

## 3.c Model Predictive Control and Dynamic Real-time Optimization of Steam Cracking Units

Francesco Rossi<sup>1</sup>, Maurizio Rovaglio<sup>2</sup>, Flavio Manenti<sup>3,\*</sup>

<sup>1</sup> Purdue University, Forney Hall of Chemical Engineering, 480 Stadium Mall Drive, West Lafayette, IN 47907-2100, United States

<sup>2</sup> Schneider-Electric S.p.A., Via Circonvallazione Est 1, 24040 Stezzano, Italy

<sup>3</sup> Politecnico di Milano, Dipartimento CMIC “Giulio Natta”, Centre for Sustainable Process Engineering Research (SuPER), Piazza Leonardo da Vinci 32, 20133 Milano, Italy

\* Corresponding Author: [flavio.manenti@polimi.it](mailto:flavio.manenti@polimi.it)

### Abstract

Unlike other parts of this book, this chapter does not deal with mathematical procedures for modeling complex reaction systems, e.g. strategies for construction/reduction of kinetic schemes, approaches to the estimation of their parameters (pre-exponential factors, activation energies, etc.), and so on. Conversely, it shows how to use accurate mathematical models of chemical processes, based on detailed kinetic schemes, for advanced control and optimization purposes. To this end, it describes two advanced model-based optimization/control strategies, called *model predictive control* and *dynamic real-time optimization*, and demonstrates the extent to which they can benefit chemical processes. The process unit, utilized to show the potential of model predictive control and dynamic real-time optimization, is a steam cracking furnace.

### Keywords

Model predictive control; Dynamic real-time optimization; Steam cracking furnace

### 3.c.1 Introduction to Model Predictive Control and Dynamic Real-time Optimization

Model predictive control (NMPC) and dynamic real-time optimization (DRTO) are advanced strategies for process optimization and control, which were first proposed by Rawlings and co-workers in the 90's (Patwardhan et al., 1990; Rawlings, 2000). Back in the day, the limited availability of computational resources and the skepticism of many industrial and academic players regarding the real efficacy of these methodologies, seemed to suggest that neither of them would ever find applications in real-life scenarios. This conclusion was ultimately incorrect. Nowadays, several companies make use of linear model predictive control strategies to operate crude oil distillation systems, polymerization processes, steam crackers, and so on. These same companies are also developing and testing dynamic real-time optimization frameworks, as they consider them viable ways of improving process performance and making strategic decisions on production schedules, based on market demand.

All of these remarks suggest that model predictive control and dynamic real-time optimization are likely to become the state of the art in process control and optimization in the near future. Therefore, it is important to discuss the rationale of NMPC and DRTO, and to report the advantages/disadvantages that these methodologies insure over conventional process optimization and control schemes.

#### 3.c.1.1 Conventional Optimization and Control Strategies for Process Management

Before discussing model predictive control and dynamic real-time optimization techniques in greater detail, it is essential to introduce some preliminary concepts about conventional process optimization and control. To this end, let us introduce four key definitions (see also *Figure 1*):

1. *Target process* (or simply *process*): a single process unit or an ensemble of interconnected process units, which must be optimized and/or controlled;
2. *Manipulated variables*: quantities that can be “freely” adjusted to modify the operating conditions of the target process (note that every manipulated variable is usually bounded, thus can take on values between a minimum and a maximum);
3. *Disturbances* or *perturbations*: quantities that affect the operating conditions of the target process and vary independently of our will;
4. *State variables* (or simply *states*): quantities that describe the operating conditions of the target process, and are univocally determined by the laws of physics for every given set of manipulated variables and disturbances.

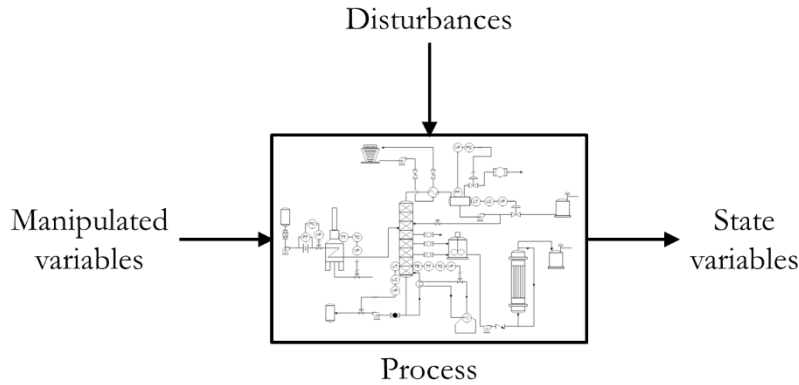


Figure 1: Schematic representation of a generic process.

Given this set of definitions, let us describe the rationale of conventional control techniques and the characteristics of common offline optimization strategies, applied to chemical processes.

Standard control systems are comprised of (interacting) feedback controllers, which force some key state variables, called the *controlled variables*, to track predefined set-point trajectories. Since every controller cannot handle more than one controlled variable at a time, the number of controllers must always be greater than or equal to the number of such variables. The rationale of these conventional controllers is simple, namely, the trajectory of a generic controlled variable is adjusted by modifying the profile of one of the manipulated variables, using predefined control laws. The latter are mathematical expressions that calculate the suggested value of the manipulated variable as a function of the deviation of the controlled variable from its set-point. The most general form of control law is shown in Eq.(1), where  $w_k$  is a generic controlled variable,  $m_i$  is the manipulated variable used to control  $w_k$ ,  $w_k^{set}$  is the set-point of  $w_k$ ,  $m_i^{bias}$  is the bias signal of the controller,  $e$  is the deviation of  $w_k$  from  $w_k^{set}$ , and  $K_C$ ,  $t_I$  and  $t_D$  are tuning parameters of the controller, called *proportional gain*, *integral time* and *derivative time*, respectively.

$$m_i = m_i^{bias} + K_C \left[ e + \frac{1}{t_I} \int_0^t e dt' + t_D \frac{de}{dt} \right] \quad (1)$$

where

$$e = (w_k - w_k^{set})$$

$$-\infty < K_C < +\infty \quad \dot{\cup} \quad t_I, t_D > 0$$

This very brief and simplified description of conventional control systems suggests some important remarks on conventional controllers:

1. They do not exploit the features of the target process (note that Eq.(1) does not include any model-based terms);

2. They can only compensate for deviations of the controlled variables from their set-points, which have already taken place;
3. They may not perform satisfactorily when the target process exhibits highly nonlinear dynamic response and the values of single manipulated variables significantly impact on the trajectories of multiple controlled variables (the latter situation implies strong interactions between different control loops);
4. They require set-point trajectories as input data.

We will go back to discussing bullets 1-3 later, now let us concentrate on bullet 4. Since conventional control systems require set-points, it is essential to calculate them so as to achieve maximum process performance (most of the times, process performance means economic profit). In the past, the optimal set-points used to come from both empirical knowledge of the process and analysis of historical data. Nowadays, it is common practice to rely on offline optimization strategies to compute the optimal set-point values. Specifically, first, a steady-state process simulation and proper optimization algorithms are employed to calculate those values of the manipulated variables that insure maximum process performance at nominal conditions (i.e. all of the disturbances are set to their nominal values). Then, these optimal manipulated variables are utilized to estimate the optimal values of the controlled variables, using the aforementioned steady-state process simulation. The latter are the optimal set-points, which are transferred to the control system. Note that this type of optimization procedure, called *real-time optimization* (RTO), is performed offline, e.g. once a day. Therefore, it differs from dynamic real-time optimization in several aspects, which will be reported in the next section.

### 3.c.1.2 Conventional Algorithms for Model Predictive Control and Dynamic Real-time Optimization

Nonlinear model predictive control (NMPC) and dynamic real-time optimization (DRTO) are model-based techniques for process control and online process optimization, respectively. Let us first introduce NMPC and then describe DRTO.

In nonlinear model predictive control, we make use of a dynamic process model and of an optimization algorithm to compute optimal trajectories of the manipulated variables of the process, which are defined as those trajectories that ensure all the controlled variables track their setpoint curves at best. Let us first explain the rationale of this control methodology, and then use it to formulate a typical NMPC algorithm.

To introduce the rationale of NMPC, assume it is possible to measure and/or estimate the current values of the process states and of the most relevant disturbances, and consider that all the setpoint profiles are known. Under these conditions, we can utilize a dynamic model to predict the future operating conditions of the process (all of the state profiles), and to calculate the overall deviation of the controlled variables from their setpoints, for every potential set of trajectories of the manipulated variables. By minimizing this overall deviation with an optimization algorithm, we can then estimate optimal profiles of the manipulated variables (the latter are the optimization “variables” of this minimization problem). Finally, this optimal control policy can be applied to the process over time.

This naïve version of NMPC is simple and intuitive, but exhibits a major flaw, namely, it does not consider that the aforementioned optimal control policy depends on the process disturbances. The latter have been considered fixed and have been set to their current values, but may vary in the future. In that case, the optimal control strategy, computed at this stage, may no longer be optimal. Therefore, optimal trajectories of the manipulated variables must be re-computed iteratively, on a time scale that is much smaller than the typical settling time of the process. In addition, only a small portion of such trajectories must be utilized for control purposes, specifically that part which lies between two of their consecutive re-evaluations. This iterative scheme, used to account for the presence of disturbances, is called *receding horizon approach*, and constitutes one of the main components of an actual NMPC algorithm.

