

Politecnico di Milano

School of Industrial and Information Engineering
Master of Science degree in Automation and Control
engineering

Design of a nonlinear Model Predictive Control algorithm with symbolic computations

Supervisor
Professor Riccardo Scattolini

Master of science dissertation of
Stefano Luca Brambilla
Matricola: **836974**



Academic Year 2015/2016

Contents

1	Introduction	1
1.1	Classic approach to MPC	1
1.1.1	Limitations of Linear MPC	2
1.2	Trajectory MPC	3
1.3	Objective of the thesis	4
1.4	Thesis organization	4
2	Model of the system	5
2.1	Variables definition	7
2.2	Model equations	8
2.3	Open loop response	12
2.4	Level control	15
3	Linear MPC	21
3.1	Optimization problem	22
3.2	Tuning of the MPC	23
3.3	Kalman Observer	24
3.4	Results	24
3.4.1	Re-boiler reference temperature	25
3.4.2	Condenser reference temperature	26
3.4.3	Response to the disturbances	27
4	Trajectory MPC	31
4.1	Control algorithm	31
4.2	Control scheme implementation	33
4.2.1	Linearization around the nominal trajectory	35
4.2.2	Optimization problem solution	36
4.3	Challenges and limitations of the implementation	37
4.4	Results	38
4.4.1	Condenser reference temperature	38
4.4.2	Re-boiler reference temperature	38
4.4.3	Response to the disturbances	38
5	Conclusion	45

List of Figures

1.1	Hierarchy of control system in a typical process plant with MPC structure	2
1.2	Trajectory MPC conceptual scheme	3
2.1	Reactor-distillation process network	5
2.2	Continuous stirred tank reactor	6
2.3	Trays example	7
2.4	Trays states: steady-state values	13
2.5	Temperature response to a 5% increase of Q_{cstr}	14
2.6	T_{cstr} response to a 5% increase of Q_{cstr}	14
2.7	Temperature response to a 5% increase of Q_{cond}	15
2.8	Temperature response to a 5% increase of Q_{reb}	16
2.9	T_{cstr} response to $\pm 5\%Q_{cstr}$	16
2.10	Response to a 10% positive step variation of F	17
2.11	Response to a 10% positive step variation of D	18
2.12	Response to a 10% positive step variation of V	19
2.13	Non-linear behavior of T_{cond} and T_{reb}	20
3.1	MPC control scheme with integral action	21
3.2	Linear MPC control scheme	24
3.3	T_{reb} response to a step variation of T_{reb}^o	25
3.4	Input increments due to a step in T_{reb}^o	25
3.5	T_{cstr} response to a step variation of T_{reb}^o	26
3.6	T_{cond} response to a step variation of T_{reb}^o	26
3.7	T_{cond} response to a step variation of T_{cond}^o	27
3.8	Input increments due to a step in T_{cond}^o	27
3.9	T_{cstr} response to a step variation of T_{cond}^o	28
3.10	T_{reb} response to a step variation of T_{cond}^o	28
3.11	T_{cstr} response to a step variation of M_{cond}	29
3.12	T_{cond} response to a step variation of M_{cond}	29
3.13	T_{reb} response to a step variation of M_{cond}	30
3.14	Input increments due to a step in M_{cond}	30
4.1	Trajectory MPC control scheme with integral action	34
4.2	Nominal trajectory computation scheme	36
4.3	Nominal trajectory computation step	37
4.4	Comparison of the T_{cond} responses to a step variation of T_{cond}^o	39
4.5	Input increments due to a step in T_{cond}^o	39
4.6	Comparison of the T_{cstr} responses to a step variation of T_{cond}^o	40
4.7	Comparison of the T_{reb} responses to a step variation of T_{cond}^o	40
4.8	Comparison of the T_{reb} responses to a step variation of T_{reb}^o	41

4.9	Input increments due to a step in T_{reb}^o	41
4.10	Comparison of the T_{cstr} responses to a step variation of T_{reb}^o	42
4.11	Comparison of the T_{cond} responses to a step variation of T_{reb}^o	42
4.12	Comparison of the T_{cstr} responses to a step variation of M_{cond}	43
4.13	Comparison of the T_{cond} responses to a step variation of M_{cond}	43
4.14	Comparison of the T_{reb} responses to a step variation of M_{cond}	44
4.15	Input increments due to a step in M_{cond}	44

List of Tables

2.1	Process states	8
2.2	Process inputs and outputs	8
2.3	CSTR parameters	9
2.4	Process parameters	11
2.5	Process parameters values	11
2.6	Model parameters	11
2.7	Equilibrium state inputs	12
2.8	Steady state values	12
2.9	Temperature variations due to a 5% heat/coolant variation	13

Abstract

This thesis focuses on implementing a nonlinear Model Predictive Control algorithm based on a time-varying model of the system. A systematic procedure to implement the control algorithm has been developed using the Matlab[®] Symbolic Toolbox. The controller has been implemented to perform the temperature control in a distillation process system. A second MPC controller based on the linear model of the system has been implemented and the results obtained with the two controllers have been compared.

Abstract

Lo scopo di questa tesi è lo sviluppo di una procedura sistematica per l'implementazione di un controllore Model Predictive Control non lineare utilizzando un software per effettuare calcoli simbolici. Il sistema su cui è stato implementato il controllore MPC è un processo costituito da un reattore e una torre di distillazione collegati in feedback. Il sistema in questione è stato scelto per la complessità del suo modello e per la sua ampia diffusione nell'ambito dell'industria di processo. Il controllore MPC è utilizzato per controllare i valori delle temperature nell'impianto. Il controllore implementato si basa su un modello tempo variante dell'impianto, ottenuto linearizzando il sistema attorno alla traiettoria nominale degli stati. Tale traiettoria è l'evoluzione degli stati del sistema ottenuta simulando il modello non lineare del sistema stesso utilizzando gli ingressi ottimali calcolati all'istante di tempo precedente. Il calcolo delle matrici simboliche del sistema linearizzato è stato effettuato utilizzando il Toolbox Simbolico di Matlab[®]. Per confrontare i risultati ottenuti con il controllore MPC non lineare, un controllore basato sull'algoritmo Model Predictive Control lineare è stato a sua volta implementato sul sistema. Per questo controllore si è supposto di non avere a disposizione la misura degli stati. La stima del valore degli stati è ottenuta con un filtro di Kalman. Il secondo controllore è stato implementato allo scopo di avere un riferimento su cui confrontare i risultati ottenuti con il controllore non lineare.

Chapter 1

Introduction

The Model Predictive Control, MPC in short, is a family of advanced control algorithms that uses an explicit process model to predict the future response of a plant. The algorithm attempts, at each time step, to optimize the future plant behavior by computing a sequence of the manipulated input variables. Only the first input of the optimal sequence is applied to the system and a new sequence is computed at the next time step, this principle is called Receding Horizon. This type of algorithm has had a huge impact in the industrial field since its development at the beginning of the 80s. It was originally developed for the petroleum refineries and power plants, but nowadays it is widely used in all the process industry and some applications can be found in other engineering fields. An in depth study on the MPC history and development can be found in [3], a survey of its applications in the industrial field can be found in [4]. To understand the reasons behind its popularity it is important to list the main characteristics of the MPC algorithms. The main advantage of Model Predictive Control algorithms is the possibility to formulate the control problem as an optimization one, with the additional possibility to define constraints on state and input variables. Another important factor, is the possibility to use process models derived empirically, by means of simple experiments on the plant, in the controller synthesis. This characteristics make MPC algorithms very flexible and suitable for many different applications. In the hierarchy of the control system, model predictive controllers are usually applied at an intermediate level, as shown in Figure 1.1, between the real time optimization layer and the low level control. The real time optimization layer computes the optimal operating conditions of the overall plant based on a static model of the plant, while the MPC level task is to compute dynamically the reference signals for the lower level control. At the lower level the actuators are controlled at a higher frequency than the upper levels.

1.1 Classic approach to MPC

The most classic definition of the MPC algorithm is the version based on a linear discrete time system in state space form. The model describing this type of systems is

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases} \quad (1.1)$$

The state $x \in R^n$ is assumed to be measurable, $u \in R^m$ is the control variable and $y \in R^p$ is the output variable. As stated in the previous paragraph the algorithm solves an optimization problem at each time step k . The optimization problem consists of computing the optimal control sequence $u(k), u(k+1), \dots, u(k+N-1)$ that minimizes the finite horizon quadratic cost

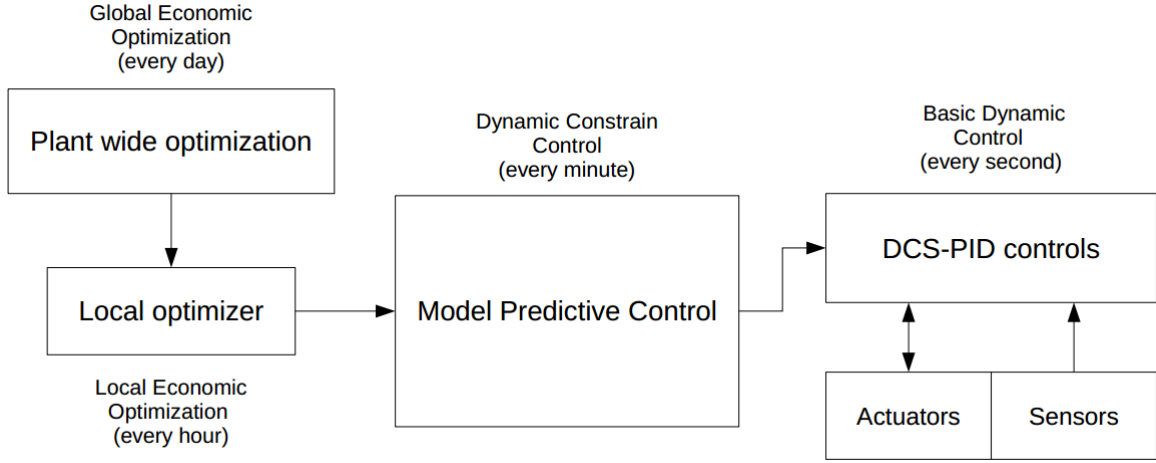


Figure 1.1: Hierarchy of control system in a typical process plant with MPC structure

function

$$J(x(k), u(\cdot), k) = \sum_{i=0}^{N_p-1} \left(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right) + \|x(k+N_p)\|_S^2 \quad (1.2)$$

where $Q = Q' \geq 0$, $R = R' > 0$, $S = S' \geq 0$ are matrices of suitable dimension. The positive integer N_p is called prediction horizon and defines the amplitude of the window over which the prediction of the system evolution is performed. In the solution of the optimization problem, it is easy to consider constraints on the future state, input and output variables, in the form

$$\begin{aligned} u_{min} &\leq u(k+i) \leq u_{max}, & i = 0, 1, \dots, N_p - 1 \\ x_{min} &\leq x(k+i) \leq x_{max}, & i = 1, \dots, N_p \\ y_{min} &\leq y(k+i) \leq y_{max}, & i = 1, \dots, N_p \end{aligned}$$

The solution of the optimization problem is the optimal control sequence

$$\vec{u}^o(k : k + N_p - 1 | k) = [u^o(k|k) \dots u^o(k + N_p - 2 | k) \quad u^o(k + N_p - 1 | k)]$$

According to the Receding Horizon principle only the first element is applied to the system $u(k) = u^o(k|k)$. At the next time step $k + 1$, a new optimization control is solved based on the values of the states and inputs at time k .

1.1.1 Limitations of Linear MPC

The classic linear MPC algorithm presents mainly two limitations: the first one is related to the prediction horizon N_p , while the second is inherited to the model considered for the system. To include all the process dynamics in the prediction horizon a large value for N_p is required, but as the horizon increases also the number of vector optimization variables $u(k+i)$ increases, thus leading to higher computational load for the solution of the optimization problem. To mitigate this effect, a control horizon $0 < N_u < N_p$ can be defined. A new set of constraints is applied to the control problem in the form

$$u(k+i) = u(k+i-1), \quad i = N_u, \dots, N_p - 1$$

The smaller the control horizon N_u is chosen, the higher is the reduction in computational costs for the algorithm. However as N_u decreases a less aggressive control action is obtained, an effect that can be desirable or not, depending on the particular case. A typical choice in the industrial field is called Mean level control and is based on the choice to set $N_u = 1$. With this choice the control variable is set equal on all the prediction horizon, but due to the Receding Horizon strategy, the actual control variable changes at each time step. The second limitation is related to the disadvantages of linear models for the description of complex systems. Linear models are useful to simplify the definition and solution of control problems, but are approximations of the actual behavior of the system. In the Model Predictive Control field, this approximation leads to possible mismatch between the predicted evolution of the system and the actual behavior of the state and output variables. A possible solution to this problem would be to use a nonlinear model of the system, but this approach has two main caveats. The definition of the optimization problem becomes more complex and to find a solution for this optimization problems is not always feasible in some cases. Additionally the computational load using a nonlinear model is even higher than the linear model case. A possible solution to the issues related to the use of a nonlinear model is presented in the next paragraph.

1.2 Trajectory MPC

The main idea behind this MPC algorithm is to compute at each time step the nominal trajectory of the states of the system during the prediction horizon, to linearize the system around this trajectory and to use the obtained model to compute the optimal control sequence at the next time step. The optimization problem solved with this algorithm is still in the form of the Linear MPC case, but the model used is a time varying linear system. To further improve the prediction of the system trajectory, the linearization error is considered in the optimization problem. The

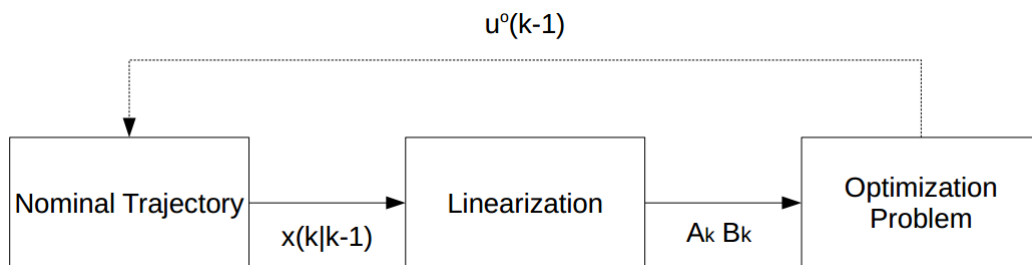


Figure 1.2: Trajectory MPC conceptual scheme

main advantage of this algorithm, with respect to the Linear MPC, is better prediction capability, at the expense of heavier computational load. The main issue of the Trajectory MPC algorithm is the difficulty in the definition of the optimization problem, as it will be shown in Chapter 4. The development for a different application of this control algorithm can be found in [5]. The MPC algorithm based on the linearization of the model around the nominal trajectory is a very powerful control algorithm, but due to the complexity of the algorithm, there are no systematic procedures to realize this control scheme. For this reason it is not a simple task to realize this type of controller for complex systems.

1.3 Objective of the thesis

In this thesis a Trajectory MPC control scheme is realized for a complex plant model using Matlab[®] Simulink software. This thesis aims to find a systematic procedure to realize this control algorithm. In order to achieve the main goal, the following guidelines has been followed

- procedure should not depend on model size and structure
- realize the on-line linearization using software tools
- the controller structure should not depend on the control parameters

The second goal of this thesis is to compare the results obtained with a traditional linear MPC to the ones obtained with the trajectory MPC.

1.4 Thesis organization

In the following chapter, the system considered in this thesis for the control is presented. First a description of the plant is given and the model equations are described. Then the initial equilibrium conditions of the model are defined and the open loop input/output responses of the model are analyzed. In chapter 3, a traditional linear Model Predictive Controller is designed for the system. The MPC algorithm used utilizes an integrator to ensure reference tracking. The procedure to define and solve the optimization problem is presented, an explanation on the choice for the control parameters is given and the most significant results obtained with the controller are presented. Chapter 4 presents the Trajectory MPC algorithm. The procedure used to realize the controller is described and the problems encountered are stated. A comparison between the results obtained with this control algorithm and the traditional MPC is given at the end of the chapter. The final considerations on the results obtained and some possible development of this work are presented in chapter 5.

Chapter 2

Model of the system

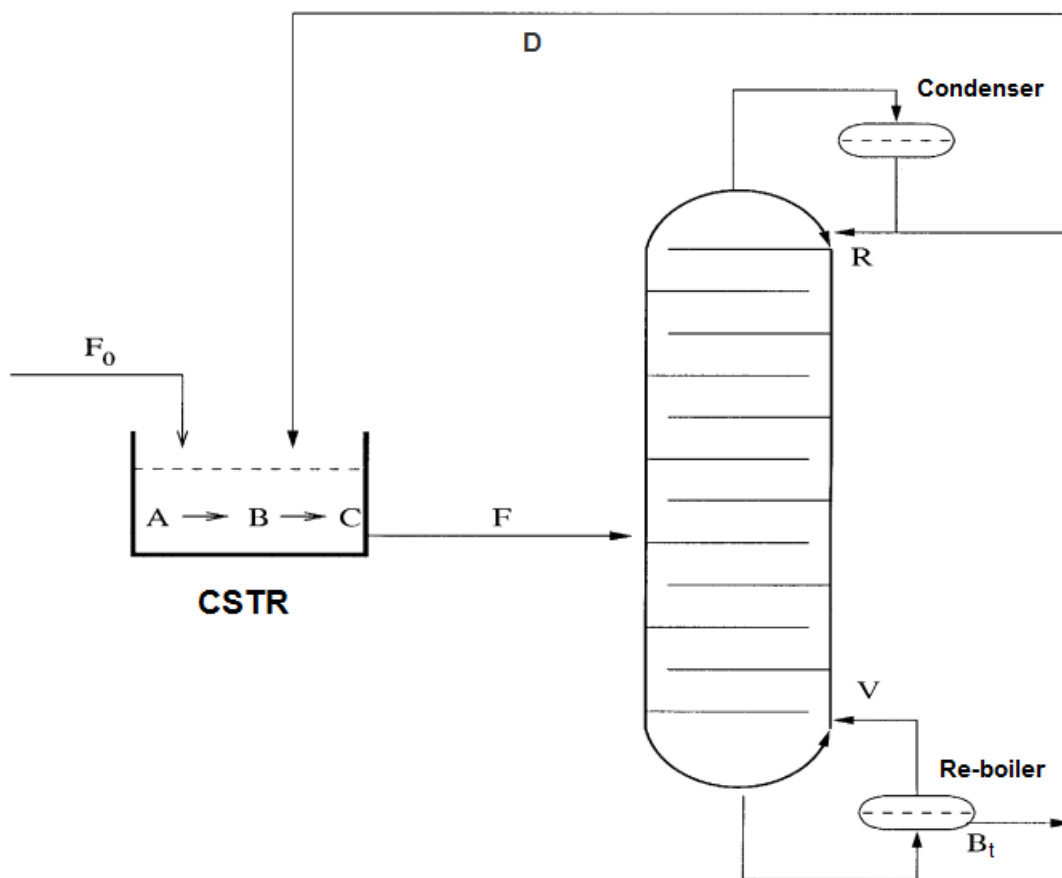


Figure 2.1: Reactor-distillation process network

The system considered in this thesis is a reactor-distillation process network, shown in figure 2.1. The system is described in the articles [1] and [2]. This type of process is widely used in the chemical process industry when one or more components are generated from chemical reactions and the resulting multi-component mixture has to be separated. It consists of a continuously stirred tank reactor (CSTR), where reactions take place, and a distillation column, also called distillation tower, whose goal is to separate the mixture. In the scheme considered they are connected through a recycle loop, feed D in the figure, connecting the outlet coming from the

top of the distillation tower to the reactor.

Continuous stirred tank reactors are the most standard of the continuous reactors used in chemical processes. CSTR are open systems, where material is free to enter or exit. They operate on a steady-state basis, so conditions in the reactor don't change with time. CSTR are composed of a tank, usually of constant volume, and a stirring system to mix reactants together. In the considered system, two exothermic reactions, $A \xrightarrow{k_1} B \xrightarrow{k_2} C$, take place inside the reactor. Component A is the reactant, B is the desired product, while C is the by-product. Reactor is fed with a fresh feed of component A at flow-rate F_0 . The outlet of the reactor, feed F , is sent to the distillation column entering from the so called feed tray.

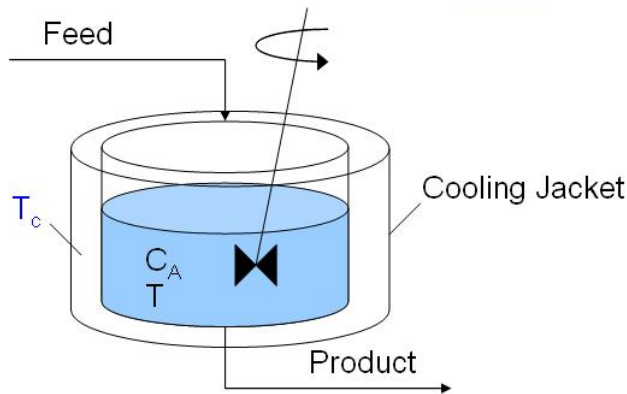


Figure 2.2: Continuous stirred tank reactor

Industrial distillation columns are usually large, vertical cylindrical vessels with diameters ranging from about sixty centimeters to a dozen of meters, and variable heights from some meters to sixty or more. Inside the tower a feed stream is separated into two fractions, an overhead distillate product and a bottom product. The "lightest" products, those with the lowest boiling point or highest volatility, exit from the top of the column and the "heaviest" products, those with the highest boiling point, exit from the bottom of the column. At the top of the column a condenser is used to condense the vapor stream produced by the tower, while at the bottom a re-boiler heats the mixture to further separate the components. To improve the separation, distillation towers are normally made by horizontal plates or trays, and reflux is used to recycle part of the overhead liquid product inside the top of the tower. Trays are devices used to provide the required number of equilibrium stages. The stage at the tower bottom has the highest pressure and temperature. Progressing upwards in the tower, pressure and temperature decrease for each succeeding stage. Due to the fact that each component reacts differently to pressure and temperature conditions, separation of the components inside the column occurs. Figure 2.3 shows an example of two trays inside a distillation tower, liquid flows downwards into the tower, stopping at each tray, while vapor rises passing through the bubble caps and partially condensing. Continuous distillation is usually performed at steady-state. Unless the process is disturbed due to changes in feed flow-rate or composition, heating, condensing or ambient temperature, steady-state is normally maintained inside the column. This feature makes distillation columns very popular in the industrial field, because if the feed rate and composition are kept constant, product rate and quality are also constant.

In the scheme considered in this work, the distillation column has internally 15 trays, the feed tray corresponding to tray number 12. Component B of the mixture has the highest boiling point, while A the lowest. Therefore the bottom outlet of the tower, the bottom product B_t is mainly composed of the desired product B , while the upper outlet D is mainly composed of A .

The outlet coming from the condenser is fed back into the CSTR as an additional input.

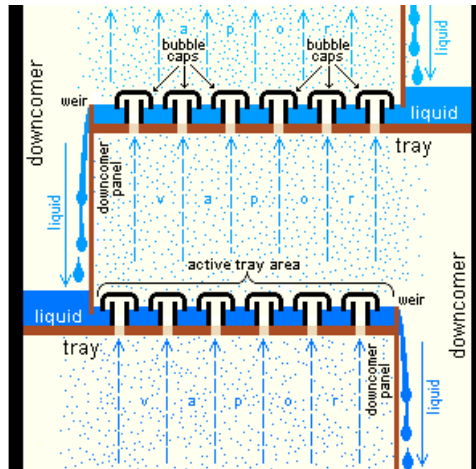


Figure 2.3: Trays example

Reasons behind the choice of this system in the context of this work are multiple. First of all, as stated before, this type of process network are common in the chemical engineering field where MPC control is widely used. Another reason is that, due to the internal feedback generated by the feed D and the high nonlinearity of the model, shown in following paragraph 2.2, the control of the system is not trivial and requires an advanced control technique. Therefore, the system represents a challenging workbench for for the analysis and synthesis of Model Predictive Control algorithms

2.1 Variables definition

The dynamic model of the system is based on energy and material balances under standard modeling assumptions. In particular the model of the reactor assumes perfect mixing and spatially uniform heat conduction, the mixture flowing outside the reactor has thus the same composition and enthalpy of the mixture inside. Reactions inside the CSTR are first order elementary reactions. To obtain the dynamic model of the multicomponent distillation, stage-by-stage methods and batch rectification are applied. It is assumed vapour-liquid equilibrium in each tray, perfect mixing of liquid and vapour, negligible vapour holdup, constant molar-liquid holdup on each tray, and adiabatic process for all the entire distillation process. Thermodynamic properties of the mixtures are obtained assuming ideal behavior in both liquid and vapour phase. The obtained model has 74 states, corresponding to the liquid holdups in CSTR, condenser and re-boiler. Enthalpy of liquid mixtures and liquid-phase species compositions are states for each considered element of the system, namely CSTR, condenser, re-boiler and each tray. In particular for the reactor, the composition of component C is not a state, leading to the reactor having only four states, while each other component of the system is described by five states. A list of all the states with their symbols used in the model equations can be found in Table 2.1.

Table 2.1: Process states

M_R	CSTR liquid holdup	[mol]
M_0	condenser liquid holdup	[mol]
M_{N+1}	re-boiler liquid holdup	[mol]
$H_{L,R}$	Enthalpy of the liquid mixture in the CSTR	[J/mol]
$H_{L,0}$	Enthalpy of the liquid mixture in condenser	[J/mol]
$H_{L,j}$	Enthalpy of the liquid mixture in tray number j	[J/mol]
$H_{L,N+1}$	Enthalpy of the liquid mixture in re-boiler	[J/mol]
$x_{A,R}$	Species A composition inside CSTR	
$x_{B,R}$	Species B composition inside CSTR	
$x_{i,0}$	Species i composition inside condenser	
$x_{i,j}$	Species i composition inside tray number j	
$x_{i,N+1}$	Species i composition inside re-boiler	

The considered inputs are the feed flow-rate for the tower, the feedback flow-rate flowing into the CSTR from the top of column and the vapor flow-rate inside the distillation column. Additional inputs are the external coolant/heat inputs of CSTR, condenser and re-boiler. Flow-rates for the input feed stream of the CSTR and bottom outlet of the column are considered as constant parameters for the system. Outputs of interest for the system are the liquid holdups inside the reactor, condenser and re-boiler, and their respective temperatures.

Table 2.2: Process inputs and outputs

D	Internal feedback flow-rate
F	Column feed flow-rate
V	Vapor flow-rate inside column
Q_{cstr}	External heat input to the CSTR
Q_{cond}	External heat input to the condenser
Q_{reb}	External heat input to the re-boiler
M_R, M_0, M_{N+1}	Liquid hold-ups
T_{cstr}	Temperature of the CSTR
T_{cond}	Temperature of the condenser
T_{reb}	Temperature of the re-boiler

2.2 Model equations

In this paragraph model equations are presented, starting from the models of the CSTR and distillation tower. Then the relationship between the different thermodynamic properties of the species are developed from the assumptions presented in the previous paragraph. The process parameter values are collected at the end of the paragraph in Table 2.6. According to the assumptions made previously, the resulting dynamic model of the CSTR is

$$\dot{M}_R = F_0 + D - F \quad (2.1)$$

$$\dot{x}_{A,R} = \frac{F_0(1 - x_{A,R}) + D(x_{A,0} - x_{A,R})}{M_R} - k_1 e^{-E_1/R_g T_{cstr}} x_{A,R} \quad (2.2)$$

$$\dot{x}_{B,R} = \frac{-F_0 x_{B,R} + D(x_{B,0} - x_{B,R})}{M_R} + k_1 e^{-E_1/R_g T_{cstr}} x_{A,R} - k_2 e^{-E_2/R_g T_{cstr}} x_{B,R} \quad (2.3)$$

$$\begin{aligned} \dot{H}_{L,R} = & \frac{F_0(H_{L,F_0} - H_{L,R}) + D(H_{L,0} - H_{L,R})}{M_R} + \frac{Q_{cstr}}{M_R} + \\ & - k_1 e^{-E_1/R_g T_{cstr}} x_{A,R} \Delta H_{r1} - k_2 e^{-E_2/R_g T_{cstr}} x_{B,R} \Delta H_{r2} \end{aligned} \quad (2.4)$$

The first equation is obtained from mass balance, it states that the variation of the liquid holdup inside the reactor depends linearly on the input and output flow-rates. Equations (2.2) and (2.3) represent the variation of the species composition for components A and B respectively. They are composed by a linear part and a nonlinear one. The linear part corresponds to the variations caused by the different mixture compositions flowing into the reactor, while the nonlinear ones are the consequence of the variation due to the reactions developed inside. The composition for component C is obtained through the following relationship $x_{A,R} + x_{B,R} + x_{C,R} = 1$. Similarly to previous equations, equation (2.4) is composed by a linear and a nonlinear part, both corresponding to the same causes. It is important to note that the heat input Q_{cstr} is linearly related to the value of $\dot{H}_{L,R}$. Table 2.3 contains the meaning of the different parameters appearing in the equations.

Table 2.3: CSTR parameters

$\Delta H_{r1}, \Delta H_{r2}$	Heat of reactions 1 and 2
k_1, k_2	Reaction coefficients
E_1, E_2	Activation energy
F_0	Reactor's input flow-rate
H_{L,F_0}	Enthalpy of reactor's input mixture
R_g	Gas constant
e	Euler's number

The dynamic model of the distillation column can be divided into three parts: condenser, trays and re-boiler. The components of the distillation column have been numerated from the top to the bottom, starting from the condenser labeled as component 0. For the condenser the following dynamic model is obtained

$$\dot{M}_0 = V - R - D \quad (2.5)$$

$$\dot{x}_{i,0} = \frac{V}{M_0} (y_{i,1} - x_{i,0}) \quad (2.6)$$

$$\dot{H}_{L,0} = \frac{V}{M_0} (H_{V,1} - H_{L,0}) + \frac{Q_{cond}}{M_0} \quad (2.7)$$

Same considerations done for equation (2.1) can be applied to equation (2.5). Equation (2.6) shows that the variation of the liquid-phase composition for each species $i = A, B, C$ is linearly dependent on the vapor-phase composition of the same species $y_{i,1}$ coming from the topmost tray of the tower and the liquid-phase composition of same species in the condenser. Also for (2.7) the variation of the enthalpy of the liquid-phase depends linearly on the enthalpy of the gas mixture coming from tray 1 and the enthalpy of the mixture inside the condenser, but also depends on the external coolant input Q_{cond} .

The dynamic model of the trays is different depending on the position inside the column with respect to the external feed F , three cases arise. For the trays situated over the feed tray the model is

$$\dot{x}_{i,j} = \frac{1}{M_j} [V(y_{i,j+1} - y_{i,j}) + R(x_{i,j-1} - x_{i,j})], \quad 1 \leq j < f \quad (2.8)$$

$$\dot{H}_{L,j} = \frac{1}{M_j} [V(H_{V,j+1} - H_{V,j}) + R(H_{L,j-1} - H_{L,j})], \quad 1 \leq j < f \quad (2.9)$$

Equation (2.8) represents the variation of the liquid-phase composition for species i inside the j -th tray. It is composed of two elements: the first one is the difference between the vapor-phase composition $y_{i,j+1}$ coming from the tray below and the one contained in the current tray j , weighted by the vapor flow-rate V . Similarly, the second element is the difference between the liquid-phase composition in the current tray with the one of the tray above, weighted by the flow-rate of the liquid reflux R . In case of the top tray, $j = 1$, the difference between the liquid-phase composition in the current tray is performed with the one of the condenser. The same structure can be found in equation (2.9), where the vapor $H_{V,j}$ and liquid $H_{L,j}$ enthalpies differences are weighted by the corresponding vapor and liquid reflux flow-rates. Both equations are weighted by the inverse of the liquid holdup contained in the tray, in this work it is considered constant and equal to $M_j = 170$.

The dynamic model of the feed tray is

$$\dot{x}_{i,f} = \frac{1}{M_f} [V(y_{i,f+1} - y_{i,f}) + R(x_{i,f-1} - x_{i,f}) + F(x_{i,R} - x_{i,f})], \quad j = f \quad (2.10)$$

$$\dot{H}_{L,f} = \frac{1}{M_f} [V(H_{V,f+1} - H_{V,f}) + R(H_{L,f-1} - H_{L,f}) + F(H_{L,R} - H_{L,f})], \quad j = f \quad (2.11)$$

The structure of equations (2.10) and (2.11) is the same of (2.8) and (2.9) respectively. The main difference is that a new element, corresponding to the variation caused by the mixture liquid-phase concentrations and enthalpy coming from the CSTR, appears.

The model for trays below the feed tray is

$$\dot{x}_{i,j} = \frac{1}{M_j} [V(y_{i,j+1} - y_{i,j}) + (R + F)(x_{i,j-1} - x_{i,j})], \quad f < j \leq N \quad (2.12)$$

$$\dot{H}_{L,j} = \frac{1}{M_j} [V(H_{V,j+1} - H_{V,j}) + (R + F)(H_{L,j-1} - H_{L,j})], \quad f < j \leq N \quad (2.13)$$

The model equations maintain the same structure as the trays above the feed tray. The only difference is that the liquid flow-rates is now the sum of the reflux R and the feed F . The term N corresponds to the number of trays and it is equal to 15.

The dynamic model of the re-boiler is

$$\dot{M}_{N+1} = R + F - V - B_t \quad (2.14)$$

$$\dot{x}_{i,N+1} = \frac{1}{M_{N+1}} [(R + F)(x_{i,N} - x_{i,N+1}) - V(y_{i,N+1} - x_{i,N+1})] \quad (2.15)$$

$$\dot{H}_{L,N+1} = \frac{R + F}{M_{N+1}} (H_{L,N} - H_{L,N+1}) - \frac{V}{M_{N+1}} (H_{V,N+1} - H_{L,N+1}) + \frac{Q_{reb}}{M_{N+1}} \quad (2.16)$$

Equation (2.14) has the same structure of (2.1) and (2.5), the variation of the liquid holdups is a linear combination of the feeds entering and exiting the re-boiler. The derivative of the liquid-phase compositions $\dot{x}_{i,N+1}$ is linearly related to two elements, as shown in (2.15). The first element is the difference between the liquid-phase composition of tray N from the liquid-phase composition inside the re-boiler, weighted by the flow-rate of the liquid stream coming from the tray above. The second is the difference of the vapor-phase composition of the tray above with respect to the liquid-phase composition contained in the re-boiler. Equation (2.16) maintains a structure similar to the one corresponding to $\dot{H}_{L,0}$, see equation (2.7), with the addition of an element, that is the difference between the enthalpy of the liquid mixture contained in the bottom tray and the one in the re-boiler.

According to the assumption of ideal behavior in both liquid and vapor-phases it is possible to obtain the expression for the enthalpy of a vapor mixture

$$H_V = \sum_i^{A,B,C} y_i h_{V,i}^0 + (T - T_0) \sum_i^{A,B,C} y_i C_{pV,i} \quad (2.17)$$

where T_0 is the reference temperature and its value is 373.15 [K]. In this work, the enthalpy of a liquid mixture H_L is known, so it is possible to obtain the value of temperature using the following expression

$$T = \frac{H_L - \sum_i^{A,B,C} x_i (h_{V,i}^0 - \Delta H_i^{Vap})}{\sum_i^{A,B,C} x_i C_{pV,i}} \quad (2.18)$$

For ideal liquid-vapor mixture, the relationship between the vapor-phase molar composition and the liquid-phase molar composition of each species is determined using Raoult's law

$$y_i = \frac{\alpha_i x_i}{\sum_k^{A,B,C} \alpha_k x_k} \quad (2.19)$$

All process parameters appearing in the last equations are assumed to be constant. Their meaning is explained in Table 2.4, while Table 2.5 contains their nominal values.

Table 2.4: Process parameters

$C_{pV,i}$	Heat capacity of each species at vapor-phase
$h_{V,i}^0$	Enthalpy of each species in vapor state at reference temperature
ΔH_i^{Vap}	Enthalpy of vaporization of each species
α_i	Relative volatilities of each species

Table 2.5: Process parameters values

	A	B	C
$C_{pV,i}$ [J/mol K]	1.86	2.01	2.00
ΔH_i^{Vap} [J/mol]	83.333	86.111	85.556
$h_{V,i}^0$ [J/mol]	283.889	369.844	394.444
α_i	5.5	1.2	1.0

Table 2.6: Model parameters

k_1	2.4 [1/s]	k_2	4.0 [1/s]
E_1	9500 [J/mol]	E_2	12000 [J/mol]
ΔH_{r1}	2500 [J/mol]	ΔH_{r2}	5500 [J/mol]
F_0	100 [mol/s]	H_{L,F_0}	61.06 [J/mol]
B_t	100 [mol/s]	R	290 [mol/s]
R_g	8.314 [J/K/mol]	T_0	373.15 [K]
M_j	170 [mol]		

2.3 Open loop response

In this paragraph, the open loop relationships between input and outputs are analyzed. System has been initialized at the equilibrium state corresponding to the inputs contained in Table 2.7. The steady-state values for the states of CSTR, condenser and re-boiler are shown in Table 2.8.

Table 2.7: Equilibrium state inputs

\bar{F}	1880 [mol/s]	\bar{D}	1780 [mol/s]
\bar{V}	2070 [mol/s]	\bar{Q}_{cstr}	3.58×10^5 [J/s]
\bar{Q}_{cond}	-2.00×10^5 [J/s]	\bar{Q}_{reb}	2.335×10^5 [J/s]

Table 2.8: Steady state values

M_R	1300 [mol]	$H_{L,R}$	195.02 [J/mol]
M_0	1125 [mol]	$H_{L,0}$	204.05 [J/mol]
$M_N + 1$	1425 [mol]	$H_{L,N+1}$	369.41 [J/mol]
$x_{A,R}$	0.795	$x_{B,R}$	0.212
$x_{A,0}$	0.802	$x_{B,0}$	0.179
$x_{A,N+1}$	0.0016	$x_{B,N+1}$	0.796

Values for the species C compositions $x_{C,R}$, $x_{C,0}$ and $x_{C,N+1}$ are obtained using the relationship $x_{A,i} + x_{B,i} + x_{C,i} = 1$. The steady-state values of the distillation column states have been derived through simulation starting from a generic initial condition, the obtained values are shown graphically in Figure 2.4. A trend can be observed in the trays steady-state values. The values of a state between the topmost tray, tray number 1, and the next ones until the feed tray, tray 12, are slowly increasing. Only exception is the concentration of species A that slowly decreases as the tray number increases. After the feed tray the rate at which the state changes value increases drastically due to the fact that the outlet of the CSTR is added to the mixture contained in the trays.

The effect of a step change in the values of the heat/coolant inputs is analyzed. These inputs do not affect the values of the liquid holdup states. Three simulations were performed, one for each input. In each simulation, a step of magnitude corresponding to 5% of the steady-state value of the heat/coolant input is performed at simulation time 10s. The temperature variations due to a step in the heat input of the CSTR are shown in figure 2.5. It is important to note that all three output temperatures are affected by the input. The response of each temperature to Q_{cstr} is the one of a nonlinear system, as it can be argued from Figure 2.6 containing the response of T_{cstr} only. The variation of each output is around 4 ~ 4.5 Kelvin and takes about 40 or more seconds. Similar results are obtained if a step variation of one of the two other

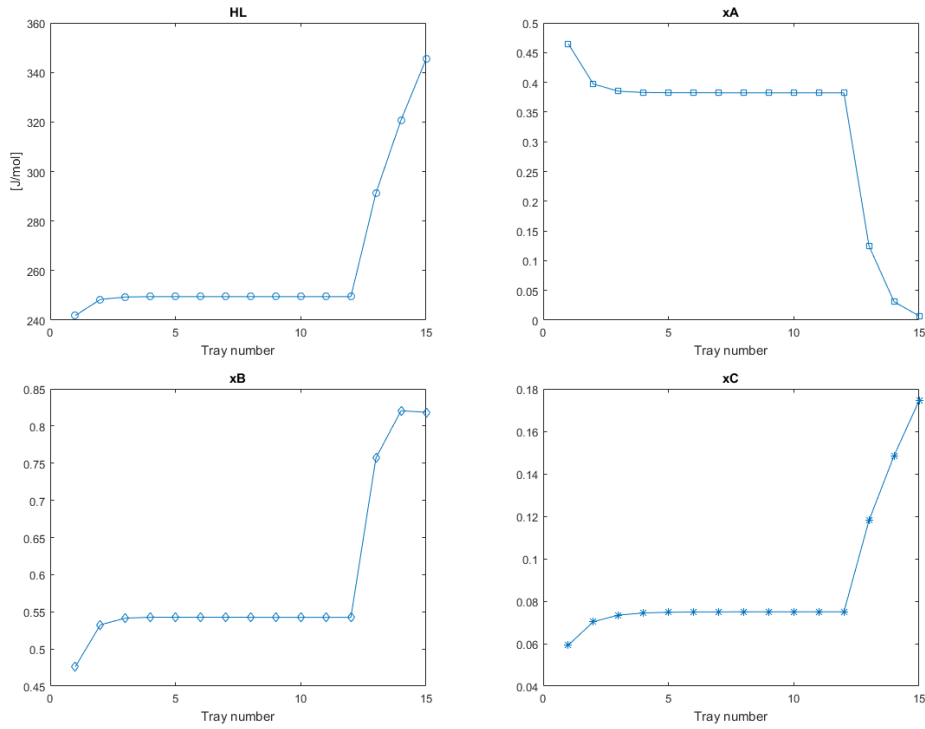


Figure 2.4: Trays states: steady-state values

inputs is performed but with different amplitudes in the responses, as shown in Figures 2.7 and 2.8. Numerical values of the temperature variations in the three cases are reported in Table 2.9. Changing the direction of the input variation generates the same results from a practical standpoint. The final values of the temperatures are slightly different from the one obtained in the previous simulations. The biggest difference happens for Q_{cstr} , if two step of amplitude $\pm 5\% \bar{Q}_{cstr}$ are applied, the actual final variation of the temperatures differs of about 0.04[K]. An example of this is shown in Figure 2.9, where the responses of T_{cstr} to two input steps of amplitude $\pm 5\% \bar{Q}_{cstr}$ are plotted together.

Table 2.9: Temperature variations due to a 5% heat/coolant variation

	[J/s]	ΔT_{cstr} [K]	ΔT_{cond} [K]	ΔT_{reb} [K]
ΔQ_{cstr}	1.79×10^5	4.5072	4.5041	4.3412
ΔQ_{cond}	-10^5	-2.5289	-5.5063	-2.4268
ΔQ_{reb}	1.17×10^5	2.2952	4.9993	15.0030

The effect of a variation in the input flow-rates is now considered. In all the simulations shown, a step input has been given to the system, one flow-rate at a time, of amplitude equal to 1% of the initial steady-state value. Recalling equations (2.1), (2.5) and (2.14), each of the input flow-rates appears in two state equations, one with positive and one negative sign. This causes that when a positive amplitude step is applied to the system, one of the liquid holdups decreases. During each simulation, as soon as one of the liquid holdups experiences a negative variation corresponding 10% of his initial value, each input has been reset to its initial steady-state value. This has to be done in order to prevent reaching negative values of the holdups, that will give non meaningful results from a physical stand point. Figure 2.10 shows the response of the outputs

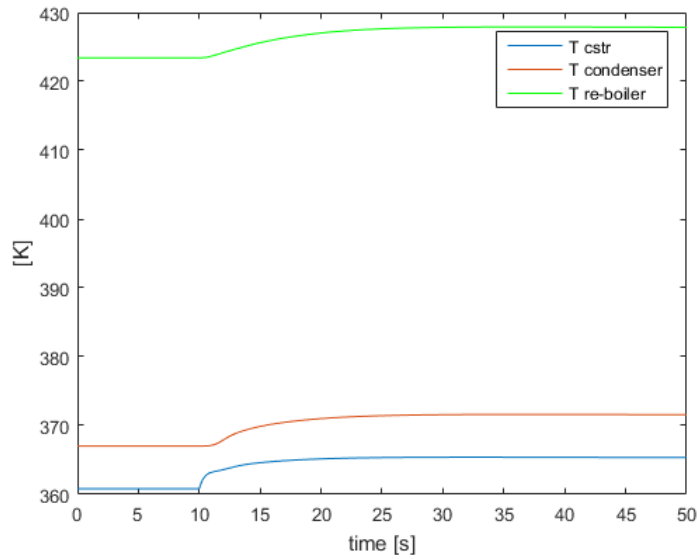


Figure 2.5: Temperature response to a 5% increase of Q_{cstr}

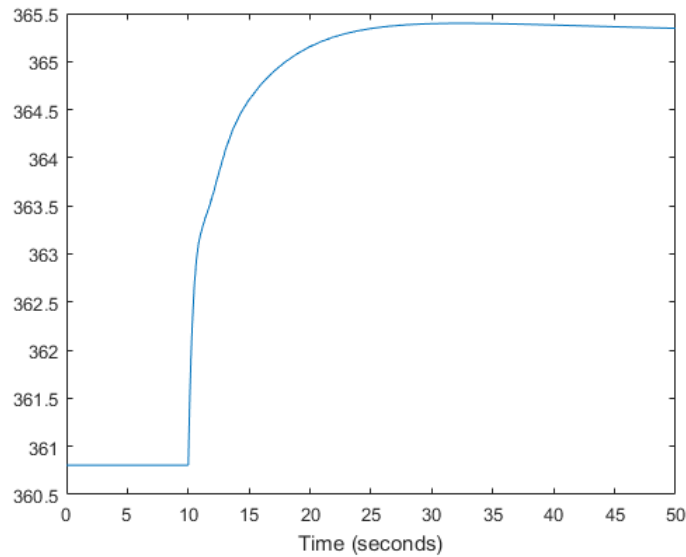


Figure 2.6: T_{cstr} response to a 5% increase of Q_{cstr}

to a variation of input F . The liquid holdups affected by this input are M_{cstr} and M_{reb} . The relationship between this output and the input is the one of a pure integrator, as stated in the model equation paragraph. All temperatures are increased by a positive variation of input F and their evolution is typical of a nonlinear system. Similar considerations can be done when a positive step variation happens for the other two flow-rate inputs D and V . Their effects on the system are shown in Figures 2.11 and 2.12 respectively. The high non-linearity of the temperature/flow-rate relationship can be experienced fully in the response of T_{cond} and T_{reb} to V , as shown in Figure 2.13. Analyzing the different responses it appears that the system presents a two-time-scale behavior. It is common in this type of systems and is caused by the

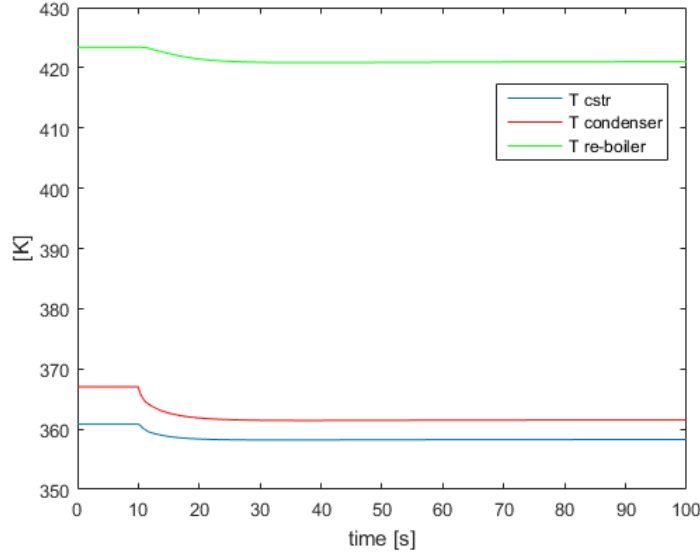


Figure 2.7: Temperature response to a 5% increase of Q_{cond}

large value of the recycle stream D . The liquid holdups outputs have faster responses with respect to the temperatures and the other states. Controlling the entire system using one single MPC controller is not recommended, due to the fact that to accomplish good performance of the level control a fast sample time should be considered, but the high complexity of the overall system does not allow this. For these reasons, in this work, two separate control strategies are used for the regulation of the different outputs. A proportional controller has been designed for the regulation of the liquid holdups, presented in next paragraph, while temperature regulation is achieved through MPC control.

2.4 Level control

To control the levels a simple proportional controller has been designed. At each liquid holdup output a single flow-rate has been associated: flow-rate F is used to control M_R , D for M_0 and V for M_{N+1} . The designed controllers are

$$F = \bar{F}(1 - k_c(M_{R_{ref}} - M_R)) \quad (2.20)$$

$$D = \bar{D}(1 - k_c(M_{0_{ref}} - M_0)) \quad (2.21)$$

$$V = \bar{V}(1 - k_c(M_{N+1_{ref}} - M_{N+1})) \quad (2.22)$$

where the coefficient $k_c = 10^{-4}$ has been used. The idea is to have the control action composed of two parts. A constant part corresponding to the initial steady-state flow-rate value and a part proportional to the output error. The goal of this controller is to have fast response and to keep the liquid holdup level as constant as possible.

(INSERIRE GRAFICI)

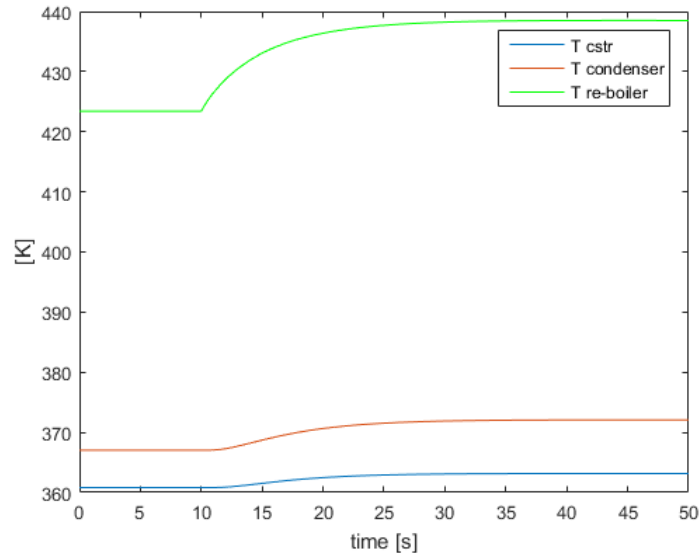


Figure 2.8: Temperature response to a 5% increase of Q_{reb}

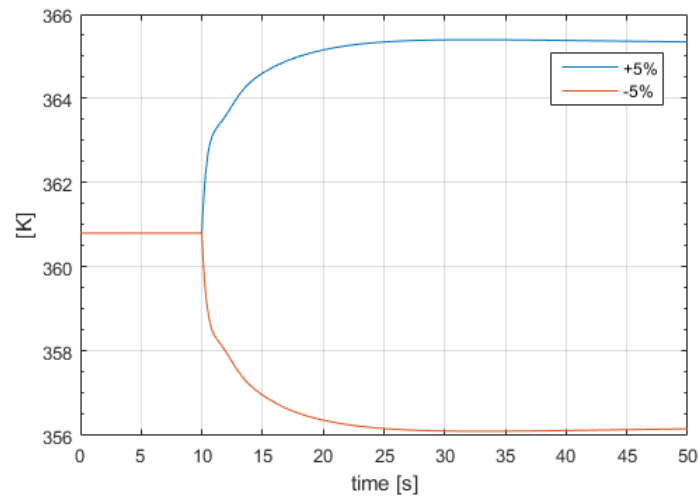


Figure 2.9: T_{cstr} response to $\pm 5\% Q_{cstr}$

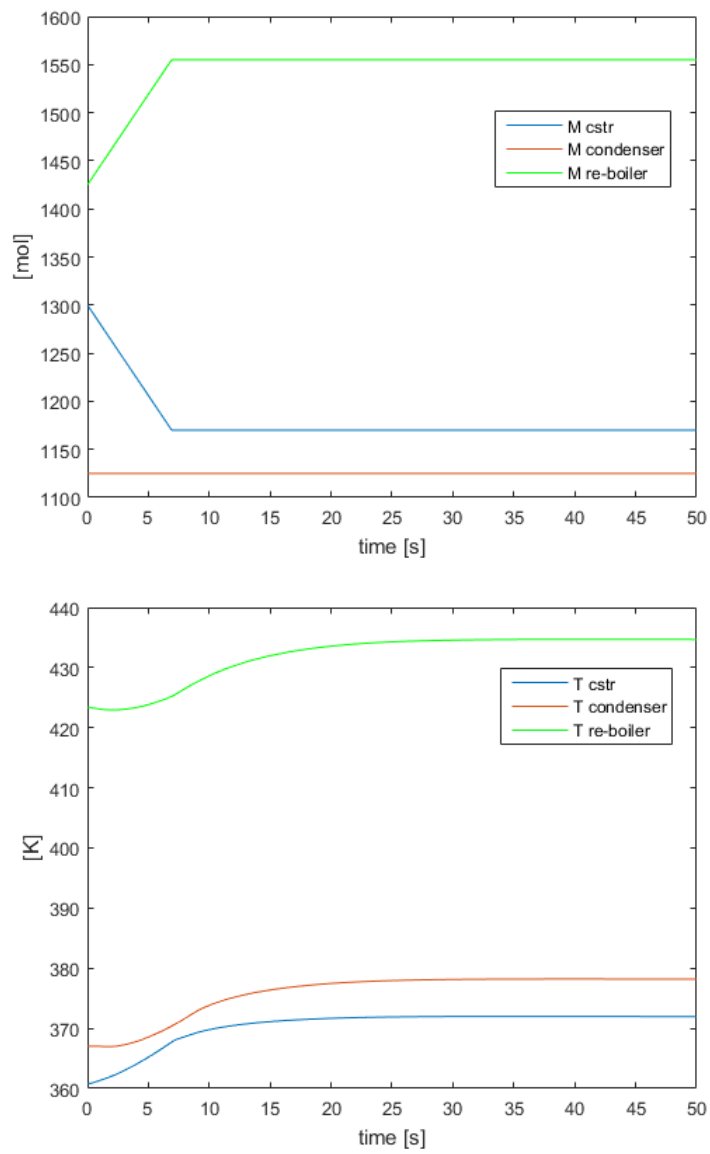


Figure 2.10: Response to a 10% positive step variation of F

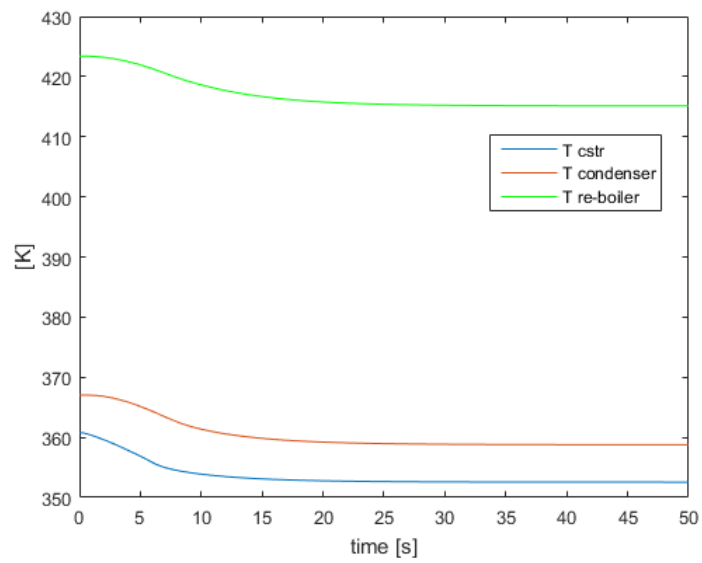
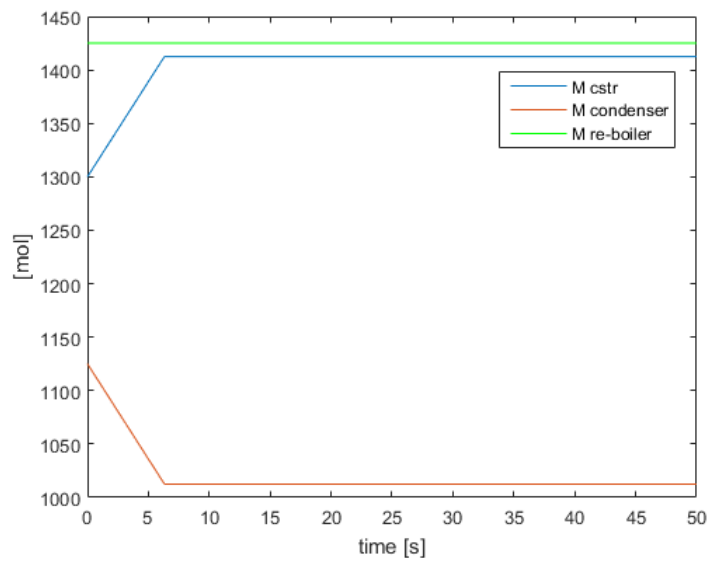


Figure 2.11: Response to a 10% positive step variation of D

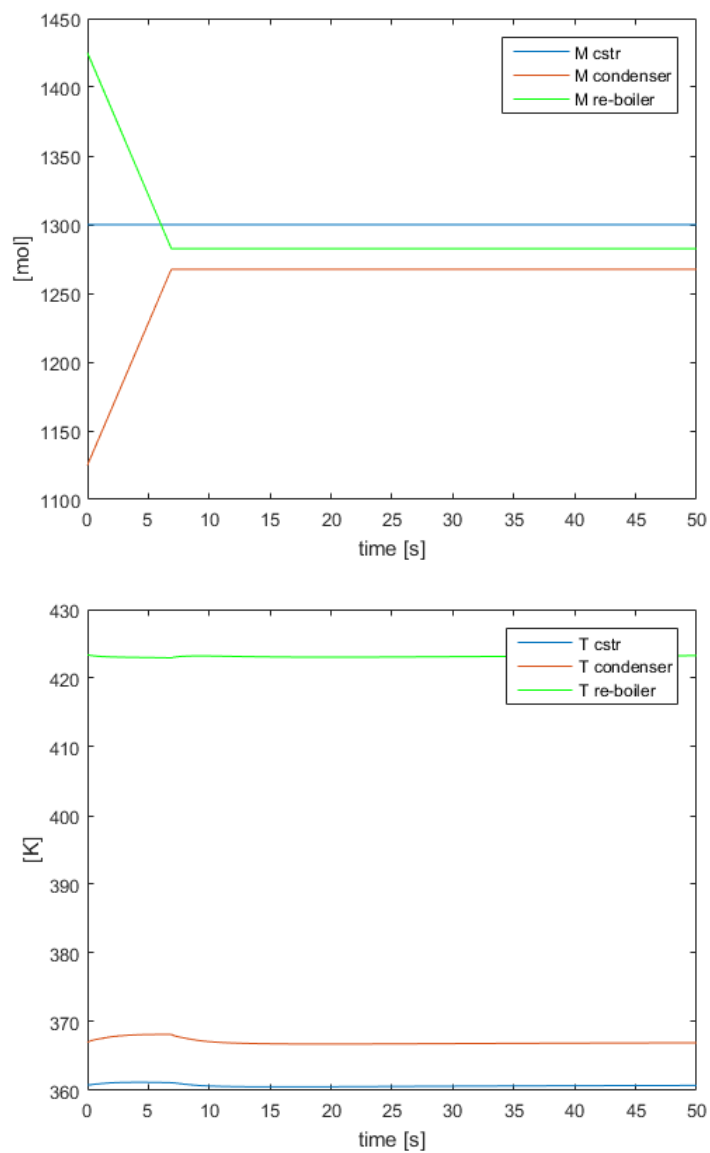


Figure 2.12: Response to a 10% positive step variation of V

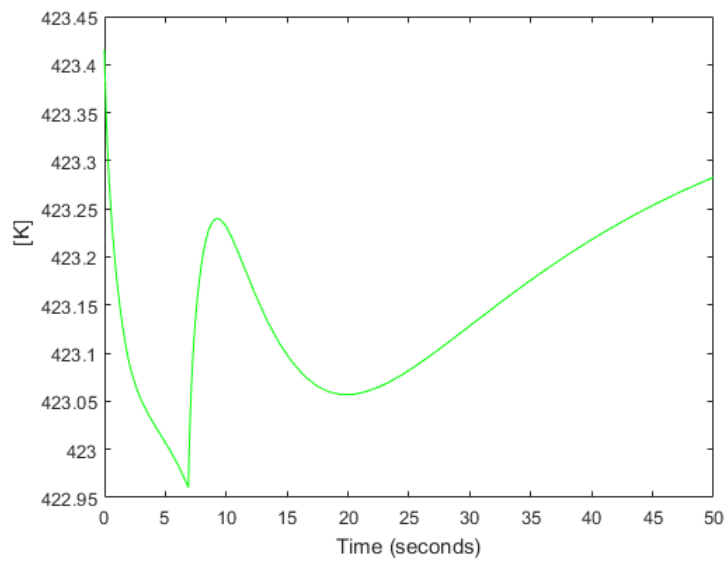
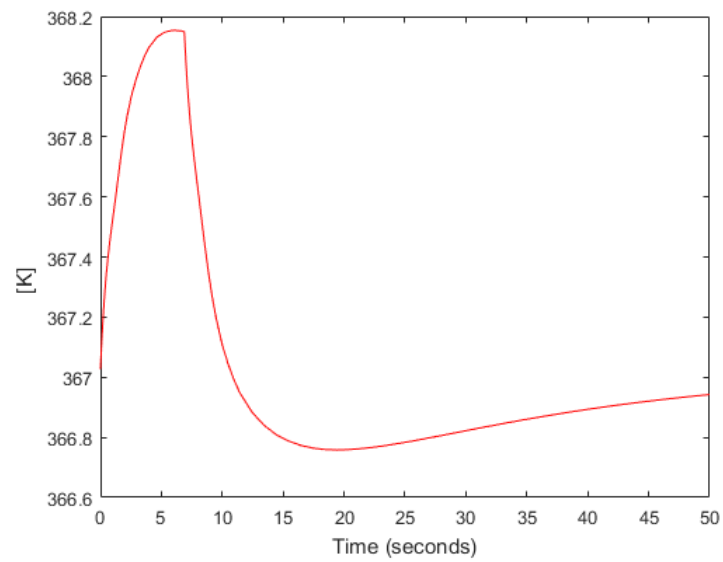


Figure 2.13: Non-linear behavior of T_{cond} and T_{reb}

Chapter 3

Linear MPC

In this chapter, regulation of the temperatures is obtained with a traditional Model Predictive Control scheme based on a linearized model of the system. The linear state-space model of the system is obtained directly from the Simulink model using the Control Design toolbox. Linearization has been performed around the equilibrium state described in chapter 2.3 considering as inputs the three heat/coolant inputs and as outputs the temperatures. The obtained model is in the form

$$\begin{cases} \Delta\dot{x}(t) = A_c\Delta x(t) + B_c\Delta u(t) \\ \Delta y(t) = C_c\Delta x(t) \end{cases} \quad (3.1)$$

The model is asymptotically stable and the number of states is $n = 71$. The states $\Delta x(t) = x(t) - \bar{x}$ of the considered model are the variations with respect to the nominal values of the enthalpies and the species composition in each element of the tower and in the CSTR. Given that MPC control strategies require a discrete-time model of the system, a sample time $T_s = 30[s]$ is chosen and discretization of the model is performed using the c2d Matlab function. Discrete-time model of the system is

$$\begin{cases} \Delta x(k+1) = A_d\Delta x(k) + B_d\Delta u(k) \\ \Delta y(k) = C_d\Delta x(k) \end{cases} \quad (3.2)$$

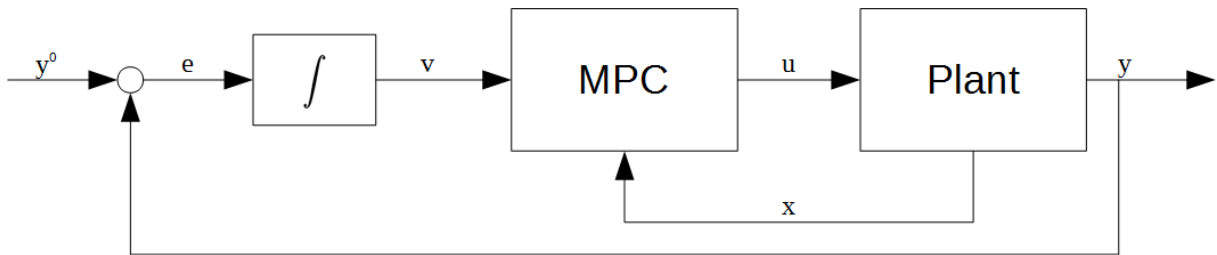


Figure 3.1: MPC control scheme with integral action

To ensure reference signal tracking and robustness with respect of external disturbances an integral action has been introduced in the control scheme, as shown in Figure 3.1. The enlarged system equations are

$$\begin{aligned} \Delta x(k+1) &= A_d\Delta x(k) + B_d\Delta u(k) \\ v(k+1) &= v(k) + e(k+1) \end{aligned} \quad (3.3)$$

where $e(k) = y^o - y(k)$, where y^o is the output reference. Substituting the expression of $e(k)$, the second equation of (3.3) can be re-written as

$$v(k+1) = v(k) + y^o - C_d \Delta x(k+1) = v(k) + y^o - C_d A_d \Delta x(k) - C_d B_d \Delta u(k) \quad (3.4)$$

The state and control variable increments are defined as $\delta x(k) = \Delta x(k) - \Delta x(k-1)$ and $\delta u(k) = \Delta u(k) - \Delta u(k-1)$. The model is re-written subtracting from the first expression of (3.3) and from (3.4) their values at the previous time instant. The obtained expression for the enlarged system is

$$\begin{bmatrix} \delta x(k+1) \\ e(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & 0 \\ -C_d A_d & I_{p \times p} \end{bmatrix}}_A \begin{bmatrix} \delta x(k) \\ e(k) \end{bmatrix} + \underbrace{\begin{bmatrix} B_d \\ C_d B_d \end{bmatrix}}_B \delta u(k) \quad (3.5)$$

$z(k) = \begin{bmatrix} \delta x(k) \\ e(k) \end{bmatrix}$ is the enlarged state of the system with the integrators. The overall model of the enlarged system is

$$\begin{cases} z(k+1) = Az(k) + B\delta u(k) \\ y(k) = Cz(k) \end{cases} \quad (3.6)$$

Where $C = [0_{n \times n} \quad I_{n \times p}]$.

3.1 Optimization problem

The optimization problem consists of computing the sequence $[\delta u(k), \dots, \delta u(k+N_p)]$ that minimizes, at each time step, the finite horizon quadratic cost function

$$J(z(k), \delta u(\cdot), k) = \sum_{i=0}^{N_p-1} \left(\|z(k+i)\|_Q^2 + \|\delta u(k+i)\|_R^2 \right) + \|z(k+N_p)\|_S^2 \quad (3.7)$$

where $Q = Q' \geq 0$, $R = R' > 0$, $S = S' \geq 0$ are matrices of suitable dimensions. The constant N_p is the prediction horizon and defines the number of discrete-time steps considered for the solution of the optimization problem. Defining the vectors containing the state and control input

$$Z(k) = \begin{bmatrix} z(k+1) \\ z(k+2) \\ \vdots \\ z(k+N_p-1) \\ z(k+N_p) \end{bmatrix}, \quad \Delta U(k) = \begin{bmatrix} \delta u(k) \\ \delta u(k+1) \\ \vdots \\ \delta u(k+N_p-2) \\ \delta u(k+N_p-1) \end{bmatrix}$$

and matrices A_c and B_c as

$$A_c = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p+1} \\ A^{N_p} \end{bmatrix}, \quad B_c = \begin{bmatrix} B & 0 & 0 & \dots & 0 & 0 \\ AB & B & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ A^{N_p-2}B & A^{N_p-3}B & A^{N_p-4}B & \dots & B & 0 \\ A^{N_p-1}B & A^{N_p-2}B & A^{N_p-3}B & \dots & AB & B \end{bmatrix} \quad (3.8)$$

it follows that

$$Z(k) = A_c z(k) + B_c \Delta U(k) \quad (3.9)$$

The finite horizon quadratic cost function, equation (3.7), can be re-written as

$$J(z(k), \delta u(\cdot), k) = (A_c z(k) + B_c \Delta U(k))' Q_T (A_c z(k) + B_c \Delta U(k)) + \Delta U'(k) R_T \Delta U(k) \quad (3.10)$$

Where matrices Q_T and R_T are

$$Q_T = \begin{bmatrix} Q & 0 & \cdots & 0 & 0 \\ 0 & Q & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Q & 0 \\ 0 & 0 & \cdots & 0 & S \end{bmatrix}, \quad R_T = \begin{bmatrix} R & 0 & \cdots & 0 & 0 \\ 0 & R & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & R & 0 \\ 0 & 0 & \cdots & 0 & R \end{bmatrix} \quad (3.11)$$

Expanding (3.10) and grouping the common factors, the finite horizon cost function becomes

$$J(z(k), \delta u(\cdot), k) = z'(k) A_c z(k) + 2z'(k) A_c' Q_T B_c \Delta U(k) + \Delta U'(k) (B_c' Q_T B_c + R_T) \Delta U(k) \quad (3.12)$$

First term of equation (3.12) does not depend on $\Delta U(k)$ and thus can be ignored. The obtained optimization problem is in linear quadratic form

$$\bar{J}(z(k), \delta u(\cdot), k) = 2z'(k) A_c' Q_T B_c \Delta U(k) + \Delta U'(k) (B_c' Q_T B_c + R_T) \Delta U(k) \quad (3.13)$$

The quadratic term coefficient ($B_c' Q_T B_c + R_T$), can be computed off-line because it only depends on constant matrices. The linear term depends on the state and changes at each iteration of the algorithm. The solution of the optimization problem is obtained using the Matlab[®] function *quadprog*. The constraints considered for the optimization problem are in the form

$$A_{vinc} \Delta U = B_{vinc} \quad (3.14)$$

where $A_{vinc} = \begin{bmatrix} I_{n_e \times n_e} \\ -I_{n_e \times n_e} \end{bmatrix}$ and $B_{vinc} = \begin{bmatrix} U_{max} \\ -U_{min} \end{bmatrix}$. Parameter $n_e = n + p$ is the enlarged state size, while U_{max} and U_{min} are vectors of the same size ($n_e, 1$) containing respectively the maximum and minimum possible variations of the heat/coolant inputs. In all of the performed simulations the maximum variation of the heat/coolant inputs is $\delta u_{max} = \pm 400 [J/s]$. According to the Receding Horizon approach, at each time step k , only the first element of the optimal control sequence $\Delta U(k)$ is used. The value of the inputs sent to the system are obtained as $u(k) = u(k-1) + \delta u(k)^o$.

3.2 Tuning of the MPC

Tunable elements of the MPC are the matrices Q , R and S ; Q and S have to be symmetric and semi-positive definite, while R has to be symmetric and positive definite. The Q matrix is diagonal, with its values set according to the normalization approach, each non zero element of the matrix is $Q(i, i) = \frac{1}{|\bar{x}(i)|}$, $i = 1, 2, \dots, n$, where $|\bar{x}(i)|$ are the value of the states at the initial condition described in chapter 2.3. Matrix S is set equal to matrix Q . The choice of matrix R is similar to the one for matrix Q , but its elements on the diagonal are equal to $R(1, 1) = \frac{1}{|Q_{cstr}|}$, $R(2, 2) = \frac{1}{|Q_{cond}|}$ and $R(3, 3) = \frac{1}{|Q_{reb}|}$. The normalization approach has been chosen due to the fact that order of magnitude of the states and the inputs is very different at the desired steady-state conditions, as shown in the previous chapter in Tables 2.7 and 2.8. To reduce computational costs and have a less aggressive control strategy, a control horizon $N_u = 3$ has been chosen. This results in the additional constraints $\delta u(k+i) = 0$ with $i = N_u, \dots, N_p - 1$, for the optimization problem.

3.3 Kalman Observer

In the linear MPC application the states of the system have been considered as unmeasurable. For this reason Kalman predictor has been designed to obtain an estimate of the state increments $\delta\hat{x}(k|k-1)$. The optimal predictor is

$$\delta\hat{x}(k+1|k) = (A_d - \bar{L}C_d)\delta\hat{x}(k|k-1) + B_d\delta u(k) + \bar{L}e(k) \quad (3.15)$$

Where the gain is $\bar{L} = A_d\bar{P}C_d'(C_d\bar{P}C_d' + \tilde{R})^{-1}$. Matrix \bar{P} is the unique positive definite solution of the stationary Riccati equation

$$\bar{P} = A_d\bar{P}A_d' + \tilde{Q} - A_d\bar{P}C_d'(C_d\bar{P}C_d' + \tilde{R})^{-1}C_d\bar{P}A_d' \quad (3.16)$$

The value of \bar{L} has been obtained using the Matlab function *lqr*. Matrices \tilde{Q} and \tilde{R} are set as $I_{n \times n}$ and $I_{p \times p}$ respectively. The linear MPC control scheme, complete of the level controller and Kalman filter, is shown in Figure 3.2.

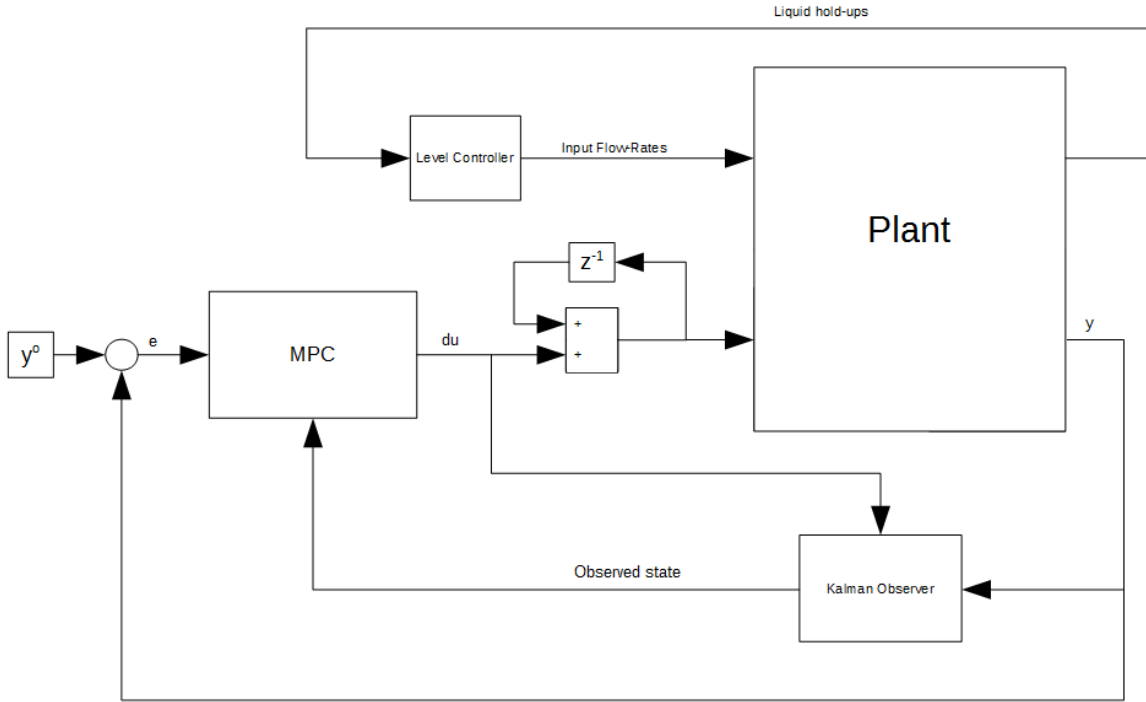


Figure 3.2: Linear MPC control scheme

3.4 Results

In this section the closed loop system responses to variations in the heat/coolant inputs are presented. A set of simulations is performed considering a step variation of one reference temperature at a time. The amplitude of the step is equal to +5% of the initial reference value and happens at time $t = 1000[s]$ in all of the simulations. The effect of a variation in the liquid hold-ups is then analyzed, considering only as an example the liquid hold-up in the condenser M_{cond} . The variation considered is a step of amplitude equal to +5% of the initial value occurring at time $t = 600[s]$.

3.4.1 Re-boiler reference temperature

The first case considered is the step variation of the re-boiler reference temperature T_{reb}^o . The temperature of the re-boiler reaches the new reference value in around 2000 seconds, as shown in Figure 3.3. During the transient some of the inputs saturate at their maximum values, as it is shown in Figure 3.4, leading to an almost linear increment in the value of T_{reb} . The responses of the other two output temperatures, T_{cstr} and T_{cond} , are shown in Figures 3.5 and 3.6 respectively. Due to the high non-linearity and interconnection of the system, a variation of the T_{reb}^o reference temperature affects all the temperatures. The maximum variation of T_{cstr} and T_{cond} is less than 5[K] and the MPC is able to restore the initial values in a time comparable with the response time of T_{reb} .

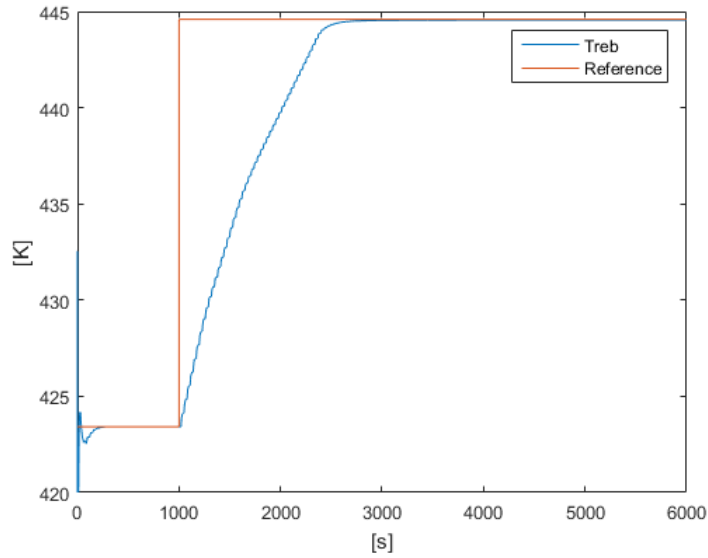


Figure 3.3: T_{reb} response to a step variation of T_{reb}^o

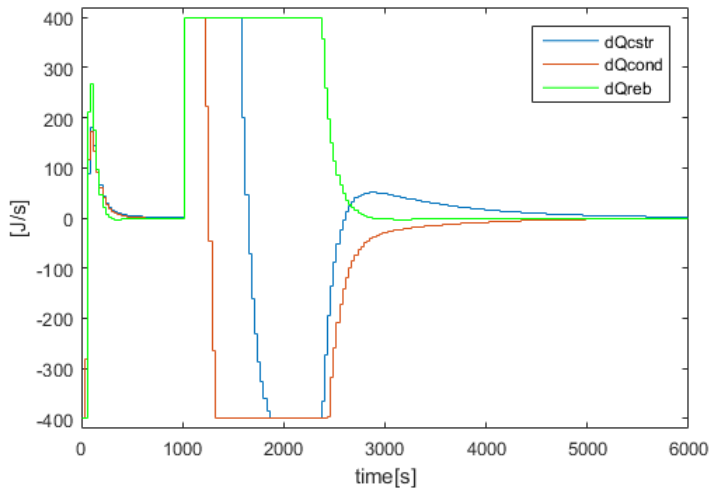


Figure 3.4: Input increments due to a step in T_{reb}^o

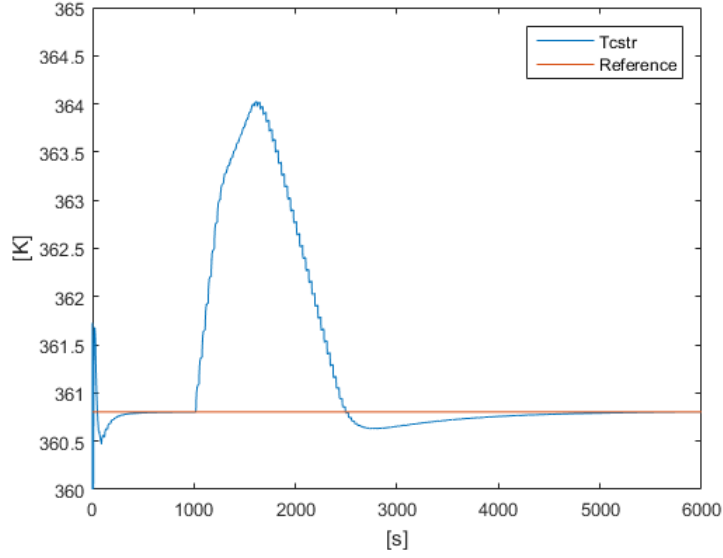


Figure 3.5: T_{cstr} response to a step variation of T_{reb}^o

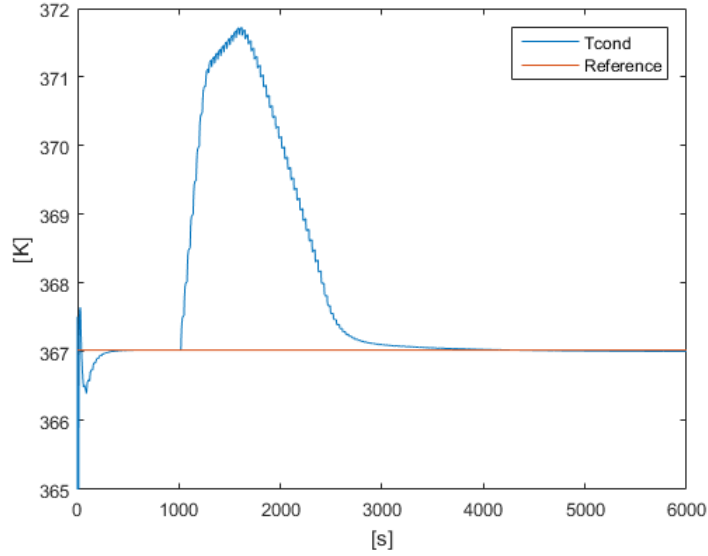


Figure 3.6: T_{cond} response to a step variation of T_{reb}^o

3.4.2 Condenser reference temperature

The responses to the variation of T_{cond}^o are now considered. Figure 3.7 shows the response of the condenser temperature T_{cond} . The response time of T_{cond} is around 7000 seconds, longer with respect to the previous case. Similarly to the first simulation the input increments saturate during the transient. Also in this case, the MPC is able to recover the values of T_{cstr} and T_{reb} after an initial variation, as shown in Figures 3.9 and 3.10. The worst case is the T_{cstr} temperature case, that takes around 10000 seconds to return to the initial value.

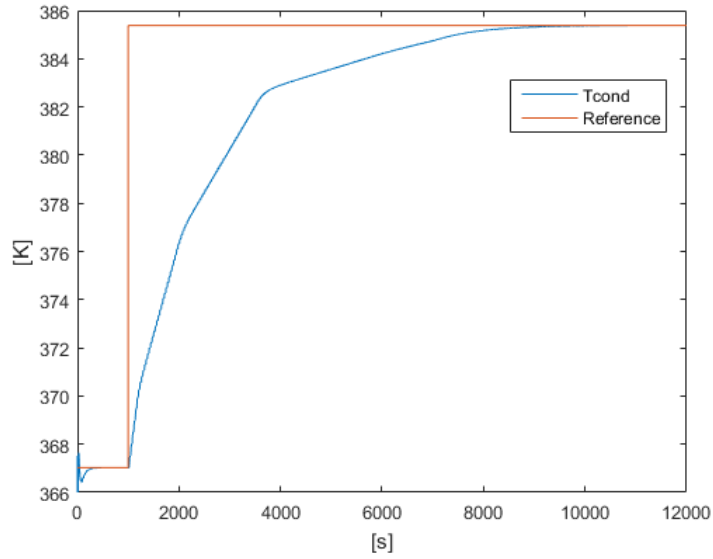


Figure 3.7: T_{cond} response to a step variation of T_{cond}^o

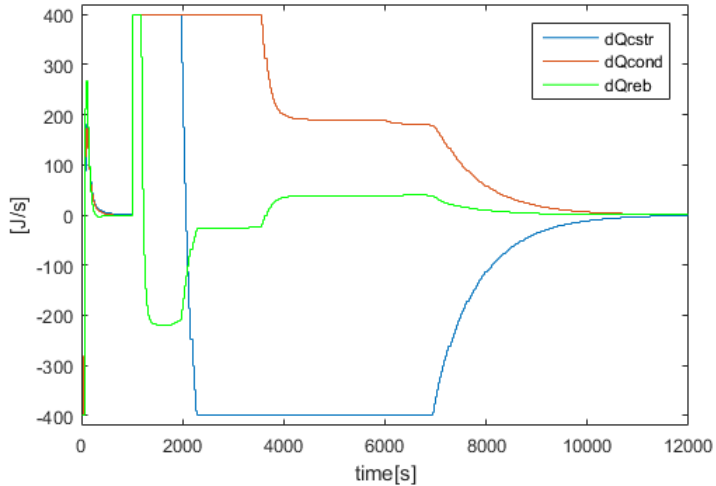


Figure 3.8: Input increments due to a step in T_{cond}^o

3.4.3 Response to the disturbances

In this section the effect of a variation in one of the liquid hold-ups is considered. The effect of a step change in any of the liquid hold-ups is similar, so only the case for the one contained in the condenser is shown here, reason is that it leads to the higher effect on the output temperatures. A step variation of amplitude +10% of the initial steady-state value of M_{cond} is performed at 600 seconds. All the output temperatures are affected by the disturbance, their initial values are recovered in less than 100 seconds, faster with respect to the thermal dynamics. The output responses are shown in Figures 3.11, 3.12 and 3.13. It is interesting to note that the disturbance is easily recovered by the MPC at low control cost, as shown in Figure 3.14.

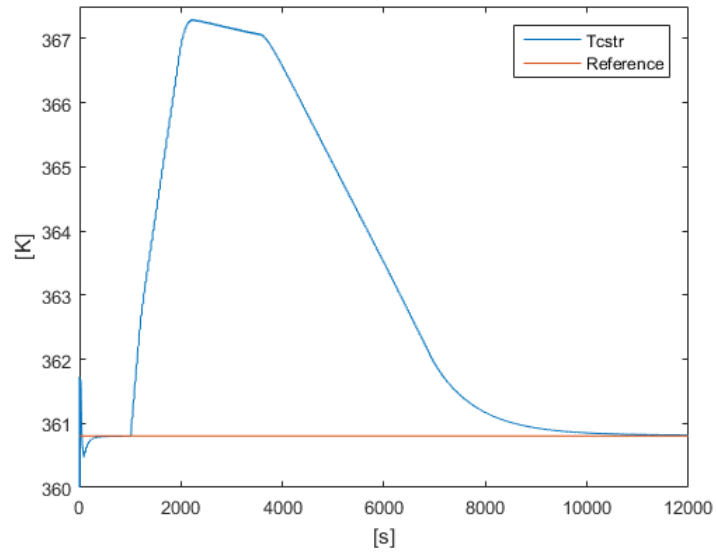


Figure 3.9: T_{cstr} response to a step variation of T_{cond}^o

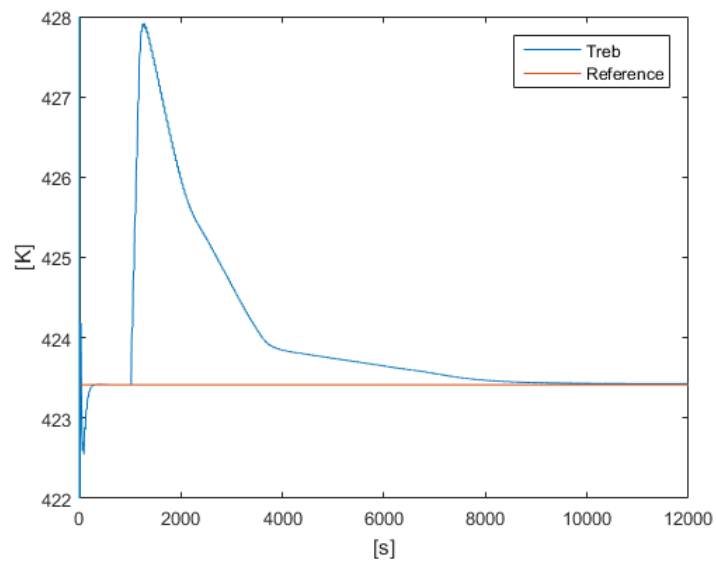


Figure 3.10: T_{reb} response to a step variation of T_{cond}^o

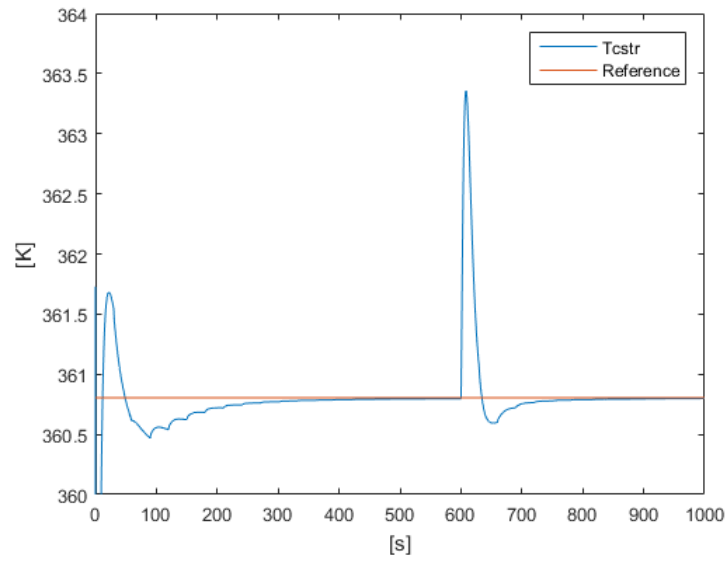


Figure 3.11: T_{cstr} response to a step variation of M_{cond}

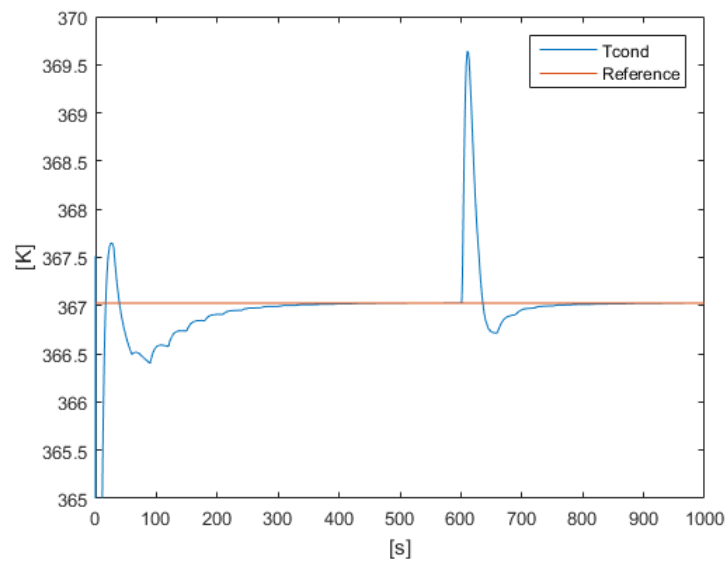


Figure 3.12: T_{cond} response to a step variation of M_{cond}

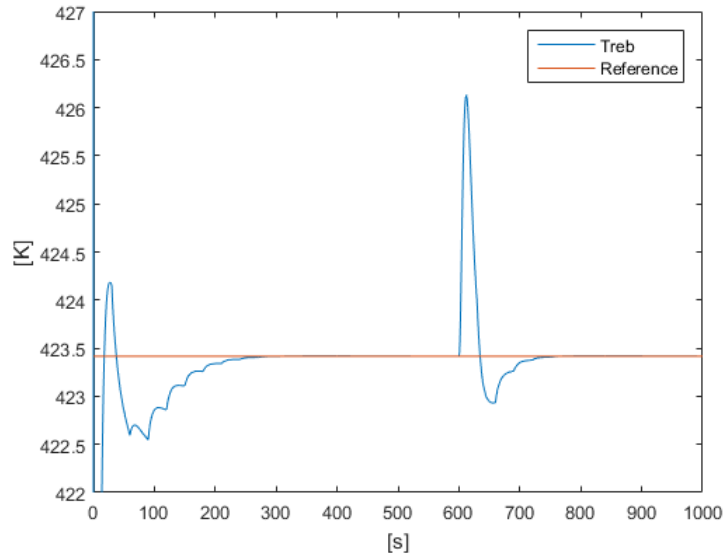


Figure 3.13: T_{reb} response to a step variation of M_{cond}

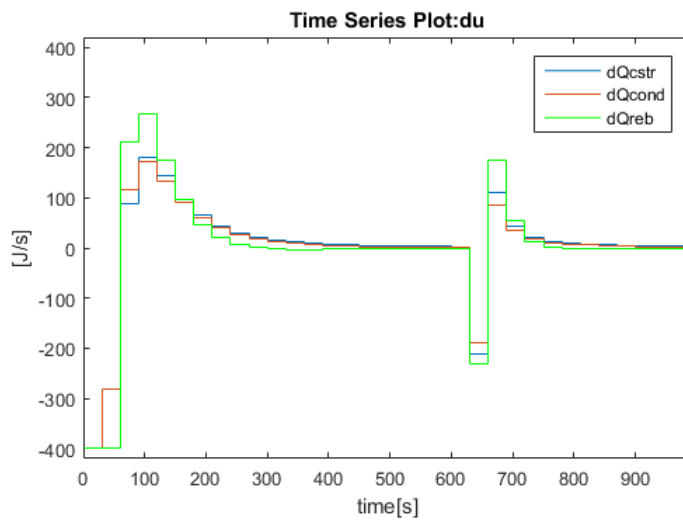


Figure 3.14: Input increments due to a step in M_{cond}

Chapter 4

Trajectory MPC

In this chapter a procedure to implement a Model Predictive Control algorithm based on the linearization of the system around the nominal trajectory is presented. First the control algorithm, in a generic scenario, is explained in detail and the optimization problem to be solved is defined. In the second paragraph the procedure to realize the control scheme for the system considered in this thesis is shown. The last paragraph contains the results obtained with the control scheme compared to the traditional linear MPC results described in the previous chapter.

4.1 Control algorithm

The nonlinear model of a system is

$$\dot{x}(t) = f_c(x(t), u(t), d(t)) \quad (4.1)$$

Fixing a particular sampling time T_s , the system model can be discretized, obtaining

$$x(k+1) = f(x(k), u(k), d(k)) \quad (4.2)$$

where $x(k)$ are the state variables, $u(k)$ the inputs and $d(k)$ the disturbances on the system. At time instant $k-1$ the optimal control sequence is computed using the criterion presented later in this chapter. The obtained sequence is

$$\vec{u}^o(k-1 : k+N_p-2|k-1) = [u^o(k-1|k-1) \quad u^o(k|k-1) \dots u^o(k+N-2|k-1)]$$

According to the Receding Horizon principle, only the first input is applied to the system

$$u(k-1) = u^o(k-1|k-1)$$

At the next time instant k , the following sequence can be defined

$$\vec{u}^o(k : k+N_p-1|k-1) = [u^o(k|k-1) \dots u^o(k+N-2|k-1) \quad u(k+N-1|k-1) = u^o(k+N-2|k-1)]$$

Considering $\hat{x}(k|k) = x(k)$, the values of the state variables in the next prediction steps can be obtained simulating the system (4.2) with the expression

$$\hat{x}(k+i+1|k) = f(\hat{x}(k+i|k), u^o(k+i|k-1), d(k+i)), \quad i = 0, 1, \dots, N_p-1 \quad (4.3)$$

where N_p is the prediction horizon. To perform this computations the values of the disturbances $d(k)$ are assumed to be constant over the prediction horizon

$$d(k+i) = d(k), \quad i = 0, 1, \dots, N_p-1$$

The sequence of values obtained $\hat{x}(k+i|k)$ is called the nominal trajectory referred to the optimal control sequence. Defining the quantities

$$\begin{aligned}\tilde{\delta}\hat{x}(k+i|k) &= x(k+i) - \hat{x}(k+i|k) \\ \tilde{\delta}u(k+i|k-1) &= u(k+i) - u^o(k+i|k-1) \\ \tilde{\delta}d(k+i) &= d(k+i) - d(k+i) = 0 \\ i &= 0, 1, \dots, N_p - 1\end{aligned}$$

the linearization of the system around the computed nominal trajectory is

$$\begin{aligned}x(k+i+1) &= \hat{x}(k+i+1|k) + \tilde{\delta}\hat{x}(k+i+1|k) = \\ &f(\hat{x}(k+i|k), u^o(k+i|k-1), d(k+i)) + A_{k+i}\tilde{\delta}\hat{x}(k+i|k) + B_{k+i}\tilde{\delta}u(k+i|k-1) + M_{k+i}\tilde{\delta}d(k+i)\end{aligned}\quad (4.4)$$

The matrices A_{k+i} , B_{k+i} and M_{k+i} of the linearized model around the nominal trajectory are obtained as

$$\begin{aligned}A_{k+i} &= \left. \frac{\partial f}{\partial x} \right|_{\hat{x}(k+i|k), u^o(k+i|k-1)} \\ B_{k+i} &= \left. \frac{\partial f}{\partial u} \right|_{\hat{x}(k+i|k), u^o(k+i|k-1)} \\ M_{k+i} &= \left. \frac{\partial f}{\partial d} \right|_{\hat{x}(k+i|k), u^o(k+i|k-1)}\end{aligned}\quad (4.5)$$

Defining as the linearization error

$$H(k+i) = f(\hat{x}(k+i|k), u^o(k+i|k-1), d(k+i)) - A_{k+i}\hat{x}(k+i|k) - B_{k+i}u(k+i|k-1) - M_{k+i}d(k+i)\quad (4.6)$$

equation (4.4) can be written as

$$x(k+i+1) = A_{k+i}x(k+i) + B_{k+i}u(k+i) + M_{k+i}d(k+i) + H(k+i)\quad (4.7)$$

At time k , all the required informations to compute the linearization error $H(k+i)$ are available. The obtained model is time varying and is used in the formulation of the optimization problem.

The future control variables are defined as

$$\begin{aligned}u(k) &= u(k-1) + \delta u(k) \\ u(k+1) &= u(k) + \delta u(k+1) = u(k-1) + \delta u(k) + \delta u(k+1) \\ &\dots \\ u(k+N_p-1) &= u(k-1) + \delta u(k) + \delta u(k+1) + \dots + \delta u(k+N_p-1)\end{aligned}\quad (4.8)$$

Or defining the following matrices

$$U(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}, \quad \Delta U(k) = \begin{bmatrix} \delta u(k) \\ \delta u(k+1) \\ \vdots \\ \delta u(k+N_p-1) \end{bmatrix}, \quad \Phi_1 = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix}, \quad \Phi_2 = \begin{bmatrix} I_m & 0 & 0 & 0 \\ I_m & I_m & 0 & 0 \\ I_m & I_m & \ddots & 0 \\ I_m & I_m & I_m & I_m \end{bmatrix}$$

The set of equations (4.8) can be written in compact form as

$$U(k) = \Phi_1 u(k-1) + \Phi_2 \Delta U(k)\quad (4.9)$$

Defining the following matrices

$$X(k+1) = \begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+N_p) \end{bmatrix}, \quad D(k) = \begin{bmatrix} d(k) \\ d(k+1) \\ \vdots \\ d(k+N_p-1) \end{bmatrix}, \quad \bar{H}_k = \begin{bmatrix} H(k) \\ H(k+1) \\ \vdots \\ H(k+N_p-1) \end{bmatrix}$$

$$\bar{A}_k = \begin{bmatrix} A_k \\ A_{k+1}A_k \\ A_{k+2}A_{k+1}A_k \\ \vdots \end{bmatrix}, \quad \bar{B}_k = \begin{bmatrix} B_k & 0 & 0 & 0 \\ A_{k+1}B_k & B_{k+1} & 0 & 0 \\ A_{k+2}A_{k+1}B_k & A_{k+1}B_{k+1} & B_{k+2} & 0 \\ \dots & \dots & \dots & \ddots \end{bmatrix},$$

$$\bar{M}_k = \begin{bmatrix} M_k & 0 & 0 & 0 \\ A_{k+1}M_k & M_{k+1} & 0 & 0 \\ A_{k+2}A_{k+1}M_k & A_{k+1}M_{k+1} & M_{k+2} & 0 \\ \dots & \dots & \dots & \ddots \end{bmatrix}$$

the linearization of the system around the nominal trajectory (4.7) can be rewritten as

$$X(k+1) = \bar{A}_k x(k) + \bar{B}_k (\Phi_1 u(k-1) + \Phi_2 \Delta U(k)) + \bar{M}_k D(k) + \bar{H}_k \quad (4.10)$$

The optimization problem to be solved at each time step k is then defined as

$$\min_{\Delta u(k)} J = (X(k+1) - X^o)' Q_T (X(k+1) - X^o) + \Delta U'(k) R_T \Delta U(k) + U'(k) R_{uT} U(k) \quad (4.11)$$

The control matrices are defined as

$$Q_T = \begin{bmatrix} Q & 0 & \dots & 0 & 0 \\ 0 & Q & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & Q & 0 \\ 0 & 0 & \dots & 0 & S \end{bmatrix}, \quad R_T = \begin{bmatrix} R & 0 & \dots & 0 & 0 \\ 0 & R & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & R & 0 \\ 0 & 0 & \dots & 0 & R \end{bmatrix}, \quad R_{uT} = \begin{bmatrix} R_u & 0 & \dots & 0 & 0 \\ 0 & R_u & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & R_u & 0 \\ 0 & 0 & \dots & 0 & R_u \end{bmatrix}$$

where $Q = Q' \geq 0$, $R = R' > 0$, $R_u = R'_u > 0$, $S = S' \geq 0$ are matrices of suitable dimensions. As in the traditional MPC algorithm, constraints on the future state and control variables can be set in the form

$$\begin{aligned} X_{min} &\leq X(k) \leq X_{max} \\ U_{min} &\leq U(k) \leq U_{max} \\ \Delta U_{min} &\leq \Delta U(k) \leq \Delta U_{max} \end{aligned}$$

4.2 Control scheme implementation

In this thesis, the nonlinear model of the system considered is in the form

$$\begin{cases} \dot{x} = f_c(x(t), u(t), d(t)) \\ y(t) = g_c(x(t)) \end{cases} \quad (4.12)$$

where $x(t)$ are the same state variables considered in the Linear MPC case, $u(t)$ are the heat/coolant inputs, $y(t)$ are the output temperatures and $d(t)$ are the liquid holdups of the CSTR,

condenser and re-boiler. Defining the sampling time $T_s = 30[s]$, the discrete time model of the system is obtained using the Forward Euler method. The resulting model is

$$\begin{cases} x(k+1) = f(x(k), u(k), d(k)) \\ y(k) = g(x(k)) \end{cases} \quad (4.13)$$

As for the Linear MPC case, an integral action has been introduced in the control scheme, see Figure 4.1. The control algorithm is changed to consider the enlarged system. The nominal

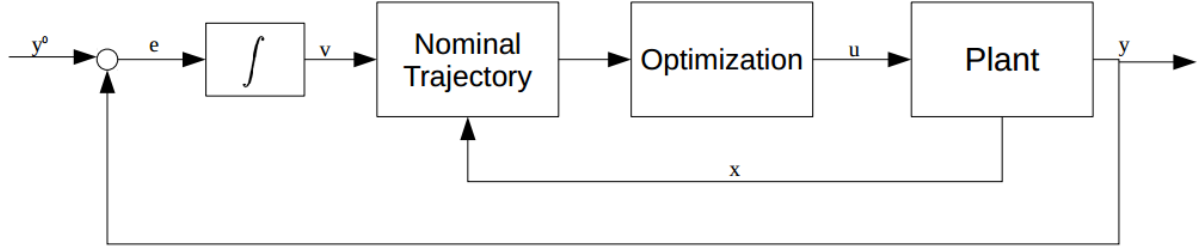


Figure 4.1: Trajectory MPC control scheme with integral action

trajectory is computed as shown in the previous paragraph 4.1. Defining $z(k)$ as the enlarged system state, as presented in Chapter 3, the obtained linearization around the nominal trajectory is

$$z(k+i+1) = \tilde{A}_{k+i}z(k+i) + \tilde{B}_{k+i}\delta u(k+i) + \tilde{M}_{k+i}\delta d(k+i) + \tilde{H}(k+i) \quad (4.14)$$

where

$$\tilde{A}_{k+i} = \begin{bmatrix} A_{k+i} & 0 \\ -C_{k+i}A_{k+i} & I_{p \times p} \end{bmatrix}, \quad \tilde{B}_{k+i} = \begin{bmatrix} B_{k+i} \\ C_{k+i}B_{k+i} \end{bmatrix}, \quad \tilde{M}_{k+i} = \begin{bmatrix} M_{k+i} \\ C_{k+i}M_{k+i} \end{bmatrix} \quad (4.15)$$

The matrices A_{k+i} , B_{k+i} and M_{k+i} are obtained, as in the previous case, from (4.5). Similarly, the matrices C_{k+i} are obtained from

$$C_{k+i} = \left. \frac{\partial g}{\partial x} \right|_{\hat{x}(k+i|k)} \quad (4.16)$$

Taking into consideration the enlarged state $z(k)$, the linearization error is computed as

$$\tilde{H}(k+i) = \begin{cases} H(k+i) \\ g(\hat{x}(k+i|k)) - C_{k+i}\hat{x}(k+i|k) \end{cases} \quad (4.17)$$

The first element is the linearization error in the evaluation of the state variables, computed using equation (4.6), while the second element computes the linearization error for the output variables. The optimization problem structure does not change with respect to the case presented in the previous paragraph. However, instead of the vector containing the future states $X(k+1)$, the vector containing the enlarged states $Z(k+1)$ is used.

$$Z(k+1) = \begin{bmatrix} z(k+1) \\ z(k+2) \\ \vdots \\ z(k+N_p) \end{bmatrix} \quad (4.18)$$

The optimization problem solved at each time step becomes

$$\min_{\Delta u(k)} J = Z'(k+1)QZ(k+1) + \Delta U'(k)R\Delta U(k) + U'(k)R_u U(k) \quad (4.19)$$

where the matrices \bar{A}_k , \bar{B}_k , \bar{M}_k and \bar{H}_k are obtained as

$$\bar{A}_k = \begin{bmatrix} \tilde{A}_k \\ \tilde{A}_{k+1}\tilde{A}_k \\ \tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{A}_k \\ \vdots \end{bmatrix}, \quad \bar{B}_k = \begin{bmatrix} \tilde{B}_k & 0 & 0 & 0 \\ \tilde{A}_{k+1}\tilde{B}_k & \tilde{B}_{k+1} & 0 & 0 \\ \tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{B}_k & \tilde{A}_{k+1}\tilde{B}_{k+1} & \tilde{B}_{k+2} & 0 \\ \dots & \dots & \dots & \ddots \end{bmatrix} \quad (4.20)$$

$$\bar{H}_k = \begin{bmatrix} \tilde{H}(k) \\ \tilde{H}(k+1) \\ \vdots \\ \tilde{H}(k+N_p-2) \end{bmatrix}, \quad \bar{M}_k = \begin{bmatrix} \tilde{M}_k & 0 & 0 & 0 \\ \tilde{A}_{k+1}\tilde{M}_k & \tilde{M}_{k+1} & 0 & 0 \\ \tilde{A}_{k+2}\tilde{A}_{k+1}\tilde{M}_k & \tilde{A}_{k+1}\tilde{M}_{k+1} & \tilde{M}_{k+2} & 0 \\ \dots & \dots & \dots & \ddots \end{bmatrix}$$

Recalling Equation (4.9), vector $Z(k+1)$ can be rewritten as

$$Z(k+1) = \bar{A}_k z(k) + \bar{B}_k(\Phi_1 u(k-1) + \Phi_2 \Delta U(k)) + \bar{M}_k D(k) + \bar{H}_k \quad (4.21)$$

The expressions of $Z(k+1)$ and $U(k)$, obtained from (4.21) and (4.9) respectively, are substituted in the cost function J of the optimization problem (4.19). Neglecting the elements that are independent from the control increments $\Delta U(k)$, an equivalent cost function \bar{J} is obtained

$$\begin{aligned} \bar{J} = & \Delta U'(k)\Phi_2'(\bar{B}_k'Q_T\bar{B}_k + R_T + R_{uT})\Phi_2\Delta U(k) + \\ & + 2(z'(k)\bar{A}_k'Q_T\bar{B}_k + u'(k-1)\Phi_1'(\bar{B}_k'Q_T\bar{B}_k + R_{uT}) + \bar{H}_k'Q_T\bar{B}_k)\Phi_2\Delta U(k) \end{aligned} \quad (4.22)$$

The optimization problem to be solved becomes

$$\min_{\Delta U(k)} \bar{J} = \min_{\Delta U(k)} (\Delta U'(k)Q_{uad}\Delta U(k) + L_{in}\Delta U(k)) \quad (4.23)$$

where

$$\begin{aligned} Q_{uad} = & \Phi_2'(\bar{B}_k'Q_T\bar{B}_k + R_T + R_{uT})\Phi_2 \\ L_{in} = & 2(z'(k)\bar{A}_k'Q_T\bar{B}_k + u'(k-1)\Phi_1'(\bar{B}_k'Q_T\bar{B}_k + R_{uT}) + \bar{H}_k'Q_T\bar{B}_k)\Phi_2 \end{aligned} \quad (4.24)$$

The following constraints have been considered for the problem

$$\Delta U_{min} \leq \Delta U(k) \leq \Delta U_{max} \quad (4.25)$$

where ΔU_{min} and ΔU_{max} are the vectors containing the minimum and maximum input possible variations of the heat/coolant inputs of the system. The optimization problem is in linear quadratic form and the solution is obtained using the Matlab[®] function *quadprog*.

4.2.1 Linearization around the nominal trajectory

The main challenge posed by the control algorithm is to compute the linearization around the nominal trajectory. Equation (4.3) is solved to find each value of the nominal trajectory. To simplify the problem of the nominal trajectory computation, instead of using the optimal control sequence values $u^o(k+i|k-1)$, the control inputs can be supposed to be constant and equal to the last imposed value $u(k-1)$. This approximation is useful to simplify the code used to compute the values of the nominal trajectory. At each time step k of the controller, the value $\hat{x}(k+i|k)$ of

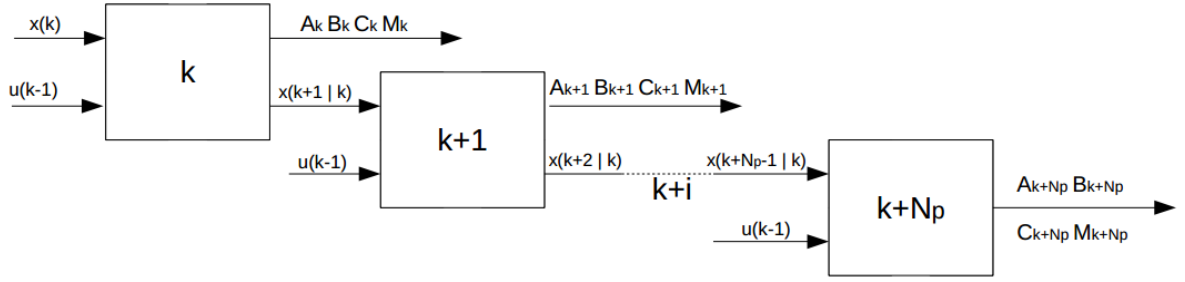


Figure 4.2: Nominal trajectory computation scheme

the nominal trajectory is obtained using only the control inputs $u(k-1)$ applied at the previous time step and the previous value of the nominal trajectory $\hat{x}(k+i-1|k)$. The computation of matrices A_{k+i} , B_{k+i} , C_{k+i} and M_{k+i} , as for equations (4.5) and (4.16), is done using the values of $\hat{x}(k+i|k)$ and $u(k-1)$. The same two inputs are required to compute the linearization error $H(k+i)$. For this reasons, the scheme in Figure 4.2 has been implemented. Inside each block, the value of the nominal trajectory computed in the previous block $\hat{x}(k+i-1|k)$ and the inputs $u(k-1)$, are used to compute the next step of the nominal trajectory $\hat{x}(k+i+1|k)$, the values of the linearized model matrices and the linearization error. To obtain the first value of the nominal trajectory, the first block is initialized with the measure of the states $x(k)$. In the scheme developed the number of blocks is equal to the value of N_p . Figure 4.3 shows in detail the scheme of each block. The nonlinear dynamics of the system are obtained using the forward Euler method on the continuous time model of the system

$$\hat{x}(k+i+1|k) = \hat{x}(k+i|k) + T_{s1} f_c(\hat{x}(k+i|k), u(k-1), d(k+i)) \quad (4.26)$$

The nonlinear system has been simulated using a sampling time $T_{s1} \ll T_s$. This choice has been made to reduce the numerical instability and improve the precision of the method used. Sampling the output of the simulation at $T_s = 30[s]$ the values of the nominal trajectory are obtained. The matrices of the linearized model along the nominal trajectory have been derived using the Matlab[®] symbolic toolbox. In the symbolic toolbox environment, the partial derivatives of the continuous time model of the system have been computed. Using this particular toolbox, it is possible to realize automatically a *Matlab function* block to be used in the Simulink environment. In the simulator the symbolic expression of the matrices are evaluated at each simulation step, their values are sampled at T_s and then discretized using the *c2d* Matlab[®] function. The obtained values of the matrices and $\hat{x}(k+i|k)$ are used in the block labeled as $H(k+i)$ to compute the value of the linearization error according to equation (4.17).

4.2.2 Optimization problem solution

The values of A_{k+i} , B_{k+i} , C_{k+i} and M_{k+i} are used to build the enlarged system matrices (4.15). The optimization problem matrices, shown in (4.20), are then assembled using the obtained enlarged system matrices. The optimization problem is solved inside a block that receives as inputs the values of the matrices \bar{A}_k , \bar{B}_k , \bar{M}_k and \bar{H}_k along with the values of $u(k-1)$ and $z(k)$. At each time step k the values of Q_{uad} and L_{in} are obtained according to the equations (4.24) and the optimization problem is solved using the Matlab[®] function *quadprog*.

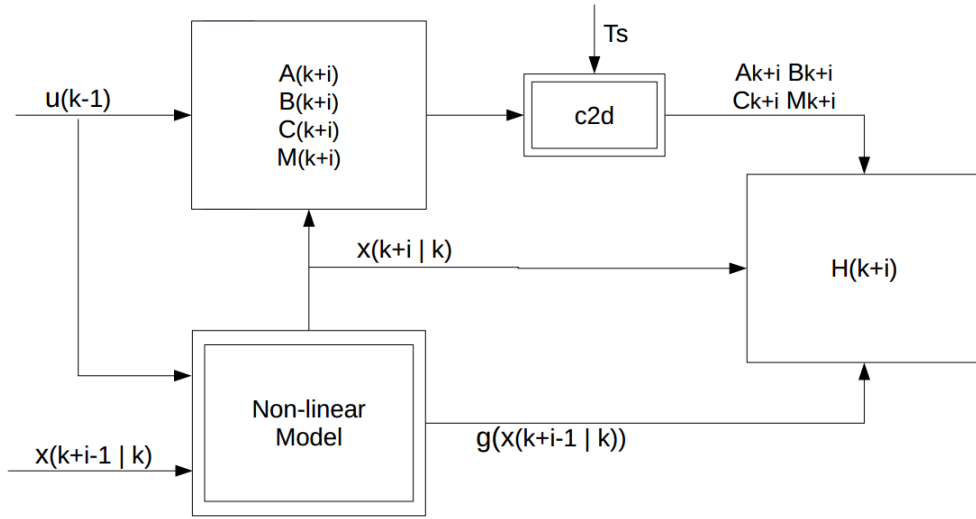


Figure 4.3: Nominal trajectory computation step

4.3 Challenges and limitations of the implementation

The procedure followed to implement the Trajectory Model Predictive Controller is

- The real time model of the system is realized using Simulink blocks
- The matrices of the linearized model of the continuous time system are computed using the Symbolic Toolbox of Matlab[®]
- The obtained matrices and the nonlinear model of the system are used to build the block shown in Figure 4.3, that computes the value of a single step in the nominal trajectory $\hat{x}(k+i|k)$ and the linearization error $H(k+i)$
- The nominal trajectory is obtained combining a number of the previous blocks equal to the prediction horizon N_p
- The optimization problem elements Q_{uad} and L_{in} are computed using the values of the matrices evaluated at each time step of the simulation
- The optimization problem is solved using the Matlab[®] function *quadprog*

The main challenge posed in the implementation of this particular control algorithm is the computation of the linearized system matrices. When dealing with systems with a large number of states and/or complex models, as in the considered case, the solution of the partial derivatives of equations (4.5) and (4.16) would be unfeasible without the use of a software. The possibility to automatically compute the partial derivatives of the nonlinear system model to obtain the linearized system matrices with the Symbolic Toolbox greatly simplifies the procedure to implement this control algorithm. The other advantage of this particular toolbox is the possibility to automatically generate the Simulink blocks to be used inside the simulator. In the developed simulator the nonlinear model of the system has been realized using *Matlab function* blocks. A possible alternative would have been to obtain the nonlinear model Simulink code using the same procedure followed for the linearized system matrices using the Symbolic Toolbox. This alternative could be of interest if the model of the system is very complex or a more compact

structure of the simulator is desirable. An important feature of the developed simulator, is that all the controller parameters values are easily accessible and modifiable without changing the implemented scheme. This characteristics is very important because it makes the operation of tuning of the controller a simple task. The main limitation found using this particular software is the difficulty to develop a scheme for the computation of the nominal trajectory whose structure is independent from the value of the prediction horizon N_p . However the realized scheme is easily modifiable if a different value of the prediction horizon is used. In fact the only necessary modification would be to add or remove the required number of identical blocks inside the nominal trajectory computation scheme shown in Figure 4.2.

4.4 Results

In this paragraph, the responses of the Trajectory MPC scheme are analyzed. The same inputs shown in chapter 3.4 are considered. First the responses to an input step for two different reference temperature are shown. Then the response to a variation in the condenser liquid holdup M_{cond} is considered. The amplitude of the steps is equal to +5% of the initial value and occur at time $t = 1000[s]$ for the temperatures and $t = 600[s]$ for the liquid holdup step input. To obtain comparable results with the classic MPC approach the tunable matrices Q and R have been chosen with the same criterion used in the previous case. Matrix R_u has been set equal to matrix R . The prediction horizon is set to $N_p = 10$, while the control one is $N_u = 3$.

4.4.1 Condenser reference temperature

The response to a step variation in the condenser reference temperature T_{cond}^o is analyzed. Figure 4.4 compares the responses of the condenser temperature T_{cond} obtained with the Trajectory MPC to the one obtained with the Linear MPC. Using the Trajectory MPC the response is faster, the new reference value is reached in less than 6000 seconds. The input increments δu plots are shown in Figure 4.5. The responses of T_{cstr} and T_{reb} are shown in Figures 4.6 and 4.7 respectively. In both cases the obtained response is faster than the Linear MPC case, but the initial variation caused by the initial reference change is slightly higher than the previous case.

4.4.2 Re-boiler reference temperature

In this section the effect of a step change in the re-boiler reference temperature T_{reb}^o is presented. The obtained response of the re-boiler temperature using the Trajectory MPC with respect to the Linear MPC response is almost identical, as shown in Figure 4.8. Similarly to the condenser reference temperature step variation case, the Trajectory MPC recovers the values of the other two temperatures slightly faster than the Linear MPC, as shown in Figures 4.10 and 4.11.

4.4.3 Response to the disturbances

The effect of a step variation in one of the liquid holdups is similar in all three cases. The effect of a +5% step variation of the liquid holdup contained in the condenser M_{cond} is shown in Figures 4.12, 4.13 and 4.14. The responses obtained with the Trajectory MPC are faster than the Linear MPC ones. The temperatures sustain a variation of less than 2 Kelvin before reaching their steady state values. The disturbance is easily recovered by the Trajectory MPC and the control cost required is very low, as shown in Figure 4.15.

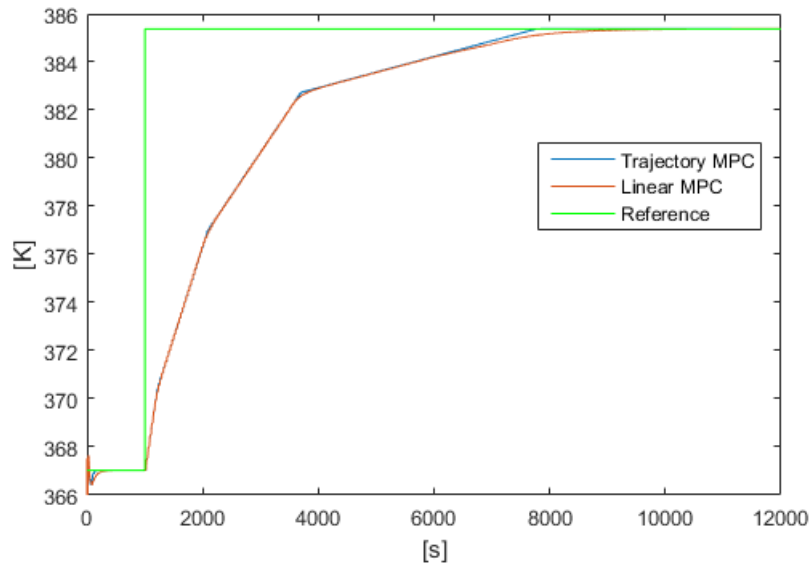


Figure 4.4: Comparison of the T_{cond} responses to a step variation of T_{cond}^o

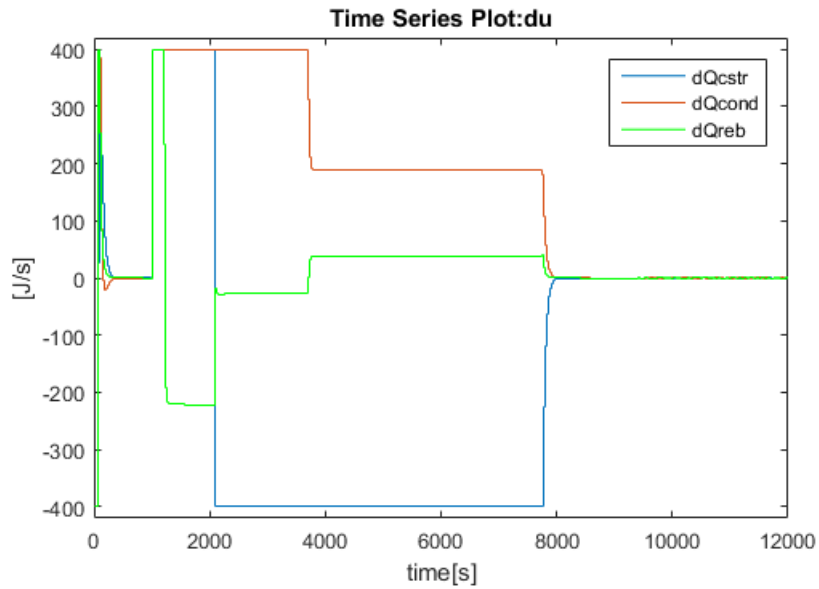


Figure 4.5: Input increments due to a step in T_{cond}^o

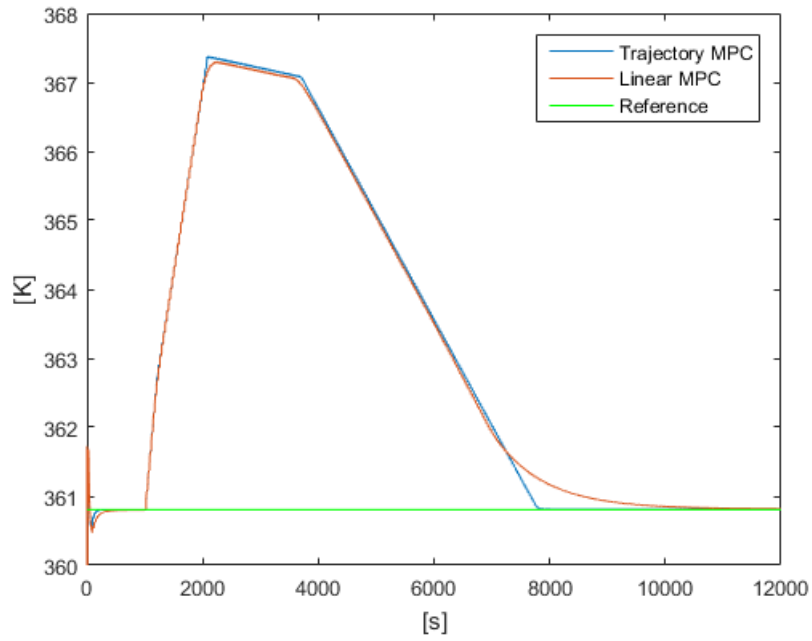


Figure 4.6: Comparison of the T_{cstr} responses to a step variation of T_{cond}^o

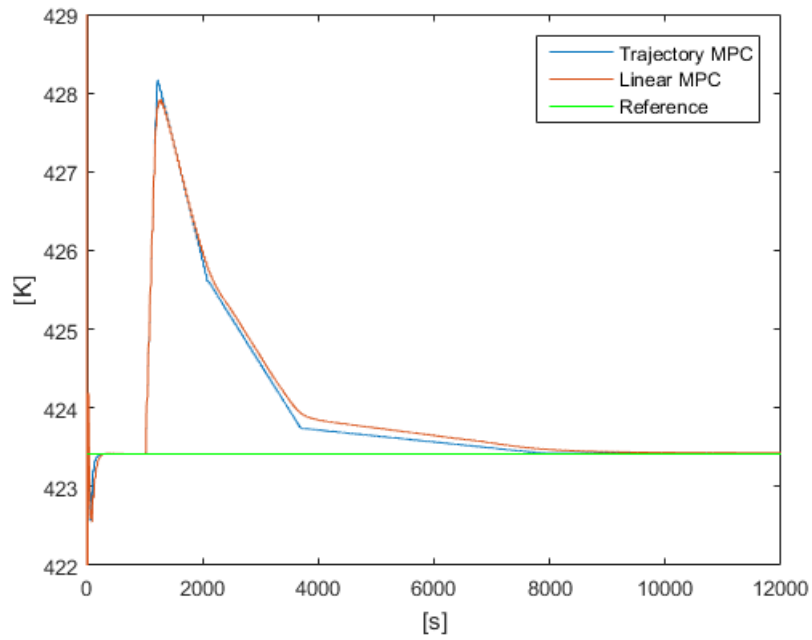


Figure 4.7: Comparison of the T_{reb} responses to a step variation of T_{cond}^o

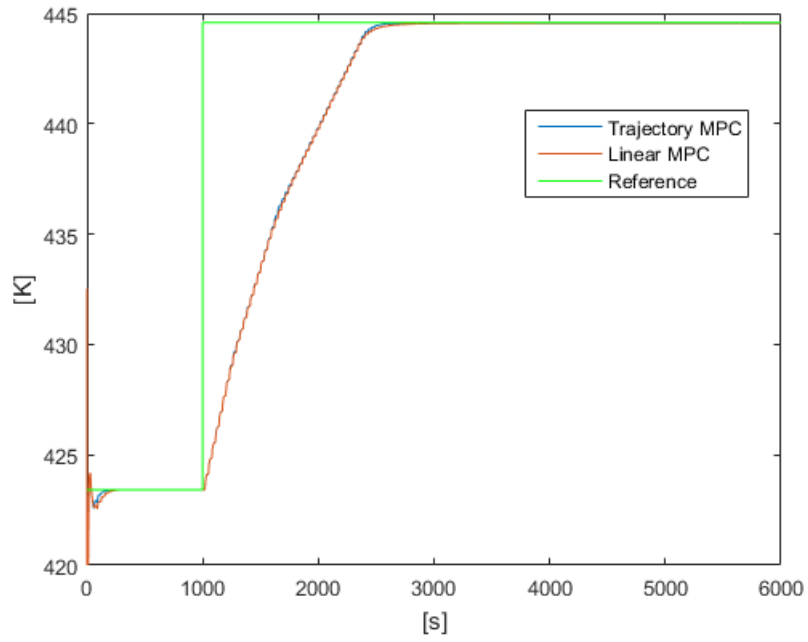


Figure 4.8: Comparison of the T_{reb} responses to a step variation of T_{reb}^o

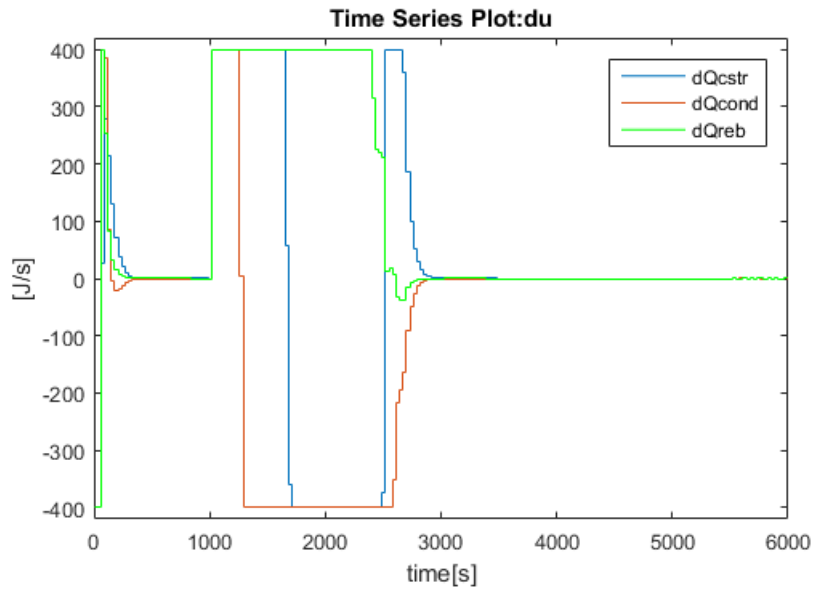


Figure 4.9: Input increments due to a step in T_{reb}^o

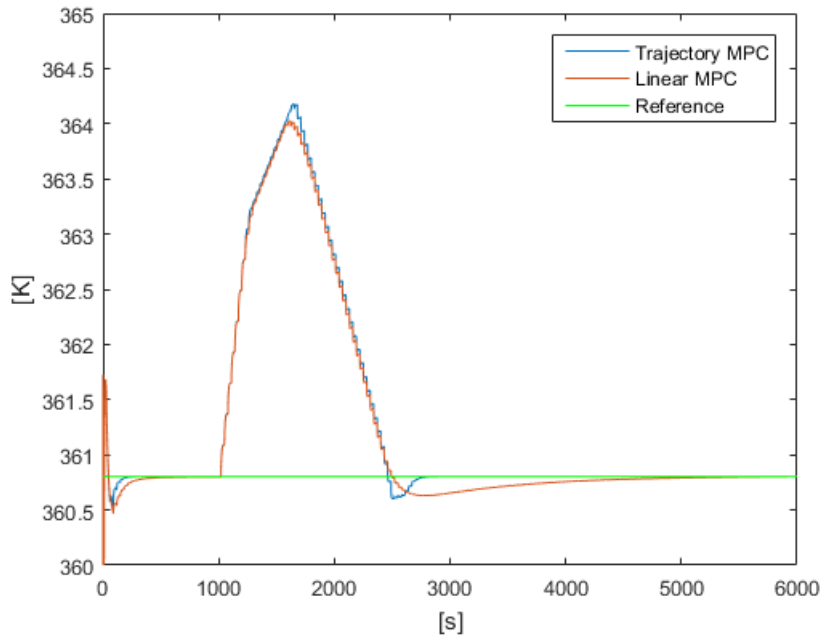


Figure 4.10: Comparison of the T_{cstr} responses to a step variation of T_{reb}^o

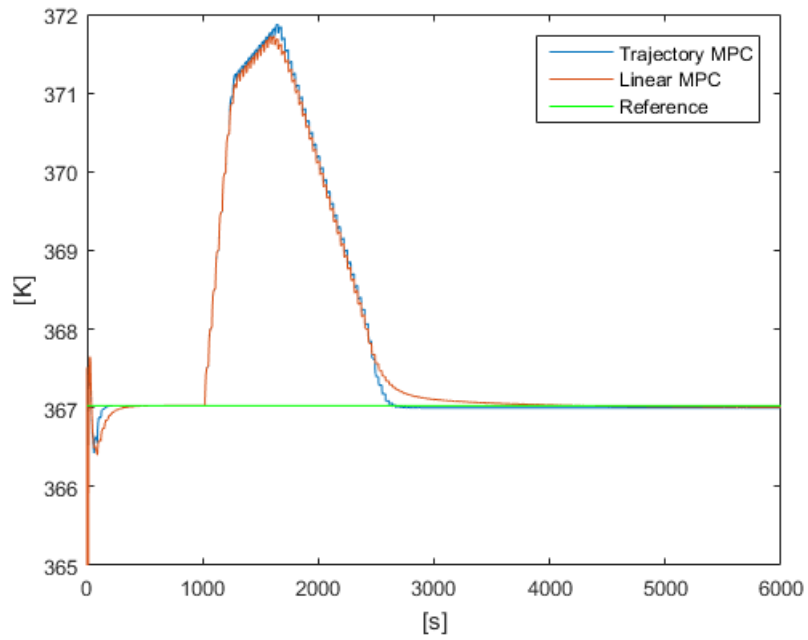


Figure 4.11: Comparison of the T_{cond} responses to a step variation of T_{reb}^o

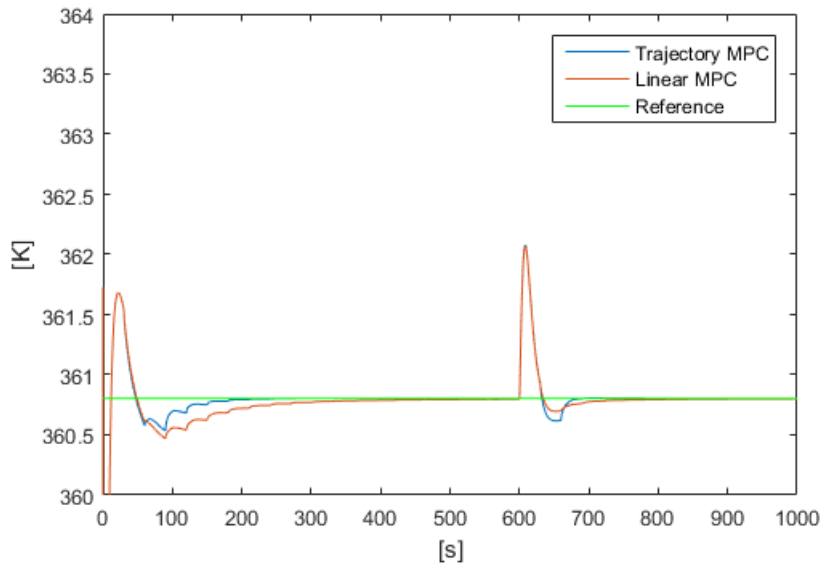


Figure 4.12: Comparison of the T_{cstr} responses to a step variation of M_{cond}

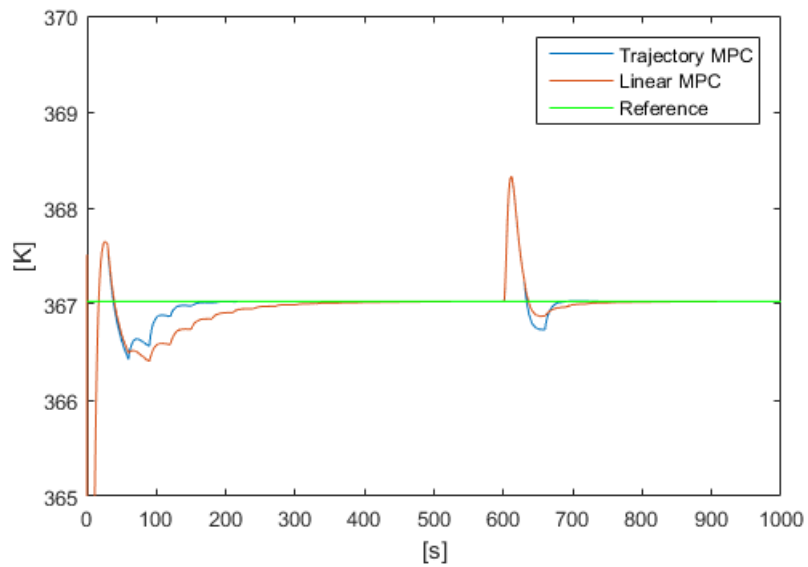


Figure 4.13: Comparison of the T_{cond} responses to a step variation of M_{cond}

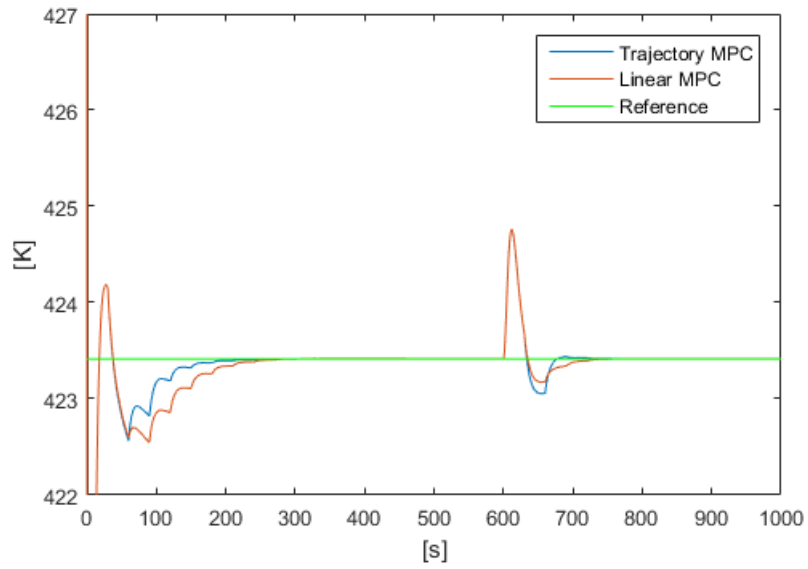


Figure 4.14: Comparison of the T_{reb} responses to a step variation of M_{cond}

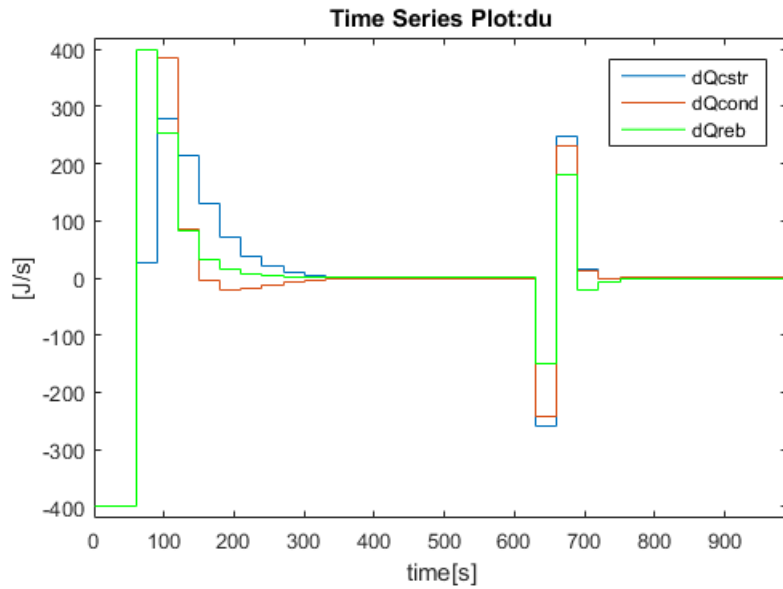


Figure 4.15: Input increments due to a step in M_{cond}

Chapter 5

Conclusion

In this thesis two model predictive controllers have been designed for the control of the temperatures in a reactor-distillation process network. First a MPC controller based on the linear model of the system has been designed. An integral action has been introduced to ensure reference signal tracking and improve the robustness of the controller with respect to external disturbances. For this controller the system states have been considered unmeasurable. A Kalman Filter has been designed to estimate the value of the states to be used in the solution of the optimization problem. The designed controller is able to follow precisely the reference signals and counteract effectively the effect of external disturbances. The second controller has been designed based on the time-variant model obtained linearizing the system along the nominal trajectory computed at each time step of the controller. The objective set for this thesis was to develop a systematic procedure to implement this controller. The main goals of the procedure where:

- the symbolic linearized system matrices expressions should be obtained using software tools
- the obtained procedure should be independent on the system model
- the control structure should be independent on the control parameters

The first goal is related to the fact that the computation of the linearized system matrices without using software tools can be difficult when dealing with systems with a large number of states and/or complex models. The second and third goals have been set to obtain a procedure that can be applied to any controlled plant. In chapter 4 the designed procedure to realize a Trajectory MPC with an integral action using the Matlab[®] Simulink software, is shown. The main steps of the proposed procedure are

- realize the nonlinear model using Simulink blocks or the Symbolic Toolbox
- compute the expressions of the linearized system matrices using the Symbolic Toolbox
- use the models above to compute the nominal trajectory at each step of the controller
- compute the model of the system linearized over the nominal trajectory
- solve an optimization problem based on the linearized model

The obtained procedure has been designed with the proposed goals in mind. The choice of using the Matlab[®] Symbolic Toolbox to realize the second step of the procedure gives many advantages. First is the possibility to easily write the nonlinear system model equations and solve the partial derivatives required to obtain the expressions of the linearized model matrices. The

second advantage is the possibility to realize automatically, using the same toolbox, the blocks to be used inside the Simulink environment. The obtained control scheme is not independent on the value of the prediction horizon N_p , because the number of blocks that compute the nominal trajectory is equal to its value. However the scheme is easily adaptable if the value of N_p changes. The Trajectory MPC control scheme implemented gives slightly better results than the first controller with respect to the reference signal tracking and disturbance rejection.

In conclusion, Model Predictive Control is a family of powerful control algorithms widely used in the industrial field for many different applications. Nonlinear MPC algorithms are a possible solution to some of the issues related to the traditional MPC algorithms based on the linearized model of the system. In this thesis a systematic procedure has been designed for the realization of the nonlinear Model Predictive Control algorithm based on a time-variant linearized model of the system. The proposed procedure has been designed to be applicable to many different plant models and utilizes the Matlab[®] Symbolic Toolbox to compute the values of the nominal trajectory and obtain the linearized model of the system.

Bibliography

- [1] X. Chen, M. Heidarinejad, J. Liu, D. Munõz de la Peña, P.D. Christofides, Model predictive control of nonlinear singularly perturbed systems: Application to a large-scale process network, *Journal of Process Control* 21 (2011) 1296-1305.
- [2] A. Kumara, P. Daoutidis, Nonlinear dynamics and control of process systems with recycle, *Journal of Process Control* 12 (2002) 475-484
- [3] M. Morari, J.H. Lee, Model predictive control: past, present and future, *Computers and Chemical Engineering* 23 (1999), 667-682
- [4] S.J. Qina, T.A. Badgwell, A survey of industrial model predictive control technology, *Control Engineering Practice* 11 (2003) 733-764
- [5] P. Falcone, F. Borrelli, H.E. Tseng, J. Asgari, D. Hrovat, