PROGRAMA DE DOCTORADO EN

INGENIERÍA INDUSTRIAL

TESIS DOCTORAL:

INTEGRATION OF ECONOMIC MPC AND MODIFIER ADAPTATION IN SLOW DYNAMIC PROCESSES WITH STRUCTURAL MODEL UNCERTAINTY

Presentada por Erika Oliveira da Silva

para optar al grado de

Doctor/a por la Universidad de Valladolid

Dirigida por:

D. César de Prada Moraga

D. Daniel Andrés Navia López

“I make a vow

Right here and now

I'm gonna spend my time this way”

Why – Robert Larson

This work was supported by the Regional Government of Castilla y León and the EU-FEDER (CLU 207-09, CL-EI-2021-07, UIC 233), as well as by project InCO4In, from Spanish AEI under project PGC2018-099312-B-C31, both with FEDER funds. It has also been supported by a pre-doctoral contract from the *Consejería de Educación de la Junta de Castilla y León*, Orden EDU/556/2019 under the title "Optimization in Industry 4.0".

The author thanks the European Social Fund and the *Consejería de Educación de la Junta de Castilla y León*.

Agradecimentos

A mis directores de tesis, César de Prada y Daniel Navia, por todo el conocimiento que me han aportado y consejos. En especial a César, por confiar en mí y ofrecerme otra vez la oportunidad de hacer un doctorado en España después de tantos años.

A los profesores del grupo de investigación "Control y supervisión de procesos" por la formación recibida y a los técnicos de laboratorio Teresa Álvarez y Graciano Garcia. A mis compañeros de trabajo: José Luis Pitarch, Carlos Palacín, María Marcos, Pablo Riquelme. En especial, quisiera agradecer a mi familia latina: Tomás García, Rogelio, Daniel Montes, Yuri, Felipe y Irina y mi familia vallisoletana: Cristian Pablos, Suní Rodriguez y Elena Rojo por todos los momentos "familia" que pasamos juntos.

A todos los compañeros del Departamento de Ingeniería Química. En especial a Priscila Scarcelli, Grazielle Rúas que estuvieron poco tiempo, pero se convirtieron en amigas para toda la vida.

A los ingenieros de Petronor Sergio, Rafael por el apoyo técnico al caso de estudio del splitter de propano-propileno.

A Aline, Fabiana, Irina, Rafaella por la infinita amistad, soporte que me han dado y felices recuerdos.

A Andrea, Lucile e Yosra, las mejores compañías de piso que una podría tener durante el periodo más incierto y duro del confinamiento.

A minhas amigas de "infância": Karina Seal, Gabriela Valença, Flávia Miranda e mais recentes: Thais Lima, Sara Lima e família que me ajudaram e apoiaram minha vinda a Espanha e me ensinaram que família vai além de compartir laços sanguíneos.

Aos meus pais, Erick e Maria Luiza, que sempre incentivaram a minha vontade de aprender.

Ao meu irmão, Erick, que sempre esteve ao meu lado, mesmo com a grande distancia física que nos separa.

A José y Palmira del bar Central Park que siempre me han recibido con una grande sonrisa y por las comidas riquísimas que me hicieran sentir como en casa.

A la música y a todos los amigos que hice a partir de ella, como María San Miguel y Galen Fraser que me recibieron con los brazos abiertos a su familia fiddle. En especial a María por inspirarme como músico y profesora y hacerme ver que la docencia siempre fue parte de mi vida y seguirá siendo.

A las profesionales de salud Fernanda Ximenes, Lidia Sanz, Maria Luisa Rojo que me ayudaron a cuidar de mi cuerpo y mente y permitieron que yo terminara esta tesis antes que ella terminara conmigo.

Summary

English

The rise of digitalization and increasing competitiveness has increased the interest in process optimization in industry. Decisions in industry are usually based on the automation pyramid. The automation pyramid has different layers that cover the different decisions to be made for different time horizons. One of them is Real-Time Optimization, known by its acronym RTO. Traditional RTO uses a steady-state nonlinear model of the process to optimize a plant's economic objective subject to process constraints. This is the technology currently used in commercial RTO applications. However, no model is a perfect representation of reality, and structural and parametric model uncertainties make the optimum calculated by RTO do not match those of the actual process. One way to address this problem is to modify the optimization problem so that the Necessary Conditions of Optimality (NCO) of the problem match those of the actual plant. This strategy is known as Modifier Adaptation (MA) methodology.

The MA methodology requires the gradient values of the real plant and the model to calculate the modifiers. There are several ways to accurately estimate model gradients, but estimation of the real process gradients are more difficult. In addition, the need to use stationary data is a limitation of RTO with MA, especially for slow dynamic systems.

Objectives

This thesis focuses on ways to mitigate the weaknesses of RTO and MA unification that we consider most critical for its application in industry. To this end, it is proposed to couple the RTO and control layers with the concepts of the Modifier Adaptation (MA) methodology by estimating process gradients or directly the MA modifiers using transient data. It also aims to apply the developed algorithms in different simulation case study, in a laboratory plant and in an industrial case.

Methodology

To achieve the objective, the methodology followed consisted of studying the fundamentals of predictive control and MA, as well as the problems posed by the estimation of gradients using transient data. Optimization methods were used to estimate these gradients/modifiers.

In addition, different case studies were studied: Williams Otto Reactor, Hybrid Pilot Plant, and Propane-Propylene Fractionator. A first-principles dynamic model was developed for each case study. In the case of the propane-propylene fractionation unit, the model was also validated with real process data. In addition, data reconciliation was used to deal with the presence of unreliable measurements in the plant. Subsequently, an economic predictive control (eMPC) architecture integrated with MA is proposed to bring a process to its true optimal operating point. To accelerate the time to optimum, it is proposed to update the MA modifiers at the same frequency as the controller. For this purpose, the process gradients or directly the MA modifiers should be estimated using process transient data.

Then, two proposals for estimating process gradients or modifiers, Dynamic Modifier Estimation (DME) and Transient MA (TMA), were studied. The DME algorithm aims to directly estimate the MA modifiers without the need to explicitly calculate the process and model gradients. The DME uses an optimization problem that attempts to minimize the difference between the modified cost function or constraints and the transient process measures using a moving horizon. The decision variables of the optimization problem are the dynamic modifiers that will match the static modifiers required by MA when the process reaches steady state.

On the other hand, the TMA is based on previous work using a Taylor series expansion that relates the outputs, inputs, and gradients of the process to be estimated. The previous method was formulated for an RTO (Real Time Optimization) context and did not take into account dynamic effects, so only the dependence of past decision variables was considered. Therefore, optimization and control were applied to different time scales. In this thesis, the Taylor series has been extended to include the effect of time (the derivative over time). This derivative can be estimated by a polynomial approximation that requires past data using the Nordsieck vector. Once the time dependence is computed, recursive identification algorithms can be implemented to estimate an approximation of the dynamic derivatives of the process with respect to the decision variables.

Results and conclusions

Both DME and TMA algorithms were first applied to a simulation of the Williams-Otto reactor. The results with the DME algorithm showed that this new approach can reach values quite close to the actual economic operating optimum, despite the parametric and structural mismatch between the model and the process. The TMA algorithm was also able to reach the process optimum in the Williams-Otto example with or without active constraints.

The TMA algorithm has also been applied in a hybrid laboratory plant that mimics the behavior of a CSTR with Van de Vusse reactions. The concept of hybrid plants is based on the fact that some process phenomena can be replaced by computations of measured variables and their effect on the process can be physically implemented, at least partially, by suitable actuators. The inclusion of the modifiers calculated with the TMA has significantly increased the process benefit for both experimental examples, reducing the suboptimality related to the process-model mismatch.

Finally, the two proposed methods were applied in a virtual environment similar to the real process of a propane-propylene fractionator of a refinery located in northern Spain. For this case study, the TMA method was modified to improve its performance. The results presented show that both DME and TMA could improve the process performance during the transient period for some cases. However, the applicability of MA with transient data depends on the process under consideration.

Español

La expansión de la digitalización y la creciente competitividad aumentó el interés en la optimización de procesos en la industria. Las decisiones en la industria normalmente se basan en la pirámide de automatización. Sus diferentes capas abarcan las diferentes decisiones que se debe tomar para diferentes horizontes de tiempo. Una de ellas es la Optimización en Tiempo Real, conocida por la sigla en inglés RTO. El RTO tradicional usa un modelo no lineal estacionario del proceso para optimizar un objetivo económico de la planta frente a restricciones del proceso. Esta es la tecnología usada actualmente por las aplicaciones comerciales de RTO. Sin embargo, ningún modelo es una representación perfecta de la realidad y las incertidumbres estructurales y paramétricas de los modelos hacen que los óptimos calculados por la RTO no coincidan con los del proceso real. Una forma de abordar este problema es modificar el problema de optimización de modo que las condiciones necesarias de optimalidad del problema (NCO) se igualen a los de la planta real. Esa estrategia es conocida como la metodología de adaptación de modificadores (Modifier Adaptation, MA).

La metodología MA necesita de los valores de gradiente de la planta real y del modelo para el cálculo de los modificadores. Hay diversas formas de estimar los gradientes del modelo con exactitud, sin embargo, la estimación en proceso real es más difícil. Además, la necesidad de usar datos en estacionario sigue siendo una limitación fundamental de la RTO con MA, principalmente para sistemas dinámicos lentos.

Objetivos

Esta tesis se enfoca en formas de mitigar las debilidades de la unificación RTO y MA que consideramos las más críticas para su aplicación en la industria. Para eso se propone que las capas de RTO y control se unan con los conceptos de la metodología de adaptación de modificadores (Modifier Adaptation, MA) estimando los gradientes de proceso o directamente los modificadores de MA usando datos de transitorio. Además tiene como objetivo aplicar los algoritmos desarrollados en diferentes casos de estudio en simulación, en una planta de laboratorio y en un caso industrial.

Metodología

Para alcanzar el objetivo, la metodología seguida consistió en estudiar los fundamentos de control predictivo y MA así como los problemas que presentaba la estimación de gradientes en transitorio. Se utilizaron métodos de optimización para estimar estos gradientes/modificadores.

Además se estudiaron los distintos casos de estudio: Reactor Williams Otto, Planta Piloto Híbrida y fraccionadora de propano-propileno. Para cada caso de estudio, se desarrolló un modelo dinámico de primeros principios. En el caso de la fraccionadora de propano-propileno, el modelo también se validó con datos reales del proceso. Además, se recurrió a la reconciliación de datos para hacer frente a la presencia de mediciones poco fiables en la planta.

En seguida, se propuso una arquitectura de un controlador predictivo económico (eMPC) integrado con MA de forma que permita a llevar un proceso a su punto de operación óptimo real. Con el objetivo de acelerar el tiempo al óptimo, se propone que los modificadores del MA se actualicen a la misma frecuencia del controlador. Para ello, los gradientes de proceso o directamente los modificadores del MA son estimados usando datos del transitorio del proceso.

Las dos propuestas presentadas para esa estimación son el Dynamic Modifier Estimation (DME) y Transient MA (TMA). El algoritmo de DME tiene como objetivo estimar directamente los modificadores de MA, sin necesidad de calcular explícitamente los gradientes del proceso y del modelo. El DME utiliza un problema de optimización que trata de minimizar la diferencia entre la función costo o restricciones modificadas con las medidas del proceso en transitorio usando un horizonte móvil. Las variables de decisión del problema de optimización son los modificadores dinámicos que coincidirán con los modificadores estáticos requeridos por MA cuando el proceso alcance el estado estacionario.

Por otro lado, el TMA está basado en un trabajo antecedente donde se utiliza una expansión en serie de Taylor que relaciona las salidas, entradas y gradientes del proceso que se desea estimar. El método anterior fue formulado para un contexto RTO (Real Time Optimization) y no tenía en cuenta los efectos dinámicos, por lo que sólo se consideró la dependencia de las variables de decisión pasadas. Debido a ello, la optimización y el control se aplicaron a diferentes escalas de tiempo. En esta tesis, la serie de Taylor fue expandida de forma que considera el efecto del tiempo (la derivada en el tiempo). Esta derivada se puede estimar mediante una aproximación polinómica que requiere datos pasados utilizando el vector de Nordsieck. Una vez calculada la dependencia temporal, se pueden implementar algoritmos de identificación recursiva para estimar una aproximación de las derivadas dinámicas del proceso con respecto a las variables de decisión.

Resultados y conclusiones

Ambos algoritmos fueron primeramente aplicados a una simulación del reactor de Williams-Otto. Los resultados con el algoritmo de DME, demostraron que este nuevo enfoque puede alcanzar valores bastante cercanos al punto óptimo de funcionamiento económico real a pesar del desajuste paramétrico y estructural entre el modelo y el proceso. El algoritmo TMA también ha sido capaz de alcanzar al óptimo de proceso en el ejemplo de Williams-Otto con o sin restricciones activas.

El algoritmo TMA también se ha aplicado en una planta de laboratorio híbrida que emula el comportamiento de un CSTR con las reacciones de Van de Vusse. El concepto de plantas híbridas se basa en que algunos fenómenos del proceso pueden ser reemplazados por cómputos de variables medidas y su efecto sobre el proceso puede ser implementado físicamente, al menos parcialmente, mediante actuadores adecuados. La inclusión de los modificadores calculados con el TMA ha aumentado significativamente el beneficio del proceso para ambos ejemplos experimentales, reduciendo la suboptimalidad relacionada con el desajuste proceso-modelo.

Finalmente, las dos metodologías propuestas se aplicaron en un ambiente virtual similar al proceso real de una fraccionadora de propano-propileno de una refinería al norte de España. Para este estudio de caso, el método TMA fue modificado para mejorar su performance. Los resultados presentados muestran que tanto DME como TMA pueden mejorar la performance del proceso durante el periodo transitorio. La aplicabilidad del MA con datos transitorios depende del proceso considerado.

Contents

1 Introduction

1.1 Current State of Real-Time Optimization Applications in the Process Industry

The objective of process optimization is to make the right operational decisions that minimize production costs and maximize profits while fulfilling safety, environmental and quality constraints.

Typically, industrial decision-making is organized as shown in Figure 1.1. Each layer uses different timescales and models to achieve its objectives. The first layer is Enterprise Resource Planning (ERP) and is responsible for production planning over a long time horizon. Most refineries and large chemical plants use linear programming (LP) with an economic objective for decision making in this layer. The next level corresponds to the Manufacturing Execution Systems (MES), which are responsible for scheduling operations and deal with the assignment of products and tasks to appropriate equipment and timing over a period of days or hours. Real-Time Optimization (RTO) comes next, using real-time process measurements to calculate the optimal setpoints to be applied to the process (the solid line indicates this real-time communication, while the dashed lines in ERP and MES may not be automatic or real-time). Typically, a Model Predictive Control (MPC) layer and then a Distributed Control System (DCS) layer with the basic controller, actuators and sensors in the field are responsible for making the process achieve these setpoints.

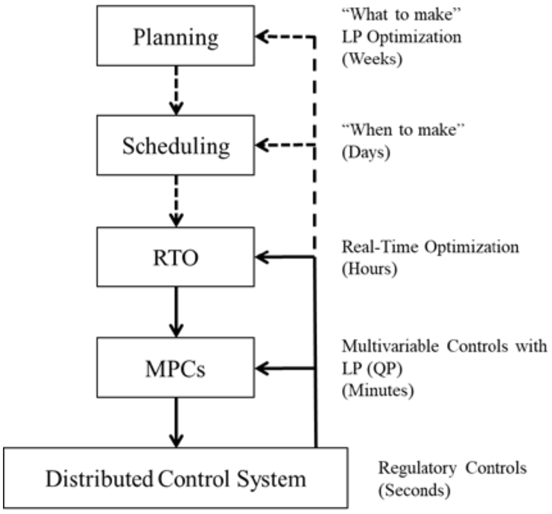


Figure 1.1: Industry Decision Hierarchy (Darby et al., 2011)

In this thesis, special attention is given to the RTO and MPC layers. The RTO is a set of algorithms and techniques that automatically compute, using real-time data, an estimate of the optimum operating point of a process at steady state, taking into account economic criteria. Ideally, a perfect RTO layer would solve the problem (1.1), where ϕ_p is the process cost function to be minimized, $\boldsymbol{u}$ are the decision variables between the lower and upper limits $[\boldsymbol{u}^L, \boldsymbol{u}^U]$, $\boldsymbol{y}_p$ are process variables, $\boldsymbol{g}_p$ a set of inequality constraints and $\boldsymbol{f}_p$ the "real" process model in steady state. The subscript p indicates that the variable or function corresponds to the real process.

$$\begin{gathered}\min_{\boldsymbol{u}} \phi_p(\boldsymbol{u}, \boldsymbol{y}_p) \\ s.t \;\; \boldsymbol{g}_p(\boldsymbol{u}, \boldsymbol{y}_p) \leq 0 \\ \boldsymbol{f}_p(\boldsymbol{u}, \boldsymbol{y}_p) = 0 \\ \boldsymbol{u}^L \leq \boldsymbol{u} \leq \boldsymbol{u}^U\end{gathered} \tag{1.1}$$

Since the correct mapping between input and output $\boldsymbol{f}_p$ is generally unknown, the RTO layer uses a rigorous nonlinear steady-state model $\boldsymbol{f}_m$ that attempts to predict both $\boldsymbol{f}_p$ and constraints $\boldsymbol{g}_p$, equation (1.2). Therefore, the performance of RTO is highly dependent on the accuracy of the model $\boldsymbol{f}_m$, $\boldsymbol{g}_m$, and on reliable measured data. A good RTO model should represent the process under a wide range of operating conditions, match the real process optimum, and not violate the constraints during optimization. In addition, the time to solve the RTO optimization problem must be less than the considered frequency of execution of the RTO (in traditional RTO, this would be the time to reach steady state), so there is a trade-off between model accuracy and computational complexity.

$$\begin{gathered}\min_{\boldsymbol{u}} \phi_m(\boldsymbol{u}, \boldsymbol{y}) \\ s.t \;\; \boldsymbol{g}_m(\boldsymbol{u}, \boldsymbol{y}) \leq \boldsymbol{0} \\ \boldsymbol{f}_m(\boldsymbol{u}, \boldsymbol{y}) = 0 \\ \boldsymbol{u}^L \leq \boldsymbol{u} \leq \boldsymbol{u}^U\end{gathered} \tag{1.2}$$

Equation (1.2) can be rewritten as equation (1.3) using $\boldsymbol{f}_m(\boldsymbol{u}, \boldsymbol{y})$ to make the cost function and constraints dependent on the inputs $\boldsymbol{u}$.

$$\begin{aligned} &\min_{\boldsymbol{u}} \phi(\boldsymbol{u}) \\ &s.t \;\; \boldsymbol{g}(\boldsymbol{u}) \leq \boldsymbol{0} \\ &\boldsymbol{u}^L \leq \boldsymbol{u} \leq \boldsymbol{u}^U \end{aligned} \tag{1.3}$$

As it is well known, even a really good model does not exactly match the real process due to three factors: parametric uncertainty (model parameters are different from the process), structural mismatch (simplification of complex phenomena, omission of some dynamics in the model), and process disturbances (Marchetti et al., 2016).

An intuitive strategy to address the model-process mismatch is to perform a model-parameter adaptation, using process data to update the parameters of the nonlinear steady-state model before solving the economic optimization. This idea is also referred to as the two-step approach, and it is probably the only strategy applied in commercial RTO systems today (Câmara et al., 2016).

A commercial RTO application will have a structure similar to Figure 1.2. In practice, the measurements need to be pre-processed before they can be used in a steady-state optimization algorithm. The first thing to check is the variable range or variable rate change of the variables. Then another steps are performed and should take into account (Bhat and Saraf, 2004):

- Steady State Detection (SSD): detects when the steady state has been reached. SSD uses statistical methods to verify this.
- Gross Error Detection (GED): this step verifies instrument failure, measurement bias, presence of leaks that can add errors to the measurements
- Data Reconciliation (DR): this step aims to eliminate the random noise from the measurements so that they satisfy the material and energy balances. Then the parameters of the $\boldsymbol{f}_m$ model are updated to best fit the online process data (model parameter update).
- Optimization: subsequently, the optimization problem (1.3) is solved using the updated model.

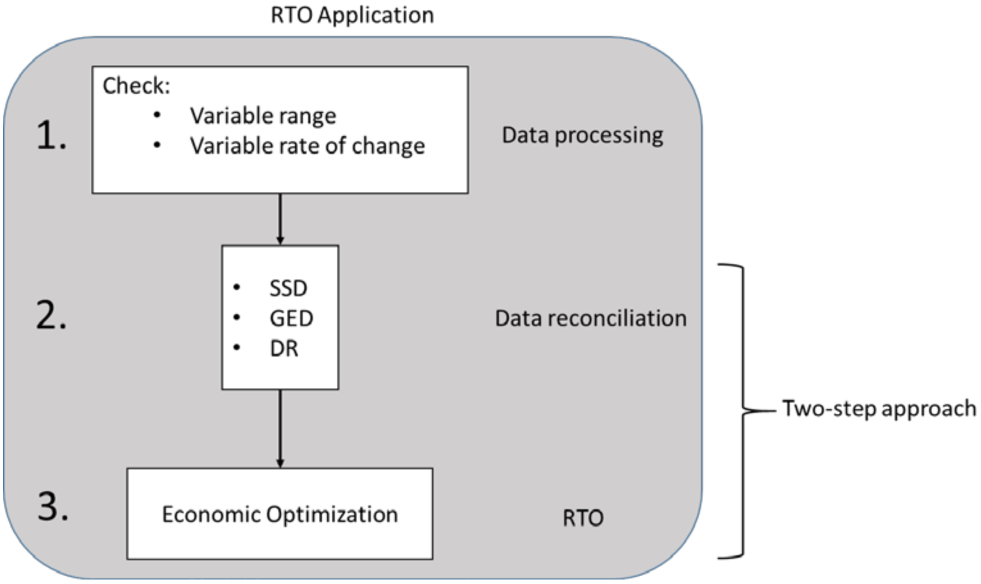


Figure 1.2: Inside a RTO application.

The performance issues of the current commercial RTO applications have been previously reported by Câmara et al. (2016), D. Quelhas et al. (2013) and Darby et al. (2011). One of the main challenges reported by the authors is the plant-model mismatch. Model uncertainty or mismatch is one of the main causes of suboptimal operating conditions, violation of constraints and the long period of time to achieve final process stabilization with no guarantee that the process will achieve an optimal operation (D. Quelhas et al., 2013). If the model is structurally correct, the online model parameter adaptation (the two-step approach) can converge to the plant optimum in one iteration (Chachuat et al., 2009). However in case of structural mismatch between process and model, this strategy does not guarantee to find the process optimum when convergence is achieved (Chachuat et al., 2009; D. Quelhas et al., 2013; Yip and Marlin, 2004).

In order to minimize the problems related to the structural uncertainty of the model, other algorithms have been studied. These algorithms consider different ways of using process measurements to compensate for the mismatch problem, such as: Direct Input Adaptation (Self-Optimizing Control or Extremum seeking) and Modifier Adaptation (Chachuat et al., 2009). A brief description of these algorithms is given in the following sections.

Besides structural mismatch, another challenge for current commercial RTO applications is that some industrial processes never reach steady state due to frequent disturbances, changing parameters and demand. In these processes, the frequency execution of a traditional RTO could be very large, making the use of these applications questionable. In this case, the idea of considering the dynamics of the process should be useful and then Dynamic RTO or economic MPC (eMPC) have been proposed. They are also presented in the following sections.

1.2 Direct Input Adaptation: Self-Optimizing control and Extremum seeking control

Unlike the model-parameter adaptation algorithms presented in the two-step approach, the Direct Input Adaptation does not require the solution of successive optimization problems. The main objective of these algorithms is to use feedback control to maintain some variables related to optimal plant performance at their set points values. Therefore, the challenge is to choose the right variables to control that will drive the plant to its optimal operation, since these variables change with the active set. Self-optimizing control and Extremum seeking control are examples of direct input adaptation algorithms (Chachuat et al., 2009).

Self-optimizing control aims to track a constant setpoint, as a function of the process variables, that is close to the process optimum, considering an acceptable loss, defined as the difference between the actual value and the true optimum value (Skogestad, 2000). This set point can be a measurement, a linear combination of measurements, or the gradient of the economic cost function.

The idea is to find a self-optimizing variable that is a constrained optimum or a flat optimum, cases (a) and (b) in Figure 1.3, respectively. The case (c) in Figure 1.3 is difficult to solve because the cost function is sensitive to the value of the self-optimizing variable.

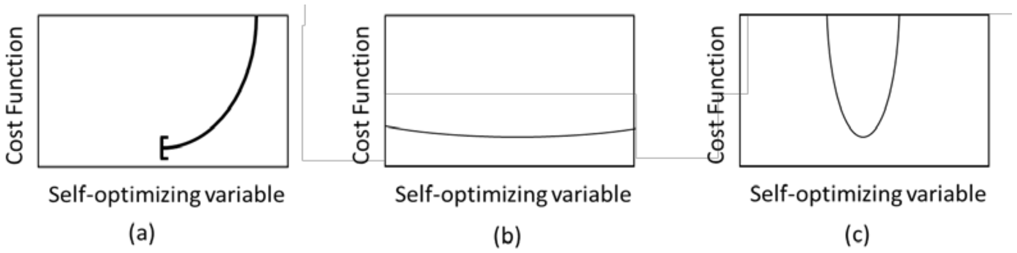


Figure 1.3: Problems in implementing self-optimizing control.
(a) Constrained optimum, (b) Unconstrained flat optimum, and (c) Unconstrained sharp optimum (Skogestad, 2000).

Therefore, a good self-optimizing variable must satisfy these four requirements:

a) Its optimum must be insensitive to the disturbances.
b) It must be a variable that is easy to measure and control.
c) Its value is sensitive to the manipulated variables.
d) If there is more than one variable, they must not be closely correlated.

Extremum seeking control aims to find and maintain the extremum value of a static map between input and cost function. In the classical approach, a slow periodic dither signal (sinusoidal wave) is superimposed on to the input signal, equation (1.4).

$$u(t) = u + a \sin \omega t \tag{1.4}$$

A slow frequency ω is used so that the plant map could be considered static. This requires three different time scales between the plant (fast), the sinusoidal perturbation (intermediate) and a convergence to the optimum (slow). This separation of time scales can result in very slow convergence for most chemical or biochemical processes (Krishnamoorthy and Skogestad, 2022).

1.3 Modifier Adaptation

In order to incorporate ability to deal with structural mismatch into the two-step approach, further developments have incorporated process gradient information into the calculation to satisfy the necessary conditions of optimality (NCO) of the plant and the model. Roberts (1979) proposed a modification of the two-step approach to account for the differences between the process and model gradients. This idea was later given the acronym ISOPE, Integrated System Optimization and Parameter Estimation (Roberts, 1995; Roberts and Williams, 1981).

During this period, several variations of the ISOPE algorithm were developed to overcome some limitations of the original algorithm. BrdyŚ et al. (1986) presented a proposal to handle an optimization problem with process constraints in the outputs. Almost ten years later, Brdyś and Tatjewski (1994) included a dual control effect in the ISOPE framework to force the process to follow the optimum and to produce an output with sufficient excitation to compute the future control signals, including a constraint to guarantee the excitation needed to estimate the process gradients.

Then, almost twenty years later, Tatjewski (2002) concluded that adding a bias correction (model shift) term, equation (1.5), to the output of the predicted model would sufficient to satisfy the output-matching condition, equation (1.6). Including the bias, the value of the model outputs matches the process value without the need to update the model parameters.

$$\boldsymbol{a}_k := \boldsymbol{y}_p(\boldsymbol{u}_k) - \boldsymbol{y}(\boldsymbol{u}_k, \boldsymbol{\theta}_k) \tag{1.5}$$

$$\boldsymbol{y}(\boldsymbol{u}_k, \boldsymbol{\theta}_k) = \boldsymbol{y}_p(\boldsymbol{u}_k) \tag{1.6}$$

Subsequently, Gao and Engell (2005) proposed a new name for the algorithm: iterative gradient-modification optimization, since the name ISOPE is no longer appropriate (the parameter estimation step is not necessary). The authors also defined new modifiers for the constraints and proposed an alternative to keep the process excitation to estimate good gradients without the dual constraint.

Finally, in 2009, Marchetti et al. (2009) formalized the Modifier Adaptation (MA) methodology. In the traditional MA (Marchetti et al., 2009), at the current steady state k, additional terms involving past inputs $\boldsymbol{u}_{k-1}^*$ and modifiers $\boldsymbol{\lambda}$, $\boldsymbol{\gamma}$, ε are added to the original optimization problem in (1.2), formulating the modified problem as problem (1.7).

$$\begin{gathered} \min_{\boldsymbol{u}} \phi_M := \phi(\boldsymbol{u}) + \boldsymbol{\lambda}_k^T(\boldsymbol{u} - \boldsymbol{u}_{k-1}^*) \\ s.t \quad g_{M,i}(\boldsymbol{u}) := g_i(\boldsymbol{u}) + \boldsymbol{\gamma}_{k,i}^T(\boldsymbol{u} - \boldsymbol{u}_{k-1}^*) + \varepsilon_{k_i} \leq 0 \, , i = 1, \dots, n_g \\ \boldsymbol{u}^{\boldsymbol{L}} \leq \boldsymbol{u} \leq \boldsymbol{u}^{\boldsymbol{U}} \end{gathered} \tag{1.7}$$

The variables ϕ_M and $g_{M,i}$ are the modified cost function and constraints respectively, $\boldsymbol{u}_{k-1}^*$ are the actual values of the manipulated variables calculated and applied to the process in the previous steady state. n_g is the number of nonlinear constraints. The modifiers are given by equations (1.8)-(1.10).

$$\boldsymbol{\lambda}_k^T = \left.\frac{\partial \phi_p}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^*} - \left.\frac{\partial \phi}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^*} \tag{1.8}$$

$$\boldsymbol{\gamma}_{k_i}^T = \left.\frac{\partial g_{p_i}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^*} - \left.\frac{\partial g_i}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^*} \tag{1.9}$$

$$\varepsilon_{k_i} = g_{p_i}(\boldsymbol{u}_{k-1}^*) - g_i(\boldsymbol{u}_{k-1}^*)\text{ , } i = 1, \dots, n_g \tag{1.10}$$

$\boldsymbol{\lambda}_k^T$ and $\boldsymbol{\gamma}_k^T$ are the first-order modifiers and correct the curvature of the model, and ε_{k_i} is the zero-order modifier and corrects the offset for each constraint, see Figure 1.4.

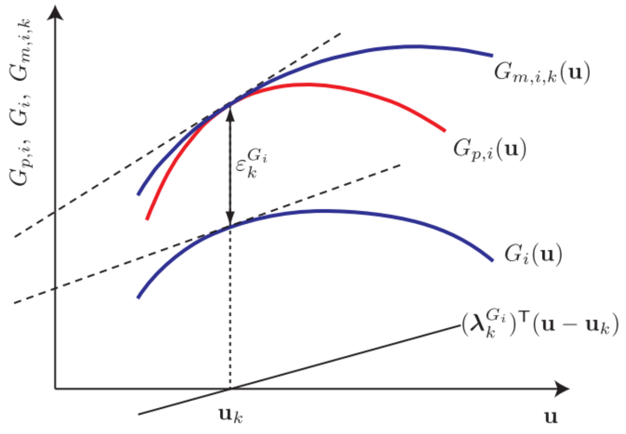


Figure 1.4: First order modification in the constraints (figure from Marchetti et al. (2016))

The total number of modifiers required depends on the number of constraints, n_g, and the number of inputs, n_u as shown in equation (1.11).

$$n_{mod} = n_g + n_u(n_g + 1) \tag{1.11}$$

Problem (1.7) is solved iteratively by calculating the values of $\boldsymbol{u}_k^*$ that are applied to the process. When the next steady state is reached, the newly measured process outputs are used to update the modifiers. In general, to avoid excessive correction, the strategy used is to filter these modifiers before using the original values in the optimization problem, equation (1.12) where $K = \operatorname{diag}(K_1, \dots, K_{n_u}),\ K_i \in (0,1]$.

$$\begin{aligned} \boldsymbol{\lambda}_k &= (I - K^\lambda)\boldsymbol{\lambda}_{k-1}^T + K^\lambda \left(\left.\frac{\partial \phi_p}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^*} - \left.\frac{\partial \phi}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^*} \right) \\ \boldsymbol{\gamma}_k &= (I - K^\gamma)\boldsymbol{\gamma}_{k-1}^T + K^\gamma \left(\left.\frac{\partial \boldsymbol{g}_p}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^*} - \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^*} \right) \end{aligned} \tag{1.12}$$

$$\boldsymbol{\varepsilon}_k = (I - K^{\boldsymbol{\varepsilon}})\boldsymbol{\varepsilon}_{k-1}^T + K^{\boldsymbol{\varepsilon}}\left(\boldsymbol{g}_p(\boldsymbol{u}_{k-1}^*) - \boldsymbol{g}(\boldsymbol{u}_{k-1}^*)\right)$$

When the MA modifiers are computed correctly, the RTO problem reaches the process optimum by convergence, regardless of the quality of the model. However, the model must at least satisfy a model adequacy requirement: the reduced Hessian of the Lagrangian in the real optimum must be positive definite (François and Bonvin, 2013).

Other equivalent formulations are possible, for example, in the output MA or MAy approach, the modifiers are used directly in the predicted output, instead of adding an additional term to the cost function (Papasavvas et al., 2019).

1.4 Dynamic RTO and eMPC

Many industrial processes with frequent grade changes would benefit from the use of dynamic information in the optimization layer. This dynamic information could be included in the RTO layer by replacing the steady-state optimization problem with a dynamic optimization problem as in a DRTO (dynamic RTO). The other option is to consider an economic optimization layer within the MPC (an economic MPC, eMPC).

Both DRTO and eMPC use the economic information in the objective function and a dynamic model as a constraint in the optimization problem. The main difference is that DRTO is used in the same hierarchical structure of RTO and calculates the setpoints to a control layer, and eMPC calculates the control trajectory to achieve the economic objective function (Ellis et al., 2014).

1.5 Using transient measurements for gradient estimation

Mansour and Ellis (2003) present several algorithms for estimating process gradients using steady state and transient data. The advantage of using transient data is to avoid the waiting time for steady state. The idea is to identify a linear or nonlinear dynamic model online using the process measurements and then calculate the steady-state gradient from this model.

Others alternatives to using transient measurements in steady-state optimization have been studied. François and Bonvin (2014) presented an estimation of the gradients using Neighboring Extremals (NE). The method estimates the gradient based on a variational analysis around the nominal operating point for the nominal parameter values where the gradient is zero (unconstrained problem). However the method is not valid in the case of structural mismatch. Afterward, Rodríguez-Blanco et al. (2017) developed a method that uses a truncated Taylor expansion of the process cost combined with identification algorithms to estimate process gradients. This second method is able to estimate gradients in the presence of parameter or structural mismatch.

1.6 Unification RTO + MPC + MA

The mismatch between the RTO+MA and MPC models can lead to a poor economic performance if the setpoint calculated in the upper layer is unreachable by the controller layer. A logical way to solve this problem is to integrate these two layers.

Recent works have presented different formulations of the RTO+MA+MPC problem. Vaccari and Pannocchia (2017) used the so-called economic MPC (eMPC), where the optimization problem has an economic objective function. Their contribution includes the modifiers from the MA methodology to achieve the NCO of the process despite the plant-model mismatch. The implementation has two layers: the first is a steady-state economic objective problem and the second is a dynamic optimization problem. The eMPC is applied to an example of a continuous stirred tank reactor using a state space model in the controller with a parameter mismatch. In this example, the steady state gradient of the process is assumed to be known for the calculation of the MA modifiers. Subsequently, Pannocchia (2018) and Vaccari and Pannocchia (2018) extended their previous work by estimating the process gradients from a local linear input-output model obtained from online data and identification algorithms. To guarantee the necessary excitation, a random signal was included for a period of time, which increased the convergence time of the algorithm.

Then, Hernández and Engell (2019) presented an alternative formulation for economic control based on MA, where the corrections are made in the nominal dynamic plant model instead of the objective function. They identified linear dynamic models to approximate the true plant map, and then computed the plant gradients. Since the estimated linearized model is only valid in the neighborhood of the current state, the sequence of optimal control moves is limited to this region.

Faulwasser and Pannocchia (2019) used the output modifier adaptation (MAy) and the eMPC from their previous work without the terminal constraint. The authors emphasized the importance of estimating correct plant gradients. In this work, the gradients were assumed to be known. After, Vaccari et al. (2020) presented a technique for direct estimation of the modifiers using steady-state perturbations and a Broyden update algorithm. Subsequently, Vaccari et al. (2021) presented an extension of Vaccari et al. (2020) to compare two techniques for estimating either plant gradients or modifiers using steady-state measurements via Broyden's update and linear regression. The eMPC schemes were tested in simulation on two benchmark examples, highlighting the fact that the use of transient measurements to estimate the process gradients can be an interesting approach to speed up convergence.

As can be seen from the aforementioned papers, the investigation of how to correctly estimate the process gradients/modifiers to be used in a framework for RTO+MA+MPC is an open issue.

1.7 Motivation

Although MA is a powerful methodology that guarantees the true optimum of a process despite the use of incorrect models, the industrial application of MA is almost non-existent (Marchetti et al., 2016). One of the reasons for this is that many industrial processes never reach a steady state due to frequent disturbances or slow dynamics, so there are not enough steady-state data for the gradient estimation step.

The motivation of this thesis, therefore, is to develop, from the idea of the traditional MA, a new methodology that could be easily applied in an industrial process. This new methodology will consider the unification of RTO and MPC layers with MA and will use new algorithms to estimate the process gradients using transient measurements.

1.8 Objectives

The general objective of this thesis is to study strategies to make RTO with MA applicable in slow dynamic processes. This thesis proposes the unification of the RTO and control layers and the use of transient data to estimate the steady-state gradients required by MA. Moreover, the proposed tools are applied to a simulation case, an experimental laboratory plant, and then to a real industrial process.

Along this line, in order to achieve the main objective, the following specific objectives are proposed:

- Study the Modifier Adaptation Methodology
- Study Aspen DMC
- Identify the main challenges of the Modifier Adaptation methodology to be applied in slow process.
- Propose a unification of the Real-Time Optimization layer, the MPC layer and Modifier Adaptation
- Propose new methods to estimate MA modifiers using transient measurements that can be performed with the same sampling time of the MPC, with the aim of reducing the convergence time of the method in processes with slow dynamics.
- Application of the proposed solutions in a simulation case
- Application of the proposed solutions in a laboratory plant
- Application of the proposed solutions in an industrial case

1.9 Organization of the thesis

The thesis is divided into 7 chapters as follows:

Chapter 2 presents a proposal of RTO, MPC and MA unification.

Chapter 3 presents the first proposed algorithm that directly estimates the MA modifiers, the Dynamic Modifier Estimation algorithm. This chapter shows the application of the method in the Williams-Otto reactor case study.

Chapter 4 presents the second proposed algorithm that is able to improve the performance of the RTO using an identification algorithm and transient measurements, the TMA. This method is applied in simulation in the Williams-Otto reactor case study.

Chapter 5 presents the results of the TMA algorithm applied to a laboratory plant. The plant consists of a hybrid reactor based on the Van der Vusse reactions.

Chapter 6 presents the application of the DME and a modification of the TMA algorithms applied to an industrial case: a propane-propylene splitter.

Chapter 7 presents the conclusions.

1.10 Contributions and publications

Journal contributions:

Oliveira-Silva, E., de Prada, C., Navia, D., 2021. Dynamic optimization integrating modifier adaptation using transient measurements. Comput. Chem. Eng. 149, 107282. https://doi.org/10.1016/j.compchemeng.2021.107282

Oliveira-Silva, E., de Prada, C., Montes, D., Navia, D., 2023. Economic MPC with Modifier Adaptation using Transient Measurements. Comput. Chem. Eng. 108205. https://doi.org/10.1016/j.compchemeng.2023.108205

Book chapters:

Oliveira-Silva, E., de Prada, C., Navia, D., 2021. Economic MPC with Modifier Adaptation using Transient Measurements, in: Computer Aided Chemical Engineering. Elsevier Masson SAS, pp. 1253–1258. https://doi.org/10.1016/B978-0-323-88506-5.50193-5

Congress contributions:

Silva, E. O. da, Rodríguez-Blanco, T., and Prada, C. de, “Aplicación de la metodología de adaptación de modificadores con datos de transitorio para RTO de una columna fraccionadora de propano-propileno,” XXXIX Jornadas de Automática, Badajoz, 2018.

Oliveira-Silva, E., de Prada, C., "Methodology to achieve convergence in a rigorous dynamic model of a superfractionator", X Congress of Eurosim, Logroño, La Rioja, 2019.

Oliveira-Silva, E., de Prada, C., Navia, D. Economic MPC with Modifier Adaptation using Transient Measurements, ESCAPE-31, Istanbul 2021.

Oliveira-Silva, E., de Prada, C., Navia, D., 2022. Simulation platform of an industrial propylene-propane splitter integrated to Advanced Process Control for Real

Time Optimization experiments. IFAC-PapersOnLine 55, 673–678. https://doi.org/10.1016/j.ifacol.2022.07.521

Montes, D., Zamarreño, J.M., Pitarch, J.L., Oliveira da Silva, E., Prada, C. De, 2021. Implementación de capas superiores de la pirámide de automatización en una planta piloto híbrida, in: XLII JORNADAS DE AUTOMÁTICA : LIBRO DE ACTAS. Servizo de Publicacións da UDC, pp. 403–410. https://doi.org/10.17979/spudc.9788497498043.403

2 Economic MPC with MA (eMPC+MA)

This chapter presents a unification of RTO, MPC, and MA architecture that is simple enough to be applied to real processes. The proposed architecture is shown in Figure 2.1. It is composed of three modules: a module integrating RTO, MPC and MA, which performs the core task, i.e., an economic MPC (eMPC) with MA, a Moving Horizon Estimator (MHE) for estimating states and disturbances, and a third one for computing the values of the MA modifiers. This scheme is executed at regular time intervals denoted by the sub-index k.

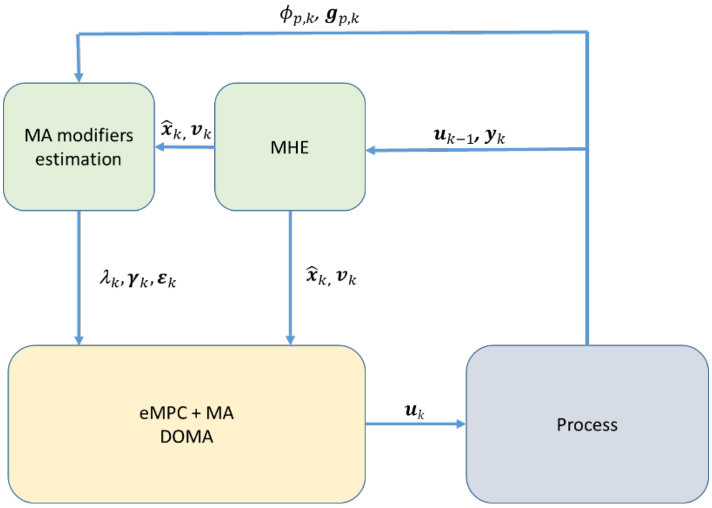

$\phi_{p,k}, \boldsymbol{g}_{p,k}$
MA modifiers estimation
$\hat{\boldsymbol{x}}_k, \boldsymbol{v}_k$
MHE
$\boldsymbol{u}_{k-1}, \boldsymbol{y}_k$
$\lambda_k, \boldsymbol{\gamma}_k, \boldsymbol{\varepsilon}_k$
$\hat{\boldsymbol{x}}_k, \boldsymbol{v}_k$
eMPC + MA
DOMA
$\boldsymbol{u}_k$
Process

Figure 2.1: Architecture of eMPC+MA.

At each iteration k, which coincides with the sampling time of all modules, the values of the manipulated variables applied to the process in the past interval $\boldsymbol{u}_{k-1}$ and the current process measurements $\boldsymbol{y}_k$ are collected and sent to the MHE module to estimate the current model states $\hat{\boldsymbol{x}}_k$ and disturbances $\boldsymbol{v}_k$. The solution given by the MHE module ($\hat{\boldsymbol{x}}_k$ and $\boldsymbol{v}_k^*$), is used by the MA modifiers estimation module with current values of the process economic cost function $\phi_{p,k}$ and constraints $\boldsymbol{g}_{p,k}$, to estimate online the values of the cost and constraints modifiers $\boldsymbol{\lambda}_k$, $\boldsymbol{\gamma}_k$ and $\boldsymbol{\varepsilon}_k$, after filtering. Next, the modifiers are used in the eMPC+MA to compute new control actions over a given horizon. Then, the first control action of the horizon, corresponding to the current time $\boldsymbol{u}_k$, is applied to the process at time k. Finally, at the next sample time, the whole procedure is repeated, following a moving horizon policy, as in MPC. All of the three modules presented here are described in the following sections.

2.1 Economic MPC with MA (eMPC+MA)

This module incorporates an economic target and the modifiers from the MA methodology into a dynamic optimization problem, searching for the control moves $\Delta\boldsymbol{u}_{k+i}\ i = 0,1,2,\dots,n_u - 1,$ that minimize the cost function (2.1) subject to constraints (2.2) to (2.8). Note that the problem is formulated in the continuous time domain (t). The cost function (2.1) considered is in steady state, so its value is calculated at the end of the prediction horizon, $\phi(t_{pred})$, and using the value of the manipulated variable, in steady state, $\bar{\boldsymbol{u}}$ in MA terms.

The process is represented by a continuous dynamic model (2.2) and (2.3), which is assumed to be continuously differentiable. In problem (2.1)-(2.9), $\boldsymbol{x} \in \mathbb{R}^{n_x}$ represent the states, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ the control actions, $\boldsymbol{y} \in \mathbb{R}^{n_y}$ the measured outputs, $\hat{\boldsymbol{x}} \in \mathbb{R}^{n_x}$ and $\boldsymbol{v}_k \in \mathbb{R}^{n_v}$ are the actual states and disturbances, estimated by the MHE module (section 2.2). Since the model is formulated in continuous time, $\boldsymbol{x}$, $\boldsymbol{u}$ and $\boldsymbol{y}$ are functions of t, but for simplicity this dependence has been omitted throughout the document unless necessary (e.g. (2.7)). Using a control vector parameterization approach, the control actions are only allowed to change at regular time intervals $\Delta t = t_k - t_{k-1}$. Denoting k the current sampling time, the control actions $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$, computed and applied at time t_k, are kept constant within each time interval $[t_k, t_{k+1})$, as in (2.7). The current and future control moves, denoted as $\Delta\boldsymbol{u}_{k+i}$, $i = 0,1,2,\dots,n_u - 1$ defined in (2.6), are the decision variables of problem (2.1)-(2.9). The selection of the control horizon n_u and other tuning parameters follows the usual rules of MPC (Shah and Engell, 2011).

The model allows to compute predictions of the cost function and constraints over a future horizon from t_k to the final prediction horizon t_{pred}, long enough for the model to reach steady state. n_{pred} refers to the number of time instants from t_k to reach t_{pred}.

The control moves are computed every sampling time from the current time t_k to a control horizon t_{k+n_u} , after which, $\Delta\boldsymbol{u}_{k+i} = 0$, but only the first control move $\Delta\boldsymbol{u}_k$ is applied to the process.

$$\min_{\substack{\Delta u_{k+i} \\ i=0,1,\dots n_u-1}} \quad \phi(t_{pred}) + \boldsymbol{\lambda}_k^{\boldsymbol{T}}(\overline{\boldsymbol{u}} - \boldsymbol{u}_{k-1}) + \sum_{i=0}^{n_u-1} \Delta\boldsymbol{u}_{k+i}^T \, \boldsymbol{Q}_u \Delta\boldsymbol{u}_{k+i} \tag{2.1}$$

$$s.t. \quad \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}_k) = \boldsymbol{0}, \ \forall t \in [t_k, t_{pred}] \tag{2.2}$$

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}, \boldsymbol{v}_k) = \boldsymbol{0}, \forall t \in [t_k, t_{pred}] \tag{2.3}$$

$$\boldsymbol{g}(\boldsymbol{u}) + \boldsymbol{\gamma}_k^T(\overline{\boldsymbol{u}} - \boldsymbol{u}_{k-1}) + \boldsymbol{\varepsilon}_k \leq \boldsymbol{0}, \ \forall t \in [t_k, t_{pred}] \tag{2.4}$$

$$\boldsymbol{u}^{\boldsymbol{L}} \leq \boldsymbol{u}_{k+i} \leq \boldsymbol{u}^{\boldsymbol{U}}, \qquad i = 0,1, \dots n_{pred} - 1 \tag{2.5}$$

$$\boldsymbol{u}_{k+i} = \boldsymbol{u}_{k+i-1} + \Delta\boldsymbol{u}_{k+i}, \qquad i = 0,1, \dots n_{pred} - 1 \tag{2.6}$$

$$\boldsymbol{u}(t) = \boldsymbol{u}_{k+i}\,, \qquad t \in [t_{k+i}, t_{k+i+1}], \qquad i = 0,1, \dots n_{pred} - 1 \tag{2.7}$$

$$\Delta\boldsymbol{u}_{k+i} = 0\,, \ \ i = n_u, \dots n_{pred} - 1 \tag{2.8}$$

$$\boldsymbol{x}(t_k) = \widehat{\boldsymbol{x}}_k \tag{2.9}$$

The cost function (2.1) consists of three terms:

1. The first term, $\phi(t_{pred})$, corresponds to an economic objective computed with the value of the model variables and control actions at the end of the prediction horizon t_{pred}, where the variables are expected to reach steady state. The objective is to achieve and maintain the process operating steadily at the real process optimum. Examples of economic objectives can be: maximizing benefits or production, or minimizing costs or energy. We assume that the value of $\phi_{p,k}$ at any time instant k can be computed from process measurements and control actions.
2. The second one, $\boldsymbol{\lambda}_k^T(\overline{\boldsymbol{u}} - \boldsymbol{u}_{k-1})$, is an additional MA-type term responsible for modifying the economic cost function ϕ to match the NCO of the real plant in steady state. Here $\overline{\boldsymbol{u}} = \boldsymbol{u}_{k-1} + \sum_{i=0}^{n_u-1} \Delta\boldsymbol{u}_{k+i}$ is the final value of the control actions. Note that this term is equivalent to the term added to the cost function in the traditional MA formulation. The modifier $\boldsymbol{\lambda}_k$ is computed at each sampling time k by the external module DME and is held constant as $\boldsymbol{\gamma}_k$ and $\boldsymbol{\varepsilon}_k$ in problem (2.1)-(2.9).
3. Finally, the third term, $\sum_{i=0}^{n_u-1} \Delta\boldsymbol{u}_{k+i}^T \, \boldsymbol{Q}_u \Delta\boldsymbol{u}_{k+i}$ penalizes changes in the manipulated variables, which increases stability and contributes to model adequacy and convexification (François and Bonvin, 2013). $\boldsymbol{Q}_u$ is a positive definite matrix, with weighting factors on the control moves $\Delta\boldsymbol{u}$, which can be considered as tuning factors for normalization and stabilization, as in the current practice of MPC.

The cost function is subjected to the model (2.2), (2.3), and the inequality constraints (2.4) and (2.5). In (2.4), $\boldsymbol{g}(\boldsymbol{u})$ are computed using (2.2) and (2.3), the constraint modifiers $\boldsymbol{\gamma}_k$ and $\boldsymbol{\varepsilon}_k$ are calculated in the MA modifiers estimation module, and are kept constant at each iteration k.

To solve (2.1)-(2.9), at each sampling time k, the initial value of the model states $\boldsymbol{x}(t_k)$ and the disturbances are initialized to the value $\widehat{\boldsymbol{x}}_k$ and $\boldsymbol{v}_k$ respectively (2.9), values estimated by the MHE module.

The eMPC+MA presented in (2.1)-(2.9) was proposed considering the standard formulation (1.7) of MA, but other alternative and equivalent formulations are possible if the output MA (MAy) (Papasavvas et al., 2019) approach is chosen. In this case, the modifiers are used directly in the predicted output, instead of adding an extra term to the cost function. Similarly, the dynamic optimization problem could had been formulated as a two-step problem, as in (Vaccari et al., 2020), where a static optimization is first performed to compute optimal steady-state objectives, and then the results are used as terminal constraints in a horizon optimal control problem (FHOCP). Nevertheless, in the formulation of Vaccari et al. (2020), it may happen that the targets computed by the objective optimization problem are not achievable by the FHOCP because path constraints are not considered in the computation of the optimal targets. This problem can be mitigated by the proposed formulation (2.1)-(2.9), where the search takes place only within the feasible region.

2.2 Moving Horizon Estimation (MHE)

To provide offset-free behavior for the eMPC+MA problem (2.1)-(2.9), as well as to compute the initial values of states and disturbances, an augmented state estimator can be employed. Extended Kalman Filter (EKF) or Moving Horizon Estimation (MHE) are two well-known methods for this task, and either of them could be used in our scheme. Nevertheless, MHE was chosen because it fits better into the optimization framework, allows estimation of specific disturbances affecting the nonlinear model, and includes constraints in the formulation when necessary (Huang et al., 2010; Rawlings et al., 2019; Vaccari and Pannocchia, 2018). The MHE uses past information from measurements and control actions to estimate the values of the states and disturbances $\boldsymbol{v}$, solving the dynamic optimization problem (2.10)-(2.17).

$$\min_{\substack{\boldsymbol{x}_{k-ne}\boldsymbol{v}_{k-i}\\ i=1,\dots,n_e}} \sum_{i=0}^{n_e-1} \Delta\boldsymbol{y}_{k-i}^T\boldsymbol{Q}_y\Delta\boldsymbol{y}_{k-i} + \ \Delta\boldsymbol{x}_{k-n_e}^T\boldsymbol{Q}_x\Delta\boldsymbol{x}_{k-n_e} + \sum_{i=1}^{n_e}\boldsymbol{v}_{k-i}^T\boldsymbol{Q}_v\boldsymbol{v}_{k-i} \tag{2.10}$$

$$s.t. \qquad \boldsymbol{f}(\dot{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{0}, \forall t \in \left[t_{k-n_e}, t_k\right], \ \boldsymbol{x}\left(t_{k-n_e}\right) = \boldsymbol{x}_{k-n_e} \tag{2.11}$$

$$\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{y},\boldsymbol{v}) = \boldsymbol{0}, \ \forall t \in \left[t_{k-n_e}, t_k\right] \tag{2.12}$$

$$\boldsymbol{g}(\boldsymbol{u},\boldsymbol{y}) \leq \boldsymbol{0}, \ \forall t \in \left[t_{k-n_e}, t_k\right] \tag{2.13}$$

$$\boldsymbol{u}(\boldsymbol{t}) = \boldsymbol{u}_{\boldsymbol{k-i}}, \qquad \boldsymbol{v}(\boldsymbol{t}) = \boldsymbol{v}_{k-i}, \qquad t \in [t_{k-i}, t_{k-i+1}], \qquad i = 1, \dots, n_e \tag{2.14}$$

$$\boldsymbol{v}^{\boldsymbol{L}} \leq \boldsymbol{v}_{k-i} \leq \boldsymbol{v}^{\boldsymbol{U}}, \qquad i = 1 \dots n_e \tag{2.15}$$

$$\Delta\boldsymbol{y}_{k-i} = \boldsymbol{y}_{k-i} - \boldsymbol{y}_{P,k-i}, \qquad i = 0, \dots, n_e - 1 \tag{2.16}$$

$$\Delta\boldsymbol{x}_{k-n_e} = \boldsymbol{x}_{k-n_e} - \widehat{\boldsymbol{x}}_{k-n_e} \tag{2.17}$$

Problem (2.10)-(2.17), has also been formulated in continuous time, and it is executed at each sampling time k, using the same nonlinear model (2.11), (2.12) of the eMPC+MA problem, considering a past horizon $t \in \left[t_{k-n_e}, t_k\right]$. In this past horizon, the control variables $\boldsymbol{u}_{k-i}$ applied to the process in $[t_{k-i}, t_{k-i+1}]$, and the process measurements collected in t_{k-i}, i.e., $\boldsymbol{y}_{P,k-i}$, are known. The past horizon of the MHE is illustrated in Figure 2.2.

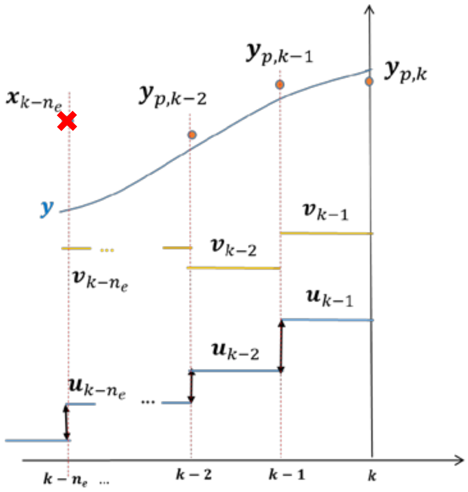


Figure 2.2: Past values for the MHE estimation.

Note that the decision variables of the MHE problem are the values of the states at time t_{k-n_e} ($\boldsymbol{x}_{k-n_e}$), represented by a red “x” in Figure 2.2, and the past disturbances $\boldsymbol{v}_{k-i}, i = 1, \dots, n_e$, represented by yellow lines in Figure 2.2. The MHE assumes that if $\boldsymbol{u}_{k-i}$ and $\boldsymbol{v}_{k-i}$ were applied to the model (2.11) and (2.12) starting from $\boldsymbol{x}_{k-n_e}$, then the corresponding model output at sampling times $k-i, \ i = 0, \dots, n_e - 1$, i.e., $\boldsymbol{y}_{k-i}$, must be as close as possible to $\boldsymbol{y}_{P,k-i}$. This aim corresponds to the first term in the cost function (2.10), which also incorporates two additional terms: the last one minimizes the magnitude of the estimated disturbances $\boldsymbol{v}_{k-i}$, while the second one is a prior weighting that penalizes the distance of the decision variable $\boldsymbol{x}_{k-n_e}$, with respect to the one that was previously estimated at sampling time $k-n_e$ ($\widehat{\boldsymbol{x}}_{k-n_e}$), as in (2.17). $\boldsymbol{Q}_x$, $\boldsymbol{Q}_v$, and $\boldsymbol{Q}_y$ are positive definite matrices, with weighting and normalization factors. The problem also includes inequality constraints to bind the disturbances in the allowed range (2.15), as well as others (2.13) to avoid unwanted values of the variables.

The solution of problem (2.10)-(2.17) gives $\boldsymbol{x}_{k-n_e}^*$ and $\boldsymbol{v}_{k-i}^*, i=1\ldots n_e$. This implies that once problem (2.10)-(2.17) has been solved, its solution can be used to estimate the initial value of the model state at time t_k, i.e., $\widehat{\boldsymbol{x}}_k$, which is required by the eMPC+MA problem in (2.9). To obtain $\widehat{\boldsymbol{x}}_k$ the model equations (2.18) must be integrated over $t\in\left[t_{k-n_e},t_k\right]$ starting from $\boldsymbol{x}_{k-n_e}^*$, using the estimated disturbances $\boldsymbol{v}_{k-i}^*$, $i=1,\ldots,n_e$, and applying $\boldsymbol{u}_{k-i}$.

$$\boldsymbol{f}(\dot{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}^*)=0\ ,\ \forall t\in\left[t_{k-n_e},t_k\right],\ \boldsymbol{x}\left(t_{k-n_e}\right)=\boldsymbol{x}_{k-n_e}^* \tag{2.18}$$

Once (2.18) has been solved, the states calculated at time t_k are defined as the initial value for problem (2.1)-(2.9) as it is stated in (2.19). Note that is (2.9) is similar to equation (2.19), the difference being the order of assignment.

$$\widehat{\boldsymbol{x}}_k=\boldsymbol{x}(t_k) \tag{2.19}$$

The value of the disturbances $\boldsymbol{v}_k$ required in the eMPC+MA problem (2.1)-(2.9), is taken as the estimation given by the MHE at the last sampling period $\boldsymbol{v}_{k-1}^*$, this is $\boldsymbol{v}_k=\boldsymbol{v}_{k-1}^*$.

2.3 MA modifiers estimation using transient measurements

When proposing a method for on-line estimation of process derivatives, the overall performance of the algorithms must be taken into account. Accurate estimation of the static process gradients, which requires collecting information about the process in this state, allows precise detection of the true optimum of the process. However, the time spent in the estimation is an important issue, since it affects the time of operation in suboptimal conditions. It may even make the implementation of MA impractical for processes with long settling times when process gradients are estimated with steady-state data. Then, the use of transient measurements seems to be an interesting alternative to reduce the convergence time for all the RTO+MA+MPC approaches already mentioned in Section 1.6.

The MA modifiers estimation module in Figure 2.1 is responsible to calculate the MA modifiers to be used in the controller. In this thesis, two different algorithms have been developed. The first one is the Dynamic Modifier Estimation (DME) and the second one is the Transient MA (TMA). This module will have different steps depending on the algorithm chosen:

1. Choose between the process gradient estimation (TMA) or modifier estimation (DME).
2. If the TMA is selected:
    a. Estimate the process gradient using TMA algorithm
    b. Calculate the model gradient using the finite difference method
    c. Calculate the modifiers as in (1.8) and (1.9)
3. If DME is used
    a. Calculate the modifiers using the DME algorithm
4. Calculate the zero order modifiers as in (1.10)
5. Apply the filter as in (1.12)

In the next chapters 3 and 4, the DME and TMA methods are explained in detail with the results of the application to the Williams-Otto Reactor case study. In a few words, DME uses an on-line optimization algorithm to make the estimated modifiers converge to the steady-state modifiers required by MA. TMA estimates directly the dynamic gradients using the available transient measurements and an identification algorithm and makes small corrections to improve the process performance. Neither method requires an input-output process model to compute the process gradients or modifiers from transient data. Table 2.1 shows the main features of the two algorithms.

Table 2.1: Main features of the proposed algorithms.

	Dynamic Modifier Estimation	**Transient MA**
Type of data	Transient measurements	Transient measurements
Type of problem	Optimization problem	Identification Algorithm
Result of the problem	Estimation of MA modifiers	Estimation of dynamic process gradients

3 Dynamic Modifier Estimation (DME)

This chapter presents an algorithm to estimate the MA modifiers during the transient, called Dynamic Modifier Estimation (DME). DME is used in the eMPC+MA architecture presented earlier and is applied to a benchmark example, the Williams-Otto reactor.

3.1 DME algorithm

DME has the task of computing the modifiers $\boldsymbol{\lambda}_k, \boldsymbol{\gamma}_k$ used in the eMPC+MA module (Figure 2.1), while $\boldsymbol{\varepsilon}_k$ can be directly computed from (1.10). From their definitions in (1.8) and (1.9), several approaches have been proposed in the literature to compute modifiers based on the estimation of the process gradients from available measurements. As mentioned above, the use of transient measurements to estimate modifiers opens the door to the application of MA in slow dynamic processes. Win this sense, the MA modifiers could be seen as correction terms that are added to the economic optimization problem so that the modified problem solution matches the process optimum. From this point of view, they can be estimated directly without explicitly computing process and model gradients. DME will try to estimate the modifiers directly using historical data, which, after filtering, will be incorporated into the eMPC+MA module.

The idea behind the DME approach is explained next, considering only the economic cost function in a continuous form for simplicity. Consider a vector $\boldsymbol{\theta}$ such that, if given an appropriate value, the modified cost function appearing in equation (3.1) reproduces the measured process cost function ϕ_p over the past time interval $t = [t_{k-n_d}, t_k]$ as in Figure 3.1. n_d is a small integer number, the DME past horizon, which may or may not coincide with the past horizon of the MHE.

$$\phi_p(t) \approx \phi(t) + \boldsymbol{\theta}(t)^T \Delta\boldsymbol{u}(t) + \Delta\boldsymbol{u}(t)^T \boldsymbol{Q} \Delta\boldsymbol{u}(t) + \sum_{j=1}^{n_u-1} \Delta\boldsymbol{u}_j(t)^T \boldsymbol{Q} \Delta\boldsymbol{u}_j(t) \qquad (3.1)$$

Where $\phi_p(t)$ and $\phi(t)$ represent, respectively, the measured process cost function at a given time t in the past interval, and the model-estimated cost obtained with the $\Delta\boldsymbol{u}$ applied to the process in the interval of the DME past horizon. Both ϕ_p and ϕ are time functions obtained either by interpolation of measured values or by model integration. In equation (3.1), the term $\Delta\boldsymbol{u}(t)^T\boldsymbol{Q}(t)\Delta\boldsymbol{u}(t)+\sum_{j=1}^{n_u-1}\Delta\boldsymbol{u}_j(t)^T\boldsymbol{Q}\Delta\boldsymbol{u}_j(t)$ corresponds to the optimal value of the quadratic term in (2.1), computed in the resolution of the eMPC+MA at the past time $t=[t_{k-n_d},t_k]$. The control moves $\Delta\boldsymbol{u}(t)$ in $\boldsymbol{\theta}(t)^T\Delta\boldsymbol{u}(t)$ and $\Delta\boldsymbol{u}(t)^T\boldsymbol{Q}\Delta\boldsymbol{u}(t)$ are those actually applied to the process in the past, and the summation term is the future moves calculated but not applied to the process due to the moving horizon strategy of the eMPC+MA.

Considering that the output variables are functions of the manipulated variables $\boldsymbol{u}$, one can compute the derivative of both sides of (3.1) w.r.t. $\boldsymbol{u}$ applied to the process and obtain (3.2). Note that the derivative of the summation term in (3.1) is zero because this term does not have the variable $\boldsymbol{u}$ applied to the process:

$$\frac{\partial\phi_p}{\partial\boldsymbol{u}}\approx\frac{\partial\phi}{\partial\boldsymbol{u}}+\boldsymbol{\theta}^T+2\Delta\boldsymbol{u}^T\boldsymbol{Q}_u \tag{3.2}$$

Now, if we rearrange equation (3.2) and use the definition of the modifiers, we could obtain an estimation of $\boldsymbol{\lambda}$ using (3.3) at any transient time by calculating the vector $\boldsymbol{\theta}$ that matches the past process cost as in (3.1).

$$\boldsymbol{\lambda}^T=\frac{\partial\phi_p}{\partial\boldsymbol{u}}-\frac{\partial\phi}{\partial\boldsymbol{u}}\approx\boldsymbol{\theta}^T+2\Delta\boldsymbol{u}^T\boldsymbol{Q}_u \tag{3.3}$$

In this way, we could consider the value of $\boldsymbol{\theta}^T+2\Delta\boldsymbol{u}^T\boldsymbol{Q}_u$ at t_k as an estimation of the cost modifiers at k. Thus, when the system reaches steady state, $\boldsymbol{\theta}^T+2\Delta\boldsymbol{u}^T\boldsymbol{Q}_u$ will correspond to the steady-state modifiers.

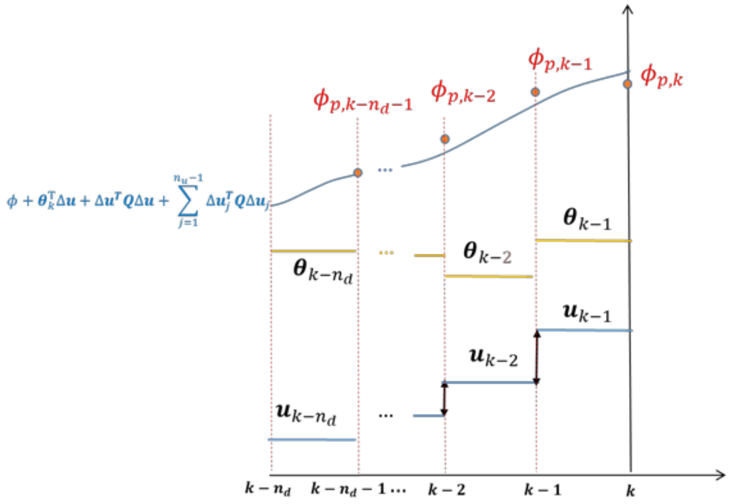

$\phi_{p,k-n_d-1}$
$\phi_{p,k-2}$
$\phi_{p,k-1}$
$\phi_{p,k}$
$\phi + \boldsymbol{\theta}_k^T \Delta \boldsymbol{u} + \Delta \boldsymbol{u}^T \boldsymbol{Q} \Delta \boldsymbol{u} + \sum_{j=1}^{n_u-1} \Delta \boldsymbol{u}_j^T \boldsymbol{Q} \Delta \boldsymbol{u}_j$
$\boldsymbol{\theta}_{k-1}$
$\boldsymbol{\theta}_{k-2}$
$\boldsymbol{\theta}_{k-n_d}$
$\boldsymbol{u}_{k-1}$
$\boldsymbol{u}_{k-2}$
$\boldsymbol{u}_{k-n_d}$
...
$k-n_d$
$k-n_d-1$ ...
$k-2$
$k-1$
k

Figure 3.1. Past process cost function and evolution of the extended model cost function corresponding to past values of control actions and values of $\boldsymbol{\theta}_{k-i}$ and $\boldsymbol{u}_{k-i}$.

Finding such values of $\boldsymbol{\theta}$ can be formulated as an optimization problem in the interval $\left[t_{k-n_d}, t_k\right]$, where the cost process evolution ϕ_p and the applied control moves are known. The cost function of the optimization problem would include the squared difference between the process cost and the RHS of equation (3.1) in the previous sampling times. Then, the value of the modifiers (3.3) corresponding to the last interval $[t_{k-1}, t_k]$ is used in the eMPC+MA problem to compute the next optimal control moves. The same idea can be extended to compute the constraints modifiers $\boldsymbol{\gamma}$, using in the cost function the squared difference between the process and the modified constraint from (2.4).

Formally, the DME problem can be formulated as in (3.4)-(3.9) for $\boldsymbol{\lambda}_k$, and (3.11)-(3.16) for $\boldsymbol{\gamma}_k$. At iteration k we know the past actions applied $\boldsymbol{u}_{k-i}$ and the vectors $\boldsymbol{\lambda}_{k-i}$ and $\boldsymbol{\gamma}_{k-i}$ used in previous iterations, as well as the values of the process cost function $\phi_{p,k-i}$ and the constraints $\boldsymbol{g}_{p,k-i}$ obtained. The only unknowns are $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}^{DME}$, which are conveniently parameterized. In (3.4)-(3.9) and (3.11)-(3.16), n_d is the DME past horizon.

$$\min_{\substack{\boldsymbol{\theta}_{k-i-1},\\ i=0\ldots n_d-1}} \sigma_\phi \int_{t_{k-n_d}}^{t_k} \Delta\phi^2 dt + \sum_{i=0}^{n_d-1} (\Delta\boldsymbol{\theta}_{k-i-1}^T \boldsymbol{Q}_\theta \Delta\boldsymbol{\theta}_{k-i-1}) \tag{3.4}$$

$$\begin{gathered} s.t.\ \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}_k) = \boldsymbol{0} \\ \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}, \boldsymbol{v}_k) = \boldsymbol{0} \end{gathered} \tag{3.5}$$

$$\begin{gathered} \Delta\phi(t) = \phi_p(t) - [\phi(t) + \boldsymbol{\theta}(t)^T \Delta\boldsymbol{u}(t) + \boldsymbol{S}(t)] \\ \boldsymbol{S}(t) = \Delta\boldsymbol{u}(t)^T \boldsymbol{Q}_{\boldsymbol{u}} \Delta\boldsymbol{u}(t) \end{gathered} \tag{3.6}$$

$$\begin{gathered} \boldsymbol{\theta}(t) = \boldsymbol{\theta}_{k-i-1}, \qquad t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \ldots, n_d - 1 \\ \boldsymbol{\theta}_{k-i-1} = \bar{\bar{\boldsymbol{\theta}}}_{k-i-1} - \Delta\boldsymbol{\theta}_{k-i-1}\ , \qquad i = 0, \ldots, n_d - 1 \end{gathered} \tag{3.7}$$

$$\begin{gathered} \boldsymbol{u}(t) = \boldsymbol{u}_{k-i-1}\ , t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \ldots, n_d - 1 \\ \Delta\boldsymbol{u}_{k-i-1} = \boldsymbol{u}_{k-i} - \boldsymbol{u}_{k-i-1}\ , \quad i = 0,1, \ldots n_d - 1 \end{gathered} \tag{3.8}$$

$$\boldsymbol{x}(t_{k-n_d}) = \hat{\boldsymbol{x}}_{k-n_d} \tag{3.9}$$

In the first term of the cost function (3.4), expanded in (3.6), $\boldsymbol{S}$ is the value of the quadratic term that penalizes changes in the manipulated variable from (2.1) solved in the previous iteration. σ_ϕ is a scalar weight for this difference. Note that (3.4), (3.6) also include another quadratic term that tries to smooth the changes over the values computed in the previous iterations (3.7) and helps to convexify the problem, where $\boldsymbol{Q}_\theta \in \mathbb{R}^{\boldsymbol{n_u} \times \boldsymbol{n_u}}$ a diagonal matrix with weighting factors for each manipulated variable. The optimization is subject to the dynamic model (3.5). The problem uses the value $\hat{\boldsymbol{x}}_{k-n_d}$ that was estimated by the MHE as the initial state (3.9) and includes the disturbances $\boldsymbol{v}_k$ to maintain consistency with the other modules. Note also that ϕ and ϕ_p are computed as continuous functions interpolating the past iterations $k - i$. $\bar{\bar{\boldsymbol{\theta}}}$ is the vector of optimal solutions of the problem (3.4)-(3.9) obtained in the last iteration.

The solution of (3.4)-(3.9), $\boldsymbol{\theta}_{k-1}^*$, is considered equal to $\boldsymbol{\theta}_k$ and allows to estimate the cost modifiers in iteration k using (3.10):

$$\boldsymbol{\lambda}_k^{DME} = \boldsymbol{\theta}_k{}^{\boldsymbol{T}} + 2\Delta\boldsymbol{u}_{k-1}^T \boldsymbol{Q}_{\boldsymbol{u}}, \qquad \boldsymbol{\theta}_k = \boldsymbol{\theta}_{\boldsymbol{k-1}}^* \tag{3.10}$$

In the same way, we can formulate a similar problem (3.11)-(3.16) to compute the modifiers $\boldsymbol{\gamma}$ of each constraint. $\boldsymbol{\sigma}_g := \left[\sigma_{g,1}, \ldots, \sigma_{g,n_g}\right]^T$ is a column vector with the weight values for each constraint and $\sigma_\gamma^j \in \mathbb{R}^{n_u \times n_u}$ a diagonal matrix with the weight values for each constraint modifier, $j = 1 \ldots n_g$.

$$\min_{\substack{\boldsymbol{\gamma}_{k-i-1}^{DME} \\ i=0\ldots n_d-1}} \sum_{j=0}^{n_g} \sigma_{g,j} \int_{t_{k-n_d}}^{t_k} \Delta g_j{}^2 dt + \sum_{j=1}^{n_g} \sum_{i=0}^{n_d-1} \left(\Delta\boldsymbol{\gamma}_{k-i-1}^{j,DME^{T}} \sigma_\gamma^j I \Delta\boldsymbol{\gamma}_{k-i-1}^{j,DME} \right) \tag{3.11}$$

$$\begin{gathered} s.t.\ \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}_k) = \mathbf{0} \\ \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}, \boldsymbol{v}_k) = \mathbf{0} \end{gathered} \tag{3.12}$$

$$\Delta\boldsymbol{g}(t) = \boldsymbol{g}_P(t) - \boldsymbol{g}(t) - \boldsymbol{\gamma}^{DME}(t)\Delta\boldsymbol{u}(t), \qquad i = 0 \ldots n_d - 1 \tag{3.13}$$

$$\begin{gathered} \boldsymbol{\gamma}^{DME}(t) = \boldsymbol{\gamma}_{k-i-1}^{DME}, \qquad t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \ldots, n_d - 1 \\ \Delta\boldsymbol{\gamma}_{k-i}^{DME} = \boldsymbol{\gamma}_{k-i}^{DME} - \overline{\overline{\boldsymbol{\gamma}}}_{k-i}, \qquad i = 1, \ldots, n_d \end{gathered} \tag{3.14}$$

$$\boldsymbol{u}(t) = \boldsymbol{u}_{k-i-1}, \qquad t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \ldots, n_d - 1 \tag{3.15}$$

$$\boldsymbol{x}(t_{k-n_d}) = \hat{\boldsymbol{x}}_{k-n_d} \tag{3.16}$$

The solution to problems (3.11)-(3.16) is $\gamma_{k-1}^{DME^*}$. Similar to the estimation of $\boldsymbol{\theta}_k$, the value of γ_k^{DME} is taken to be equal to $\gamma_{k-1}^{DME^*}$. Therefore, when the system reaches steady state, γ_k^{DME} will be equal to the steady-state modifiers.

Finally, (1.10), (3.4)-(3.16) give the new values of the cost and constraint modifiers $\boldsymbol{\Lambda}_k^{DME} = (\boldsymbol{\lambda}_k^{DME}, \gamma_k^{DME}, \boldsymbol{\varepsilon}_k^{DME})$, which, after filtering (3.17), can be used as estimations of the modifiers at iteration k in the eMPC+MA problem. The value of $\boldsymbol{\varepsilon}_k^{DME}$ is obtained directly using (1.10), while the superscript "DME" has been added to differentiate this value from that applied to the eMPC+MA and MHE problems, obtained after filtering. The term $\boldsymbol{\Lambda}_{k-1}$ is the vector of filtered modifiers applied to the process in the previous iteration $k-1$ and α is a gain matrix.

$$\begin{gathered} \boldsymbol{\Lambda}_k = \boldsymbol{\alpha}\boldsymbol{\Lambda}_{k-1} + (\boldsymbol{I} - \boldsymbol{\alpha})\boldsymbol{\Lambda}_k^{DME} \\ \boldsymbol{\Lambda}_k = (\boldsymbol{\lambda}_k, \boldsymbol{\gamma}_k, \boldsymbol{\varepsilon}_{\boldsymbol{k}}) \end{gathered} \tag{3.17}$$

Where α can be represented by a block-diagonal matrix given by (3.18).

$$\boldsymbol{\alpha} := diag(b_1, \ldots, b_{n_u,} q_1\boldsymbol{I}_{n_u}, \ldots, q_{n_g}\boldsymbol{I}_{n_u}, d\boldsymbol{I}_{n_g}) \tag{3.18}$$

The gain entries $b_1, \ldots, b_{n_u,} q_1, \ldots, q_{n_g}, d$ are taken in (0,1]. n_g is the number of constraints $\boldsymbol{g}$ in the problem and n_u the number of manipulated variables.

eMPC+MA algorithm using DME to estimate MA modifiers.

i. Initialization

1. Collect $N \coloneqq \max\{n_d, n_e\}$ previous data (controller sampling time) of variables $\boldsymbol{u}_{k-i}$, $\boldsymbol{y}_{p,k-i}$, $\phi_{p,k-i}$, $\boldsymbol{g}_{p,k-i}$ where $i = 1, \dots, N$.

MHE module: given $\boldsymbol{Q}_x$, $\boldsymbol{Q}_y$ and $\boldsymbol{Q}_v$

2. Initialize $\widehat{\boldsymbol{x}}_{k-n_e}$
   a. For measured states, consider $\widehat{\boldsymbol{x}}^M_{k-n_e} = \boldsymbol{y}^S_{p,k-n_e}$, being $\widehat{\boldsymbol{x}}^M_{k-n_e}$ the subset of states that are measured, and $\boldsymbol{y}^S_{p,k-n_e}$ the subset of $\boldsymbol{y}_{p,k-n_e}$ containing the measured states in $k - n_e$
   b. For unmeasured states, use the values predicted by the model with $\boldsymbol{v}_{k-i} = \boldsymbol{0}, i = 1 \dots n_e$
3. Solve the MHE problem (2.10)-(2.17) to find the past values of the states $\boldsymbol{x}^*_{k-n_e}$ and disturbances $\boldsymbol{v}^*_{k-i}, i = 1 \dots n_e$.
4. Evaluate the estimated state $\widehat{\boldsymbol{x}}_k$ integrating (2.18) from t_{k-n_e} to t_k. Estimate the value of the disturbance using $\boldsymbol{v}_k = \boldsymbol{v}^*_{k-i}$
5. Update $\widehat{\boldsymbol{x}}_{k-n_e} = \boldsymbol{x}^*_{k-n_e}$.

DME module: given $\boldsymbol{\sigma}_\phi$, $\boldsymbol{Q}_\theta$, $\boldsymbol{\sigma}_g$, $\boldsymbol{\sigma}^j_\gamma$ where $j = 1 \dots n_g$, and $\boldsymbol{\alpha}$

6. Initialize previous values of $\boldsymbol{\lambda}_{k-i}$ and $\boldsymbol{\gamma}_{k-i}, i = 1 \dots n_d$, using an identification method such as recursive least squares. The idea here is to start with good estimators of the modifiers to improve the convergence of the DME module.
7. Evaluate $\boldsymbol{\varepsilon}_{k-1}$ using (1.10)
8. Solve problems (3.4) and (3.11) to obtain $\boldsymbol{\theta}^*_{k-1}, \gamma^{DME^*}_{k-1}$.
9. Use (3.10) to obtain $\boldsymbol{\lambda}^{DME}_k$ and (4.c) to obtain $\boldsymbol{\varepsilon}^{DME}_k$**.** Make $\gamma^{DME}_k = \gamma^{DME^*}_{k-1}$
10. Use (3.17) and get the filtered modifiers $\boldsymbol{\Lambda}_k$.

eMPC+MA module: given $\boldsymbol{Q}_u$

11. Go to step 19

ii. **For next iterations**

MHE module:

12. Collect n_e previous data of variables $\boldsymbol{u}_{k-i}$, $\boldsymbol{y}_{p,k-i}$, $\phi_{p,k-i}$, $\boldsymbol{g}_{p,k-i}$ with $i = 1, \dots, n_e$.
13. Solve problem (2.10)-(2.17) to find the past values of the states $\boldsymbol{x}^*_{k-n_e}$ and the disturbances $\boldsymbol{v}^*_{k-i}$.
14. Evaluate the estimated state $\hat{\boldsymbol{x}}_k$ integrating (2.18) from t_{k-n_e} to t_k. Estimate the value of the disturbance using $\boldsymbol{v}_k = \boldsymbol{v}^*_{k-i}$
15. Update $\hat{\boldsymbol{x}}_{k-n_e} = \boldsymbol{x}^*_{k-n_e}$

DME module:

16. Solve problems (3.4) and (3.11) to obtain $\boldsymbol{\theta}^*_{k-1}, \gamma^{DME^*}_{k-1}$.
17. Use (3.10) to obtain $\boldsymbol{\lambda}^{DME}_k$ and (1.10) to obtain $\boldsymbol{\varepsilon}^{DME}_k$. Make $\gamma^{DME}_k = \gamma^{DME^*}_{k-1}$
18. Use (3.17) and get the filtered modifiers $\boldsymbol{\Lambda}_k$.

eMPC+MA module:

19. Solve problem (2.1)-(2.9) using $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{v}_k$ from MHE, and $\boldsymbol{\Lambda}_k$ calculated in DME.
20. Apply $\Delta\boldsymbol{u}_k$ to the process.
21. Wait for the next sampling time and update $k = k + 1$
22. Go to step 12.

3.2 Implementation in the Williams-Otto reactor case study

The Williams-Otto reactor benchmark example (Williams and Otto, 1960) has been used by several authors to evaluate the performance of different RTO approaches. In this process, an intermediate component of the reactions is not considered in the model used in the optimization layer, which introduces an important structural model-process mismatch. In this section, the DME algorithm is used with the previously presented architecture of eMPC+MA to calculate the MA modifiers to make the process reach the true optimum.

3.2.1 Process Description

The process consists of a continuous stirred tank reactor (CSTR) in which the reactants A and B are combined to produce four species C, E, G, P in three different reactions. Figure 3.2 shows the schematic of the simulated reactor operating at temperature T. In Figure 3.2, F_A and F_B represent the volumetric flows of the feeds containing A and B at molar concentrations given by X_{A0} and X_{B0} respectively; F_R represents the volumetric flow of the reactor effluent with molar concentrations of the components denoted by $X_i, i \in \{A, B, C, E, G, P\}$. The system has two degrees of freedom (F_B and T) that can be manipulated by a supervisory layer, while F_A can be considered as a measured disturbance. Since the outlet of the reactor is located at the top of the vessel, the total reaction volume (V_R) can be considered constant. In addition, the mixer makes it possible to assume that the system behaves as an ideal CSTR. It is also assumed that the measurements of the molar concentrations of C and B in the effluent are not available, while the other variables are measured. Products P and E are the valuable ones, while and G is a by-product and C is an intermediate component of the reactions.

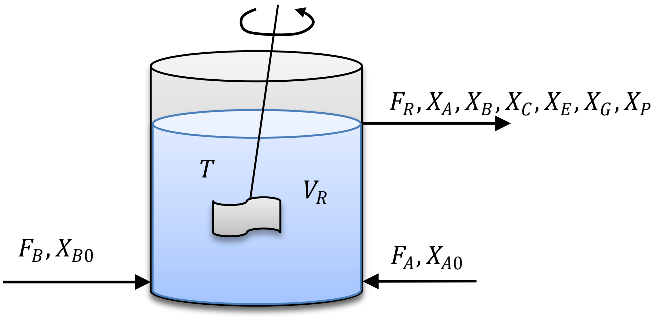


Figure 3.2: Schematic of the Williams-Otto reactor.

The process-model mismatch has been emulated using a simplified dynamic model to solve the model-based optimization in the controller. The simplified model considers only two reactions and neglects the existence of component C, which implies a structural mismatch. In addition, parametric uncertainty has been incorporated by using different values for the model parameters and the real process model parameters. The model that emulates the real process and the simplified model are based on mass balances for each component and the elemental kinetics with Arrhenius temperature dependence. The equations of both models are presented below.

3.2.2 Rigorous dynamic model

A rigorous dynamic model is constructed and it will be used as a virtual plant. The three reactions are considered:

$$A + B \overset{k_1}{\rightarrow} C \qquad\qquad B + C \overset{k_2}{\rightarrow} P + E \qquad\qquad C + P \overset{k_3}{\rightarrow} G$$

The rate of each reaction is given by equations (3.19)-(3.21). The rate constant of each reaction, k_i, is described by the Arrhenius equation (3.22), which depends on the pre-exponential factor $k_{i,0}$, the activation energy E_{ai} for each reaction, and T.

$$r_1 = k_1 X_A X_B \tag{3.19}$$

$$r_2 = k_2 X_B X_C \tag{3.20}$$

$$r_3 = k_3 X_C X_P \tag{3.21}$$

$$k_i = k_{i,0} e^{-(E_{ai}/T)}, \qquad i = 1,2,3 \tag{3.22}$$

A cooling system maintains the temperature of the reactor around the desired value *T*. It is assumed that no significant change in the density or similar physical properties takes place.

The nonlinear dynamics of the process is given by equations (3.23)-(3.28), which, together with the total mass balance (3.29), are used in a simulation of the real plant. Table 3.1 presents the parameters used in the simulation.

$$V_R \frac{dX_A}{dt} = F_A X_{A0} - F_R X_A - V_R r_1 \tag{3.23}$$

$$V_R \frac{dX_B}{dt} = F_B X_{B0} - F_R X_B - V_R r_1 - V_R r_2 \tag{3.24}$$

$$V_R \frac{dX_C}{dt} = -F_R X_C + V_R r_1 - V_R r_2 - V_R r_3 \tag{3.25}$$

$$V_R \frac{dX_E}{dt} = -F_R X_E + V_R r_2 \tag{3.26}$$

$$V_R \frac{dX_G}{dt} = -F_R X_G + V_R r_3 \tag{3.27}$$

$$V_R \frac{dX_P}{dt} = -F_R X_P + V_R r_2 - V_R r_3 \tag{3.28}$$

$$F_R = F_A + F_B \tag{3.29}$$

Table 3.1: Parameters of Williams-Otto reactor real process.

$k_{1,0}$	9.9594×10^6	$l/(mol\,min)$
$k_{2,0}$	8.66124×10^9	$l/(mol\,min)$
$k_{3,0}$	1.6047×10^{13}	$l/(mol\,min)$
E_{a1}	6666.7	K
E_{a2}	8333.3	K
E_{a3}	11111	K
X_{A0}	10	mol/l
X_{B0}	10	mol/l
V_R	2105	l
F_A	112.35	l/min

3.2.3 Simplified dynamic model

In the simplified model used in the supervisory layer, only five components and two reactions are considered:

$$A + 2B \xrightarrow{k_{1\_model}} P + E \quad A + B + P \xrightarrow{k_{2\_model}} G$$

The rate of each reaction is given by equations (3.30) and (3.31). The rate constant of each reaction, k_{i_appr}, is described by the Arrhenius equation (3.32), which depends on the pre-exponential factor, $k_{i,0\_model}$, the activation energy E_{ai_model} for each reaction, and the temperature inside the reactor, T.

$$r_{1\_model} = k_{1\_appr} X_A X_B^2 \tag{3.30}$$

$$r_{2\_model} = k_{2\_appr} X_A X_B X_P \tag{3.31}$$

$$k_{i\_appr} = k_{i,0\_model} e^{-(E_{ai\_model}/T)}, \qquad i = 1,2 \tag{3.32}$$

The dynamic model includes the following equations, which represent the mass balances of the various components considered:

$$V_R \frac{dX_A}{dt} = F_A X_{A0} - F_R X_A - V_R r_{1_{model}} - V_R r_{2_{model}} \tag{3.33}$$

$$V_R \frac{dX_B}{dt} = F_B X_{B0} - F_R X_B - 2 V_R r_{1\_model} - V_R r_{2\_model} \tag{3.34}$$

$$V_R \frac{dX_E}{dt} = -F_R X_E + V_R r_{1\_model} \tag{3.35}$$

$$V_R \frac{dX_G}{dt} = -F_R X_G + V_R r_{2\_model} \tag{3.36}$$

$$V_R \frac{dX_P}{dt} = -F_R X_P + V_R r_{1_{model}} - V_R r_{2_{model}} \quad (3.37)$$

$$F_R = F_A + F_B \quad (3.38)$$

Model parameters are summarized in Table 3.2.

Table 3.2: Parameters of Williams-Otto model.

$k_{1,0\_model}$	1.3134×10^8	$l/(mol\, min)$
$k_{2,0\_model}$	2.586×10^{13}	$l/(mol\, min)$
E_{a1_model}	8077.6	K
E_{a2_model}	12438.5	K
X_{A0}	10	mol/l
X_{B0}	10	mol/l
V_R	2105	l
F_A	112.35	l/min

3.2.4 Comparison between process and model optimum

The purpose of the optimal operation is to maximize the operating profit, which is computed as the difference between the value of the useful products E and P and the cost of the raw materials A and B. Table 3.3 gives the prices of these products. If a perfect steady state model of the process were available and there were no disturbances, the optimal operating point could be obtained by solving the optimization problem (3.39). Here, the inflow F_B and the reactor temperature T are constrained to operate within a certain range.

$$\begin{gathered} \max_{\boldsymbol{u}=[F_B,T]} \phi = F_R(X_P p_P + X_E p_E) - F_A X_{A0} p_A - F_B X_{B0} p_B \\ s.t.\, equations\ (3.23) - (3.28)\ in\ steady\ state \\ 180\ l/min \leq F_B \leq 360\ l/min \\ 75ºC \leq T \leq 100º\mathrm{C} \\ 0 \leq X_A \leq 1.2\ \mathrm{mol}/l \\ 0 \leq X_G \leq 0.5\ \mathrm{mol}/l \end{gathered} \quad (3.39)$$

The real optimum of the process in steady state is $F_B = 293.55\ \mathrm{l/min}$ and $T = 89.98$ ºC and the value of the objective function is $\phi = 11594.40€$ (black point in Figure 3.3). As can be seen in Figure 3.3, which displays the value of the benefit ϕ for a range of values of the manipulated variables, the cost is in a relatively flat zone, which can cause convergence problems for the optimizers.

