ON THE USE OF TRANSIENT PROCESS DATA FOR REAL TIME OPTIMIZATION

Rodrigo Curvelo Santos

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Química, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Química.

Orientadores: Argimiro Resende Secchi
Maurício Bezerra de Souza Júnior

Rio de Janeiro
Fevereiro de 2020

ON THE USE OF TRANSIENT PROCESS DATA FOR REAL TIME OPTIMIZATION

Rodrigo Curvelo Santos

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA QUÍMICA.

Examinada por:

_____
Prof. Argimiro Resende Secchi, D.Sc.

_____
Prof. Maurício Bezerra de Souza Júnior, D.Sc.

_____
Prof. Alessandro Jacoud Peixoto , D.Sc.

_____
Prof. Lizandro de Sousa Santos, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2020

Curvelo Santos, Rodrigo

On the use of transient process data for real time optimization/Rodrigo Curvelo Santos. – Rio de Janeiro: UFRJ/COPPE, 2020.

XXII, 98 p.: il.; 29,7cm.

Orientadores: Argimiro Resende Secchi

Maurício Bezerra de Souza Júnior

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Química, 2020.

Referências Bibliográficas: p. 93 – 98.

1. Otimização. 2. RTO. 3. DRTO. 4. Van de Vusse. I. Secchi, Argimiro Resende *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Química. III. Título.

Agradecimentos

Ao professor Argimiro, por ser o maior exemplo que tive ao longo da minha trajetória acadêmica, um exemplo de conhecimento, didática e organização. Obrigado pela paciência e por todo o sacrifício e tempo expendidos para a manutenção da qualidade do PEQ.

Ao professor Maurício, por ser um excelente professor e pela proposta do tema abordado. Serei eternamente grato pela sua simpatia e otimismo na realização deste trabalho, assim como por todo o conhecimento absorvido nas diversas reuniões que tivemos ainda na disciplina de Controle.

Aos amigos do PEQ. Especialmente aos que tive mais contato, Lucas, Ana e Pedro. A este por ter disposto tanto tempo a me ajudar neste trabalho. O quanto aprendi conversando contigo não pode ser posto em palavras, muito obrigado de verdade; Lucas por ser literalmente incrível, como profissional e como pessoa e Ana por ser tão amável e companheira. Obrigado a vocês dois por cederem seu sofá, por abrirem minha mente para novas culturas e, principalmente, por fazerem parte de uma das melhores épocas da minha vida. Aos amigos que tive a oprtunidade de conhecer mais a fundo na Noruega, Allyne e Otávio, por serem um ponto familiar em um lugar tão longe de casa. Vocês são pessoas maravilhosas.

À minha famìlia, por todo o suporte educacional ao longo da minha vida e por me apoiarem no caminho que escolhi trilhar.

Aos meus amigos da graduação da EQ, que ajudaram a amenizar o constante estresse da faculdade e me fizeram sair da minha zona de conforto.

Ao meu melhor amigo e irmão de consideração, Vinícius, o qual chega a ser difícil de agradecer por tanto que já fez por mim. Muito obrigado, meu mano, sem você eu não seria um Engenheiro e muito menos a pessoa que sou hoje.

Ao meu amor e parceira para toda a vida, Bianca, pelo suporte emocional, por me socorrer nos momentos de desespero, por não me fazer desistir nunca e principalmente, por me mostrar o que é ser forte de verdade. Você é minha inspiração diária.

Á UFRJ e todos os funcionários.

Ao CNPq e CAPES pelo financiamento público da pesquisa brasileira.

Acknowledgments

I am also truly grateful to the Process System Engineering group for the wonderfull time I spent at NTNU. Special thanks to Professor Sigurd Skogestad and Dinesh Krishnamoorthy, for being my first international whole models in the field. It is impossible to quantify what I've learned and experienced during my time as an exchange stundent at NTNU.

I am thankful to The International Partnerships for Excellence Education, Research and Innovation (INTPART), for funding part of this work and my stay in Trondheim.

"The pursuit of knowledge is a journey with no destination
As long as people live, there is always something to learn
And so we learn, we record, we speak
Points connect, thus forming our history, our agency, our identity
This world we live in is not for, or of, any one individual - and neither is knowledge
And so I write and I speak of all that I learned
To plant seeds of knowledge that will be harvested by those who follow me
To you who have found my writings, read well these words
That they help you build a bridge to a brighter future"

"Knowledge is Light" - Susanna Grotoff (*"Octopath Traveler"*)

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

INVESTIGAÇÃO DO USO DE INFORMAÇÃO TRANSIENTE DE PROCESSO PARA A OTIMIZAÇÃO EM TEMPO REAL

Rodrigo Curvelo Santos

Fevereiro/2020

Orientadores: Argimiro Resende Secchi
Maurício Bezerra de Souza Júnior

Programa: Engenharia Química

Na literatura há alguns trabalhos recentes (KRISHNAMOORTHY *et al.*, 2018; MATIAS and LE ROUX, 2018) que propõe uma abordagem RTO híbrida que combina o melhor de ambas as metodologias tradicional e dinâmica, ou seja, a etapa de estimação de parâmetros ocorre fazendo uso de modelos dinâmicos e medidas transientes enquanto que a etapa de otimização é realizada através de modelos estacionários. KRISHNAMOORTHY *et al.* (2018) mostraram em seu trabalho que esta abordagem, quando aplicada a um sistema de produção de óleo *gas-lift*, fornece resultados de performance similares ao RTO dinâmico, com tempos computacionais equivalentes ao RTO tradicional. Entretanto, investigações sistemáticas a respeito da utilização desta metodologia híbrida para diferentes classes de modelos de processos dinâmicos ainda são ausentes na literatura. O objetivo central deste trabalho é avaliar a abordagem citada aplicando em um maior número de classes de sistemas dinâmicos onde comportamentos complexos podem ser observados (i.e. regiões de fase não-mínima, overshoots, resposta inversa, etc). O estudo de caso escolhido foi o CSTR de Van de Vusse (VAN DE VUSSE, 1962), usado como benchmark para algoritmos de controle e otimização. Os resultados deste trabalho mostraram que estas abordagens divergem substancialmente em operação transiente, o que é realçado pelo alto grau de não-linearidade do sistema em questão e, principalmente, que a abordagem híbrida apresenteu melhor performance computacional e em termos da função objetivo.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ON THE USE OF TRANSIENT PROCESS DATA FOR REAL TIME OPTIMIZATION

Rodrigo Curvelo Santos

February/2020

Advisors: Argimiro Resende Secchi
Maurício Bezerra de Souza Júnior

Department: Chemical Engineering

In the literature, there are some recent works (KRISHNAMOORTHY *et al.*, 2018; MATIAS and LE ROUX, 2018) that propose a hybrid RTO approach that couples the best of both traditional and dynamic RTO (i.e., the parameter estimation step proceeds using dynamic models and transient data while the optimization step is done using stationary models). KRISHNAMOORTHY *et al.* (2018) showed that this methodology, when applied to a gas-lift system, provides similar performance results in comparison to dynamic RTO, spending similar computation time to traditional RTO. However, systematic investigations on the use of this hybrid RTO approach for different classes of dynamic process models still lack in the literature. The main goal of this work is to evaluate this hybrid RTO methodology by applying it to more classes of dynamic systems where complex phenomena can occur (i.e. overshoots, inverse responses and so on). The chosen case study was the Van de Vusse CSTR (VAN DE VUSSE, 1962), a benchmark system for control and optimization algorithms. The results show that these RTO approaches mainly diverge during transient operation, fact that is highlighted by the high nonlinearity of this system and, the most important, that the hybrid approach has better performance, coth computationally and cost function wise.

Contents

List of Figures

List of Tables

List of Symbols

A	discrete-time linear system state matrix
A_c	continuous-time linear system state matrix
A_R	heat exchange area of the cooling jacket, m^2
B	discrete-time linear system input matrix
B_c	continuous-time linear system input matrix
C	state-space model output matrix
E	expected value
C_A	concentration of component A, mol/L
$C_{A,in}$	reactor inlet concentration of component A, mol/L
C_B	concentration of component B, mol/L
c_p	heat capacity of the reaction mixture, kJ/kgK
d	dynamic system disturbance vector
E_i	activation energy for reaction i divided by the universal gas constant, K
f	discrete-time dynamic system state equations
F	reactor inlet flow, mol/h
F_{obj}	objective function

F_k model derivative matrix in respect of the states

g_{SS} discrete-time stationary system equations

h discrete-time dynamic system output equations

H_k output model derivative in respect of the states

I identity matrix

k_i reaction rate constant for reaction i

k_{0i} frequency factor of Arrhenius Law for reaction i

K_w heat transfer coefficient of the cooling jacket, kJ/hKm^2

K_k Kalman gain

L_k model derivative matrix in respect of the process noise

M control horizon

M_k output model derivative in respect of the measurement noise

N prediction horizon

p dynamic system parameter vector

P_k covarianve matrix of the estimation error

Q setpoint weight matrix

Q_k process noise covariance matrix

R_k measurement noise covariance matrix

S target weight matrix

T_{in} reactor inlet temperature, $^\circ C$

T_k	cooling jacket temperature, $^\circ C$
u	dynamic system input vector
u^*	dynamic system optimal input vector
u^{target}	input target vector
U^*	optimal set of control actions
v	measurements Gaussian white noise
V	reactor volume, L
w	process Gaussian white noise
W	control action size weight matrix
x	dynamic system state vector
$\hat{x}$	dynamic system estimated state vector
$\hat{x}^+$	*posteriori* estimate of the state vector
$\hat{x}^-$	*priori* estimate of the state vector
y	dynamic system output vector
y_{meas}	dynamic system measured output vector
y^{SP}	output setpoint vector
ΔH_i	molar enthalpy for reaction i, kJ/mol
ρ	specific mass of the reaction mixture, kg/L
θ	dynamic system parameter-disturbance vector
$\hat{\theta}^*$	dynamic system optimal estimated parameter-disturbance vector

List of Abbreviations

CasADi	open-source tool for nonlinear optimization and algorithmic differentiation
CEKF	constrained Kalman filter
CSTR	continuous stirred tank reactor
DRTO	dynamic real time optimization
EKF	Extended Kalman filter
EMPC	economic model predictive control
HRTO	hybrid real time optimization
HRO	heuristic random optimization
LP	linear programming
MA	modifier adaptation
MHE	moving horizon estimation
MPC	model predictive control
NCO	necessary conditions for optimality
NLP	nonlinear programming
NMPC	nonlinear model predictive control
QP	quadratic programming

RTO real time optimization

SQP sequential quadratic programming

SOC self-optimizing control

Chapter 1

Introduction

This chapter is focused on exposing a scope of the current practice of real-time optimization as well as challenges associated with the successful implementation of this type of structure. At the end, the motivation, goals and structure of the dissertation are introduced.

1.1 Optimization of industrial processes

Companies from several sectors of the economy are constantly aiming to produce more while spending less resources. Thus, for a company to remain competitive, process optimization has become a necessity over the years and it is no longer just an improvement in efficiency, but a critical factor for business competitiveness. Therefore, constant studies and refinement of process optimization and control techniques became more relevant than ever. Proper use of such techniques allows an industrial plant to operate with its design variables closer to its operational limits, while maintaining plant integrity and technical demands of the products.

Objectively, process optimization directly enables cost reductions, end-product quality improvements, safe and in-plant operation environmental limits. Some of the the main optimization techniques are: *real time optimization* (RTO), *NCO tracking* and *self-optimizing control* (SOC), with RTO approaches to be focused on this work. RTO Algorithms have a large participation in the industry, being employed in many fields of the industry, especially in petrochemical processes (MERCANGÖZ and DOYLE III, 2008; ROTAVA and ZANIN, 2005; RUIZ, 2009; YOUNG, 2006).

It is well known that industrial processes usually consist of different pieces of equipment and operations, resulting in an often high number of decision variables. Optimal operation requires the process to achieve goals on different

timescales, from relatively slow actions such as planning to corrective actions that need to be fast. Therefore, from years ago to the present day, process operation is usually sectioned into several decision making layers that obey a certain hierarchy, each with its own time horizon (DARBY *et al.*, 2011; SKOGESTAD and POSTLETHWAITE, 2007), as shown in Figure 1.1. Process information flows with the upper layers providing setpoints to the lower layers (KRISHNAMOORTHY, 2019). These layers are succinctly described as follows:

- Asset management: refers to governance and mostly long-term decisions, such as investments in infrastructure, human resources and intellectual properties;
- Plant-wide scheduling: consists of arranging, controlling and optimizing work and workloads in a process;
- Real time optimization: this layer's goal is to maximize/minimize a specific objective, usually related to revenue/operational costs;
- Supervisory control: receives the operational setpoint from the RTO layer and aims to calculate optimal control actions with some optimization capability. Commonly played multivariable control methods with *quadratic programming* (QP) such as *model predictive control* (MPC);
- Regulatory control: responsible for maintaining the plant stable and consists typically of *proportional-integral-derivative* controllers (PID).

This work will focus more deeply into the RTO and supervisory control layers. In the 1960s, the first optimizing solutions applying model-based predictive control were reported in the literature (KALMAN *et al.*, 1960), although only in the 1980s the first real-time static optimization solutions came to fruition (CHEN and JOSEPH, 1987).

1.2 Motivation

Despite the well-known potential of applying real-time optimization algorithms, their implementation still represents a major challenge in many situations. Firstly, RTO is not responsible for manipulating any process variable. It only determines the setpoint that corresponds to the optimality state of the system and which is forwarded to the advanced control layers. Thus, it is possible that certain setpoints are not employed because there is incompatibility between the RTO and MPC internal models. Similarly, the internal objective function of the

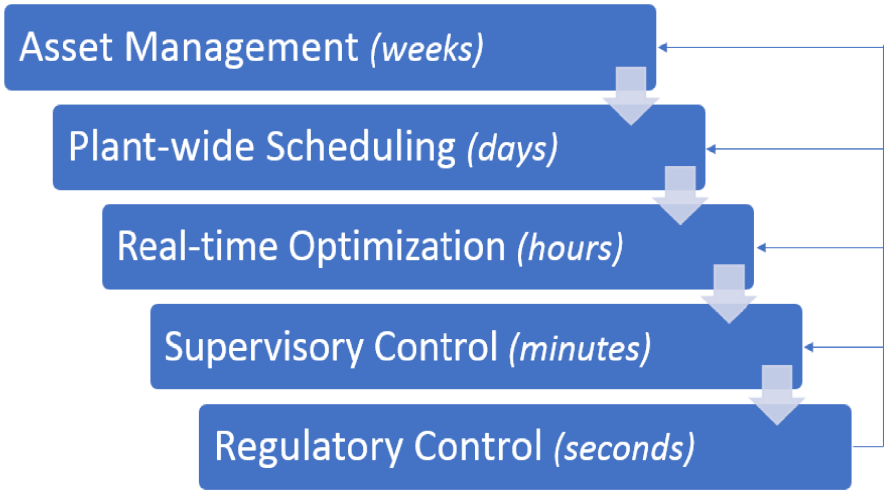

Figure 1.1: Typical control hierarchy of a process with each layer's time-scale. *Source*: adapted from KRISHNAMOORTHY *et al.* (2018).

RTO layer has an economical bias, the objective function in the MPC layer has accuracy and prediction purposes. As a result there are two layers that receive distinct information with distinct goals (DARBY *et al.*, 2011).

KRISHNAMOORTHY (2019) points out that the main challenges associated with the practical use of RTO are: development cost of the model, model uncertainties, numerical robustness, frequent product changes, dynamic limitations and problem formulation. Regarding RTO implementation, the dynamic limitation is a major problem, as it usually uses stationary process models in order to calculate the reference of optimum operation. This practice has a great operational setback because the process becomes extremely dependent on the steady state, where any system disturbance or model uncertainty shall probably compromise plant operation. In DARBY *et al.* (2011), the authors considered this the main limitation of the traditional RTO. Such a challenge favors the use of dynamic optimization.

As the name suggests, dynamic real-time optimization (DRTO) techniques rule out the steady state detection step while making use of dynamic models. In DRTO, a reference trajectory is calculated and the advanced control problem is solved simultaneously (HELBIG *et al.*, 2000). In conjunction with DRTO algorithms, there has also been great development of the economic MPC (EMPC) (AMRIT *et al.*, 2011; CHEN *et al.*, 2012; HUANG *et al.*, 2011). In this case, the economic and control problems are solved simultaneously in the same layer. In contrast, many authors point to a number of problems associated with DRTO/EMPC (CAMPOS *et al.*, 2009). These problems will be best discussed in the next chapter, but it can already be inferred that the main limiting factor is the computational cost involved in solving a nonlinear dynamic optimization

problem in large scale processes.

Driven by these limitations, some authors (KRISHNAMOORTHY *et al.*, 2018; MATIAS and LE ROUX, 2018) have suggested an alternative hybrid approach for real-time optimization that combines the strengths of traditional RTO and DRTO: the static optimization combined with the dynamic adaptation of the model. The case studies addressed in both papers will be briefly discussed in the following chapter. As this is a very recent approach, the literature still lacks of a larger number of case studies that can validate it, including cases that present high nonlinear behavior at certain operating ranges.

1.3 Objectives

Given the above, the general goal of this work is to evaluate this hybrid approach by employing a benchmark problem as case study: the Van der Vusse CSTR (VAN DE VUSSE, 1962). The main purpose is to analyze the performance of the approach in adverse operational conditions from the dynamic point of view, like inverse responses and overshoots. HRTO's performance shall be compared with traditional RTO and DRTO.

In order to this specific goal to be attained, the following secondary objectives are listed:

- Process model definition and implementation;
- Study concerning the dynamics of the case study in order to determine different operational regimes that present complex behavior;
- Stationary and dynamic optimization problems formulation and optimization solver;
- Implementation of a steady-state detection algorithm for traditional RTO;
- Application of advanced control based on the nonlinear process model coupled with a static parameter estimator (for RTO) or a observer (for DRTO and HRTO) for setpoint attainment under uncertainties;
- Integration of process optimizer, control and estimator into a single framework for each RTO approach.

1.4 Document structure

The present work is organized as follows: Chapter 2 contains the literature review, the strands of optimization and advanced control are discussed in more

details and with greater mathematical rigor. In addition, KRISHNAMOORTHY *et al.* (2018)'s and MATIAS and LE ROUX (2018)'s works will also be briefly discussed, whose applied similar online model parameter estimation approaches. Chapter 3 is focused on the case study adopted, featuring deeper discussions regarding its model. Chapter 4 features the methodology applied. Chapter 5 presents and discusses the results obtained and Chapter 6 ends this dissertation with the conclusion and final remarks.

Chapter 2

Literature review

2.1 Contextualization of the different RTO techniques

2.1.1 Stationary real time optimization - RTO

As stated in Chapter 1, an RTO system acts with the intent of operating a plant as close as possible to its optimality without violating constraints that each subsystem may be subjected to. The most common structure of the traditional RTO is illustrated in Figure 2.1. In this structure, the steady-state model used is parameterized by a group of parameters, whether uncertain or unknown, and these are updated through real-time stationary process measurements. After this step, the optimization proceeds with the updated steady-state model, generating the optimal setpoint for each of the controlled variables (CHEN and JOSEPH, 1987; MARLIN and HRYMAK, 1997). In view of the above, the traditional RTO can also be called *model parameter adaptation* (MPA).

Looking at Figure 2.1, the traditional RTO framework is composed of the following main steps (CHEN and JOSEPH, 1987):

- Steady-state detection, as well as process data preprocessing: only responsible for assessing whether the plant is operating near enough stationarity. In this phase, statistics that use process measurement variations are usually employed;

- Steady-state parameter estimator: in this case an online parameter estimator, a step that encompasses the data reconciliation and adaptation of the RTO Steady-state model. The model parameters and non-measured disturbances are updated in order to match process data through regression techniques;

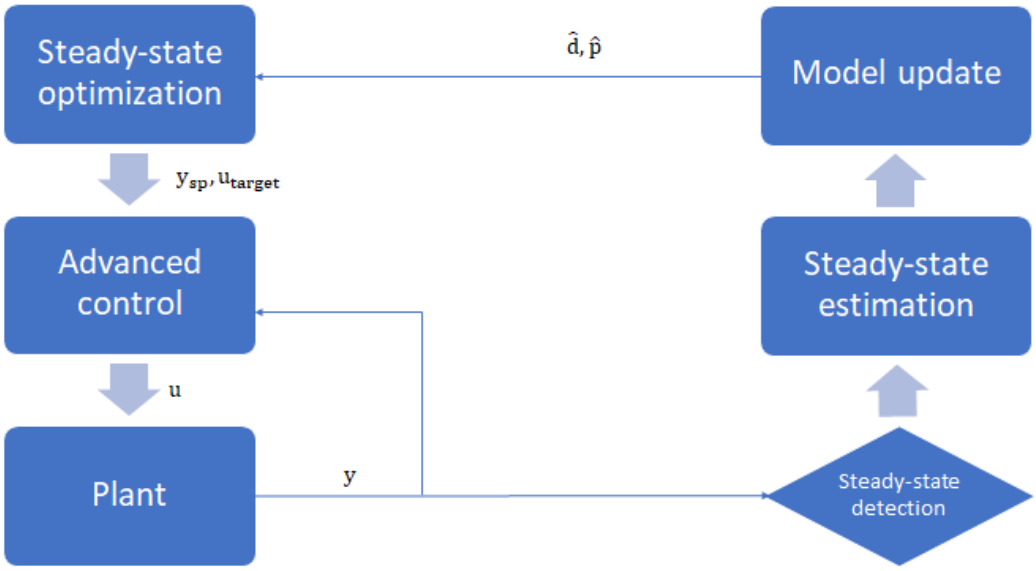

Figure 2.1: Architecture of a traditional RTO with steady-state model adaptation.

- Steady-state optimization: with the process model properly adjusted plus the constraints and the economic objective function formulated as well, the optimization problem is solved and returns the optimal setpoint of the controlled variables and any target values of the manipulated variables that may be used by the control layer;
- Advanced control: usually an MPC, this step is responsible for computing the output of the steady-state optimization layer and, with its internal model updated, calculate the optimal control actions that shall be received by the plant.

On behalf of the crucial steps of parameter estimation and steady-state optimization, this technique is commonly referred to as *two-step approach*, being practically the only one used in the industry (MARLIN and HRYMAK, 1997; NAYSMITH and DOUGLAS, 1995; TRIERWEILER, 2014).

Representing the process through a discrete-time nonlinear model:

$$x_{k+1} = f(x_k, u_k, p_k, d_k) \tag{2.1}$$

$$y_k = h(x_k, u_k)$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, $y_k \in \mathbb{R}^{n_y}$, $p_k \in \mathbb{R}^{n_p}$ e $d_k \in \mathbb{R}^{n_d}$ are respectively the states, inputs, process measurements, parameters and disturbances at time step

$k \in \mathbb{N}$. Therefore, f and h are nonlinear functions such that $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_x}$ e $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$ and consist of process's constitutive equations (balances, equipment design equations, etc). To simplify the notation, concatenating the process parameters and disturbances into a single parameter vector $\theta = [p^T, d^T]^T \in \mathbb{R}^{n_\theta}$.

Since it is a traditional RTO, assuming the plant operating in steady state it can be stated that:

$$f(x, u, \theta) = 0 \tag{2.2}$$

This way, x is a function of u and θ, and it can also be implied that:

$$y = g_{SS}(u, \theta) \tag{2.3}$$

where $g_{SS} : \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_y}$ is the input-output model. The subscript SS indicates that this is a stationary model.

As detailed above, once the stationarity of the plant is determined at a given sampling time k, the parameters of the steady-state model are adjusted. This model update is nothing more than an optimization problem that seeks to minimize the quadratic error between process outputs and expected outputs from the steady-state model. Thus:

$$\hat{\theta}_k^* = arg\, min_{\theta_k} \; ||y_{meas} - g_{SS}(u_k, \theta_k)||_2^2 \tag{2.4}$$

where the superscript in $\hat{\theta}^*$ implies that it is the optimal parameters vector and y_{meas} is the plant measurement output vector.

In the optimization step, usually consisted of an *nonlinear programming* (NLP) problem, the optimal vector of the decision variables is determined by the model updated with the optimal vector of the parameters calculated in the previous step. Writing the formulated optimization problem:

$$u_{k+1}^{*} = arg\,min_u\ J(y,u) \tag{2.5}$$

$$s.t.\ \ y = g_{SS}(u, \hat{\theta}_k^{*})$$

$$l(y,u) \leq 0$$

where, u_{k+1}^{*} is the optimal input vector (target), which will proceed into the advanced control layer alongside the optimal states vector (setpoint) if the plant remains at the same steady-state detected at sampling time k; $J : \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$ is the economic objective function (in this case as cost-related function that will be minimized); while $l : \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_l}$ are all equality or inequality constraints of the system.

Another optimization problem between the RTO and MPC layers is usually formulated as a LP or QP whose goal is to determine the operating point of the plant from the economic point of view and provide it to the MPC as a realizable setpoint, setting up a widespread cascade control scheme in the industry (QIN and BADGWELL, 1997; RAO and RAWLINGS, 1999; SORENSEN and CUTLER, 1998; YING and JOSEPH, 1999).

As mentioned in Chapter 1, the main challenge involving traditional RTO algorithms is the steady-state detection step. This procedure usually makes use of statistics or heuristics and tolerances of reference are employed to assess whether the system is sufficiently close to steady-state operation (CAO and RHINEHART, 1995). Thus, any mistaken validation of these tolerances is reason enough for a faulty detection of the process stationarity, i.e., the steady-state model update would be mistakenly performed through transient process measurements. Several studies have, in fact, found out that the use of process measurements in order to adjust model parameters does not guarantee its suitability for optimizing (FORBES *et al.*, 1994; ROBERTS, 1978). This problem is known as *plant-model mismatch*. This becomes an even more recurring issue in processes that have slow dynamics, frequent transitions and possible non-measured disturbances (JAMALUDIN and SWARTZ, 2017). The work of WHITE (1998) stated that, for perturbations with longer periods of time, traditional RTO procedures are not realizable.

Some authors have proposed solutions to such problem. Since the steady-state detection step demands long sampling periods, it has been proposed reducing the time interval between each update of the setpoint intended for the control layer, as well as the real-time evolution of such setpoints through heuris-

tic searches (SEQUEIRA *et al.*, 2002). In this case, the proposed procedure improves the optimization and control procedures.

In QUELHAS *et al.* (2012), the authors discuss the most commonly observed challenges of implementing real-time optimization algorithms, such as possible measurement failures, corrupted or incomplete processes, as well as performance adversity related to convergence in a optimization problem and, as previously mentioned, inconsistencies between the optimization layer and the advanced control layer. Many of these traditional RTO vulnerabilities are extremely common in real-world industrial processes and often a consequence of decision making at the plant design layer.

2.1.2 Dynamic real-time optimization - DRTO

Unlike traditional RTO, DRTO algorithms perform the calculation of a trajectory of the optimal decision variables of the plant through a dynamic model of the process. The DRTO schema is similar to its stationary counterpart: with the collection of plant information, it updates the dynamic model and calculates a setpoint to be implemented by the control layer. However, the following key points that differ DRTO from the earlier discussed RTO are:

1. Steady-state detection is no longer needed, with only initialization and preprocessing of the data from the plant;

2. State estimator (observer): the estimation step is now composed of a online state-parameter estimator. This step encompasses the dynamic reconciliation of data and adaptation of the dynamic model of DRTO and the model parameters are updated to match the process data. As the model in this case is composed of differential-algebraic equations in time, it is necessary to estimate the initial state vector of the system at the current sampling time. As the dynamic reconciliation, it is reinforced that the plant does not need to be in the steady state, performed at each sampling cycle of plant information. Different methods can be employed in order to perform the dynamic estimation for nonlinear systems, as some examples are recursive methods suach as: *recursive squares*, *extended Kalman filtering* (EKF), *constrained Kalman filtering* (CEKF), *unscented Kalman filtering* (UKF) and methods based on optimization as *moving horizon estimation* (MHE), among others;

3. Dynamic optimization: with the properly adjusted process model, the optimization problem is solved and returns the optimal trajectory in a certain horizon for each of the design variables. This step is the motivating agent

of most studies in the optimization area and can be solved through a variety of approaches, most by adapting the discretization of the system time meshes (MAGALHÃES, 2010) and making use of NLP solvers.

In Figure 2.2, a simplified scheme of the procedure described above follows.

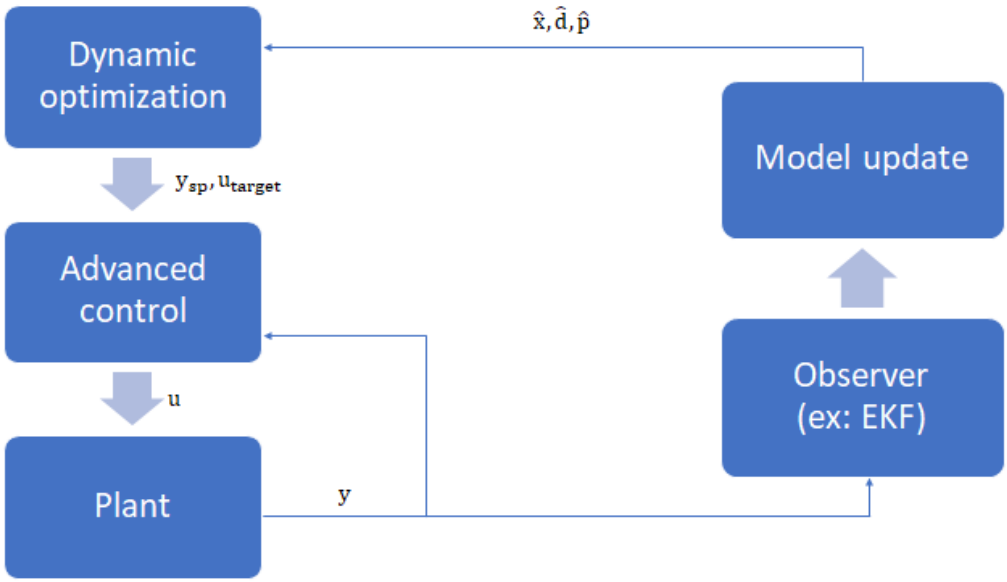

Figure 2.2: General architecture of a DRTO with dynamic model adaption.

With the similarity between these two approaches, having as main steps the estimation of parameters/states and the optimization itself, the illustrated structure also configures a two-step approach.

At this moment, in order to write the equations for DRTO, returning to the general modeling proposed in Equation (2.1) there is now the estimation of states and parameters that is given by the following optimization problem:

$$\hat{\theta}_k = arg\,min_\theta \; ||y_{meas,k} - h(x_k, u_k)|| \tag{2.6}$$

$$s.t. \quad x_k = f(x_{k-1}, u_{k-1}, \theta)$$

The dynamic optimization problem is given as:

$$u_t^* = arg\,min_{u_t} \sum_{t=k}^{k+N} J(y_t, u_t) \;\; or \;\; arg\,min_{u_t} J(y_N, u_N) \tag{2.7}$$

$$s.t. \;\; x_{t+1} = f(x_t, u_t, \hat{\theta}_k)$$

$$y_t = h(x_t, u_t)$$

$$l(y_t, u_t) \leq 0 \qquad \forall \; t \in [k, k+N]$$

$$x_k = \hat{x}_k$$

where N is the optimization horizon, $\hat{x}_k$ is the state estimate at the current sampling time k and u_t^* indicates the optimal input vector for each time step t, configuring several optimal vectors, i.e., an optimal trajectory. It is noteworthy that, in this case, the cost function can be formulated as a sum for the whole trajectory or only as its value ate the end of the horizon.

Despite the steady-state detection step limitation not being present in DRTO algorithms, another major problem arises: the computational cost associated with dynamic optimization. Solving a dynamic optimization problem on nonlinear systems can be a highly costly task even with the clusters in today's computational labs, especially for large-scale systems. Given by this fact alone, the industry in general still opts for the application of the traditional RTO (KRISHNAMOORTHY *et al.*, 2018).

2.1.3 Hybrid real-time optimization - HRTO

The HRTO method consists of the steady-state optimization of an economic objective function, in addition with the adaptation of the dynamic model of the process. Thus, it is possible to use real-time transient process measurements (and by doing so eliminating both steady-state waiting time and mass process data discarding). With this, the updated model parameters are used in the steady-state optimization step, drastically reducing the computational effort required in order to achieve solving the problem when compared with the DRTO approach. This combined approach illustrated in Figure 2.3.

Thus, the implementation of the technique also becomes a combination of the approaches studied in the previous subsections, containing both discrete-time dynamic model (Eq. 2.1) and steady-state model (Eq. 2.3). Therefore, the

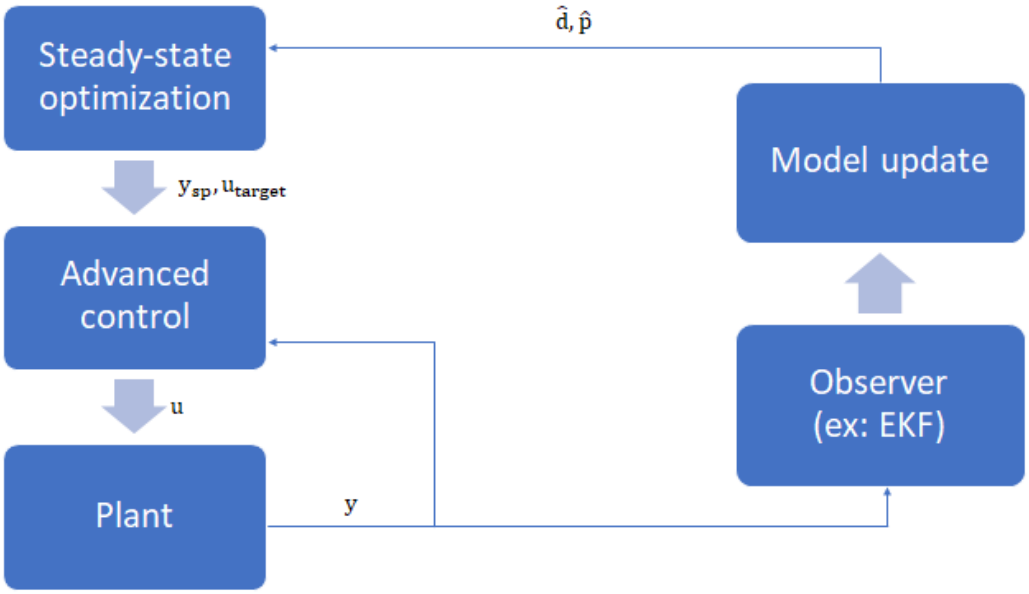

Figure 2.3: HRTO architecture.

estimation step becomes:

$$\hat{\theta}_k = arg\,min_\theta \; ||y_{meas,k} - h(x_k, u_k)|| \tag{2.8}$$

$$s.t. \quad x_k = f(x_{k-1}, u_{k-1}, \theta)$$

While the optimization problem is given by its stationary version:

$$u^*_{k+1} = arg\,min_u \; J(y, u) \tag{2.9}$$

$$s.t. \quad y = g_{SS}(u, \hat{\theta}_k)$$

$$l(y, u) \leq 0$$

Unlike traditional RTO, the steady-state optimization problem is solved at every time step k making use of $\hat{\theta}_k$ in order to update the steady-state model employed, whether the system is at steady-state or not.

KRISHNAMOORTHY *et al.* (2018) proposed an hybrid RTO (HRTO) making use of transient plant data for dynamic model adaptation followed by steady-

state optimization. The method was applied to a gas production network and performances comparisons were made between RTO, DRTO and HRTO approaches.

In their work, KRISHNAMOORTHY *et al.* (2018) formulate the optimization problem in order to determine the gas injection rate of each well that maximized oil production, without violating the operational constraints of the process. The total profit of the process was applied as the objective function of the optimization problem.

In order to numerically solve the presented optimization problem, the authors used as environment MATLAB R2017a commercial software with CasADi v3.0.1 package together with *IPOPT* v3.12.2 (WÄCHTER and BIEGLER, 2006). CasADi is an open-source tool for NLP problems and algebraic differentiating available in many environments like MATLAB and Python, while IPOPT (*Interior-Point Optimizer*) is a library written in C and FORTRAN intended for solving a large number of NLP problems. An NMPC and an EKF for state and parameter estimation were used in the advanced control layer. In the parameter estimation step, in all cases, plant variables of simple measurement were employed, such as oil flow and pressure on the riser's head, to estimate the gas-oil ratio (GOR) in each well, taken as unknown parameters.

The hybrid algorithm performed similarly to the DRTO. The optimal values for the objective function were relatively close between the hybrid and dynamical approaches, while the CPU time spent were consistent with traditional RTO. The authors concluded that, when comparing these approaches, the standard RTO and hybrid RTO are preferred thanks to the advantages steady-state optimization has. The dynamic optimization problem in this case consisted of over 3000 variables, in contrast of only 22 of the steady-state optimization. Between the standard RTO and HRTO, the first one performed considerably worse thanks to the steady-state waiting time, with HRTO operating at the same steady-state as the DRTO.

Analogous to KRISHNAMOORTHY *et al.* (2018), in this work the author proposed an approach of continuously adjusting the optimal setpoint through steady-state optimization and online dynamic model updates, with also the use of online estimation methods. The authors named this strategy *real-time optimization with persistent adaptation* (ROPA) and applied it in three separate case studies, comparing the performances between this approach and traditional RTO. The case studies tested were: the Williams-Otto reactor, a fluid catalytic cracking unit and a complete plant containing one reactor and one separation step.

For the parameter estimation step it was implemented a *reduced rank extended Kalman filter* (rEKF) for the first case study and an EKF for the third one. rEKF

is a variant of the previously described EKF that has some appearances in the literature regarding chemical processes. It differs from the known EKF only by a lower order approximation of the covariance of the error estimation matrix (PHAM *et al.*, 1998).

A classic MPC was implemented for the advanced control layer. For the second case study no multivariable control method was applied, with a PI controller being implemented.

Regarding the results, it was once again stated the best performance of the hybrid approach when compared with traditional RTO. Since the author did not implement an DRTO, it is not possible to compare its performance with ROPA. An interesting observation is that the final case study illustrated one more advantage of the hybrid approach: the possibility of decoupling between subsystems. The algorithm ensures model updating by dividing the estimation problem into decoupled subproblems. By dividing the plant model into partitions, the online estimation was applied to the submodels of the plant that had low parameter update frequency, which allows the update all sections of the model.

KRISHNAMOORTHY *et al.* (2018)'s and MATIAS and LE ROUX (2018)'s works share a lot of similarities and are the base of the approach developed in this work. Despite this, there are several researches concerning the use of transient data for steady-state optimization prior to theirs.

GOLSHAN *et al.* (2008) used in their work an approch called *heuristic random optimization* (HRO) and applied it to the well-known Tennessee Eastman process. The HRO method was first presented by LI and RHINEHART (1998) and its main feature is the use of several heuristic principles while depending only on the value of the objective function, not depending on gradients (RAO, 2019). The main contribution of this work was a comparison between HRO, an heuristic method, with *sequential quadratic programming* (SQP), considered the best among deterministic optimization approaches (RAO, 2019). The results showed that HRO could be applied online and had as main advantages over SQP: no dependency on the initial point neither on gradient information and it does not get stuck in local optimum, even though these features come with the cost of lower convergence times.

FRANÇOIS and BONVIN (2014)'s work was based on the *modifier adaptation* (MA) method, a RTO variant. MA methodes incorporate plant information to the optimization problem through modifiers with the goal of matching the output and gradient predictions from the model with plant measurements and estimated gradients. A zeroth order modifier is applied for the objective function while a first order one is applied for the constraints and they are updated iter-

atively as new plant measurements become available (MATIAS and JASCHKE, 2019). This technique was applied to a CSTR and the results showed that the MA using transient process data was competitive with the standard RTO algorithms. On the same line, DE AVILA FERREIRA *et al.* (2017) implemented an almost identical scheme, mainly differing on the use of dynamic models for gradient estimates.

2.2 Model predictive control - MPC

2.2.1 Introduction

Model-based predictive control is an advanced control technique that makes explicit use of a model to predict process output in future time moments, characterizing a prediction horizon. By means of the model, predictions are made about the behavior of the plant and, next, the implementation of the control actions takes place in a moving or receding horizon. In order to determine these control actions, an optimization problem is solved internally (CAMACHO and BORDONS, 2004)

Thus, the basic structure of an MPC consists of four elements: specifying a reference path for the process outputs, prediction of future outputs, computation of the control actions sequence and error prediction update. Thus, the algorithm followed by an MPC follows the following steps (CAMACHO and BORDONS, 2007; DOS SANTOS, 2007):

1. Receiving the reference path from the RTO layer, i.e., a sequence of future setpoints;

2. Based on the information passed from the manipulated variables and on the reference path, the algorithm computes the prediction the process outputs on a prediction horizon and the deviation from the reference path;

3. With these deviations, an optimization routine is pursued to minimize a quadratic objective function consisting of future errors and control actions. Thus, a sequence of future control actions is suggested within the adopted control horizon.

Standard MPC capabilities, according to OGUNNAIKE and RAY (1994), include: ease of handling multiple inputs and multiple outputs as well as dead-time processes, inverse response and other nonlinearities, as well the often lack of necessity of a robust internal model, being discrete linear formulations such

as convolution models, discrete state-space models and discrete transfer function models. Such factors made MPC become a widespread control strategy in the industry, being applied to most diverse types of equipment such as reactors, distillation columns, extruders, cracking units, jet engines, among others (QIN and BADGWELL, 1997).

It is possible to note that, in an ideal scenario where the model is perfectly representative of the system, eventually the deviation from the setpoints become zero within the prediction horizon, as seen in Figure 2.5 (SEBORG *et al.*, 2010).

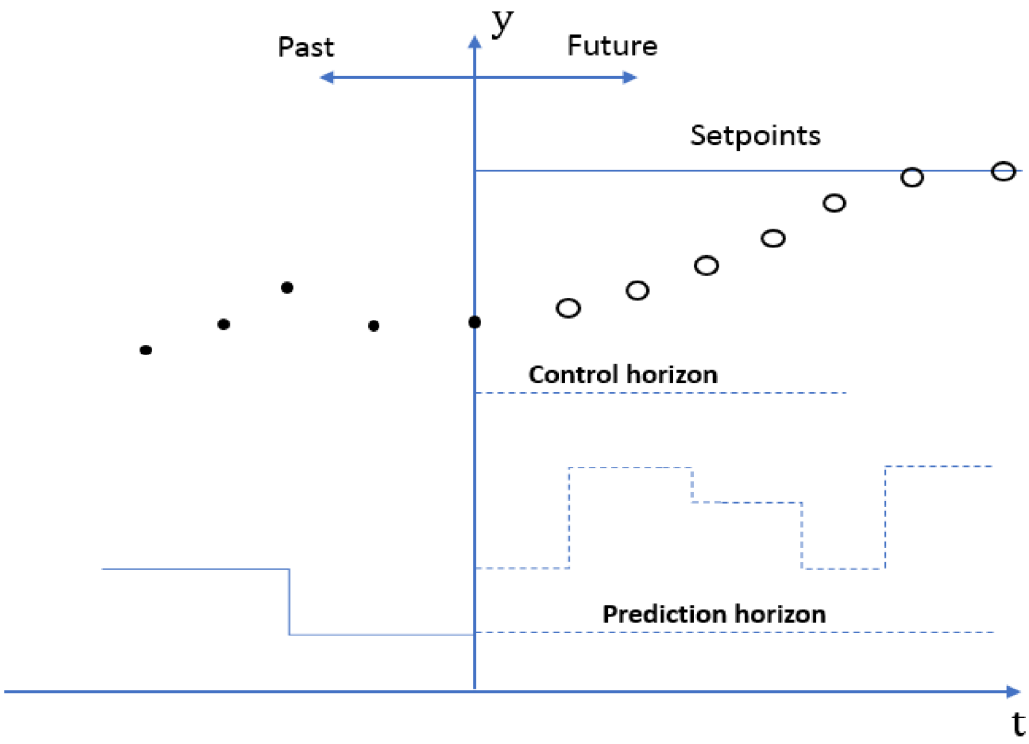

Figure 2.4: Illustration of an ideal MPC. *Source*: adapted from THOMAZ (2017).

Thus, the main difference between classic controllers and this class of model-based controllers is, whereas the former makes use of the present system error for the determination of control actions, the MPC uses the present process states as well as the predictions future errors from its internal model, then performing the optimization routine in order to minimize such errors. It is also noteworthy that, as MPC's internal model will never be perfect: the optimization calculation for the determination of control actions is performed at each sampling time, while only the first optimal control action from the whole horizon will be implemented. Thus, at the next sampling time, the process is repeated based on new information from the plant, a strategy known as *descending horizon* (QIN and BADGWELL, 1997).

2.2.2 Brief history

The pioneer works of this idea appeared in the 1960s (KALMAN, 1960). In this study, the author employed a state-space model, as well as a state estimator, and a function with an objective function with quadratic criterion, typical perks of a traditional MPC detailed earlier. In contrast, the control horizon used did not characterize the so-called moving horizon, but an infinite horizon was used. KALMAN (1960) has defined this approach as *linear-quadratic-gaussian* control (LQG). However, it was not until the 1970s that the widespread MPC concept known today came to prominence.

The first groups of engineers to design and market what would be considered the first generation of model-based predictive control were ADERSA (RICHALET *et al.*, 1978) and Shell Oil (CUTLER and REMARKER, 1980).

The first one developed technique was called *model predictive heuristic control* (MPHC) or *model algorithmic control* (MAC) and employed an internal model based on impulse response, reference trajectory and a vector of control actions, so that the difference between the quoted trajectory and the prediction from the model should be as small as possible. In this study, the authors reported applications in industrial systems such as polymer production plants, thermal generation and catalytic cracking.

The second group, consisted of Shell Oil engineers, developed the *dynamic matrix control* (DMC) algorithm which adopted an analogous approach to MPHC but using a model based on response to truncated step. In this paper, the authors introduced the concept of control horizon, being the main responsible for the emergence of the MPC in the late 70s. However, in this first generation the constraints to the process had not been introduced, which occurred only in the 1980s in the form of the *quadratic dynamic matrix control* (QDMC) algorithm (GARCIA and MORSHEDI, 1986a). The DMC was formulated as a QP, which allowed the insertion of constraints explicitly.

Model-based predictive control has evolved greatly over the years, and it was not until the late 1990s that the first versions of MPC based on nonlinear models emerged (HENSON, 1998; QIN and BADGWELL, 2000). Named after NMPC, this methodology emerged in order to expand the already wide model-based predictive control applicability for highly nonlinear systems.

2.2.3 Classic MPC implementation

As detailed in the previous subsections, the built-in MPC internal prediction model is linear, discrete and time-invariant. Considering, for example, a linearized state-space dynamic model by means of a Taylor series expansion trun-

cated in the first-order term. Thus, with this expansion around the steady state-nominal vectors of states and inputs, x_{SS} and u_{SS}, still having θ_{SS} as the nominal vector of the parameters:

$$x_{k+1} \approx Ax_k + Bu_k \tag{2.10}$$

$$y_k \approx Cx_k$$

This is the process model in its discrete-time state-space formulation. The matrix A is the system matrix, B is the input matrix and C is the output matrix. For a continuous-time linear system, A and B are the first-order partial derivatives from the Taylor series expansion, resembled as A_c and B_c in this case:

$$A_c = \left.\frac{\partial f(x,u,\theta)}{\partial x}\right|_{x=x_{SS},\, u=u_{SS},\, \theta=\theta_{SS}} \tag{2.11}$$

$$B_c = \left.\frac{\partial f(x,u,\theta)}{\partial u}\right|_{x=x_{SS},\, u=u_{SS},\, \theta=\theta_{SS}} \tag{2.12}$$

The discrete-time approximation is give by:

$$A = e^{A_c \Delta t} \tag{2.13}$$

$$B = e^{A_c \Delta t} B_c \int_0^{\Delta t} e^{-A_c \Delta t} d\tau = A\left[I - e^{-A_c \Delta t}\right] A_c^{-1} B_c \tag{2.14}$$

It is common to introduce integral action into MPC controllers using incremental variables at each time step for system states and inputs. This notation is called speed formulation and is a common approach in the design of offset-free MPCs (BETTI *et al.*, 2012; GONZÁLEZ *et al.*, 2008; PANNOCCHIA and RAWLINGS, 2001; WANG, 2004). Rewriting the model in Eq. (2.10) with $\Delta x_k = x_k - x_{k-1}$ and $\Delta u_k = u_k - u_{k-1}$:

$$\Delta x_{k+1} = A\Delta x_k + B\Delta u_k \tag{2.15}$$

$$\Delta y_k = C\Delta x_k$$

$$y_{k+1} = y_k + C\Delta x_{k+1} = y_k + CA\Delta x_k + CB\Delta u_k$$

The model above will then be used to calculate the optimal sequence of inputs at each time step within the control horizon:

$$U^* = [u^*_{k|k}, u^*_{k+1|k}, ..., u^*_{k+M-1|k}] \tag{2.16}$$

$$M \leq N \tag{2.17}$$

where M and N are the control and prediction horizons, respectively.

Thus, the sequence of the manipulated variables of the process is given by solving the following optimization problem:

$$U^* = arg\,min_U \sum_{i=1}^{N} ||y_{k+i|k} - y_k^{SP}||_Q^2 + \sum_{i=0}^{M-1} ||\Delta u_{k+i}||_W^2 \tag{2.18}$$

$$s.t.\ \ \Delta x_{k+i+1} = A\Delta x_{k+i} + B\Delta u_{k+i},\ i = 0, ..., N-1$$

$$\Delta y_{k+i} = C\Delta x_{k+i},\ i = 1, ..., N$$

$$y^{min} \leq y_{k+i} \leq y^{max},\ i = 1, ..., N$$

$$u^{min} \leq u_{k+i} \leq u^{max},\ i = 0, ..., M-1$$

$$\Delta u^{min} \leq \Delta u_{k+i} \leq \Delta u^{max},\ i = 0, ..., M-1$$

where the subscript $k+i|k$ indicates the prediction at $k+i$, calculated with plant information obtained at k. Q and W are the proper dimensioned positive defined matrices that weight the deviation from the setpoint and from the increment of the manipulated variable, respectively. This latter is also commonly referred to as the motion suppression coefficient and the matrices are commonly chosen as time invariant, i.e., not functions of i. These matrices are the MPC's main parameters that must be tuned by the user. The *min* and *max* supercripts indicate, respectively, the lower and upper bounds consistent with

the constraints employed in the RTO layer preceding the MPC. Finally, y^{SP} is the setpoint coming from the RTO.

For simplicity, Eq. (2.17) was written in terms of weighted Euclidean norms, defined as:

$$||y_{k+i|k} - y_k^{SP}||_Q^2 = (y_{k+i|k} - y_k^{SP})^T Q\,(y_{k+i|k} - y_k^{SP}) \tag{2.19}$$

$$||\Delta u_{k+i}||_W^2 = \Delta u_{k+i}^T W\,\Delta u_{k+i} \tag{2.20}$$

MPC's quadratic cost function has several variations. The ultimate goal remains being setpoint tracking but one can consider a reference trajectory for the manipulated variables as well (CAMACHO and BORDONS, 2007). Thus, including a target trajectory for the inputs into the MPC cost function:

$$\sum_{i=1}^{N} ||y_{k+i|k} - y_k^{SP}||_Q^2 + \sum_{i=0}^{M-1} \left[||\Delta u_{k+i}||_W^2 + ||u_{k+i|k} - u_{k+i|k}^{target}||_S^2 \right] \tag{2.21}$$

with matrix S playing the same role as Q and W.

Thus, a classic MPC is defined by its linear internal model, linear constraints and a quadratic cost function, making possible the use of QP solvers.

Regarding NMPC, the notable difference is that this variant is characterized by the use of nonlinear system models in the prediction. By this fact alone, since the problem becomes an NLP, the computational cost is increased drastically. Thus, the need of using an NMPC instead of an MPC should be well evaluated, being deeply dependant on the process behavior itself.

2.3 State estimation - Kalman filtering

2.3.1 Introduction

In control theory, the state of a dynamic system is defined as the minimum set of variables that, when combined with its known value at a given initial time and the system manipulated variables, exactly describes the system's condition at any time step (OGATA, 2010).

If this set of variables is not completely known or contains inaccuracies such as statistical noise or other uncertainties, or even if the system output consists of linear or non-linear state combinations, online state estimators are required (MACIEJOWSKI, 2002).

Kalman filtering (KF), also known as linear quadratic estimation (LQE) is a technique that aims to produces estimates of unknown variables based on several measurements observed over time. These estimates tend to be more precise than those based on a single measurement alone and they are performed by a joint probability distribution over the variables for each timeframe. .

The first KF version was formalized in the 1960s by KALMAN (1960). This formulation was based on linear discrete-time systems and it is the basis of several developments that would come to fruition later. For nonlinear systems, the most common alternative is the *extended Kalman filtering* (EKF).

On the next subsections, the fundamentals of discrete-time KF and EKF will be highlighted based on SIMON (2006).

2.3.2 The discrete-time Kalman filter - KF

Considering the following discrete-time linear system:

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} \tag{2.22}$$

$$y_k = C_k x_k + v_k$$

where w_k and v_k are process noise defined as white, zero-mean, uncorrelated, and have known covariance matrices Q_k and R_k:

$$w_k \sim (0, Q_k) \tag{2.23}$$

$$v_k \sim (0, R_k)$$

The KF mainly uses two different estimations of the same state variables: a *priori* estimate ($\hat{x}_k^-$) and a *posteriori* estimate ($\hat{x}_k^+$). The first one is defined as the estimate of the states at time step k before the measurements at this time step are processed and if the output measurement y_k is known, the estimate $\hat{x}_k^+$ is a *posteriori* estimate. Therefore, the *posteriori* estimate at time k is carried out to the next step $k+1$ as a *priori* estimate.

It is important to highlight that the initial state estimate, i.e., at time $k = 0$, is a *posteriori* estimate and it is also the expected value of the initial state:

$$\hat{x}_0^+ = E(x_0) \tag{2.24}$$

Denoting P_k as the covariance matrix of the estimation error, one can write the equations for the *priori* and *posteriori* covariance of the estimation error as:

$$P_k^- = E\Big[(x_k - x_k^-)(x_k - x_k^-)^T\Big] \tag{2.25}$$

$$P_k^+ = E\Big[(x_k - x_k^+)(x_k - x_k^+)^T\Big]$$

Using the process model to compute the *priori* estimate using the *posteriori* estimate from the previous time step:

$$\hat{x}_k^- = A_{k-1}\hat{x}_{k-1}^+ + B_{k-1}u_{k-1} \tag{2.26}$$

Similarly for the covariance matrix of the estimation error:

$$P_k^- = A_{k-1}P_{k-1}^+A_{k-1}^T + Q_{k-1} \tag{2.27}$$

Using recursive least squares, the *posteriori* estimates of x_k and P_k can be expressed as:

$$\begin{aligned} K_k &= P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1} \\ &= P_k^+ C_k^T R_k^{-1} \end{aligned} \tag{2.28}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - C_k\hat{x}_k^-) \tag{2.29}$$

$$\begin{aligned} P_k^+ &= (I - K_kC_k)P_k^-(I - K_kC_k)T + K_kR_kK_k^T \\ &= ((P_k^-)^{-1} + C_k^TR_k^{-1}C_k)^{-1} \\ &= (I - K_kC_k)P_k^- \end{aligned} \tag{2.30}$$

where K_k is denoted as the *Kalman gain* at time step k.

One crucial point that must be noted is that the first *posteriori* state estimate, $x_0^+ = E(x_0)$, and the first *posteriori* covariance matrix of the estimation error, P_0^+, must be guessed in order to initialize the KF algorithm. It is also noteworthy that the expressions for the Kalman gain and the covariance matrix of the estimation error do not depend on the measurements y_k, but only on the noise and system parameters, Q_k, R_k, A_k and C_k. Thus, implying that K_k, P_k^- and P_k^+ can be computed offline and stored in memory before process operation initializes. Although this fact does not stand true for KF formulations designed for nonlinear systems.

2.3.3 The discrete-time extended Kalman filter - EKF

This time, consider the following discrete-time nonlinear system:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{2.31}$$

$$y_k = h(x_k, v_k)$$

$$w_k \sim (0, Q_k)$$

$$v_k \sim (0, R_k)$$

which f and h are nonlinear functions.

Performing a first-order Taylor series expansion of f and h around the *posteriori* estimate x_{k-1}^+:

$$\begin{aligned} x_k &= f(x_{k-1}^+, u_{k-1}, 0) + \left.\frac{\partial f}{\partial x}\right|_{x_{k-1}^+} (x_{k-1} - x_{k-1}^+) + \left.\frac{\partial f}{\partial w}\right|_{x_{k-1}^+} w_{k-1} \\ &= f(x_{k-1}^+, u_{k-1}, 0) + F_{k-1}(x_{k-1} - x_{k-1}^+) + L_{k-1} w_{k-1} \\ &= F_{k-1} x_{k-1} + [f(x_{k-1}^+, u_{k-1}, 0) - F_{k-1} x_{k-1}^+] + L_{k-1} w_{k-1} \\ &= F_{k-1} x_{k-1} + \tilde{u}_{k-1} + \tilde{w}_{k-1} \end{aligned} \tag{2.32}$$

where the matrices F_{k-1} and L_{k-1} and the vectors $\tilde{u}_{k-1}$ and $\tilde{w}_{k-1}$ are:

$$F_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{x_{k-1}^+} \tag{2.33}$$

$$L_{k-1} = \left.\frac{\partial f}{\partial w}\right|_{x_{k-1}^+}$$

$$\tilde{u}_{k-1} = f(x_{k-1}^+, u_{k-1}, 0) - F_k \hat{x}_k^+$$

$$\tilde{w}_{k-1} \sim (0, L_K Q_k L_k^T)$$

Performing the same to the output equation with an first order expasion around x_k^-:

$$y_k = H_k x_k + z_k + \tilde{v}_k \tag{2.34}$$

$$H_k = \left.\frac{\partial h}{\partial x}\right|_{x_k^-}$$

$$M_k = \left.\frac{\partial h}{\partial v}\right|_{x_k^-}$$

$$z_k = h(x_k^-, 0) - H_k \hat{x}_k^-$$

$$\tilde{v}_k \sim (0, M_k R_k M_k^T)$$

Now, applying the KF equations from the previous subsection:

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + L_{k+1} Q_{k+1} L_{k+1}^T \tag{2.35}$$

$$K_K = P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \tag{2.36}$$

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, 0) \tag{2.37}$$

$$z_k = h(\hat{x}_k^-, 0) - H_k \hat{x}_k^- \tag{2.38}$$

$$\begin{aligned} \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k \hat{x}_k^- - z_k) \\ &= \hat{x}_k^- + K_k(y_k - h(\hat{x}_k^-, 0)) \end{aligned} \tag{2.39}$$

$$P_k^+ = (I - K_k H_k) P_k^- \tag{2.40}$$

Therefore, the EKF algorithm consists of the following steps:

1. System's model and noise definitiom;
2. EKF's initialization with $\hat{x}_0^+ = E(x_0)$ and $P_0^+ = E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]$;
3. Computation of F_{k-1} and L_{k-1};
4. P_k^- and $\hat{x}_k^-$ calculation;
5. Computation of H_k and M_k;
6. K_k, P_k^+ and $\hat{x}_k^+$ calculation;
7. Back to step 3 for instant $k+1$.

2.4 Final remarks

As detailed on Section 2.1 the works of KRISHNAMOORTHY *et al.* (2018) and MATIAS and LE ROUX (2018) explored a similar hybrid approach to real time optimization, where a steady-state optimization step is performed alongside the adaptation of the dynamic model, usually with a Kalman filter variant.

Each work contributed in their own way to this subject but one observation can be taken from both: better performance from the hybrid variant. It is noteworthy that MATIAS and LE ROUX (2018) did not perform DRTO simulations. Besides the fact that DRTO is not usually applied in industry, due the setbacks mentioned in Chapter 1, the DRTO results would provide the best possible reference in order to quantify and compare RTO and ROPA performances.

Even though the hybrid method stood out as the best performing RTO in both works, the subject is far from matured and it needs to be applied into several different system classes.

Since, in KRISHNAMOORTHY *et al.* (2018)'s work, the main divergence between DRTO and HRTO was notable in transient operation, a natural idea that can take place concerning the subject is applying it to complex dynamics systems, which is the main contribution of this work.

Chapter 3

Case study - Van de Vusse reactor

3.1 Introduction

The Van der Vusse reactor is a CSTR that contains a reaction system in which cyclopentadiene is used to synthesize cyclopentanol by electrophilic addition of water. The reaction is catalyzed by an acid solution (Diels-Alder reaction or cycloaddition), with the generation of cyclopentenediol and dicyclopentadiene as side products due to the high reactivity of reagents and products (ENGELL and KLATT, 1993). The complete chemical reactions are presented bellow:

$$C_5H_6 + H_2O/H^+ \xrightarrow{k_1} C_5H_7OH + H_2O/H^+ \xrightarrow{k_2} C_5H_8OH_2 \tag{3.1}$$

$$2\,C_5H_6 \xrightarrow{k_3} C_{10}H_{12} \tag{3.2}$$

Adopting the following nomenclature for the components: cyclopentadiene (A), cyclopentadiene (B), cyclopentenediol (C) and dicyclopentadiene (D), the simplified scheme becomes:

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C \tag{3.3}$$

$$2\,C \xrightarrow{k_3} E \tag{3.4}$$

Concerning this process, some hypothesis were considered in order to obtain its model:

- Continuous stirred-tank reactor (CSTR), i.e., perfect mixing;

- Constant volume;
- Incompressible fluid;
- Constant heat capacity;
- Elementary kinetic for all reactions;
- Reaction rate constants described by Arrhenius Law;
- No inlet flow of B, C and D;
- Negligible dynamics on the jacket.

Given the above, the complete system model proposed consists of the following differential equations, containing the mass balance equations for components A and B and the energy balance for the reactor.

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{Ain} - C_A) - k_1(T)C_A - k_3(T)C_A^2 \tag{3.5}$$

$$\frac{dC_B}{dt} = -\frac{F}{V}C_B + k_1(T)C_A - k_2(T)C_B \tag{3.6}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_{in} - T) + \frac{K_w A_R}{\rho c_p V}(T_k - T) + \frac{1}{\rho c_p}\left[k_1(T)C_A(-\Delta H_1) + k_2(T)C_B(-\Delta H_2) + k_3(T)C_A^2(-\Delta H_3)\right] \tag{3.7}$$

which C_A and C_B are the concentrations of components A and B in the reactor, C_{Ain} and T_{in} are the concentration of A and the temperature in the inlet respectively, T is the reactor temperature, T_k is the temperature of the jacket and F/V is the reactor spatial velocity. k_1, k_2 and k_3 are the rate constants of each reaction, dependent on the reactor temperature and expressed by Arrhenius Law:

$$k_i(T) = k_{0i} \exp\left(-\frac{E_i}{T(K)}\right), \quad i = 1, 2, 3 \tag{3.8}$$

which k_{0i} and E_i are the pre-exponential frequency factor of the Arrhenius Law and the activation energy of the reaction i divided by the universal gas constant.

Regarding the remain parameters, ΔH_i is the molar enthalpy of the reaction i, ρ and c_p are the specific mass and specific heat capacity of the reaction mixture,

K_w and A_R are the heat transfer coefficient and heat exchange area of the cooling jacket.

3.2 Motivation

The Van de Vusse CSTR is a well-known benchmark for multivariable control algorithms and nonlinear optimization. This is because this process combines nonlinear behavior in several operational ranges with a relatively simple modeling, which avoids possible disturbances on the results by its model reduction (MAGALHÃES, 2010).

As far as process optimization is concerned, it is usually aimed at maximize the yield of component B with the least generation. Some alternatives consider the conversion of B added to its selectivity.

Thus, for process control, there is often the need to maintain as controlled variables the concentration of B and the reactor temperature for safety reasons. For these purposes, the temperature of the jacket and spatial velocity will be chosen as manipulated variables.

Therefore, a control system containing *multiple inputs and multiple outputs* (MIMO) is featured, as shown in the scheme of Figure 3.1.

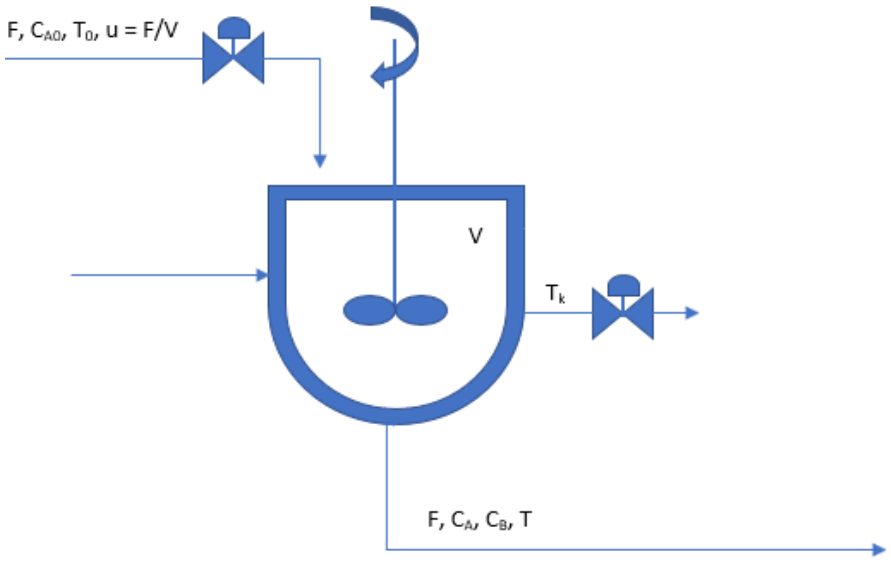

Figure 3.1: MIMO control diagram for Van de Vusse CSTR. *Source*: adapted from CUNHA NETO (2005).

The concentration of A and temperature on the inlet will act as process disturbances.

3.3 System dynamics

In the literature, there are several works reporting the nonlinear behaviors of the Van de Vusse CSTR.

In OGUNNAIKE and RAY (1994), the authors performed tests for a Van der Vusse CSTR presented in the work of ENGELL and KLATT (1993), characterizing different possible operational regimes. CUNHA NETO (2005) also deeply studied this system in his work in order to verify its stationary behavior and to evaluate the nonlinearity.

The most notable results consists of the steady-state profiles while varying the spatial velocity, whose can be seen in Figures 3.2, 3.3, 3.4, 3.5, 3.6 and 3.7. The model parameters adopted in this work are based on ENGELL and KLATT (1993) and are listed in Table 3.1:

Table 3.1: Van de Vusse CSTR parameters. *Source*: ENGELL and KLATT (1993).

k_{10} (h^{-1})	1.287 x 10^{12}
k_{20} (h^{-1})	1.287 x 10^{12}
k_{30} (L/mol h)	9.043 x 10^{9}
E_1 (K)	-9.758.3
E_2 (K)	-9.758.3
E_3 (K)	-8560.0
-ΔH_1 (kJ/mol A)	-4.20
-ΔH_2 (kJ/mol B)	11.00
-ΔH_3 (kJ/mol C)	41.85
ρ (kg/L)	0.9342
c_p (kJ/kg K)	3.01
K_w (kJ/h K m^2)	4032.0
A_R (m^2)	0.215
V (L)	10

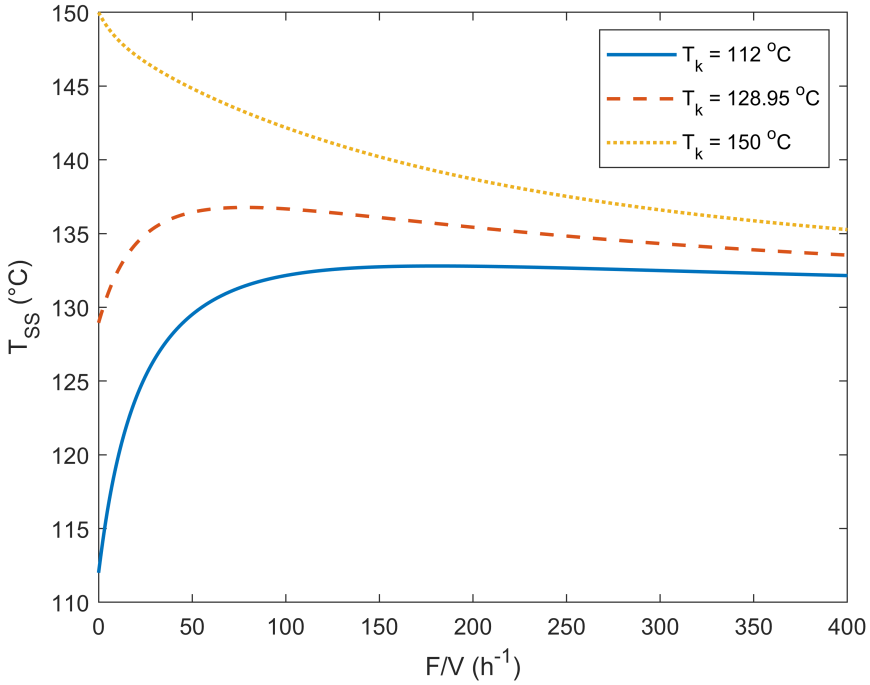

Figure 3.2: Stationary reactor temperature profiles as function of the spatial velocity different jacket temperatures with fixed inlet temperature and concentration of A (130 °C and 5.1 *mol / L*).

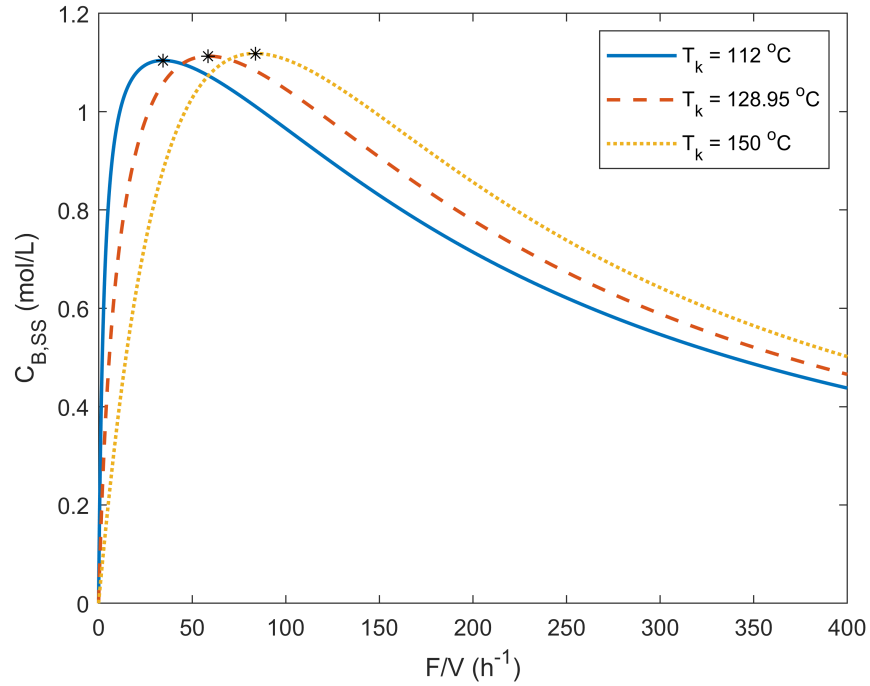

Figure 3.3: Stationary yield of B profiles as a function of the spatial velocity for different jacket temperatures with fixed inlet temperature and concentration of A (130 °C and 5.1 *mol / L*).