The naïve NMPC strategy, previously described, exhibits an additional complexity. Specifically, it requires solution of an optimization problem, in which the optimization “variables” are unknown functions of time. Although the *calculus of variations* offers methods and tools to solve this category of problems, e.g. the Pontryagin's maximum principle, it is often impractical to use them in real-life

scenarios. Therefore, the trajectories of the manipulated variables are usually discretized, using parametric piecewise functions (in this chapter, we will employ piecewise constant functions). The discretization intervals, over which all these piecewise functions are defined, are called *control intervals*. Due to this simplification, the original optimization problem can be reformulated such that its optimization variables are no longer functions of time but time-invariant parameters, embedded in the piecewise functions selected. The resulting, simplified optimization problem can be solved using conventional optimization routines.

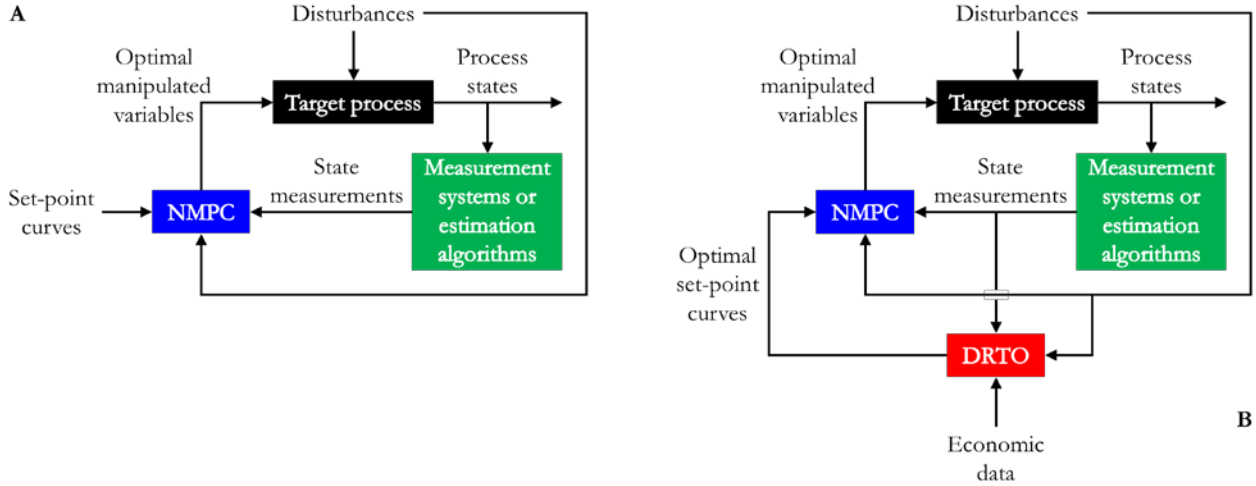


Figure 2: Interconnections among NMPC, DRTO and target process (A – set-points are fixed; B – set-points are computed by DRTO).

By applying the two suggested improvements to the naïve NMPC strategy, described previously, we obtain the conventional NMPC algorithm, reported below (Figure 2 and Figure 3 should help the reader to visualize some of these concepts):

1. [offline step] Build a dynamic model of the target process and select a suitable optimization algorithm.
2. [offline step] Define the interval over which this dynamic model is used to predict the operating conditions of the process and to calculate the deviation of the controlled variables from their set-points. This interval is referred to as *prediction horizon* (Figure 3, charts B<sub>1</sub> and B<sub>2</sub>).
3. [offline step] Choose the type of discretization grid applied to the trajectories of the manipulated variables (Figure 3, charts A<sub>1</sub> and A<sub>2</sub>), which implies selecting the number of control intervals, their length and the type of parametric piecewise functions. The product of the number of control intervals and of their length is the so-called *control horizon* (the control horizon is always shorter than the prediction horizon, thus the actual number of discretization intervals equals the number of control intervals plus one).
4. [offline/online step] Select proper set-point profiles for all the controlled variables. These set-point curves may or may not vary over consecutive NMPC steps (Figure 2, schematics A).
5. [online step] Measure and/or estimate the current values of process states and disturbances (Figure 2, schematics A).
6. [online step] Compute optimal trajectories of the manipulated variables of the process via optimization. In this computation, assume all the disturbances are fixed and set to their last known values (Figure 2, schematics A).
7. [online step] Identify the values taken on by the manipulated variables in the first control interval, and apply them to the target process (Figure 2, schematics A, and Figure 3, charts A<sub>1</sub> and A<sub>2</sub>).
8. [online step] Wait for the duration of a control interval, then go back to step 5.

Before moving on, note that we have provided a very basic and conventional algorithm for NMPC, similar to that proposed originally by Rawlings and co-workers. However, there exist more advanced strategies for nonlinear model predictive control, which prove to be more reliable and effective, e.g. *closed-*

*loop* NMPC (Van and Bosgra, 2006). The description of these methodologies goes beyond the scope of this chapter. In addition, note that the NMPC algorithm, previously reported, assumes that the computational time, needed to perform step 6, is negligible. The reader should be aware that there may be scenarios, in which we cannot introduce this simplification. Therefore, some NMPC frameworks have been proposed, which explicitly account for the time needed to complete the optimization phase. Again, the description of any of these advanced strategies goes beyond the scope of this chapter.

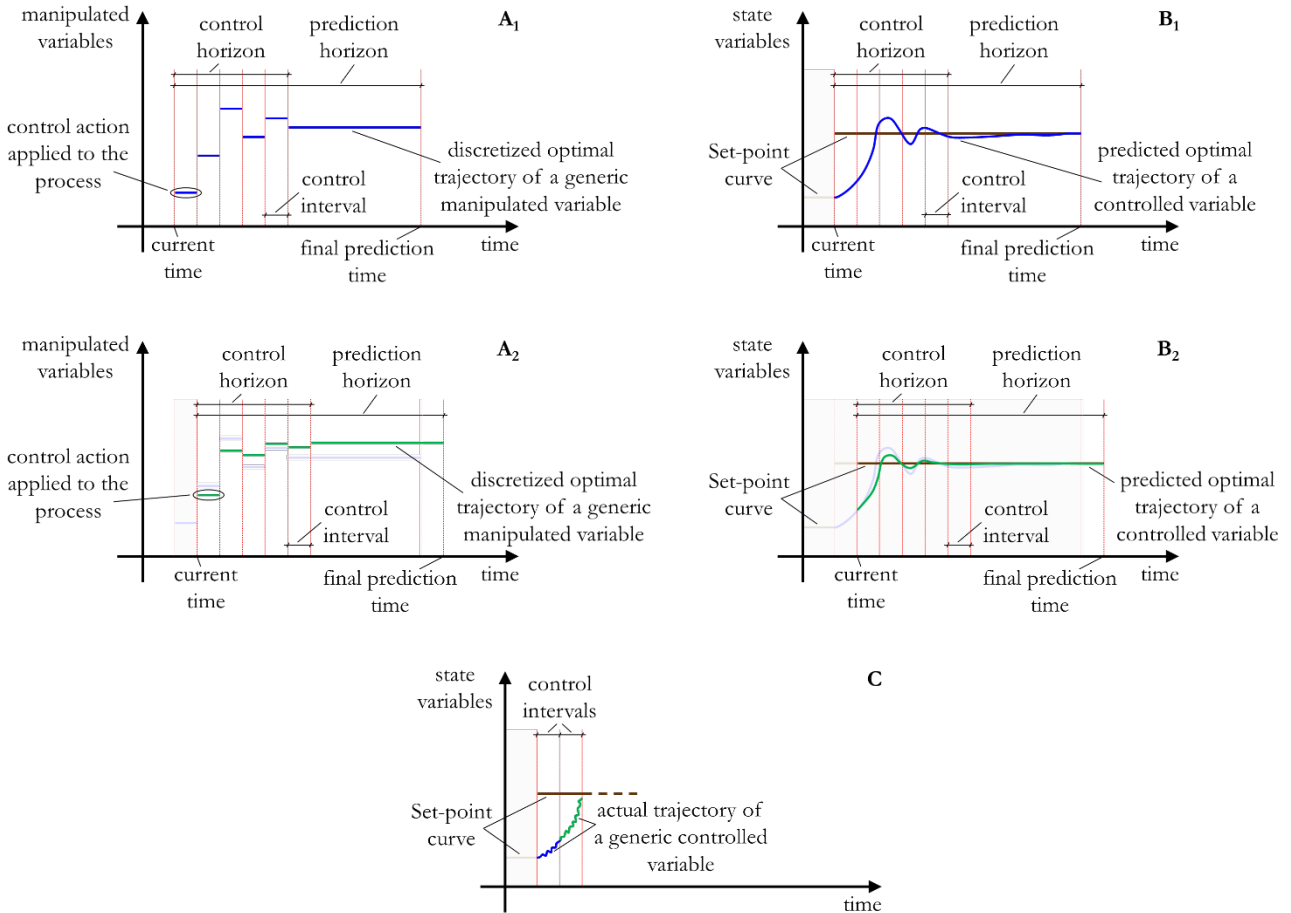


Figure 3: Graphical representation of the receding horizon approach (A<sub>1</sub>, B<sub>1</sub> and A<sub>2</sub>, B<sub>2</sub> – optimal profiles of manipulated variables and process states associated with two consecutive NMPC steps; C – closed-loop response of the process).

After describing a typical algorithm for NMPC, we can move on to DRTO. In dynamic real-time optimization, we utilize a dynamic process model and an optimization algorithm to compute optimal set-point trajectories for all the controlled variables of the process, which are defined as those trajectories that allow maximizing operating margins. This definition suggests that DRTO is conceptually identical to NMPC, except that the first yields optimal set-points while the second returns optimal control policies. In other words, DRTO suggests how to run a process so as to attain maximum profits, while NMPC allows us to enforce these suggestions at best. It is therefore important to mention that dynamic real-time optimization and model predictive control are not mutually exclusive and thus can be combined into advanced process management systems, in which the first provides the second with optimal set-point profiles over time (Figure 2, schematics B).

Given the strong similarities between NMPC and DRTO, we do not provide a detailed, step-by-step algorithm for DRTO. The reader can refer to the algorithm for NMPC, previously discussed, and make all the required extrapolations/modifications. To this end, it is important to highlight that both dynamic real-time optimization and model predictive control require iterative solution of the same type of optimization problem, where the optimization “variables” are identical but the objective functions are different. Specifically, the objective function, utilized in NMPC, measures the overall deviation of the

profiles of the controlled variables from their set-points, while that, used in DRTO, is selected as an indicator of the economic performance of the process, computed over the entire prediction horizon.

As a final remark, note that this section discusses only the most important aspects of dynamic real-time optimization and model predictive control. However, the scientific literature offers a variety of papers and books, which can serve as useful complement to this introductory material. All the interested readers are encouraged to make use of these additional references.

### 3.c.1.3 Pros and Cons of Model Predictive Control and Dynamic Real-time Optimization

To conclude this introduction to model predictive control and dynamic real-time optimization, let us comment on their advantages and disadvantages over conventional control systems and standard real-time optimization. We will first compare conventional control systems with NMPC and then standard real-time optimization with DRTO.

The principal pros and cons of NMPC over PID-based control systems are summarized in *Table 1* (symbols  $\uparrow$  and  $\downarrow$  represent pros and cons, respectively). The reader can appreciate that NMPC is more powerful than conventional control systems, namely, it insures faster and more effective disturbance rejection and set-point tracking as well as providing useful capabilities of prediction and automatic prevention of safety hazards. In addition, if we need to deal with multiple input multiple output processes, then NMPC performs satisfactorily even when there exist strong interactions among different control loops (recall that interacting control loops are such that variations in the manipulated variable of any loop significantly affect the trajectories of the controlled variables of all the other loops). On the other hand, NMPC is more computationally demanding than conventional PID-based control systems, needs an accurate dynamic model of the target process, and requires case-specific tuning procedures. Moreover, it can be applied only when we can accurately measure all the most important process disturbances. Therefore, model predictive control should be preferably applied to processes, which exhibit highly nonlinear behavior and can be accurately modeled, using small-scale/medium-scale systems of differential-algebraic equations.

*Table 1. Major advantages and downsides of NMPC and conventional PID-based control systems.*

Conventional control systems (PID-based)	Nonlinear model predictive control (NMPC)
$\uparrow$ Model-free	$\downarrow$ Model-based
$\uparrow$ Minimal computational effort required	$\downarrow$ Significant computational effort required (especially for large-scale models)
$\uparrow$ Always applicable	$\downarrow$ Applicable only in the presence of accurate process models
$\uparrow$ Disturbance measurements not required	$\downarrow$ Disturbance measurements required
$\downarrow$ Slower disturbance rejection	$\uparrow$ Faster disturbance rejection (especially for systems with strongly nonlinear response)
$\downarrow$ Slower set-point tracking	$\uparrow$ Faster set-point tracking (especially for systems with strongly nonlinear response)
$\downarrow$ Performance strongly affected by potential interactions between control loops (RGA <sup>a</sup> not diagonally dominant)	$\uparrow$ Performance weakly affected by potential interactions between control loops
$\downarrow$ Unable to predict/prevent safety hazards	$\uparrow$ Capable of predicting/preventing safety hazards
$\uparrow$ Easy controller tuning (standard procedures available)	$\downarrow$ Complex controller tuning (problem-specific procedures required)

<sup>a</sup> RGA stands for *relative gain array*

After comparing NMPC with conventional control systems, we can comment on the pros and cons of dynamic real-time optimization over standard real-time optimization (see *Table 2*, where symbols  $\uparrow$  and  $\downarrow$  represent pros and cons, respectively). DRTO appears to be more flexible and powerful than RTO because it allows optimization of the transient of the target process between initial and optimal operating conditions, thus ensuring higher economic benefits than conventional real-time optimization. Moreover, it can serve to optimize shutdown and startup operations, and can automatically mitigate the economic loss, arising from emergency shutdown procedures. On the other hand, it is more computationally demanding than RTO, needs accurate dynamic models of the process, and requires case-specific tuning. Consequently, DRTO should be preferably applied to processes, which produce high value-added products, and can be accurately modeled, using small-scale/medium-scale systems of differential-algebraic equations.

*Table 2. Major advantages and downsides of DRTO and RTO.*

Real-time optimization (RTO)	Dynamic real-time optimization (DRTO)
[ $\uparrow$ ] Model-based (steady-state models required)	[ $\downarrow$ ] Model-based (dynamic models required)
[ $\uparrow$ ] Affordable computational effort required	[ $\downarrow$ ] Significant computational effort required (especially for large-scale models)
[ $\downarrow$ ] Unable to optimize the process transient between initial and optimal operating conditions	[ $\uparrow$ ] Capable of optimizing the process transient between initial and optimal operating conditions
[ $\downarrow$ ] Unable to optimize startup/shutdown	[ $\uparrow$ ] Capable of optimizing startup/shutdown
[ $\downarrow$ ] Useless in emergency shutdown circumstances	[ $\uparrow$ ] Capable of minimizing economic losses in emergency shutdown circumstances
[ $\uparrow$ ] No tuning required	[ $\downarrow$ ] Problem-specific tuning procedures required

As a final remark, note that both DRTO and NMPC show significant advantages over their conventional counterparts, i.e. RTO and PID-based control systems, but also a few downsides. However, many of these disadvantages, e.g. high computational cost and the need for accurate models, may be overcome in the near future. Therefore, we can envision that DRTO and NMPC will soon become the state of the art for process optimization and control.

### **3.c.2 Mathematical Formulation and Numerical Solution of Model Predictive Control and Dynamic Real-time Optimization Problems**

After describing model predictive control and dynamic real-time optimization in qualitative terms, we can discuss the most important aspect of their mathematical formulation, namely, the structure of the optimization problem, solved in the optimization phase (NMPC/DRTO algorithm – step 6). Note that we will not provide any details regarding *state estimation techniques* (Simon, 2006), used to approximate the trajectories of the unmeasured process states (NMPC/DRTO algorithm – step 5), as this goes beyond the scope of this chapter. The reader can find additional information on this topic in scientific papers and dedicated books. In addition to this, we will also report useful information on the two numerical solution procedures, usually adopted to solve the aforementioned optimization problems, i.e. the *sequential approach* and the *simultaneous approach* (these two methodologies are typically referred to as *direct approaches*). We will describe the first in greater detail and only outline the second, as all the application studies, reported in the following section, rely on the sequential approach. Again, we will not elaborate on other types of numerical solution strategies, e.g. all the *indirect approaches* based on variational calculus, but encourage the reader to refer to scientific papers and dedicated books to gain additional insight into them.

### 3.c.2.1 Mathematical Formulation of the Optimization Phase

The optimization problem, solved in the optimization phase of every NMPC/DRTO algorithm, serves to compute optimal profiles of the manipulated variables of the target process via minimization of an appropriate objective function. The latter differs between NMPC and DRTO but always includes integral terms, which depend on the trajectories of some process states, and/or end-point terms, which depend on the values taken on by some state variables at the end of the prediction horizon. Therefore, we must minimize this function subject to the dynamic model of the process, which represents the first set of constraints, embedded in the aforementioned optimization problem. The second set of constraints, which must be enforced, is composed of the lower/upper bounds on process states and manipulated variables, which insure that all of these quantities lie within the allowable operating region of the process. Finally, we may need to impose additional equality and/or inequality constraints to enforce other case-specific restrictions. Note that we will not consider the latter types of restrictions, as they are often unnecessary and, if present, they may be eliminated using proper reformulation techniques (*penalty function method* and/or addition of new state variables to the process model). If necessary, this assumption can be easily removed by the reader.

On the basis of this qualitative information, we can finally give a rigorous mathematical formulation to the optimization problem, solved in the optimization phase of conventional algorithms for NMPC and DRTO (see Eqs.(2)-(5)). More specifically, Eq.(2), Eq.(3) and Eq.(4) introduce some preliminary definitions, while Eq.(5) shows the actual formulation of the optimization problem. The nomenclature, used in these four equations, is explained below (*Figure 4* can help the reader to visualize the graphical significance of some of the following definitions):

- $\mathbf{m}$ ,  $\mathbf{w}$  and  $\mathbf{d}$  represent the manipulated variables, state variables and disturbances of the target process;
- $m_i$ ,  $w_k$  and  $d_l$  are generic entries of vectors  $\mathbf{m}$ ,  $\mathbf{w}$  and  $\mathbf{d}$ ;
- $N_M$ ,  $N_S$  and  $N_D$  stand for the number of manipulated variables, state variables and disturbances;
- $Dt_p$ ,  $t_*$  and  $t_p$  indicate the prediction horizon, the current time and the prediction time ( $t_p$  simply equals  $t_* + Dt_p$ );
- $J$  is the objective function of the optimization problem, while  $J_s$  and  $J_t$  are its two principal components, called *stage cost* and *terminal cost*;
- $\mathbf{I}_M$  is the coefficient matrix of the dynamic model of the process, which is supposed to be either a system of differential equations (ODE) or a system of differential-algebraic equations (DAE);
- $\mathbf{h}$  is a vector field, which represents the right-hand side functions of the process model;
- Superscript  $*$  stands for “computed/measured/estimated at  $t_*$ ”, while superscripts *MIN* and *MAX* stand for “minimum” and “maximum”;