Table 3.3: Cost of the reactants and price of products.

p_A	$7.623\ €/mol$
p_B	$11.434\ €/mol$
p_P	$114.338\ €/mol$
p_E	$5.184\ €/mol$

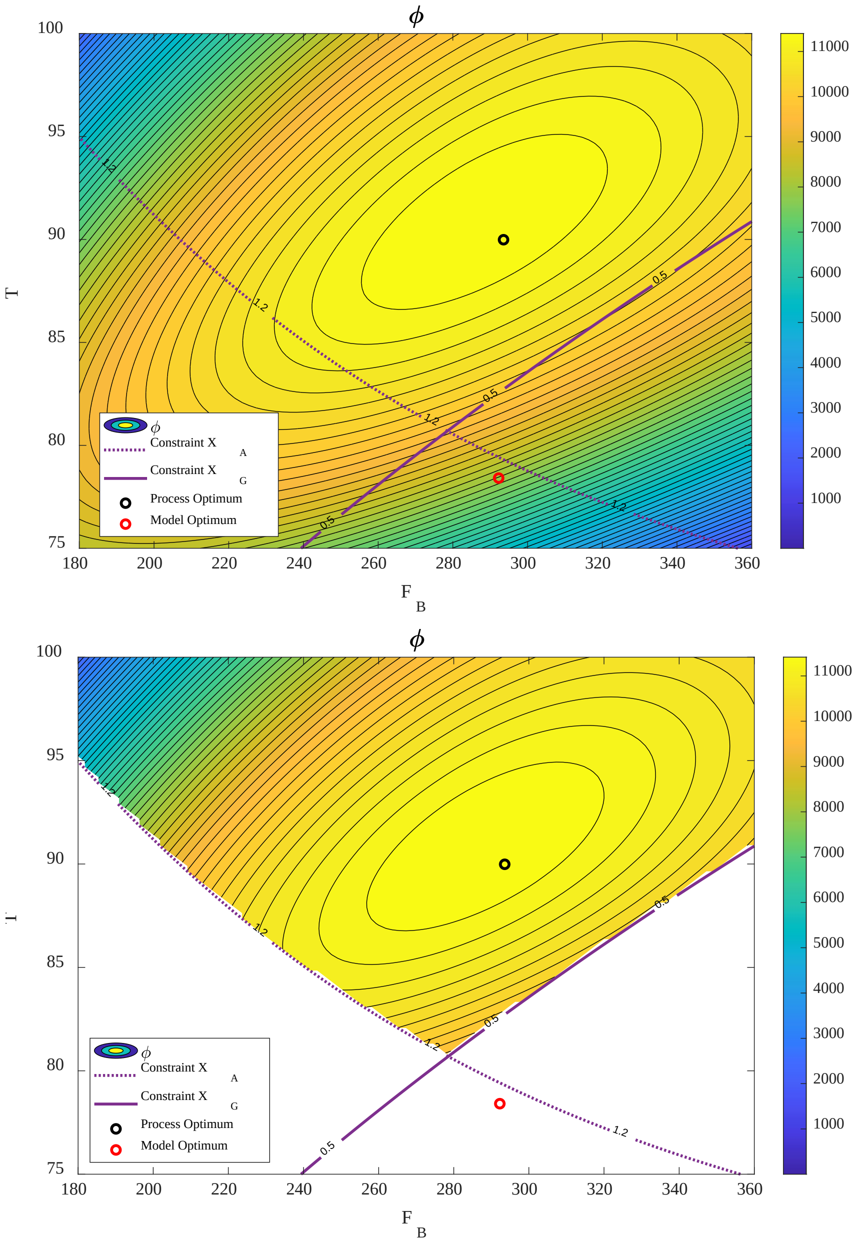


Figure 3.3. Objective function values and feasible region of problem (3.39). The contour 1.2 (dotted line) and 0.5 (solid line) are the problem constraints in (3.39) for X_A and X_G respectively. The black and red dots correspond to the plant and model optimum, respectively.

Actually, the mapping between input-output is unknown so, an RTO formulation for this process should be like problem (3.40). The model optimum in steady state is

$F_B = 292.26\ \mathrm{l/min}$ and $T = 78.41$ ºC and the value of the objective function in the real process is $\phi = 9075.3€$ (red dot in Figure 3.3).

$$\begin{gathered}
\max_{\boldsymbol{u}=[F_B,T]} \phi = F_R(X_P p_P + X_E p_E) - F_A X_{A0} p_A - F_B X_{B0} p_B \\
s.t.\, equations\ (3.30) - (3.37)\ in\ steady\ state \\
180\ l/min \leq F_B \leq 360\ l/min \\
75ºC \leq T \leq 100ºC \\
0 \leq X_A \leq 1.2\ \mathrm{mo}l/l \\
0 \leq X_G \leq 0.5\ \mathrm{mo}l/l
\end{gathered} \qquad (3.40)$$

3.2.5 eMPC+MA + DME

eMPC+MA solves the modified problem given by (3.41). The economic objective function considers the variables as computed at the end of the prediction horizon (denoted by a bar above the variables in part A of ϕ_M). The states $\bar{X}_P$ and $\bar{X}_E$ are predicted using equations (3.30)-(3.37), adding the disturbances $\boldsymbol{v}$ from MHE. The two terms with λ_1 and λ_2 correspond to the MA cost modifiers for each control variable in part B of ϕ_M. The last two terms are the penalties on the control efforts to smooth the control actions and contribute to model adequacy and convexity in part C.

$$\begin{gathered}
\max_{\boldsymbol{u}=[F_B,T]} \phi_M = \underbrace{\bar{F}_R\big((\bar{X}_P)p_P + (\bar{X}_E)p_E\big) - F_A X_{A0} p_A - \bar{F}_B X_{B0} p_B}_{A} \\
+ \underbrace{\lambda_{1,k}\big(\bar{F}_B - F_{B,k-1}\big) + \lambda_{2,k}(\bar{T} - T_{k-1})}_{B} \\
\underbrace{+ \sigma_{u_1} \sum_{i=1}^{n_u-1} \big(\Delta F_{B,k+i}\big)^2 + \sigma_{u_2} \sum_{i=1}^{n_u-1} (\Delta T_{k+i})^2}_{C} \\
s.t.\, equations\ (3.30) - (3.37)\ \text{with disturbances}\ v \\
\varDelta F_{B,k+i} = F_{B,k+i} - F_{B,k+i-1}, \qquad i = 0 \ldots n_{pred} - 1 \\
\varDelta T_{k+i} = T_{k+i} - T_{k+i-1}, \qquad i = 0 \ldots n_{pred} - 1 \\
F_B(t) = F_{B_{k+n_u}} \ ,\ t \in \big[t_{k+n_u}, t_{pred}\big] \\
T(t) = T_{k+n_u} \ , t \in \big[t_{k+n_u}, t_{pred}\big] \\
180 \frac{l}{min} \leq F_{B,k+i} \leq 360 \frac{l}{min}, \qquad i = 0 \ldots n_{pred} - 1 \\
75ºC \leq T_{k+i} \leq 100ºC, \qquad i = 0 \ldots n_{pred} - 1
\end{gathered} \qquad (3.41)$$

Note that the cost function has two objectives: to maximize the benefit at the end of the prediction horizon and to achieve a smooth and stable operation. This compromise is given by the weighting factors β_i. The MHE module provides the initial value for the state and disturbances at the current iteration k.

Similarly, the DME problem solved at each sampling time k is given by (3.42). Here, ϕ_P refers to the value measured in the process, as well as the concentrations $X_{p,j}$ (for each component j), while ϕ is a value estimated from the model (3.30)-(3.37) with disturbances. ϕ_P is calculated as a continuous function of time, interpolating the measurements in the interval $[t_{k-3}, t_k]$. $\Delta\boldsymbol{u}$ and $\boldsymbol{\theta}$ are also calculated as continuous function of time t, using the equation in (3.42).

$$\min_{\substack{\boldsymbol{\theta}_{k-i-1} \\ i=0\ldots n_d-1}} \sigma_\phi \int_{t_{k-n_d}}^{t_k} \Delta\phi^2 dt + (\Delta\boldsymbol{\theta}_{k-i-1}^T \boldsymbol{Q}_\theta \Delta\boldsymbol{\theta}_{k-i-1})$$

$$\begin{gathered}
s.t. equations\ (3.30) - (3.37)\ \text{ with disturbances } v \\
\Delta\phi(t) = \big(\phi_p(t) - (\phi(t) + \boldsymbol{\theta}^T \Delta\boldsymbol{u}(t) + \Delta\boldsymbol{u}(t)^T \boldsymbol{Q}_u \Delta\boldsymbol{u}(t))\big) \\
\phi_p(t) = F_R\big(X_{p,P}(t)p_P + X_{p,E}(t)p_E\big) - F_A X_{A0} p_A - F_B X_{B0} p_B \\
\phi(t) = F_R(X_P(t)p_P + X_E(t)p_E) - F_A X_{A0} p_A - F_{B,k-i} X_{B0} p_B \\
\boldsymbol{u}_{k-i-1} = \left[F_{B,k-i-1}\ T_{k-i-1}\right]^T, \qquad i = 0 \ldots n_d - 1 \\
\boldsymbol{u}(t) = \boldsymbol{u}_{k-i-1}, t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \ldots, n_d - 1 \\
\boldsymbol{\theta}(t) = \boldsymbol{\theta}_{k-i-1}, \qquad t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \ldots, n_d - 1 \\
\Delta\boldsymbol{\theta}_{k-i-1} = \boldsymbol{\theta}_{k-i-1} - \bar{\bar{\boldsymbol{\theta}}}_{k-i-1-1}, \qquad i = 0 \ldots n_d - 1
\end{gathered} \tag{3.42}$$

The solution $\boldsymbol{\theta}_{k-1}^*$ of (3.42) after filtering (3.17) is used to compute $\boldsymbol{\lambda}_k$ as in equation (3.10). Now, the problem (3.41) has all the information it needs to be solved.

3.2.6 Results

The results of the proposed method are given next. The starting point is given in Table 3.4 and corresponds to a value of the process states far from the optimum.

Table 3.4: Starting point of the simulation.

X_A	0.8264	mol/l
X_B	4.8075	mol/l
X_C	0.0843	mol/l
X_E	1.2618	mol/l
X_G	0.2059	mol/l
X_P	1.0558	mol/l
F_B	350	l/min
T	82	ºC

The eMPC+MA problem could be solved numerically using either simultaneous or sequential approaches. The first one implies a full discretization of the model, e.g. using orthogonal collocation on finite elements, which transforms the dynamic optimization into an NLP problem that can be solved with algorithms such as IPOPT (Wächter and Biegler, 2006). This approach significantly increases the size of the problem to be solved, but allows a good treatment of path constraints and facilitates the use of automatic differentiation to calculate exact derivatives. Depending on the specific case considered, the sequential approach may also offer a good alternative by combining a dynamic simulation of the model with an NLP solver like SNOPT (Gill et al., 2005). An advantage of this method is the smaller size of the optimization problem and the fact that it does not require state discretization, but it requires integration of the extended system to compute exact derivatives, and it is more difficult to deal with path constraints and unstable systems. For the MHE problem, similar numerical methods can be used. In the presented case study, a sequential approach was used.

Table 3.5: Parameters used in the simulation for eMPC+MA+DME.

Module	**Parameter**	**Value**
MHE	σ_y $(\boldsymbol{Q}_y = \sigma_y \boldsymbol{I})$	250
MHE	σ_x $(\boldsymbol{Q}_x = \sigma_x \boldsymbol{I})$	1×10^{-2}
MHE	σ_v $(\boldsymbol{Q}_v = \sigma_v \boldsymbol{I})$	1×10^{-5}
MHE	n_e	3
DME	σ_ϕ	10
DME	$\boldsymbol{Q}_\theta$	$\begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$
DME	n_d	3
eMPC+MA	σ_{u_1}	0.05
eMPC+MA	σ_{u_2}	0.44
eMPC+MA	α	0.95
eMPC+MA	n_u	3
eMPC+MA	n_{pred}	30

The reactor and the three dynamic optimization problems eMPC+MA, MHE and DME, are formulated in the continuous domain in the simulation environment EcosimPro/Proosis (EA Int., 2020), a modern object-oriented software. They are solved with a sampling time of 2 minutes. The optimization problems have been solved using a sequential approach in which a dynamic simulation of the model is connected to SNOPT, a reduced-space SQP optimization algorithm. Exact derivatives of cost functions and constraints are generated by integrating of the extended system as provided by the IDAS integration software. The entire problem is solved in an average time of 3 seconds per iteration on a PC running Windows10 with a four-core i7 processor running at 3GHz and with 16 GB of memory. The eMPC+MA module starts at time $t = 8\, min$ to collect past measurements. The MHE, DME and eMPC+MA parameters used are listed in Table 3.5.

Figure 3.4 shows the states estimated by the MHE (blue line) along with the measured values from the process (red line). The estimated disturbances for the five states of the model appear at the bottom.

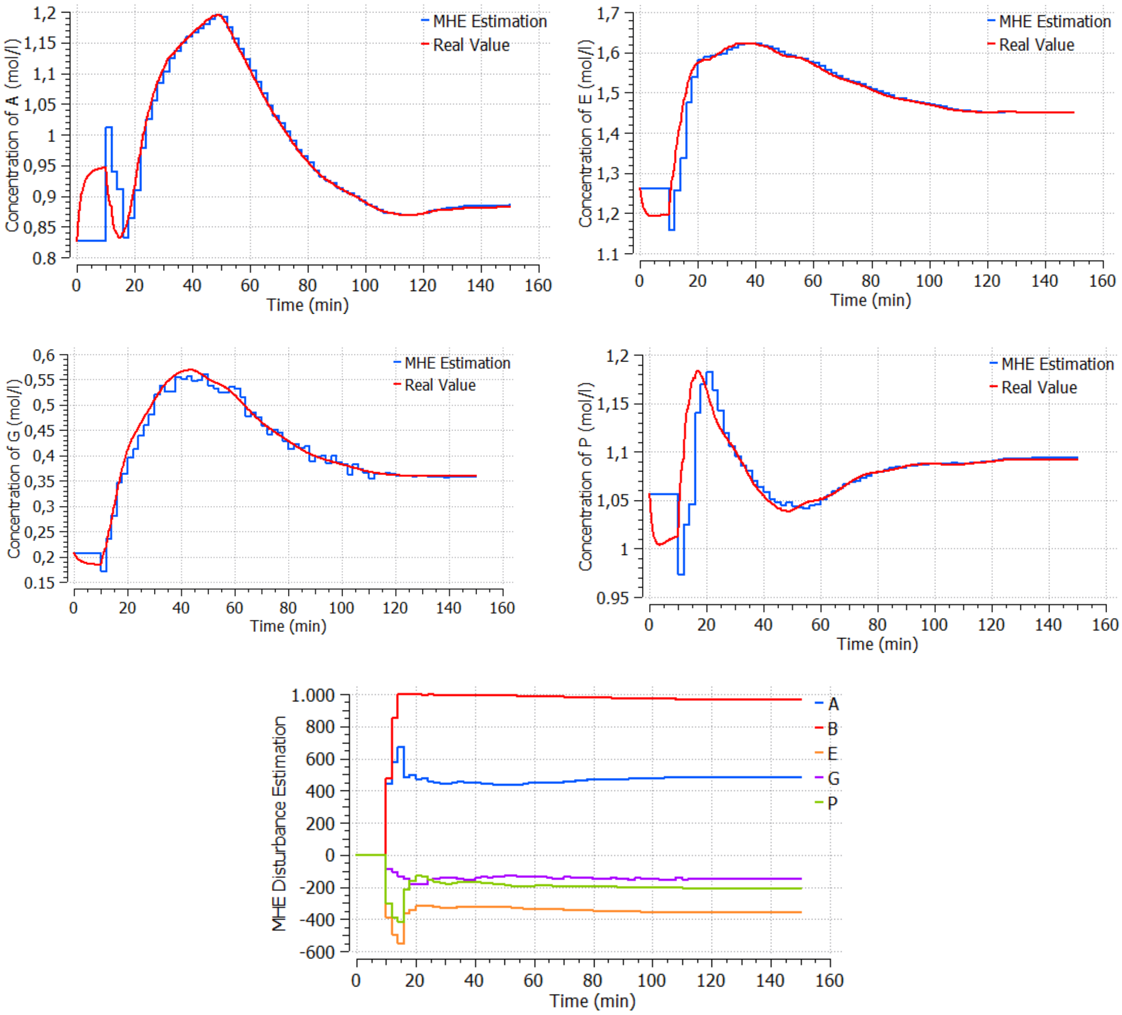


Figure 3.4. Results of the MHE module

The lambda modifiers computed by DME after filtering are presented in Figure 3.5. Figure 3.6 shows the time evolution of the process cost function and Figure 3.7 shows the manipulated variables.

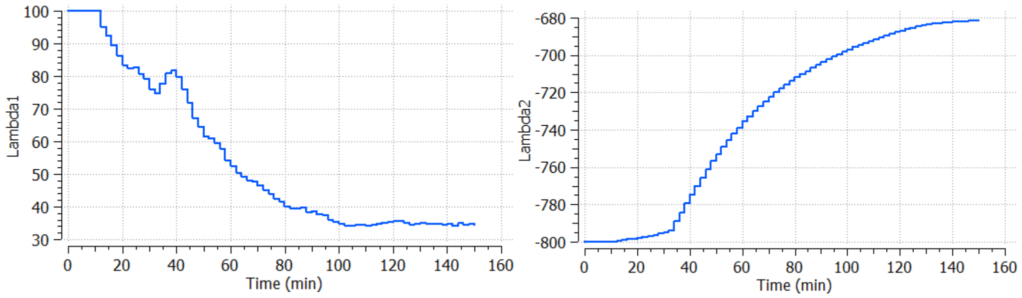


Figure 3.5. Values of lambda modifiers calculated from the solution of the online DME.

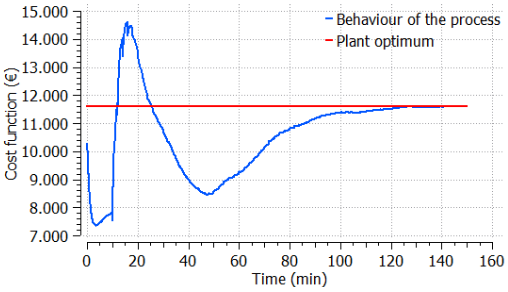


Figure 3.6. Cost function value during simulation eMPC+MA+DME.

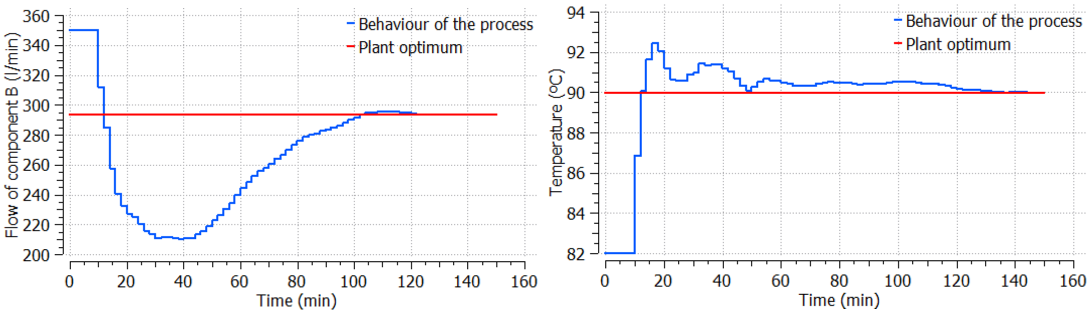


Figure 3.7. Manipulated variables over time eMPC+MA+DME.

As can be seen, after an initial transient, the closed loop system stabilizes at about 140 minutes, which is less than four times the open-loop response of the reactor (40 minutes), reaching an average benefit quite close to the real process optimum.

The next figures (Figure 3.8 and Figure 3.9) compare the results of using eMPC without modifiers, eMPC+MA+DME and eMPC+MA with correct modifiers. The application of eMPC+MA (dark blue line) clearly improves the cost function of the process compared to the application of the eMPC without the correction in the economic function (green line). For comparison, the closed-loop evolution of the eMPC+MA with real optimum modifiers (assuming they were known) is also added (orange line). It can be seen that using the real values of the modifiers in the steady state, the system reaches the optimal operation in 50 minutes, which is almost a third of the convergence time of the eMPC+MA using DME to estimate the modifiers. This reduction in the performance of the proposed algorithm can be explained by the use of transient measurements (rather than steady-state measurements) to estimate the modifiers. Nevertheless, the algorithm was able to reach a stationary point in the neighbourhood of the process optimum in a short time. The small errors obtained can be attributed to the low sensitivity that this variable can have in ϕ_p close to the optimum (flat zone commented from Figure 3.3). The order of magnitude of the ratio between the time to convergence using DME and considering the known gradients reaches a value of about 3 (140 minutes /50 minutes).

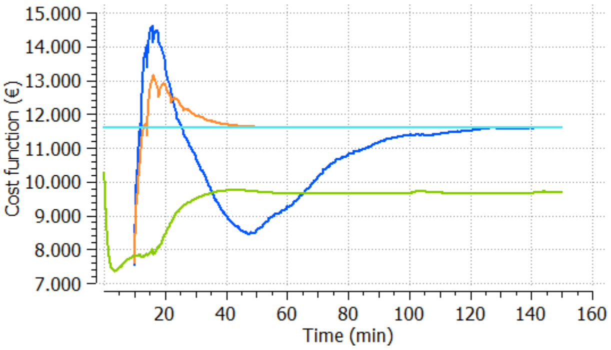


Figure 3.8: Comparison of the cost function with eMPC+MA+DME (dark blue), eMPC+MA with perfect modifiers (orange) and without MA (green). The light blue line corresponds to the plant optimum.

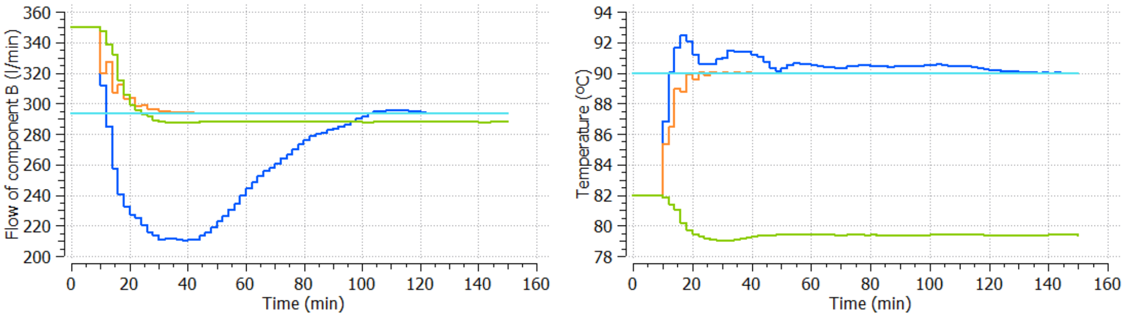


Figure 3.9: Comparison of the control actions with eMPC+MA with DME (dark blue), MA with perfect modifiers (orange) and without MA (green). The light blue line corresponds to the plant optimum.

Finally, noise has been added to the process, which is reflected in 4% oscillations in the cost function, as shown in Figure 3.10. The noise degrades the performance of the DME algorithm by creating a small offset in the decision variables.

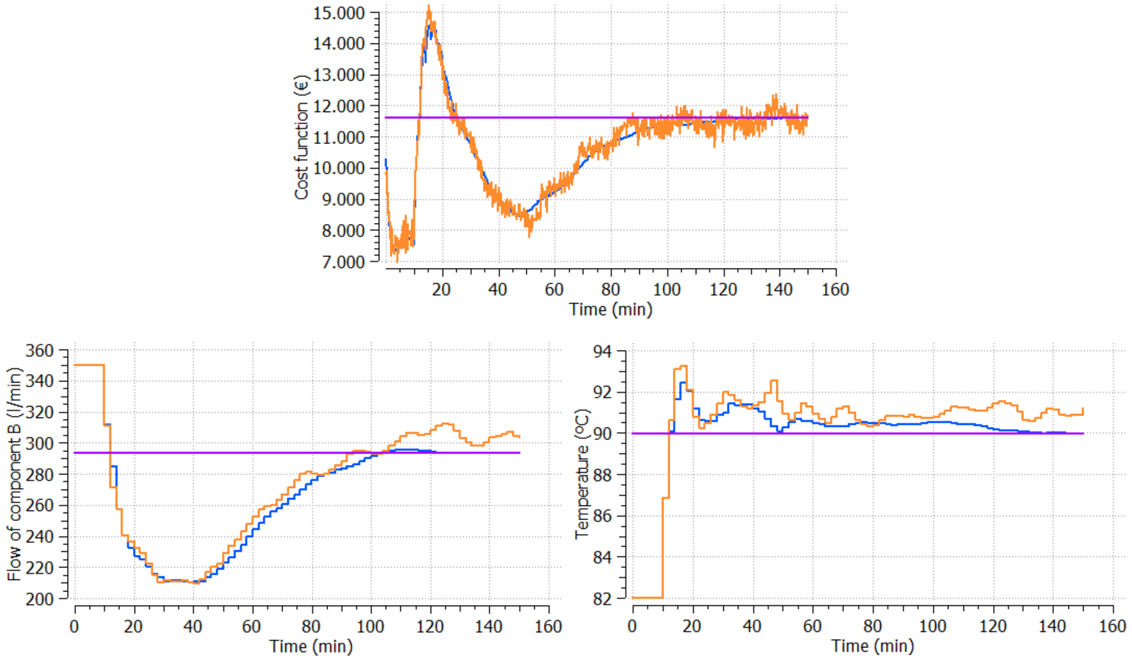


Figure 3.10: Up: Cost function without (dark blue) and with (orange) noise added to the process. Bottom: control efforts without (dark blue) and with (orange) noise added to the process. The light purple line corresponds to the plant optimum.

3.2.7 Conclusions

In this chapter, we presented the DME algorithm that aims to estimate the MA modifiers directly, without the need to explicitly calculate the process and model gradients. The case study presented showed that the eMPC+MA+DME approach can achieve values quite close to the real economic optimum operating point, despite the parametric and structural mismatch between model and process. Therefore, the DME algorithm seems to be a powerful tool to be applied in real processes with slow dynamics.

4 Transient MA (TMA)

This chapter presents an algorithm to estimate the dynamic process gradients during the transient, called Transient MA (TMA). TMA is used in the eMPC+MA architecture presented in Chapter 2 and is applied to the same benchmark example, the Williams-Otto reactor, described in Section 3.2.

4.1 TMA algorithm

For generalization purposes, let us define the variable $\boldsymbol{z}_{p_k}$ as the set of process variables for which we need to estimate the derivatives, i.e., $\boldsymbol{z}_{p_k}^T \coloneqq \left[\boldsymbol{\phi}_{p_k}, \boldsymbol{g}_{p_k}^T\right]$. The dynamical behaviour of a system from a certain instant in time can be approximated as a contribution of the free dynamic evolution of the process and the effect of current and future input moves from the steady state. The approximation is correct for linear systems and may contain some error for nonlinear systems.

4.1.1 On-line predictions of $\mathbf{z}_{p_k}$

We are interested in developing an expression for the value of $\mathbf{z}_p$ at time k using plant information up to time $k-1$. The contribution of what can be called the free dynamic evolution of a variable of a process can be computed from data using the values of the variable $\mathbf{z}_p$ available at time $k-1$ and previous time instants, e.g. using a polynomial approximation that goes through the previous $k-n$ values of $\mathbf{z}_p$ as in Figure 4.1. This polynomial can then be used to extrapolate the values of the variable at time k. For extrapolation, values of the time derivatives of $\mathbf{z}_{p_{k-1}}$ are required, and we can perform first, second or higher-order extrapolations with these derivatives.

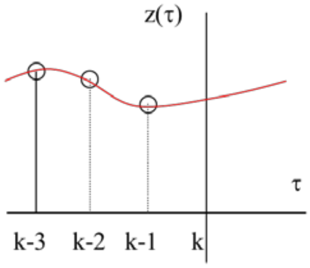


Figure 4.1: Polynomial approximation of a function.

In the context of numerical methods to solve time-dependent systems of differential equations (ODE), Backward Differentiation Formulas give approximations to a derivative of a dependent variable $\boldsymbol{z}(t)$ at a time t_k, in terms of the value of $\boldsymbol{z}_k$ and earlier values (Bank et al., 1993). In addition, Nordsieck (1962) derived a procedure based on a polynomial expression of n^{th} order to estimate the time derivatives for numerical integration methods of ODE systems. Combining these two approaches, one can implement an estimator for the time derivative of process variables using, for instance, a third-degree Nordsieck vector using past data, as presented in equation (4.1).

$$\begin{bmatrix} \boldsymbol{z}_{P_{k-1}} \\ \left.\frac{\widehat{\mathrm{d}\boldsymbol{z}_p}}{dt}\right|_{k-1} \Delta t_{k-1} \\ \left.\frac{\widehat{\mathrm{d}^2\boldsymbol{z}_p}}{dt^2}\right|_{k-1} \frac{\Delta t_{k-1}^2}{2} \\ \left.\frac{\widehat{\mathrm{d}^3\boldsymbol{z}_p}}{dt^3}\right|_{k-1} \frac{\Delta t_{k-1}^3}{6} \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 6 & 0 & 0 & 0 \\ 11 & -18 & 9 & -2 \\ 6 & -15 & 12 & -3 \\ 1 & -3 & 3 & -1 \end{bmatrix}\begin{bmatrix} \boldsymbol{z}_{P_{k-1}} \\ \boldsymbol{z}_{P_{k-2}} \\ \boldsymbol{z}_{P_{k-3}} \\ \boldsymbol{z}_{P_{k-4}} \end{bmatrix} \tag{4.1}$$

For example, a first-order estimation of the free evolution of $\mathbf{z}_{p_k}$ could be computed using (4.1) as in Equation (4.2).

$$\begin{aligned} \mathbf{z}_{p_{k,free}} &\approx \mathbf{z}_{p_{k-1}} + \left.\frac{\widehat{\mathrm{d}\boldsymbol{z}_p}}{\mathrm{dt}}\right|_{k-1} \Delta t \\ &= \mathbf{z}_{p_{k-1}} + \frac{1}{6}\left(11\boldsymbol{z}_{P_{k-1}} - 18\boldsymbol{z}_{P_{k-2}} + 9\boldsymbol{z}_{P_{k-3}} - 2\boldsymbol{z}_{P_{k-4}}\right) \end{aligned} \tag{4.2}$$

Notice that equation (4.1) also calculates higher-order terms with respect to time that can be used in higher-order extrapolations.

Obviously, equation (4.2) reflects the influence of almost all past control actions on the evolution of the system except the effect of the recent control actions $\Delta\boldsymbol{u}_{k-1}$ performed at time $k-1$ that also impact on the value of $\mathbf{z}_{p_k}$. Notice that, if there is a delay d, this comment extends to $\Delta\boldsymbol{u}_{k-d}$. So, in order to predict correctly the value of $\mathbf{z}_{p_k}$, we have to consider the additional effects of the changes of these control variables on $\mathbf{z}_{p_k}$. The corresponding changes on $\mathbf{z}_{p_k}$, $\Delta\mathbf{z}_{p_{k,forced}}$, have to be estimated as if the plant were at steady state at the moment when the changes in $\Delta\boldsymbol{u}_{k-1}$ took place, to make them independent of the evolution due to its past history as reflected in (4.2). Approximations of different orders for this component $\Delta\mathbf{z}_{p_{k,forced}}$ can be obtained by means of the expansion (4.3), where it is clear that the $\Delta\mathbf{z}_{p_{k,forced}}$ depends only on the most recent actions and not on past history:

$$\Delta\mathbf{z}_{p_{k,forced}} \approx \left.\frac{\partial\boldsymbol{z}_p}{\partial\boldsymbol{u}_{k-1}}\right|_{k-1} \Delta\boldsymbol{u}_{k-1} + \frac{1}{2}\Delta\boldsymbol{u}_{k-1}{}^T \left.\frac{\partial^2\boldsymbol{z}_p}{\partial\boldsymbol{u}_{k-1}{}^2}\right|_{k-1} \Delta\boldsymbol{u}_{k-1} + \cdots \tag{4.3}$$

Here $\Delta\boldsymbol{u}_{k-i}$ refers to $\Delta\boldsymbol{u}_{k-i} \coloneqq \boldsymbol{u}_{k-i} - \boldsymbol{u}_{k-i-1}$, and $\left.\frac{\partial z_p}{\partial \boldsymbol{u}_{k-i}}\right|_{k-i}$ stands for the partial derivative of $\boldsymbol{z}_p$ with respect to the decision variables applied at t_{k-i}, estimated at time instant t_{k-i} starting from steady state.

Joining together (4.2) and (4.3), “free response” and “forced response”, we can write an expression for the prediction of $\mathbf{z}_{p_k}$ based on plant data such as the equation (4.4):

$$\begin{aligned}\Delta\mathbf{z}_{p_k} &= \mathbf{z}_{p_k} - \mathbf{z}_{p_{k-1}} \\ &= \left.\frac{\partial\boldsymbol{z}_p}{\partial\boldsymbol{u}_{k-1}}\right|_{k-1} \Delta\boldsymbol{u}_{k-1} + \frac{1}{2}\Delta\boldsymbol{u}_{k-1}{}^T \left.\frac{\partial^2\boldsymbol{z}_p}{\partial\boldsymbol{u}_{k-1}{}^2}\right|_{k-1} \Delta\boldsymbol{u}_{k-1} + \left.\frac{\mathrm{d}\boldsymbol{z}_p}{\mathrm{d}t}\right|_{k-1} \Delta t \\ &\quad + R_n\end{aligned} \tag{4.4}$$

where $\Delta\boldsymbol{z}_{p_k}$ represents the current change in the process variables with respect to the ones measured in the previous sampling time. R_n is a noise term that reflects the different errors that result from the polynomial approximations and for adding (4.2) and (4.3) as the system is normally nonlinear. (4.4) shows a first order approximation of the "free" response but other higher order approximations could have been used.

The interest of (4.4) resides in the fact that we have process data and the (unknown) process gradients related to the ones necessary for MA in the same expression. However, it is important to remark that the derivatives estimated from equation (4.4) are not the steady-state process gradients, but dynamic ones, that change at every time step. The derivatives in (4.4) describe the effect of a change in $\boldsymbol{u}$ on $\boldsymbol{z}_p$ from steady state at a certain time instant during the transient.

The problem now is to estimate the unknown process gradients. The terms of equation (4.4) can be truncated including terms up to a certain order, as in equation (4.5), where we considered, for instance, that the delay is one sampling period, using a second-order approximation with respect to the decision variables and a first-order one w.r.t. time.

$$\Delta \boldsymbol{z}_{p_k} = \left.\frac{\partial \boldsymbol{z}_p}{\partial \boldsymbol{u}_{k-1}}\right|_{k-1} \Delta \boldsymbol{u}_{k-1} + \frac{1}{2}\Delta \boldsymbol{u}_{k-1}^T \left.\frac{\partial^2 \boldsymbol{z}_p}{\partial {\boldsymbol{u}_{k-1}}^2}\right|_{k-1} \Delta \boldsymbol{u}_{k-1} + \left.\widehat{\frac{\mathrm{d}\boldsymbol{z}_p}{\mathrm{d}t}}\right|_{k-1} \Delta t_{k-1} + R_n \qquad (4.5)$$

After that, one can rearrange the terms of equation (4.4), defining a vector $\boldsymbol{\varphi}_{k-1}$ with known variables (measured or estimated), and the vector $\widehat{\boldsymbol{\theta}}_{k-1}$ with the coefficients to be estimated, as equation (4.6) shows.

$$\begin{gathered} \Delta \boldsymbol{z}_{P_k} = \widehat{\boldsymbol{\theta}}_{k-1} \boldsymbol{\varphi}_{k-1}^T + R_n \\ \widehat{\boldsymbol{\theta}}_{k-1} \coloneqq \left[\left.\frac{\partial \boldsymbol{z}_p}{\partial \boldsymbol{u}_{k-1}}\right|_{k-1} \quad \left.\frac{\partial^2 \boldsymbol{z}_p}{\partial {\boldsymbol{u}_{k-1}}^2}\right|_{k-1} \quad \hat{\eta} \right] \\ \boldsymbol{\varphi}_{k-1}^T \coloneqq \left[\Delta \boldsymbol{u}_{k-1}^T \quad \frac{1}{2}\Delta \boldsymbol{u}_{k-1} \Delta \boldsymbol{u}_{k-1}^T \quad \left.\widehat{\frac{\mathrm{d}\boldsymbol{z}_p}{\mathrm{d}t}}\right|_{k-1} \Delta t_{k-1} \right] \end{gathered} \qquad (4.6)$$

Here, vector $\widehat{\boldsymbol{\theta}}_{k-1}$ contains the unknown values to be estimated, where the gradients of the process variables with respect to the decision variables appear in the first positions of the vector. Vector $\boldsymbol{\varphi}_{k-1}^T$ contains values of past moves known at time k or that can be computed with information up to time k as the time derivatives. To consider the estimation error of the time derivatives, an additional coefficient $\hat{\eta}$ has been defined. As equation (4.6) has the form of a typical model used in parameter identification problems, current data of the measured $\Delta \boldsymbol{z}_{P_k}$ and $\boldsymbol{\varphi}_{k-1}^T$ can be used to estimate the process derivatives by solving a recursive identification problem.

During the transient, the estimated gradients are not the required gradients for MA, so that, over this time, the controller will only implement partial corrections. These partial corrections made during the transient will improve the overall performance of the real process in comparison with a standard RTO.

4.1.2 Identification algorithm

As we mentioned before, equation (4.6) has the form of a typical linear model used in parameter identification problems, so that we can use a recursive identification algorithm to estimate the unknown vector $\boldsymbol{\theta}$ containing the process derivatives.

Notice that $\boldsymbol{\theta}$ is a time-varying vector, as it contains the dynamic gradients, so that the recursive identification should take into account this circumstance. Also, notice that only the identified components of $\boldsymbol{\theta}$ corresponding to the first-order gradients $\partial \boldsymbol{z}_p / \partial \boldsymbol{u}$ are required for the MA corrections. As the iterations run, if the identification algorithm performs well, and the process converges to a steady state, the identified gradients will be closer and closer to the ones computed at steady state.

Of course, the success of this scheme is linked to the ability of the identification algorithm to converge to the time-varying parameters, which depends for each particular application on both the identification method chosen and the excitation of the control signals.

In this work, the normalized least mean square algorithm (NLMS) has been used as identification algorithm. NLMS is formulated in the parameter space, using as target to minimize directly the module of the distance, V, from the current estimate of $\boldsymbol{\theta}$ to the real value of the parameters $\boldsymbol{\theta}^*$ as in equation (4.7). The NLMS algorithm has been chosen because it is easy to implement, and it is computationally less expensive than other recursive methods (Isermann and Münchhof, 2011). Furthermore, the gain of the estimator is different from zero, so it can be applied to time-varying problems (Goodwin and Sin, 1984). In equation (4.8), σ is a small positive constant to prevent the numerical difficulties associated with a denominator close to zero, and μ is a gain constant that must be between 0 and 2 in order to decrease the difference between the current estimate of $\boldsymbol{\theta}$ and the real one (Richalet, 1991).

$$V = \|\theta - \theta^*\| \tag{4.7}$$

$$\widehat{\boldsymbol{\theta}}_k = \widehat{\boldsymbol{\theta}}_{k-1} + \mu \frac{\left[\Delta \boldsymbol{z}_{P_{k-1}} - \boldsymbol{\varphi}_k^T \widehat{\boldsymbol{\theta}}_{k-1}\right]}{\sigma + \|\boldsymbol{\varphi}_k^T\|^2}, \qquad 0 < \mu < 2 \tag{4.8}$$

NLMS is globally exponentially convergent to the real $\boldsymbol{\theta}$ if some conditions are fulfilled (Goodwin and Sin, 1984).

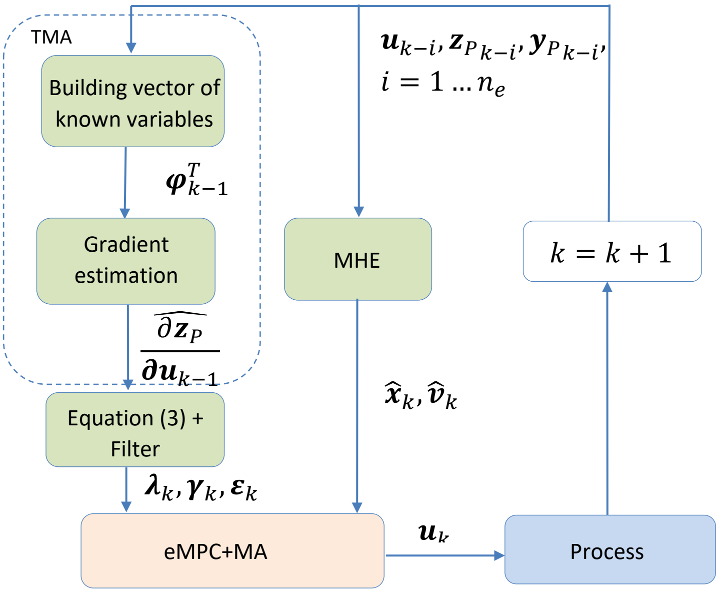

TMA
Building vector of known variables
$\boldsymbol{\varphi}_{k-1}^T$
Gradient estimation
$\dfrac{\widehat{\partial \boldsymbol{z}_P}}{\partial \boldsymbol{u}_{k-1}}$
Equation (3) + Filter
$\boldsymbol{\lambda}_k, \boldsymbol{\gamma}_k, \boldsymbol{\varepsilon}_k$
$\boldsymbol{u}_{k-i}, \boldsymbol{z}_{P_{k-i}}, \boldsymbol{y}_{P_{k-i}},$
$i = 1 \dots n_e$
MHE
$\widehat{\boldsymbol{x}}_k, \widehat{\boldsymbol{v}}_k$
$k = k + 1$
eMPC+MA
$\boldsymbol{u}_k$
Process

Figure 4.2: eMPC+TMA architecture.

After estimating the process gradients, equations (1.8)-(1.10) are applied to finally calculate the modifiers. Notice that the estimated gradients are not exactly the ones required by MA, as they correspond to the change of $\boldsymbol{z}_p$ w.r.t $\boldsymbol{u}$ computed at steady state and not to the change of the steady value of $\boldsymbol{z}_p$ w.r.t. $\boldsymbol{u}$, but they will be used as the best approximation available to the true process gradients. Figure 4.2 describes the two-step identification procedure proposed (TMA), coupled with the eMPC+MA problem. In Figure 4.2 and hereafter, the hat over the variables stands for estimated quantities.

4.2 Possible modification of TMA algorithm: TMA$_m$

TMA is not able to estimate the process gradient at steady state, but at a time prior to steady state. In fact, if we consider a step model, the value estimated by TMA is the first coefficient g_1, see Figure 4.3. If the MVs are kept constant after this step, the process would reach the steady state with a final value of g_n. Since the present splitter case study has a dynamic behaviour with a similar gain around the operating point, so a constant relationship between these coefficients could be calculated.

To improve the TMA algorithm, a correction to the estimated process gradient was added. It consists of changing the estimated process gradient by a factor equal to the difference between the steady-state gradient of the model and the current model gradient. Note that in our case, since the model is linear and invariant, this correction factor is equal to $relation = g_n - g_1$ of each submodel, which can be seen in Figure 4.3. g_1 is estimated by the TMA algorithm presented previously, then the process gradient at steady state can be approximated as $g_n = relation + g_1$. g_n is the process gradient value used to calculate the modifiers.

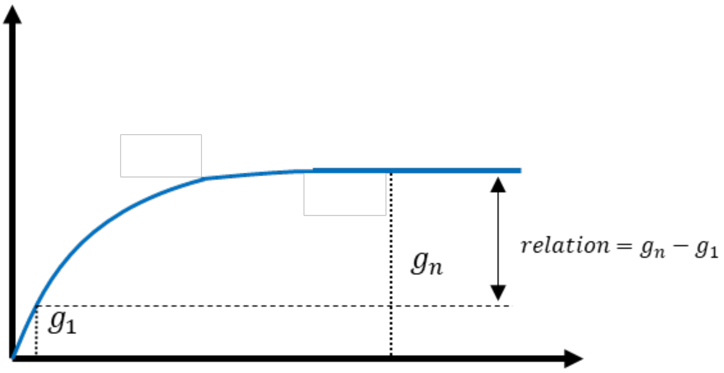


Figure 4.3: Step model of the process.

This strategy will be tested in the case study of Chapter 6. The TMA algorithm with this new approach, will be referred to as TMA$_m$.

4.3 Implementation in the Williams-Otto reactor case study

4.3.1 eMPC+MA+TMA

To evaluate the performance of the TMA algorithm, again, the Williams-Otto case study mentioned in Chapter 3 was again selected. This time, two scenarios were considered. The first scenario, represented by equation (4.9), and it consists of optimizing an economic objective function with no inequality constraints in the dependent variables, which implies that only the modifiers of the objective function are used. The second scenario, defined as equation (4.10), includes constraints on the molar concentration of A and G in the effluent. The upper bounds for X_A and X_G have been chosen such that these constraints are active in the optimum of the process. This implies that the modifiers of these constraints are necessary for the convergence of the algorithm.

$$\max_{\substack{\Delta F_{B_{k+i}}, \Delta T_{k+i} \\ i=0\ldots n_u-1}} \left\{ \bar{F}_R(\bar{X}_P p_P + \bar{X}_E p_E) - F_A X_{A0} p_A - \bar{F}_B X_{B0} p_B \right. \\ + \lambda_{1,k}\left(\bar{F}_B - F_{B,k-1}\right) + \lambda_{2,k}(\bar{T} - T_{k-1}) \\ \left. + \beta_1 \sum_{i=0}^{n_u-1} \left(\Delta F_{B,k+i}\right)^2 + \beta_2 \sum_{i=0}^{n_u-1} (\Delta T_{k+i})^2 \right\}$$

s.t. nonlinear simplified dynamic model plus disturbances from MHE,

$$\forall t \in \left[t_k, t_{pred}\right]$$

$$F_B(t) = F_{B_{k+i}}, \qquad t \in [t_{k+i}, t_{k+i+1}], \qquad i = 0, \ldots n_{pred} - 1 \tag{4.9}$$

$$T(t) = T_{k+i}, \qquad t \in [t_{k+i}, t_{k+i+1}], \qquad i = 0, \ldots n_{pred} - 1$$

$$F_{B_{k+i}} = F_{B_{k+i-1}} + \Delta F_{B_{k+i}}, \qquad i = 0, \ldots, n_{pred} - 1$$

$$T_{k+i} = T_{k+i-1} + \Delta T_{k+i}, \qquad i = 0, \ldots, n_{pred} - 1$$

$$\Delta F_{B_{k+i}} = 0, \qquad i = n_u, \ldots n_{pred} - 1$$

$$\Delta T_{k+i} = 0, \qquad i = n_u, \ldots n_{pred} - 1$$

$$180 \frac{L}{min} \leq F_{B_{k+i}} \leq 360 \frac{L}{min}, \qquad i = 0 \ldots n_{pred} - 1$$

$$75ºC \leq T_{k+i} \leq 100º\text{C}, \qquad i = 0 \ldots n_{pred} - 1$$

$$\max_{\substack{\Delta F_{B_{k+i}}, \Delta T_{k+i} \\ i=0\ldots n_u-1}} \left\{ \bar{F}_R(\bar{X}_P p_P + \bar{X}_E p_E) - F_A X_{A0} p_A - \bar{F}_B X_{B0} p_B \right.$$

$$+ \lambda_{1,k}\left(\bar{F}_B - F_{B,k-1}\right) + \lambda_{2,k}(\bar{T} - T_{k-1})$$

$$\left. + \beta_1 \sum_{i=0}^{n_u-1} \left(\Delta F_{B,k+i}\right)^2 + \beta_2 \sum_{i=0}^{n_u-1} (\Delta T_{k+i})^2 \right\} \qquad (4.10)$$

s.t. nonlinear simplified dynamic model and same constraints from (4.9)

$$X_A\left(t_{pred}\right) + \gamma_{1,k}\left(\bar{F}_B - F_{B,k-1}\right) + \gamma_{2,k}(\bar{T} - T_{k-1}) + \varepsilon_{1,k} \leq 0.8$$

$$X_G\left(t_{pred}\right) + \gamma_{3,k}\left(\bar{F}_B - F_{B,k-1}\right) + \gamma_{4,k}(\bar{T} - T_{k-1}) + \varepsilon_{2,k} \leq 0.35$$

In both scenarios, the MHE module provided the initial states and the disturbances considered at each iteration k of the controller. To evaluate the effect of the proposed eMPC+TMA algorithm, each problem was solved with three strategies: (a) estimating the process gradients using the TMA, (b) estimating the process gradients neglecting the time derivative in equation (4.4), equivalent to the gradient estimation strategy proposed by Rodríguez-Blanco et al. (2017), and (c) solving the eMPC without modifiers (equivalent to setting the value of the modifiers to zero). In addition, another experiment was performed for problem (4.9), including an additive Gaussian noise in the measurements of the molar concentrations of the effluent. This experiment was solved only with strategy (a), as the objective was to evaluate the effect of the measurement noise on the behavior of the TMA.

The simulation of the reactor and the dynamic optimization problem in the MHE and the economic controller were formulated in a continuous-time domain in MATLAB. The controller was implemented using the *nlmpc* object. Both the MHE and the controller were solved using the sequential quadratic algorithm, available in the *fmincon* NLP solver. On average, the whole problem (MHE + controller) took 6 seconds to be solved in each sampling time of the supervisory layer, on a PC running Windows10, Intel Core i5 processor running at 3.2GHz, and 16GB of memory.

For problems (4.9) and (4.10), the monitoring layer was executed every 2 minutes. The control and prediction horizons were set to as $n_u = 3$ and $n_{pred} = 30$, respectively.

The MA filter was set to $K = 0.7$ and the move suppression parameter for both decision variables was $\beta_1 = \beta_2 = 0.5$. The NLMS parameters were set to $\mu = 0.1$ and $\sigma = 1e - 4$. The first 8 minutes were used to collect the data needed for the identification algorithm. The controller started at minute 10.

4.3.1.1 Unconstrained Problem

Figure 4.4 shows the trajectories of the objective function and the decision variables using the three solution strategies tested in problem (4.9). It can be seen that each algorithm reaches a different steady state, depending on the method used to estimate the value of the modifiers.

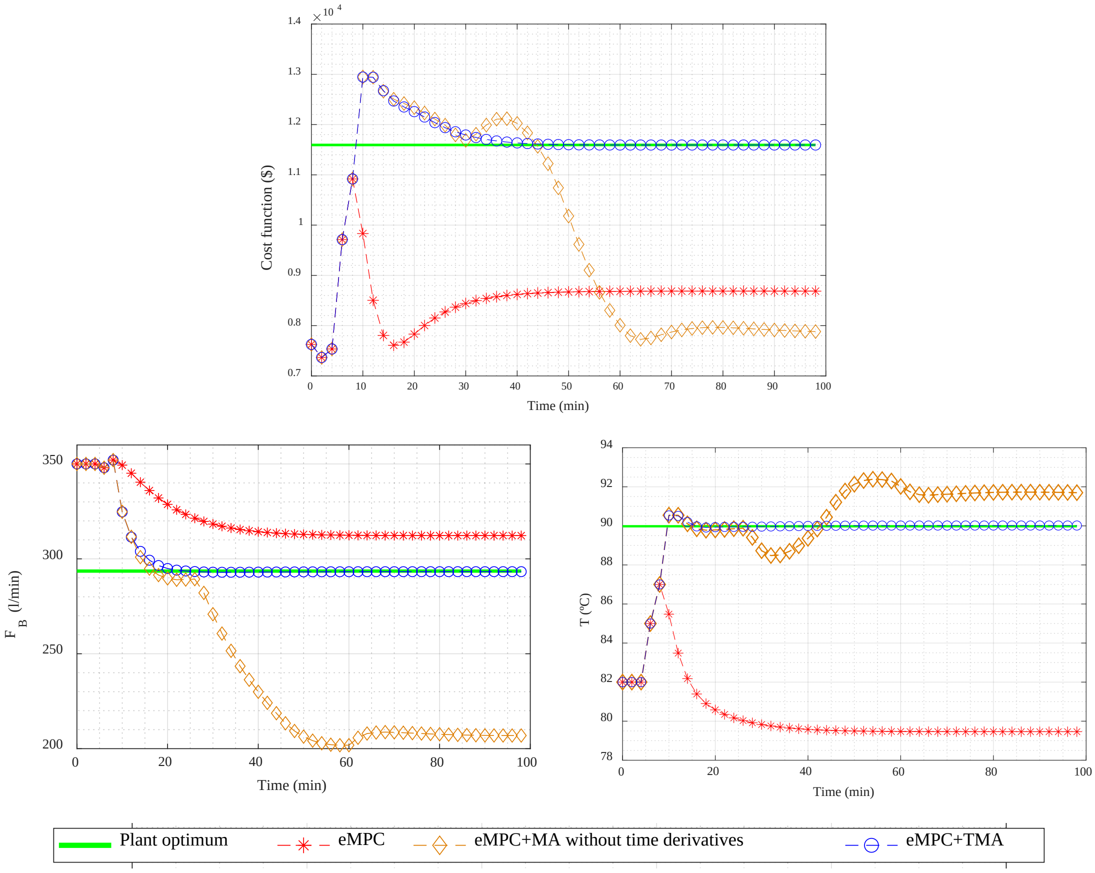


Figure 4.4: Evolution of cost function and manipulated variables for problem (4.9).

For the case of the eMPC, as no correction is implemented in the objective function, the controller drives the process into the optimum of the model ($F_B = 307.2\ L/\text{min}$ and $T = 79.8ºC$), which is identical to the expected results of RTO without the modifiers to correct the structural uncertainty. These decision variables applied to the simplified stationary model without MHE corrections (the same dynamic model where the derivatives are set to zero) would result in a profit of ~11259 €. However, as can be seen in Figure 4.4, these values applied to the real process achieve a much lower profit (~8686€). Meanwhile, for the case of the eMPC+TMA algorithm, the controller detects the real optimum of the process and drives the system to a profit of ~11593 €, a value far greater than the profit obtained without MA. Also, notice that in the eMPC+TMA case the closed-loop system stabilizes around 50 min, which is, approximately, twice the time of the reactor open-loop response time (30 min), and similar to the stabilization time of the eMPC. As the system converges on the expected value, it can be assumed that the TMA allows a correct approximation of the process gradients. This implies that the first-order correction of the objective function agrees in finding the stationary condition of the process. It can be noticed that before $t \approx 25$ min the decision variables proposed by the controller are closer to the optimum of the process, but after this time the supervisory layer suggests a different steady state condition that is kept until the end of the experiment. The failure in detecting the optimum for the eMPC+MA can be explained considering that in the interval from 20 to 25 min the changes in the decision variables become smaller than the previous ones; however, the cost function is affected by the process dynamics. This implies that the effect of time in the objective function emerges as relatively more important than the effect of the decision variables, a condition that produces a miscalculation in the gradient of the process due to not including the time derivative in this case.

Figure 4.5 presents the evolution of the eMPC+TMA with measurement noise in the molar concentrations. Note that the algorithm is capable of finding the optimum of the process, stabilizing the system in about 60 minutes, which is 20% larger than the noise-free example. This difference can be attributed to the oscillations that the temperature presents before minute 25, and because of the additional time that the controller needs to bring the value of F_B closer to the optimum. This is a consequence of the effect that noise has in the MHE and in the gradient estimation step, probably related to the linear model assumed in time dependence. The loss of optimality associated with the extra time needed to converge, due to the additive noise of a standard deviation of 5% with respect to the expected span of the measurement, seems reasonable, especially considering that the process optimum was detected despite the measurement noise.

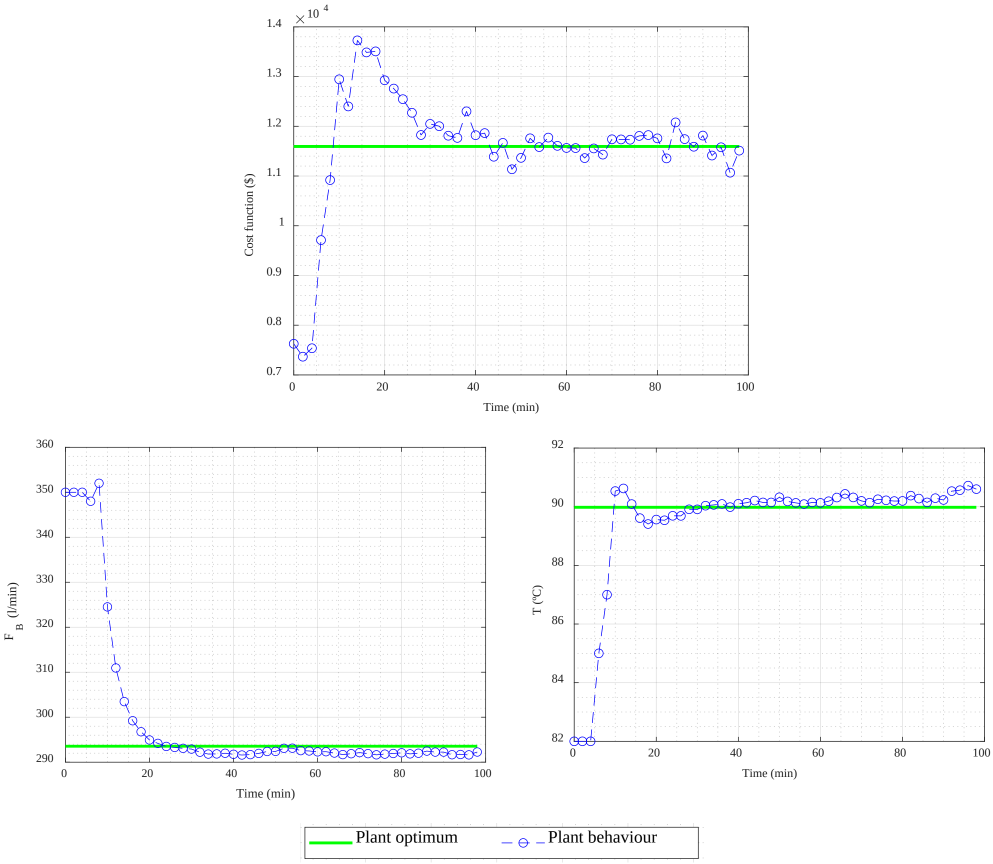


Figure 4.5: Evolution of cost function and manipulated variables for problem (4.9) with measurement noise.

4.3.1.2 Constrained Problem

The evolution of the cost function, molar fractions of A and G, and decision variables for problem (4.10) are presented in Figure 4.6 for the three strategies tested. Similar behavior to the one observed for the unconstrained case can be seen, i.e., the value of the modifiers determines the steady state reached. As in this case the NCO of the process implies that the inequality constraints on X_A and X_G are active, the accuracy of $\boldsymbol{\lambda}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\varepsilon}$ is relevant to detect the optimum of the process. The eMPC converges to a point that is the model optimum at steady state and is not the process optimum, as the eMPC problem does not consider first and zeroth-order corrections in the model. The effect of using these corrections can be seen in the trajectory of the eMPC+TMA. Notice that this controller is capable of detecting the optimum of the process and converges to this value, which implies an increase in the profit of the process of almost 20% as a result of the correct estimation of the modifiers using the TMA strategy. Also, notice that both constraints are satisfied after a while, even though X_A starts at an unfeasible value. The evolution of the process using eMPC+MA shows that the algorithm cannot converge to the process optimum, although the modifiers have been included. As discussed for the unconstrained case, this difference can be attributed to the effect of the miscalculation of the modifiers because the value of the time derivatives is neglected in this process gradient estimation procedure. This miscalculation becomes more critical when the system tends to stabilize, as can be observed in the interval from $t = 20$ to $t = 40$ min. In this interval, the decision variables remain practically constant while the cost function and the constrained variables are still changing in time because of the inertia of the process. These findings support the need to include the dynamics in the process gradient estimation step to calculate the modifiers used in the eMPC+MA formulation. An additional consequence of not including the dynamics in the calculation of the modifiers is the increment in the settling time of the controller because of the changes in the value of the proposed steady state.

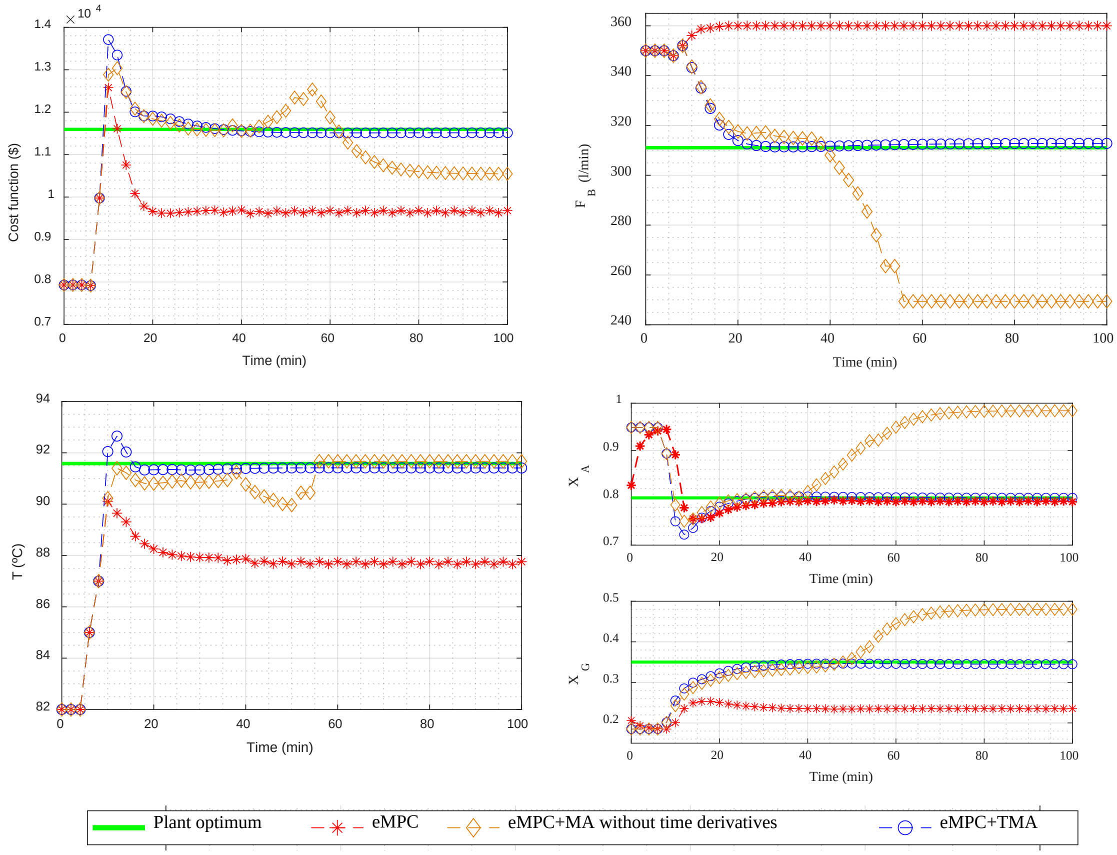


Figure 4.6: Evolution of cost function and manipulated variables for problem (4.10).

4.3.2 Conclusions

In this chapter, we presented the second algorithm developed to estimate the dynamic process derivatives, which is based on the use of a two-step identification algorithm named TMA. The TMA uses the available measurements from the process to estimate the time derivative using the Nordsieck's vector, followed by the estimation of the process dynamic derivatives with respect to the decision variables using a recursive normalized least squares algorithm. Although the algorithm does not estimate the steady-state gradients of the cost function and constraints, but those at each time step, the results obtained in the Williams-Otto simulation were better than the standard eMPC. In fact, considering good initial values for the estimation, the result could also achieve the true optimum of the plant.

It can also be concluded that for a better estimation of the dynamic process gradients from the transient measurements using the proposed approximated model of the dependent variables, it is necessary to include not only the decision variables but also the dependence on time. Besides, it is important to ensure that the process has the necessary excitation so that the identification method can correctly estimate the gradients.

5 Hybrid Laboratory Plant Case Study

Chapter 5 presents the results of a second case study for evaluating the performance of the TMA methodology. Now in real time, operating a plant of the Process Control Laboratory of the ISA Department, University of Valladolid.

5.1 Description of the process

The second case study for evaluating the performance of the TMA methodology is a system implemented in a hybrid laboratory plant (Figure 5.1) that emulates the behaviour of a CSTR with the Van de Vusse reactions (Van de Vusse, 1964). Hybrid plants are real processes with some simulated components. They have been used in metallurgical processes as an alternative to validate the performance of supervisory algorithms in more realistic situations than those given by computational experiments (Bergh and Yianatos, 2014; Navia et al., 2019, 2016). The concept of hybrid plants is based on the fact that some process phenomena can be replaced by computations of measured variables and their effect on the process can be physically implemented, at least partially, by means of proper actuators. In our case, fluid dynamics and thermal effects can be emulated using an experimental setup and a fluid with characteristics similar to the original, while chemical mechanisms can be emulated using a first-principles model that uses the experimental measurements to close the degrees of freedom. This configuration allows the definition of two kinds of decision variables: experimental and virtual. Experimental decision variables are associated with the variables that can be manipulated in the actual experimental setup, such as flows, pressures, or liquid levels. Virtual decision variables can be any variable that affects the output of the process, such as the simulated properties of the raw materials. The use of hybrid plants permits the study of the effect of experimental errors on the behaviour of supervisory control algorithms, knowing part of the modelling mismatch, and facilitates the maintenance of pilot plants in a laboratory.

The schematic of the experimental setup is presented in Figure 5.1 (Montes et al., 2021). It consists of an exothermic CSTR with four temperature sensors (TT-01 to 04) and two flowmeters (FT-01 and 02). The final control elements (experimental decision variables) are the peristaltic reagent pump (P-101) and the coolant valve (V-101). The P-102 pump allows emulating the reactants leaving the reactor by overflow. The power of two heating coils located at the bottom of the vessel can be manipulated by a power amplifier in J-101. The system uses water as the working fluid and the chemical reactions are simulated using a chemical model running in real time with process data. The heat released by the chemical reactions is computed by the simulation and its value is applied to the process using the J-101 power amplifier as if a true exothermic reaction were taking place in the reactor.

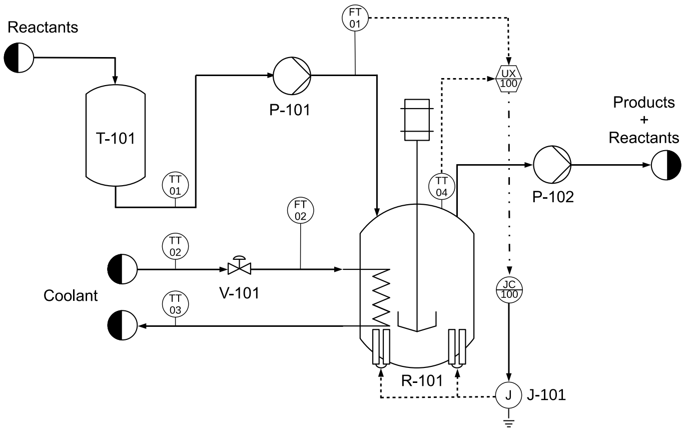


Figure 5.1: Laboratory plant P&ID.

The chemical model, integrated into the process as component UX-100, was obtained from the description of the system. The process consists of the three Van de Vusse reactions, where component B is the desired product:

$$A \overset{k_1}{\rightarrow} B \overset{k_2}{\rightarrow} C$$

$$2A \overset{k_3}{\rightarrow} D$$

Assuming that the physical properties of water are constant within the range of the operating conditions and that the vessel is perfectly agitated, a first-principles model of the reactions can be derived based on mass balances for each component. Equations (5.1) to (5.4) represent the dynamic mass balances for each compound, whereas equations (5.5) to (5.7) describe the respective reaction rates, assuming elemental kinetics and Arrhenius dependence with temperature. As mentioned earlier, equation (5.8) computes the heat that is applied to the reactor by the J-101 amplifier in real-time.

$$V\frac{dC_A}{dt} = q(C_{A0} - C_A) + V(-r_1 - 2r_3) \tag{5.1}$$

$$V\frac{dC_B}{dt} = -qC_B + V(r_1 - r_2) \tag{5.2}$$

$$V\frac{dC_C}{dt} = -qC_C + Vr_2 \tag{5.3}$$

$$V\frac{dC_D}{dt} = -qC_D + Vr_3 \tag{5.4}$$

$$r_1 = k_{10}e^{-\frac{E_1}{RT}}C_A \tag{5.5}$$

$$r_2 = k_{20}e^{-\frac{E_2}{RT}}C_B \tag{5.6}$$

$$r_3 = k_{30}e^{-\frac{E_3}{RT}}C_A^2 \tag{5.7}$$

$$P = \frac{1000}{60}V\sum_{j=1}^{3} -r_j\Delta H_j \tag{5.8}$$

where C_i is the molar concentration of component i in the effluent, q is the total volumetric flow, C_{A0} is the molar concentration of component A in the influent, and V is the reaction volume. For the kinetic expressions, r_j, k_{j0}, and E_j are the reaction rate, exponential constant, and activation energy of reaction j, respectively, while T is the temperature in the reactor. Equation (5.8) defines the conversion between the heat generated by the reactions (ΔH_j is the heat of the reaction j) and the heat power supplied by the electrical resistances, denoted by P.

To clarify the relationship between the experimental and the simulated variables, Table 5.1 summarizes how the value of each variable is obtained. Note that as E_j, k_{j0} and ΔH_j are parameters that are assumed to be known so that the model of the process can be simulated.

Table 5.1: Description of the source of the variables for the hybrid process.

Variable	Source
$C_i, i \in \{A, B, C, D\}$	Calculated from the chemical model
$r_j, j = \{1,2,3\}$	Calculated from the chemical model
P	Calculated from the chemical model
q	Measured from FT-01
T	Measured from TT-04
V	Defined as the volume of the vessel
C_{A0}	Defined by the user (fixed value)

The value of the heat of reaction calculated from the chemical model is used as the set-point of J-101, which makes it possible to emulate the effect of the exothermic reactions in the experimental setup.

In the control layer (eMPC+TMA and MHE), a simplified nonlinear dynamic model has been implemented that neglects the existence of reaction 3 and component D, which defines a structural uncertainty in the available model. The experimental system has two decision variables: the flow rate of the reactants (manipulated with P-101) and the flow rate of the coolant (manipulated with V-101). The model is based on mass and energy balances and a description of how heat is transferred from the vessel to the coolant. Equations (5.9) to (5.16) summarize the model.

$$V\frac{dC_A}{dt} = q(C_{A0} - C_A) + V(-r_1) \tag{5.9}$$

$$V\frac{dC_B}{dt} = -qC_B + V(r_1 - r_2) \tag{5.10}$$

$$V\frac{dC_C}{dt} = -qC_C + Vr_2 \tag{5.11}$$

$$r_1 = k_{10}e^{-\frac{E_1}{RT}}C_A \tag{5.12}$$

$$r_2 = k_{20}e^{-\frac{E_2}{RT}}C_B \tag{5.13}$$

$$\rho C_p V\frac{dT_{out}}{dt} = q\rho C_p(T_{in} - T_{out}) - Q - V\sum_{j=1}^{2} r_j \Delta H_j \tag{5.14}$$

$$\rho C_p V_c\frac{dT_{c,out}}{dt} = F_R\rho C_p\left(T_{c,in} - T_{c,out}\right) + Q \tag{5.15}$$

$$Q = \alpha_1 F_R^{\alpha_2}\,(T_{out} - T_{c,out}) \tag{5.16}$$

where T and T_C are the temperatures of the reactant and coolant streams, respectively. The sub-indexes in and out indicate whether the temperature is of the inlet or outlet stream. C_p and ρ are the heat capacity and density of the coolant, respectively. Q is the heat transfer, F_R is the coolant flow rate, and V_C is the volume of the cooling coil. Note that equation (5.16) approximates Newton's law of cooling, where the total heat transfer coefficient has been defined as a function of the coolant flow rate with parameters α_1 and α_2.

The values of the parameters used in this case study are given in Table 5.2. In Table 5.2 and hereafter, "Chemical model" refers to the model used to emulate the physicochemical mechanisms in the hybrid plant (equations (5.1) to (5.8)), while "eMPC model" refers to the model implemented in the supervisory layer (equations (5.9) to (5.16)).

Table 5.2: Value of the parameters used in the case study.

Parameter	Chemical Model	eMPC Model	Parameter	Chemical Model	eMPC Model
k_{10} (min^{-1})	9.95×10^{10}	6.40×10^{9}	ΔH_3 $(\mathrm{kJ} \cdot \mathrm{mol}^{-1})$	-66.54	-
k_{20} (min^{-1})	9.99×10^{11}	4.84×10^{10}	V (L)	11.5	11.5
k_{30} $(\mathrm{L} \cdot \mathrm{mol}^{-1} \cdot \mathrm{min}^{-1})$	9.99×10^{12}	-	V_c (L)	-	1.0
E_1 $(\mathrm{kJ} \cdot \mathrm{mol}^{-1})$	59.0	52.1	ρ $(\mathrm{kg} \cdot \mathrm{L}^{-1})$	-	1.0
E_2 $(\mathrm{kJ} \cdot \mathrm{mol}^{-1})$	77.62	70.0	C_p $(\mathrm{kJ} \cdot \mathrm{kg}^{-1} \cdot {}^{\circ}\mathrm{C}^{-1})$	-	4.18
E_3 $(\mathrm{kJ} \cdot \mathrm{mol}^{-1})$	71.11	-	C_{A0} $(\mathrm{mol} \cdot \mathrm{L}^{-1})$	5.0	5.0
ΔH_1 $(\mathrm{kJ} \cdot \mathrm{mol}^{-1})$	-21.22	-48.75	α_1	-	1.8
ΔH_2 $(\mathrm{kJ} \cdot \mathrm{mol}^{-1})$	-2.68	-34.50	α_2	-	0.8

Note that the uncertainty of the model-based optimization can be considered structural because both reaction 3 and component D have been ignored and the heat transfer model is a simplification. In addition, there is parametric uncertainty as shown in Table 5.2.

The aim of the supervisory layer is to maximize the economic benefit, which can be defined as the income related to selling the desired product B minus the costs of the reactants and coolant, manipulating q and F_R. Analogous to Section 4.3, two dynamic optimization problems have been considered for this case study. The first problem consists in the optimization of the benefit with no inequality constraints in the dependent variables, implying that only the modifiers of the cost function are needed as presented in equation (5.17). The second problem includes an additional inequality constraint on the reactor temperature. For this constrained problem, we have used the output MA (MAy) formulation to solve the optimization problem. This MAy formulation includes the modifiers in the output variables as presented in equation (5.18). According to Papasavvas et al. (2019), this formulation improves the convergence of the MA type algorithms. In both problems, their formulations follow the structure of eMPC+MA presented in Chapter 2, while the value of the prices and costs are given in Table 5.3.

$$\max_{\substack{\Delta q_{k+i}, \Delta F_{R_{k+i}} \\ i=0\ldots n_u-1}} \begin{Bmatrix} \bar{q}(C_B p_B - C_{A0} p_A) - \bar{F}_R p_R + \lambda_{1,k}(\bar{q} - q_{k-1}) \\ + \lambda_{2,k}\left(\bar{F}_R - F_{R_{k-1}}\right) \\ + \beta_1 \sum_{i=0}^{n_u-1} (\Delta q_{k+i})^2 + \beta_2 \sum_{i=0}^{n_u-1} \left(\Delta F_{R_{k+i}}\right)^2 \end{Bmatrix}$$

s.t. Nonlinear model (5.9) to (5.16) plus disturbances from MHE , $\forall t \in [t_k, t_{pred}]$

$$\begin{gathered} q(t) = q_{k+i}, \qquad t \in [t_{k+i}, t_{k+i+1}], \qquad i = 0, \ldots n_{pred} - 1 \\ F_R(t) = F_{R_{k+i}}, \qquad t \in [t_{k+i}, t_{k+i+1}], \qquad i = 0, \ldots n_{pred} - 1 \\ q_{k+i} = \ q_{k+i-1} + \Delta q_{k+i}, \qquad i = 0, \ldots, n_{pred} - 1 \\ F_{R_{k+i}} = \ F_{R_{k+i-1}} + \Delta F_{R_{k+i}}, \qquad i = 0, \ldots, n_{pred} - 1 \\ \Delta q_{k+i} = 0, \qquad i = n_u, \ldots n_{pred} - 1 \\ \Delta F_{R_{k+i}} = 0, \qquad i = n_u, \ldots n_{pred} - 1 \\ 0.3 \frac{\mathrm{L}}{\mathrm{min}} \leq q_{k+i} \leq 1.2 \frac{\mathrm{L}}{\mathrm{min}}, \qquad i = 0 \ldots n_{pred} - 1 \\ 6 \frac{\mathrm{L}}{\mathrm{min}} \leq F_{R_{k+i}} \leq 15 \frac{\mathrm{L}}{\mathrm{min}}, \qquad i = 0 \ldots n_{pred} - 1 \end{gathered} \tag{5.17}$$

$$\max_{\substack{\Delta q_{k+i},\Delta F_{R_{k+i}} \\ i=0\ldots n_u-1}} \left\{ \bar{q}\left(C_{B,MA}p_B - C_{A0}p_A\right) - \bar{F}_R p_R + \beta_1 \sum_{i=0}^{n_u-1} (\Delta q_{k+i})^2 + \beta_2 \sum_{i=0}^{n_u-1} \left(\Delta F_{R_{k+i}}\right)^2 \right\} \tag{5.18}$$

s.t. Same constraints from problem (5.17)

$$C_{B,MA} = \overline{C_B} + \lambda_{1,k}(\bar{q} - q_{k-1}) + \lambda_{2,k}\left(\bar{F}_R - F_{R_{k-1}}\right) + \varepsilon_{1,k}$$

$$T_{R,MA} = \overline{T_R} + \gamma_{1,k}(\bar{q} - q_{k-1}) + \gamma_{2,k}\left(\bar{F}_R - F_{R_{k-1}}\right) + \varepsilon_{2,k}$$

$$T_{R,MA} \leq 38^{\circ}\mathrm{C}$$

Table 5.3: Costs and prices for the case study.

Parameter	Description	Value
p_B	Price of component B	$18.0\ € \cdot \mathrm{mol}^{-1}$
p_A	Cost of component A	$0.2\ € \cdot \mathrm{mol}^{-1}$
p_R	Cost of coolant	$3.0\ € \cdot \mathrm{L}^{-1}$

5.2 Results

To evaluate the effect of the proposed algorithm, each problem was solved with two strategies: (a) estimating the process gradients using the TMA and (b) solving the eMPC without modifiers. For both problems, the simulation of the equations of the hybrid process (the chemical model) was developed with EcosimPro (EA Int., 2020) and encapsulated as an OPC-DA server generated in the same environment.

5.2.1 Unconstrained Problem

For problem (5.17), the control and the prediction horizons were defined as $n_u = 3$ and $n_{pred} = 60$ respectively, and the move suppression parameter was fixed to $\beta_1 = \beta_2 = 0.5$. The parameters for the NLMS algorithm were set to $\mu = 1.6$ and $\sigma = 1e - 4$. The dynamic optimization problem was solved using the simultaneous approach, discretizing the dynamic model using the orthogonal collocation method. The resulting algebraic problem was implemented in Pyomo (Hart et al., 2017, 2011), using IPOPT as an NLP solver. The supervisory layer started at time t = 3 min and was executed every 30s, with an average computation time of 0.5s in a PC under Windows10, Intel Core i5-4460 processor at 3.2GHz and 8GB of RAM.

Figure 5.2 shows the evolution of the objective function and the decision variables of two experiments carried out under similar conditions with the eMPC+TMA (left) and the eMPC without correction (right). The two lines that appear in each figure of q and F_R correspond to: the dashed line is the setpoint (decision variable) of a PID that governs this variable (i.e., is the decision variable generated by the controller) and the continuous line is the current value of the variable. Both experiments started from the same initial steady state point: $q = 0.7 L/min$ and $F_R = 5.7\ L/min.$ It can be noticed that both algorithms converge to different steady states. Table 5.4 summarizes the 99% confidence intervals for the mean of the profit and the decision variables at the end of the experiments for both algorithms.

Table 5.4: 99% confidence intervals of the mean of the convergence point for both algorithms in problem (5.17).

Variable	**eMPC+TMA**	**eMPC**
Profit (€/min)	$[32.1, 33.5]$	$[26.7\,, 28.4]$
q (L/min)	$[1.18, 1.20]$	$[1.19\,, 1.20]$
F_R (L/min)	$[5.8, 6.2]$	$[10.8\,, 11.3]$

In terms of profit, the inclusion of the modifiers represents a significant improvement in the economic results of the process, averaging of 19%. Table 5.4 shows that the confidence interval of the decision variables obtained with the eMPC is very close to the optimum of the model, which is consistent with the results obtained for the case study in section 4.3. The convergence point of the eMPC+TMA supports the hypothesis that the inclusion of the modifiers allows improving the performance of the process by correcting the optimum computed with a model with structural modeling mismatch. However, it is not possible to evaluate whether the steady state obtained by the eMPC+TMA is the process optimum, since a “perfect model” is never available when dealing with real processes. Nevertheless, some observations can be made by analyzing the results, the prices, and the predicted consequences of the modeling mismatch. From the economic point of view, as the price of B is three orders of magnitude larger than the price of A and two orders of magnitude larger than the price of the coolant, it is expected that the process optimum result increases the production of component B as much as possible. Therefore, the flow of q is set to its upper bound, which is achieved for both algorithms (no statistical differences are observed for q in Table 5.4). Since q is fixed in the optimum, the remaining degrees of freedom (F_R) must be selected to maximize the concentration of B. Assuming unlimited cooling capacity, this is equivalent to finding the value of T_R that maximizes C_B using equations (5.1) to (5.7) for the process and equations (5.9) to (5.13) for the model, both in steady state. Unlike the original problem, the cooling assumption removes the experimental uncertainty, and a comparison can be made by solving this surrogate steady-state optimization problem with fixed $q = 1.2 \text{ L min}^{-1}$, and using the parameters given in Table 5.2. The solution shows that the process optimal temperature is 8ºC higher than the model optimum. This difference in temperature between process and model implies that the correct process optimum should reduce the use of coolant, as is observed in Figure 5.2, where F_R is close to its lower bound at the end of the experiment.

From the point of view of the process dynamic behaviour, since the system reaches a steady state in both cases, the inclusion of the modifiers does not affect the stability of the controller for this case study. However, from the evolution of the objective function, it can be noticed that the closed-loop system stabilizes in around 60 minutes for the eMPC+TMA, which is twice the stabilization time of the eMPC. This could be explained by comparing the transient behaviour of F_R for both algorithms. Notice that the eMPC detects a target value that remains constant during the whole experiment, unlike the eMPC+TMA that changes three times the target value of F_R: (1) for $t < 8$ min the system behaves in a similar way than the eMPC, looking for the optimum of the model; (2) from $t = 10$ to $t = 32$ min the controller proposes a decrease in F_R which improves the objective function; and (3) from $t = 35$ min ahead, where the expected optimum of the process is identified. The observed changes in the target value of F_R are similar to the behavior of the decision variables from section 4.3 when the estimation of the process gradients neglects the effect of time (denoted as eMPC+MA in Figure 4.4 and Figure 4.6). Probably due to the presence of measurement noise in the experimental variables, the assumption of linear dependence on time in the TMA is not enough to detect the correct value during the transient. However, as the identification algorithm implemented in the

TMA (NLMS) is recursive, the increase of its accuracy as more data is available produces successive changes in the decision variable, improving the objective function as a consequence of including the time dependency, which makes the algorithm converge to the expected value. With respect to the value of q, the model and the process coincide, and the eMPC+TMA continuously drives the process to the right value, regardless of the assumed estimation error of the experimental gradient.

An alternative to avoid the undesired behaviour observed in F_R and to reduce the convergence time of the algorithm can be adding additional time dependencies in the TMA using higher-order Taylor expansions. This may be an interesting future research direction.

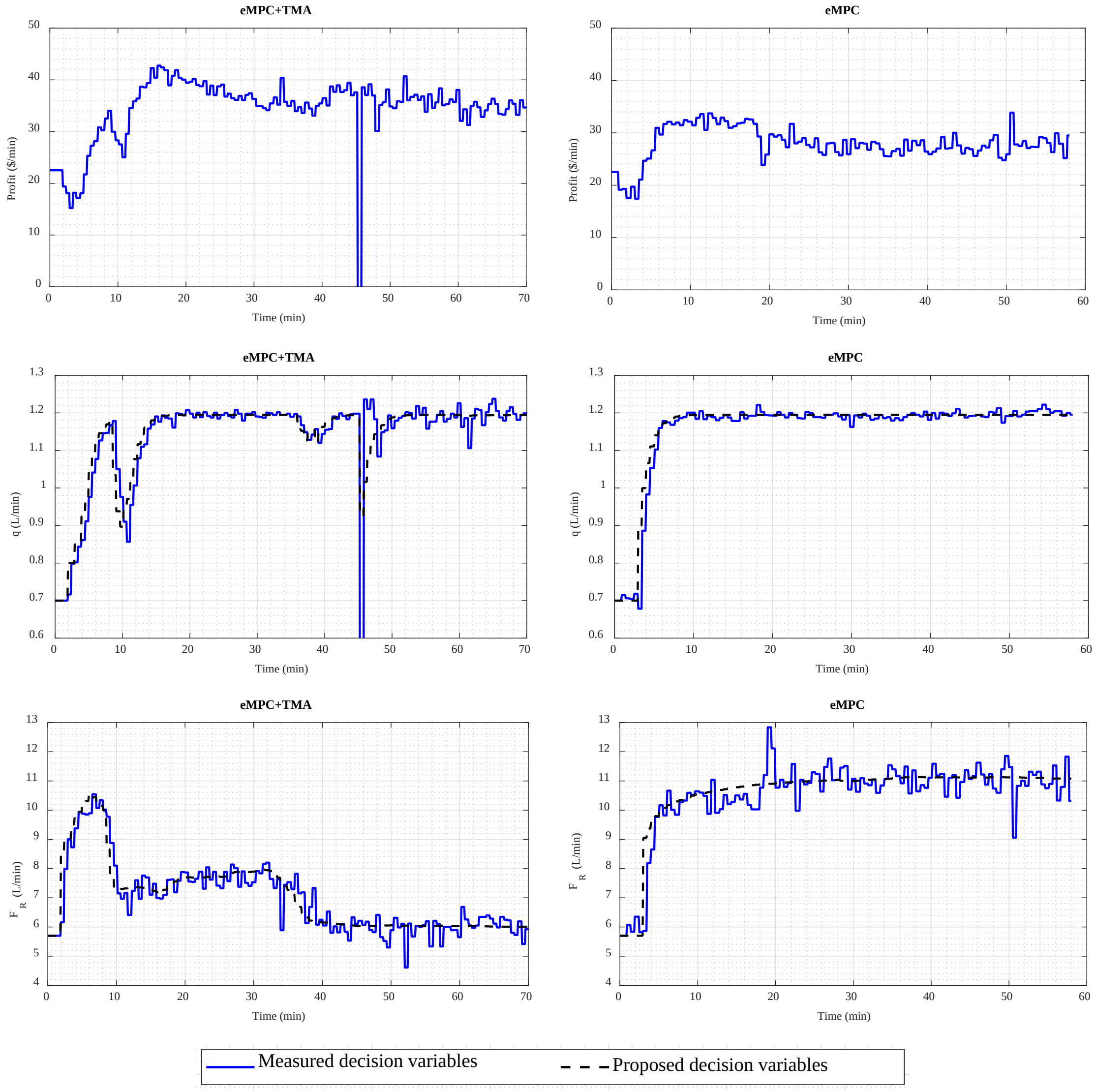


Figure 5.2: Cost function and manipulated variables over time for problem (5.17).

5.2.2 Constrained Problem

For this example, an additional constraint in the upper bound of the reactor temperature $\boldsymbol{T_R \leq 38ºC}$ has been introduced to modify the expected optima of the unconstrained case, where its final measured value was $\boldsymbol{T_R = 39.5ºC}$. The MAy (output MA) approach has been implemented in the modified problem (5.18), so the first and zeroth-order modifiers have been calculated for $\boldsymbol{C_B}$ and $\boldsymbol{T_R}$, which are the dependent variables that explicitly appear in the objective function and the inequality constraint. The first-order modifiers for $\boldsymbol{C_B}$ ($\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_2}$) and for $\boldsymbol{T_R}$ ($\boldsymbol{\gamma_1}$ and $\boldsymbol{\gamma_2}$) were calculated using their respective process gradients estimated by the TMA approach. This implies that in this example $\boldsymbol{z_P} \coloneqq [\boldsymbol{C_B}, \boldsymbol{T_R}]$ for the TMA.

The control and the prediction horizons were defined as $n_u = 3$ and $n_{pred} = 60$, respectively. The suppression parameters were set to $\beta_1 = 0.004$ and $\beta_2 = 0.002$. The parameters for the NLMS algorithm were set to $\mu = 0.05$ and $\sigma = 1e - 4$. The rest of the specifications are the same as for the unconstrained case. Both experiments started from the same initial point: $q = 0.7L/min$ and $F_R = 5.7\ L/min$.

The evolution of the objective function, the reactor temperature (constrained variable) and the decision variables are shown in Figure 5.3 for the eMPC+TMA (left) and the eMPC (right).