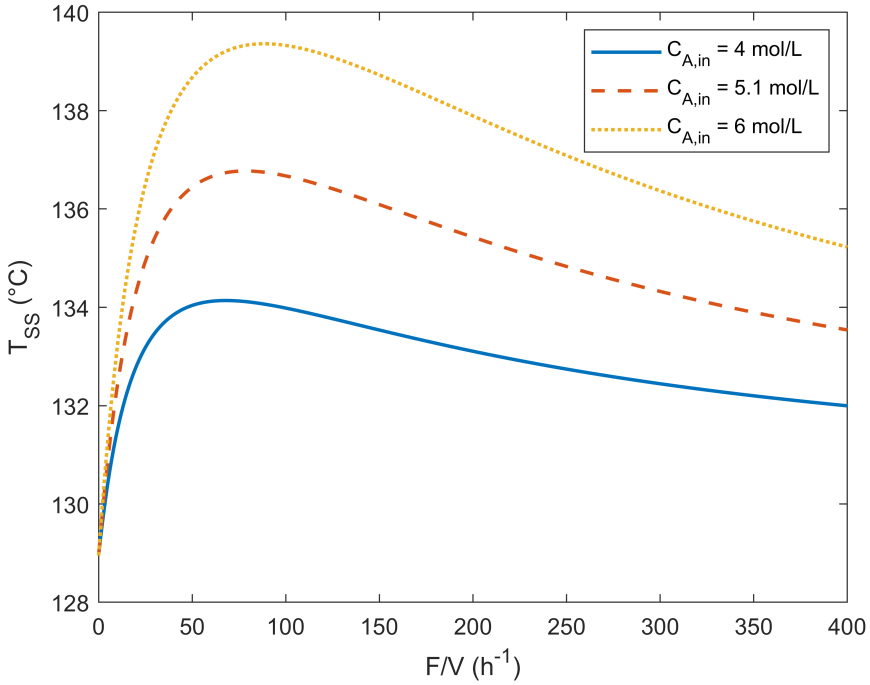

Figure 3.4: Stationary reactor temperature profiles as a function of the spatial velocity for different inlet concentrations of A with the jacket and inlet temperature fixed (128.95 $°C$ and 130 $°C$).

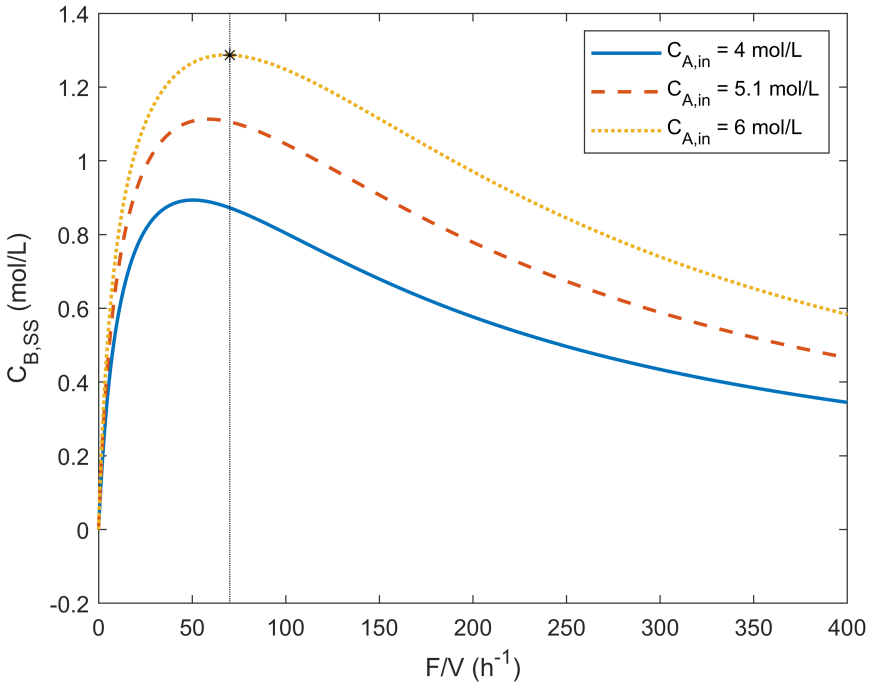

Figure 3.5: Stationary yield of B profiles as a function of the spatial velocity for different inlet concentrations of A with fixed jacket and inlet temperature (128.95 $°C$ and 130 $°C$).

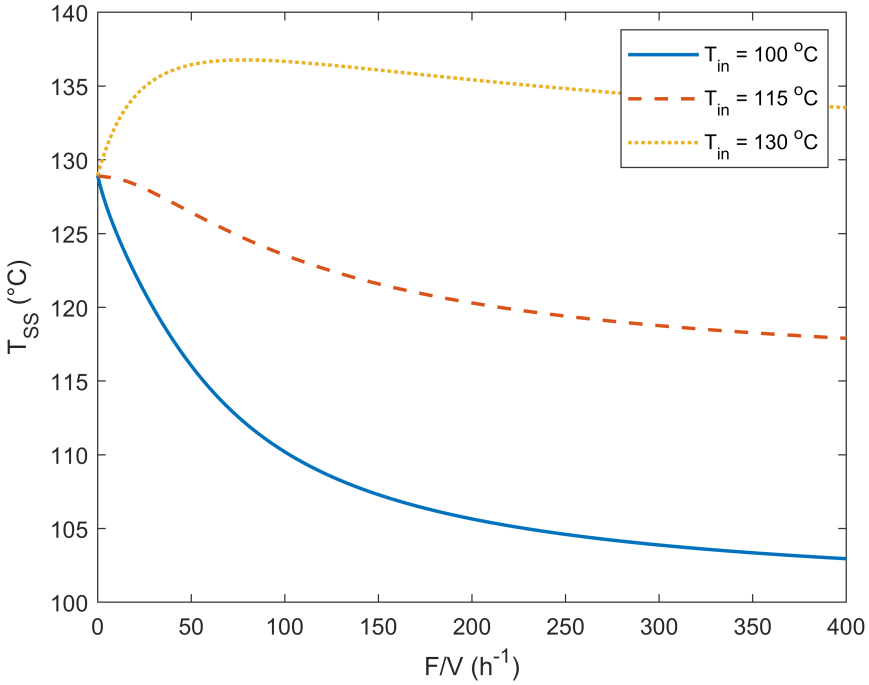

Figure 3.6: Stationary reactor temperature profiles as a function of the spatial velocity for different inlet temperatures with fixed jacket temperature and inlet concentration of A (128.95 $°C$ and 5.1 mol/L).

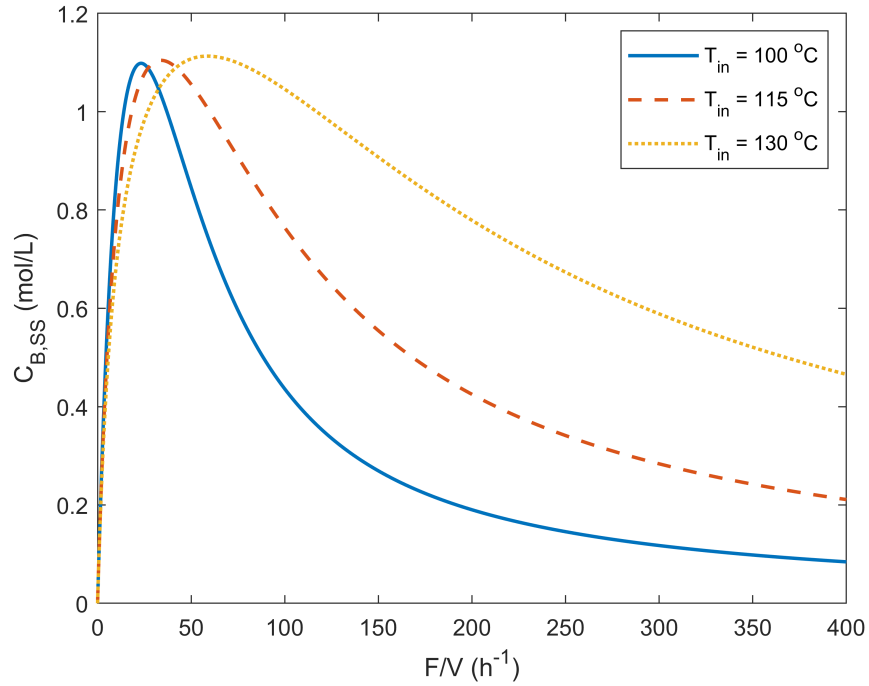

Figure 3.7: Stationary yield of B profiles as a function of the spatial velocity for different inlet temperatures with fixed jacket temperature and inlet concentration of A (128.95 $°C$ and 5.1 mol/L).

In Figures 3.2 and 3.3 it can be noted that, depending on the jacket temperature, the reactor temperature can assume ascending or descending profiles. Generally, for a jacket temperature greater than the inlet temperature, the reactor will cool down. This behavior highlights T_k as a good choice for manipulated variable once it has influence in T. Concerning the concentration of B, its optimum marginally increases with higher jacket temperatures, what will be better discussed in the optimization subsection of the methodology chapter.

In Figures 3.4 and 3.5, it is possible to note that a the process has a sign change in its gain at the optimal values for C_B and T and presents non-minimum phase behavior (inverse response) to the left of this optimum. Also, on the other hand, to the right of this peak the process shows minimum phase behavior with overshoots (MAGALHÃES, 2010). Thus, for operations with lower values of F/V it is likely that the process will contain complex dynamics features.

OGUNNAIKE and RAY (1994) point out several operational regimes for this particular process based on the stationary profile C_B *x* F/V. The authors also make recommendations regarding the process controllability in those operational ranges:

1. To the right of the maximum value for C_B, since the profile is almost linear, the process gain is fairly constant, with low-order simple dynamics. No advanced multivariable control technique is required, with two single-loop PI controllers being probably enough;

2. Near the optimum, the process goes through a gain change, wich represents the appearance of both minimum and non-minimum phase dynamics. Since the process has high degree of nonlinearity at this operational range, only a well designed multivariable nonlinear controller shall perform well;

3. To the left of the maximum, the process shows slight gain changes. The mild curve shows a low degree of nonlinearity and non-minimum phase, which represents difficult dynamics. PI controller's performance will be limited by the non-minimum phase behavior, with inverse response compensations being required.

By varying the inlet temperature, the optimal yield of B also slightly increases and the stationary profile behavior changes drastically, with the degree of nonlinearity being higher for lower values of T_{in}. For the reactor temperature, the changes are more notable since its profile shows a transition between a descending to ascending behavior, presenting higher degree of nonlinearity for lower values of F/V, as expected.

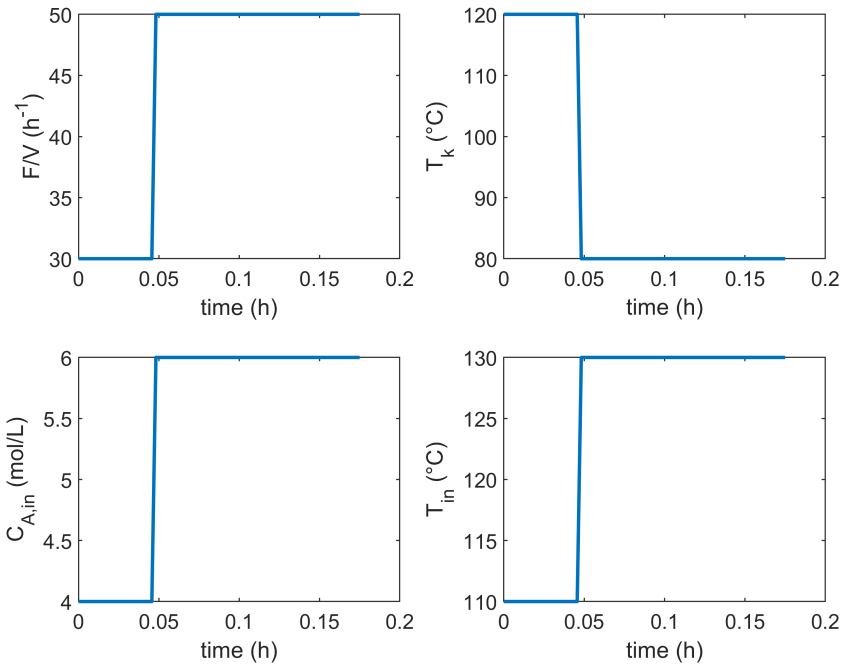

Figure 3.8: Perturbation applied.

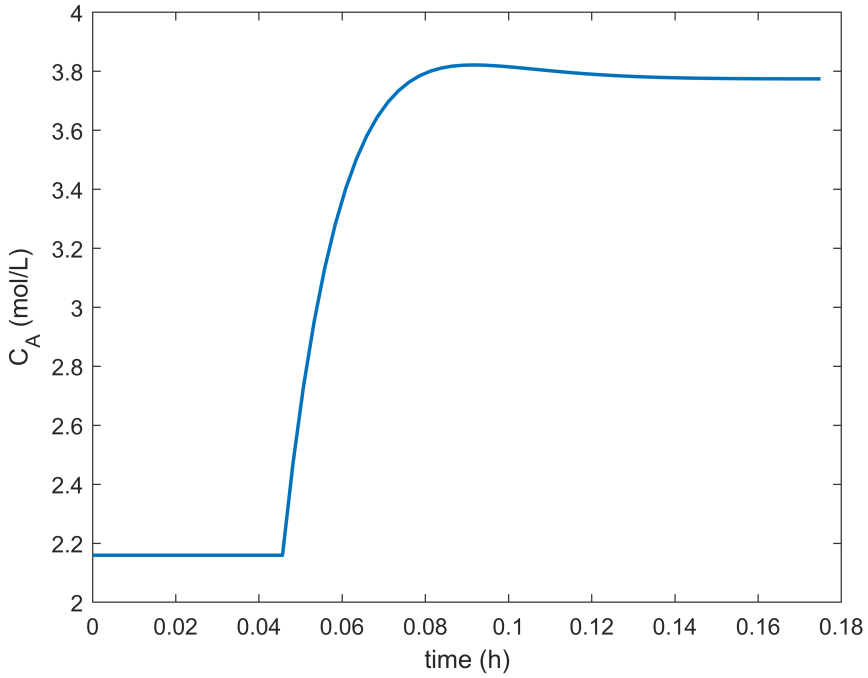

Figure 3.9: A concentration transient profile.

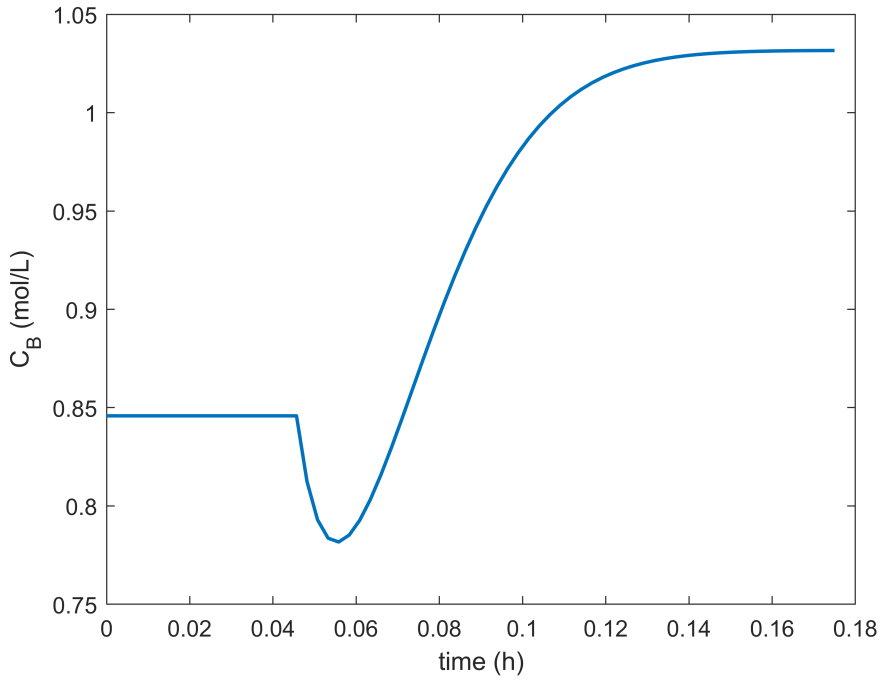

Figure 3.10: B concentration transient profile.

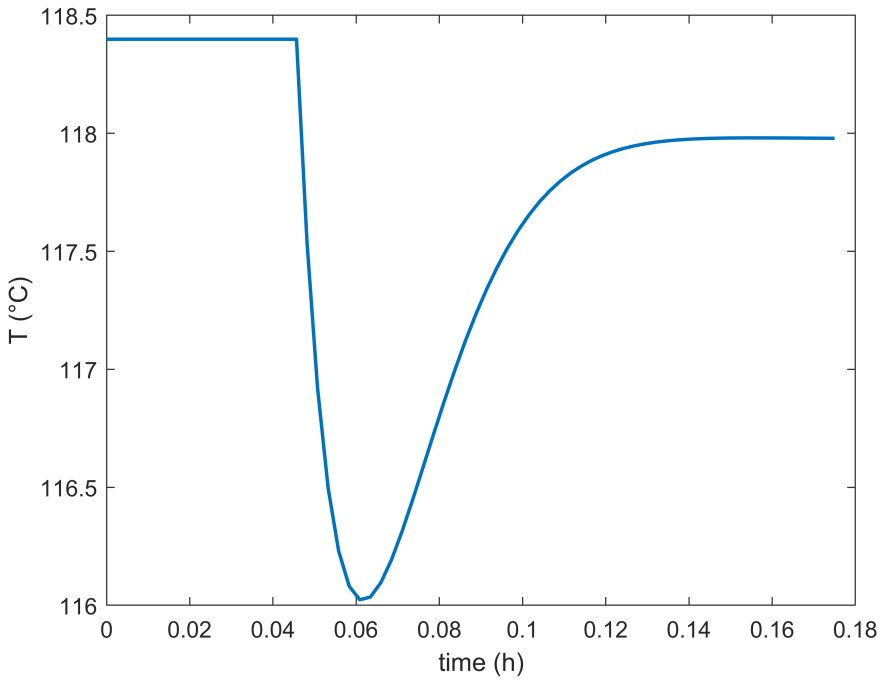

Figure 3.11: Reactor temperature transient profile.

In order to illustrate this complex dynamics, a transient simulation is performed for this system. Initially the system is at steady state and a perturbation is imposed around time $0.048h$ in F/V, T_k, $C_{A,in}$ and T_{in} is imposed (Figure 3.8)

and the state profiles are shown in Figures 3.9 to 3.11.

Figure 3.10 illustrates a *undershoot* for the T profile and Figure 3.9 clearly shows a inverse response for C_B, even though for a relatively short period o time. This rather fast dynamics highlights the use of small sampling times. Since the main goal is to evaluate the RTO approaches for systems when subject these complex behaviors, its of extreme relevance that plotting allows the observation of this kind of responses.

All the results regarding Van de Vusse CSTR dynamic behavior will be useful when defining operational ranges for multiple simulations in order to evaluate how each RTO method performs.

Chapter 4

Methodology

This chapter aims to detail the methods applied into this dissertation, going through the advanced control and online estimation layers, as well into the optimization steps and the real-time optimization frameworks as a whole.

In this work, the states considered are C_A, C_B and T; the inputs are F/V and T_k; and the disturbances are $C_{A,in}$ and T_{in}. The disturbances were considered as non-measured while performing the RTOs, and there are no plant-model mismatches in terms of the system parameters.

Looking for the addition of more realism into the simulations, white Gaussian noise of 0.1% was added to the disturbances and output measurements by the expression:

$$\begin{aligned} d &= d \times (1 + 0.001 \times randn()) \\ y &= y \times (1 + 0.001 \times randn()) \end{aligned} \tag{4.1}$$

where the $randn()$ function returns a random scalar drawn from the standard normal distribution.

All simulations developed in the course of this dissertation were carried out using Matlab R2018 scientific computing environment. The algorithmic differentiation package CasADi v3.0.1 was the main resource applied in order to implement all scripts. The optimization, estimation and control formulations were performed by the author on a 3.4GHz workstation with 8GB memory.

Also, the parameters used for the CSTR in this work are the ones from the work of ENGELL and KLATT (1993), from Table 3.1.

4.1 RTO, DRTO and HRTO frameworks

Two variations of RTO architecture were considered: the first one almost identical to the RTO scheme previously exposed in Chapter 2, but with the feedback of the estimated disturbances into the NMPC as well; and in the second variation the online estimation is performed by an EKF, besides the feedback of the estimated variables to the NMPC. The schema of both RTO algorithms can be seen in Figures 4.1 and 4.2.

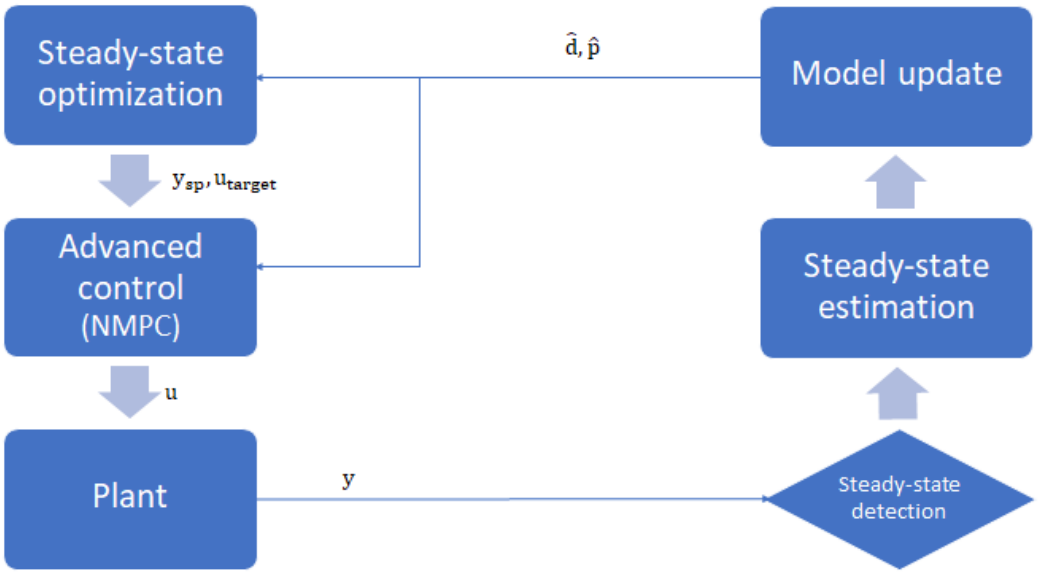

Figure 4.1: RTO architecture with a least-squares steady-state estimation step.

The reasoning behind feeding the controller with the estimated variables is simply for setpoint control purposes because the process disturbances contain uncertainties but, concerning the first RTO mentioned above, this feedback will only occur when steady states are detected. This RTO configuration will still use a steady-state least-squares estimator, thus performing only at steady states. When steady-state detection step does not determine the stationarity of the plant, the previously estimated disturbances will be fed into the NMPC. Lastly, this steady-state estimation layer only estimates the non-measured disturbances, thus the controller always receives the last states from the plant because the states are not estimated at this configuration, that is, assuming that all states are measured.

The problems with this structure are: since the non-measured disturbances are only estimated at steady-state, setpoint control is compromised when the process is not at steady-state due the fact the NMPC is still performing with the

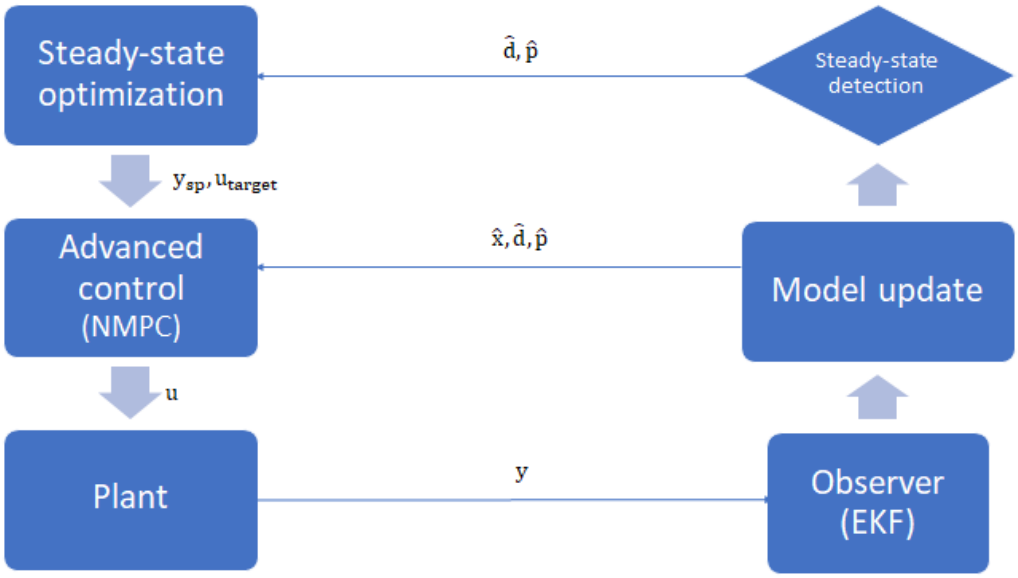

Figure 4.2: RTO architecture with an EKF.

last estimated disturbances. Also, for comparison purposes, the NMPC model set up should be the same for all RTOs. Therefore, for the second configuration, an EKF was designed instead of the steady-state estimator.

For this one, the estimation is performed at every sampling time, independent of the steady state, and the states will be estimated alongside the disturbances and these estimations will be used by the NMPC. The steady-state detection will only determine whether the stationary optimization step should occur. When the stationarity of the process is stated, then the optimization model is updated with the estimated disturbances.

For both dynamic and hybrid RTOs, the only difference is the use of the steady-state optimization for the later, not needing the estimated states, as it can be seen in Figures 4.3 and 4.4. In all RTOs performed, the optimization step is carried every 10 time steps, with the sampling time being 9 seconds and the plant (simulator) is solved by the IDAS integrator (HINDMARSH *et al.*, 2005).

4.2 Optimization problem formulation

4.2.1 Steady-state optimization

Regarding the optimization of the Van de Vusse CSTR, the literature mainly presents only two variations of cost functions, both intending to maximize the

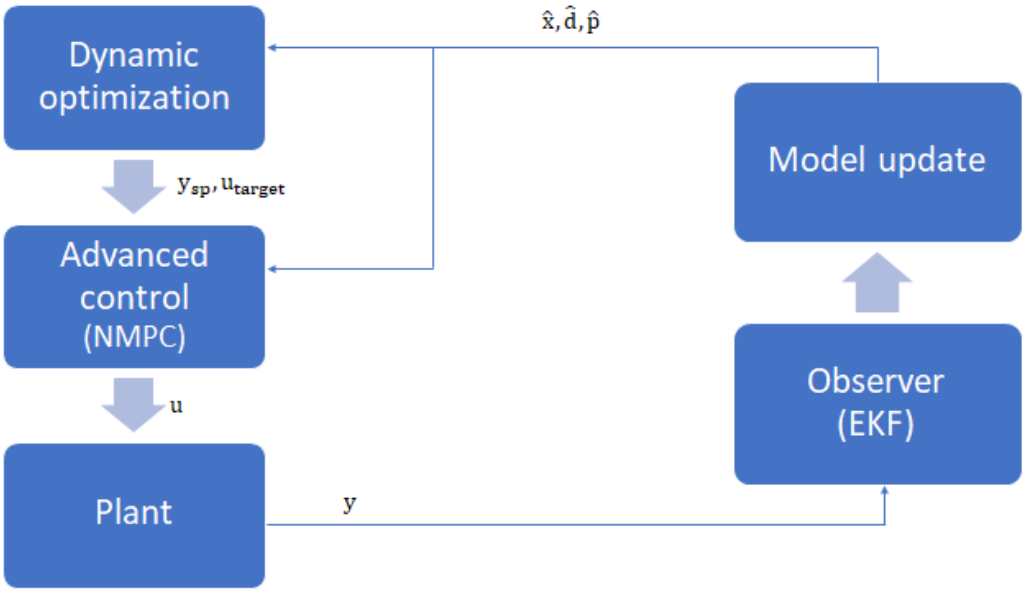

Figure 4.3: DRTO architecture.

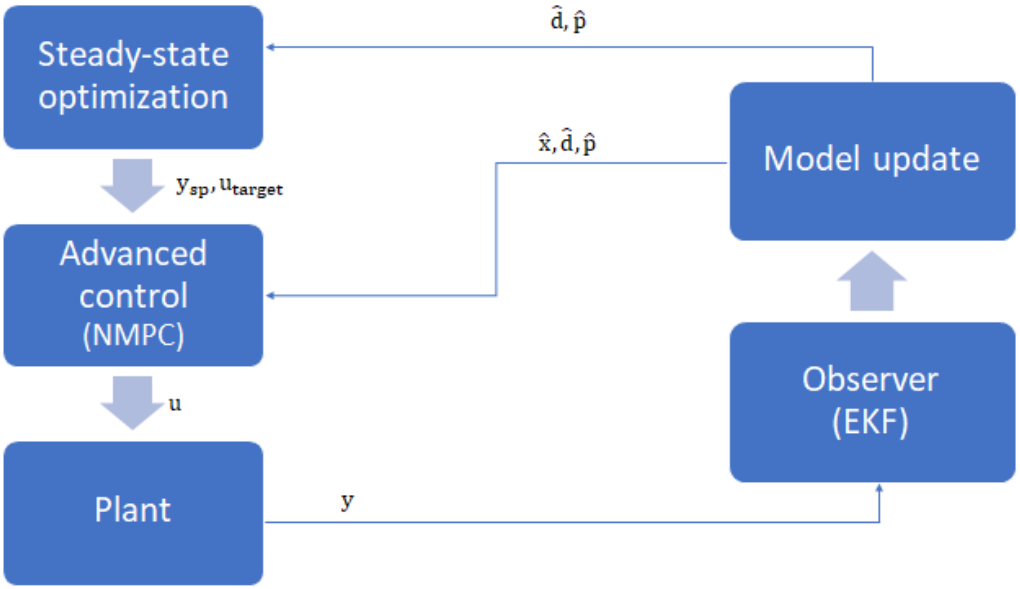

Figure 4.4: HRTO architecture.

desired product B. The most widely applied one is purely the yield of B, C_B; with the second being the sum of selectivity and conversion of B, $\frac{C_B}{C_{A,in}-C_A} + \frac{C_B}{C_{A,in}}$.

From Figure 3.3 it was shown that the concentration of B always raises

alongside the jacket temperature, which is a problem for the optimization. The first optimizations performed using this objective function always saturated the jacket temperature, i.e., it reached its upper bound quickly and the controller mostly manipulated the spatial velocity. In order to prevent this from happening, the same stationary profile was plotted for the selectivity-conversion objective function (Figure 4.5):

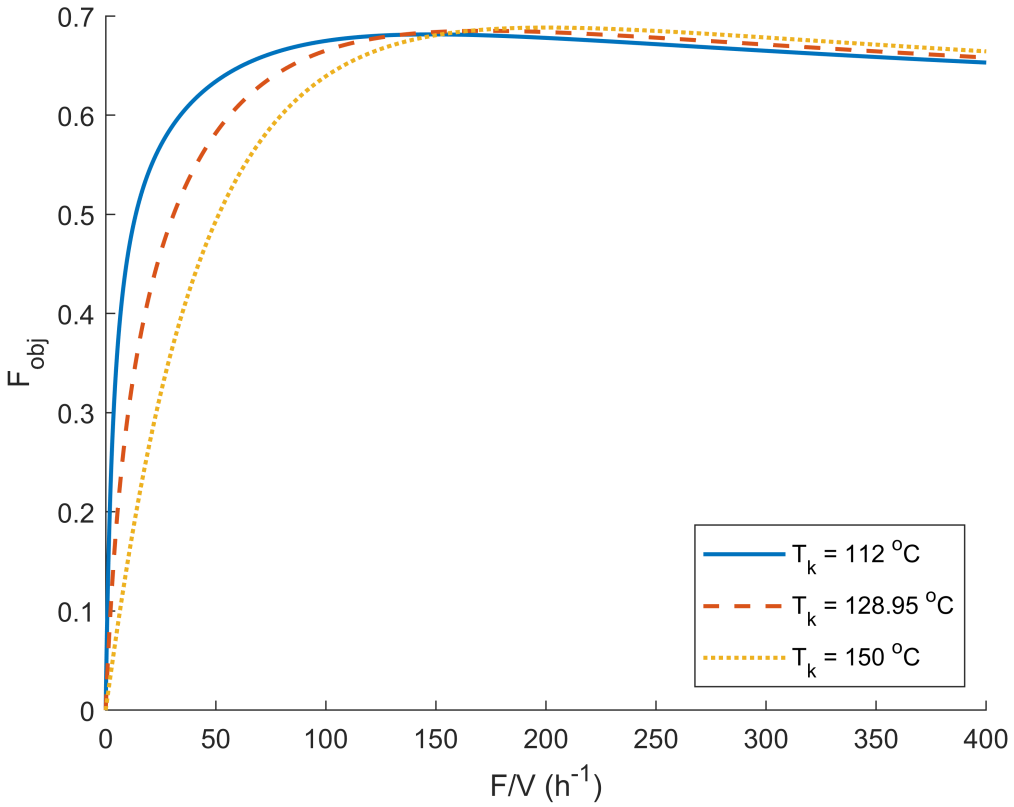

Figure 4.5: Selectivity plus conversion objective function.

It can be noted that the the same occurs when using this cost function, it raises with greater values of T_k and the optimizer will saturate it to its upper bound.

From this problem it was proposed a new variation of cost function for the Van de Vusse CSTR. Both these functions introduced earlier only take into account the productivity factor of the process, neglecting the energy cost related to higher jacket temperatures. This way, the objective function proposed simply the previous selectivity-conversion function applied with the addition of a penalty for higher values of T_k:

$$J = \frac{C_B}{C_{A,in} - C_A} + \frac{C_B}{C_{A,in}} - w\frac{T_k}{100} \tag{4.2}$$

It is well known that the main problem of multi-objective optimization's main problem is the conflict between the objective functions. The solution is not unique and depends deeply on the weight w. Thus, after performing some steady-state simulations, it was chosen $w = 0.06$ such that the objective function is optimal for intermediary values of jacket temperature.

Thus, the steady-state optimization problem formulated is given by:

$$max_{F/V,T_k} \frac{C_B}{C_{A,in} - C_A} + \frac{C_B}{C_{A,in}} - 0.06\frac{T_k}{100} \tag{4.3}$$

$$s.t. \quad \frac{F}{V}(C_{Ain} - C_A) - k_1(T)C_A - k_3(T)C_A^2 = 0$$

$$\frac{F}{V}C_B - k_1(T)C_A + k_2(T)C_B = 0$$

$$\frac{F}{V}(T_{in} - T) + \frac{K_w A_R}{\rho c_p V}(T_k - T) + \frac{1}{\rho c_p}\left[k_1(T)C_A(-\Delta H_1) + k_2(T)C_B(-\Delta H_2) + k_3(T)C_A^2(-\Delta H_3)\right] = 0$$

$$k_i(T) = k_{0i}\exp\left(-\frac{E_i}{T(K)}\right), \quad i = 1, 2, 3$$

$$0.1 \leq C_A \leq 5\ mol/L$$

$$0.1 \leq C_B \leq 2\ mol/L$$

$$30 \leq T \leq 200\ ^\circ C$$

$$10 \leq F/V \leq 400\ h^{-1}$$

$$30 \leq T_k \leq 200\ ^\circ C$$

This NLP was structured using CasADi and solved using IPOPT version 3.12.2., an interior-point open-source tool developed by the Department of Chemical Engineering of Carnegie Mellon University, Pittsburgh, PA, USA (WÄCHTER and BIEGLER, 2006).

4.2.2 Dynamic optimization

Similarly to the previous subsection, the dynamic optimization problem can be expressed as:

$$max_{F/V,T_k} \frac{C_B(t_N)}{C_{A,in} - C_A(t_N)} + \frac{C_B(t_N)}{C_{A,in}} - 0.06\frac{T_k(t_N)}{100} \tag{4.4}$$

$$s.t. \quad \frac{dC_A}{dt} - \frac{F}{V}(C_{Ain} - C_A) + k_1(T)C_A + k_3(T)C_A^2 = 0$$

$$\frac{dC_B}{dt} + \frac{F}{V}C_B - k_1(T)C_A + k_2(T)C_B = 0$$

$$\frac{dT}{dt} - \frac{F}{V}(T_{in} - T) - \frac{K_w A_R}{\rho c_p V}(T_k - T) -$$

$$\frac{1}{\rho c_p}\left[k_1(T)C_A(-\Delta H_1) + k_2(T)C_B(-\Delta H_2) + k_3(T)C_A^2(-\Delta H_3)\right] = 0$$

$$k_i(T) = k_{0i}\exp\left(-\frac{E_i}{T(K)}\right), \quad i = 1, 2, 3$$

$$0.1 \leq C_A \leq 5\ mol/L$$

$$0.1 \leq C_B \leq 2\ mol/L$$

$$30 \leq T \leq 200\ ^\circ C$$

$$10 \leq F/V \leq 400\ h^{-1}$$

$$30 \leq T_k \leq 200\ ^\circ C$$

The cost function was chosen to be its value at the end of the horizon for comparison purposes. It is expected that the RTO, DRTO and HRTO present the same steady state but mainly diverging in transient operation. The prediction horizon N was set to 40 time steps and the control horizon M to 10. This means that the inputs and states will be calculated through the whole horizon of 40 but the inputs are set to be constant after 10 control actions.

The DRTO was also structured using CasADi and the resulting NLP is solved with IPOPT. In order to implement this problem, it was applied a third-order direct collocation scheme for discretization.

4.3 Steady-state detection

The steady-state detection step, necessary for the standard RTO, was based on the use of maximum covariance when the process is at the steady state. Several open-loop simulations were performed in the presence of process and measurement noise and the maximum covariance for each state while at steady state was estimated. Thus, the algorithm is quite simple: the covariance is measured at every time step and if the detection finds it bellow this maximum covariance for all states, this process is said to be at steady-state operation. Each sampling is composed of measurements from the last five time steps.

4.4 NMPC

As exposed earlier in this work, a classic MPC should not be able to perform well in operational ranges that present complex nonlinear dynamics. This fact alone encourages the use of an NMPC.

The NMPC dynamic optimization problem was built in similar fashion to the optimization problem of the previous section:

$$min_{F/V,T_k}\ q_1 \sum_{i=1}^{N} \left(C_{B,k+i} - C_{B,k+i}^{SP}\right)^2 + q_2 \sum_{i=1}^{N} \left(T_{k+i} - T_{k+i}^{SP}\right)^2 + \tag{4.5}$$

$$w_1 \sum_{i=0}^{M-1} \left(F/V_{k+i} - F/V_{k+i}^{target}\right)^2 + w_2 \sum_{i=0}^{M-1} \left(T_{k,k+i} - T_{k,k+i}^{target}\right)^2 +$$

$$s_1 \sum_{i=0}^{M-1} \left(\Delta F/V_{k+i}\right)^2 + s_2 \sum_{i=0}^{M-1} \left(\Delta T_{k,k+i}\right)^2$$

$$s.t. \quad \frac{dC_A}{dt} - \frac{F}{V}(C_{Ain} - C_A) + k_1(T)C_A + k_3(T)C_A^2 = 0$$

$$\frac{dC_B}{dt} + \frac{F}{V}C_B - k_1(T)C_A + k_2(T)C_B = 0$$

$$\frac{dT}{dt} - \frac{F}{V}(T_{in} - T) - \frac{K_w A_R}{\rho c_p V}(T_k - T) -$$

$$\frac{1}{\rho c_p}\left[k_1(T)C_A(-\Delta H_1) + k_2(T)C_B(-\Delta H_2) + k_3(T)C_A^2(-\Delta H_3)\right] = 0$$

$$k_i(T) = k_{0i}\exp\left(-\frac{E_i}{T(K)}\right), \quad i = 1, 2, 3$$

$$0.1 \leq C_A \leq 5\ mol/L$$

$$0.1 \leq C_B \leq 2\ mol/L$$

$$30 \leq T \leq 200\ ^\circ C$$

$$10 \leq F/V \leq 400\ h^{-1}$$

$$30 \leq T_k \leq 200\ ^\circ C$$

$$-50 \leq \Delta F/V \leq 50\ h^{-1}$$

$$-10 \leq \Delta T_k \leq 10\ ^\circ C$$

where q_i, w_i and s_i are the respective weights from the Q, W and S matrices presented in the MPC section in Chapter 2.

4.5 Estimators

4.5.1 Static estimator

For one of the standard RTOs a simple weighted least-squares estimator is applied. This step was only responsible for estimating disturbances at steady state and the estimation problem formulation is given as follows:

$$min_{C_{Ain},T_{in}} \ (y_{meas} - x_{SS})^T \ W \ (y_{meas} - x_{SS}) \tag{4.6}$$

$$s.t. \quad \frac{F}{V}(C_{Ain} - C_{A,SS}) - k_1(T)C_{A,SS} - k_3(T)C_{A,SS}^2 = 0$$

$$\frac{F}{V}C_{B,SS} - k_1(T)C_{A,SS} + k_2(T)C_{B,SS} = 0$$

$$\frac{F}{V}(T_{in} - T_{,SS}) + \frac{K_w A_R}{\rho c_p V}(T_k - T_{,SS}) +$$

$$\frac{1}{\rho c_p}\left[k_1(T)C_{A,SS}(-\Delta H_1) + k_2(T)C_{B,SS}(-\Delta H_2) + k_3(T)C_{A,SS}^2(-\Delta H_3)\right] = 0$$

$$k_i(T) = k_{0i}\exp\left(-\frac{E_i}{T(K)}\right), \quad i = 1, 2, 3$$

$$0.1 \leq C_{Ain} \leq 10 \ mol/L$$

$$90 \leq T_{in} \leq 150 \ ^\circ C$$

where W is the weight square matrix, tuned as $\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$ in this case.

4.5.2 Dynamic state-parameter estimation using EKF

As stated previously, uncertainties regarding the non-measured disturbances and the measurements are considered. The formulation that follows was based on the work of KRISHNAMOORTHY (2019):

$$d_{k+1} = d_k + w_{d,k} \tag{4.7}$$

$$y_k = h(x_k, u_k) + v_k \tag{4.8}$$

$$w_{d,k} \sim (0, Q_{d,k})$$

$$v_k \sim (0, R_k)$$

Since the goal is to estimate both states and disturbances, the extended EKF system is given by:

$$x^*_{k+1} = \begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} = f^*(x^*_k, u_k) + w^*_k \tag{4.9}$$

$$y_k = \begin{bmatrix} h(x_k, u_k) & 0 \end{bmatrix} x^*_k + v_k$$

where:

$$f^*(x^*_k, u_k) = \begin{bmatrix} f(x^*_k, u_k) + w_k \\ d_k + w_{d,k} \end{bmatrix}$$

$$w_k \sim (0, Q_k)$$

Therefore, the discrete-time extended Kalman filter for this system can be written as:

$$\hat{x}^{*-}_k = f(\hat{x}^{*+}_{k-1}, u_{k-1}) \tag{4.10}$$

$$P^-_k = F_{k-1} P^+_{k-1} F^T_{k-1} + Q^*_{k+1} \tag{4.11}$$

$$K_K = P^-_k H^T_k (H_k P^-_k H^T_k + R_k)^{-1} \tag{4.12}$$

$$z_k = h(\hat{x}_k^-, 0) - H_k \hat{x}_k^- \tag{4.13}$$

$$\begin{aligned} \hat{x}_k^{*+} &= \hat{x}_k^{*-} + K_k(y_k - H_k \hat{x}_k^- - z_k) \\ &= \hat{x}_k^{*-} + K_k(y_k - h(\hat{x}_k^-, u_k)) \end{aligned} \tag{4.14}$$

$$P_k^+ = (I - K_k H_k) P_k^- \tag{4.15}$$

where:

$$F = \left. \frac{\partial f(x^*, u)}{\partial x^*} \right|_{x^* = \hat{x}^*} \tag{4.16}$$

$$H = \left. \frac{\partial h(x, u)}{\partial x^*} \right|_{x^* = \hat{x}^*} \tag{4.17}$$

$$Q^* = \begin{bmatrix} Q & 0 \\ 0 & Q_d \end{bmatrix} \tag{4.18}$$

The filter tuning ended with the given noise covariance matrices:

$$Q = \begin{bmatrix} 0.03 & 0 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{bmatrix} \tag{4.19}$$

$$R = \begin{bmatrix} 0.03 & 0 & 0 \\ 0 & 0.005 & 0 \\ 0 & 0 & 0.8 \end{bmatrix} \tag{4.20}$$

and the covariance matrix for the estimation error is initialized as:

$$P_k = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \tag{4.21}$$

Chapter 5

Results

5.1 Closed-loop simulation

In this work, a nonlinear MPC whose internal model is the own Van de Vusse model is applied. Since the transient regions are the point of interest when comparing the RTOs, the main goal behind this choice is to represent the dynamic behavior of the system well enough.

After the model is implemented, a discretization step follows. This is not particularly needed since there are continuous-time Kalman filter formulations (SIMON, 2006), but the use of continuous-time models makes it necessary to integrate the system equations. Also, the use of discrete-time systems is encouraged when designing MPCs which require recursive model evaluations (BERNARDINO, 2019).

Figures 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 and 5.8 present some closed loop simulations. For now, two simulations were performed: the first one assuming the disturbances as measured parameters (perfect model) and the second one for non-measured disturbances. As stated previously, C_B and T are controlled, F/V and T_k are manipulated and $C_{A,in}$ and T_{in} are discturbances.

Since these simulations do not present the real-time optimization step, the setpoints were fixed and null weight was applied to the targets terms in the cost function, i.e., this NMPC takes into account only the setpoints and control action moves. The perturbations and the exact same time they were imposed, which is halfway through the simulation can be seen in the subplots in Figure 5.4. The setpoint and controller parameters used are presented in Table 5.1.

Table 5.1: NMPC parameters - closed loop simulation.

q_1	q_2	s_1	s_2	N	M	C_B^{SP}	T^{SP}
10^{-3}	1	10^{-8}	10^{-5}	40	10	0.9 mol/L	120 $^{\circ}C$

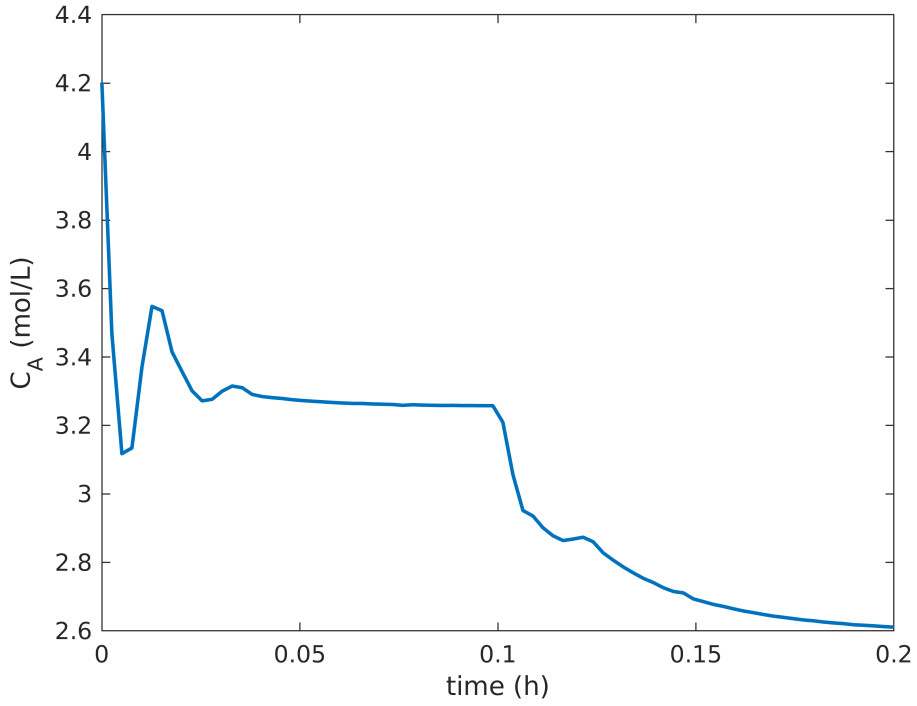

Figure 5.1: NMPC simulation with measured disturbances - concentration of A.

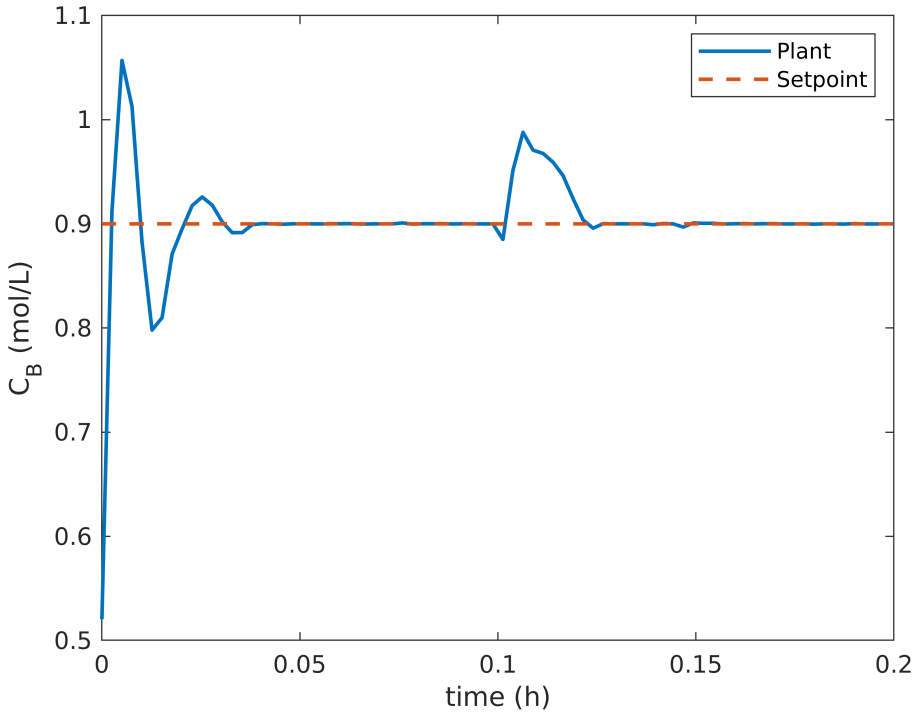

Figure 5.2: NMPC simulation with measured disturbances - concentration of B.

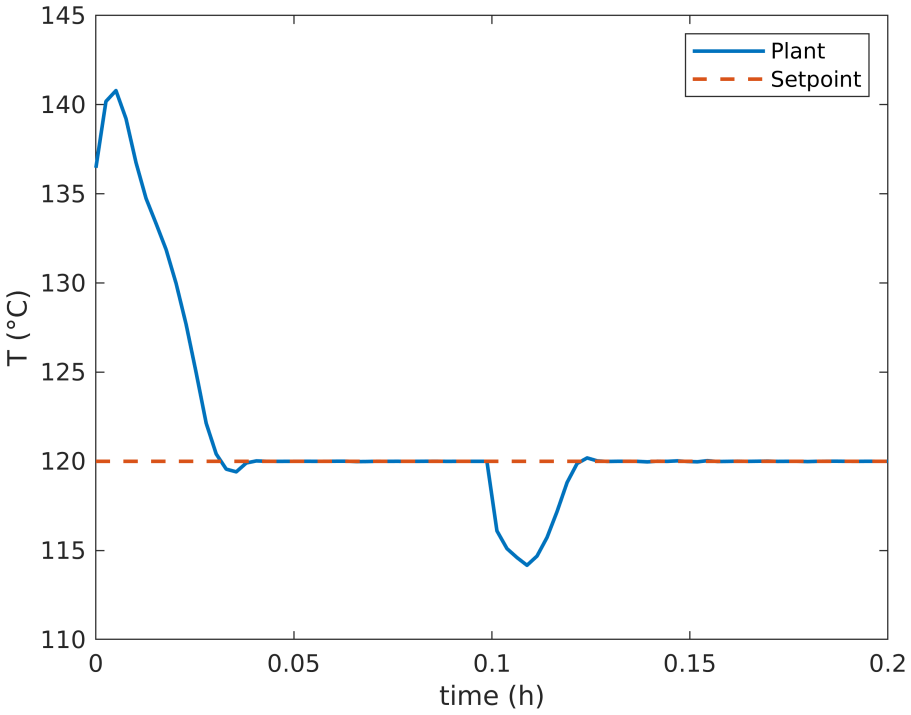

Figure 5.3: NMPC simulation with measured disturbances - reactor temperature.

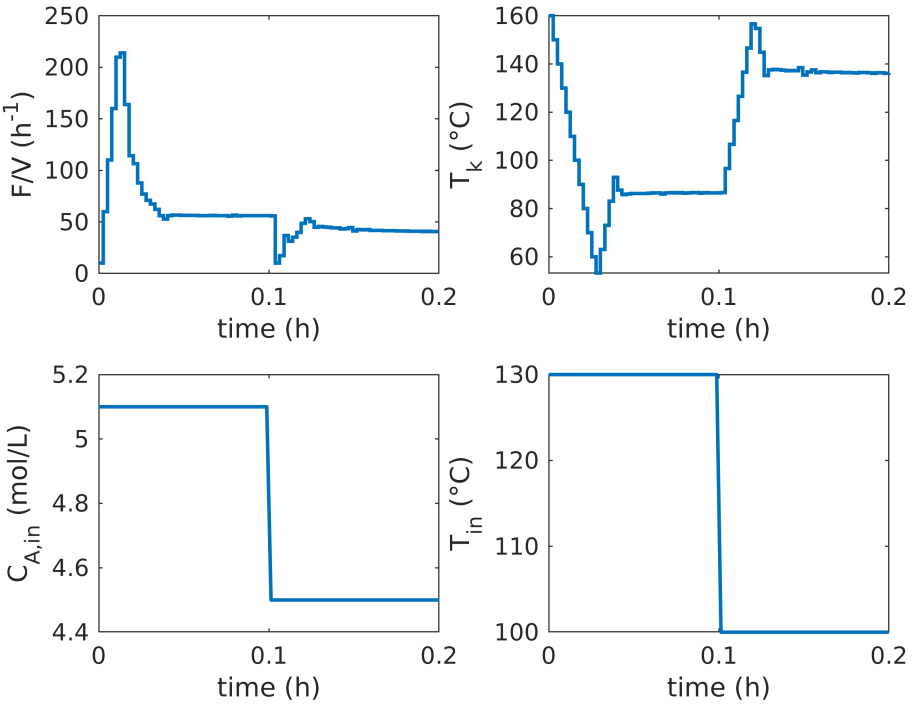

Figure 5.4: NMPC simulation with measured disturbances - inputs and disturbances.

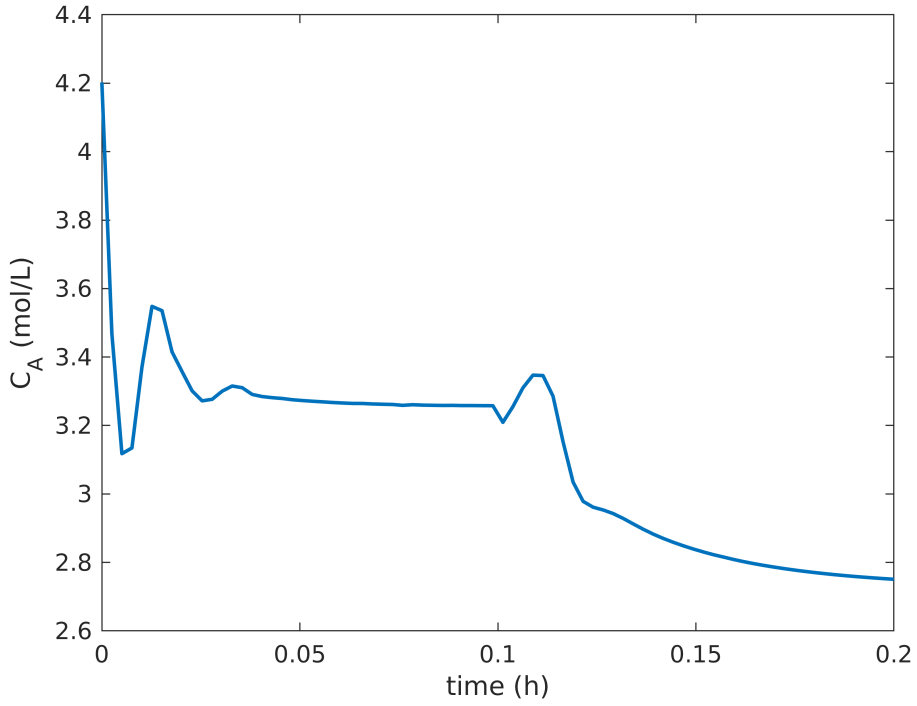

Figure 5.5: NMPC simulation with nonmeasured disturbances - concentration of A.

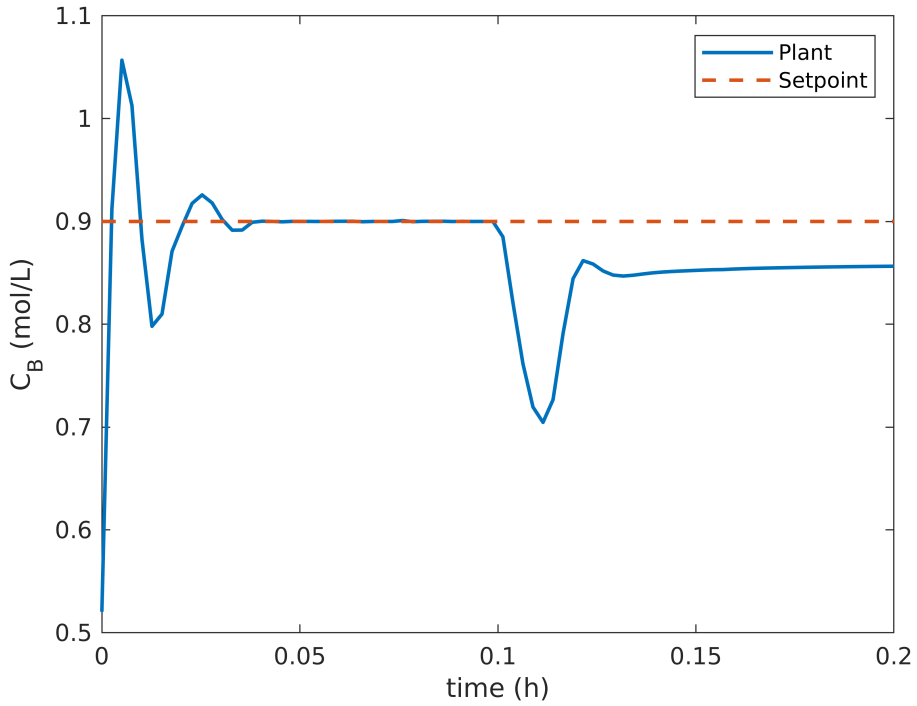

Figure 5.6: NMPC simulation with nonmeasured disturbances - concentration of B.

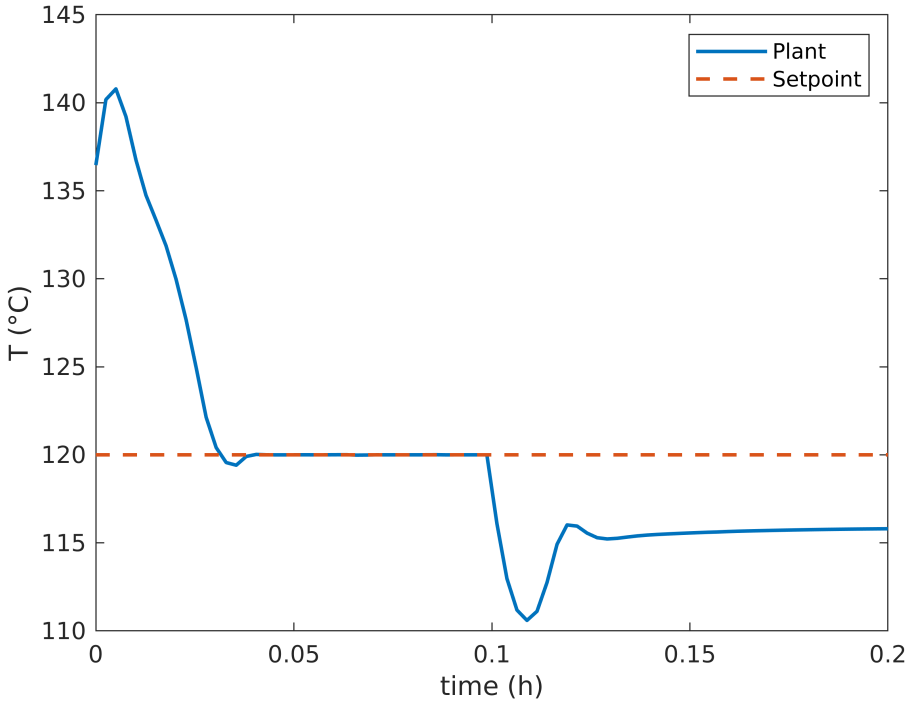

Figure 5.7: NMPC simulation with nonmeasured disturbances - reactor temperature.

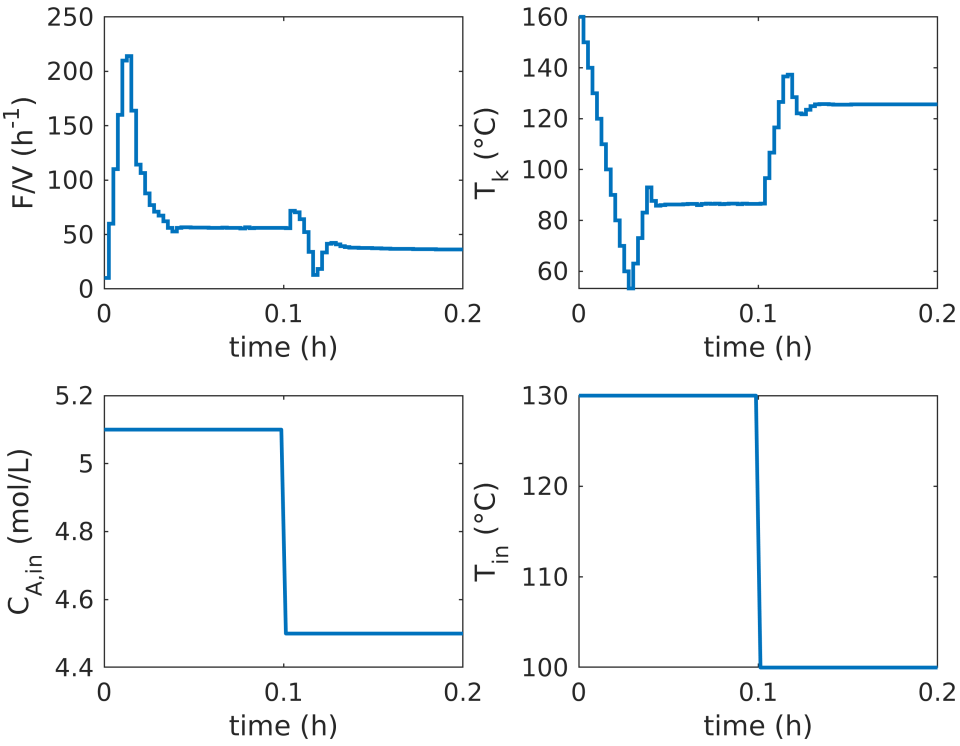

Figure 5.8: NMPC simulation with nonmeasured disturbances - inputs and disturbances.

The weights were tuned on a try and error basis based on the magnitude of the variables and on the relevance of each term as well.

The plant settles around time 0.05 h. Since this generates near 20 time steps, the prediction horizon chosen of 40 is enough. In both cases no process noise was considered.

The second simulating highlights the importance of an online estimator for cases where uncertanties play a role. The Kalman filter not only estimates the uncertain states but also is extremely important for model update throughout the simulation. Figures 5.6 and 5.7 show that the controller is deeply dependant on its internal model correction in order to reach the given setpoint. This particular result exposes the importance of correct online state-parameter estimation in process control under uncertanties.

The simulation of Figures 5.9, 5.10, 5.11 and 5.12, which presents EKF correcting the states and non-measured disturbances in the presence of process noise.

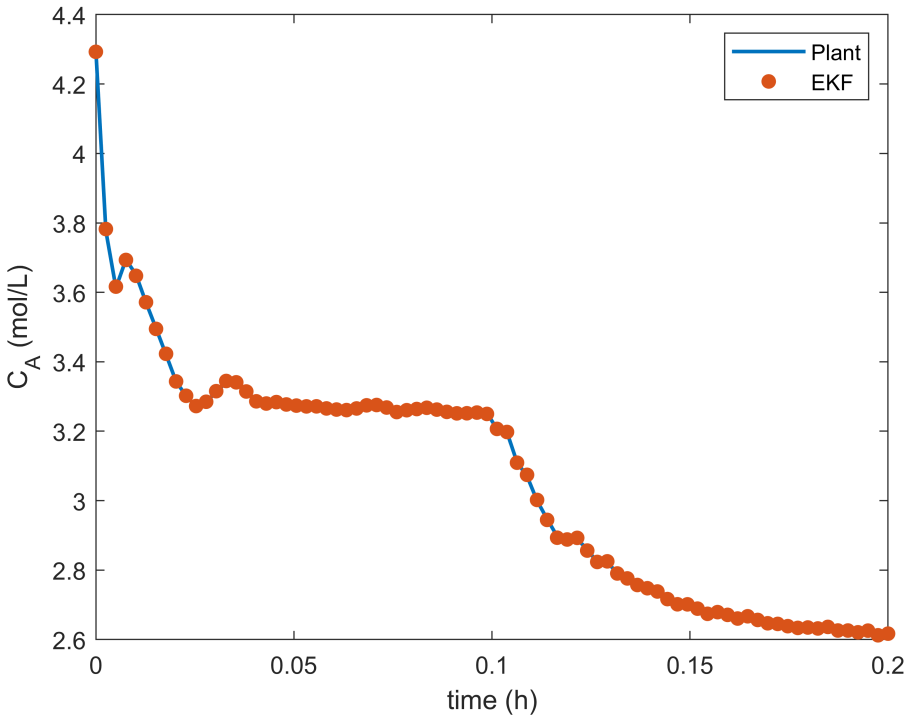

Figure 5.9: NMPC-EKF simulation with nonmeasured disturbances - concentration of A.

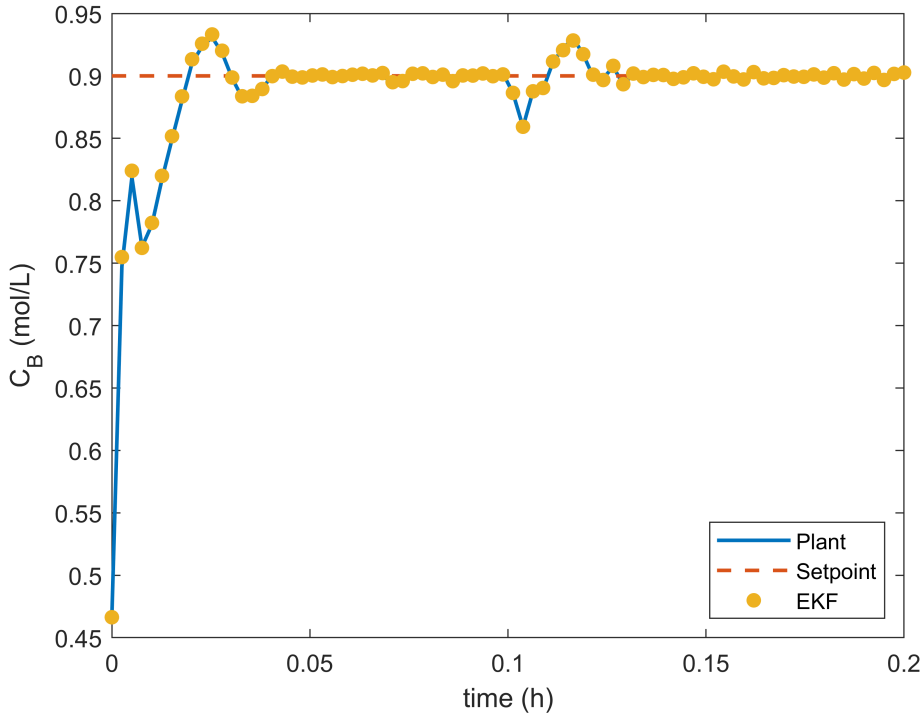

Figure 5.10: NMPC-EKF simulation with nonmeasured disturbances - concentration of B.