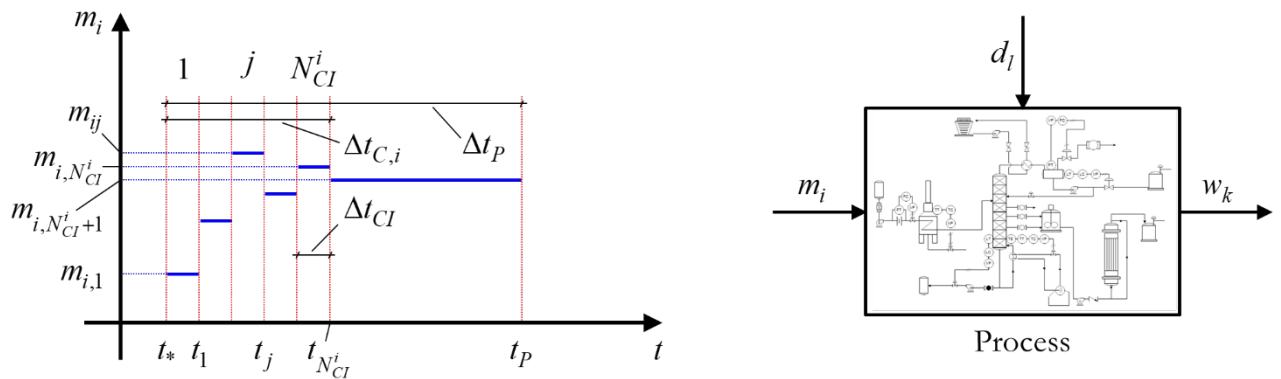


Figure 4: Logical connections between key quantities included in Eqs.(2)-(9), the target process and the discretization grid applied to the trajectories of all the manipulated variables.



Given the simple structure of Eqs.(2)-(4), we do not need to comment on them. On the other hand, some additional remarks on Eq.(5) are in order. The objective function of the optimization problem, reported in Eq.(5), is composed of an integral term and of an end-point term (this confirms the statements reported at the beginning of this section). The first is proportional to the integral of the stage cost over the prediction horizon, and serves either to compute the overall deviation between the trajectories of the controlled variables and their set-points (NMPC) or to estimate the economic performance of the process over the prediction horizon (DRTO). The second equals the terminal cost, and is often unnecessary but, if present, it is used to force some controlled variables to assume specific values at the end of the prediction horizon (NMPC) (this may help to mitigate the effects of *model mismatch*). Therefore, NMPC algorithms define the stage cost as a nonnegative function, which depends on the differences between controlled variables and set-points at any generic point in time (clearly, the stage cost must increase as such differences increase). On the other hand, DRTO strategies usually select the stage cost as the opposite of process income at any generic point in time, which can be often calculated as a linear combination of key process states and manipulated variables, multiplied by proper economic weights. The definition of terminal cost is guided by ideas similar to those used to choose the formulation of stage cost. Indeed, NMPC frameworks set it to a scaled version of the stage cost, computed at the end of the prediction horizon, while DRTO methods typically set it to zero. Examples of conventional stage cost and terminal cost functions, utilized in nonlinear model predictive control and dynamic real-time optimization, are shown in Eq.(6) and Eq.(7), respectively. In these two equations,  $w_k$  and  $j_k$  are user-supplied scale factors,  $c_k^W$  and  $c_i^M$  are economic weights, and superscripts **SP**, **P** and **TV** stand for “set-point”, “evaluated at  $t_P$ ” and “desired end-point value”, respectively.

$$\mathbf{m} = \begin{bmatrix} m_1 & \dots & m_i & \dots & m_{N_M} \end{bmatrix} \quad (2)$$

$$\mathbf{w} = \begin{bmatrix} w_1 & \dots & w_k & \dots & w_{N_S} \end{bmatrix} \quad (3)$$

$$\mathbf{d} = \begin{bmatrix} d_1 & \dots & d_i & \dots & d_{N_D} \end{bmatrix} \quad (4)$$

$$\begin{aligned} & \text{Min}_{\mathbf{m}, \mathbf{w}} \int_{t_0}^{t_P} J_s(\mathbf{m}, \mathbf{w}, \mathbf{d}^*) dt + J_t(\mathbf{m}, \mathbf{w}, \mathbf{d}^*) \\ & \text{s.t.} \\ & \mathbf{I}_M \frac{d\mathbf{w}}{dt} = \mathbf{h}(\mathbf{m}, \mathbf{w}, \mathbf{d}^*) \\ & \mathbf{w}(t_*) = \mathbf{w}^* \\ & \mathbf{m} \hat{=} \begin{bmatrix} m^{MIN} & m^{MAX} \end{bmatrix} \quad \mathbf{w} \hat{=} \begin{bmatrix} w^{MIN} & w^{MAX} \end{bmatrix} \end{aligned} \quad (5)$$

Before addressing the numerical solution of Eq.(5), there is one last aspect of Eq.(6) that is worth mentioning. Specifically, the formulation of stage cost for NMPC algorithms usually includes a so-called *anti-ringing term*, indicated by the acronym **AR**, which estimates the degree of smoothness of the profiles of the manipulated variables of the process (the higher this level of smoothness, the smaller the value of the anti-ringing term). This term insures that optimal trajectories of the manipulated variables, obtained by solving Eq.(5), always exhibit oscillations with low frequency and small amplitude, thus do not induce excessive mechanical stress on process units and valves. There exist multiple equivalent formulations of the anti-ringing term, most of which depend on manipulated variable derivatives. However, we will not report any details on this topic, as it goes beyond the scope of this chapter.

$$\begin{aligned} J_s(\mathbf{m}, \mathbf{w}, \mathbf{d}^*) &= \sum_{k=1}^{N_s} w_k (w_k - w_k^{SP})^2 + AR(\mathbf{m}) \\ J_t(\mathbf{m}, \mathbf{w}, \mathbf{d}^*) &= \sum_{k=1}^{N_s} j_k (w_k^P - w_k^{TV})^2 \end{aligned} \quad (6)$$

$$\begin{aligned} J_s(\mathbf{m}, \mathbf{w}, \mathbf{d}^*) &= \sum_{k=1}^{N_s} c_k^W w_k + \sum_{i=1}^{N_M} c_i^M m_i \\ J_t(\mathbf{m}, \mathbf{w}, \mathbf{d}^*) &= 0 \end{aligned} \quad (7)$$

As a final remark, note that this section does not describe the mathematical formulation of model predictive control and dynamic real-time optimization algorithms in a very rigorous fashion. Conversely, it only focuses on some essential aspects, and devotes more effort to explaining the physical significance of formulas and equations. For additional information on this topic, the reader is invited to refer to dedicated books.

### 3.c.2.2 Sequential and Simultaneous Approaches to the Solution of the Optimization Phase

The numerical solution of the optimization problem, shown in Eq.(5), is quite challenging due to some of its features, namely, the presence of optimization “variables”, which are unknown functions of time, and the presence of differential-algebraic constraints, which are not supported as is by any optimization package. The first issue has been discussed earlier, and has been solved by discretizing the trajectories of the manipulated variables of the process, using proper parametric piecewise functions (*Figure 4* displays the piecewise constant discretization strategy, adopted in this chapter). This way, the optimization variables, included in Eq.(5), become time-invariant parameters of the piecewise constant functions selected (i.e. the values taken on by the manipulated variables in every discretization interval). The second issue requires reformulation of Eq.(5) so as to either eliminate the differential-algebraic constraints or convert them into tractable algebraic restrictions. The sequential approach relies on the first strategy while the simultaneous approach utilizes the second. Note that the most relevant difference between sequential and simultaneous solution methods is indeed the rationale used to handle differential-algebraic constraints. After this general introduction, let us first describe the sequential approach, then introduce the simultaneous approach and, finally, compare these two numerical solution strategies.

The sequential approach relies on the following rationale (refer to *Figure 5* and Manenti (2011) for additional details):

1. Elimination of all the differential-algebraic constraints from Eq.(5) by embedding them into the objective function of the optimization problem;
2. Conversion of all the lower/upper bounds on the process states into a penalty term, which is added to the aforementioned objective function;
3. Solution of the resulting small-scale bound-constrained optimization problem with conventional optimization algorithms.

The mathematical manipulations, described in bullets 1 and 2, lead to some important consequences. More specifically, every iteration of the optimization procedure, performed in step 3, requires a certain number of evaluations of the objective function of the optimization problem, which in turn involves integration of the dynamic model of the process. Therefore, the sequential approach cannot efficiently deal with large-scale process models, as the latter may not be solved quickly. In addition to this first downside, note that all the lower/upper bounds on the process states are enforced through a penalty term (see step 2), which makes it difficult to guarantee complete absence of bound violations. Consequently, the sequential approach should be avoided in the presence of inviolable bound constraints on the process states. On the other hand, the optimization problem, which must be solved in step 3, has very interesting properties, namely, it is small-scale and bound-constrained. Thus, we can easily solve it,

using derivative-free optimization packages. This implies that the sequential approach may allow direct solution of those instances of Eq.(5), which exhibit discontinuities in the process model and/or in the stage cost and/or in the terminal cost. To summarize, the sequential approach shows some downsides as well as some advantages, which make it more suitable for application scenarios, where small-scale to medium-scale process models are used, and the target process may exhibit discontinuous behavior (e.g. multi-stage batch processes).

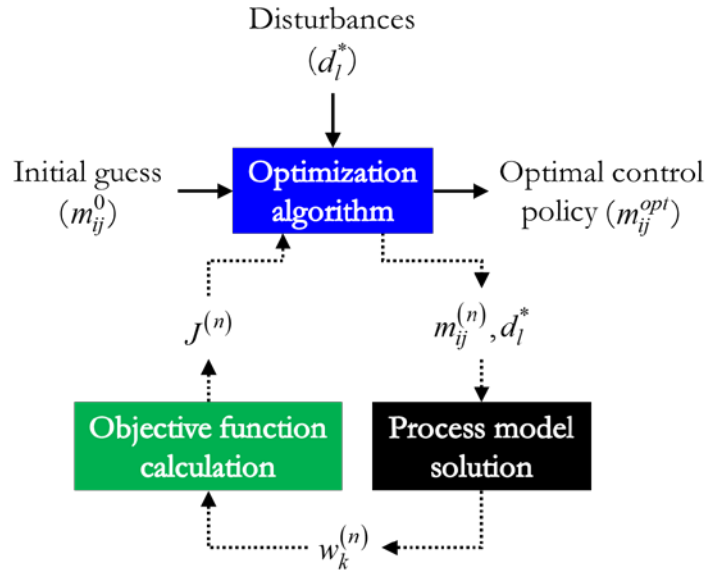


Figure 5: Rationale of the sequential approach ( $m_{ij}$  indicate the discretized profiles of the manipulated variables (Figure 4); superscripts 0, opt and (n) stand for “initial guess”, “optimal value” and “at iteration n”).

Before moving on to the simultaneous approach, we report a possible formulation of the penalty term, mentioned in step 2, and the structure of the optimization problem, which must be solved in step 3. This information is reported in Eq.(8) and Eq.(9), where  $m_{ij}$  indicate the discretized profiles of the manipulated variables (Figure 4),  $Y_p$  is the aforementioned penalty term,  $J_p$  is the objective function of the aforementioned optimization problem, and  $\alpha$  is a user-defined scale factor.

$$Y_p(m_{ij}, \mathbf{w}, \mathbf{d}^*) = \alpha \left| \sum_{k=1}^{N_s} \frac{1}{w_k^{MAX} - w_k^{MIN}} \int_0^{t_p} m_k |w_k - w_k^{MIN}| + l_k |w_k - w_k^{MAX}| dt \right| \quad (8)$$

where

$$m_k = \begin{cases} 1 & \text{if } w_k < w_k^{MIN} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$l_k = \begin{cases} 1 & \text{if } w_k > w_k^{MAX} \\ 0 & \text{otherwise} \end{cases}$$

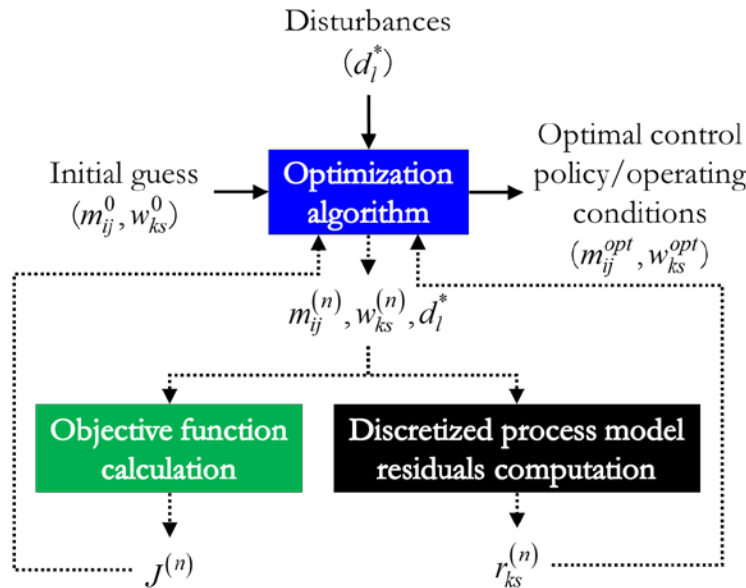
$$\begin{aligned} & \text{Min}_{m_{ij}} \{ J(m_{ij}, \mathbf{w}, \mathbf{d}^*) + Y_p(m_{ij}, \mathbf{w}, \mathbf{d}^*) \} \\ & \text{st.} \\ & m_{ij} \in [m_i^{MIN}, m_i^{MAX}] \end{aligned} \quad (9)$$

# process model embedded in the objective function

Some additional remarks on these two equations may prove useful for the reader. First, note that the penalty term, reported in Eq.(8), is defined as a summation of integrals of absolute values of nondimensionalized lower/upper bound violations, multiplied by the absolute value of the objective function of Eq.(5) and by a scale factor, called *penalty coefficient*. This mathematical formulation is very smart because it allows simple identification of nearly optimal values of the penalty coefficient, which are almost problem-independent. This is a significant attainment, as penalty coefficients may often be difficult to tune. Second, consider that Eq.(8) includes inherently non-smooth terms, thus is not suitable for sequential NMPC/DRTO implementations that do not rely on derivative-free optimizers. In the latter case, smooth penalty terms should be employed. Finally, note that Eq.(8) and Eq.(9) confirm that the sequential approach leads to optimization problems, which are easy to implement and solve. The reader may argue that the objective function of the minimization problem, shown in Eq.(8), is difficult to compute because it includes integral terms. However, we can easily eliminate these terms by adding additional state variables to the process model.

The simultaneous approach relies on a rationale completely different than that of the sequential approach. Specifically, it involves the following steps (refer to *Figure 6* and to Lang and Biegler (2007) for additional information):

1. Discretization of the differential-algebraic constraints, embedded in Eq.(5), using appropriate numerical techniques (usually we employ *orthogonal collocation on finite elements* or other collocation methods);
2. Solution of the resulting large-scale constrained optimization problem with conventional optimization algorithms.



*Figure 6: Rationale of the simultaneous approach ( $m_{ij}$  and  $w_{ks}$  indicate the discretized profiles of manipulated variables (Figure 4) and process states;  $r_{ks}$  represent the residuals of the algebraic equations yielded by discretization of the process model; superscripts  $0$ ,  $opt$  and  $(n)$  stand for “initial guess”, “optimal value” and “at iteration  $n$ ”).*

The mathematical manipulations, described in bullet 1, lead to some relevant consequences. More specifically, the discretization of the dynamic model of the process implies discretization of the profiles of the process states, which is performed via parametric piecewise functions. Note that the parameters of these functions become new optimization variables. Also, consider that we need accurate estimates of all the state profiles, such that we can enforce their bound constraints. Therefore, the optimization problem, which must be solved in step 2, will be large-scale. As a consequence, the choice of the initial guess of all the optimization variables and the formulation of process model, stage cost and terminal cost become crucial, because the presence of discontinuities in any of the latter terms and/or a bad initial

guess are likely to cause serious convergence issues. On the other hand, the numerical solution of the aforementioned optimization problem usually requires less computational time than that of its counterpart, yielded by the sequential approach, especially when we deal with large-scale process models. Therefore, we should always prefer the simultaneous approach in the latter circumstance but we should also be aware of the proper implementation techniques, needed to avoid potential convergence problems.

After describing both the sequential and the simultaneous approaches, we end this section with *Table 3*. In this table, we summarize the main differences between these two numerical solution strategies as well as the advantages/disadvantages in utilizing one or the other. The reader can appreciate that the sequential approach is simpler to understand and apply, yields optimization problems that are easy to solve, and can cope with discontinuities in process model, stage cost and terminal cost. However, it is not applicable to large-scale process models, and does not allow easy handling of equality/inequality constraints. On the other hand, the simultaneous approach can handle all types of constraints and can be applied to any type of process model, independently of its size. However, it requires significant implementation effort, yields optimization problems that may not be easy to solve, and cannot cope with discontinuities in process model, stage cost and terminal cost.

*Table 3. Major advantages and downsides of sequential and simultaneous solution strategies.*

Sequential approach	Simultaneous approach
[↑] Low implementation effort required	[↓] Significant implementation effort required
[↓] Less computationally efficient (especially for large-scale models)	[↑] More computationally efficient (especially for large-scale models)
[↓] Applicable only to small-scale/medium-scale models	[↑] Always applicable
[↑] Robust (absence of convergence issues in the optimization phase but less accurate problem solutions)	[↓] Efficient (more accurate problem solutions but potential presence of convergence issues in the optimization phase)
[↓] Equality/inequality constraints allowed only upon proper problem reformulation	[↑] Every type of constraint allowed
[↑] Discontinuities allowed (only upon use of derivative-free optimization algorithms)	[↓] No discontinuities allowed

To conclude, both the sequential and the simultaneous approaches exhibit pros and cons. This is the reason why a new numerical solution strategy, called *quasi-sequential approach* (Hong et al., 2006), has been recently proposed. This strategy combines the sequential and the simultaneous approaches, retains most of their positive aspects, and rejects their most significant drawbacks.

### **3.c.3 Latest Developments in Model Predictive Control and Dynamic Real-time Optimization**

Most conventional algorithms for model predictive control and dynamic real-time optimization exhibit two major downsides, namely, the need for accurate process models and the absence of general tuning procedures. Similarly, most standard strategies for solving NMPC/DRTO problems may be too computationally intensive, especially when it is vital to utilize large-scale process models. Therefore, the process systems engineering research community keeps working on novel algorithms for NMPC/DRTO (Dones et al., 2010; Dua et al., 2008; Logist et al., 2012; Rossi et al., 2014) as well as innovative numerical solution strategies for this class of problems (Bansal et al., 2003).

Unfortunately, no general tuning approach for model predictive control and dynamic real-time optimization has been found yet. Although some methods, which rely on *multi-objective optimization*, have been proposed (Vallerio et al., 2014), this is still an open issue. Conversely, recent scientific papers have introduced *robust NMPC/DRTO* algorithms that no longer need accurate process models (Gros et al.,

2006; Li et al., 2008; Lucia et al., 2014; Nagy and Braatz, 2007; Rossi et al., 2016). These algorithms return conservative results, which guarantee absence of operational constraint violations, even though the predictions of the process model do not match the real dynamic response of the process.

Regarding new and more efficient numerical solution strategies for NMPC/DRTO problems, recent manuscripts have introduced the quasi-sequential approach, which has been outlined at the end of the previous section, and the so-called *advanced step NMPC/DRTO* (Zavala et al., 2008; Zavala and Biegler, 2009). The latter is a new concept of model predictive control and dynamic real-time optimization, which decouples the optimization phase from the application of optimal control actions to the process. This way, the time between any two consecutive control actions is no longer bound to the computational time, needed to carry out the optimization phase. This device allows broadening the application domain of model predictive control and dynamic real-time optimization strategies.

In view of these recent developments, we can confirm that most of the typical downsides of NMPC and DRTO algorithms are being rapidly resolved. Therefore, we can expect these model-based control and optimization strategies to take over their conventional counterparts in the near future, thus generating significant benefits in terms of process performance and/or safety.

### **3.c.4 Dynamic Real-time Optimization of Steam Cracking Units**

In order to show the most important benefits, offered by NMPC/DRTO strategies over conventional offline optimization and process control approaches, we address the optimization of a well-known type of process unit, i.e. a steam cracker. More specifically, we compare the economic performance, achieved by conventional RTO, with that, yielded by DRTO. In both scenarios, the regulatory control system is composed of PI and PID controllers, which receive optimal set-points from RTO/DRTO.

Since we make use of conventional DRTO algorithms, it is essential for the dynamic model of the steam cracking unit to be accurate and reliable. Therefore, the steam cracker model, employed in this study, combines mass, energy and momentum conservation equations with detailed kinetic schemes, provided by dedicated commercial packages. The size of this dynamic model is remarkable, but not large enough to require application of simultaneous implementation strategies. In addition to this, the use of commercial packages would make it difficult to use the simultaneous approach, as the resulting model is not fully *equation-oriented*. For these reasons, we have pursued a sequential implementation strategy.

#### **3.c.4.1 Integration of Existing Software Packages for Modeling and (Dynamic) Real-time Optimization Purposes**

The steam cracker model, the RTO platform and the DRTO framework (*Figure 7*), utilized in this case study, result from the combination of different commercial simulation packages and C++ numerical libraries, developed at Politecnico di Milano. These packages and libraries are listed below:

- DynSim™ (Schneider-Electric): widespread commercial dynamic simulation suite, used by major engineering, procurement and construction (EPC) societies for purposes of design, validation and control of chemical processes;
- ROMeo™ (Schneider-Electric): package for dynamic simulation and real-time optimization, which represents the industrial state of the art in this field;
- RobOpt™ and BzzMath (Politecnico di Milano): comprehensive set of numerical tools, including robust derivative-free optimizers and numerical solvers for systems of differential and differential-algebraic equations (Buzzi-Ferraris and Manenti, 2012, 2013, 2014; Manenti et al., 2009);
- SPYRO® (Pyrotec-Technip): well-established program for the simulation of the coils of steam cracking furnaces.

The integration of these four tools into a full-fledged DRTO platform is facilitated by object-oriented programming, which allows embedding both the efficient ODE/DAE solvers of the BzzMath library and the commercial package SPYRO® into DynSim™. After this preliminary step, DynSim™ can be used to build the required dynamic process model, which is then imported into ROMeo™. The latter allows us to solve DRTO problems, using the same optimization tools designed for RTO.

It is important to stress that the DRTO framework, shown in Figure 7, would not exist without the ODE/DAE solvers of BzzMath, as these integrators allow reliable solution of highly nonlinear systems at a very affordable computational cost. This makes it possible for the aforementioned DRTO framework to meet the tight computational efficiency requirements, needed for online applications.

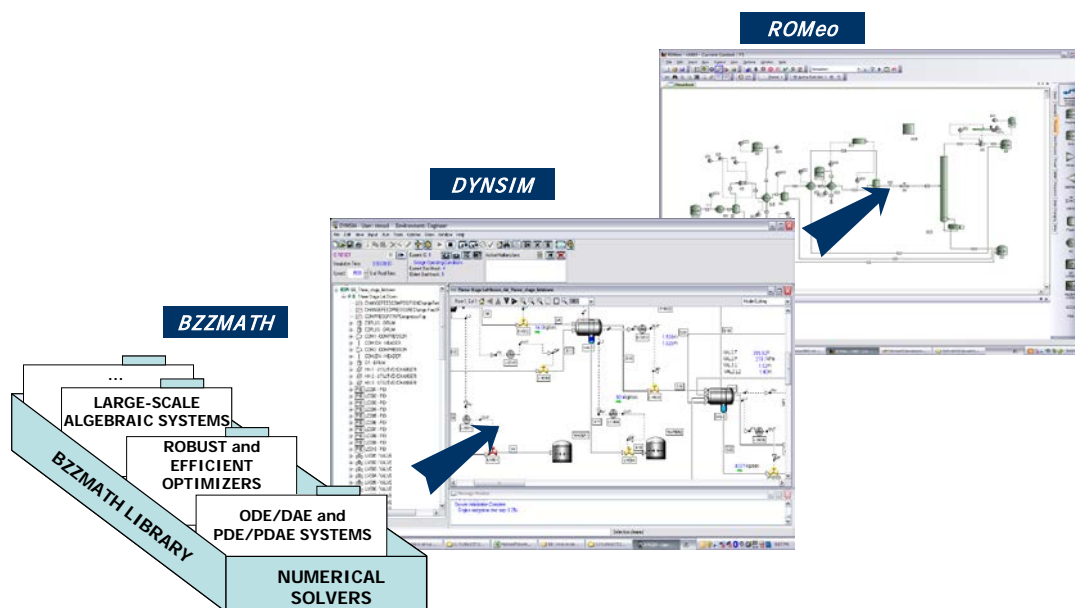


Figure 7. Rationale of the dynamic real-time optimization framework based on DynSim™, ROMeo™, BzzMath and SPYRO®.

### 3.c.4.2 Model of the Radiant Section of the Steam Cracking Furnace

Steam cracking furnaces are process units, devoted to producing ethylene and propylene from a stream of light hydrocarbons (e.g. light naphtha) and steam. These massive pieces of equipment are comprised of two key components, namely, a coil bundle, where the cracking reactions take place, and a furnace, which provides heat to the coils (recall that cracking reactions are endothermic). The coils are always placed in the radiant section of the furnace, where the temperature is higher than 800°C and the heat transfer efficiency is optimal, because all of the cracking reactions are very endothermic, thus require significant amount of energy to take place. Moreover, since steam cracking is a very fast reaction process (Dente et al., 1992), the typical residence time of the mixture of hydrocarbons and steam in the coils is in the order of some milliseconds. This very short contact time suffices to convert hydrocarbons into ethylene and propylene but is not long enough to allow production of substantial amounts of coke. On the other hand, the convective section of the furnace contains gas-gas heat exchangers, which are utilized to pre-heat the mixture of hydrocarbons and steam, fed to the coils. This pre-heating procedure serves to increase the energy efficiency of the process, thus reducing fuel consumption and operating costs (recall that the purchase of raw material and the consumption of fuel are the two major sources of cost in steam cracking units). For the convenience of the reader, the schematics of a typical steam cracking furnace are shown in Figure 8.

The aforementioned description of a typical steam cracker shows the intrinsic complexity of this type of process unit, which is comprised of multiple highly interacting sections. For the sake of simplicity, this case study will only address the optimization of the radiant section of a steam cracking furnace. To this end, let us report additional details on this specific section of the furnace. Note that the residence time of the feed in the coils is usually kept fixed, thus the steam cracker performance primarily depends on the coil outlet temperature (COT). This is because the COT directly affects the so-called *cracking severity* (ratio of the flowrates of any two products of interest at the coils outlet), which in turn defines the economic performance of the process. Therefore, the COT is certainly the most important controlled variable of the radiant section of the steam cracking furnace. As the reader knows, every controlled variable must be paired up with a manipulated variable, thus we need to select a process variable suitable

for this purpose. It is evident that the most sensible choice of manipulated variable is the fuel flowrate, as this quantity directly affects the COT. Thus, we choose to control the COT via a temperature controller, which manipulates the fuel flowrate, as shown in *Figure 9*. This temperature controller also transmits the current value of the fuel flowrate to a flow controller, which adjusts the flow of combustion air accordingly (*Figure 9*). Finally, the set-point values of coil outlet temperature, sent to the aforementioned temperature controller, are calculated by either RTO or DRTO.

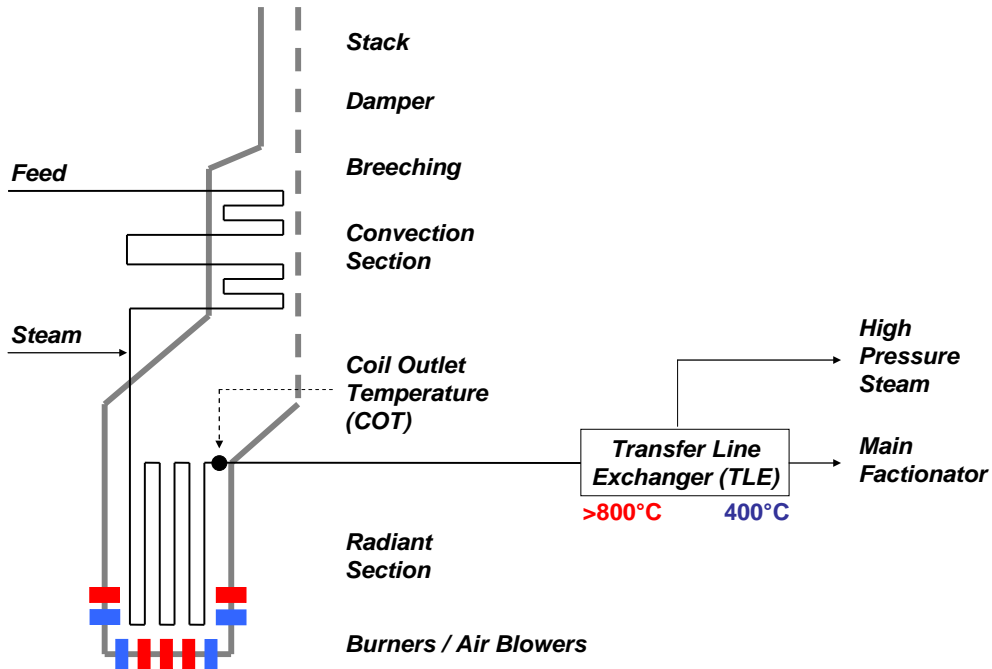


Figure 8. Half plan slice of a steam cracking furnace.

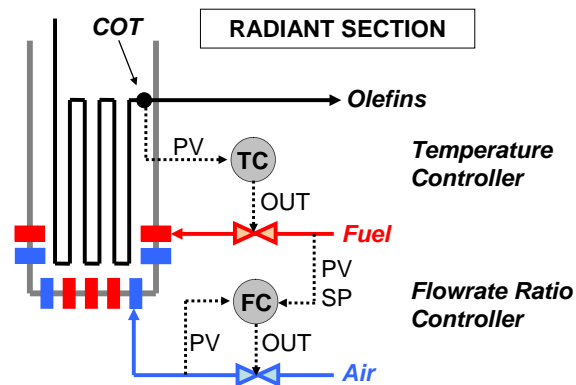


Figure 9. Schematics of the radiant section of the steam cracker complemented by the control scheme (PV stands for process variable; OUT means controller output; SP indicates a set-point value).

Before reporting some numerical results, let us offer two final pieces of information. First, we outline how DynSim™ has been used to construct the dynamic model of the radiant section of the steam cracker. This model is composed of two principal components, i.e. the coil bundle model and the radiant section model. The first is offered by SPYRO®, while the second has been implemented in C++ and imported into DynSim™ as custom unit operation. The integration of these two sub-models has been performed within DynSim™, which implies accounting for all the energy fluxes, exchanged between them, through DynSim™ heat streams. Second, we report which stage cost and terminal cost functions have been used in the DRTO platform. The terminal cost has been set to zero because we do not need to consider any end-point economic term. On the other hand, the stage cost is set to the income, generated by the steam



cracker at any generic time point, which consists of a linear combination of the propylene, ethylene, fuel, and feed flowrates. The weighting coefficients of this linear combination are the product prices, fuel cost and raw materials cost, respectively. Clearly, this definition of stage cost introduces very crude assumptions, e.g. no downstream purification costs, but is sufficiently accurate for the scope of this chapter.

### 3.c.4.3 Results of the Application of RTO and DRTO to the Radiant Section of the Steam Cracking Furnace

The most important results of the application of RTO and DRTO to the radiant section of the steam cracker, analyzed in this case study, are reported in *Figure 10*. This collection of charts shows the variations in cracking severity and fuel flowrate, induced by both real-time optimization and dynamic real-time optimization, in response to modifications in the prices of ethylene and propylene.

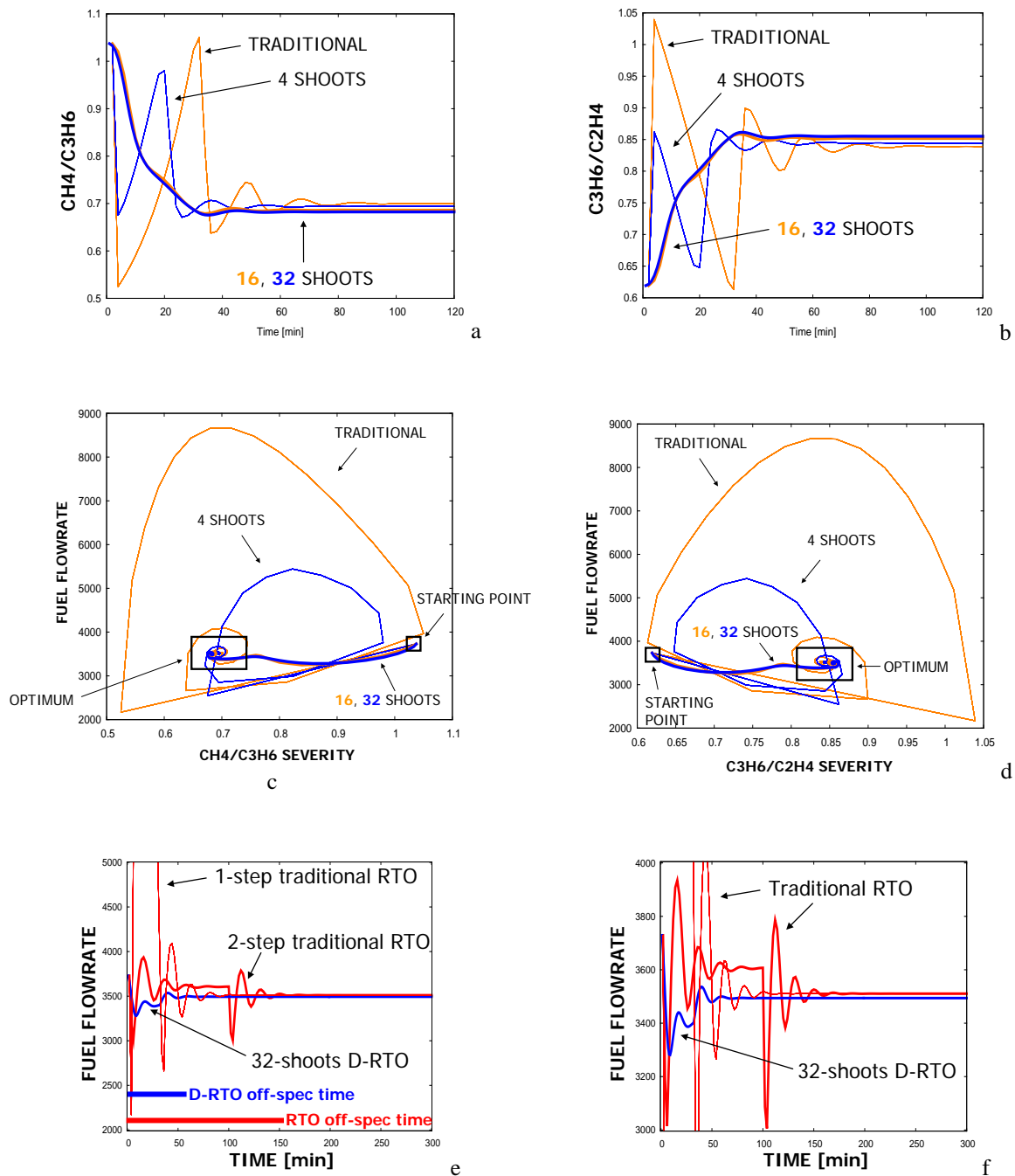


Figure 10. Results of the application of RTO and DRTO to the steam cracker (a, b – optimal trajectories of the  $\text{CH}_4 - \text{C}_3\text{H}_6$  and  $\text{C}_3\text{H}_6 - \text{C}_2\text{H}_4$  cracking severities; c, d – optimal transition between initial and optimal operating conditions achieved by RTO and DRTO; e, f – optimal trajectories of the fuel flowrate yielded by RTO and DRTO).

It is evident that both RTO and DRTO yield the same optimal steady-state conditions. However, RTO does not provide any information on how to transition from the initial operating conditions to these optimal steady states, thus this decision is left to the control room operators and to the control system. Given the highly nonlinear behavior of the steam cracking unit, a single step transition leads to very long settling times as well as extreme fluctuations in the profiles of both fuel flowrate and cracking severities. On the other hand, a two-step transition allows mitigating the oscillations in the trajectories of fuel flowrate but leads to even longer settling times. Conversely, DRTO automatically suggests how to transition from initial operating conditions to optimal steady states, thus guides the regulatory control system in this challenging operation with no need for human intervention. As the number of steps, performed in this transition, increases from 4 to 32, we observe a progressive reduction of the fluctuations in the profiles of both cracking severity and fuel flow together with a significant reduction in settling time. The latter consequence is particularly relevant, as it implies substantial reduction in the amount of off spec products, produced by the steam cracker.

In conclusion, this brief case study suggests that DRTO is significantly more effective and safer than RTO, as it insures higher economic benefits and helps to prevent safety hazards (the steam cracker experiences dynamic instability issues, when it is optimized by RTO). In addition, it can also relieve control room operators of some of their duties, thus increasing the level of automation in chemical processes. Although these considerations have been shown through a simple example, they apply to many different application scenarios, as the scientific literature has demonstrated over the last two decades (Balasubramhanya and Doyle, 1997; Busch et al., 2007; Gopalakrishnan and Biegler, 2013; Mahadevan et al., 2001; Manenti and Rovaglio, 2008; Pierucci et al., 1996; Rossi et al., 2015, 2016, 2017; Viganò et al., 2010; Weng et al., 2014). Therefore, this case study confirms that DRTO is expected to take over RTO in the near future, especially in those industrial sectors that deal with high value-added goods.

### 3.c.5 Conclusions

This chapter has described model predictive control and dynamic real-time optimization, in terms of their rationale and mathematical formulation. In addition, it has covered two of the most common numerical solution strategies for NMPC and DRTO problems. For the sake of clarity, we have devoted particular effort to highlighting the physical significance of all the mathematical expressions involved, using schematics, charts and ad-hoc descriptions. On the other hand, we have not provided rigorous mathematical formulations for all the methods and algorithms discussed, as this information can be found in scientific publications and dedicated books.

The conceptual description of NMPC and DRTO, reported in the first portion of this chapter, has been complemented with a case study, which involves optimization of the radiant section of a steam cracking unit, and allows comparing the efficacy and reliability of RTO with those of DRTO. The results of this application study show that DRTO outperforms RTO, in terms of economic benefits, reliability and capabilities to promote process safety. Therefore, dynamic real-time optimization is expected to take over RTO in the next few years, especially in those industrial sectors that produce high value-added products.

### 3.c.6 Disclaimer

ROMeO and DynSim are trademarks of Schneider Electric. SPYRO is a registered product of Technip-Pyrotec, originally developed by Politecnico di Milano.

### 3.c.7 References

Balasubramhanya, L.S., & Doyle, F.J., 1997, Nonlinear Control of a High-purity Distillation Column Using a Traveling-wave Model. *AIChE Journal* 43, 703-714.

- Bansal, V., Sakizlis, V., Ross, R., Perkins, J.D., & Pistikopoulos, E.N., 2003, New Algorithms for Mixed-integer Dynamic Optimization. *Computers & Chemical Engineering* 27, 647-668.
- Busch, J., Oldenburg, J., Santos, M., Cruse, A., & Marquardt, W., 2007, Dynamic Predictive Scheduling of Operational Strategies for Continuous Processes Using Mixed-logic Dynamic Optimization. *Computers & Chemical Engineering* 31, 574-587.
- Buzzi-Ferraris, G., & Manenti, F., 2012, BzzMath: Library Overview and Recent Advances in Numerical Methods. *Computer Aided Chemical Engineering* 30, 1312-1316.
- Buzzi-Ferraris, G., & Manenti, F., 2013, *Nonlinear Systems and Optimization for the Chemical Engineer: Solving Numerical Problems*. Wiley-VCH, Weinheim, Germany.
- Buzzi-Ferraris, G., & Manenti, F., 2014, *Differential and Differential-Algebraic Systems for the Chemical Engineer*. Wiley-VCH, Weinheim, Germany.
- Dente, M., Pierucci, S., Ranzi, E., & Bussani, G., 1992, New Improvements in Modeling Kinetic Schemes for Hydrocarbon Pyrolysis Reactors. *Chemical Engineering Science* 47, 2629-2634.
- Dones, I., Manenti, F., Preisig, H.A., & Buzzi-Ferraris, G., 2010, Nonlinear Model Predictive Control: a Self-Adaptive Approach. *Industrial & Engineering Chemistry Research* 49(10), 4782-4791.
- Dua, P., Kouramas, K., Dua, V., & Pistikopoulos, E.N., 2008, MPC on a Chip - Recent Advances on the Application of Multi-parametric Model-based Control. *Computers & Chemical Engineering* 32, 754-765.
- Gopalakrishnan, A., & Biegler, L.T., 2013, Economic Nonlinear Model Predictive Control for Periodic Optimal Operation of Gas Pipeline Networks. *Computers & Chemical Engineering* 52, 90-99.
- Gros, S., Srinivasan, B., & Bonvin, D., 2006, Robust Predictive Control Based on Neighboring Extremals. *Journal of Process Control* 16, 243-253.
- Hong, W., Wang, S., Li, P., Wozny, G., & Biegler, L.T., 2006, A Quasi-sequential Approach to Large-scale Dynamic Optimization Problems. *AIChE Journal* 52, 255-268.
- Lang, Y.D., & Biegler, L.T., 2007, A Software Environment for Simultaneous Dynamic Optimization. *Computers & Chemical Engineering* 31, 931-942.
- Li, P., Arellano-Garcia, H., & Wozny, G., 2008, Chance Constrained Programming Approach to Process Optimization Under Uncertainty. *Computers & Chemical Engineering* 32, 25-45.
- Logist, F., Vallerio, M., Houska, B., Diehl, M., & Van Impe, J., 2012, Multiobjective Optimal Control of Chemical Processes Using ACADO Toolkit. *Computers & Chemical Engineering* 37, 191-199.
- Lucia, S., Andersson, J.A.E., Brandta, H., Diehl, M., & Engell, S., 2014, Handling Uncertainty in Economic Nonlinear Model Predictive Control: A Comparative Case Study. *Journal of Process Control* 24, 1247-1259.
- Mahadevan, R., Agrawal, S.K., & Doyle, F.J., 2001, Differential Flatness Based Nonlinear Predictive Control of Fed-batch Bioreactors. *Control Engineering Practice* 9, 889-899.
- Manenti, F., 2011, Considerations on Nonlinear Model Predictive Control Techniques. *Computers & Chemical Engineering* 35, 2491-2509.
- Manenti, F., Dones, I., Buzzi-Ferraris, G., & Preisig, H.A., 2009, Efficient Numerical Solver for Partially Structured Differential and Algebraic Equation Systems. *Industrial & Engineering Chemistry Research* 48(22), 9979-9984.
- Manenti, F., & Rovaglio, M., 2008, Integrated multilevel optimization in large-scale poly(ethylene terephthalate) plants. *Industrial & Engineering Chemistry Research* 47(1), 92-104.
- Nagy, Z.K., & Braatz, R.D., 2007, Distributional Uncertainty Analysis Using Power Series and Polynomial Chaos Expansions. *Journal of Process Control* 17, 229-240.
- Pierucci, S., Brandani, P., Ranzi, E., & Sogaro, A., 1996, An industrial application of an on-line data reconciliation and optimization problem. *Computers & Chemical Engineering* 20, S1539-S1544.
- Rawlings, J.B., 2000, Tutorial Overview of Model Predictive Control. *IEEE Control Systems Magazine* 20, 38-52.
- Patwardhan, A.A., Rawlings, J.B., & Edgar, T.F., 1990, Nonlinear model predictive control. *Chemical Engineering Communications* 87, 123-141.

- Rossi, F., Casas-Orozco, D., Reklaitis, G., Manenti, F., & Buzzi-Ferraris, G., 2017, A Computational Framework for Integrating Campaign Scheduling, Dynamic Optimization and Optimal Control in Multi-unit Batch Processes. *Computers & Chemical Engineering* 107, 184-220.
- Rossi, F., Copelli, S., Colombo, A., Pirola, C., & Manenti, F., 2015, Online Model-based Optimization and Control for the Combined Optimal Operation and Runaway Prediction and Prevention in (Fed-)Batch Systems. *Chemical Engineering Science* 138, 760-771.
- Rossi, F., Manenti, F., & Buzzi-Ferraris, G., 2014, 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. *Industrial & Engineering Chemistry Research* 53, 15639-15655.
- Rossi, F., Manenti, F., Pirola, C., & Mujtaba, I., 2016, A Robust Sustainable Optimization & Control Strategy (RSOCS) for (Fed-)Batch Processes towards the Low-cost Reduction of Utilities Consumption. *Journal of Cleaner Production* 111, 181-192.
- Rossi, F., Reklaitis, G., Manenti, F., & Buzzi-Ferraris, G., 2016, Multi-scenario Robust Online Optimization and Control of Fed-batch Systems via Dynamic Model-based Scenario Selection. *AIChE Journal* 62, 3264-3284.
- Simon, D., 2006, *Optimal State Estimation: Kalman,  $H^\infty$ , and Nonlinear Approaches*. John Wiley & Sons, Hoboken, New Jersey, USA.
- Vallerio, M., Van Impe, J., Logist, F., 2014, Tuning of NMPC Controllers via Multi-objective Optimisation. *Computers & Chemical Engineering* 61, 38-50.
- Van, D., & Bosgra, H.O., 2006, Stochastic Closed-loop Model Predictive Control of Continuous Nonlinear Chemical Processes. *Journal of Process Control* 16, 225-241.
- Viganò, L., Vallerio, M., Manenti, F., Lima, N.M.N., Zuniga-Linan, L., & Manenti, G., 2010, Model Predictive Control of a CVD Reactor for Production of Polysilicon Rods. *Chemical Engineering Transactions* 21, 523-528.
- Weng, J., Shao, Z., Chen, X., Gu, X., Yao, Z., Feng, L., & Biegler, L.T., 2014, A Novel Strategy for Dynamic Optimization of Grade Transition Processes Based on Molecular Weight Distribution. *AIChE Journal* 60, 2498-2512.
- Zavala, V.M., & Biegler, L.T., 2009, The Advanced-step NMPC Controller: Optimality, Stability and Robustness. *Automatica* 45, 86-93.
- Zavala, V.M., Laird, C.D., & Biegler, L.T., 2008, Fast Implementations and Rigorous Models: Can Both Be Accommodated in NMPC? *International Journal of Robust and Nonlinear Control* 18, 800-815.