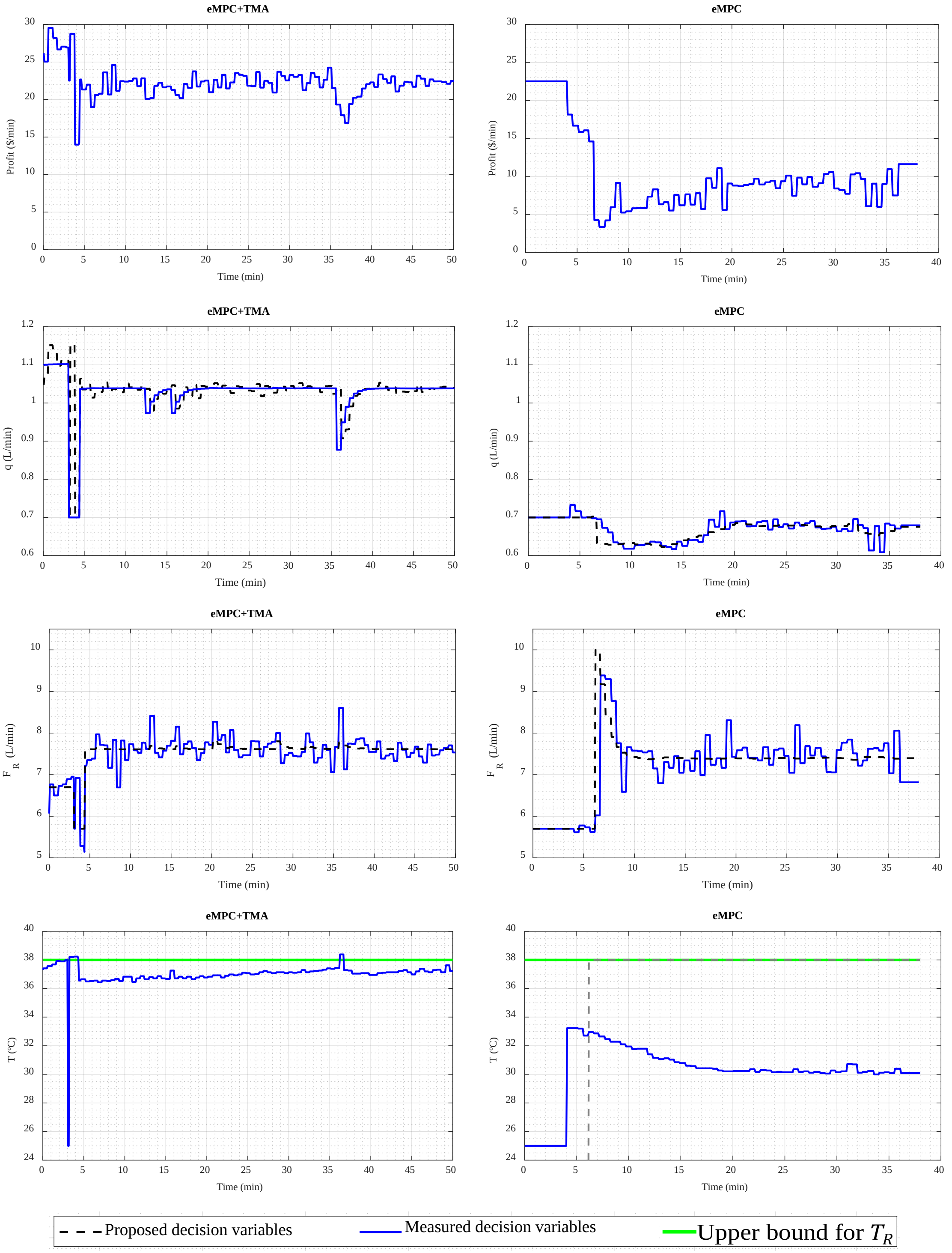


Figure 5.3: Cost function, constrained variable, and manipulated variables over time for problem (5.18).

Figure 5.3 shows a similar behavior as the previous examples, with both strategies converging to different steady states, indicating that the value of the modifiers affects the economic objective of the controller. The 99% confidence intervals for the objective function, reactor temperature, and decision variables obtained with both strategies are summarized in Table 5.5.

Table 5.5: 99% confidence intervals of the mean of the convergence point for both algorithms for problem (5.18).

Variable	**eMPC+TMA**	**eMPC**
Profit (€/min)	$[21.0, 22.0]$	$[8.6\,, 9.9]$
T (ºC)	$[37.2, 37.4]$	$[30.1, 30.2]$
q (L/min)	$[1.02, 1.04]$	$[0.67\,, 0.68]$
F_R (L/min)	$[7.5, 7.8]$	$[7.3\,, 7.6]$

Table 5.5 shows that the profits obtained with the two methods are significantly different with a 99% confidence. This difference implies that the use of the eMPC+TMA in this process reports an average 130% increase in the economic benefit of the process compared to the eMPC results. Since the upper bound of the temperature is smaller than its final value for the unconstrained case, it is expected that the algorithm proposes a final condition such that this constraint is active, increasing the coolant flow rate with respect to previous results while keeping the value of q in its upper bound. The confidence intervals of F_R from Table 5.4 and Table 5.5 show that the eMPC+TMA proposes a significant increase of 27% with respect to the unconstrained example. However, Table 5.5 and Figure 5.3 show that the algorithm is not able to detect the active constraint of T_R, even though its value is close to the upper bound. The observed "experimental optimality gap" could be caused by a constant error in the estimation of the first-order modifiers for T_R as a consequence of the experimental noise, which can also explain the decrease of the final value of q by 13% with respect to its upper bound.

Concerning the eMPC results, notice that the evolution of T_R, and the confidence interval at steady state show that the modeling mismatch causes an overestimation of T_R (this can also be seen from the eMPC results for the unconstrained case). This overestimation is reflected in 50% reduction of the proposed value of q with respect to the unconstrained solution of the eMPC. Finally, as commented before, the estimated value of the modifiers γ_1, γ_2 in the eMPC+TMA case might have persistent errors, causing the constant gap in the NCO of the modified problem. Thus, it can be concluded that the overestimation of T_R can be reduced by TMA, but it is still preserved.

5.3 Conclusions

In this chapter, the eMPC+TMA has been tested in a laboratory case study with simulation and experimental components, in both cases solving optimization problems with and without uncertain inequality constraints.

The results obtained in the hybrid plant show that the experimental errors affect the capacity of the eMPC+TMA to estimate the steady-state optimum of an uncertain process during the transient state. This can be seen in the use of more than one transition to estimate the target value or in the presence of an optimality gap that cannot be reduced by first order corrections. This allows us to conclude that it is necessary to evaluate the use of other approximations to describe the time dependence in a more robust way. However, it is important to remark that the inclusion of the modifiers calculated with the TMA has significantly increased the profit of the process for both experimental examples, reducing the optimality gap related to the process-modeling mismatch.

6 Industrial Case: Propylene-Propane Splitter

This chapter presents the results of the application of eMPC+MA in an industrial case of a propane-propylene splitter. The study has been developed as a hardware-in-the-loop system, combining a real-time optimization and control module with a rigorous dynamic simulation of the plant, as a first step to validate the developed system prior to its implementation in the process. The rigorous model has been validated using historical data.

First, a rigorous model was developed in the EcosimPro simulation software. The simulation acted as the real plant in a virtual plant environment created to perform the eMPC+MA experiments prior to its implementation in the industry. The simulation in EcosimPro was then encapsulated in an OPC-UA deck and connected to a commercial Aspen DMC controller. The same DMC configuration and model of the real process was maintained. Finally, the eMPC+MA architecture presented in Chapter 2 was added to the control structure of the splitter, using the DME or TMA algorithm presented in Chapters 3 and 4 to estimate the MA modifiers with transient data (Figure 6.1).

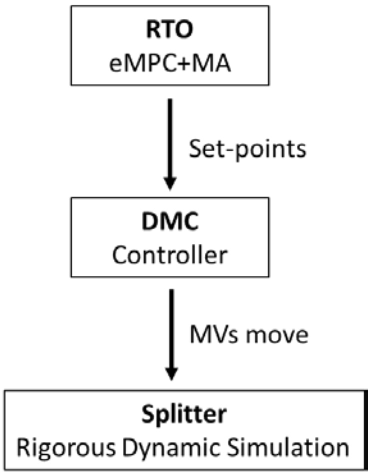


Figure 6.1. Propylene-propane splitter automation pyramid proposal.

Two economic optimization problems were defined for the splitter process. The purpose was to operate the splitter within specifications and at the best possible cost. The first problem considers the highest value product to be the column head product, propylene. In the second problem, the highest value product is considered to be the bottom product, propane. The economic optimization problem of the splitter will consider the cost of steam and the profit obtained from the sale of the products while meeting the product quality constraints (minimum concentration of propylene in the distillate).

6.1 Process description

The case study is a simulation of a real propane-propylene splitter located downstream of a Fluid Catalytic Cracking (FCC) unit at the Petronor refinery in Muskiz, Vizcaya, in northern Spain. The splitter under study is designed to produce high purity propylene from a stream of propylene, propane and a small amount of impurities (C2-C4 hydrocarbons). The splitter consists of a total condenser, a partial boiler, and 135 equilibrium stages. The splitter is operated by a DMC controller to maintain the propylene concentration in the distillate product within a range (≥ 97.5% molar) by controlling the distillate, steam flow, and top pressure (Figure 6.2). The process also has controllers to maintain the level in the accumulator vessel and base at a set point by manipulating the reflux and bottoms flow rates, respectively.

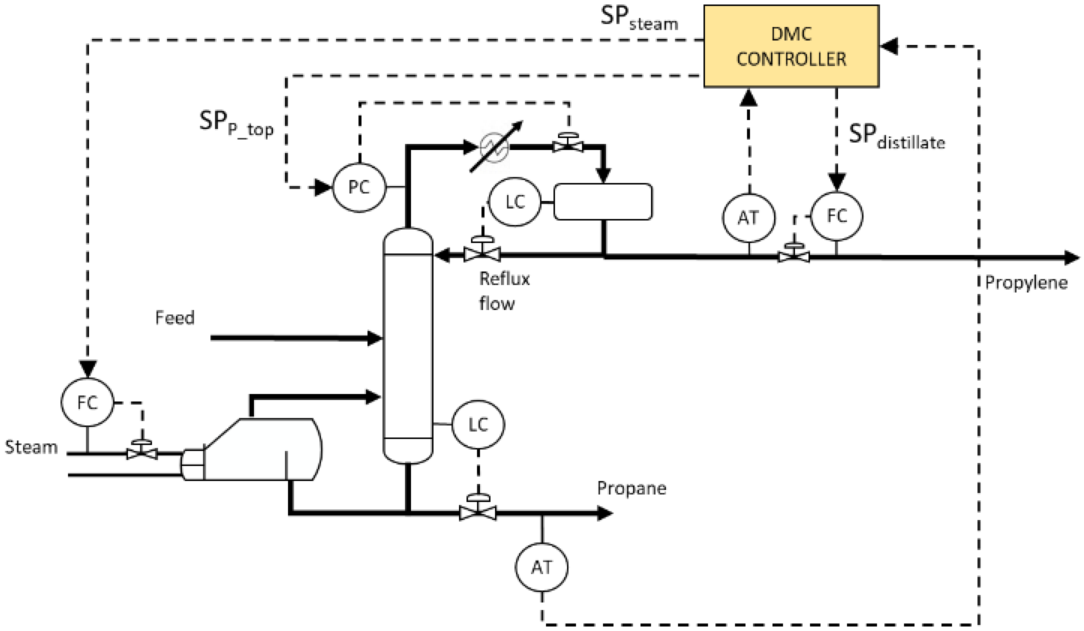


Figure 6.2. Current control structure of the propylene-propane splitter in the Petronor refinery.

In this process, both propylene and propane are end products. The distillate below 97.5% of propylene is considered out of specification. The out-of-spec product can also be sold, but at a much lower price. On the other hand, the price of the bottom product (propane) does not depend on the propylene concentration. However, a higher concentration of propylene in propane reduces the amount of the more valuable product produced.

The next subsection presents the model developed for the process.

6.1.1 Model Development

Dynamic models aim to simulate the behavior of a process over time and they are used at different stages of a plant's life cycle. For example, in the design stage, a dynamic model is useful for operability and controllability studies. During operation, dynamic simulations are used to train operators, validate safety procedures, and study different operating conditions for optimization and control. In our case, we are trying to mimic the operation of the process in response to the controller actions and disturbances.

There are several ways to describe a distillation column using a dynamic model. In general, these models use conservation laws such as mass, energy, and momentum, and time dependent constitutive equations that define the relationship between intensive variables and extensive variables (as equations of state and equilibrium equations). These equations form a system of differential-algebraic equations (DAE).

A propane-propylene splitter is a super fractionator, i.e., a distillation column that performs the separation of components with quite low relative volatility (<1.2) between the components, so the number of equilibrium stages required is very high.

The mathematical model takes into account the following simplifying hypothesis:

- Constant pressure drop.
- The feed consists of four components: propylene, propane, isobutane and ethane.
- The condenser allows subcooling.
- The column is insulated.

The described nonlinear dynamic model was developed using EcosimPro (EA Int., 2020). The model has 12090 equations and it was solved using the IDAS_SPARSE integration solver (Hindmarsh et al., 2005).

All equations used in the model are presented below.

6.1.1.1 Internal Trays

Equations (6.1) and (6.2) describe the total and individual component material balances for each stage. In equation (6.1), the variable mol_n is the molar liquid holdup at stage n. n is the number of stages in the column, varying between $n = 1$, the base, and $n = 136$, the top. l_{n+1} and l_n represent the molar liquid flow (kmol/h) coming from the upper stage that goes to the downer stage, respectively, and, v_{n-1} and v_n are the vapor flows coming from the stage below the one that goes to the upper stage.

$$\frac{dmol_n}{dt} = l_{n+1} + v_{n-1} - l_n - v_n \qquad (6.1)$$

In equation (6.2), the term $x_{j,n}$ corresponds to the liquid molar fraction of the component j in the tray n and $y_{j,n}$ corresponds to the vapor molar fraction.

$$\frac{dmol_n x_{j,n}}{dt} = l_{n+1}x_{j,n+1} + v_{n-1}y_{j,n-1} - l_n x_{j,n} - v_n y_{j,n} \qquad (6.2)$$

The dynamics of the specific liquid internal energy depends mainly on the change in composition, which is assumed to be faster than the dynamics of the total mass. Therefore, its effect on the energy balance can be negligible, avoiding a high index problem associated with the calculation of the temperature (equation (6.3)). In equation (6.3), h is the enthalpy of the liquid flow and H is the enthalpy of the vapor flow. Also note that equation (6.3) is used to calculate the vapor flow in each tray, not the temperature. Temperatures are calculated using the thermodynamic equilibrium as explained next in the Thermodynamics section.

$$h_n \frac{dmol_n}{dt} = h_{n+1}l_{n+1} + H_{n-1}v_{n-1} - h_n\, l_n - H_n v_n \qquad (6.3)$$

6.1.1.2 Thermodynamics

An equation of state (EoS) is a thermodynamic equation that describes the properties of pure substances and mixtures. The general form can be written as a function of the physical conditions, pressure P, volume V and temperature T, equation (6.4).

$$f(P,V,T) = 0 \tag{6.4}$$

The high pressures (>10 bar) inside the splitter required the use of a cubic equation of state (EoS), such as the Peng Robinson equation of state (Matsoukas, 2013). Peng Robinson is the most widely used EoS to model hydrocarbon and petroleum mixtures (Fahim, M. A.; Al-Sahhaf, T. A.; Elkilani, 2010) and it is represented by equation (6.5).

$$P_n = \frac{RT_n}{V_n - b_n} - \frac{a_n}{{V_n}^2 + 2b_nV_n - {b_n}^2} \tag{6.5}$$

The parameters a and b can be calculated for a pure substance as the parameters a_j, b_j, da_j/dT, using the critical temperature and pressure constants (T_c and P_c) and acentric factor (ω) for each pure component j, equations (6.6).

$$\begin{gathered}
a_{j,n} = 0.45724\frac{R^2{T_{c_j}}^2}{P_{c_j}}\left[1 + \Omega_j\left(1 - {T_{r_{j,n}}}^{1/2}\right)\right]^2 \\
\frac{da_{j,n}}{dT} = -0.45724\frac{R^2{T_{c_j}}^2}{P_{c_j}}\frac{\left(1 + \Omega_i\left(1 - \sqrt{T_{r_{j,n}}}\right)\right)\Omega_j}{\sqrt{T_{r_{j,n}}}} \\
T_{r_{j,n}} = \frac{T_n}{T_{c_j}} \\
\Omega_j = 0.37464 + 1.54226\omega_j - 0.26992{\omega_j}^2 \\
b_j = 0.07780\frac{RT_{c_j}}{P_{c_j}}
\end{gathered} \tag{6.6}$$

In the case of hydrocarbon mixtures, it is necessary to calculate the mixing parameters using the mixing rules a, $b, da/dT$, equations (6.7).

$$a_n = \sum_i \sum_j x_{i,n} x_{j,n} \left(1 - k_{ij}\right) \sqrt{a_{i,n} a_{j,n}}$$
$$b_n = \sum_j x_{j,n} b_j \qquad (6.7)$$
$$\frac{da_n}{dT} = \frac{1}{2} \sum_i \sum_j x_{i,n} x_{j,n} \left(1 - k_{ij}\right) \sqrt{a_{i,n} a_{j,n}} \left[\frac{1}{a_{i,n}} \frac{da_{i,n}}{dT} + \frac{1}{a_{j,n}} \frac{da_{j,n}}{dT}\right]$$

Using equations (6.8) and a, b, the dimensionless parameters A_n and B_n of the cubic polynomial equation are calculated.

$$A_n = \frac{a_n P_n}{(RT_n)^2}$$
$$B_n = \frac{b_n P_n}{RT_n} \qquad (6.8)$$

The cubic polynomial, equation (6.9), relates A_n and B_n parameters to the compressibility factor (Z_n). If there is more than one positive root in the solution, the value chosen depends on the phase of the mixture: for a liquid mixture, the smaller value is chosen; for a gas mixture, the larger value is chosen.

$$Z_n{}^3 + (B_n - 1) Z_n{}^2 + Z_n\left(A_n - 3B_n{}^2 - 2B_n\right) - A_n B_n + B_n{}^3 + B_n{}^2 = 0 \qquad (6.9)$$

The compressibility factor is related to the fugacity of liquid $\phi^{liq}{}_i$ and vapor $\phi^{vap}{}_j$, equations (6.10).

$$ln\phi_{i,n} = \frac{b_i}{b_n}(Z_n - 1) - ln(Z_n - B_n) - \frac{C_{i,n}}{2\sqrt{2}} \ln\left[\frac{\left(1 + \sqrt{2}\right)B_n + Z_n}{\left(1 - \sqrt{2}\right)B_n + Z_n}\right]$$
$$C_{i,n} = \frac{A_n}{B_n}\left(-\frac{b_i}{b_n} + \frac{2}{a_n} \sum_{j=1}^{N} x_{j,n} \sqrt{a_i a_j}\left(1 - k_{ij}\right)\right) \qquad (6.10)$$

The equilibrium composition in the vapor y_{eq} for each component is related to the K-value and the fugacity coefficients by equations (6.11) and (6.12).

$$K_{j,n} = \frac{y_{eq\,j,n}}{x_{j,n}} \qquad (6.11)$$

$$K_{j,n} = \frac{\phi^{liq}{}_{j,n}}{\phi^{vap}{}_{j,n}} \tag{6.12}$$

The Murphree tray efficiency, ε_n, is used to calculate the actual performance of the splitter, equation (6.13).

$$y_n = \varepsilon_n\left(y_{eq,n} - y_{n-1}\right) + y_{n-1} \tag{6.13}$$

Finally, the enthalpy is calculated using equation (6.14).

$$\begin{gathered} H^R{}_n = RT_n(Z_n - 1) + \frac{T_n(da/dT)_n - a_n}{2\sqrt{2}b_n}\ln\left[\frac{\left(1+\sqrt{2}\right)B_n + Z_n}{\left(1-\sqrt{2}\right)B_n + Z_n}\right] \\ H_n = \sum_i x_{i,n}H_{i,n}^{ig} + H^R{}_n \end{gathered} \tag{6.14}$$

The thermodynamic equations above are used in each tray n of the column.

6.1.1.3 Column hydraulic

A simple Francis weir equation is used to relate the liquid holdup in the tray to the liquid flow leaving the tray (Luyben, 1999), equation (6.15). The flow depends on the fluid mechanics of the tray. l_vol_i is the liquid flow rate over the weir, h_{ow} is the liquid height over the weir, and L_w is the length of the weir.

$$l\_vol_n = 3.33L_w{h_{ow}}^{1.5} \tag{6.15}$$

6.1.1.4 Feed

In the feed tray, the total, individual material balances are represented by equations (6.16) and (6.17), where f is the feed molar flow and z_j is the feed molar fraction for each component j. The subscript nf indicates the feed stage.

$$\frac{dmol_{nf}}{dt} = f + l_{nf+1} + v_{nf-1} - l_{nf} - v_{nf} \tag{6.16}$$

$$\frac{dmol_{nf}x_{j,nf}}{dt} = fz_j + l_{nf+1}x_{j,nf+1} + v_{nf-1}y_{j,nf-1} - l_{nf}x_{j,nf} - v_{nf}y_{j,nf} \tag{6.17}$$

The energy balance is represented by equation (6.18), similar to equation (6.3) .

$$h_{nf}\frac{dmol_{nf}}{dt} = fh_f + h_{nf+1}l_{nf+1} + H_{nf-1}v_{nf-1} - h_{nf}l_{nf} - v_{nf}H_{nf} \tag{6.18}$$

6.1.1.5 Top tray

In the top tray, heat and mass balances similar to those for the inner trays could be considered (equations (6.19) through (6.21)), where nt refers to the top tray ($n = 136$) and r is the molar reflux flow rate.

$$\frac{dmol_{nt}}{dt} = r + v_{nt-1} - l_{nt} - v_{nt} \tag{6.19}$$

$$\frac{dmol_{nt}x_{j,nt}}{dt} = rx_{j,r} + v_{nt-1}y_{j,nt-1} - l_{nt}x_{j,nt} - v_{nf}y_{j,nf} \tag{6.20}$$

$$h_{nt}\frac{dmol_{nt}}{dt} = h_r r + H_{nt-1}v_{nt-1} - h_{nt}l_{nt} - H_{nt}v_{nt} \tag{6.21}$$

6.1.1.6 Base and reboiler

The column base and the reboiler are assumed to be tray number 1. Equation (6.22) is the total mass balance at the base, where mol_{base} is the molar liquid holdup and l_{base} is the liquid molar flow. Equation (6.23) is the total mass balance in the reboiler, where b is the bottom molar flow and v_1 is the vapor flow leaving the reboiler. The equation (6.24) calculates the molar liquid holdup in the reboiler and the level $L_{reboiler}$. Equation (6.25) corresponds to the material balances of the individual components inside the reboiler. Equation (6.26) is the energy balance in the reboiler. The heat dynamics is faster than the composition dynamics, so equation (6.26) is used to calculate the vapor flow in the base. Equation (6.27) calculates the heat generated $Q_{reboiler}$ by the steam flow

F_S with a heat of vaporization ΔH_{Vap} in the pressure P_S, considering a steam quality τ_S. A τ_S of 0 indicates 100% liquid (condensate) and a τ_S of 1 indicates 100% steam.

$$\frac{dmol_{base}}{dt} = l_2 - l_{base} \tag{6.22}$$

$$b = l_{base} - v_1, \; l_{base} = K_{base}\sqrt{\frac{mol_{base} * MM_{base}}{A_{base}\rho_{base}}} \tag{6.23}$$

$$mol_{reboiler} = \frac{A_{reboiler}L_{reboiler}\rho_1}{MM_1} \tag{6.24}$$

$$mol_{reboiler}\frac{dx_{j,1}}{dt} = l_{base}x_{j,2} - v_1 y_{j,1} - b x_{j,1} \tag{6.25}$$

$$v_1 = \frac{h_2 l_2 - h_1 b + Q_{reboiler}}{H_1} \tag{6.26}$$

$$Q_{reboiler} = \; \tau_S F_S \cdot \Delta H_{Vap}(P_S) \tag{6.27}$$

6.1.1.7 Condenser

The splitter has a flooded condenser to control the top pressure by manipulating the flooded area of the condenser with a control valve located below. An increase in the condensate flow lowers the liquid level in the condenser and increases the area available for condensation (Luyben, 2017).

Equations (6.28)-(6.35) describe the condenser with subcooling liquid. Equation (6.28) is the vapor-molar material balance inside the condenser. mol_{v_cond} is the vapor holdup inside the condenser, v_{nt} is the vapor leaving the top tray, and l_{cond} is the amount of vapor condensing to liquid. Equation (6.29) relates the molar liquid holdup in the condenser (mol_{l_cond}) to the flow of liquid entering the liquid phase from condensation, l_{cond}, and the flow of liquid going to the accumulator vessel, l_{accum}.

$$\frac{dmol_{v\_cond}}{dt} = v_{nt} - l_{cond} \tag{6.28}$$

$$\frac{dmol_{l\_cond}}{dt} = l_{cond} - l_{accum} \tag{6.29}$$

Equations (6.30) and (6.31), respectively, represent the total latent heat lost by the vapor to condensate into a saturated liquid and the total sensible heat lost to cool the saturated liquid to a temperature below the bubble point. Both latent and sensible heat transfer processes are very fast, so the dynamics are not considered.

$$Q_{cond\_sat} = l_{cond}\left(H_k - h_{cond\_sat}\right) \tag{6.30}$$

$$Q_{cond\_sub} = l_{cond}h_{cond\_sat} - l_{accum}h_{cond\_sub} \tag{6.31}$$

Equations (6.32) and (6.33) describe the temperature change for the condensate flow and refrigerant flow, given the equipment design data and the condensate level in the condenser. The parameter α describes the percentage of the area available for condensation, equation (6.34).

$$Q_{cond\_sat} = U_v(1 - \alpha)\mathrm{A}_{cond}\left(T_{nt} - {T_{in}}^{W}\right) \tag{6.32}$$

$$Q_{cond\_sub} = U_l\alpha\mathrm{A}_{cond}\left(T_{accum} - {T_{in}}^{W}\right) \tag{6.33}$$

$$\alpha = \frac{V_{l\_cond}}{V_{T\_cond}} \tag{6.34}$$

Finally, equation (6.35) represents the total heat removed from the process flow by the refrigerant. F_W is the coolant flow, ${C_P}^{W}$ is the coolant capacity heat, and ${T_{out}}^{W}$, ${T_{in}}^{W}$ are the coolant outlet and inlet temperatures, respectively.

$$Q_{cond\_sat} + Q_{cond\_sub} = F_W{C_P}^{W}\left({T_{out}}^{W} - {T_{in}}^{W}\right) \tag{6.35}$$

The composition inside the condenser is assumed to be the same as the vapor to be condensed.

6.1.1.8 Accumulator vessel

The accumulator is a horizontal vessel that collects the liquid from the condenser. Part of the liquid is removed as distillate and the rest is returned to the splitter as reflux liquid. The total and partial mass balance can be written as (6.36) and (6.37). The accumulator vessel has a large volume, so an energy balance is also necessary (6.38).

$$\frac{dmol_{accum}}{dt} = l_{accum} - r - d \tag{6.36}$$

$$\frac{dmol_{accum}x_{j,d}}{dt} = l_{accum}y_{j,nt} - rx_{j,r} - dx_{j,d} \tag{6.37}$$

$$\frac{d(mol_{accum}h_d)}{dt} = l_{accum}h_{cond\_sub} - rh_d - dh_d \tag{6.38}$$

Note that, actually, $x_{j,r} = x_{j,d}$ is the reflux and distillate concentration leaving the accumulator vessel.

6.1.1.9 Pressure profile

The rigorous model assumes that the gas accumulates in the condenser as shown in equation (6.28). The pressure drop is assumed to be constant along the column and the maximum pressure drop between the bottom and the top is approximately 1 bar. Equation (6.39) calculates the pressure profile in the column except for the top tray, which is given by equation (6.40) (Luyben, 1999). The term mol_{v_cond} is the vapor phase holdup in the condenser and Z_{nt} is the compressibility factor calculated by Peng Robinson presented in section 6.1.1.2.

$$P_n = P_{n+1} + \Delta p \tag{6.39}$$

$$P_{nt} = Z_{nt}\frac{mol_{v\_cond}RT_{nt}}{V_{nt}} \tag{6.40}$$

6.1.1.10 PI controllers

The base/reboiler level, the accumulator level and the pressure PI controllers are modeled in the rigorous dynamic model. The first manipulates the bottom flow to maintain the level in the base, the second manipulates the reflux flow to maintain the level set point in the accumulator, and the last manipulates the liquid flow from the condenser to the accumulator to maintain the top pressure at the set point.

6.1.1.11 Parameters of the model

Calibration of a dynamic simulation requires some information about the equipment involved in the process. The size of the condenser and its vapor phase heat transfer coefficient, the size of the reboiler, and the diameter, weir length, and height of the column were obtained from the respective equipment specification sheet. Thermodynamic data were obtained from the Simulis Thermodynamics software (“Simulis Thermodynamics,” 2021) and from Wauquier (1995). Other model parameters, such as the column efficiency, were obtained from operating data that matched the model and process responses for the same inputs using data reconciliation algorithms described later.

6.1.1.12 Boundary conditions

The boundary conditions for the full model are shown in Table 6.1.

Table 6.1. Boundary conditions.

	Variable
1	Feed temperature
2	Feed flow
3	Feed pressure
4	Feed composition
5	Distillate flow
6	Steam flow to reboiler
7	Condenser refrigerant flow
8	Condenser refrigerant inlet temperature
9	Accumulator vessel level set point
10	Reboiler level set point
11	Top pressure set point

6.1.1.13 Initialization and convergence of the model

The presented splitter rigorous model contains 12090 equations that require the initial values of 553 state variables and 549 algebraic variables (1102 variables in total). It is very difficult to find a set of initial values for all these variables that will allow the IDAS DAE integration algorithm to converge on the first try. For this reason, the methodology proposed below has been applied.

The methodology constructs simplified models based on the full rigorous model, using assumptions that may not be appropriate for the process under study, but reduce the number of initial values required. In each step, the simplified models approximate the full rigorous model until a subset of good initial values is found.

The methodology applies to DAE systems with a large number of equations and requires multiple initial values for state and algebraic variables. The methodology consists of the following steps:

1. Create the simplest possible model that can calculate the controlled variables with the minimum number of initial values required. To create this first simplest model, several assumptions were considered, even though they could not be applied to the problem studied.
2. Use the design and/or operational data to predict the initial values for state and algebraic variables.
3. Simulate the model and let it reach the steady state and save the new values for the state or algebraic variables.
4. Discard one of the assumptions that does not apply to the problem and add the equations that describe the phenomena. A new set of initial values is now required, consisting of the previous set of initial values and a new set associated with the new equations added. The steady state results obtained from the previous step are used and the new initial values are again predicted from the design or operational data.
5. Simulate the model and let it reach the steady state. Save the new initial values and repeat the step 4 until all incorrect assumptions have been discarded.

Many of the initial values of the splitter model are difficult to predict, such as the composition of the components and the temperature in each tray. For this reason, five different models were constructed, from the simplest to the most complex necessary to predict well the behavior of the real plant. The simplest model considered was based on model 1 presented by Grassi in Luyben (2006).

The complete rigorous model includes equations that describe each of the phenomena below:

a) Total material balances for each tray – liquid holdup dynamics

b) Francis equation - liquid flow dynamics

c) Component Material Balances and Equilibrium Relationships (Peng Robinson) – liquid and vapor composition dynamics

d) PI level controllers – reflux flow dynamics and bottom flow dynamics

e) Energy balances – temperature dynamics

f) Condenser mass balance and head equation of state – head pressure dynamics

g) PI pressure controller

h) Sensible heat equations – condenser subcooling

Model 1 (System of equations of a + b + c)

The first model considered the total and component mass balance and the Francis equation to calculate the liquid flow from each tray. The temperature gradient between top and bottom is known from the design data and a linear profile was used to calculate the liquid densities. The vapor flow inside the column is constant and equal to the liquid evaporated in the reboiler. In this step, the bottom and reflux flows were set to a value between zero and the value from the total material mass, equation (6.1), at steady state (derivative equal to zero).

The system of equations required six boundary conditions: the feed flow, the distillate flow, the vapor flow evaporated in the reboiler, and the feed composition. The values used for the boundary conditions were the splitter design data.

The initial conditions required are the molar holdup, composition and equilibrium constant at each stage. In this step, the feed was assumed to contain 20% propane and 80% of propylene, so the composition of isobutane and ethane was zero. A linear profile for propane and propylene was used, taking into account the expected concentrations at the bottom and top from the design data. Equilibrium constant values were taken from the Scheibel and Jenny nomograph for light hydrocarbons (Hemptinne et al., 2012). After the previous simulation reached the steady state, the boundary conditions for the concentration of isobutane and ethane were increased to the real values. Each composition was changed one by one, always waiting for the simulation to reach steady state. A qualitative assessment was made considering that the lighter components will have the higher composition in the top product and the heavier components will leave the bottom products.

Model 2 (System of equations of a + b + c + d)

Step 2 consists of the system of equations from Step 1 plus the PI controllers for the accumulator vessel level and bottom. These new equations require new initial values related to the integration term of the PI controller. These initial values can be set as the steady-state values from the previous step, the reflux flow values, and the bottom flow. The steady-state results from the previous step were used as the initial values in this step. Here, the controller was tuned and the steady-state results were saved for the next step.

Model 3 (System of equations of a + b + c + d + e + f)

Model 3 uses the initial values from the previous simulation, in addition to the temperature in each tray and the mass balance in the condenser. The linear temperature profile assumed earlier was kept as the initial values in this step. The equilibrium balances combined with the energy balances calculate the dynamic behavior of the temperature inside the column. The vapor mass balance inside the condenser and the equation of state calculate the pressure in the head. Here, the pressure control was considered perfect, so the liquid holdup inside the condenser is constant ($l_{accum} = l_{cond}$).

Model 4 (System of equations of a + b + c + d + e + f + g)

Model 4 considers the PI pressure controller. The results of the last simulation are used as initial values. The new initial value required is related to the integrator term of the pressure PI controller and is equal to the value of the flow to accumulator vessel.

Model 5 (System of equations of a + b + c + d + e + f + g + h)

In plant operation, the condenser cools the top stream to less than 20 degrees below the boiling point. Product subcooling is typically used to prevent the pump cavitation due to the saturated liquid evaporation.

The final model aims to incorporate the subcooling phenomena to simulate the operating conditions. First, the design boundary conditions are maintained and the subcooling equations are added to the model. The perfect pressure controller is also maintained.

Then, the model is simulated with the design conditions and the results are compared with the design data as in the previous step. The results are saved as the new initial values. Finally, each boundary condition of Table 6.1 is changed to its real operating value. As the new steady state result is achieved, the results are saved as the new initial values until all the boundary conditions are adjusted.

In conclusion, the proposed methodology was aimed at finding good initial values for a rigorous model of a superfractionator. The complete model was divided into four models. In each of them, some of the physical phenomena involved were progressively included until all the phenomena necessary to simulate the operating conditions of the plant were considered. Accordingly, some of the overall initial conditions were successfully computed in each step, resulting in a coherent set of values capable of initializing the full plant simulation.

6.1.1.14 Data reconciliation

Once we were able to run the simulation, we approached model validation in two steps. First, we tried to estimate some unknown model parameters and improve the quality of the process data that we would use for this purpose. This was done through steady-state data reconciliation, taking into account the difficulties of a very large dynamic model like the one we were dealing with. Next, we compared the dynamic model and process responses to the same stimuli to assess the validity of the model as a representation of the plant behavior.

To find any measurement errors caused by instrument malfunction, we used hourly data for almost 4 months in 2019 to verify the mass and energy balance equations.

First, since only the propylene concentration is measured in all the inputs and outputs, we consider the presence of only propane and propylene in the mixture. Then, the volumetric flow measurements were converted to mass flow, and the molar fractions were converted to mass fractions. Thus, the amount of total mass and propylene mass flowing into and out of the column during this time period could be calculated. We assume that by integrating the transient flow rates over a period greater than the time constant of the control volumes, the accumulation is negligible compared to the inflow and outflow values. We also assume that the analyzers are more reliable than the flowmeters, so the following optimization problem could be written as in equation (6.41), where $bias_F, bias_D, bias_B$ refer to the difference between the measured and the actual values of the feed, distillate and bottoms flows, respectively.

$$\begin{gathered}
\min_{bias_F, bias_D, bias_B} \varepsilon_{total}{}^2 + \varepsilon_{C_3H_6}{}^2 \\
\varepsilon_{total} = \sum_{t=0}^{t_f} [(F_t + bias_F) - (D_t + bias_D) - (B_t + bias_B)] \\
\varepsilon_{C_3H_6} = \sum_{t=0}^{t_f} \left[(F_t + bias_F) z_{i,t} - (D_t + bias_D) x_{D,i,t} - (B_t + bias_B) x_{B,i,t}\right] \\
\text{s.t. steady state mass balances,} \\
\text{process constraints}
\end{gathered} \tag{6.41}$$

The problem was solved using the solver in Excel and the results founded were as follows: $bias_F = 0.1624$, $bias_D = 0$ and $bias_B = 0.0920$. Two open-loop simulations were performed: the first with the input flow data and the second with the bias-corrected input flow data. The percentage difference results for the propylene molar fraction concentration in the distillate are shown in Figure 6.3. The bias correction applied to the process data reduced the difference between the simulation and process data. Similar results were obtained for other key simulation variables such as bottom propylene concentration, top and bottom temperatures and reflux flow.

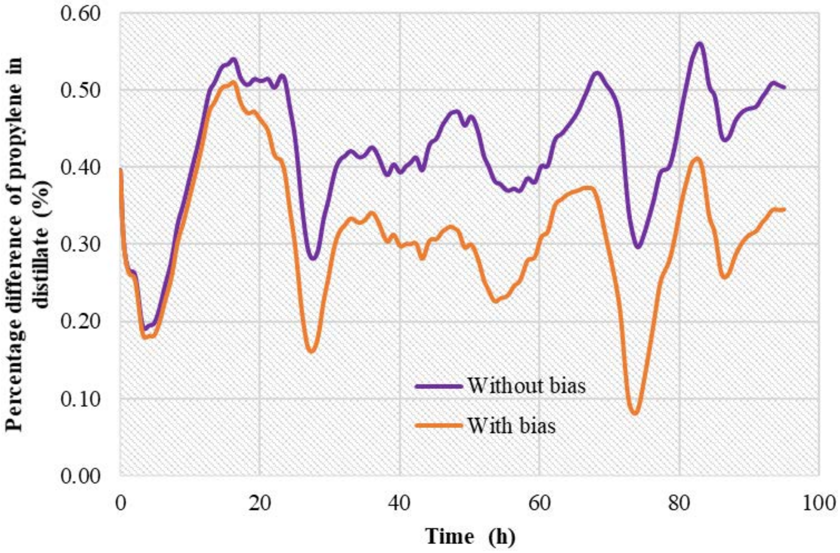


Figure 6.3 Percentage difference between simulations with and without reconciliation and process data.

Using the process data, the column efficiency ε_n in equation (6.13), was estimated to be 1.

6.1.1.15 Model validation

To validate the model, some of the historical process bias-corrected data were compared with the model results. To do so, process data from four consecutive days (one data per hour for four days in June 2019) were used as input to the EcosimPro simulation, including the values of the manipulated variables: distillate flow rate, steam flow rate to the reboiler, and the top pressure. This means that the following validation results do not consider the DMC controller (open loop simulation), while the process data were obtained in closed loop with the DMC operating the plant. Some of the input data used is shown in Figure 6.4 to Figure 6.7. The data has been normalized for confidentiality reasons.

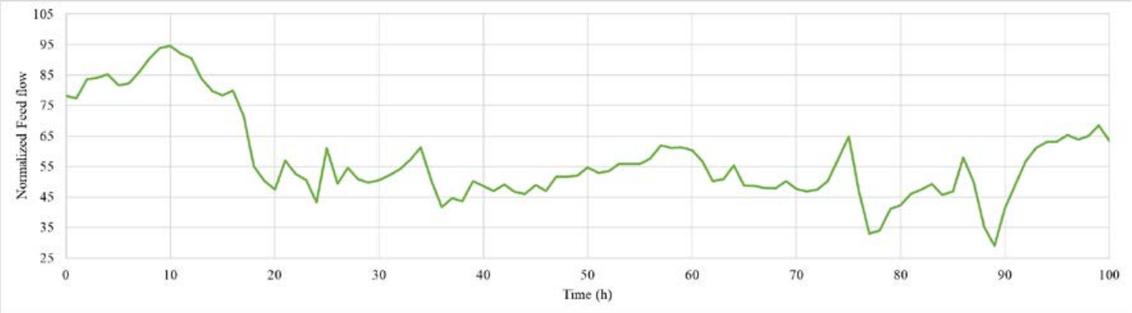


Figure 6.4: Normalized feed flow data used for model validation.

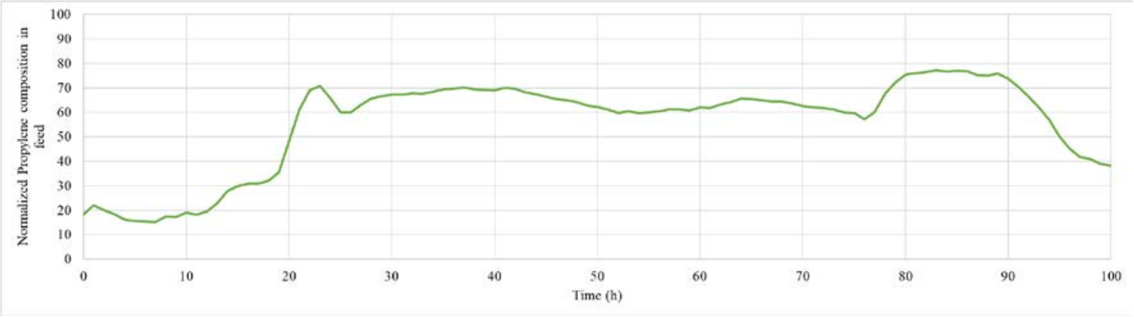


Figure 6.5: Normalized propylene concentration in feed data used for model validation.

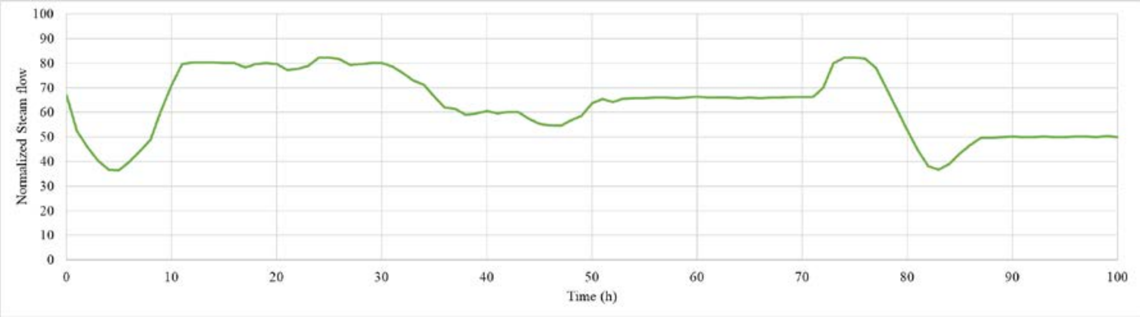


Figure 6.6: Normalized steam flow data used for model validation.

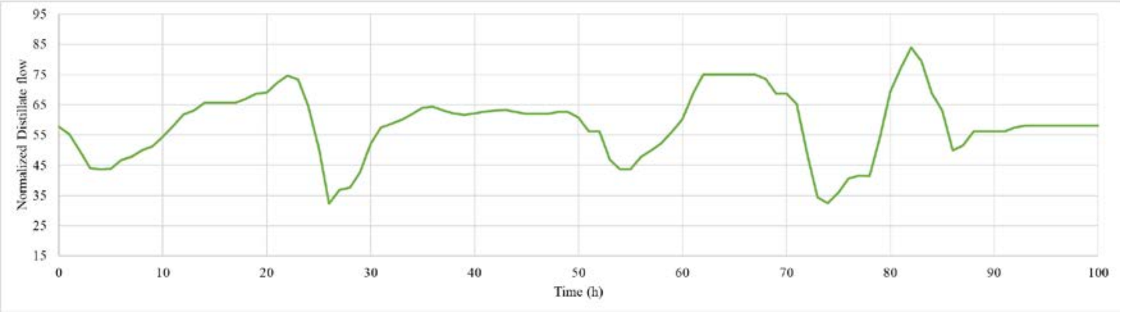


Figure 6.7: Normalized distillate flow data used for model validation.

Figure 6.8 and Figure 6.9 show the behavior of the molar fraction of propylene in the distillate and bottoms versus time.

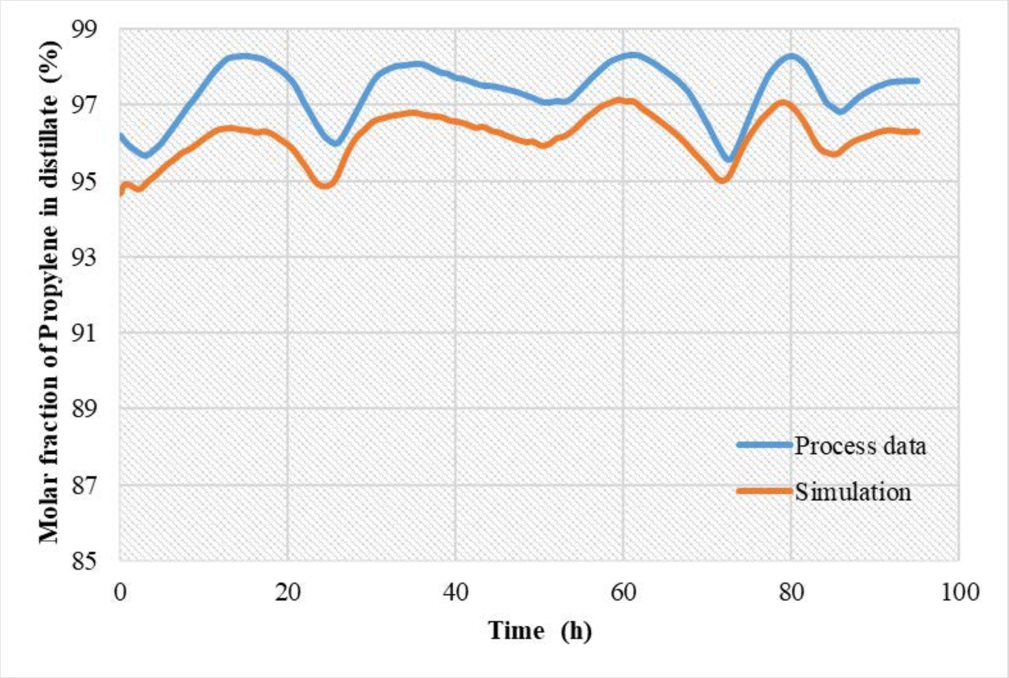


Figure 6.8 Molar fraction of propylene in distillate, open loop.

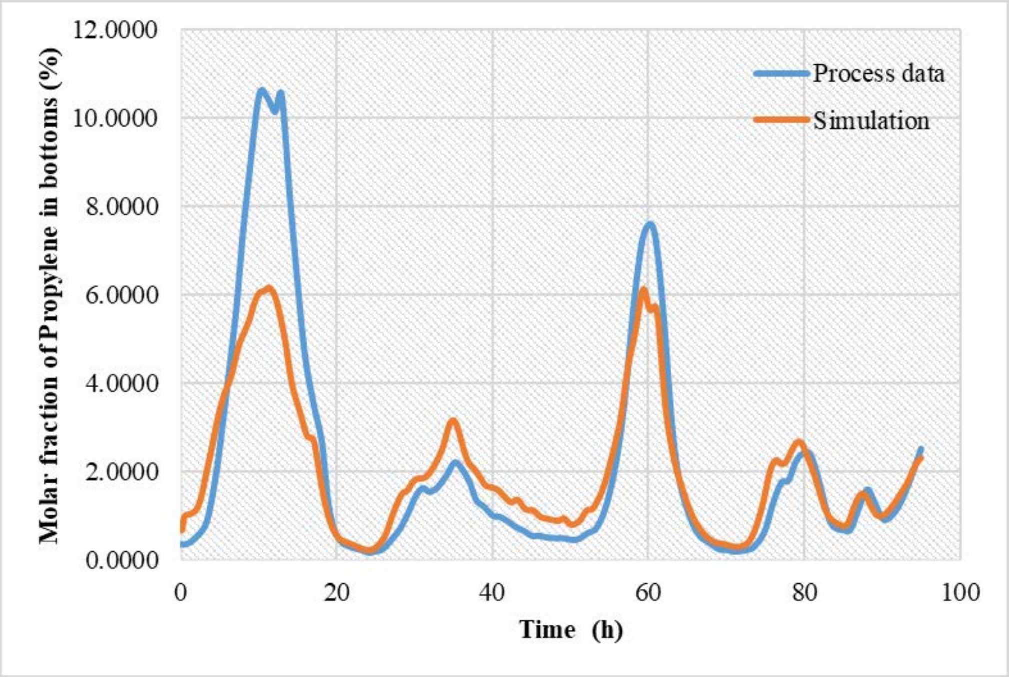


Figure 6.9 Molar fraction of propylene in bottoms, open loop.

The figures show that the rigorous model and the process have a very similar dynamic behavior. Figure 6.8 shows that the process and simulation have similar gains, and a very small bias between the values. Figure 6.9 shows that the dynamic model is able to match the real gain most of the time.

Figure 6.10 and Figure 6.11 present the percentage difference between the temperature profile at the top of the column and the temperature at the bottom, respectively. The percentage difference is defined as equation (6.42).

$$\text{Percentage difference} = \frac{|V_{meas} - V_{sim}|}{\frac{(V_{meas} + V_{sim})}{2}} \times 100 \tag{6.42}$$

where V_{meas} is the value measured and V_{sim} is the value calculated by the simulation.

The process data and simulation show almost an identical behavior with an insignificant offset. The value of the percentage difference is less than 1%. The reflux flow shown in Figure 6.12 also presents a dynamic behavior very similar to the process data. The percentage difference for this case is less than 2%. All these results show that the proposed model predicts the mass and energy dynamics of the column well.

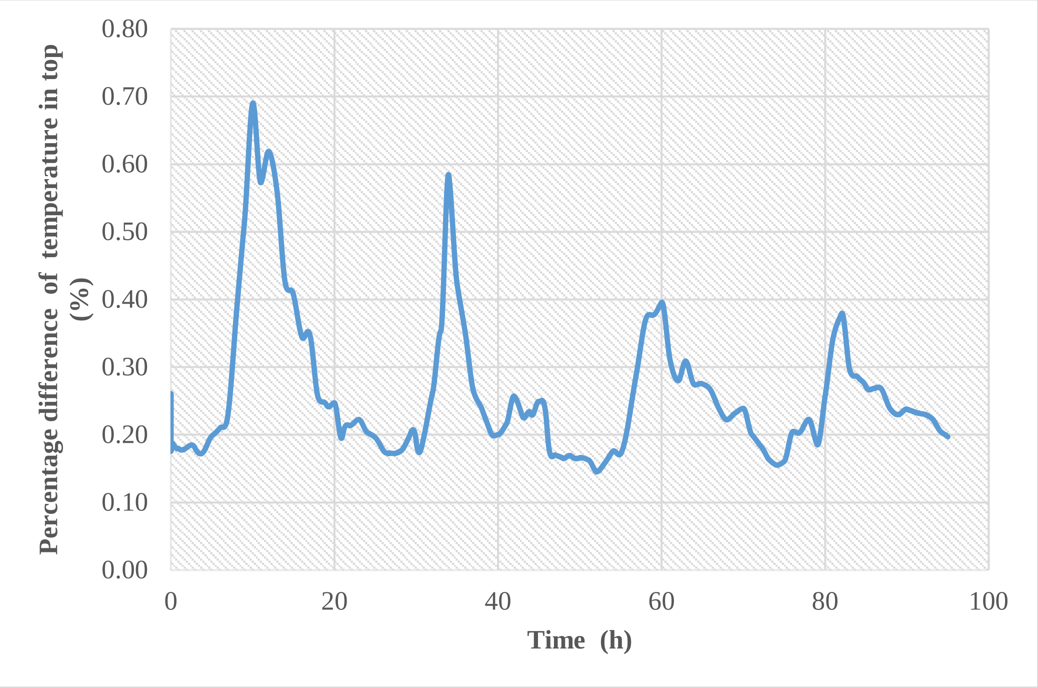


Figure 6.10 Percentage difference of temperature at the top, open loop.

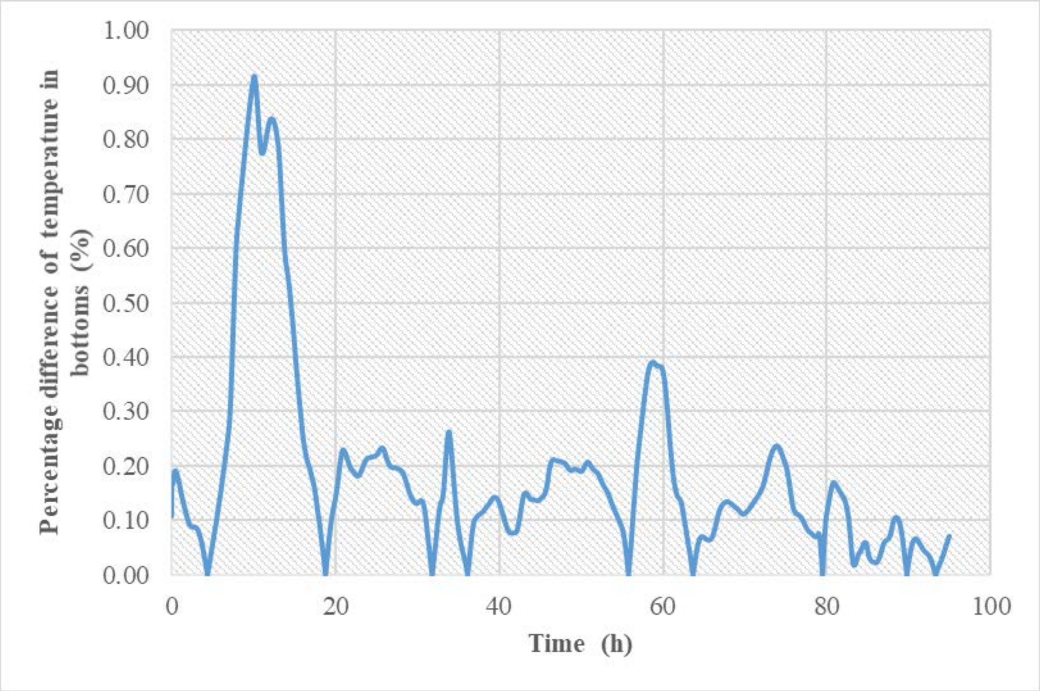


Figure 6.11 Percentage difference of the temperature at the bottom, open loop.

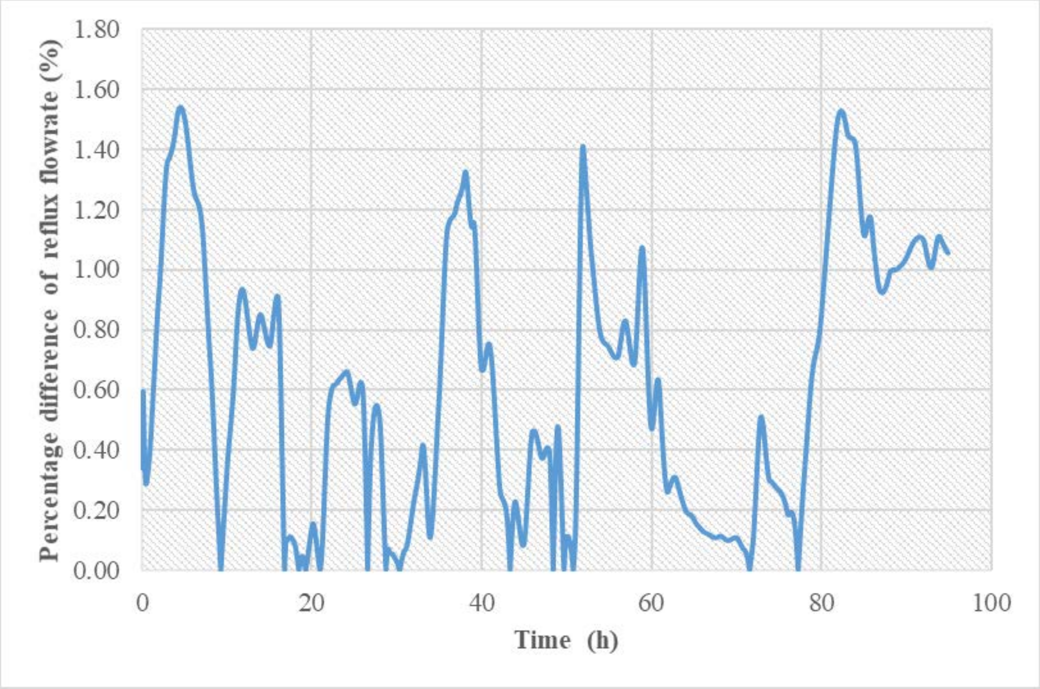


Figure 6.12 Percentage difference of reflux flow rate, open loop.

A similar closed-loop simulation was then performed, with the Aspen DMC controller connected to the EcosimPro deck. This simulation used input data from approximately nine consecutive days (one data per hour during November 2021). More recent data was used because Petronor's DMC model was updated after September 2021, and all of the algorithms presented in this thesis have already been updated with the current splitter DMC model used in the refinery.

The comparison between the results of the Aspen DMC controller in simulation and the historical data is shown in Figure 6.13. All the results presented were normalized due to confidentially reasons. It can be seen that the dynamic behavior of the curves is quite similar, presenting a small offset between them. Probably due to noise in the signal from the instruments, the process data also shows a behavior with more peaks.

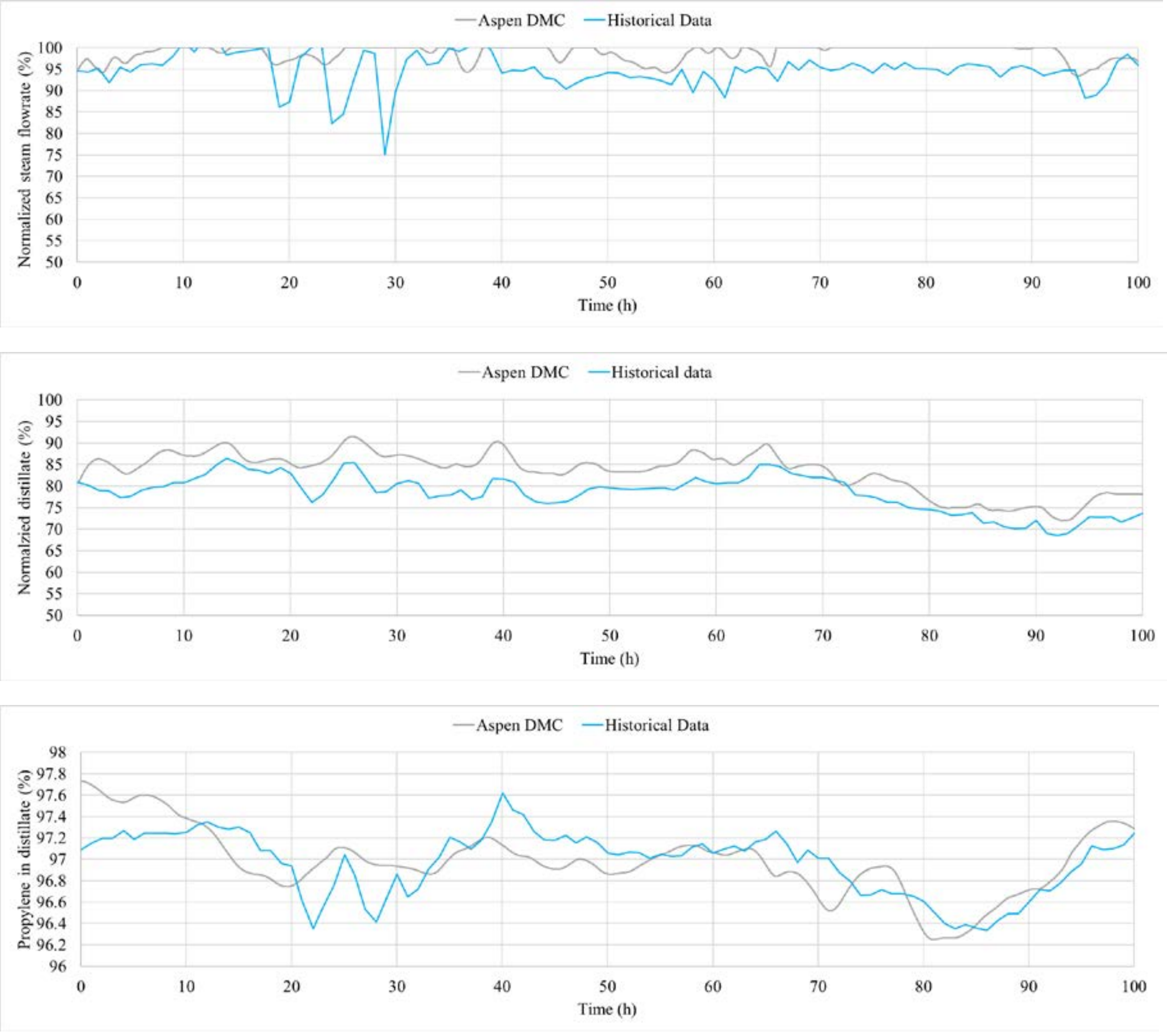


Figure 6.13: Results for the closed loop validation.

The rigorous dynamic model proposed for the splitter has a dynamic response and gains similar to the real process data. In conclusion, the present model is considered good enough for the purpose of mimicking the real plant.

6.2 RTO Architecture Proposal

In the current operation of the splitter, the DMC controller manipulates only the steam and distillate flow, not the head pressure. These two variables manipulated by the DMC are directly related to the main costs (steam) and profits (distillate production) of this plant. Therefore, it seems logical to add a Real Time Optimization (RTO) layer capable of calculating the process optimum (the setpoints for the DMC); using an objective function that takes into account the prices of these two variables and other associated costs. Typically, a nonlinear model based on first principles is used in an RTO, but in our case, due to the large size, complexity, and maintenance requirements of such a model, its use is not recommended. Instead, our goal is to use the already developed linear model of the DMC for this purpose. However, using a model with large parametric and structural uncertainties, such as the linear DMC model, can lead to suboptimality in the process, which leads to the use of MA in the optimization.

In addition, the splitter is a very slow dynamic process that takes about 18 hours to reach a steady state. Therefore, the idea of using MA to estimate the process gradients from the transient data and integrating optimization and control or considering dynamic RTO could reduce the waiting time and increase the performance.

Two different structures were evaluated for the application of economic optimization with MA in the splitter case study:

1- An integrated optimization and control layer (eMPC+MA) + Basic Control + Process.

2- An optimization layer (eMPC+MA) + a control layer (Aspen DMC controller) + Basic Control + Process.

The second option was preferred in order to minimize changes to the actual plant structure. In addition, in structure 2, the optimization layer can be quickly disconnected to return to the previous plant control structure. Therefore, structure 2 corresponds to:

1- Optimization layer: it executes eMPC, eMPC+MA+DME or eMPC+MA+ TMA_m using the dynamic linear model of the DMC. This layer calculates the manipulated variable setpoints for the controller.
2- Controller layer: this layer corresponds to the Aspen DMC Controller with the External Target option enabled.

The optimization layer performs eMPC, eMPC+MA+DME or eMPC+MA+ TMA_m calculations, depending on the user's choice. eMPC corresponds to the economic optimization problem without MA. eMPC+MA+DME solves the same economic problem as eMPC, but adds the MA modifiers in the cost function and the constraints estimated by the DME algorithm. Finally, eMPC+MA+TMA_m corresponds to the economic problem with MAy modifiers using a modification of the TMA algorithm of Section 4, which will be presented in the following sections, to compute the process gradients.

6.3 Components of the Virtual Plant Architecture

This section provides a brief description of the components and software used in the virtual plant architecture. The virtual environment will mimic different layers of the automation pyramid of the real process, as shown in Figure 6.14. The physical process and the basic controllers are represented by an EcosimPro simulation managed by a Real Time Manager (RTM) algorithm developed in Python. The simulation communicates via OPC UA with the MPC controller, represented here by an industrial controller, Aspen DMC. The optimization is represented by a dynamic RTO algorithm, which is implemented as a function developed in Matlab and called by the same RTM algorithm for maintaining synchronization between the virtual process dynamics and the optimization layer.

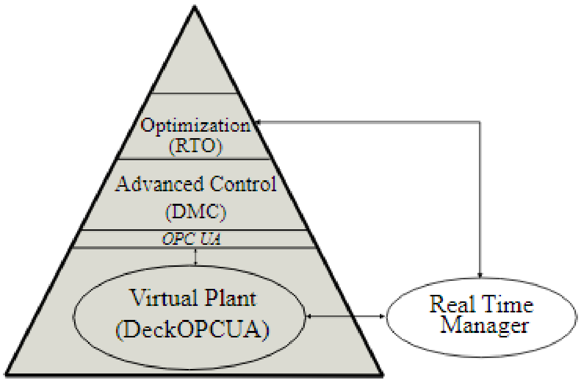


Figure 6.14: Automation pyramid with the virtual plant.

In the following sections, each part of the Virtual Plant environment will be described in detail.

6.3.1 Virtual Process

To connect the simulation with the others applications, the EcosimPro model was converted into a deck. A deck is a simulation model designed to run as a standalone black box completely independent from the main program that can be generated in the EcosimPro environment. There are several types of decks, but the most interesting one from an industrial point of view is the Deck OPC UA Server, in which the experiment is encapsulated as a binary with an OPC-UA layer to act as a server in a network. Every Deck OPC UA Server generated from EcosimPro provides a common set of nodes (see Table 6.2) corresponding to deck variables and commands (nodes corresponding to actions expressed as a variable) that allows a step by step execution of the simulation.

Table 6.2. Deck Variables and Commands.

Tag	**Data Type**	**Description**
TIME	Double	Simulation time
CINT	Double	Communication interval
TSTOP	Double	Simulation final time
command_reset	Int32	Resets the deck when assigned a number different from 0
command_run	Int32	The server runs the experiment defined in the deck when assigned a number different from 0
command_integ_cint	Int32	The server integrates CINT units of time when assigned a number different from 0

For the simulation to evolve in real time, we need an external real time manager described next.

6.3.2 Real Time Manager

The real time manager (RTM) is an OPC UA client that communicates with the Deck OPC UA server that contains the simulation. RTM uses variables and commands that controls the pace of the simulation integration.

The RTM algorithm was programed in Python and the algorithm is presented in Figure 6.15. First, we must define time-related variables of our simulation, like the CINT, TSTOP, and, finally, a speedup factor (SX) if you wish to accelerate your simulation to get results in less time than real time. Next, the connection with the Deck OPC UA Server is established and CINT and TSTOP values are written to the simulation engine. Finally, an iterative loop is executed where the manager invokes the "*command_integ_cint*" command of the Deck OPC UA Server while calculating the computation time needed for that action (ET: Elapsed Time). With this time, the algorithm knows how much time to wait until next integration/iteration so the simulation can run in real time (subject to a possible speedup factor).

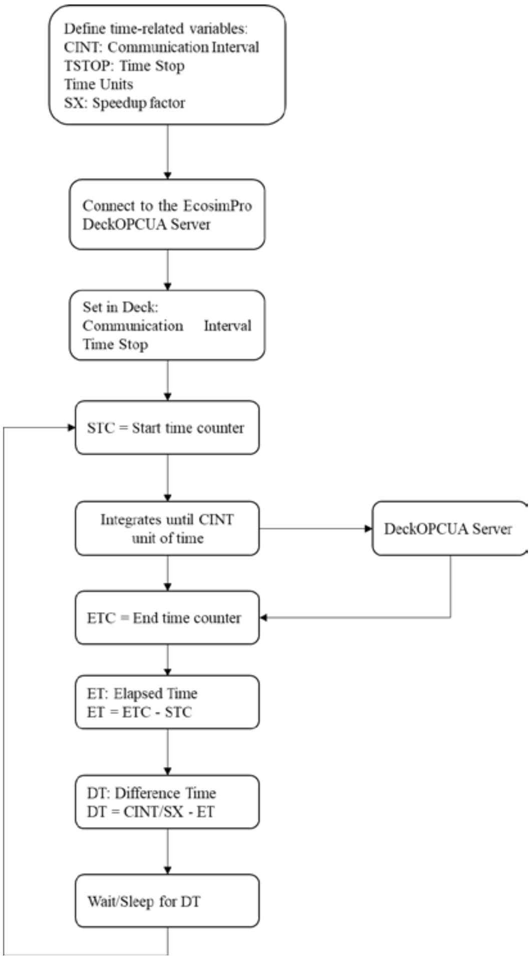


Figure 6.15 RTM algorithm.

Overall, this joint scheme of virtual plant and real-time manager implies that any third-party application with OPC UA capability could communicate with the model in the same way as the real process. In our case study, these external components will be a DMC controller and a real-time optimizer, described in the next subsections.

6.3.3 Aspen DMC controller

DMC (Dynamic Matrix Controller) is a MPC controller that aims to minimize the difference between the prediction of the controlled variables and setpoints, penalizing the MV moves (Camacho and Bordons, 1999). The software DMC plus, and its recent generation DMC3 from AspenTech® is one of the current commercial software based in this technology.

The Aspen DMC controller application has four components: model, filter, optimizer and controller (Figure 6.16). The model aims to calculate the outputs with respect to the change in the inputs. The filter is an observer that estimates the unmeasured disturbances to calculate the current prediction errors of the model. Then the conjunction of the model and filter determines the current dynamic state of the process and the future predictions. The optimizer calculates the best steady-state operating point subject to the constraints. Finally, the unconstrained controller finds the move plan to achieve the set points computed by the optimizer. All process constraints are treated in the optimizer component. To guarantee that a feasible solution will be find, it is possible to choose between minimize the global constraint violation or organize the CVs constraints into a priority order using ranking groups.

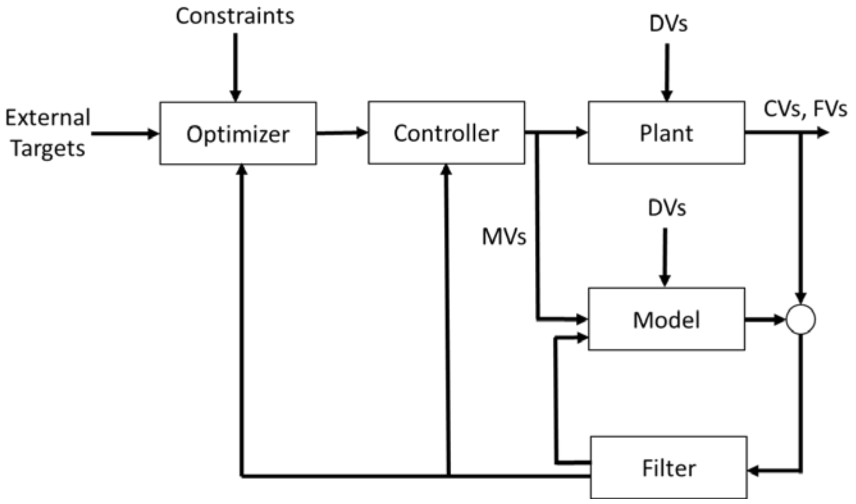


Figure 6.16: Structure of a Aspen DMC Controller (Aspen Technology Inc, 2021). DVs are disturbance variables, MVs are manipulated variables, CVs are controlled variables and FVs are filter variables.

At each sampling time, Aspen DMC runs a two-layer architecture: the optimizer and then the controller. The optimizer calculates the future targets, subject to the process constraints, for the controller as shown in Figure 6.17. The unconstrained controller then calculates the future moves to achieve these targets and applies the first move to the process.

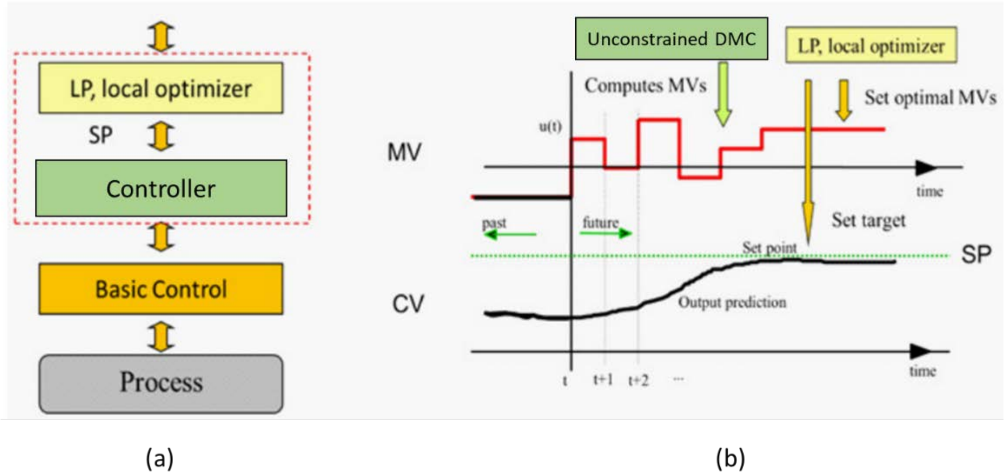


Figure 6.17: (a): The two control layers in Aspen DMC and the connection with the basic control and process; (b) The controller component predicts the future values of the CVs and MVs, considering the set-points calculated from the optimizer (de Prada et al., 2017).

6.3.3.1 Prediction Model

The first step in developing an Aspen DMC application is to build a dynamic model. Aspen DMC uses the linear model FSR (Finite Step Response) obtained from an identification algorithm.

The process input-output data used by the identification algorithm is obtained from a plant test. During the plant test, several step moves are performed in each MV and the plant data is collected. The plant test is the most critical part of the DMC project as it defines the accuracy of the model. An accurate model reduces the time spent in the next steps of controller configuration.

The resulting model represents the open-loop time response of the dependent variable to a step change in each independent variable, while holding all other independent variables constant. For example, in Figure 6.18, each curve represents the behaviour of a dependent variable listed in the top row (AI-2020, AI-2021, AI-2022) for a step change in the corresponding independent variable listed in the left column (FIC-2001, FIC-2002, etc.).

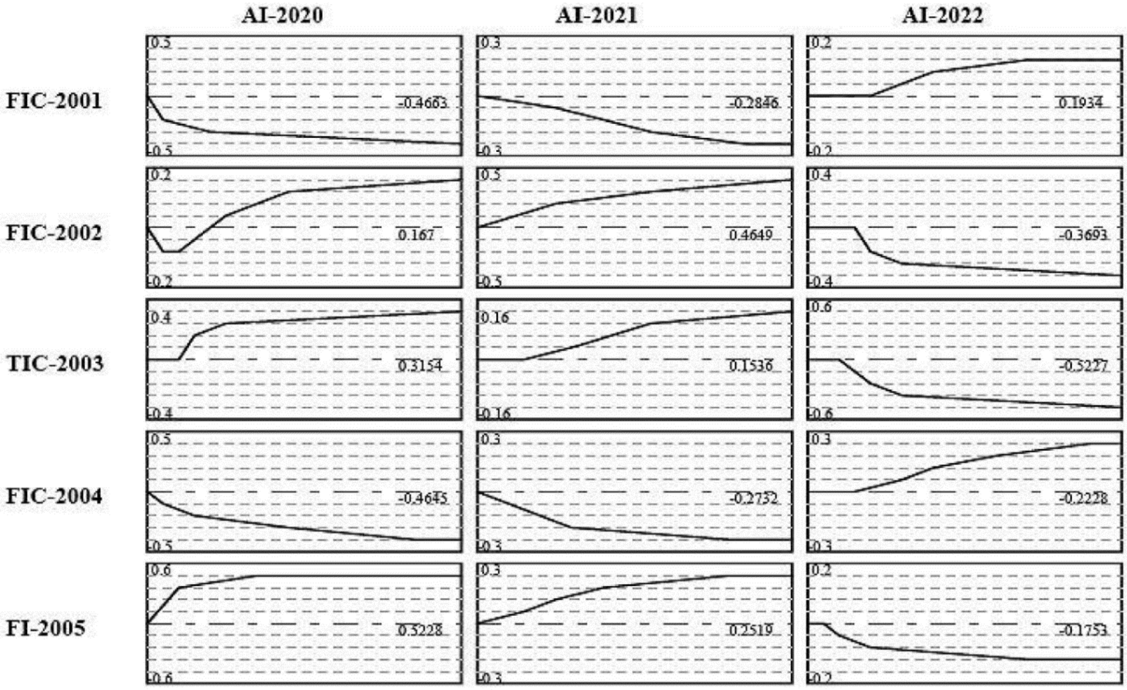


Figure 6.18: Example of a model for a complex fractionator (Aspen Technology Inc, 2021).

The model obtained from the plant test is used to compute predictions of the future behaviour of the CVs in the process. The model uses the entire independent history of changes in the MVs up to one steady-state time in the past. The changes of more than one steady-state time in the past are no longer considered because they no longer affect the current behaviour of the process.

6.3.3.2 Filter

The filter is responsible to predict the unmeasured disturbances of the process. In Aspen DMC three different disturbance models are available for the FSR model: Full Feedback, First Order or Moving Average. Full Feedback uses the difference between the prediction and measurement to calculate a bias to be applied to the prediction for the FSR model. First Order is similar to Full Feedback but permits only a fraction of the differences to be applied to the prediction. Moving Average uses an average of past values of the differences to determine the final difference to correct the prediction.

6.3.3.3 Optimizer

The optimizer uses the same linear model of the controller but in steady state. The optimizer calculates the best economic steady-state target for the CVs and MVs, and the CVs steady-state target becomes the set point for the controller. The process constraints are also considered in the optimizer component and, in fact, this is its main purpose: compute feasible targets at the end of the prediction horizon every sampling time, according to the current operating conditions. The constraints are treated in two different ways: minimizing the global violation of the constraints, and assigning a priority rank for each constraint. The user can select between a Linear Program (LP) or Quadratic Program (QP) solution type for each rank group in the constraints.

The steady-state optimization problem has a feasibility step and an economic step. The first step has the objective of minimizing the give-up for each constraint rank. The user can select a LP or QP optimization for each unique rank. In the LP optimization, the objective function minimizes the weighted sum of the slack variables added to the constraint. In the QP optimization, the objective function minimizes the weighted sum of the squared slack variables added to the constraint. Table 6.3 presents the calculation performed for the LP and QP options, where ε_1, ε_2 are give-up or slack variables and W_1, W_2 are weighting factors. In the feasibility step, the calculation starts with the lowest ranked constraint (the more important constraint) and makes it a hard constraint, i.e., the constraint is relaxed to make it feasible, and then it moves to the next highest rank and repeats the procedure. When all group ranks are hard constraints, the economic optimization step is solved, resulting in the actual steady-state targets. Therefore, the LP and QP differences are only related to the feasibility step.

Table 6.3. Difference between LP and QP calculation.

LP	QP
$min\ \theta = \varepsilon_1 W_1 + \varepsilon_2 W_2 + \cdots$ $subject\ to\ CV \leq CV_{max} + \varepsilon_1$ $CV \geq CV_{min} - \varepsilon_2$ $\varepsilon > 0$	$min\ \theta = {\varepsilon_1}^2 W_1 + {\varepsilon_2}^2 W_2 + \cdots$ $subject\ to\ CV \leq CV_{max} + \varepsilon_1$ $CV \geq CV_{min} - \varepsilon_2$ $\varepsilon > 0$

In the economic optimization step, the following objective function (6.43) is minimized. The objective function has two terms: an economic term and a CV steady-state error penalty term.

$$\theta = \Delta(operating\ cost) + M * (CV\ Steady\ State\ Limit\ Violations) \qquad (6.43)$$

In equation (6.43) M is a large positive number. The second term aims to eliminate the CV steady-state limit violation before minimizing operating cost. The operating cost is calculated as in equation (6.44).

$$\Delta(operating\ cost) = \sum_{i=1}^{n_{MV}} cost_i \cdot \Delta MV_i$$
$$\text{where } cost_i = \left.\frac{\partial(operating\ cost)}{\partial(MV_i)}\right|_{MV_{j \neq i}},\ i,j\ \in \{1, \dots, n_{MV}\} \qquad (6.44)$$

$cost_i$ is the steady-state cost defined as a value that represents the operating cost for a unitary change in MV_i variable, while holding all other MVs constant. The steady-state costs are partial derivatives. These costs can be calculated using an off-line model if one is available. If not, the costs are calculated prior to the plant test performed for the model. A positive steady-state cost means that as the variable decreases, the cost decreases and the profit increases; a negative value means the opposite: as the variable increases, the cost decreases and the profit increases. In processes where there are a few MVs, only one or two MVs dominate the economic optimization. In this case, the user can choose a steady-state cost that has the desired effect. However, in more complex processes, there is no other way but to carefully calculate the steady-state costs as described above. Notice that the choice of the $cost_i$ implies that the optimizer, in addition to finding feasible targets for the controller, will move these targets into a certain "optimal" operating region.

Aspen DMC also permits the input of MVs or CVs as target values for the controller. When the External Targets (ETs) option is selected in the DMC application, the ET calculation is performed after the CV rank feasibility stage and before the economic optimization stage. The ET adds an upper and lower constraint to maintain the CV or MV at a certain value.

Finally, Figure 6.19 shows all the parameters and measurements involved in the computation of the optimum steady state.

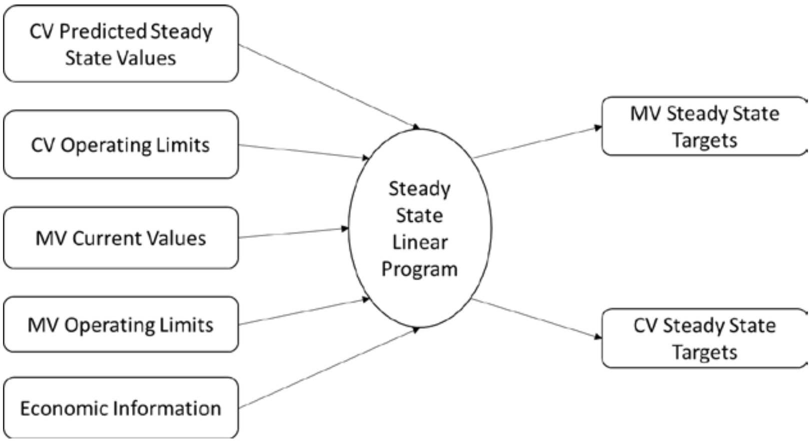


Figure 6.19: Optimizer component inputs-outputs in Aspen DMC (Aspen Technology Inc, 2021).

6.3.3.4 Controller

The controller aims to calculate the move plan to minimize the difference between the predicted value and the steady-state targets calculated in the optimizer. It means to approximate the residual, $\boldsymbol{r}$, in equation (6.45) to zero.

$$\boldsymbol{r} = \boldsymbol{A}\Delta\boldsymbol{u}_{future} - \boldsymbol{e} + \Delta\boldsymbol{u}_{future}\boldsymbol{MS}\Delta\boldsymbol{u}_{future} \approx \boldsymbol{0} \tag{6.45}$$

where $\boldsymbol{A}$ is the dynamic matrix, $\Delta\boldsymbol{u}_{future}$ are the future moves to be calculated and $\boldsymbol{e}$ is the error between the steady state targets and the open loop prediction (or free response) $\boldsymbol{A}\Delta\boldsymbol{u}_{past}$, ($\boldsymbol{e} = \boldsymbol{SP} - \boldsymbol{A}\Delta\boldsymbol{u}_{past}$). To avoid abrupt changes in the MVs, a second objective is added to minimize the magnitude of $\Delta\boldsymbol{u}_{future}$ calculated. $\boldsymbol{MS}$ is a matrix of weights called move suppression. Move suppression coefficients suppress aggressive changes in the MVs and condition the matrix prior inversion (Shridhar and Cooper, 1998). Aspen DMC also considers a parameter called the “move suppression increase factor”. This parameter primarily affects the MV move plan after the fifth move. From the fifth move to the last move in the plan, the move suppression is changed linearly over the remaining coefficients from the value of the move suppression to the value of the multiplication of the move suppression factor by the move suppression increase factor (Aspen Technology Inc, 2021).

In the controller component, the least squares method is applied to (6.45) and after taking the derivative w.r.t. $\Delta\boldsymbol{u}_{future}$ and setting it to zero, one can find the control law as equation (6.46).

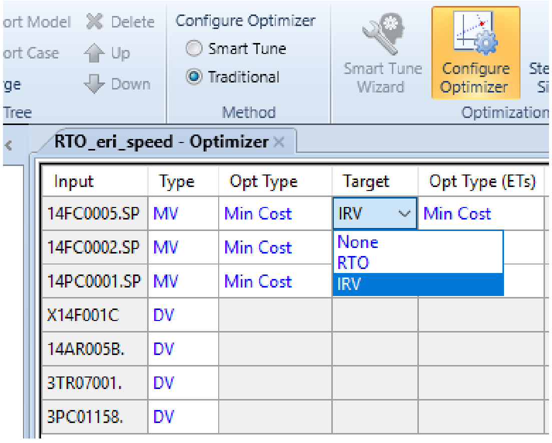
$$\Delta\boldsymbol{u}_{future} = (\boldsymbol{A}^T\boldsymbol{A})^{-\mathbf{1}}\boldsymbol{A}^T\boldsymbol{e} \tag{6.46}$$

In the case of a multivariable control, each CV has a different importance in the process. Multiplying a CV by an artificial coefficient will cause the residuals associated with each CV to be larger, and therefore the algorithm will try to minimize these errors with more emphasis. This coefficient is called the "Equal Concern Error" (ECE). The ECE does not mean that the controller allows that amount of error, but that it comparatively weights the errors in one CV more or less than another.

6.3.4 Dynamic Real Time Optimization

The dynamic optimization layer is written in MATLAB® and it is called by the Python algorithm. The current "target type" of the CVs in Petronor's original DMC application is "none", i.e., the SPs are calculated from the LP (optimizer) as previously explained. To evaluate the optimization structure proposed in this thesis, the DMC application option is changed to type "IRV" (Ideal Resting Value) as shown in Figure 6.19. The IRV mode allows external targets (ETs) to be sent to the Aspen DMC and these values are considered in the LP layer calculations. The IRV was chosen over the RTO because it is simpler than the RTO (the RTO mode in Aspen DMC requires setting five different parameters), there is no staleness check, and there is no steady-state cost exchange.

All the optimizations performed in this layer was solved using the interior-point algorithm, available in the *fmincon* NLP solver of MATLAB®.



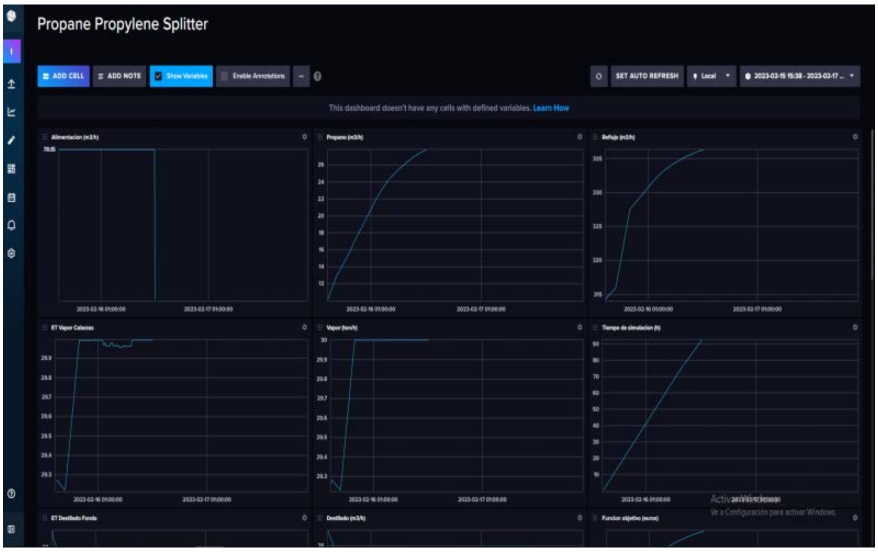
Input	Type	Opt Type	Target	Opt Type (ETs)
14FC0005.SP	MV	Min Cost	IRV	Min Cost
14FC0002.SP	MV	Min Cost		
14PC0001.SP	MV	Min Cost		
X14F001C	DV			
14AR005B.	DV			
3TR07001.	DV			
3PC01158.	DV			

Figure 6.20: Target options in Aspen DMC.

6.3.5 Database InfluxDB

InfluxDB is an open source time series database (TSDB) developed by InfluxData. It is used for storing and retrieving time series data in areas such as operational monitoring, application metrics, Internet of Things (IoT) sensor data, and real-time analytics (Influxdb, 2023).

The process variables of interest are stored in the InfluxDB database, and the software enables visualization, as shown in Figure 6.21.


Propane Propylene Splitter

This dashboard doesn't have any cells with defined variables. Learn How

Figure 6.21: Process variables that are stored in InfluxDB.

6.3.6 Connection between the parts and possible layouts of the Virtual Plant

Aspen DMC3 installations have an architecture as shown in Figure 6.22. The architecture consists of the following components:

- Aspen DMC3 online applications server(s): the machine(s) that hosts the active operation of deployed Aspen DMC3 applications. The server receives data from the Aspen CIM-IO server and hosts security assignments.
- IO sources(s): the machine(s) that provide an interface to the DCS (Distributed Control System). Here we use the Aspen CIM-IO Server.
- DMC3 Builder client(s): machine(s) where Aspen DMC3 Builder is installed. These workstations are used by the engineering staff to develop, update, test and deploy modeling and controller systems.
- Aspen APC Web Interface host: server that host the web site APC Web Interface.
- Aspen APC Web Interface clients with web browser access: machine(s) from which clients can access the APC Web Interface, allowing the engineering staff to monitor the status and manipulate the activities of the deployed control systems.

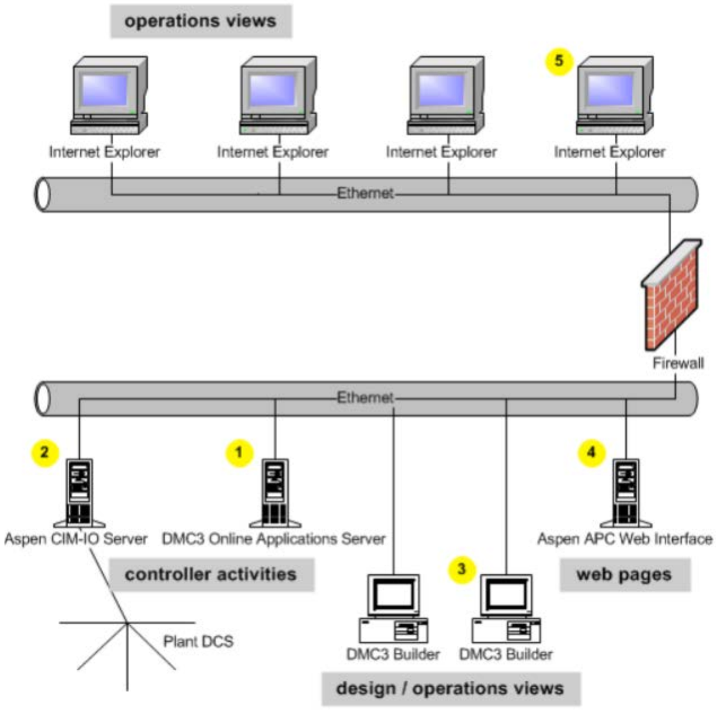
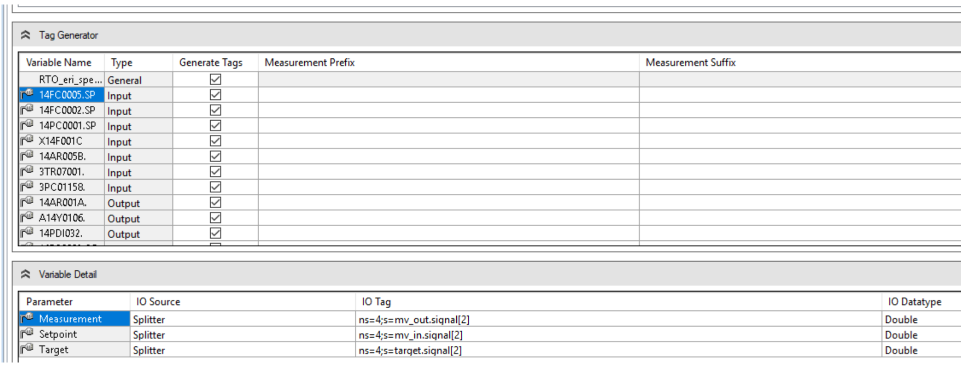


Figure 6.22: Architecture installation of Aspen DMC3 (Aspen Technology Inc, 2021).