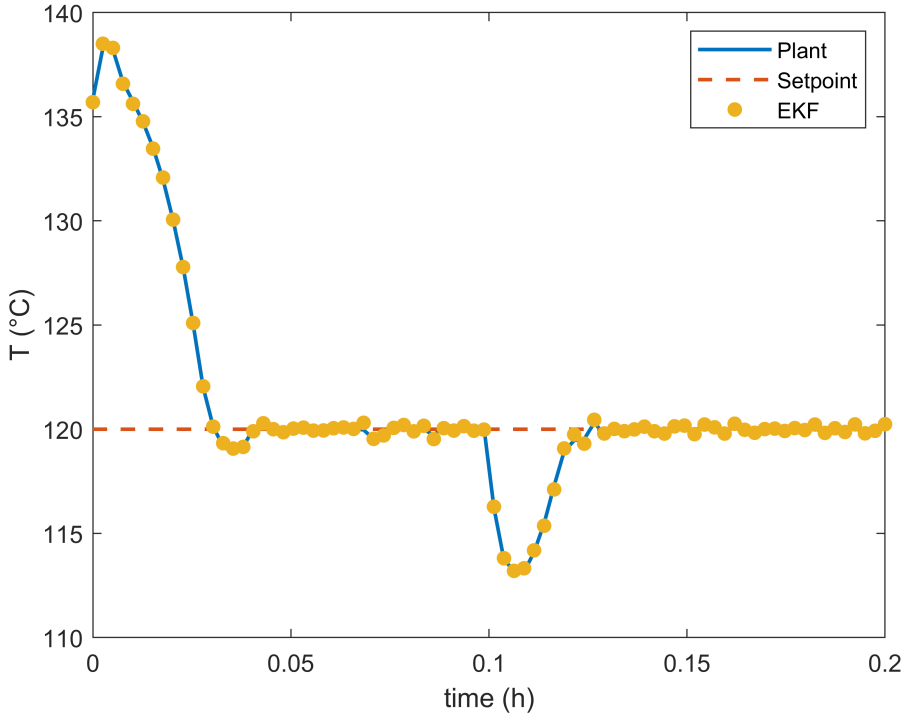

Figure 5.11: NMPC-EKF simulation with nonmeasured disturbances - reactor temperature.

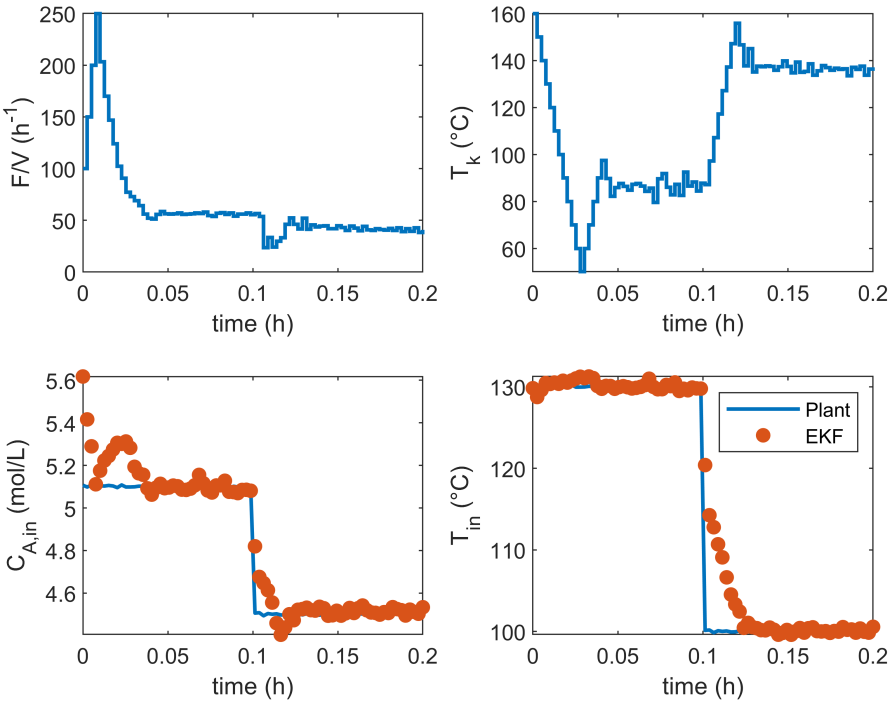

Figure 5.12: NMPC-EKF simulation with nonmeasured disturbances - inputs and disturbances.

It is possible to observe that the EKF corrects the disturbance after its change halfway through the simulation, feeding the estimated value to the NMPC, which leaves no offset at this point.

5.2 Direct collocation quadrature

The quadrature implemented for the NMPC and dynamic optimization makes use of 3 collocation points for each interval in the discretization and Legendre polynomials approximation.

A open-loop simulation was performed in order to evaluate the quadrature's performance and the results are shown in Figure 5.13.

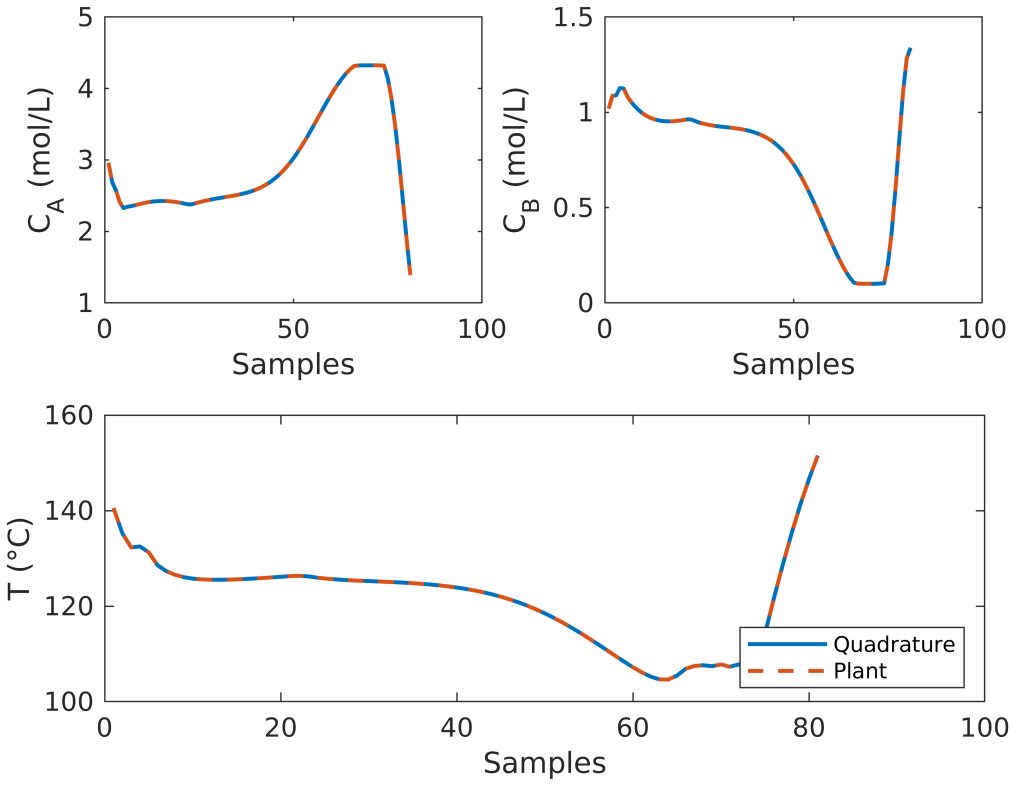

Figure 5.13: Quadrature validation.

5.3 Dynamic RTO

The DRTO was applied in this work with the intention of generating a reference, which will be used to quantify the performance of the RTO and HRTO.

In order to accomplish this feature, the DRTO simulations were performed in way that the dynamic optimization only acts at the beginning of the simulation, i.e., the setpoint and target trajectories are determined only once. The idea is to use the DRTO cost function performance as a reference tool to compare the remaining approaches, since DRTO algorithms are still too costly to be applied in industry.

The DRTO simulations start from the steady-state point of operation illustrated in Table 5.2 and disturbances are performed at $t = 0$. Thus, the RTO and HRTO simulations shall present the same disturbances, otherwise it would be pointless to compare them. The main difference is that the RTO simulations will be performed for multiple optimization cycles.

These DRTO reference results also were performed in an ideal scenario of no process noise and the exact conditions are illustrated in Table 5.3.

Table 5.2: RTO simulation steady-state start operation.

$C_A(mol/L)$	$C_B(mol/L)$	$T(^\circ C)$
3.65	0.755	125.15

Table 5.3: DRTO simulation scenarios.

Scenario	$C_{A,in}(mol/h)$	$T_{in}(^\circ C)$	Simulation time (h)	Disturbance instant (h)
1	4.5	130	0.2	0
2	5.1	110	0.2	0
3	4.5	110	0.2	0

For the NMPC, the target deviation term will be considered and its parameters are shown in Table 5.4.

Table 5.4: NMPC parameters - RTO simulations.

q_1	q_2	w_1	w_2	s_1	s_2	N	M
10^{-3}	1	10^{-8}	10^{-5}	10^{-8}	10^{-5}	40	10

The DRTO results for each scenario are present in Figures 5.14 to 5.25.

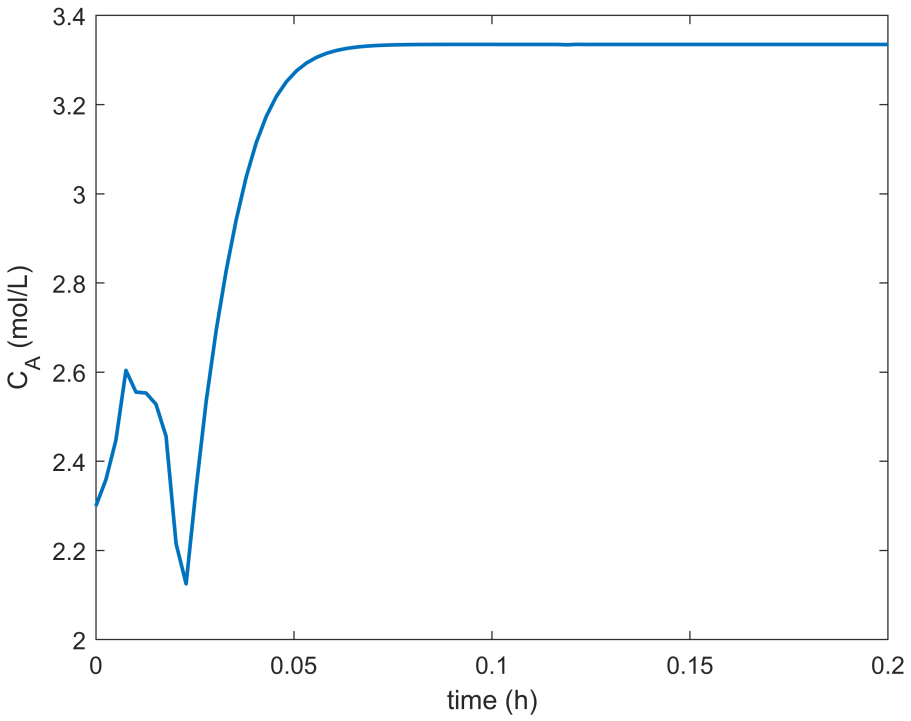

Figure 5.14: DRTO simulation scenario 1 - concentration of A.

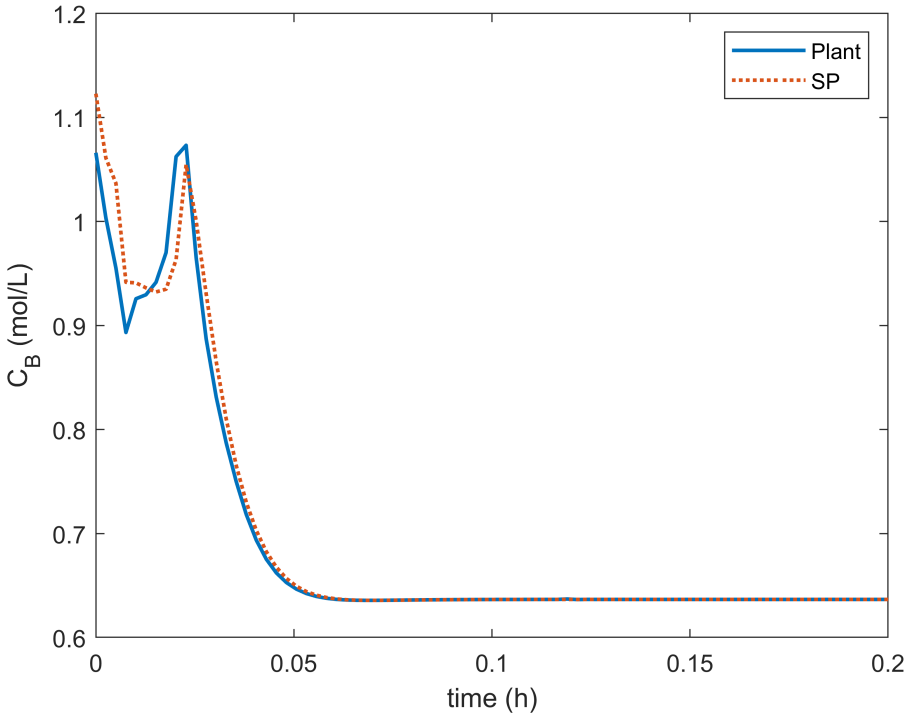

Figure 5.15: DRTO simulation scenario 1 - concentration of B.

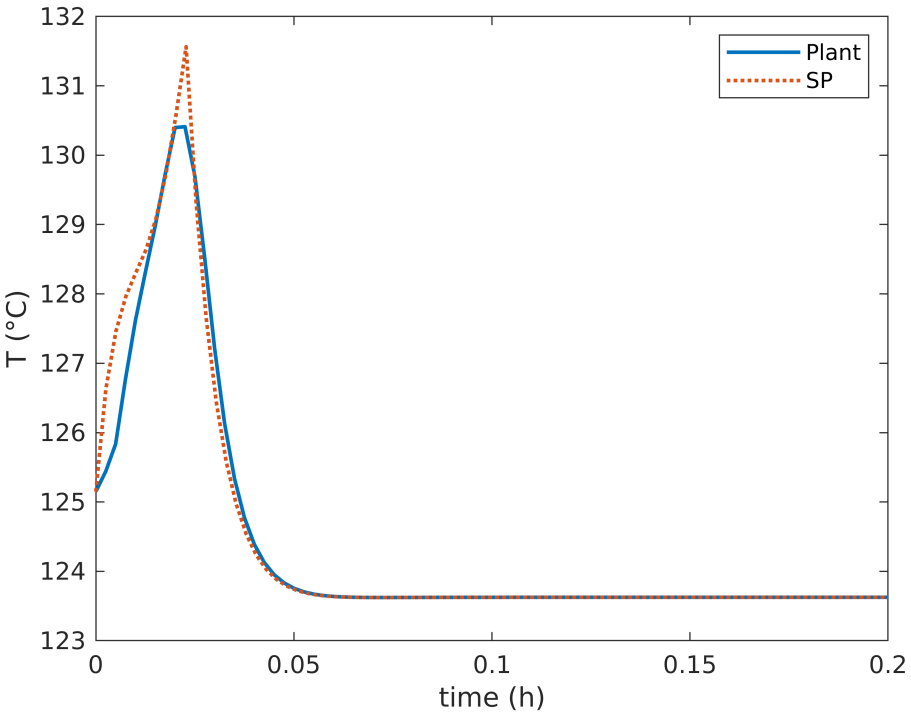

Figure 5.16: DRTO simulation scenario 1 - reactor temperature.

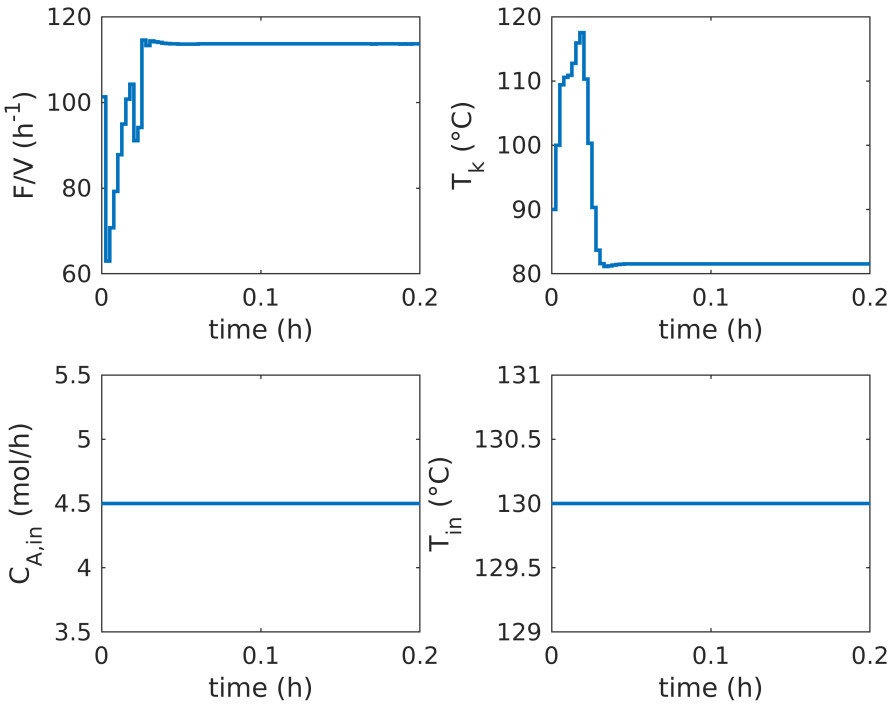

Figure 5.17: DRTO simulation scenario 1 - inputs and disturbances.

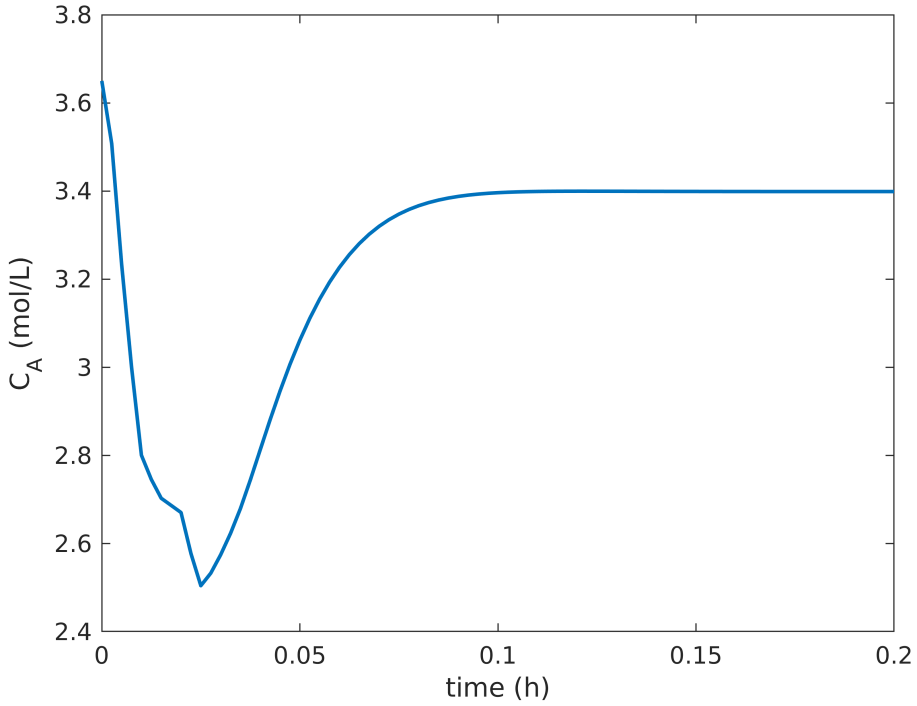

Figure 5.18: DRTO simulation scenario 2 - concentration of A.

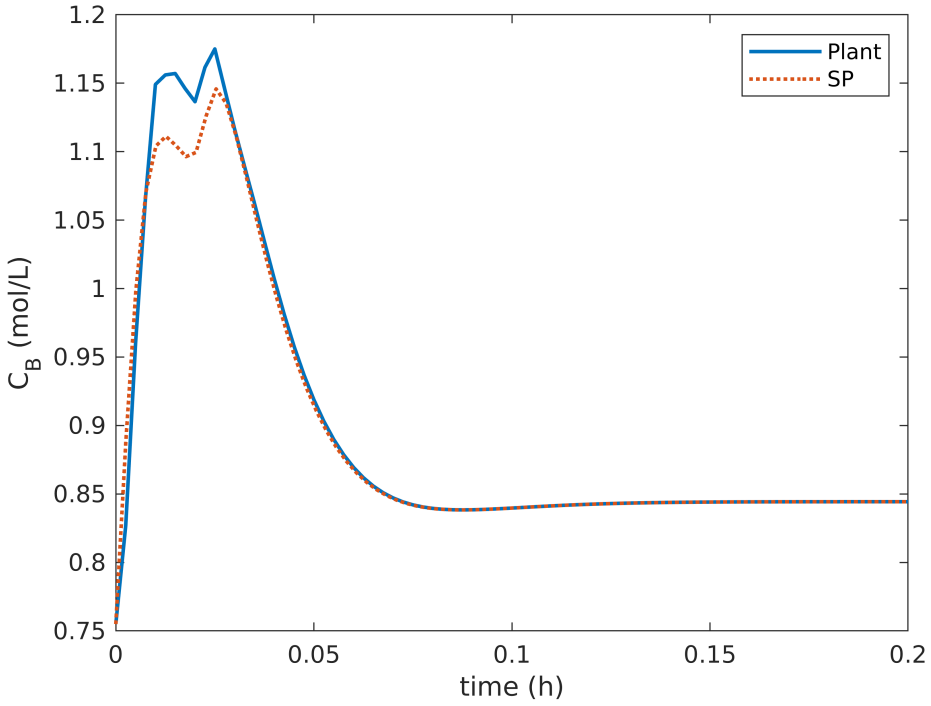

Figure 5.19: DRTO simulation scenario 2 - concentration of B.

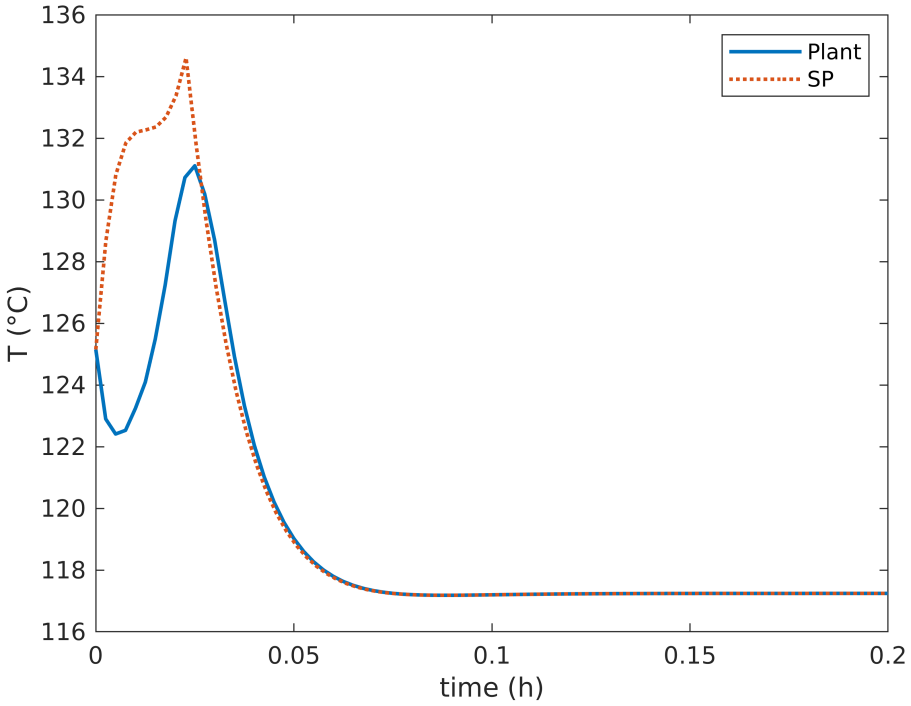

Figure 5.20: DRTO simulation scenario 2 - reactor temperature.

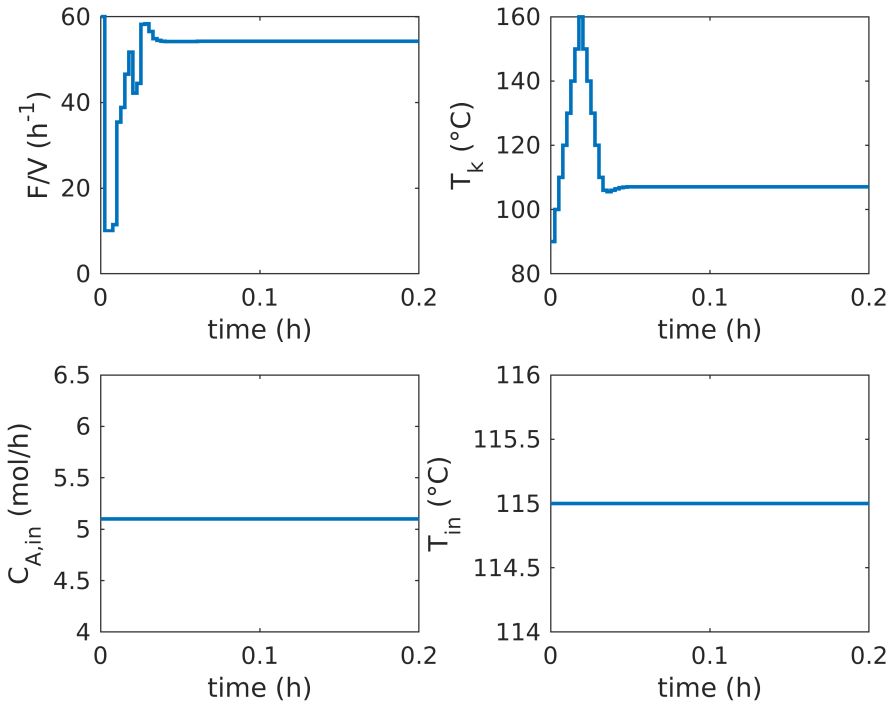

Figure 5.21: DRTO simulation scenario 2 - inputs and disturbances.

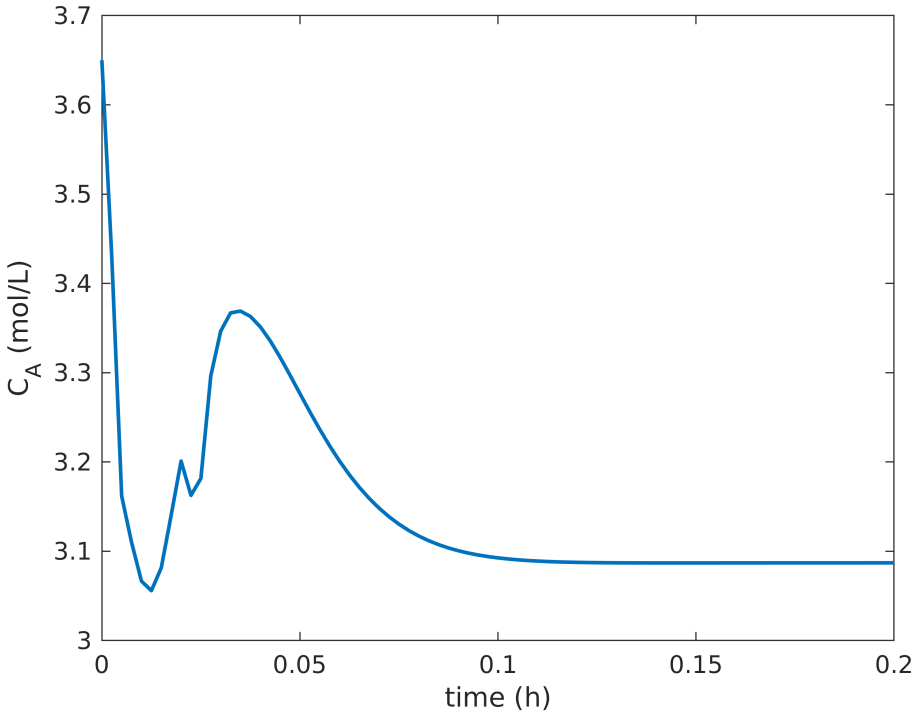

Figure 5.22: DRTO simulation scenario 3 - concentration of A.

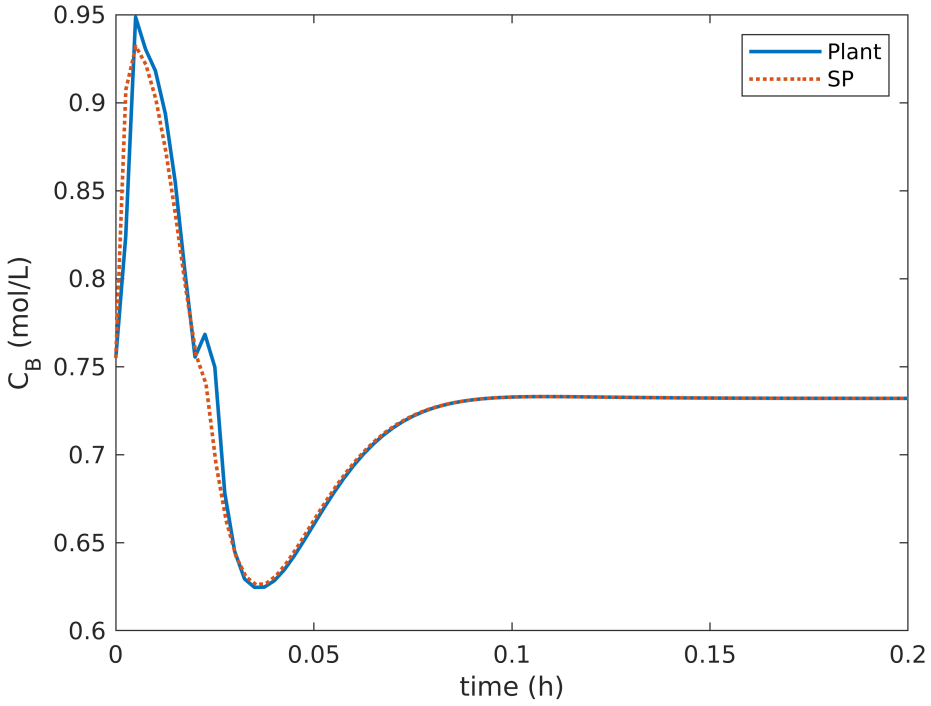

Figure 5.23: DRTO simulation scenario 3 - concentration of B.

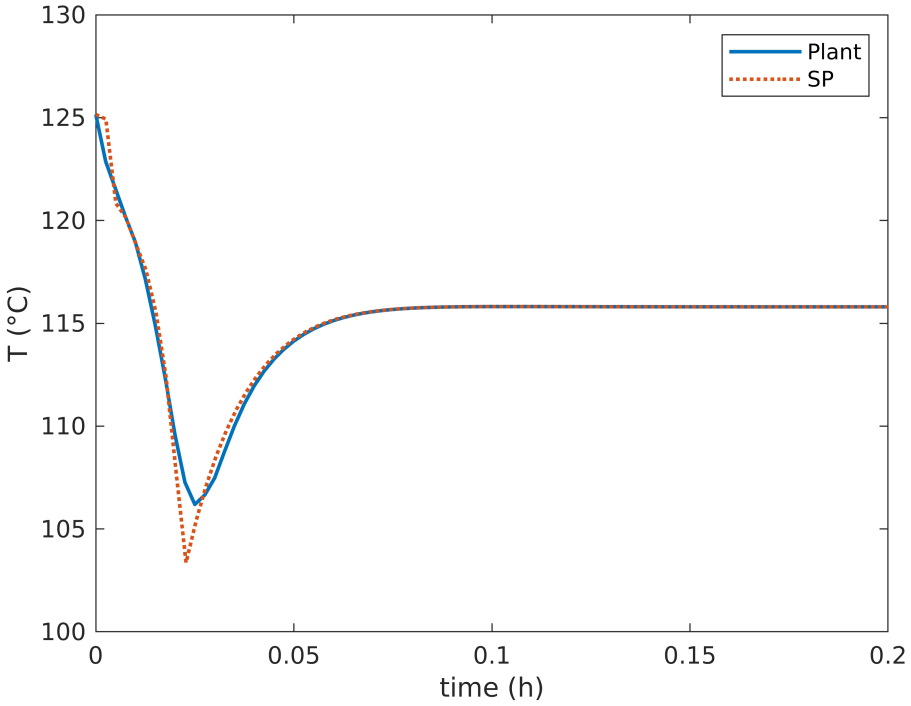

Figure 5.24: DRTO simulation scenario 3 - reactor temperature.

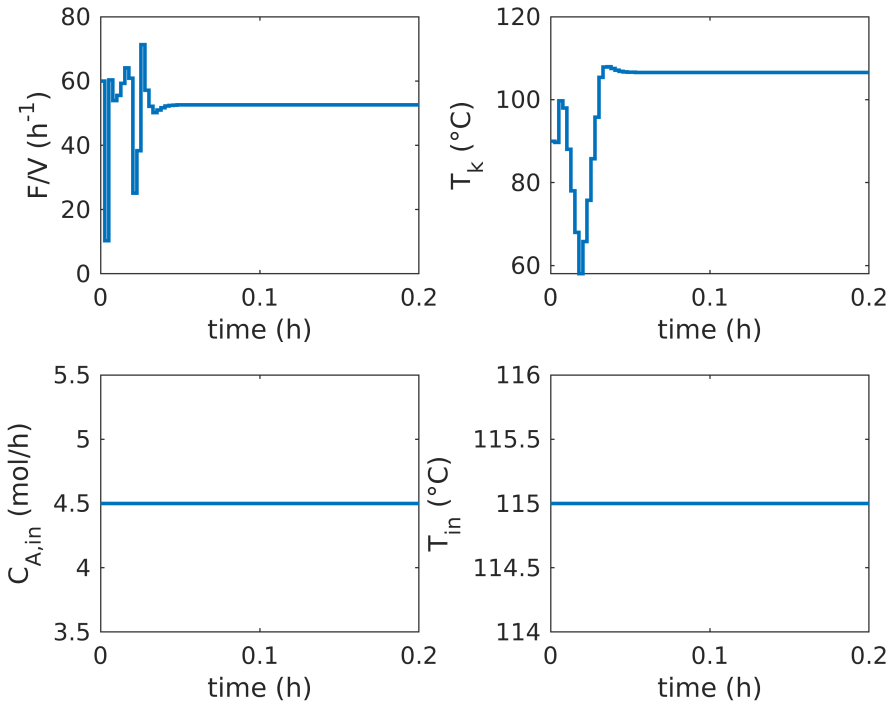

Figure 5.25: DRTO simulation scenario 3 - inputs and disturbances.

In all scenarios the DRTO was able to reach an optimal steady state, which is expected since the cost function chosen is its value at the end of the horizon. Complex dynamics behavior can be seen in all cases, especially in Figures 5.23 and 5.24. Scenario 3, i.e., with disturbances applied in both $C_{A,in}$ and T_{in}, was the the hardest one to control, with the NMPC not performing well in the undershoot range for both C_B and T profiles.

5.4 Standard RTO

Table 5.5 shows the scenarios studied with RTO and HRTO.

Table 5.5: RTO simulation scenarios.

Scenario	$C_{A,in}(mol/h)$	$T_{in}(°C)$	Simulation time (h)	Disturbance instant (h)
1	4.5	130	0.5	0.25
2	5.1	115	0.5	0.25
3	4.5	115	0.5	0.25

Figures 5.26 to 5.49. present the results for both RTOs presented in Chapter 4, the first one, named RTO-LS, coupled with a least-squares steady-state estimation step and the second one, named RTO-EKF, featuring an EKF dynamic estimator.

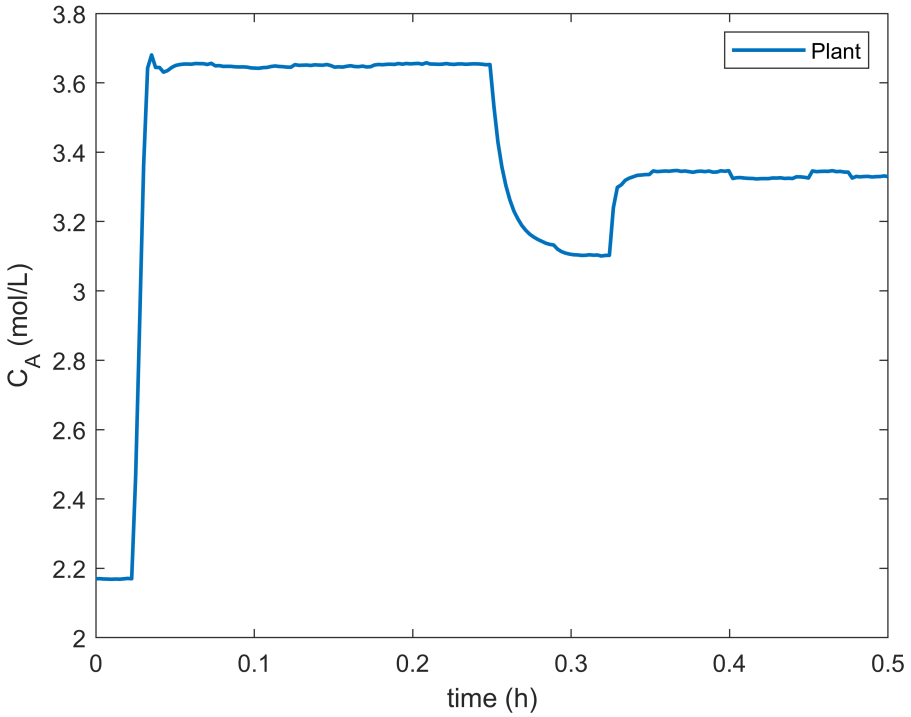

Figure 5.26: RTO-LS simulation scenario 1 - concentration of A.

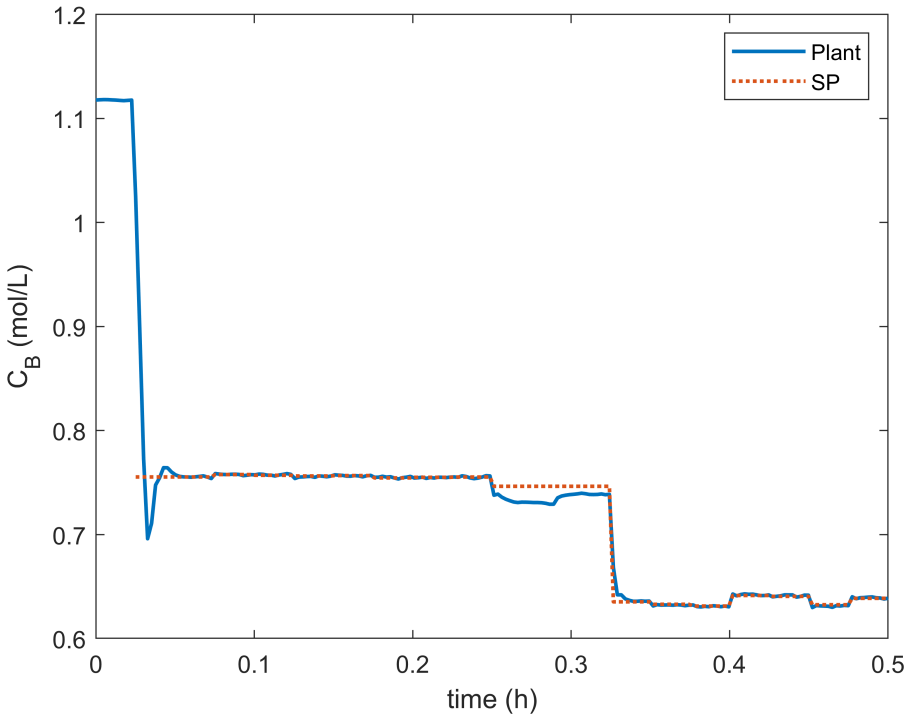

Figure 5.27: RTO-LS simulation scenario 1 - concentration of B.

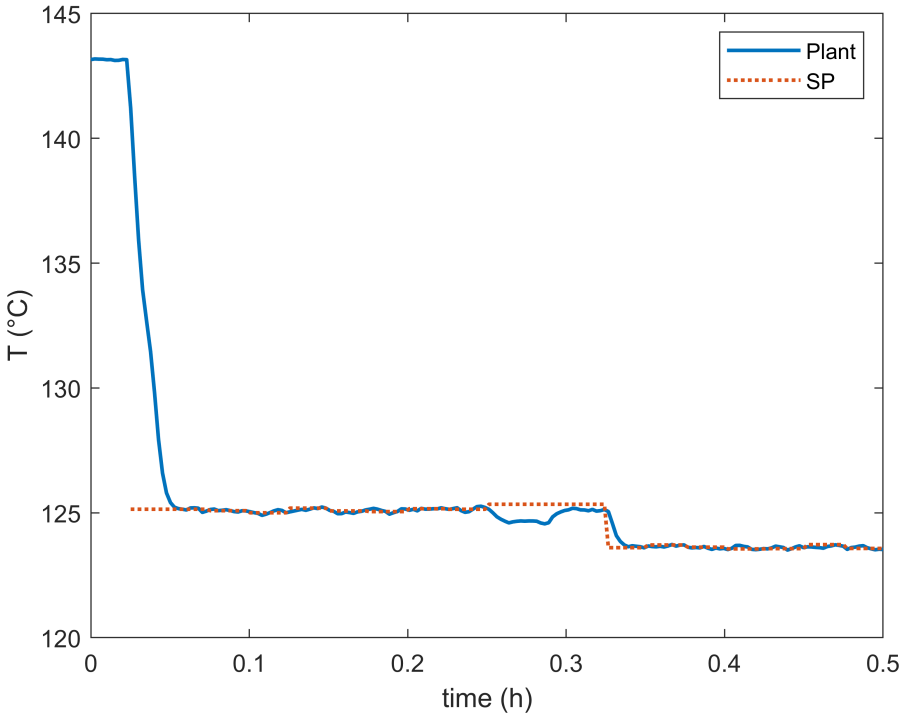

Figure 5.28: RTO-LS simulation scenario 1 - reactor temperature.

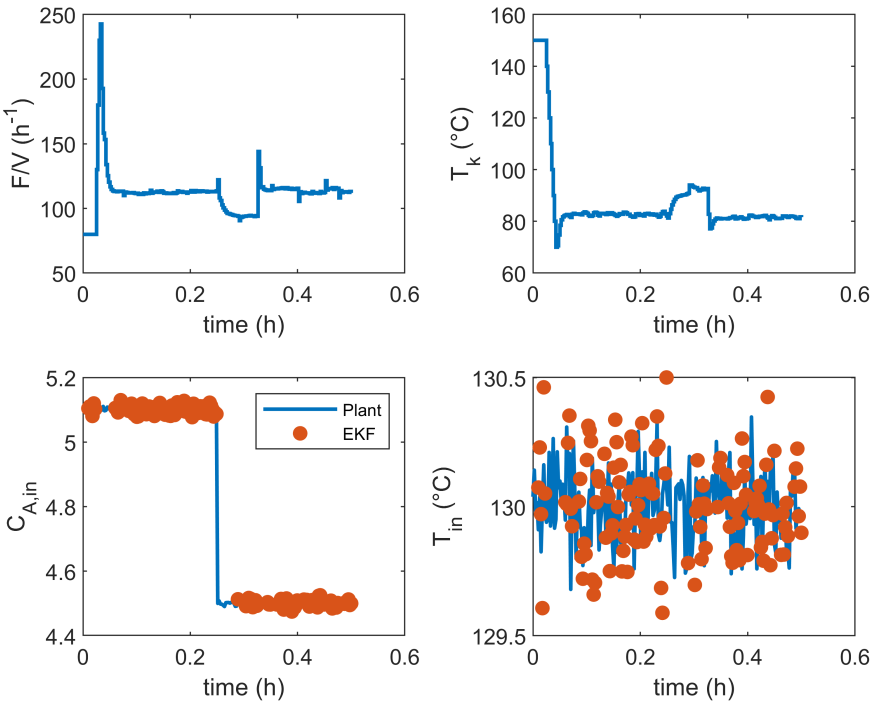

Figure 5.29: RTO-LS simulation scenario 1 - inputs and disturbances.

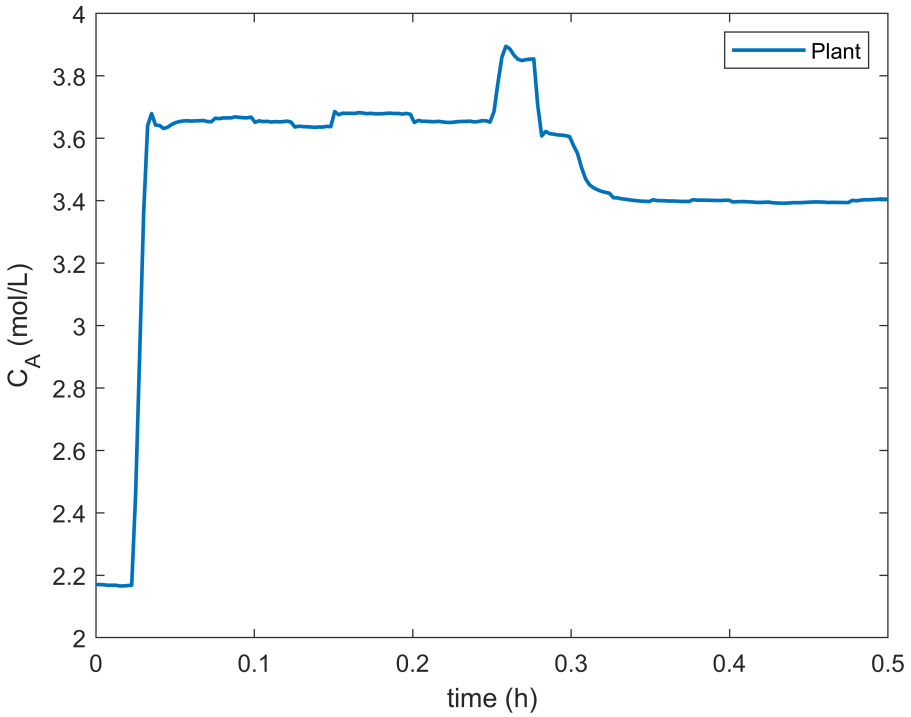

Figure 5.30: RTO-LS simulation scenario 2 - concentration of A.

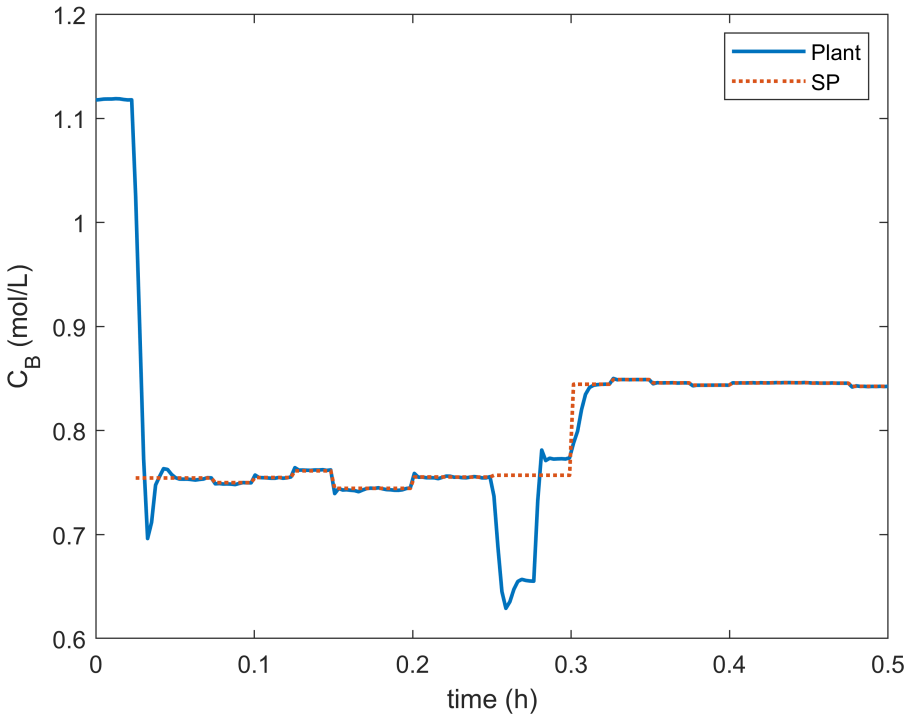

Figure 5.31: RTO-LS simulation scenario 2 - concentration of B.

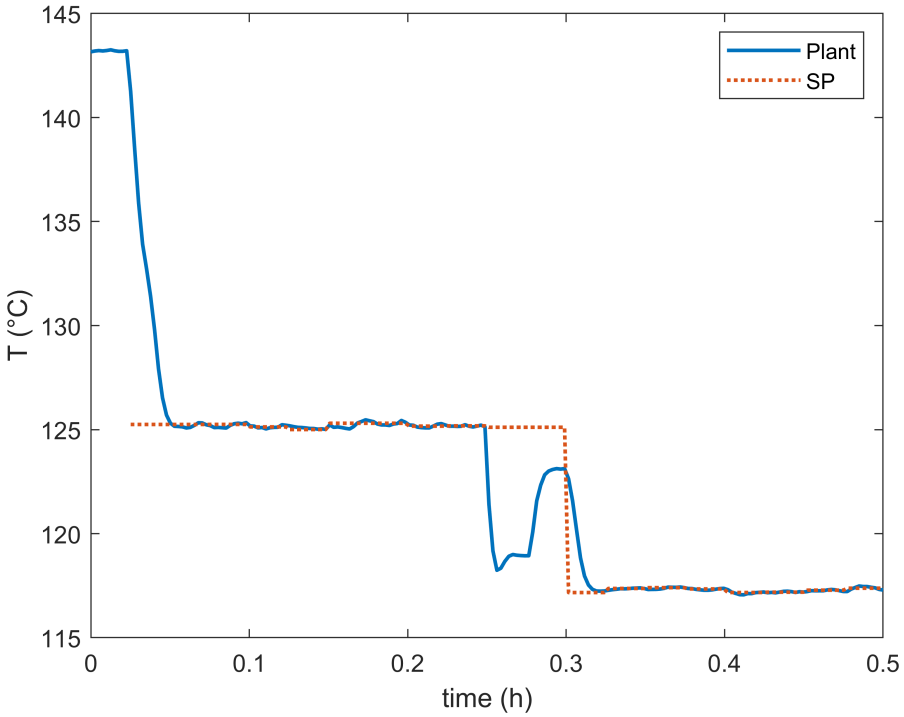

Figure 5.32: RTO-LS simulation scenario 2 - reactor temperature.

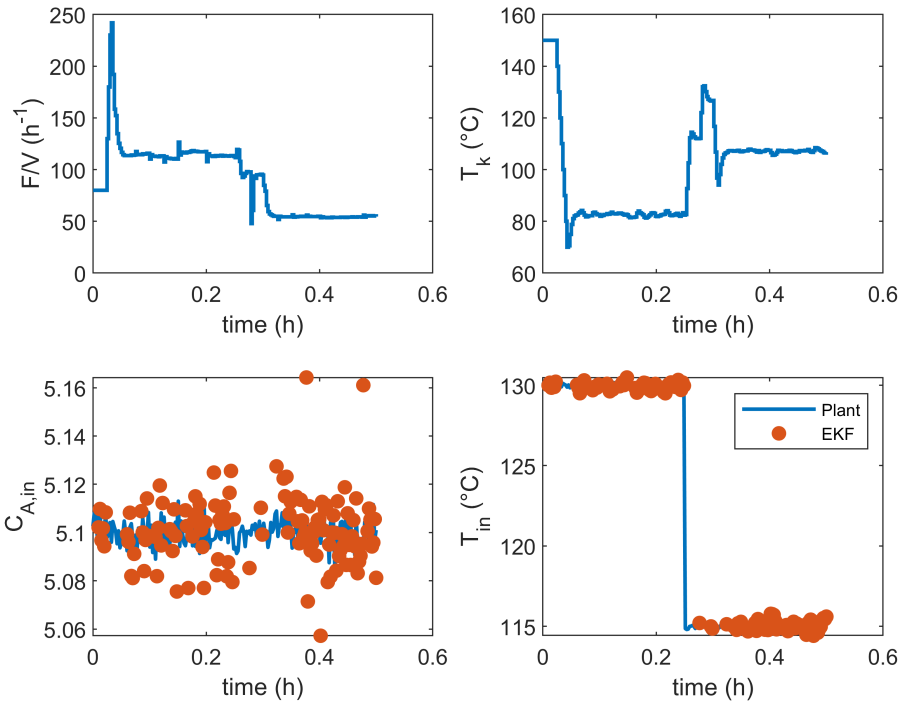

Figure 5.33: RTO-LS simulation scenario 2 - inputs and disturbances.

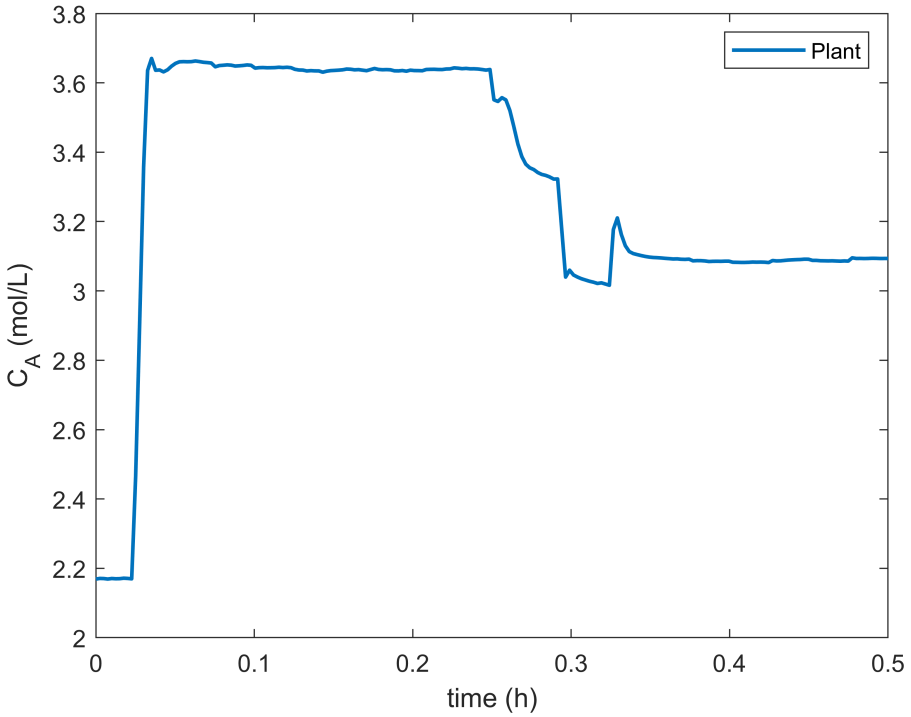

Figure 5.34: RTO-LS simulation scenario 3 - concentration of A.

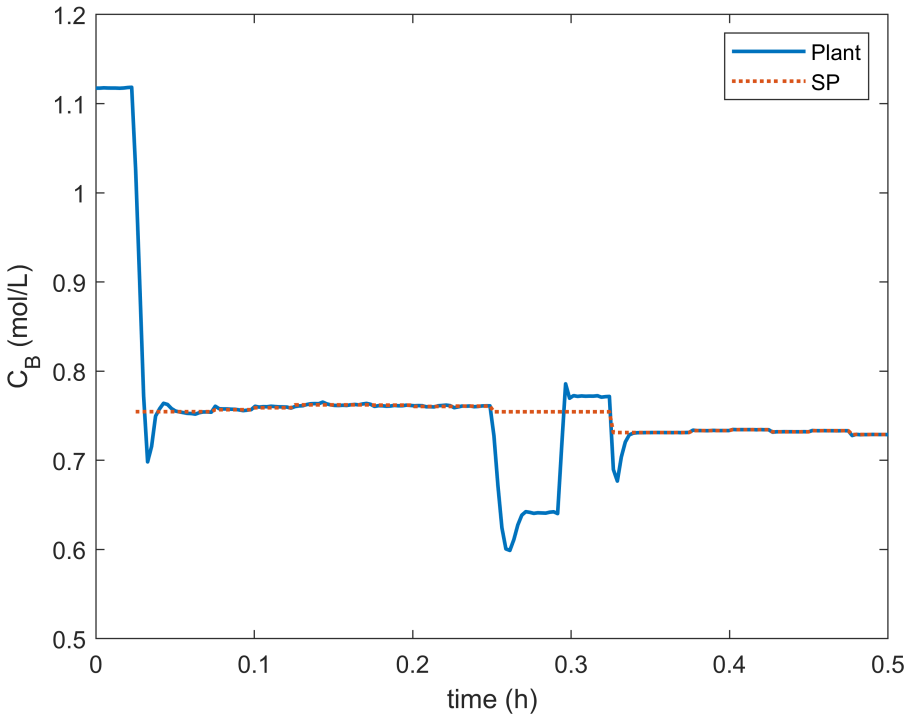

Figure 5.35: RTO-LS simulation scenario 3 - concentration of B.

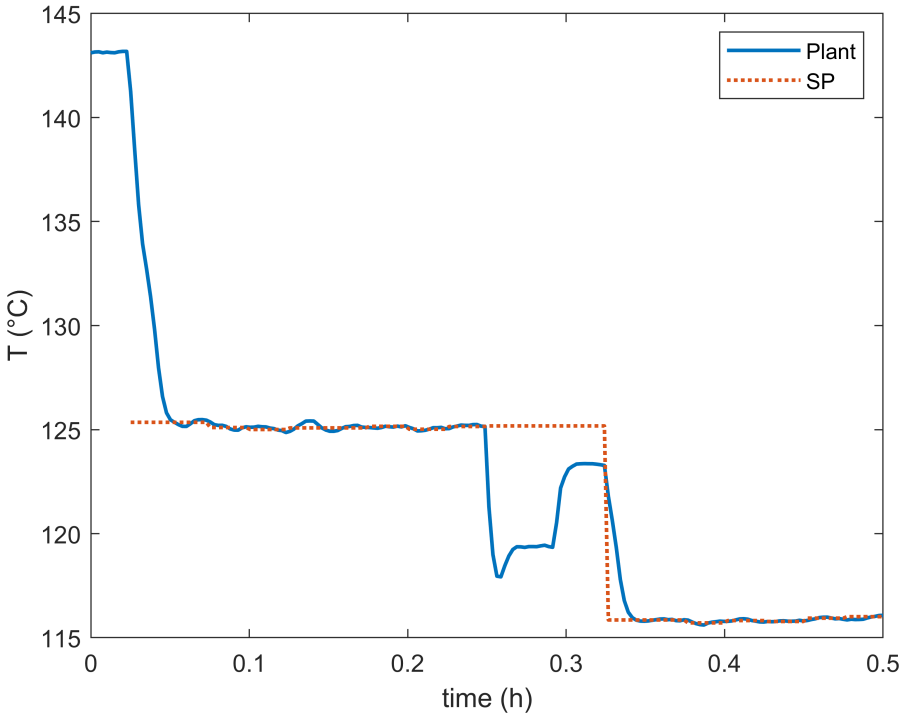

Figure 5.36: RTO-LS simulation scenario 3 - reactor temperature.

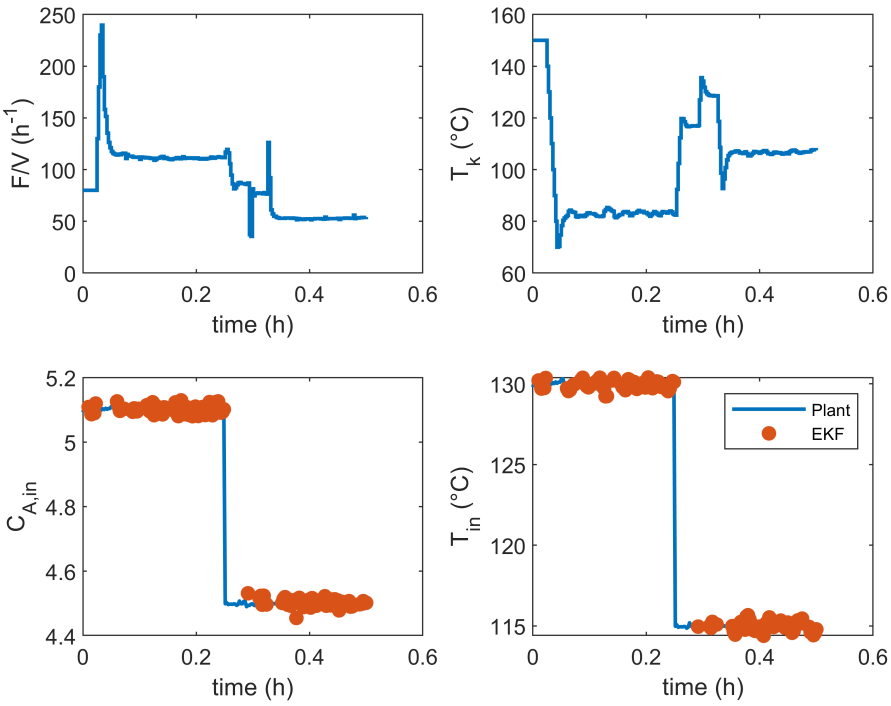

Figure 5.37: RTO-LS simulation scenario 3 - inputs and disturbances.

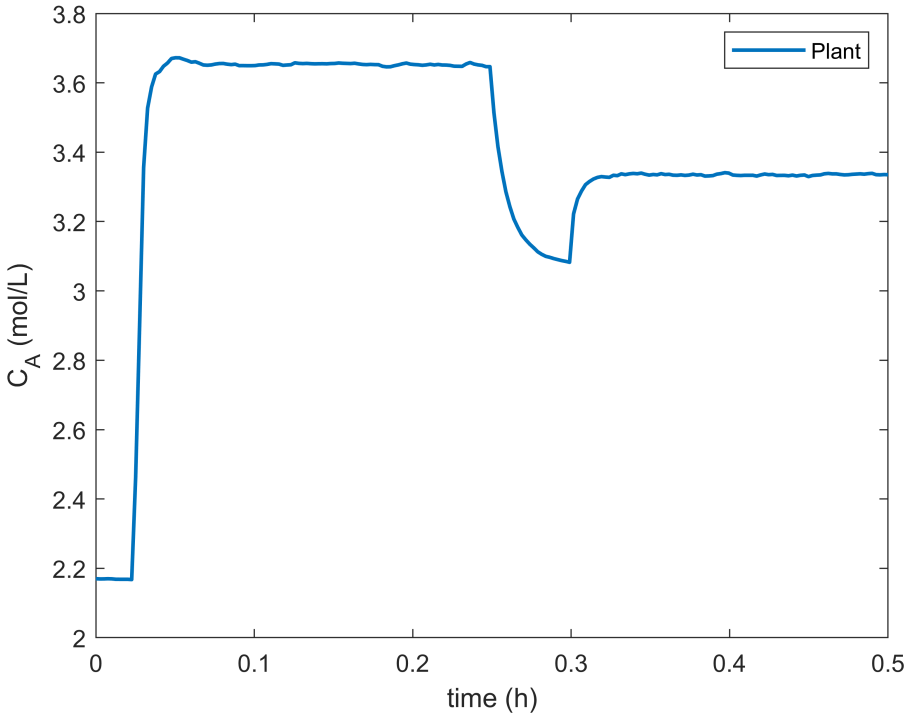

Figure 5.38: RTO-EKF simulation scenario 1 - concentration of A.

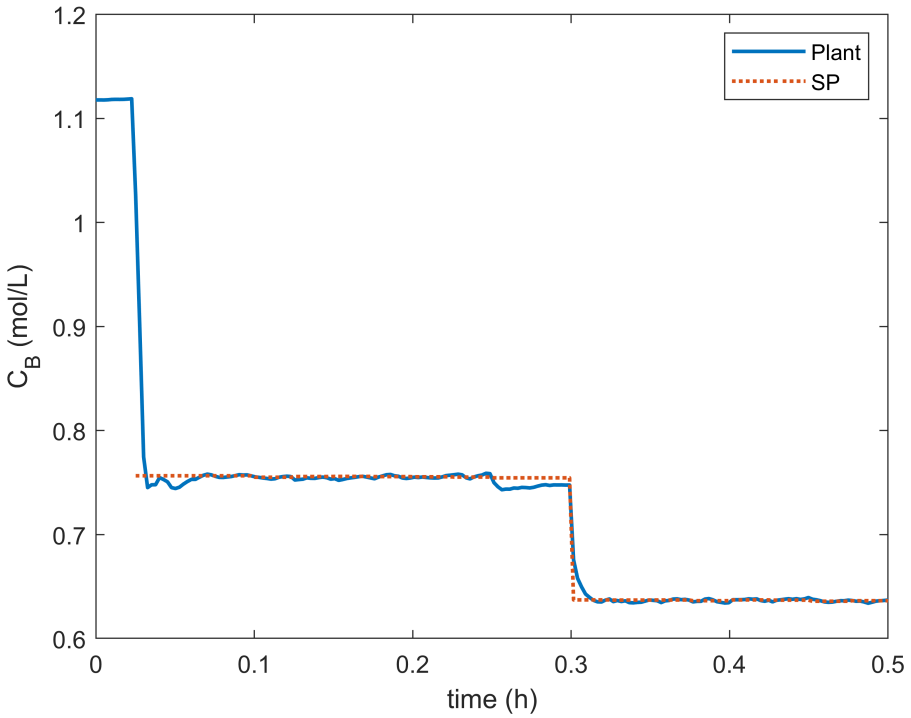

Figure 5.39: RTO-EKF simulation scenario 1 - concentration of B.

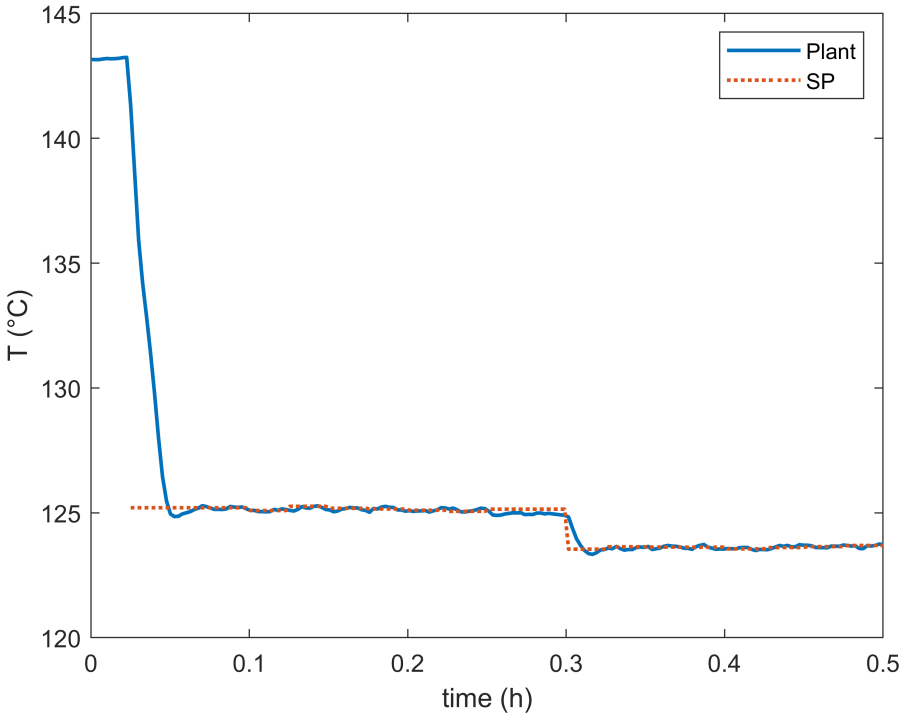

Figure 5.40: RTO-EKF simulation scenario 1 - reactor temperature.

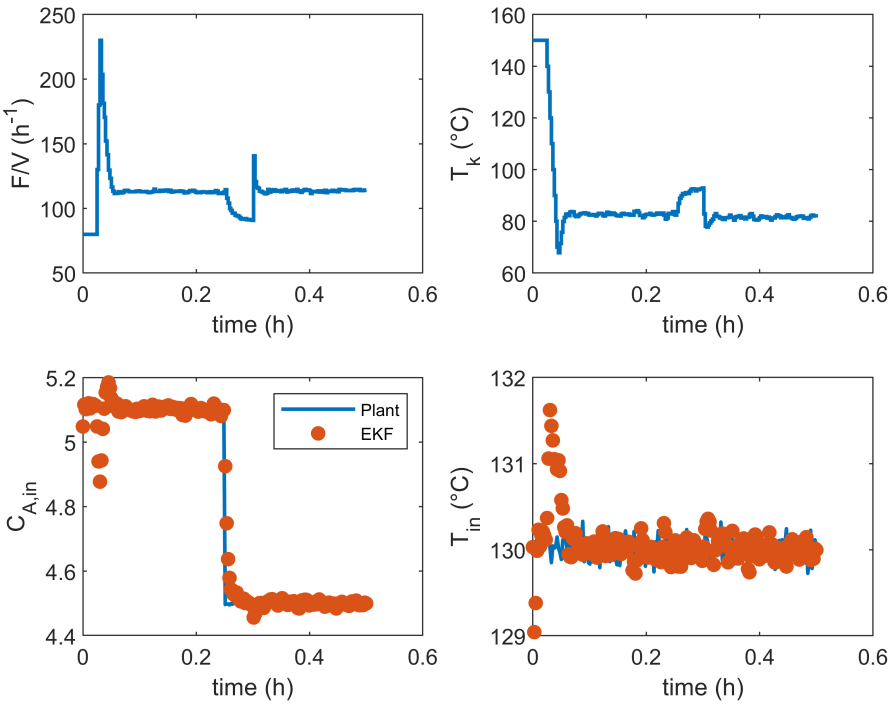

Figure 5.41: RTO-EKF simulation scenario 1 - inputs and disturbances.

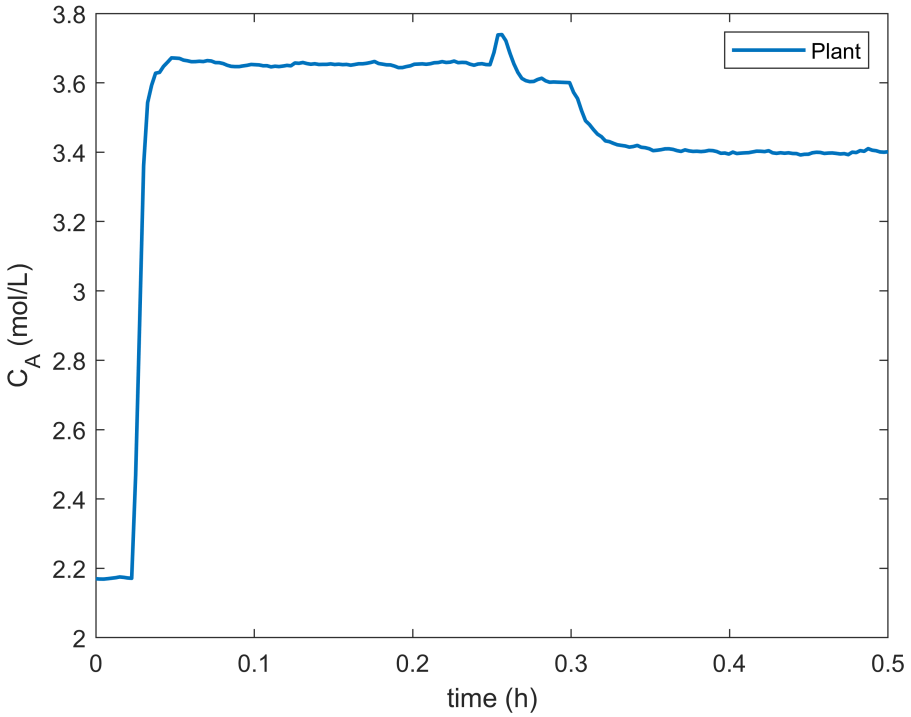

Figure 5.42: RTO-EKF simulation scenario 2 - concentration of A.

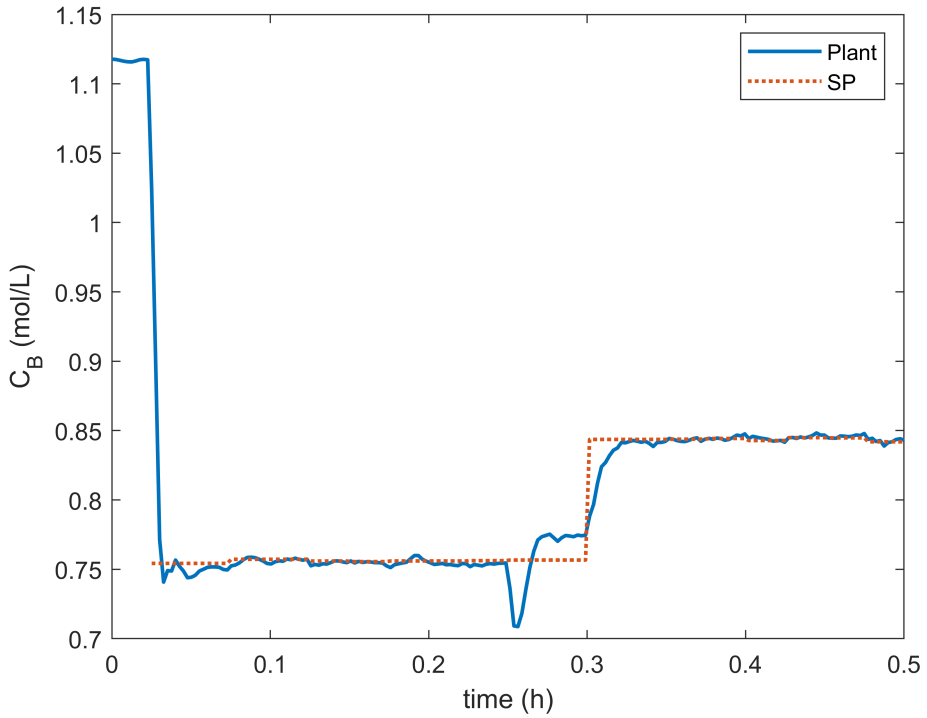

Figure 5.43: RTO-EKF simulation scenario 2 - concentration of B.

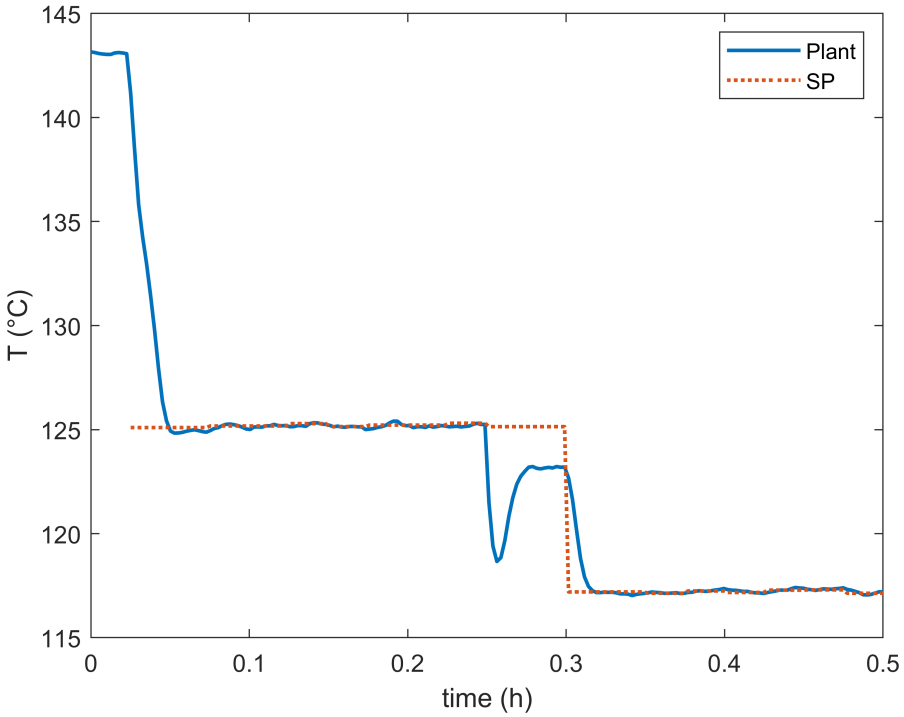

Figure 5.44: RTO-EKF simulation scenario 2 - reactor temperature.

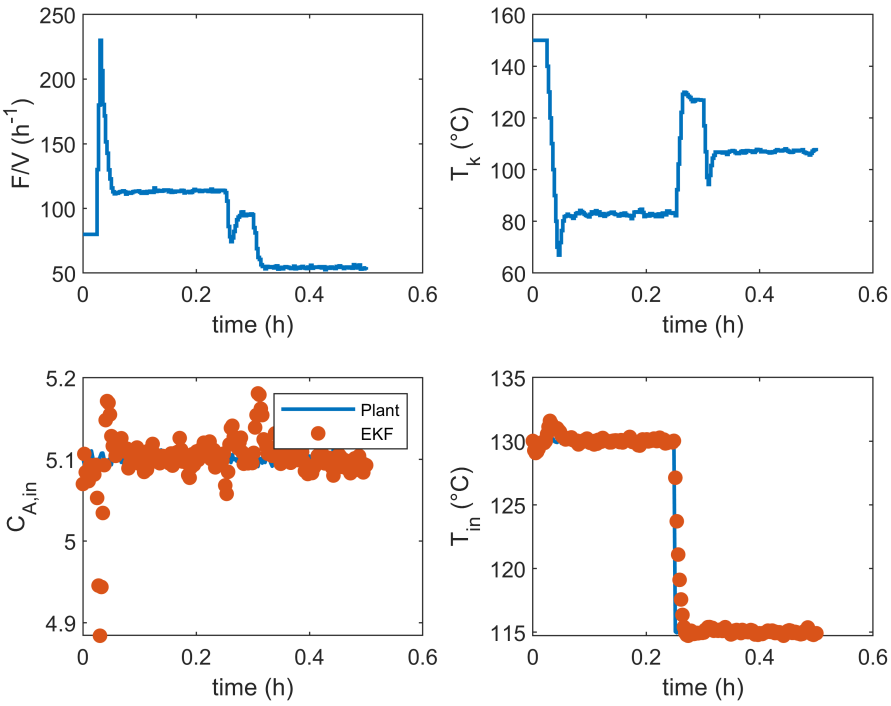

Figure 5.45: RTO-EKF simulation scenario 2 - inputs and disturbances.

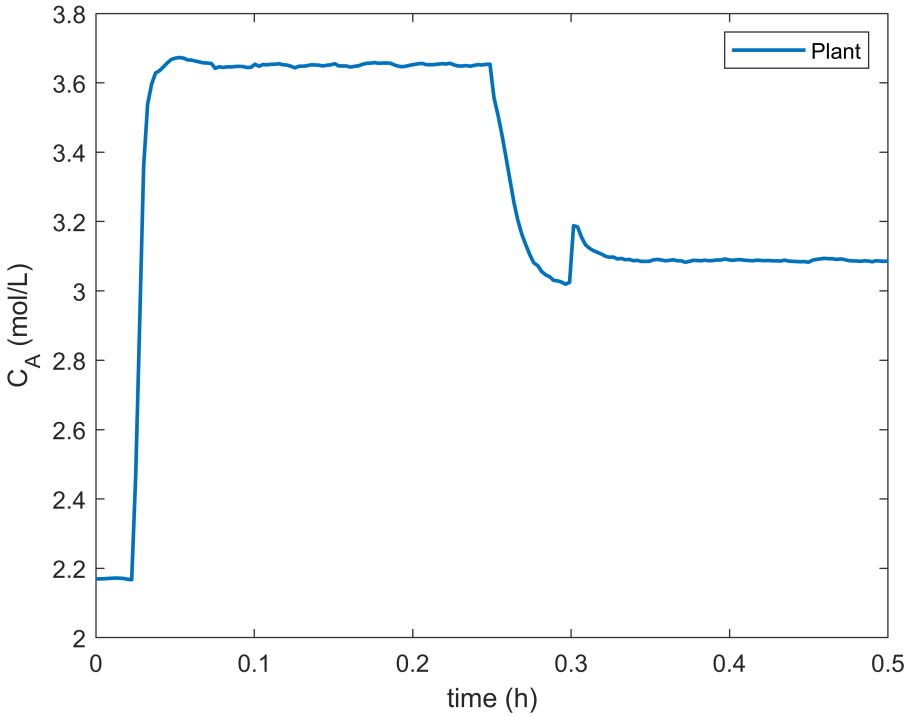

Figure 5.46: RTO-EKF simulation scenario 3 - concentration of A.

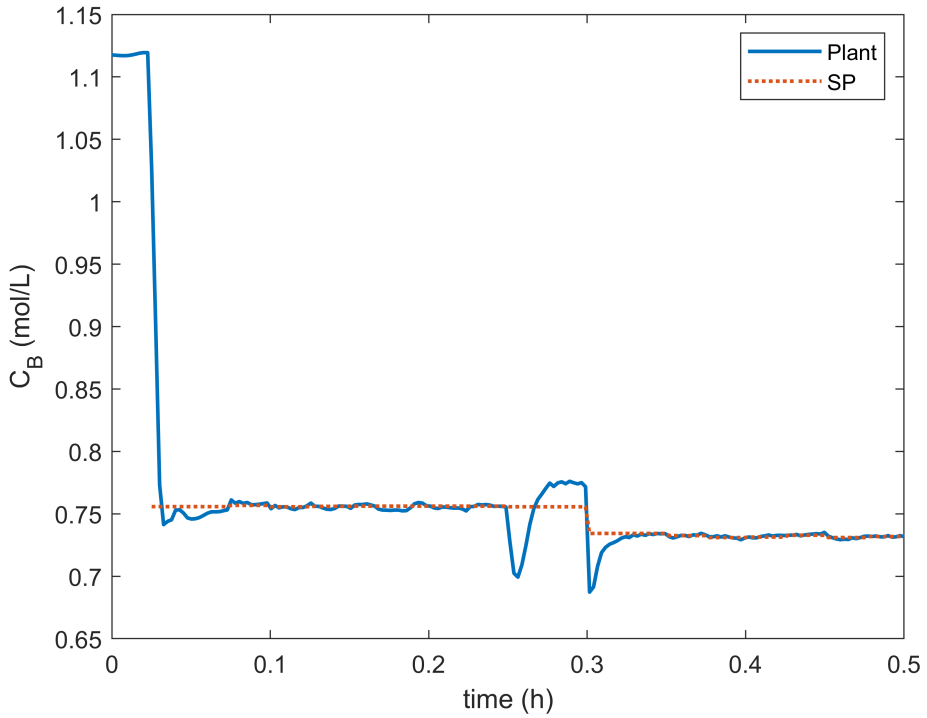

Figure 5.47: RTO-EKF simulation scenario 3 - concentration of B.

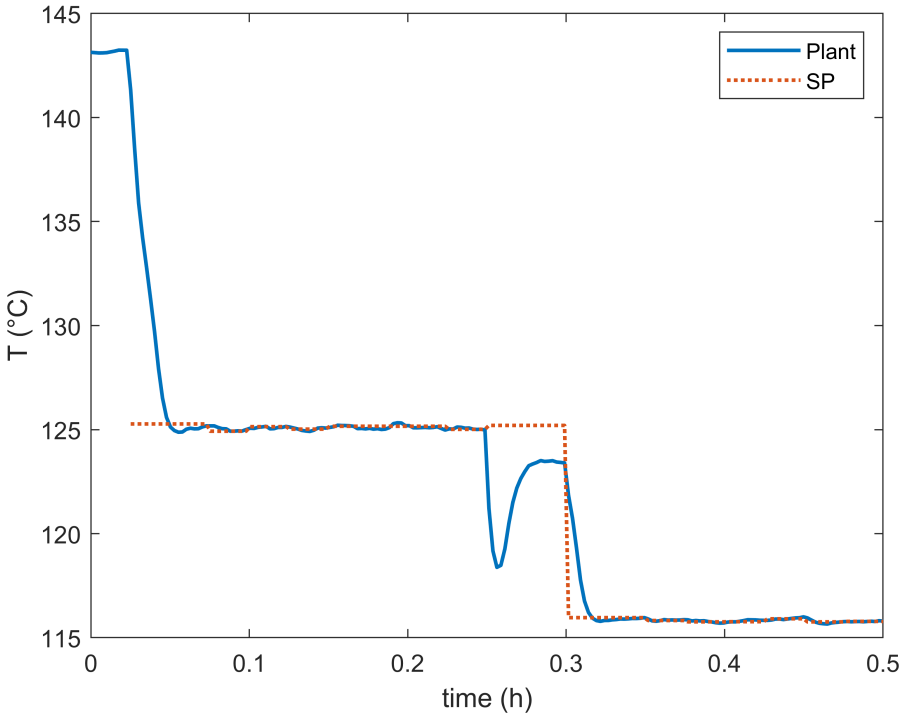

Figure 5.48: RTO-EKF simulation scenario 3 - reactor temperature.

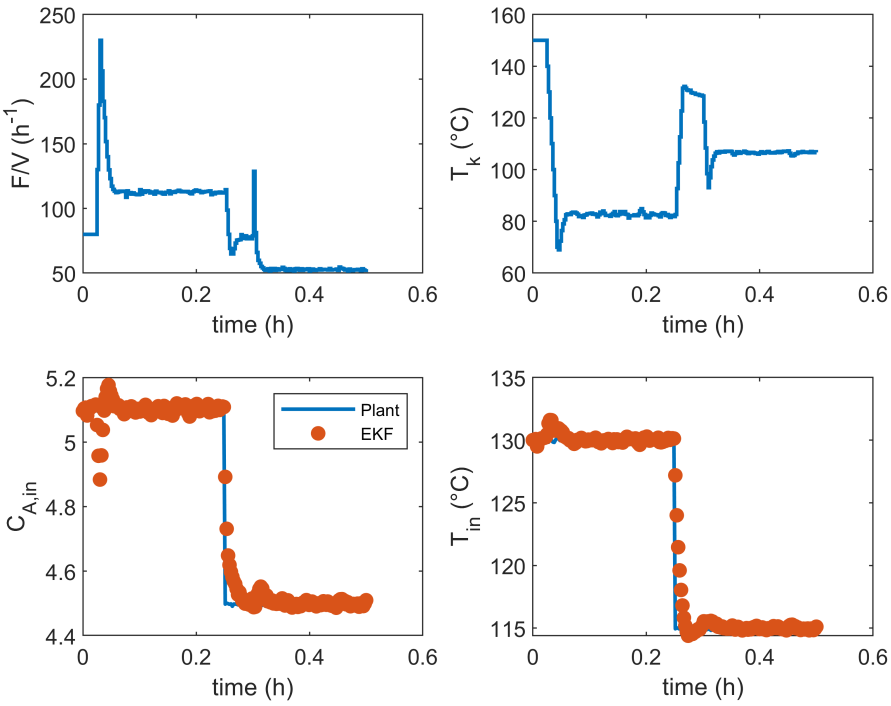

Figure 5.49: RTO-EKF simulation scenario 3 - inputs and disturbances.

It is noteworthy that the RTO with steady-state estimation has a bigger ten-

dency of leaving an offset since the disturbances are only being estimated at steady-state, as it is clearly shown in these results. Thus it makes sense that the Kalman filter use at all time steps makes the NMPC perform better, even though there are still the presence of some offset ranges. Also, since RTO-LS has no state estimation in its algorithm, the perfect NMPC model for the hypothesis was assumed for the states, i.e., the controller received the plant measurements directly without process noise.

5.5 Hybrid RTO

Figures 5.50 to 5.61 illustrate the results for the HRTO. The simulations were performed at the same conditions shown in Table 5.5.

For the HRTO, its possible the point out the constant setpoint update since the steady-state waiting time is not required anymore. In Figure 5.60 for example, it is also notable that this fact allowed a better setpoint attendance due the fact that the optimizer quickly updates it, even before the system is able to reach it before the update.

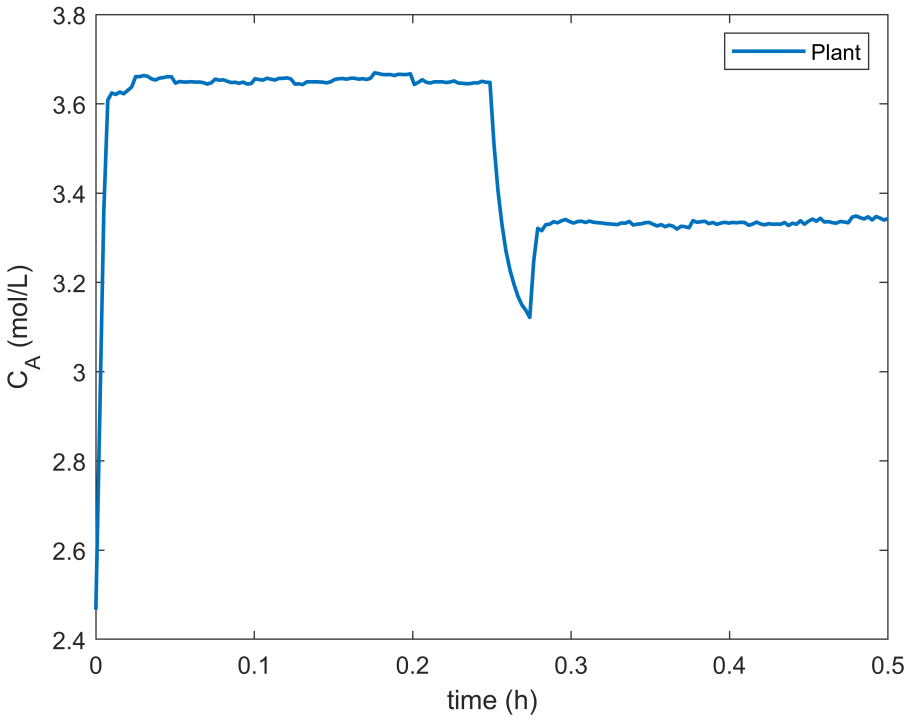

Figure 5.50: HRTO simulation scenario 1 - concentration of A.

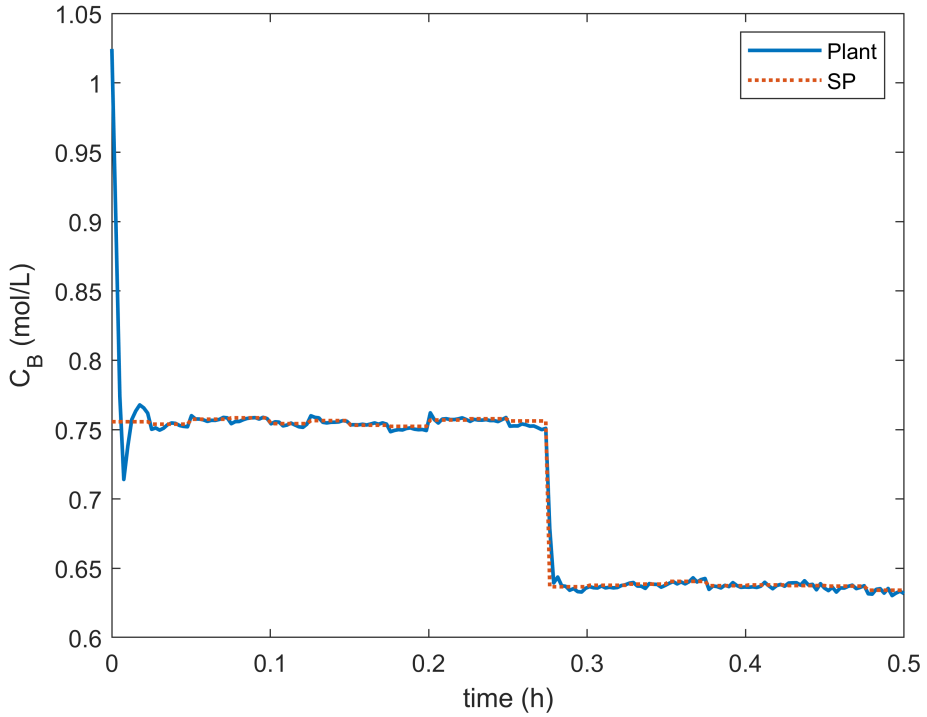

Figure 5.51: HRTO simulation scenario 1 - concentration of B.

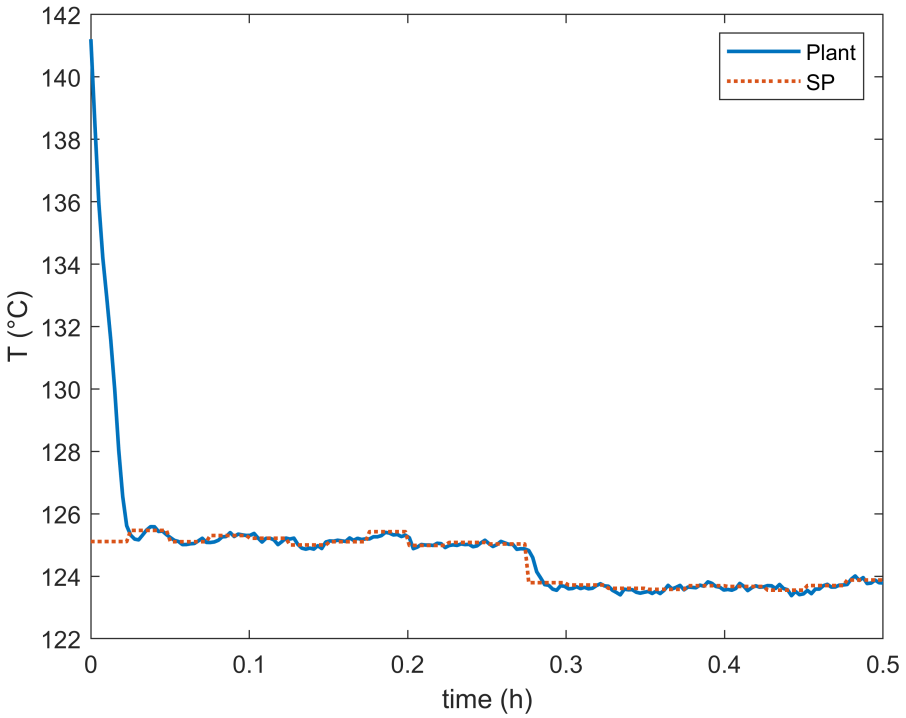

Figure 5.52: HRTO simulation scenario 1 - reactor temperature.

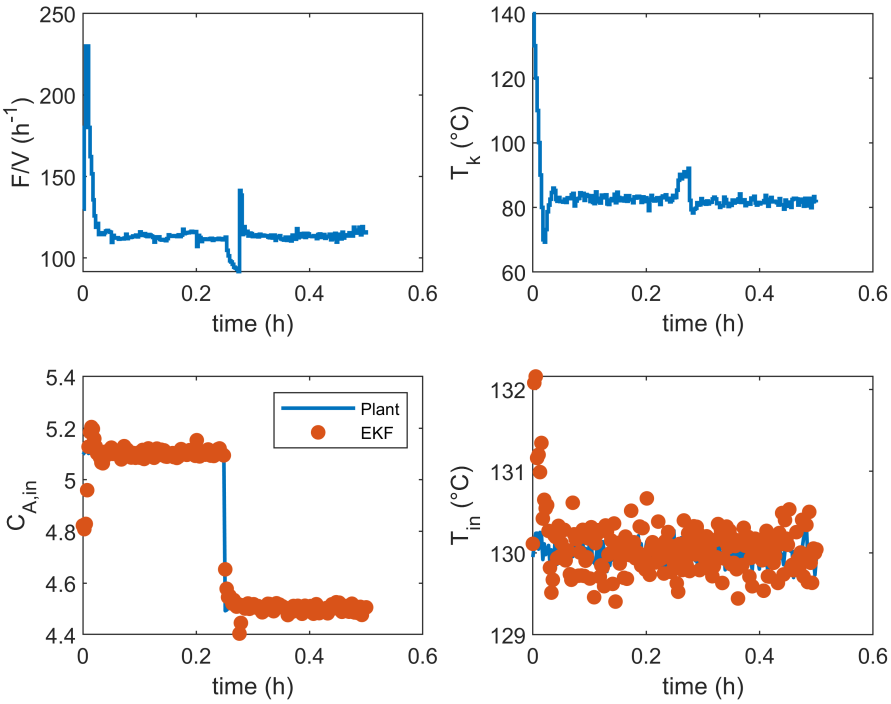

Figure 5.53: HRTO simulation scenario 1 - inputs and disturbances.

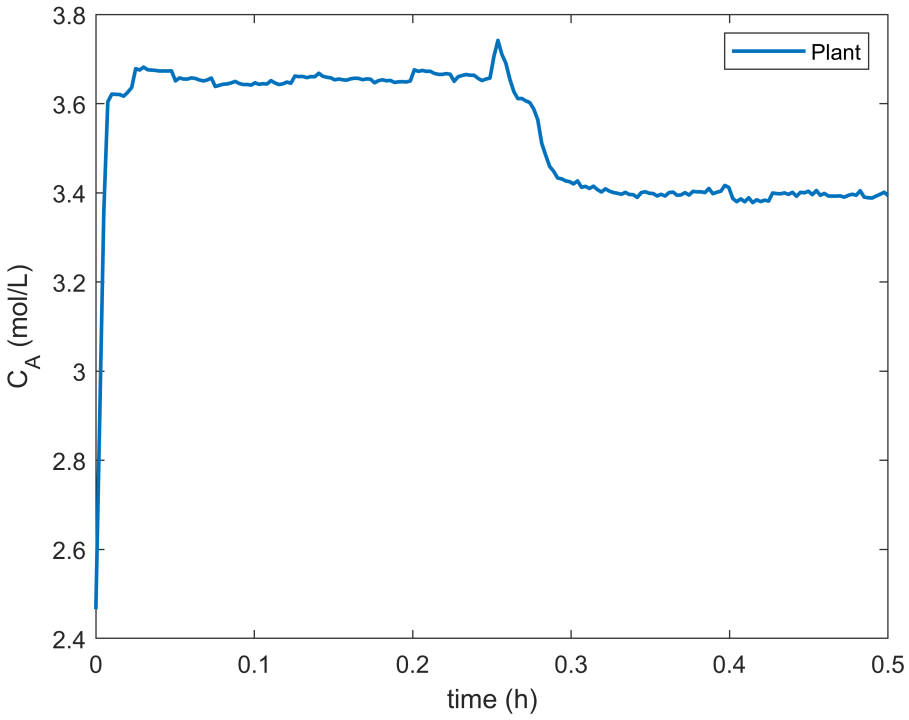

Figure 5.54: HRTO simulation scenario 2 - concentration of A.

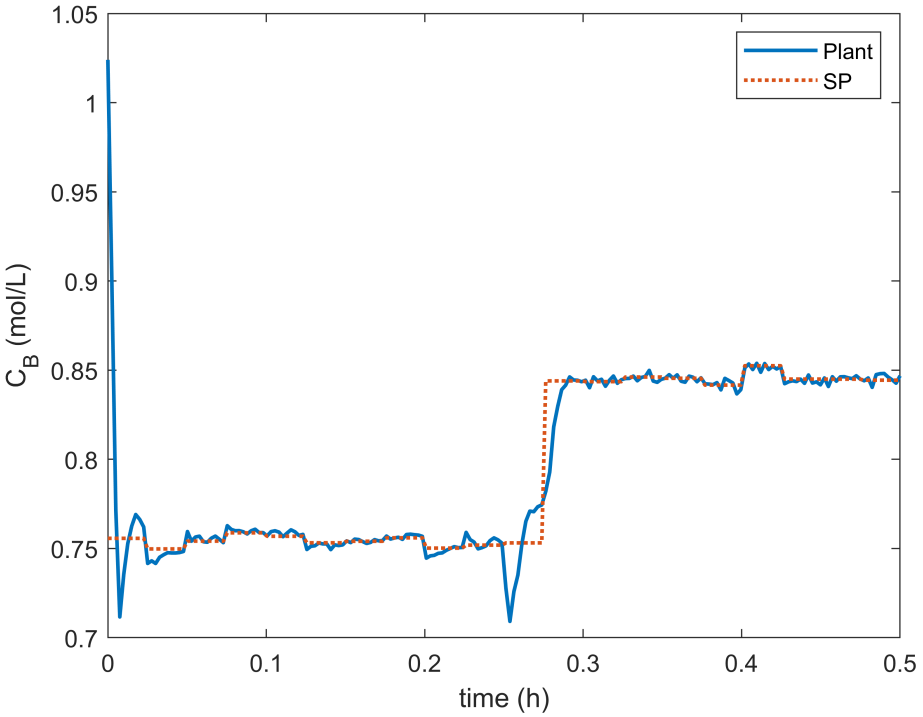

Figure 5.55: HRTO simulation scenario 2 - concentration of B.

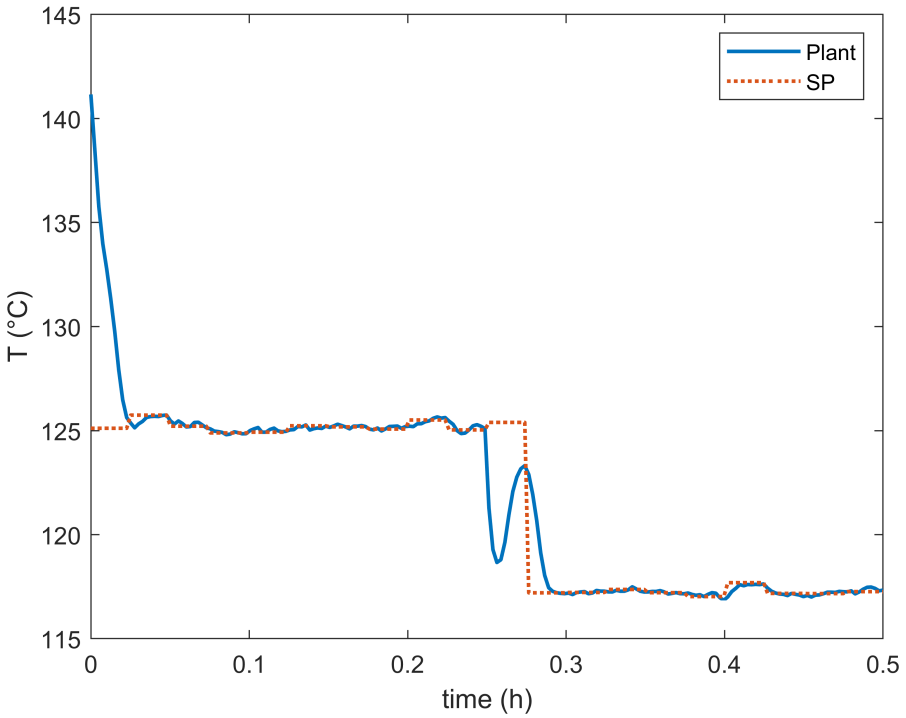

Figure 5.56: HRTO simulation scenario 2 - reactor temperature.

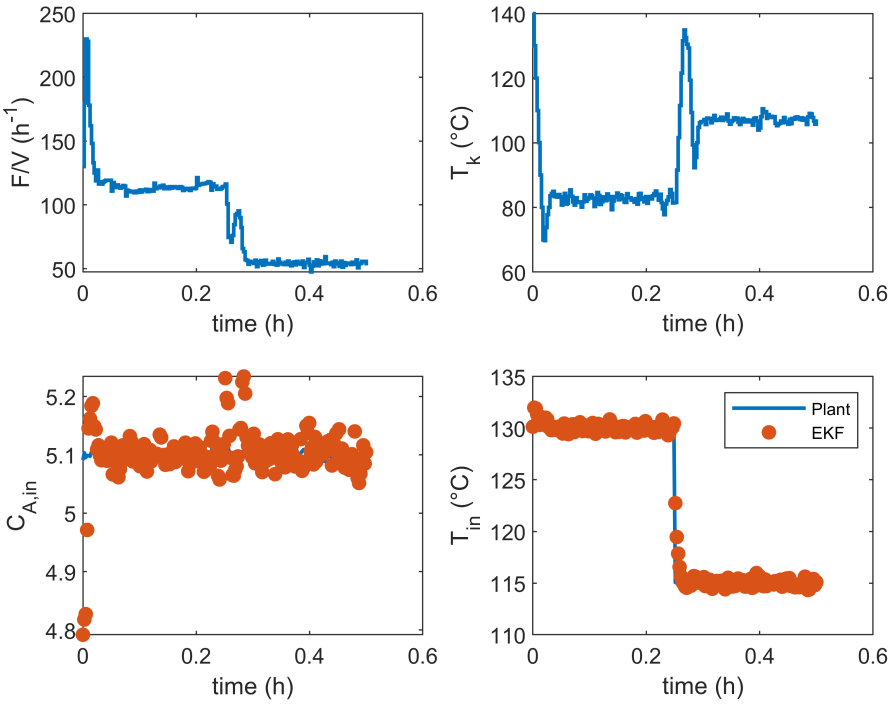

Figure 5.57: HRTO simulation scenario 2 - inputs and disturbances.

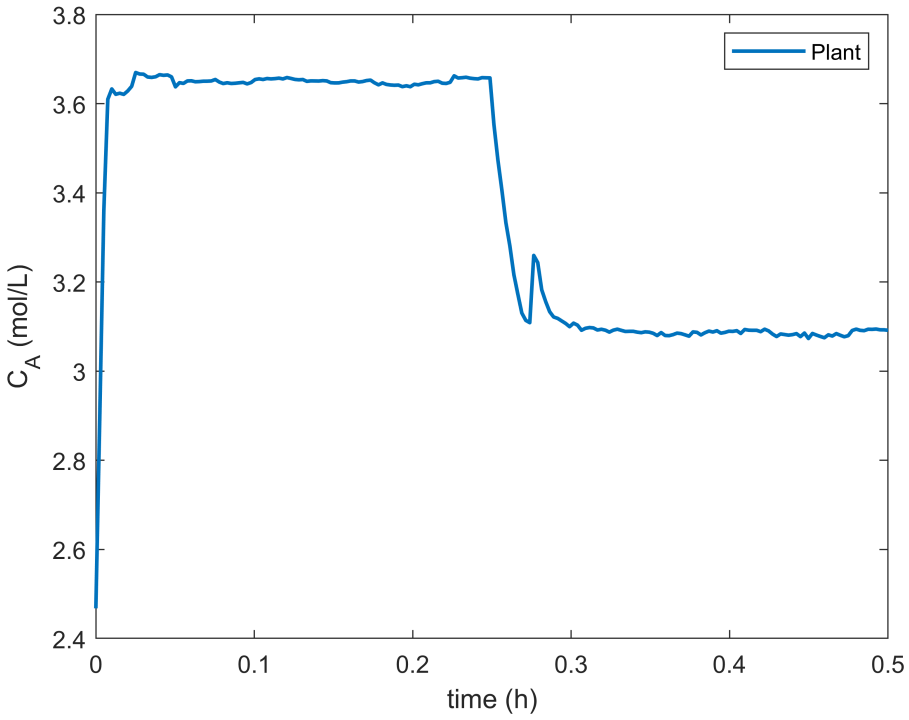

Figure 5.58: HRTO simulation scenario 3 - concentration of A.

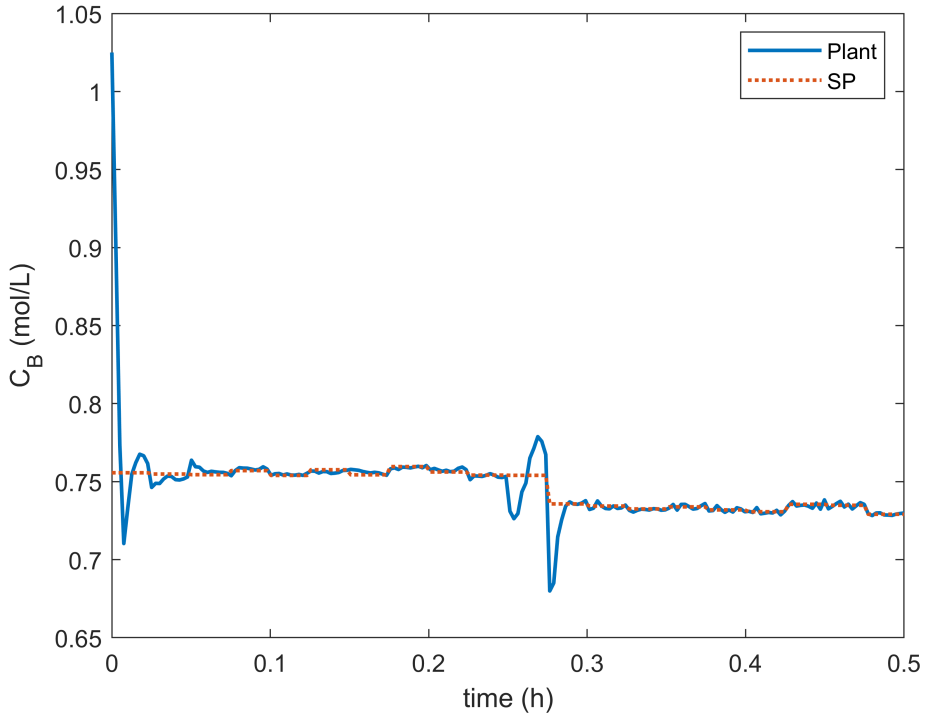

Figure 5.59: HRTO simulation scenario 3 - concentration of B.

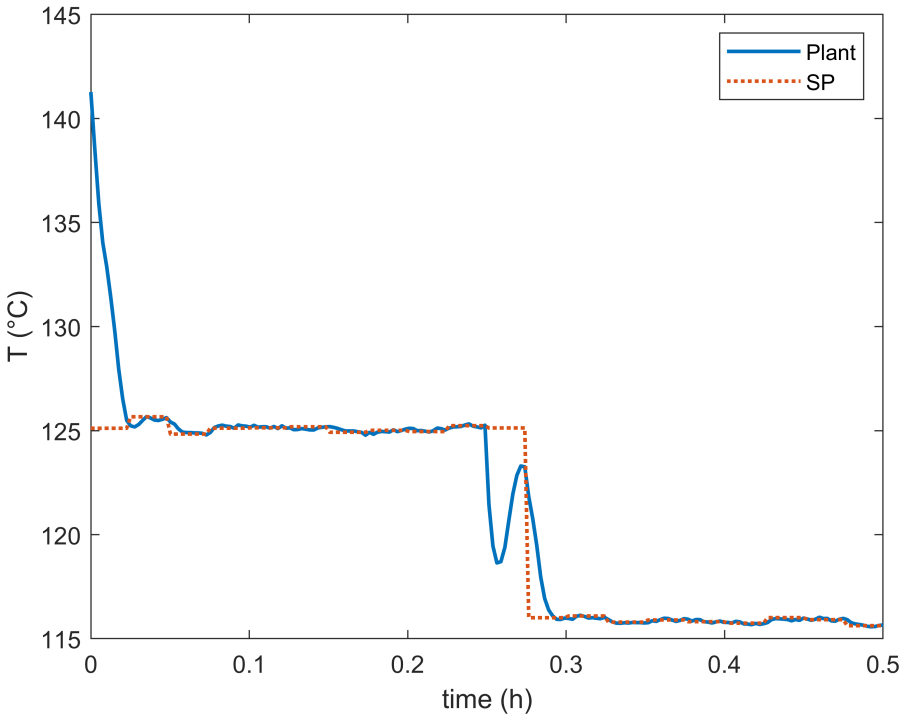

Figure 5.60: HRTO simulation scenario 3 - reactor temperature.

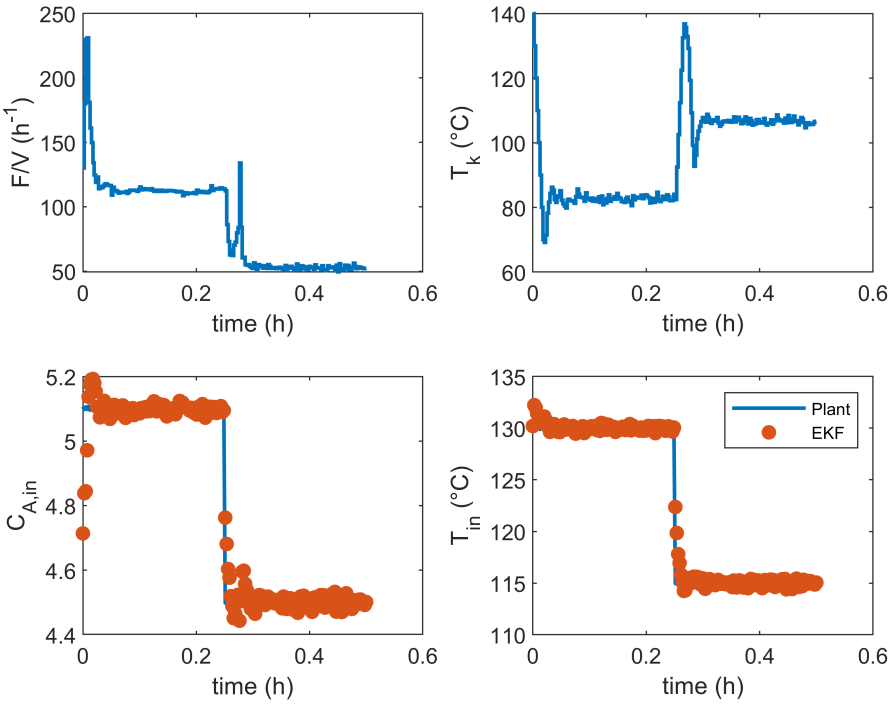

Figure 5.61: HRTO simulation scenario 3 - inputs and disturbances.

Another point is the fact that the HRTO profiles are substantially noisier when compared with the results of the previous section. This behavior is expected when taking into account that the optimizer is sensitive to process noise and is constantly updating the optimal operation point.

5.6 RTO-HRTO performance comparison

The RTO-LS approach is not considered here because, thanks to its limitations, the NMPC model needed to be modified, being pointless for it to be further compared with the other approaches. Also, it presented a substantially worse performance when compared to RTO-EKF, as discussed previously.

Figures 5.62 to 5.64 illustrate the objective function profiles for DRTO, while Figures 5.65 to 5.67 show the objective function profiles for RTO-EKF and HRTO for each scenario simulated.

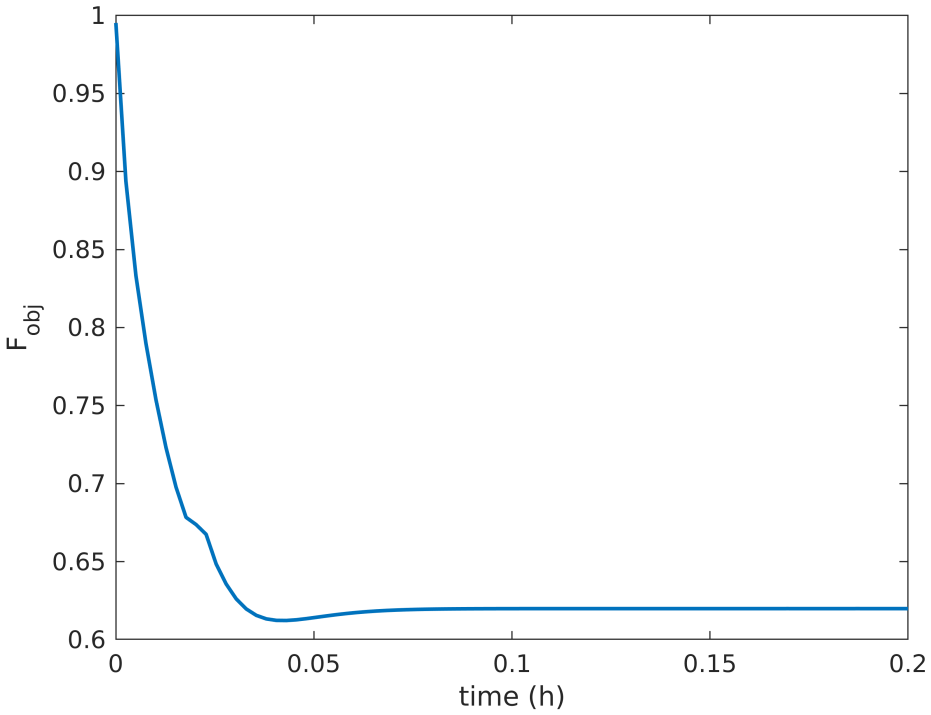

Figure 5.62: DRTO scenario 1 - objective function evaluation.

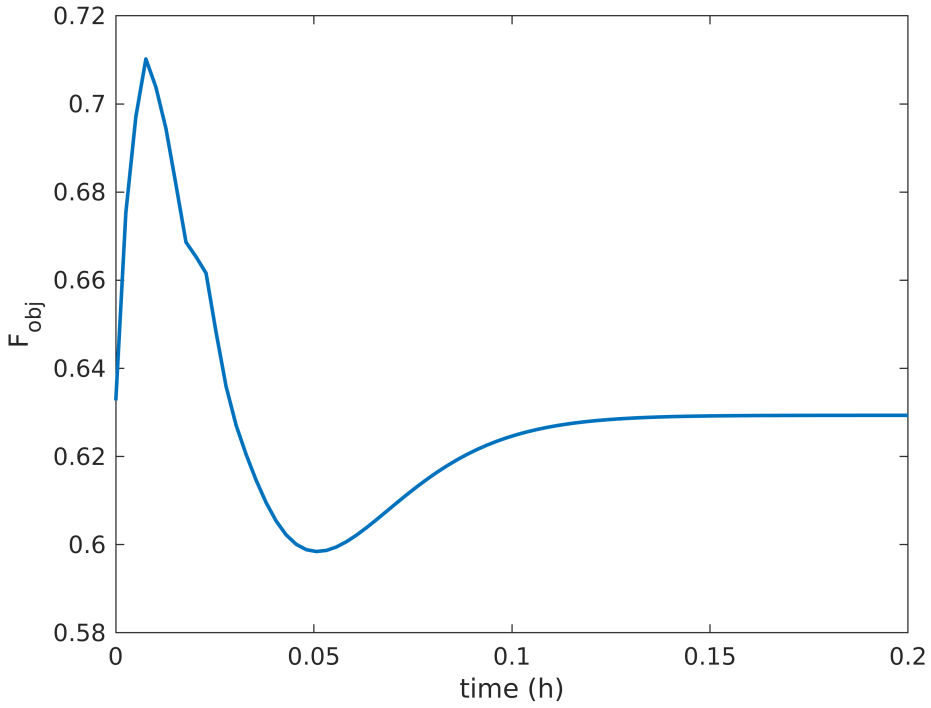

Figure 5.63: DRTO scenario 2 - objective function evaluation.

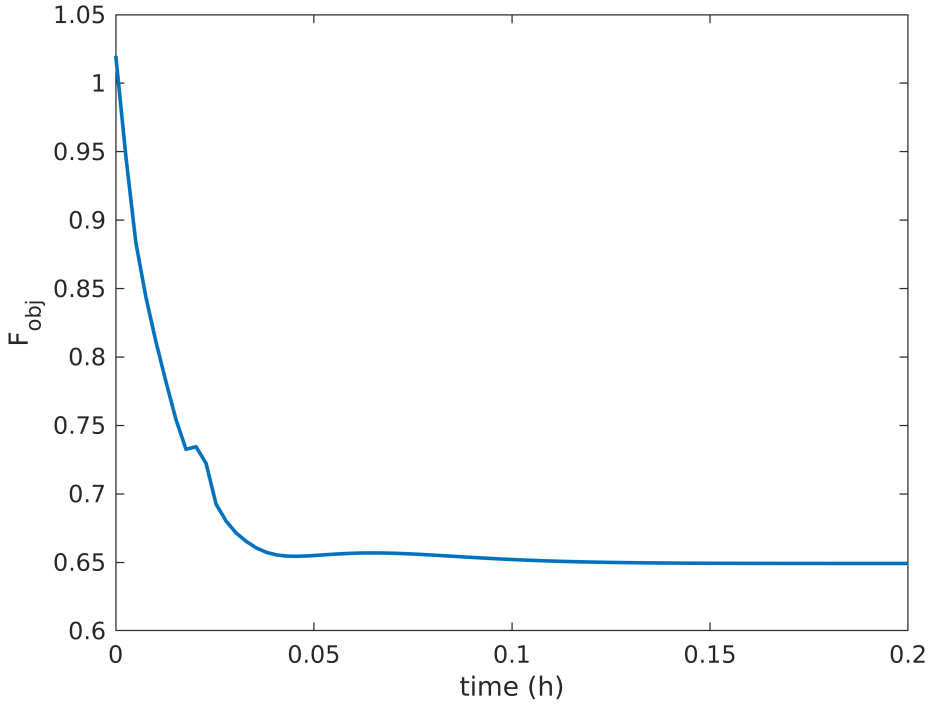

Figure 5.64: DRTO scenario 3 - objective function evaluation.

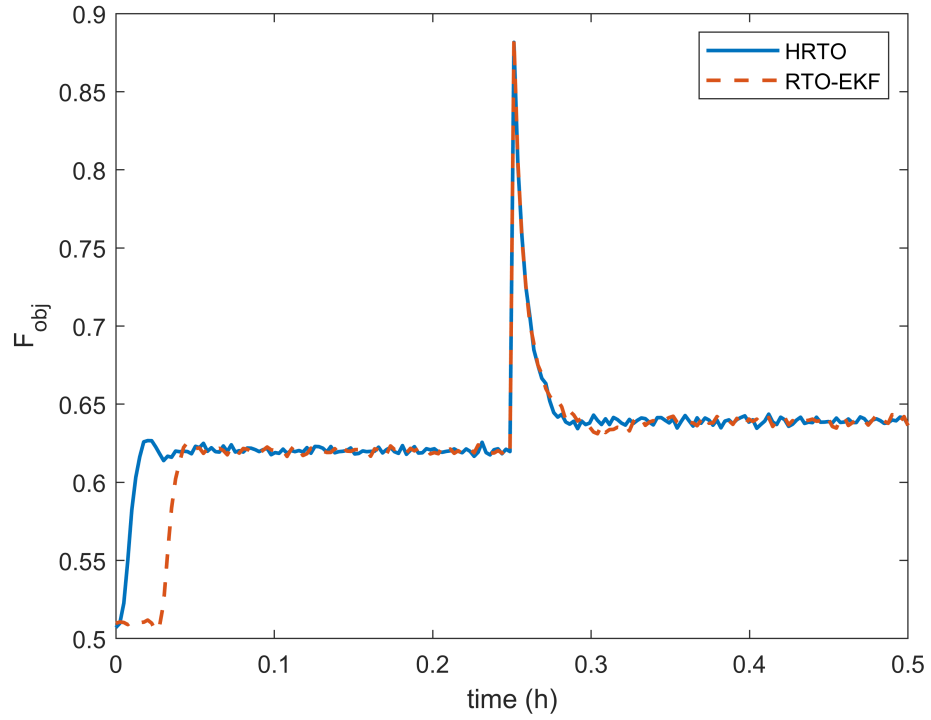

Figure 5.65: HRTO and RTO-EKF scenario 1 - objective function evaluation.

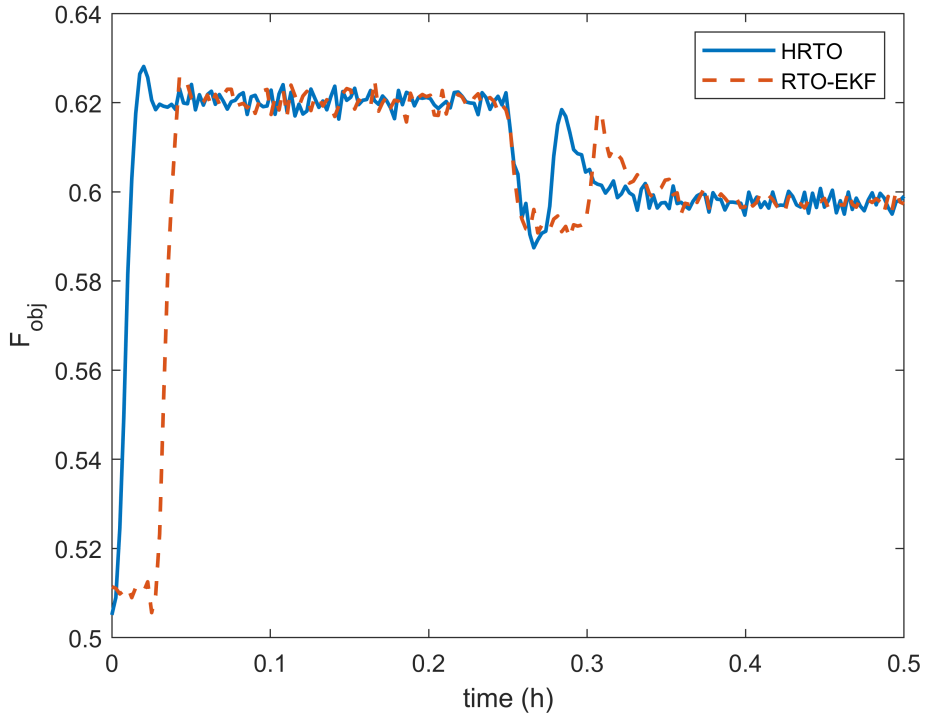

Figure 5.66: HRTO and RTO-EKF scenario 2 - objective function evaluation.

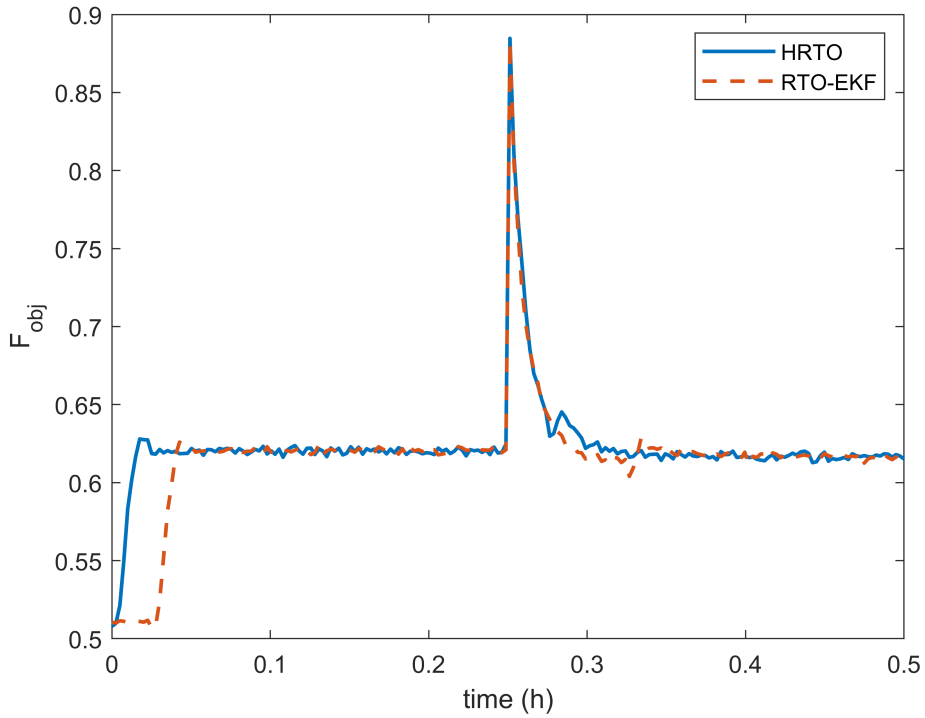

Figure 5.67: HRTO and RTO-EKF scenario 3 - objective function evaluation.

In all scenarios it can be noted that the RTO-EKF presents a moderate delay

thanks to the steady-state detection step.

In order to better evaluate RTO and HRTO performance in transient operation, the area bellow the curve in theses regions was calculated by trapezoidal numerical integration and compared with the adopted reference, the DRTO trajectory simulated in the beginning of this chapter.

Thus, defining a performance coefficient which is the ratio between the area bellow the transient operation curve for a given approach and the same area for the DRTO simulation, their values are given in Table 5.6:

Table 5.6 shows the results for *sigma* for both methods in each scenario:

Table 5.6: Transient operation performance.

Scenario	RTO-EKF	HRTO
1	0.6399	0.6399
2	0.7197	0.7270
3	0.6834	0.6905

This shows that both approaches performed similarly, with a small edge for the HRTO. This sums up with the fact that the hybrid approach outperforms the standard RTO most of steady-state operation due to the steady-state waiting time.

Chapter 6

Conclusions

In this work the hybrid RTO approach was applied to the Van de Vusse CSTR. This method mainly consists of using the dynamic model in the model update step, while the corresponding steady-state model is then used in the optimization step, therefore making the steady-state detection step no longer necessary.

Regarding transient operation, the DRTO is still expected to outperform its counterparts. This result is showed by numerically integrating the cost function curve while in transient operation and using this value for the DRTO as a reference. The performance ratios were of the order of 0.63-0.73, showing a substantial difference when compared with the optimal trajectory from the DRTO.

This considerable deviation is a consequence of the Van de Vusse CSTR complex dynamics. However, the several challenges associated with implementing dynamic optimization in real time makes it the last alternative considered in real processes.

By comparing the cost function profile for the RTO and HRTO its was shown similar behavior during most of the steady-state operation, with close performances. The HRTO stands out especially thanks to the absence of the steady-state detection step, whose makes standard RTO have a delayed performance at some points.

The main remark that can be taken from this work is the successful use of the HRTO for a benchmark system, what enhances the possibilities for this hybrid approach in industrial applications.

Possible future research directions that can be taken from this point are:

- Applying the HRTO to industrial large-scale systems: as commented during the literature revision, the HRTO was only used for relatively small systems. Its performance has yet to be stated for systems containing thousand of variables, while making the comparison with traditional RTO;
- Using transient measurements in model-free methods for MIMO sys-

tems: currently in literature there are different model-free gradient estimation schemes, but these were mainly applied to SISO systems (KRISHNAMOORTHY, 2019), there being a clear lack of model-free gradient estimation methods for large-scale multivariable processes.

- Hybrid RTO while applying metamodels: as stated before, this hybrid approach is yet to be evaluated in large-scale systems. Usually a real process has so many equations and variables that the use of metamodels is a necessity. Therefor, one possible field that can be explored is applying HRTO to a data-based process model, like Kriging models or models based on neural networks.

Bibliography

AMRIT, R., RAWLINGS, J. B., ANGELI, D., 2011, "Economic optimization using model predictive control with a terminal cost", *Annual Reviews in Control*, v. 35, n. 2, pp. 178–186.

BERNARDINO, L. F., 2019, *Health-aware control and model-based prognostics of a subsea oil and gas separation system*. Master Thesis, Universidade Federal do Rio de Janeiro.

BETTI, G., FARINA, M., SCATTOLONI, R., 2012, "An MPC algorithm for offset-free tracking of constant reference signals", *Proceedings of 51st IEEE Conference on Decision and Control*.

CAMACHO, E. F., BORDONS, C., 2004, "Control predictivo: Pasado, presente y futuro", *Revista Iberoamericana de Automatica e Informatica Industrial*, v. 1, n. 3, pp. 5–28.

CAMACHO, E. F., BORDONS, C., 2007, *Model predictive control*. London: Springer Verlag.

CAMPOS, M., H., T., LIPORACE, F., et al., 2009, "Challenges and problems with advanced control and optimization technologies", *IFAC Proceedings Volumes*, v. 42, n. 11, pp. 1–8.

CAO, S., RHINEHART, R. R., 1995, "An efficient method for on-line identification of steady state", *J. Process Control*, v. 5, pp. 363–374.

CHEN, C. Y., JOSEPH, B., 1987, "On-line optimization using a two-phase approach: an application study", *Ind. Eng. Chem. Res.*, v. 26, n. 9, pp. 1924–1930.

CHEN, X., HEIDARINEJAD, M., LIUA, J., et al., 2012, "Distributed economic MPC: Application to a nonlinear chemical process network", *Journal of Process Control*, v. 22, n. 4, pp. 689–699.

CUNHA NETO, J. V., 2005, *Controle e introdução ao diagnóstico de um reator químico*. Master Thesis, Universidade Federal do Rio de Janeiro.

CUTLER, C. R., REMARKER, B. L., 1980, "Dynamic matrix control - A computer control algorithm", *Joint automatic control conference*.

DARBY, M. L., NIKOLAOU, M., JONES, J., et al., 2011, "RTO: An overview and assessment of current practice", *Journal of Process Control*, v. 21, n. 6, pp. 874–884.

DE AVILA FERREIRA, T., FRANÇOIS, G., MARCHETTI, A. G., et al., 2017, "Use of transient measurements for static real-time optimization", *IFAC-PapersOnLine*, v. 50, n. 1, pp. 5737–5742.

DOS SANTOS, J. E. S., 2007, *Controle Preditivo N ao-Linear para Sistemas de Hammerstein*. Ph.D. Thesis, Universidade Federal de Santa Catarina.

ENGELL, S., KLATT, K. U., 1993, "Nonlinear Control of a Non-Minimum-Phase CSTR", *Proc. Of Control Conference,*, pp. 2041–2045.

FORBES, J. F., MARLIN, T. E., MACGREGOR, J. F., 1994, "Model adequacy requirements for optimizing plant operations", *Computers & Chemical Engineering*, v. 18, n. 6, pp. 497–510.

FRANÇOIS, G., BONVIN, D., 2014, "Use of transient measurements for the optimization of steady-state performance via modifier adaptation", *Industrial & Engineering Chemistry Research*, v. 53, n. 13, pp. 5148–5159.

GARCIA, C. E., MORSHEDI, A., 1986a, "Quadratic programming solution of dynamic matrix control (QDMC)", *Chemical Engineering Communications*, v. 46, n. 1-3, pp. 73–87.

GOLSHAN, M., PISHVAIE, M. R., BOOZARJOMEHRY, R. B., 2008, "Stochastic and global real time optimization of Tennessee Eastman challenge problem", *Engineering Applications of Artificial Intelligence*, v. 21, n. 2, pp. 215–228.

GONZÁLEZ, A. H., ADAM, E. J., MARCHETTI, J. L., 2008, "Conditions for offset elimination in state space receding horizon controllers: A tutorial analysis", *Chem. Engng and Proc.: Process Intensification*, v. 47, n. 12, pp. 2184–2194.

HELBIG, A., ABEL, O., MARQUARDT, W., 2000, "Structural Concepts for Optimization Based Control of Transient Processes", *Nonlinear Model Predictive Control*, v. 26, pp. 295–311.

HENSON, M. A., 1998, "Nonlinear model predictive control: current status and future directions", *Computers & Chemical Engineering*, v. 23, n. 2, pp. 187–202.

HINDMARSH, A. C., BROWN, P. N., GRANT, K. E., et al., 2005, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers", *ACM Transactions on Mathematical Software (TOMS)*, v. 31, n. 3, pp. 363–396.

HUANG, R., HARINATH, E., BIEGLER, L. T., 2011, "Lyapunov stability of economically oriented NMPC for cyclic processes", *Journal of Process Control*, v. 21, n. 4, pp. 501–509.

JAMALUDIN, M. Z., SWARTZ, C. L. E., 2017, "Dynamic Real-Time Optimization with Closed-Loop Prediction", *Process Systems Engineering*.

KALMAN, R. E., OTHERS, 1960, "Contributions to the theory of optimal control", *Bol. soc. mat. mexicana*, v. 5, n. 2, pp. 102–119.

KALMAN, R. E., 1960, "A New Approach to Linear Filtering and Prediction Problems", *Transactions of the ASME–Journal of Basic Engineering*, v. 82, n. Series D, pp. 35–45.

KRISHNAMOORTHY, D., 2019, *Novel Approaches to Online Process Optimization Under Uncertainty - Addressing the limitations of current industrial practice*. Ph.D. Thesis, Norwegian University of Science and Tecnology.

KRISHNAMOORTHY, D., FOSS, B., SKOGESTAD, S., 2018, "Steady-state real-time optimization using transient measurements", *Computers & Chemical Engineering*, v. 115, pp. 34–45.

LI, J., RHINEHART, R. R., 1998, "Heuristic random optimization", *Computers & chemical engineering*, v. 22, n. 3, pp. 427–444.

MACIEJOWSKI, J. M., 2002, *Predictive control: with constraints*. Pearson education.

MAGALHÃES, O. I. D. B., 2010, *Desenvolvimento de um Sistema de Otimização Dinâmica em Tempo Real*. Master Thesis, Universidade Federal do Rio de Janeiro.

MARLIN, T. E., HRYMAK, A. N., 1997, "Real-time operations optimization of continuous processes", *AIChE Symposium Series*, v. 93, pp. 156–164.

MATIAS, J., JASCHKE, J., 2019, "Online model maintenance via output modifier adaptation", *Industrial & Engineering Chemistry Research*, v. 58, n. 30, pp. 13750–13766.

MATIAS, J. O., LE ROUX, G. A., 2018, "Real-time Optimization with persistent parameter adaptation using online parameter estimation", *Journal of Process Control*, v. 68, pp. 195–204.

MERCANGÖZ, M., DOYLE III, F. J., 2008, "Real-time optimization of the pulp mill benchmark problem", *Computers & Chemical Engineering*, v. 32, n. 4-5, pp. 789–804.

NAYSMITH, M., DOUGLAS, P., 1995, "Review of real time optimization in the chemical process industries", *Asia-Pacific Journal of Chemical Engineering*, v. 3, n. 2, pp. 67–87.

OGATA, K., 2010. "Engenharia de controle moderno. Tradução de Heloísa Coimbra de Souza". .

OGUNNAIKE, B. A., RAY, W. H., 1994, *Process Dynamics, Modelling and Control*. Oxford University Press; 1 edition.

PANNOCCHIA, G., RAWLINGS, J. B., 2001, "The Velocity Algorithm LQR: a survey", *Texas-Wisconsin Modeling and Control Consortium*.

PHAM, D. T., VERRON, J., ROUBAUD, M. C., 1998, "A singular evolutive extended Kalman filter for data assimilation in oceanography", *Journal of Marine systems*, v. 16, n. 3-4, pp. 323–340.

QIN, S. J., BADGWELL, T. A., 1997, "An overview of industrial model predictive control technology", *AIChE Symp. Ser.*, v. 316, n. 93, pp. 232.

QIN, S. J., BADGWELL, T. A., 2000, "An Overview of Nonlinear Model Predictive Control Applications", *Progress in Systems and Control Theory*, v. 26.

QUELHAS, A. D., DE JESUS, N. J. C., PINTO, J. C., 2012, "Common Vulnerabilities of RTO Implementations in Real Chemical Processes", *The Canadian Journal of Chemical Engineering*.

RAO, C. V., RAWLINGS, J. B., 1999, "Steady States and Constraints in Model Predictive Control", *Process Systems Engineering*.

RAO, S. S., 2019, *Engineering optimization: theory and practice*. John Wiley & Sons.

RICHALET, J., RAULT, A., TESTUD, J., 1978, "Model predictive heuristic control", *Automatica*, v. 14, n. 5, pp. 413–428.

ROBERTS, P. D., 1978, "Algorithms for integrated system optimisation and parameter estimation", *Electronics Letters*, v. 14, n. 6, pp. 196–197.

ROTAVA, O., ZANIN, A., 2005, "Multivariable control and real-time optimization–an industrial practical view: here's what can be achieved with each technology and how they are integrated at this plant", *Hydrocarbon Processing*, v. 84, n. 6, pp. 61–68.

RUIZ, C. A., 2009, "Real Time Industrial Process Systems: Experiences from the Field". In: *Computer Aided Chemical Engineering*, v. 27, Elsevier, pp. 133–138.

SEBORG, D. E., MELLICHAMP, D. A., F., E. T., 2010, *Process dynamics and control*. John Wiley & Sons.

SEQUEIRA, S. E., GRAELLS, M., PUIGJANER, L., 2002, "Real-time evolution for on-line optimization of continuous processes", *Industrial & engineering chemistry research*, v. 41, n. 7, pp. 1815–1825.

SIMON, D., 2006, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons.

SKOGESTAD, S., POSTLETHWAITE, I., 2007, *Multivariable feedback control: analysis and design*, v. 2. Wiley New York.

SORENSEN, R. C., CUTLER, C. R., 1998, "LP integrates economics into dynamic matrix control", *Hydrocarbon Process Sep.*, v. 57.

THOMAZ, D. M., 2017, *Estratégia de Controle Preditivo Multivariável para um Sistema de Compressão de Gás de Plataforma do Pré-sal*. Master Thesis, Universidade Federal do Rio de Janeiro.

TRIERWEILER, J. O., 2014, "Real-Time Optimization of Industrial Processes", *Encyclopedia of Systems and Control*, p. 1132–1141.

VAN DE VUSSE, J. G., 1962, "A new model for the stirred tank reactor", *Chemical Engineering Science*, v. 17, n. 7, pp. 507–521.

WÄCHTER, A., BIEGLER, L. T., 2006, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming", *Mathematical programming*, v. 106, n. 1, pp. 25–57.

WANG, L., 2004, "A tutorial on model predictive control: Using a linear velocity-form model", *Dev. Chem. Eng. Mineral Process.*, v. 12, n. 5/6, pp. 573–614.

WHITE, D. C., 1998, "On-line Optimization: What Have We Learned?" *Hydrocarbon Process*, v. 55.

YING, C. M., JOSEPH, B., 1999, "Performance and Stability Analysis of LP-MPC and QP-MPC Cascade Control Systems", *AIChE J.*, v. 45, n. 7, pp. 1521.

YOUNG, R. E., 2006, "Petroleum refining process control and real-time optimization", *IEEE Control Systems Magazine*, v. 26, n. 6, pp. 73–83.