The OPC-UA deck was connected to the Aspen DMC controller using the Aspen Cim-IO interface manager application. The same DMC application used in the process was changed to connect to the deck, updating the IO Tag in the Deployment section in DMC3 Builder, Figure 6.23. So the DMC used in the simulation and the plant are the same (same model and configuration). The DMC were monitored using the Aspen APC Web Interface, Figure 6.24.



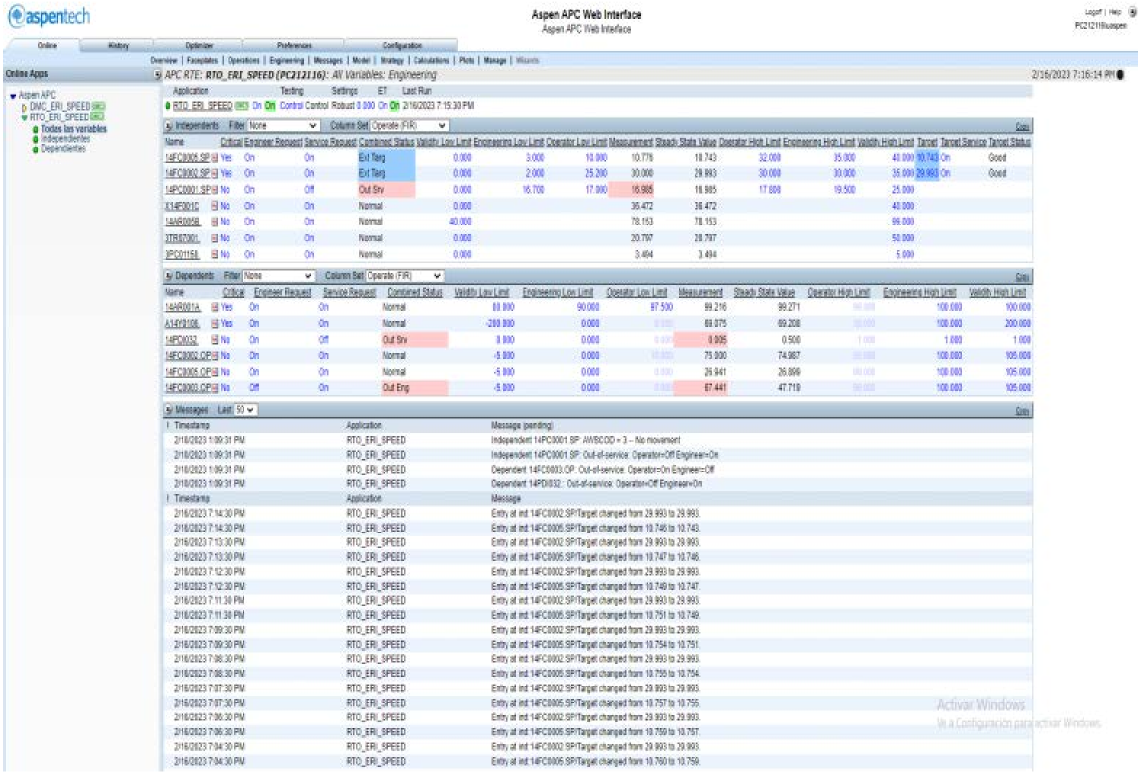
Tag Generator

Variable Name	Type	Generate Tags	Measurement Prefix	Measurement Suffix
RTO_eri_spe...	General	☑		
14FC0005.SP	Input	☑		
14FC0002.SP	Input	☑		
14PC0001.SP	Input	☑		
X14F001C	Input	☑		
14AR005B.	Input	☑		
3TR07001.	Input	☑		
3PC01158.	Input	☑		
14AR001A.	Output	☑		
A14Y0106.	Output	☑		
14PDI032.	Output	☑		

Variable Detail

Parameter	IO Source	IO Tag	IO Datatype
Measurement	Splitter	ns=4;s=mv_out.signal[2]	Double
Setpoint	Splitter	ns=4;s=mv_in.signal[2]	Double
Target	Splitter	ns=4;s=target.signal[2]	Double

Figure 6.23: Connection between deck and Aspen DMC.



APC RTE: RTO_ERI_SPEED (PC212116): All Variables: Engineering 2/16/2023 7:16:14 PM

Application			Testing		Settings		ET		Last Run
RTO_ERI_SPEED	On	On	Control	Control	Robust	0.000	On	On	2/16/2023 7:15:30 PM

Independents Filter None Column Set Operate (FIR)

Name	Critical	Engineer Request	Service Request	Combined Status	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Measurement	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	Target	Target Service	Target Status
14FC0005.SP	Yes	On	On	Ext Targ	0.000	3.000	10.000	10.776	10.743	32.000	35.000	40.000	10.743	On	Good
14FC0002.SP	Yes	On	On	Ext Targ	0.000	2.000	25.200	30.000	29.993	30.000	30.000	35.000	29.993	On	Good
14PC0001.SP	No	On	Off	Out Srv	0.000	16.700	17.000	16.985	16.985	17.800	19.500	25.000			
X14F001C	No	On	On	Normal	0.000			36.472	36.472			40.000			
14AR005B.	No	On	On	Normal	40.000			78.153	78.153			99.000			
3TR07001.	No	On	On	Normal	0.000			20.797	20.797			50.000			
3PC01158.	No	On	On	Normal	0.000			3.494	3.494			5.000			

Dependents Filter None Column Set Operate (FIR)

Name	Critical	Engineer Request	Service Request	Combined Status	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Measurement	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit
14AR001A.	Yes	On	On	Normal	80.000	90.000	97.500	99.216	99.271	[illegible]	100.000	100.000
A14Y0106.	Yes	On	On	Normal	-200.000	0.000	[illegible]	69.075	69.208	[illegible]	100.000	200.000
14PDI032.	No	On	Off	Out Srv	0.000	0.000	[illegible]	0.005	0.500	1.000	1.000	1.000
14FC0002.OP	No	On	On	Normal	-5.000	0.000	[illegible]	75.000	74.987	[illegible]	100.000	105.000
14FC0005.OP	No	On	On	Normal	-5.000	0.000	[illegible]	26.941	26.898	[illegible]	100.000	105.000
14FC0003.OP	No	Off	On	Out Eng	-5.000	0.000	[illegible]	67.441	47.719	[illegible]	100.000	105.000

Messages Last 50

Timestamp	Application	Message (pending)
2/16/2023 1:09:31 PM	RTO_ERI_SPEED	Independent 14PC0001.SP: AWSCOD = 3 -- No movement
2/16/2023 1:09:31 PM	RTO_ERI_SPEED	Independent 14PC0001.SP: Out-of-service. Operator=Off Engineer=On
2/16/2023 1:09:31 PM	RTO_ERI_SPEED	Dependent 14FC0003.OP: Out-of-service. Operator=On Engineer=Off
2/16/2023 1:09:31 PM	RTO_ERI_SPEED	Dependent 14PDI032.: Out-of-service. Operator=Off Engineer=On

Timestamp	Application	Message
2/16/2023 7:14:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0002.SP/Target changed from 29.993 to 29.993.
2/16/2023 7:14:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0005.SP/Target changed from 10.746 to 10.743.
2/16/2023 7:13:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0002.SP/Target changed from 29.993 to 29.993.
2/16/2023 7:13:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0005.SP/Target changed from 10.747 to 10.746.
2/16/2023 7:12:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0002.SP/Target changed from 29.993 to 29.993.
2/16/2023 7:12:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0005.SP/Target changed from 10.749 to 10.747.
2/16/2023 7:11:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0002.SP/Target changed from 29.993 to 29.993.
2/16/2023 7:11:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0005.SP/Target changed from 10.751 to 10.749.
2/16/2023 7:09:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0002.SP/Target changed from 29.993 to 29.993.
2/16/2023 7:09:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0005.SP/Target changed from 10.754 to 10.751.
2/16/2023 7:08:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0002.SP/Target changed from 29.993 to 29.993.
2/16/2023 7:08:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0005.SP/Target changed from 10.755 to 10.754.
2/16/2023 7:07:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0002.SP/Target changed from 29.993 to 29.993.
2/16/2023 7:07:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0005.SP/Target changed from 10.757 to 10.755.
2/16/2023 7:06:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0002.SP/Target changed from 29.993 to 29.993.
2/16/2023 7:06:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0005.SP/Target changed from 10.759 to 10.757.
2/16/2023 7:04:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0002.SP/Target changed from 29.993 to 29.993.
2/16/2023 7:04:30 PM	RTO_ERI_SPEED	Entry at ind 14FC0005.SP/Target changed from 10.760 to 10.759.

Figure 6.24: Aspen APC Web Interface.

The developed Virtual Plant can be used in 4 different architectures. The first one is shown in Figure 6.25. In this architecture, the splitter is not connected to the DMC controller, so its MVs (distillate flow and steam flow) can be changed manually inside the RTM code in Python or any other OPC UA client.

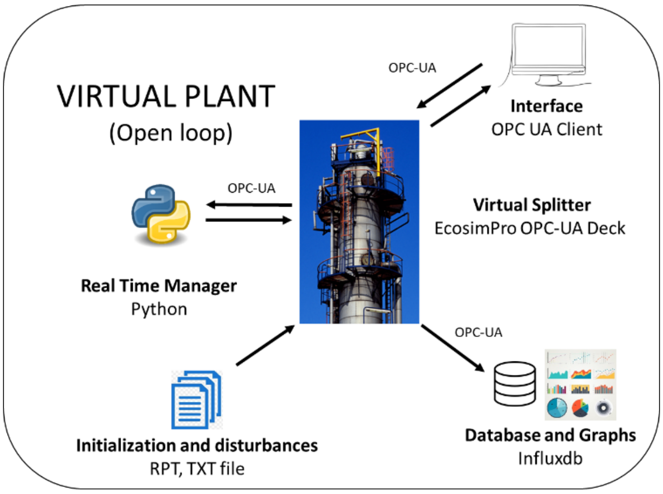


Figure 6.25: Virtual plant in open loop.

The second possible architecture is shown in Figure 6.26. In this architecture, the splitter has the same configuration as in the real plant: in closed loop with DMC controller. In this layout, the MVs targets are calculated by the DMC (subsection 6.3.3.3), which sends the values to the virtual plant via OPC UA.

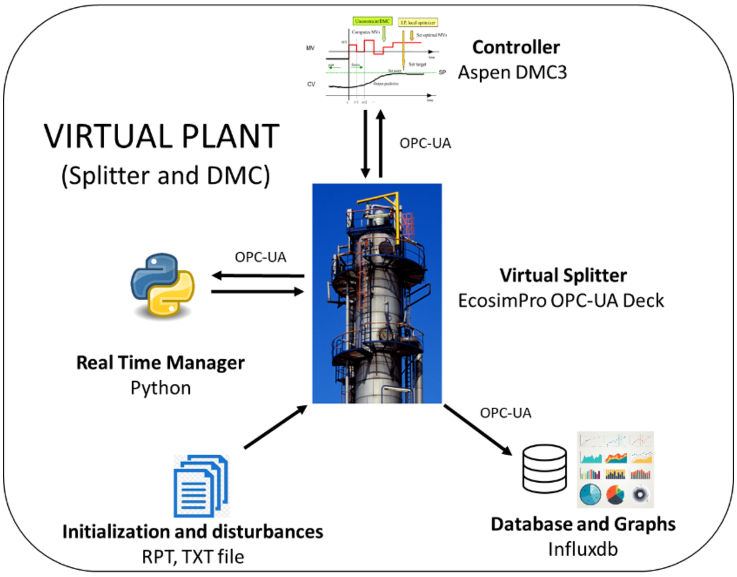


Figure 6.26: Virtual plant in closed loop using Aspen DMC.

The third architecture, Figure 6.27, considers that the eMPC sends directly the moves to the plant, i.e., the RTO and MPC layers are combined into a single layer. This layout could be interesting to make experiments faster than with the Aspen DMC Controller, as it will be explained next.

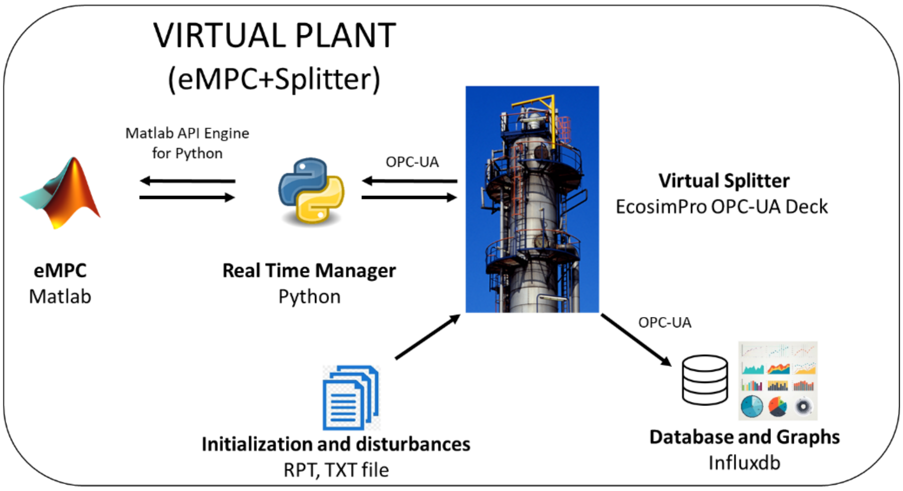


Figure 6.27: Virtual plant in closed loop using eMPC.

Finally, the last possible architecture is presented in Figure 6.28. This is the architecture of interest to test the expected benefits of using an upper supervisory layer based on the dRTO paradigm in the real process.

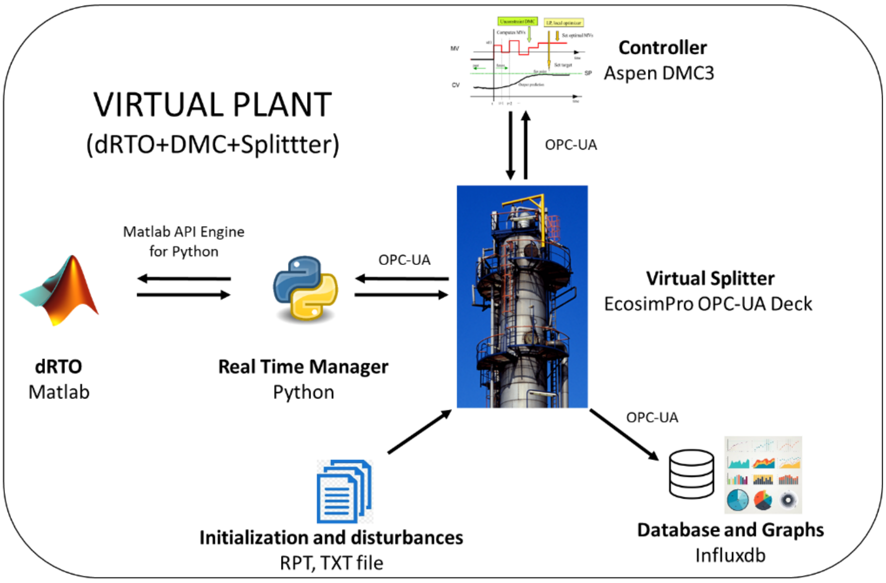


Figure 6.28: Virtual plant in closed loop with Aspen DMC and optimization layer.

The execution of any of the four configurations of the Virtual Plant is roughly done in the following steps:

1. Run the Deck OPC UA Server

2. Start Influxdb to collect data
3. If using DMC: Launch the DMC3 application using Aspen APC Web Interface
4. Set the CINT, TIME, TSTOP, speedup factor in the RTM Python code
5. Execute the RTM algorithm

6.4 Results

In this section, the results of the two economic optimization problems defined for the splitter process are presented. The optimization was performed in four different scenarios with and without disturbances.

The DMC installed in Petronor refinery has a discrete step response model with a settling time of 18h. It was developed with a sampling time of 4.5 minutes and each submodel contains 240 coefficients. The model uses a Full Feedback filter and the optimizer uses the LP type for the constraint rank.

Then, to analyse any result of an RTO, it is necessary to wait at least one steady-state time (18h). In order to reduce the time spent on the experiments, it was necessary to speed up the execution of the virtual plant. For this purpose, the execution frequency of the DMC controller was reduced, decreasing the sampling period of the model. The ASPEN DMC controller performs several operations during the cycle period (such as reading the data, solving the linear programming, calculating the movements and then sending the data to the process), so a sampling period of less than 1 min is impracticable. Therefore, the actual value of 4.5 min was reduced to 1 min, and now one steady-state time of 18h can be simulated in 4h (a speed-up factor in the simulation of 4.5). It was also necessary to improve the performance of the ASPEN DMC server computer. All results presented below were performed on a PC running Windows Server 2019, with a 24-core i9 processor running at 3 GHz and with 128 GB of memory.

The economic cost function in the RTO layer considers the steady state profit from selling propylene and propane minus the cost to produce steam. The profit, $\phi_{splitter}$, is represented by equation (6.47), where F_D, F_B and F_W are the distillate, bottoms and steam flow rates respectively. The bar above them indicates that the values are in steady state. x_d is the molar concentration of propylene in the distillate. $p_{C_3H_6,spec}$, $p_{C_3H_6,off}$, $p_{C_3H_8,spec}$, $p_{C_3H_8,off}$ and p_W are the prices of distillate on specification, distillate out of specification, bottoms product on specification, bottoms product out of specification and the cost to produce steam, respectively. The product prices depend on the propylene concentration in the distillate (≥ 97.5% molar), i.e., below this target, the price decreases as a function of the composition. All the distillate production is sent to a propylene spherical tank. When the tank is full, the final composition is measured and if it is within specification (≥ 97.5% molar propylene), the product can be sold. If the quality composition is not achieved, the product must be reprocessed, which significantly increases the production costs. To avoid this reprocessing, the dRTO problem will consider reducing the price of the off-spec product to compensate for the loss of profit in producing less distillate of higher purity. For example, if 96.5% propylene distillate is produced in a given time period, the same amount of higher purity propylene distillate (98.5%) must be produced to meet the specification in the tank. Thus, a vector of composition and prices has been constructed, see equation (6.47), and using the Modified Akima piecewise cubit hermit interpolation (*makima* function in Matlab®), the distillate price changes with the composition as shown in Figure 6.29.

$$\begin{gathered}\phi_{splitter} = p_{C_3H_6}\overline{F_D} + p_{C_3H_8}\overline{F_B} - p_W\overline{F_W}, \\ x_d = [0;\ 90;\ 95;\ 95.9;\ 96.7;\ 97.5;\ 98;\ 100] \\ p_{C_3H_6}(x_d) = [0;\ 0;\ 0;\ 0.9185;\ 0.9873;\ 1.0610;\ 1.0610;\ 1.0610]\end{gathered} \tag{6.47}$$

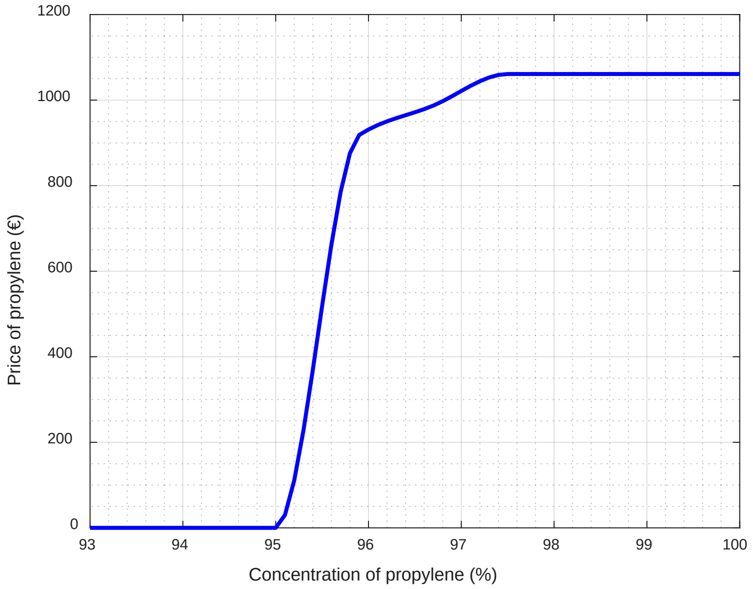


Figure 6.29: Graphical representation of distillate price with the concentration of propylene, equation (6.47).

Four different scenarios with different characteristics were considered (see Table 6.4). In scenario 1, the more valuable product is the distillate (propylene) and in scenario 2, the more valuable product is the bottoms product (propane). Only in Scenario 2, the specification of propane is a maximum of 20% propylene in the bottoms product and the price of propane varies with the concentration of propane, as shown in Figure 6.30. In Scenarios 1 and 2, the process disturbances (feed flow rate, feed composition, etc.) are held constant. Scenario 3 corresponds to the same prices as Scenario 1, but with disturbance values taken from the refinery's historical data. Finally, Scenario 4 is similar to Scenario 3 but with a different objective function that it will be presented in the section 6.4.4.

Table 6.4. Scenarios considered in the simulation.

Scenario	Price Distillate on-spec $p_{C_3H_6,spec}$ (€/tonne)	Price Distillate off-spec $p_{C_3H_6,off}$ (€/tonne)	Price Bottoms on-spec $p_{C_3H_8,spec}$ (€/tonne)	Price Bottoms off-spec $p_{C_3H_8,off}$ (€/tonne)	Cost Steam p_W (€/tonne)	Presence of Disturbances
1	1061	$p_{C_3H_6}(x_d)$	634	-	34.0	No
2	634	-	1061	$p_{C_3H_8}(x_b)$	34.0	No
3	1061	$p_{C_3H_6}(x_d)$	634	-	34.0	Yes
4*	1061	$p_{C_3H_6}(x_d)$	634	-	34.0	Yes

*different objective function

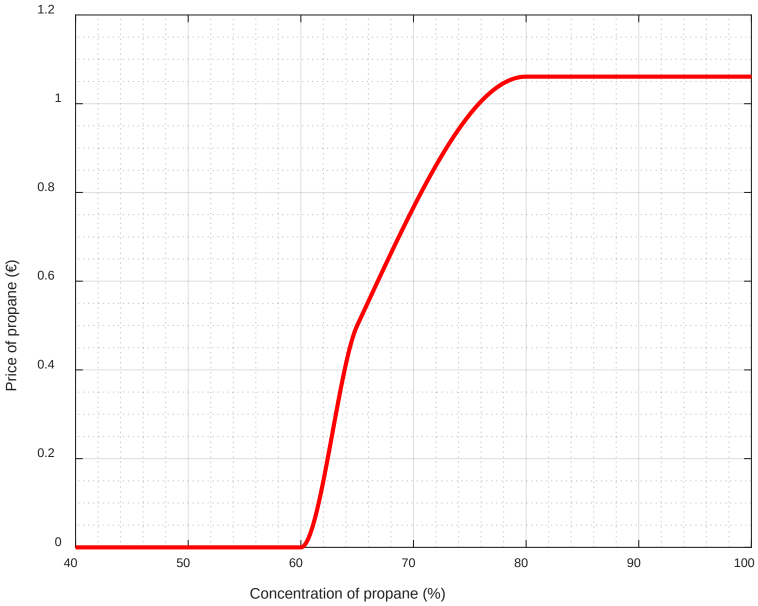


Figure 6.30: Graphical representation of bottom product price with the concentration of propane for Scenario 2.

Four different architectures were run for each scenario:

1. Aspen DMC: simulation with the ASPEN DMC controller. The current configuration of the real plant.
2. eMPC + Aspen DMC: simulation with the optimization layer without MA. For Scenarios 1 and 3, the optimization problem solved is problem (6.48), where a similar cost function of eMPC+MA is presented without the MA terms. ${F_D}^L$ and ${F_D}^U$ are the lower and upper limits of the distillate flow rate, respectively. ${F_W}^L$ and ${F_W}^U$ are the lower and upper limits for steam flow rate, respectively. Actual values are not shown for confidentiality reasons. In Scenario 2, the optimization problem solved is (6.50).
3. eMPC + DME + Aspen DMC: simulation with the optimization layer with MA. The MA modifiers are calculated using the DME algorithm. In Scenarios 1 and 3, the optimization problem solved is problem (6.49) and for Scenario 2 the problem solved is (6.51). k indicates the RTO iteration. Here, the MAy modifiers are used instead of the traditional ones. One reason for this is that in Chapter 5, MAy performed better in the constrained case without the need for a good initial point. Also, MAy requires a smaller number of modifiers in the case study.
4. eMPC + TMA$_m$ + Aspen DMC: simulation using the optimization layer with MA. The MA modifiers are calculated using the TMA$_m$ algorithm. TMA$_m$ is a modification of the TMA algorithm presented in section 4 and it is introduced in the next section 4.2. Here, the MAy modifiers are also used instead of the traditional ones. The optimization problem to be solved is the same problem (6.49) for Scenario 1 and 3, and problem (6.51) for Scenario 2.

$$\min_{F_{D,i},F_{W,i}} \left(-\phi_{splitter} + \sigma_{u_1} \textstyle\sum_{i=1}^{n_u} \left(\Delta F_{D,i}\right)^2 + \sigma_{u_2} \textstyle\sum_{i=1}^{n_u} \left(\Delta F_{W,i}\right)^2\right)$$

$$\begin{gathered} \text{s.t. dynamic linear model of Aspen DMC} \\ 97.5 - \overline{x_d} \leq 0 \\ {F_D}^L \leq F_D \leq {F_D}^U \\ {F_W}^L \leq F_W \leq {F_W}^U \\ -0.7 \leq \Delta F_D \leq 0.7 \\ -0.5 \leq \Delta F_W \leq 0.5 \end{gathered} \tag{6.48}$$

$$-7.0 \leq \sum_{i=1}^{n_u} \left(\Delta F_{D,i}\right)^2 \leq 7.0$$

$$-5.0 \leq \sum_{i=1}^{n_u} \left(\Delta F_{W,i}\right)^2 \leq 5.0$$

$$\min_{F_D,F_W} \left(-\phi_{splitter,M} + \sigma_{u_1} \sum_{i=1}^{n_u-1} \left(\Delta F_{D,k+i}\right)^2 + \sigma_{u_2} \sum_{i=1}^{n_u-1} \left(\Delta F_{W,k+i}\right)^2\right)$$

s.t. dynamic linear model of Aspen DMC

$$\phi_{splitter,M} = p_{C_3H_6,M}\overline{F_D} + p_{C_3H_8}\overline{F_B} - p_W\overline{F_W},$$

$p_{C_3H_6,M} = p_{C_3H_6}\left(\overline{x_d}_M\right)$ equation (6.47)

$$97.5 - \overline{x_d}_M \leq 0$$

$$\overline{x_d}_M = \overline{x_d} + \gamma_{1.k}\left(\overline{F_D} - F_{D,k-1}\right) + \gamma_{2,k}\left(\overline{F_W} - F_{W,k-1}\right) + \varepsilon_1$$

Modifiers are calculated using DME or TMA$_m$

$${F_D}^L \leq F_D \leq {F_D}^U \tag{6.49}$$

$${F_W}^L \leq F_W \leq {F_W}^U$$

$$-0.7 \leq \Delta F_D \leq 0.7$$

$$-0.5 \leq \Delta F_W \leq 0.5$$

$$-7.0 \leq \sum_{i=1}^{n_u} \left(\Delta F_{D,i}\right)^2 \leq 7.0$$

$$-5.0 \leq \sum_{i=1}^{n_u} \left(\Delta F_{W,i}\right)^2 \leq 5.0$$

$$\min_{F_{D,i},F_{W,i}} \left(-\phi_{splitter} + \sigma_{u_1} \sum_{i=1}^{n_u} \left(\Delta F_{D,i}\right)^2 + \sigma_{u_2} \sum_{i=1}^{n_u} \left(\Delta F_{W,i}\right)^2\right)$$

s.t. dynamic linear model of Aspen DMC

$$97.5 - \overline{x_d} \leq 0 \tag{6.50}$$

$$\overline{x_b} - 20.0 \leq 0$$

$${F_D}^L \leq F_D \leq {F_D}^U$$

$${F_W}^L \leq F_W \leq {F_W}^U$$

$$-0.7 \leq \Delta F_D \leq 0.7$$

$$-0.5 \leq \Delta F_W \leq 0.5$$

$$-7.0 \leq \sum_{i=1}^{n_u} \left(\Delta F_{D,i}\right)^2 \leq 7.0$$

$$-5.0 \leq \sum_{i=1}^{n_u} \left(\Delta F_{W,i}\right)^2 \leq 5.0$$

$$\min_{F_D,F_W} \left(-\phi_{splitter,M} + \sigma_{u_1} \textstyle\sum_{i=1}^{n_u-1} \left(\Delta F_{D,k+i}\right)^2 + \sigma_{u_2} \textstyle\sum_{i=1}^{n_u-1} \left(\Delta F_{W,k+i}\right)^2\right)$$

s.t. dynamic linear model of Aspen DMC

$$\phi_{splitter,M} = p_{C_3H_6}\overline{F_D} + p_{C_3H_8}\overline{F_B} - p_W\overline{F_W},$$

$$p_{C_3H_8,M} = p_{C_3H_8}\left(\overline{x_b}_M\right)$$

$$\overline{x_b}_M - 20.0 \leq 0$$

$$\overline{x_b}_M = \overline{x_b} + \gamma_{3.k}\left(\overline{F_D} - F_{D,k-1}\right) + \gamma_{4,k}\left(\overline{F_W} - F_{W,k-1}\right) + \varepsilon_2$$

Modifiers are calculated using DME or TMA$_m$

$$F_D{}^L \leq F_D \leq F_D{}^U \tag{6.51}$$

$$F_W{}^L \leq F_W \leq F_W{}^U$$

$$-0.7 \leq \Delta F_D \leq 0.7$$

$$-0.5 \leq \Delta F_W \leq 0.5$$

$$-7.0 \leq \sum_{i=1}^{n_u} \left(\Delta F_{D,i}\right)^2 \leq 7.0$$

$$-5.0 \leq \sum_{i=1}^{n_u} \left(\Delta F_{W,i}\right)^2 \leq 5.0$$

All of the scenarios started from the same steady state. The eMPC layer uses the same parameters of Aspen DMC controller (move suppression values, control horizon, etc), and runs at the same frequency, 4.5 min (as explained before, in the experiments the simulation was accelerated and real frequency is 1.0 min).

In all the simulations performed, the eMPC was started after 20 min and the DME or TMA$_m$ algorithm, if selected, was started after 1h in order to initialize the algorithms properly. DME or TMA$_m$ initialize later than the eMPC because these algorithms requires some past data. In the case of TMA$_m$, the identification algorithm requires a minimum excitation to make a good estimation of the parameters. At each iteration, the eMPC layer calculates the move plan and sends the first movement as an external target to the DMC controller.

6.4.1 Scenario 1

Table 6.5 presents the parameters used in the simulation of Scenario 1 to calculate the modifiers and gradients for the DME and TMA$_m$ algorithms.

Table 6.5. Parameters used in DME and TMA$_m$ for Scenario 1.

Parameters	**Values**
DME parameters	$n_{DME} = 3$ $Q_{x_d} = [0, 0]$ $\sigma_{x_d} = 0.1$ $K_{\gamma,\varepsilon} = 0.8$
TMA$_m$ parameter	$\mu = 1.0$ $K_{\gamma,\varepsilon} = 0.8$

The optimal result is to maximize the distillate flow while maintaining the quality constraint at the lower limit (97.5%). The DMC optimizer uses the values of LPCost in order to maximize the distillate flow and minimize the steam flow. However, the cost of steam is not as significant as the price of distillate.

Figure 6.31 shows that all the control architectures achieve the optimum in the steady state (after 80 hours of operation), but the performance in the transient period is very different. Figure 6.32 shows the cost function value (profit) of the process. The changes in the cost function in Figure 6.31 is directly associated with the constraint in Figure 6.32 and distillate flow in Figure 6.34.

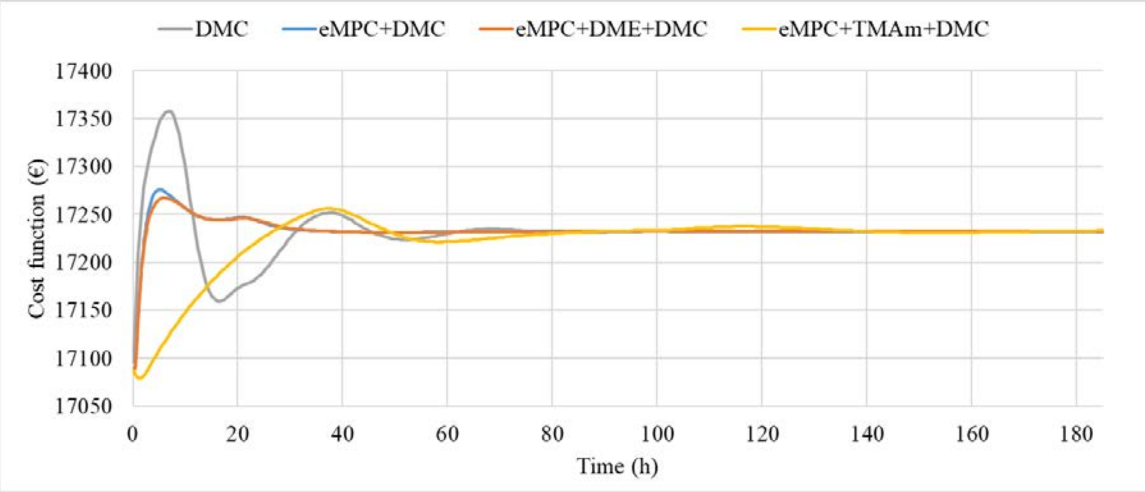


Figure 6.31: Cost function from scenario 1 with different control architecture.

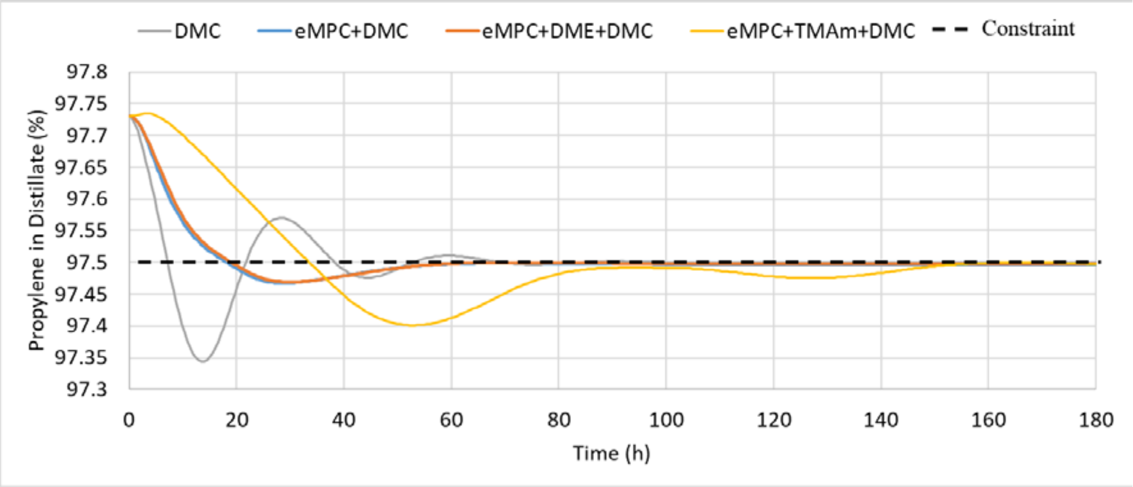


Figure 6.32: Propylene concentration from scenario 1 with different control architecture.

The profit at steady state is presented in Table 6.6. All the architectures performed similar and for this case, the inclusion of eMPC or eMPC+MA does not seem to increase the economic performance because for this case the model optimum is equal to the process optimum (at the constraint).

Table 6.6. Profit at steady state for each architecture of scenario 1.

Control Architecture	**Profit at steady state (€)**
Aspen DMC	17231.86
eMPC + Aspen DMC	17232.51
eMPC + DME + Aspen DMC	17232.34
eMPC + TMA$_m$ + Aspen DMC	17233.53

The move plan of the MVs is shown in Figure 6.33 and Figure 6.34. From a control perspective, the application of an optimization layer allows for a smoother control because the eMPC objective function aims to maximize profit at steady state minimizing the control efforts.

The TMA$_m$ presented a slower response probably because this method needs excitation and more information about the process as TMA$_m$ uses an identification algorithm to estimate the process gradients.

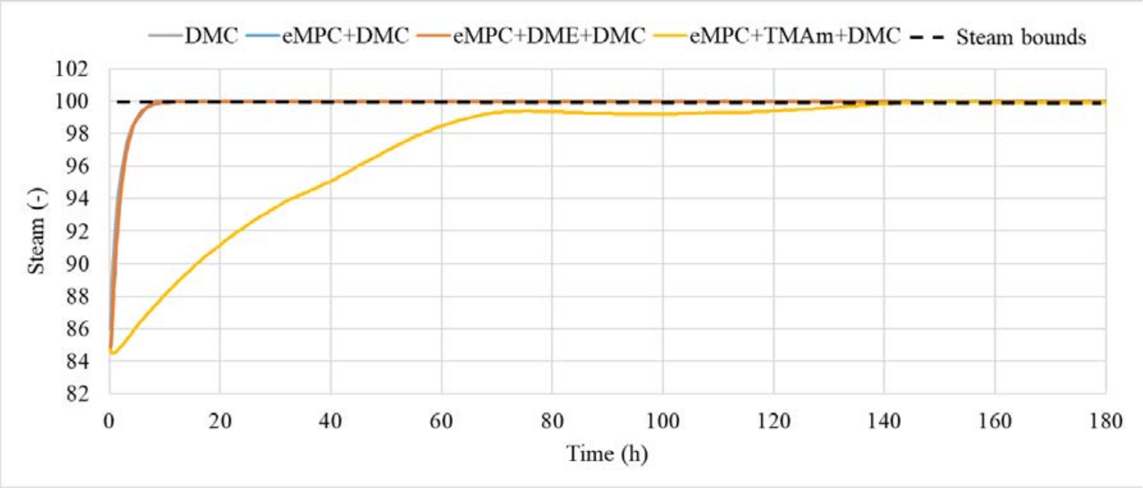


Figure 6.33: Normalized steam flow rate from scenario 1 with different control architecture.

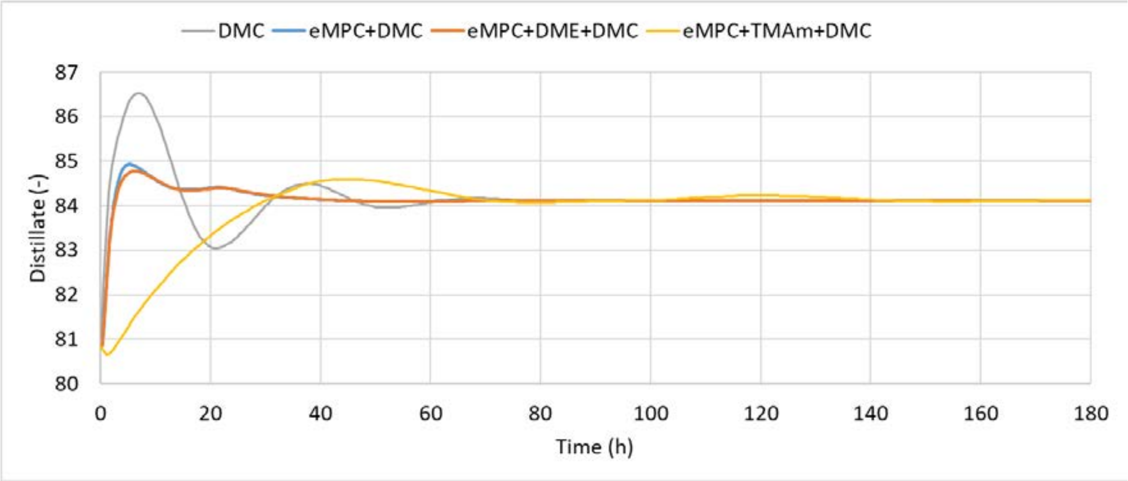


Figure 6.34: Normalized distillate flow rate from scenario 1 with different control architecture.

6.4.2 Scenario 2

Table 6.7 shows the parameters used in the Scenario 2 simulation to calculate the modifiers and gradients for the DME and TMA$_m$ algorithms, respectively.

Table 6.7. Parameters used DME and TMA$_m$ for scenario 2.

Parameters	**Values**
DME parameters	$n_{DME} = 3$ $Q_{x_b} = [0, 0]$ $\sigma_{x_b} = 0.1$ $K_{\gamma,\varepsilon} = 0.8$
TMA$_m$ parameters	$\mu = 1.0$ $K_{\gamma,\varepsilon} = 0.8$

The results of Scenario 2 are presented in Figure 6.35. In this scenario, the bottom product (propane) has the highest price and it varies with the propane concentration.

In this scenario, we assume that LP prices are not recalculated for the Aspen DMC controller. Therefore, the performance would be the same as in scenario 1 and only the cost function is updated. Scenario 2 reinforces the advantage of the proposed eMPC architecture, which is the flexibility against product prices changes. In the current plant architecture, it would be necessary to recalculate the LP prices for the Aspen DMC controller using the strategy in Section 6.3.3.3. In the eMPC or eMPC+MA, on the other hand, the prices are changed directly in the algorithm and no prior calculation or analysis is required. It would also be necessary to include the modifiers for the new variable x_b and add the corresponding constraints. However, these changes could be implemented in the algorithm and used as needed by simply setting the modifiers to zero.

The results show that eMPC and eMPC+MA reach a better profit than the current control structure. Between eMPC, eMPC with DME and eMPC with TMA the profit at steady state is similar, as can be seen in the Figure 6.35. However, eMPC with TMA have more oscillations.

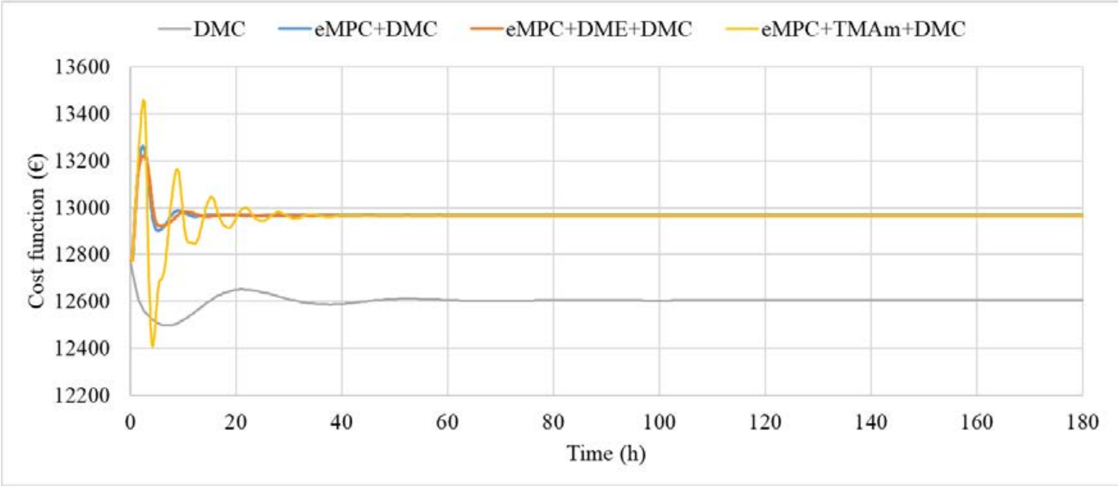


Figure 6.35: Cost function from scenario 2 with different control architecture.

The optimum is like the Scenario 1, that you are maximizing the amount of the more valuable product in specification, in this case propane in bottoms. So in this scenario, the quality constraint is again active in the optimum, as can be seen in Figure 6.36.

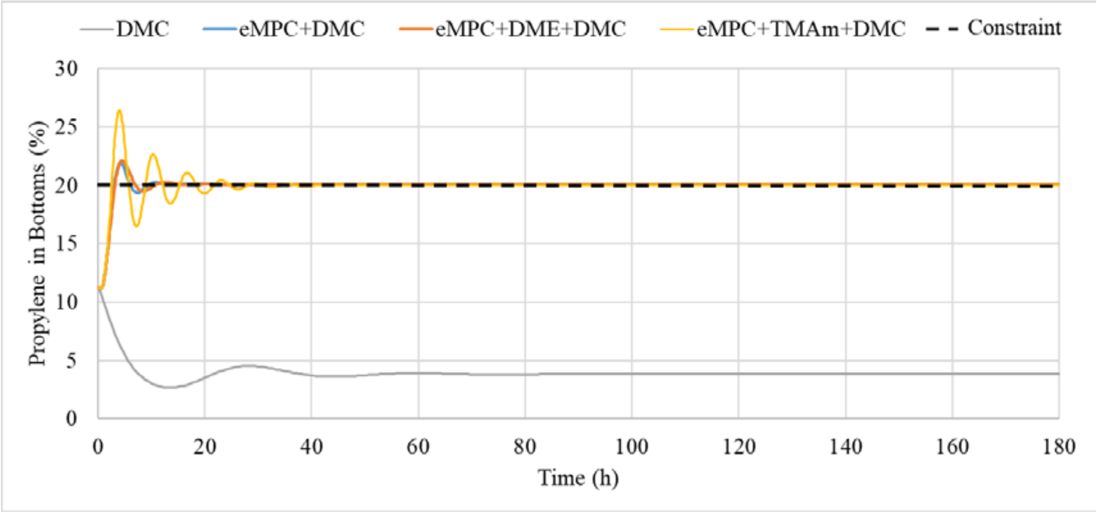


Figure 6.36: Propylene constraint from scenario 2 with different control architecture.

The profit at steady state is presented in Table 6.8. The profits of the eMPC have a larger increment compared to no optimization structure. In this case, the application of DME and $\mathrm{TMA}_{\mathrm{m}}$ algorithm do not increased the economic performance. However, the eMPC and DME maintain a smother control as expected by the objective function used.

TMA_m had a oscillating behavior compared to the others optimization structures, maybe because a miscalculation in the modifiers, probably the constant gain approximation is not valid for this operating range or the gain value is different.

Table 6.8. Profit at steady state for each architecture of scenario 2.

Control Architecture	**Profit at steady-state (€)**
Aspen DMC	12604.10
eMPC + Aspen DMC	12965.69
eMPC + DME + Aspen DMC	12965.70
eMPC + TMA_m + Aspen DMC	12965.70

Figure 6.37 and Figure 6.38 show the steam and distillate flow rates for each architecture. In the case of the distillate, the performance of DME is almost identical to that of the eMPC. In terms of control aspects, eMPC had a better performance with smoother movements than the other cases with eMPC. So, for this case, adding the MA modifiers using transient measurements does not improve the economic performance of the process.

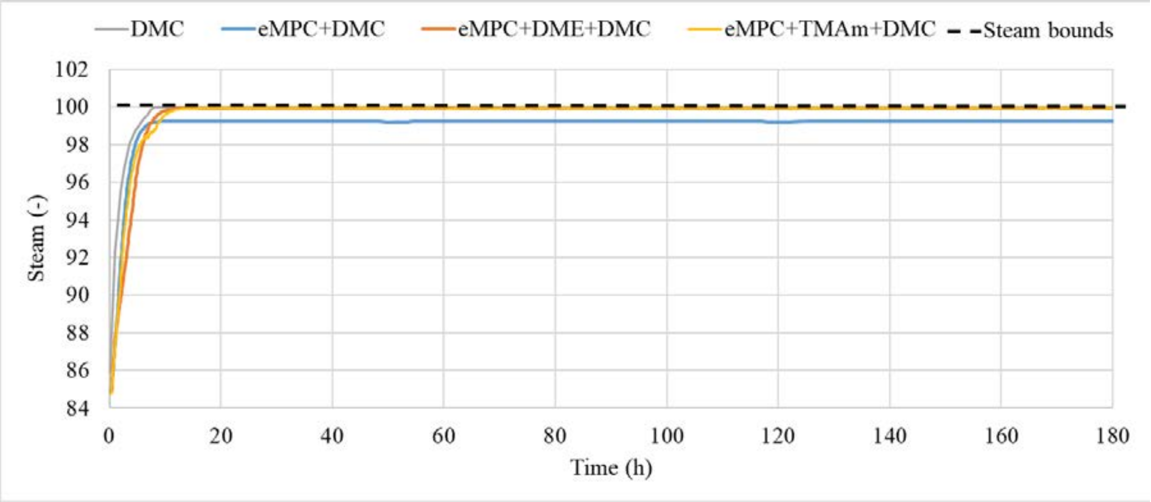


Figure 6.37: Steam flow from scenario 2 with different control architecture.

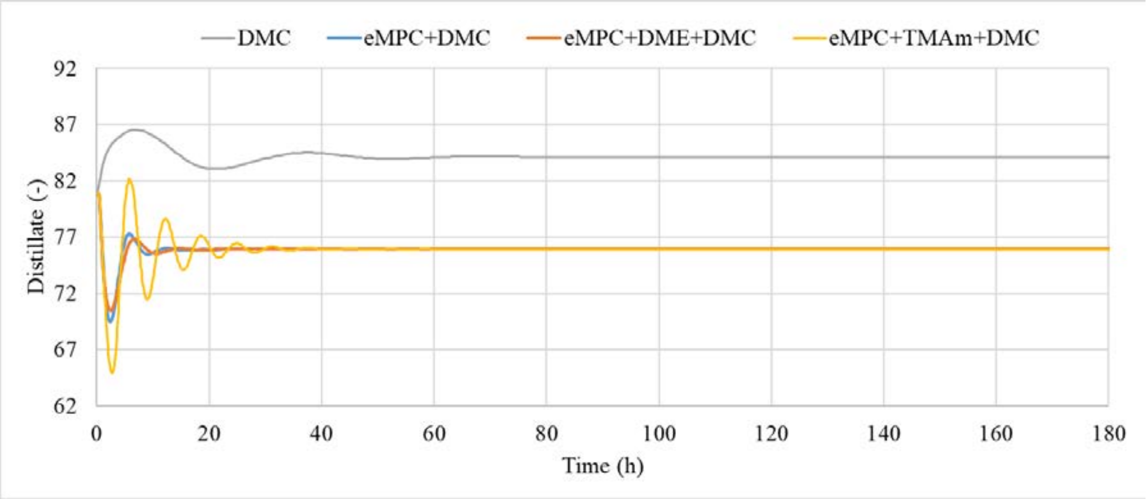


Figure 6.38: Distillate flow from scenario 2 with different control architecture.

6.4.3 Scenario 3

In this last scenario, the actual value of the process disturbances from historical data was used. The disturbances considered in the DMC model are the feed flow, the propylene concentration in the feed flow, the coolant temperature, and the steam pressure. Since these disturbances affect the values of the propylene concentration in the products and then the cost function and constraints, the MA modifiers related to these variables should also be included to have an accurate correction. Thus, the number of MA modifiers should increase from 3 to 7 when using the MAy modifiers. However, the gains related to the feed flow and concentration and the refrigerant are inferior to the gains related to the MVs. In fact, only the steam pressure gain is higher, but the price of steam does not interfere much with the cost function. For these reasons, and to simplify the calculations, the MA modifiers related to these disturbances have not been considered here. This means that the same optimization problem as in Scenario 1 was applied and only the simulation inputs were changed.

Table 6.9 shows the parameters used in the Scenario 3 simulation to calculate the modifiers and gradients for the DME and TMA_m algorithms, respectively.

Table 6.9. Parameters used DME and TMA$_m$ for scenario 3.

Parameters	Values
DME parameters	$n_{DME} = 3$ $\sigma = 10$ $Q = 0.1$ $K_{\gamma,\varepsilon} = 0.8$
TMA$_m$ parameters	$\mu = 1.5$ $K_{\gamma,\varepsilon} = 0.8$

The historical data from the real splitter were used as input information for the feed flow, feed composition of propylene, refrigerant temperature and pressure of the steam used in the reboiler are presented in Figure 6.39, Figure 6.40, Figure 6.41 and Figure 6.42, respectively. The data have been normalized for confidentiality reasons.

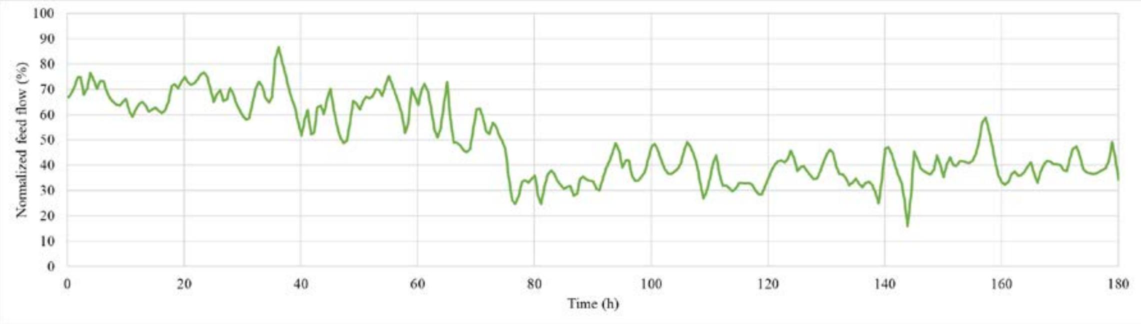


Figure 6.39: Normalized feed flow of scenario 3.

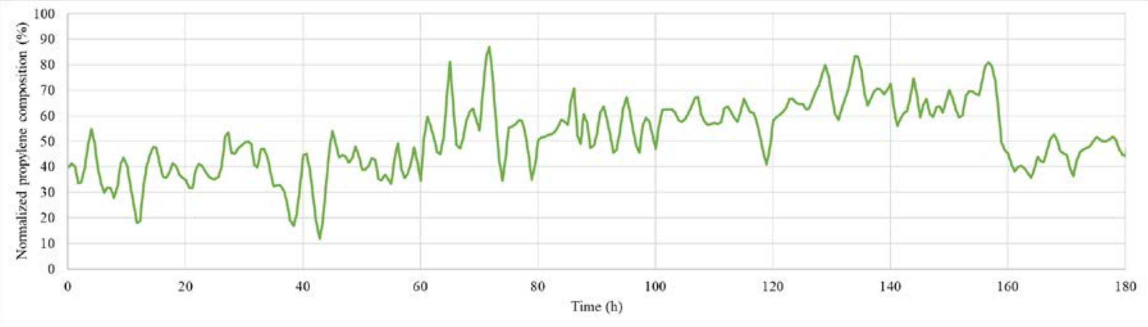


Figure 6.40: Normalized propylene composition of the feed of scenario 3.

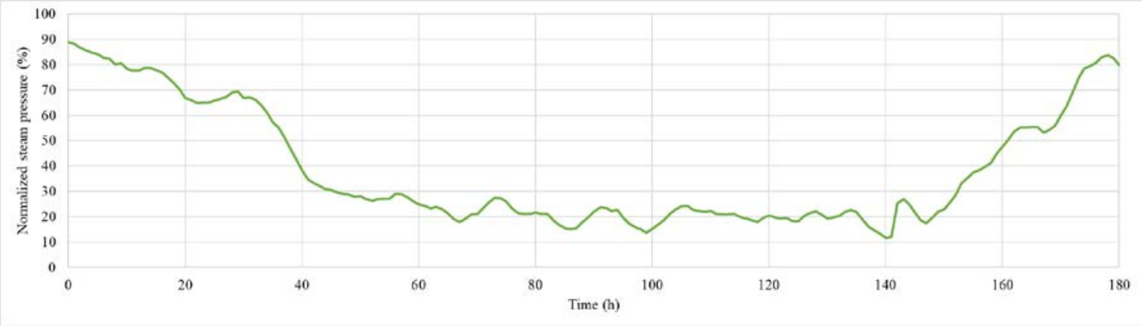


Figure 6.41: Normalized cooling water temperature of scenario 3.

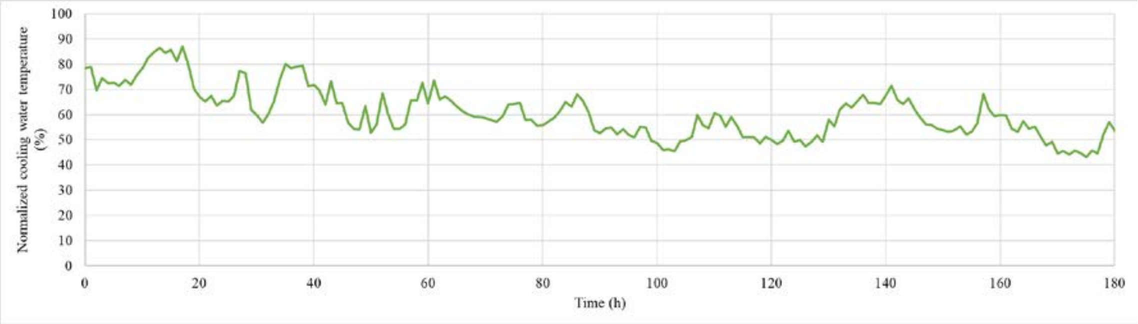


Figure 6.42: Normalized steam pressure for reboiler of scenario 3.

The Figure 6.43 shows the value of the cost function of the plant with the different architectures. The eMPC+DME+Aspen DMC architecture seems to maintain a higher value throughout the operation because the eMPC+DME compensates producing higher quality product when necessary, see Figure 6.44.

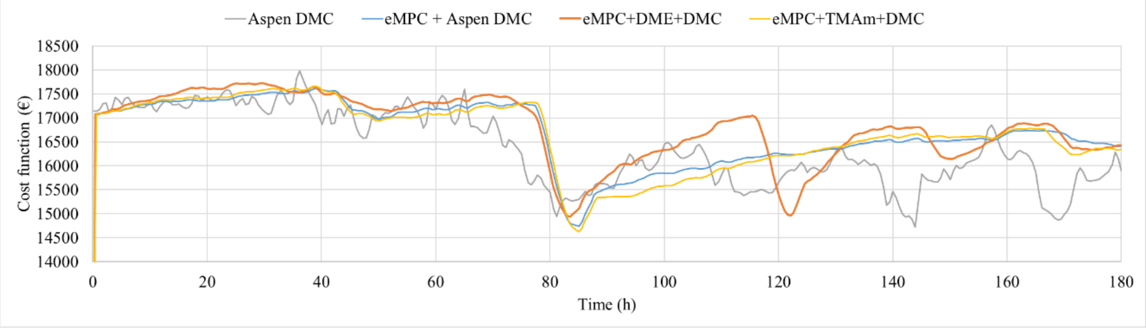


Figure 6.43: Cost function from scenario 3 with different control architecture.

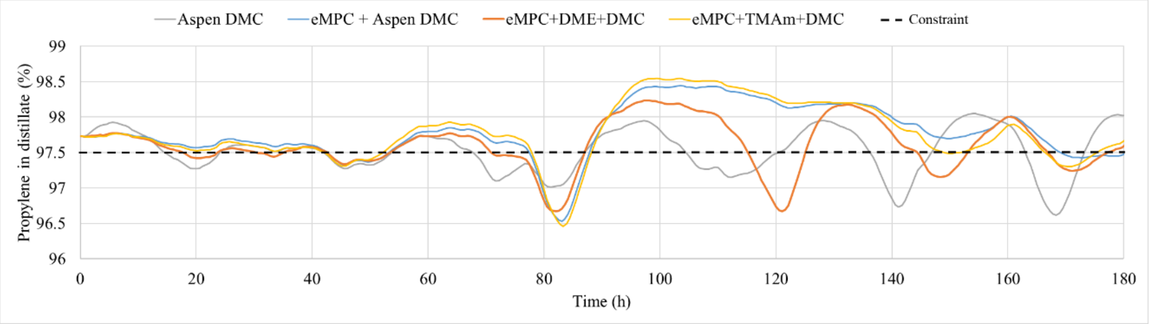


Figure 6.44: Propylene concentration from scenario 3 with different control architecture.

The total cost was also calculated for this scenario and it is shown in Table 6.10. The results show that the corrections made by the MA modifiers increase the overall profit compared to the system with DMC only. DME performed slightly better than the architecture without MA. However, TMA_m had less improvement compared to the system without MA modifiers. This is probably because the constant gain assumption is not valid in this scenario.

It is important to note that only 3 of the 7 modifiers were included in this simulation, so the profit probably could be higher if the disturbances modifiers were included. But DME still performed according to the cost function considered.

Table 6.10. Total profit for each architecture of scenario 3.

Control Architecture	**Total Profit (€)**	**Increment in profit (%)**
Aspen DMC	2.944×10^6	-
eMPC + Aspen DMC	3.002×10^6	1.984
eMPC + DME + Aspen DMC	3.024×10^6	<u>2.721</u>
eMPC + TMA_m + Aspen DMC	3.000×10^6	1.927

Figure 6.45 shows the steam used during the process. The scenarios with eMPC performed better than the case with DMC only. The DMC optimizer considers a very low LPcost for the steam and keeps the steam flow at maximum all the time, even though in some moments the value could be decreased to reduce the cost. The main objective of the DMC controller is not to increase the economic performance of the process, but to keep the plant close enough to the desired operating point chosen previously. DME had a different performance compared to the eMPC and eMPC+TMA architectures. For the TMA_m case, the behaviour is similar to the eMPC case without MA.

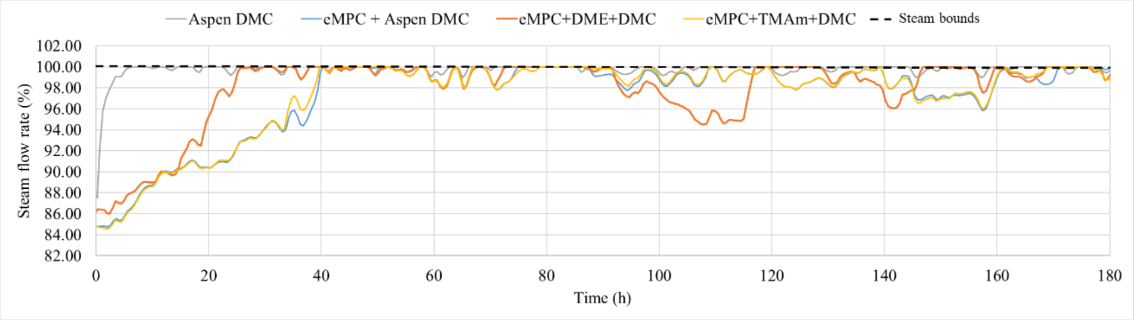


Figure 6.45: Steam flow from scenario 3 with different control architecture.

Figure 6.46 shows the value of distillate flow value for all architectures. The distillate flow rate is directly related to the quality constraint. Starting at time 70 hours in Figure 6.39, there is a decrease in the feed flow that provokes the decrease in the product quality. In the DMC architecture, this change provokes an immediate decrease in the distillate production, which reduces the profit, since DMC does not care about the economic behaviour of the process. On the other hand, eMPC tries to maximize the profit, so the decrease in the feed flow reduces the distillate, which later also reduces the quality constraint, but after some time eMPC produces a distillate with higher quality to compensate. The DME maintains the cost function in the highest value most of the time compared to the other structures. The TMA$_m$ results are similar to the case of eMPC without MA modifiers.

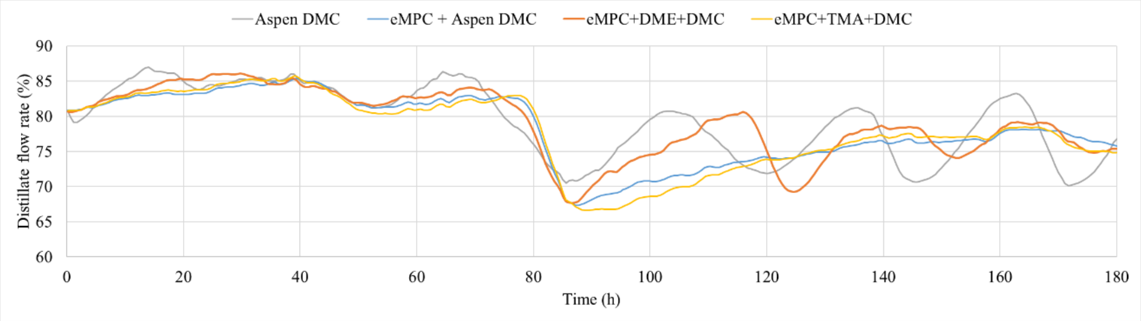


Figure 6.46: Distillate flow from scenario 3 with different control architecture.

6.4.4 Scenario 4

After a technical meeting with Petronor engineers, it was decided to change the object function from the previous scenario 3. The change is to consider the pure propylene flow rate instead of the distillate flow rate considered in problem (6.51). Therefore, the new objective function is as shown in (6.52). According to them, this new objective function could better represent the current contracts they now have with different customers.

$$\min_{F_D,F_W}\left(-\phi_{splitter,M}+\sigma_{u_1}\sum_{i=1}^{n_u-1}\left(\Delta F_{D,k+i}\right)^2+\sigma_{u_2}\sum_{i=1}^{n_u-1}\left(\Delta F_{W,k+i}\right)^2\right)$$

$$\phi_{splitter,M}=p_{C_3H_6}\overline{F_{C_3H_6}}+p_{C_3H_8}\overline{F_B}-p_W\overline{F_W} \qquad (6.52)$$

$$\overline{F_{C_3H_6}=\overline{x_d}_{M}F_D}$$

In this last scenario only Aspen DMC, eMPC without MA and eMPC+DME+DMC algorithm were applied. As the TMA algorithm did not show good results in scenario 3 this algorithm was not applied.

The Figure 6.47 shows the value of the cost function of the plant with the different architectures. It is not clear if the inclusion of the optimization layer improved the economic performance of the process. In Figure 6.48 the propylene concentration is presented.

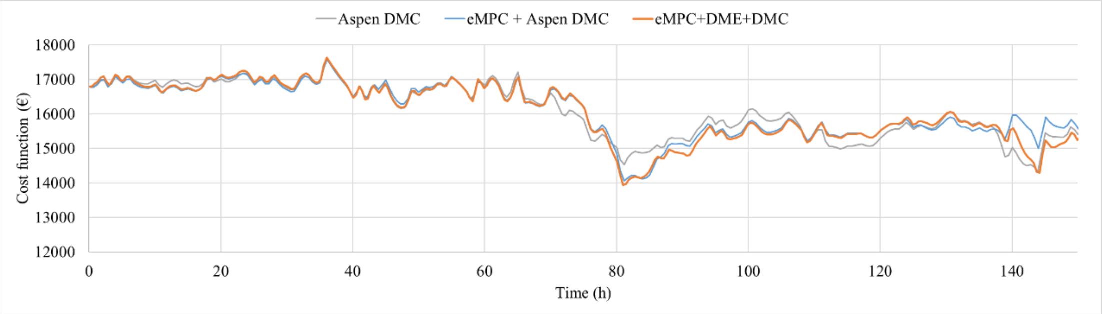


Figure 6.47: Cost function from scenario 4 with different control architecture.

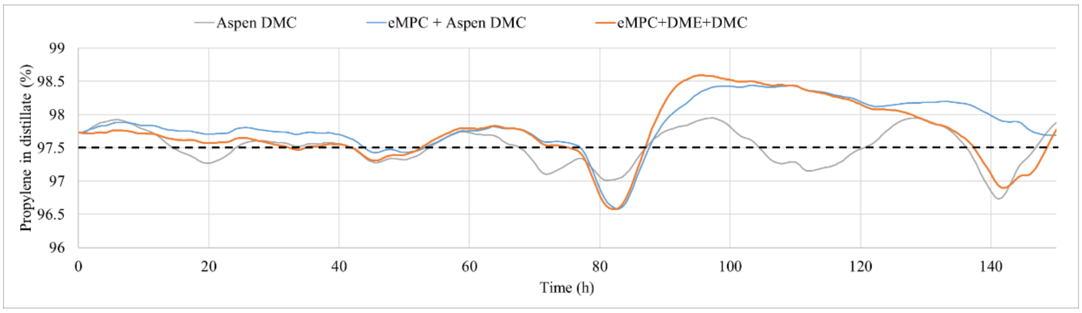


Figure 6.48: Propylene concentration from scenario 4 with different control architecture.

The total cost was also calculated for this scenario and it is shown in Table 6.11. The results show that the inclusion of a optimization layer slightly improved the economic performance of the process. However there is no significant differences between the application of MA in this case.

Table 6.11. Total profit for each architecture of scenario 4.

Control Architecture	Total Profit (€)	Increment in profit (%)
Aspen DMC	2.395×10^6	-
eMPC + Aspen DMC	2.414×10^6	0.800
eMPC + DME + Aspen DMC	2.413×10^6	0.761

Figure 6.49 and Figure 6.50 show the value of steam flowrate and distillate flow for all architectures, respectively. The case of eMPC without MA performed better in the control point of view comparing to the process without optimization and with optimization and MA.

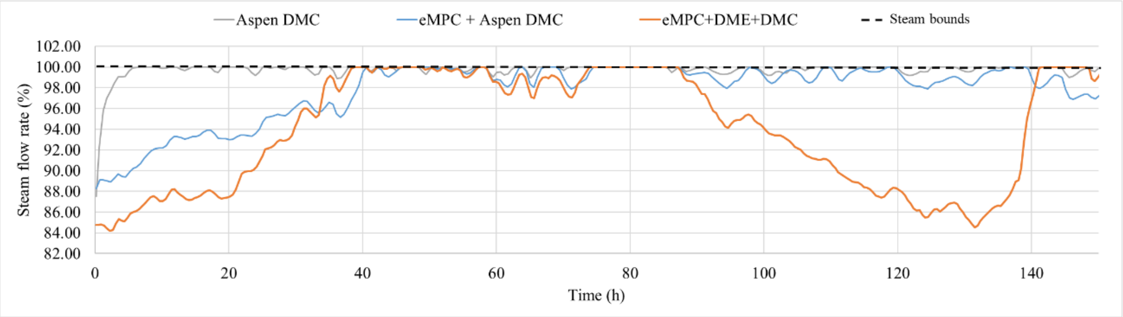


Figure 6.49: Steam flow from scenario 4 with different control architecture.

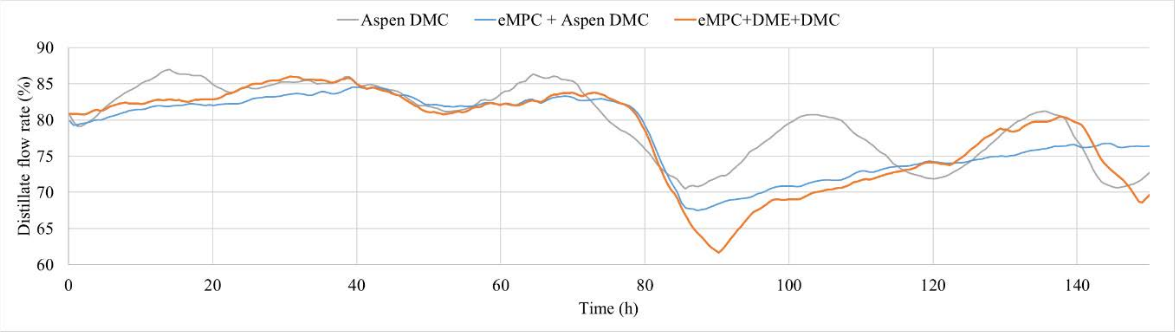


Figure 6.50: Distillate flow from scenario 4 with different control architecture.

6.5 Conclusions

In conclusion, the architecture of eMPC+MA+DMC controller with transient data can be easily implemented on top of the existing DMC layer since it was not necessary to develop a rigorous model for the optimization layer. It can also be concluded that the application of MA with transient data is valid for slow processes as the profit at steady state is achieved. Moreover, MA with transient data can also improve the economic performance of a process that suffers from constant disturbances and never reaches a steady state. Nevertheless, it could be important to consider the MA modifiers related to these disturbances to increase the accuracy of the algorithms. However, the applicability of MA with transient data depends on the process under consideration as the performance is largely affected by the accuracy of the gradients/modifiers calculation.

The main drawbacks for the application of DME are the tuning of the weighting parameters ($\sigma_{\phi}, \boldsymbol{Q}_{\theta}, \boldsymbol{\sigma}_{g}, \sigma_{\gamma}^{j}$). It is important to make a detailed study about the impact of these parameters on the performance of DME. Regarding TMA$_{\mathrm{m}}$, the algorithm still needs improvement, since in some cases it could not perform better than eMPC without MA.

7 Final Conclusions and Future Work

7.1 Final Conclusions

This thesis has presented contributions to the application of the MA methodology in slow dynamic processes by presenting a proposal to unify RTO and control layers. In addition, two algorithms were presented to estimate directly the MA modifiers and process gradients using transient data. The DME algorithm uses an optimization problem to estimate the first order modifiers. On the other hand, TMA tries to use a dynamic process gradient to enhance the economic performance of the process. This thesis also describes a methodology to find good initial values for the solution of complex DAE systems, which is usually not mentioned in the literature. The main contributions are listed below:

- A methodology for finding good initial values for a rigorous dynamic model of a propane-propylene splitter in order to achieve convergence. First, the model structure for the splitter is built with all the mass and heat balance equations for each tray, boiler and condenser, in addition to the equilibrium equations describing the molar fraction relationship between the liquid and vapor phases. The proposed methodology then constructs simplified models based on the full rigorous model, using assumptions that may not be appropriate for the process under study, but reduce the number of initial values required. In each step, the simplified models approximate the full rigorous model and a subset of initial values is found. The proposed methodology has been presented in: *“Oliveira-Silva, E., de Prada, C., "Methodology to achieve convergence in a rigorous dynamic model of a superfractionator", X Congress of Eurosim, Logroño, La Rioja, 2019.”*

- An integration of RTO layer with elements from MA and control layer. The proposed architecture includes a modified dynamic optimization, a module for estimating the model states, and an additional module for direct estimation of the modifiers during transients. This contribution was presented in the paper: *“Oliveira-Silva, E., de Prada, C., Navia, D., 2021. Dynamic optimization integrating modifier adaptation using transient measurements. Comput.*

Chem. Eng. 149, 107282." The module for direct estimation of modifiers is called Dynamic Modifier Estimation (DME) algorithm. The DME module aims to estimate the MA modifiers directly, without the need to explicitly calculate the process and model gradients. The DME uses an optimization problem that attempts to minimize the difference between the modified cost function and the constraints with the transient process measures using a moving horizon. The decision variables of the optimization problem are the dynamic modifiers that will match the static modifiers required by MA when the process reaches steady state.

- A second algorithm that use transient measurements to estimate the process gradient modifiers, TMA algorithm. TMA uses a second-order approximation with respect to the decision variables and a first-order one w.r.t. time and an estimation method to estimate a dynamic gradient. It is important to remark that the derivatives estimated are not the steady-state process gradients, but dynamic ones, that change at every time step. The derivatives describe the effect of a change in the MVs in the cost function from steady state at a certain time instant during the transient. However, the application of these values increased the economic performance of the case studies presented. The contribution was presented in congress and journal: *"Oliveira-Silva, E., de Prada, C., Navia, D. Economic MPC with Modifier Adaptation using Transient Measurements, ESCAPE-31, Istanbul 2021." and "Oliveira-Silva, E., de Prada, C., Montes, D., Navia, D., 2023. Economic MPC with Modifier Adaptation using Transient Measurements. Comput. Chem. Eng. 108205."* The last paper also presented an application of the method in a laboratory hybrid plant that mimics the behaviour of a CSTR with Van de Vusse reactions. The inclusion of the modifiers calculated with the TMA algorithm has significantly increased the process benefit for this experimental example.

- Development of a virtual platform to mimic a real industrial case to be used as a platform for RTO experiments. A virtual environment of an industrial splitter has been developed to test a dynamic real-time optimizer prior to on-site deployment. The architecture of the simulator represents the process using the EcosimPro© modelling and simulation software, and a real-time manager to maintain a controlled pace of simulation progress. The virtual plant has been integrated with the same commercial predictive controller (Aspen©DMC) as installed in the plant (even with the same configuration) and a dynamic real-time optimizer has been developed in Matlab© and implemented on top of it, allowing the virtual process to test the optimizer before deployment in the real factory. The data generated by the plant is stored using InfluxDB©, a time series data platform, for further analysis. This contribution was presented in the congress: *"Oliveira-Silva, E., de Prada, C., Navia, D., Simulation platform of an industrial propylene-propane splitter integrated to Advanced Process Control for Real Time Optimization experiments. Dycops 2022, Busan, 2022."*

- The application of MA using an industrial controller (Aspen DMC) in an example of industrial interest and the use of the existing models from the advanced control layer, to reduce the implementation effort in a real process. The propane-propylene case study presented in Chapter 6 uses the linear dynamic model already developed for the DMC controller to optimize its operation. Traditional RTO requires a rigorous model of the process, but the application of MA required only the implementation of the dRTO optimization problem and the DME/TMA algorithm to estimate the MA modifiers. As a result, the effort associated with maintaining a rigorous model are eliminated.

7.2 Future work

In the future, we would like to continue research in the areas covered in this thesis. Among them, we would like to investigate the following:

- The proposed algorithms for estimating the process gradients using transient data, TMA and TMA$_m$, need further improvement.
- Consider different structures for the eMPC economic objective function as the inclusion of an economic optimization during the transient and not only in steady state.
- The application of the MA algorithms proposed in this thesis to biological processes. Biological processes are complex, non-linear and non-stationary, which makes modelling and parameter estimation a challenging task

(Rofifah, 2020). Since MA is able to find the true optimum with bad models, it could be an interesting solution to optimize biological processes.

- Applying DME in batch processes. The DME algorithm was able to satisfy the constraints in all of the scenarios studied in Chapter 6. Thus, it seems that it could work well in batch processes subject to path constraints. Path constraints limit the inputs and outputs during the batch.

References

Aspen Technology Inc, 2021. Aspen DMC3 [WWW Document]. URL https://www.aspentech.com/en/products/msc/aspen-dmc3 (accessed 11.22.21).

Bank, R., Graham, R.L., Stoer, J., Varga, R., Yserentant, H., 1993. Solving Ordinary Differential Equations I, Math2, Springer Series in Computational Mathematics. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-78862-1

Bergh, L.G., Yianatos, J.B., 2014. Engineers training in automation of flotation processes, IFAC Proceedings Volumes (IFAC-PapersOnline). IFAC. https://doi.org/10.3182/20140824-6-za-1003.01536

Bhat, S.A., Saraf, D.N., 2004. Steady-state identification, gross error detection, and data reconciliation for industrial process units. Ind. Eng. Chem. Res. 43, 4323–4336. https://doi.org/10.1021/ie030563u

BrdyŚ, M., Chen, S., Roberts, P.D., 1986. An extension to the modified two-step algorithm for steady-state system optimization and parameter estimation. Int. J. Syst. Sci. 17, 1229–1243. https://doi.org/10.1080/00207728608926883

Brdyś, M.A., Tatjewski, P., 1994. An Algorithm for Steady-State Optimizing Dual Control of Uncertain Plants. IFAC Proc. Vol. 27, 215–220. https://doi.org/10.1016/S1474-6670(17)47650-6

Camacho, E.F., Bordons, C., 1999. Model Predictive control. Springer London, London.

Câmara, M., Quelhas, A., Pinto, J., 2016. Performance Evaluation of Real Industrial RTO Systems. Processes 4, 44. https://doi.org/10.3390/pr4040044

Chachuat, B., Srinivasan, B., Bonvin, D., 2009. Adaptation strategies for real-time optimization. Comput. Chem. Eng. 33, 1557–1567. https://doi.org/10.1016/j.compchemeng.2009.04.014

D. Quelhas, A., Castro de Jesus, N.J., Pinto, J.C., 2013. COMMON VULNERABILITIES OF RTO IMPLEMENTATIONS IN REAL CHEMICAL PROCESSES. Can. J. Chem. Eng. 91, 652–668. https://doi.org/10.1002/cjce.21738

Darby, M.L., Nikolaou, M., Jones, J., Nicholson, D., 2011. RTO: An overview and assessment of current practice. J. Process Control 21, 874–884. https://doi.org/10.1016/j.jprocont.2011.03.009

de Prada, C., Sarabia, D., Gutierrez, G., Gomez, E., Marmol, S., Sola, M., Pascual, C., Gonzalez, R., 2017. Integration of RTO and MPC in the hydrogen network of a petrol refinery. Processes 5, 1–20. https://doi.org/10.3390/pr5010003

EA Int., 2020. EcosimPro 6.2 [WWW Document]. User Man. URL https://www.ecosimpro.com/

Ellis, M., Durand, H., Christofides, P.D., 2014. A tutorial review of economic model predictive control methods. J. Process Control 24, 1156–1178. https://doi.org/10.1016/j.jprocont.2014.03.010

Fahim, M. A.; Al-Sahhaf, T. A.; Elkilani, A.S., 2010. Fundamentals of Petroleum Refining, First. ed. Elsevier, Oxford.

Faulwasser, T., Pannocchia, G., 2019. Toward a Unifying Framework Blending Real-Time Optimization and Economic Model Predictive Control. Ind. Eng. Chem. Res. 58, 13583–13598. https://doi.org/10.1021/acs.iecr.9b00782

François, G., Bonvin, D., 2014. Use of Transient Measurements for the Optimization of Steady-

State Performance via Modifier Adaptation. Ind. Eng. Chem. Res. 53, 5148–5159. https://doi.org/10.1021/ie401392s

François, G., Bonvin, D., 2013. Use of Convex Model Approximations for Real-Time Optimization via Modifier Adaptation. Ind. Eng. Chem. Res. 52, 11614–11625. https://doi.org/10.1021/ie3032372

Gao, W., Engell, S., 2005. Iterative set-point optimization of batch chromatography. Comput. Chem. Eng. 29, 1401–1409. https://doi.org/10.1016/j.compchemeng.2005.02.035

Gill, P.E., Murray, W., Saunders, M.A., 2005. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. SIAM Rev. 47, 99–131. https://doi.org/10.1137/S0036144504446096

Goodwin, G.C., Sin, K.S., 1984. Adaptive filtering prediction and control, Dover Publications. Englewood Cliffs, New York.

Hart, W.E., Laird, C.D., Watson, J.-P., Woodruff, D.L., Hackebeil, G.A., Nicholson, B.L., Siirola, J.D., 2017. Pyomo — Optimization Modeling in Python, Springer Optimization and Its Applications. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-58821-6

Hart, W.E., Watson, J.P., Woodruff, D.L., 2011. Pyomo: Modeling and solving mathematical programs in Python. Math. Program. Comput. 3, 219–260. https://doi.org/10.1007/s12532-011-0026-8

Hemptinne, J.-C. de, Ledanois, J.-M., Mougin, P., Barreau, A., 2012. Select Thermodynamic Models for Process Simulation : A Pratical Guide using a Three Steps Methodology. Technip.

Hernández, R., Engell, S., 2019. Economics optimizing control with model mismatch based on modifier adaptation. IFAC-PapersOnLine 52, 46–51. https://doi.org/10.1016/j.ifacol.2019.06.035

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S., 2005. SUNDIALS. ACM Trans. Math. Softw. 31, 363–396. https://doi.org/10.1145/1089014.1089020

Huang, R., Patwardhan, S.C., Biegler, L.T., 2010. Offset-free Nonlinear Model Predictive Control Based on Moving Horizon Estimation for an Air Separation Unit, IFAC Proceedings Volumes. IFAC. https://doi.org/10.3182/20100705-3-be-2011.00105

Influxdb, 2023. InfluxData Inc [WWW Document]. URL https://www.influxdata.com/ (accessed 3.8.23).

Isermann, R., Münchhof, M., 2011. Identification of Dynamic Systems: An Introduction with Applications, Identification of Dynamic Systems. https://doi.org/10.1007/978-3-540-78879-9_9

Krishnamoorthy, D., Skogestad, S., 2022. Real-Time optimization as a feedback control problem – A review. Comput. Chem. Eng. 161, 107723. https://doi.org/10.1016/j.compchemeng.2022.107723

Luyben, W.L., 2017. Dynamic simulation of flooded condensers. Chem. Eng. Res. Des. 118, 12–20. https://doi.org/10.1016/j.cherd.2016.12.001

Luyben, W.L., 2006. Practical distillation control. https://doi.org/10.1049/ic:20000114

Luyben, W.L., 1999. Process Modeling , Simulation and Control Chemical Engineers. Pet. Refin. Eng.

Mansour, M., Ellis, J.E., 2003. Comparison of methods for estimating real process derivatives in on-line optimization. Appl. Math. Model. 27, 275–291. https://doi.org/10.1016/S0307-904X(02)00124-5

Marchetti, A., Chachuat, B., Bonvin, D., 2009. Modifier-Adaptation Methodology for Real-Time Optimization. Ind. Eng. Chem. Res. 48, 6022–6033. https://doi.org/10.1021/ie801352x

Marchetti, A., François, G., Faulwasser, T., Bonvin, D., 2016. Modifier Adaptation for Real-Time Optimization—Methods and Applications. Processes 4, 55. https://doi.org/10.3390/pr4040055

Matsoukas, T., 2013. Fundamentals of Chemical Engineering Thermodynamics: With Applications to Chemical Processes 690. https://doi.org/10.1111/j.1475-6803.2004.00096.x

Montes, D., Zamarreño, J.M., Pitarch, J.L., Oliveira da Silva, E., Prada, C. De, 2021. Implementación de capas superiores de la pirámide de automatización en una planta piloto híbrida, in: XLII JORNADAS DE AUTOMÁTICA : LIBRO DE ACTAS. Servizo de Publicacións da UDC, pp. 403–410. https://doi.org/10.17979/spudc.9788497498043.403

Navia, D., Puen, A., Quintanilla, P., Briceño, L., Bergh, L., 2019. On dealing with measured disturbances in the modifier adaptation method for real-time optimization. Comput. Chem. Eng. 128, 141–163. https://doi.org/10.1016/j.compchemeng.2019.06.004

Navia, D., Villegas, D., Cornejo, I., de Prada, C., 2016. Real-time optimization for a laboratory-scale flotation column. Comput. Chem. Eng. 86, 62–74. https://doi.org/10.1016/j.compchemeng.2015.12.006

Nordsieck, A., 1962. On Numerical Integration of Ordinary Differential Equations. Math. Comput. 16, 22–49.

Pannocchia, G., 2018. An economic MPC formulation with offset-free asymptotic performance. IFAC-PapersOnLine 51, 393–398. https://doi.org/10.1016/j.ifacol.2018.09.332

Papasavvas, A., de Avila Ferreira, T., Marchetti, A.G., Bonvin, D., 2019. Analysis of output modifier adaptation for real-time optimization. Comput. Chem. Eng. 121, 285–293. https://doi.org/10.1016/j.compchemeng.2018.09.028

Rawlings, J.B., Mayne, D.Q., Diehl, M.M., 2019. Model predictive control:Theory, Computation, and Design, 2nd ed, Studies in Systems, Decision and Control. Nob Hill Publishing, Santa Barbara.

Richalet, J., 1991. Pratique de l'identification. Editions Hermes, Paris.

Roberts, P.D., 1995. Coping with model-reality differences in industrial process optimisation - A review of integrated system optimisation and parameter estimation (isope). Comput. Ind. 26, 281–290. https://doi.org/10.1016/0166-3615(95)00011-R

Roberts, P.D., 1979. An algorithm for steady-state system optimization and parameter estimation. Int. J. Syst. Sci. 10, 719–734. https://doi.org/10.1080/00207727908941614

Roberts, P.D., Williams, T.W.C., 1981. On an algorithm for combined system optimisation and parameter estimation. Automatica 17, 199–209. https://doi.org/10.1016/0005-1098(81)90095-9

Rodríguez-Blanco, T., Sarabia, D., Pitarch, J.L., de Prada, C., 2017. Modifier Adaptation methodology based on transient and static measurements for RTO to cope with structural uncertainty. Comput. Chem. Eng. 106, 480–500. https://doi.org/10.1016/j.compchemeng.2017.07.001

Rofifah, D., 2020. Control in Bioprocessing_ Modeling, Estimation and the Use of Soft Sensors, Paper Knowledge . Toward a Media History of Documents.

Shah, G., Engell, S., 2011. Tuning MPC for desired closed-loop performance for MIMO systems, in: Proceedings of the 2011 American Control Conference. IEEE, pp. 4404–4409. https://doi.org/10.1109/ACC.2011.5991581

Shridhar, R., Cooper, D.J., 1998. A tuning strategy for unconstrained multivariable model predictive control. Ind. Eng. Chem. Res. 37, 4003–4016. https://doi.org/10.1021/ie980202s

Simulis Thermodynamics [WWW Document], 2021. . Prosim. URL https://www.prosim.net/ (accessed 11.22.21).

Skogestad, S., 2000. Self-optimizing control: The missing link between steady-state optimization and control. Comput. Chem. Eng. 24, 569–575. https://doi.org/10.1016/s0098-1354(00)00405-1

Tatjewski, P., 2002. Iterative optimizing set-point control - The basic principle redesigned, IFAC Proceedings Volumes (IFAC-PapersOnline). IFAC. https://doi.org/10.3182/20020721-6-es-1901.00994

Vaccari, M., Bonvin, D., Pelagagge, F., Pannocchia, G., 2021. Offset-Free Economic MPC Based on Modifier Adaptation : Investigation of Several Gradient-Estimation Techniques 1–24.

Vaccari, M., Pannocchia, G., 2018. Implementation of an economic MPC with robustly optimal steady-state behavior. IFAC-PapersOnLine 51, 92–97. https://doi.org/10.1016/j.ifacol.2018.10.180

Vaccari, M., Pannocchia, G., 2016. A Modifier-Adaptation Strategy towards Offset-Free Economic MPC. Processes 5, 2. https://doi.org/10.3390/pr5010002

Vaccari, M., Pelagagge, F., Bonvin, D., Pannocchia, G., 2020. Estimation technique for offset-free economic MPC based on modifier adaptation. IFAC-PapersOnLine 53, 11251–11256. https://doi.org/10.1016/j.ifacol.2020.12.357

Van de Vusse, J.G., 1964. Plug-flow type reactor versus tank reactor. Chem. Eng. Sci. 19, 994–996. https://doi.org/10.1016/0009-2509(64)85109-5

Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106, 25–57. https://doi.org/10.1007/s10107-004-0559-y

Wauquier, J.-P., 1995. Petroleum Refining, Volume 1 - Crude Oil, Petroleum Products, Process Flowsheets, Technip. ed, Institut Français du Pétrole. Paris.

Williams, T.J., Otto, R.E., 1960. A generalized chemical processing model for the investigation of computer control. Trans. Am. Inst. Electr. Eng. Part I Commun. Electron. 79, 458–473. https://doi.org/10.1109/tce.1960.6367296

Yip, W.S., Marlin, T.E., 2004. The effect of model fidelity on real-time optimization performance. Comput. Chem. Eng. 28, 267–280. https://doi.org/10.1016/S0098-1354(03)00164-9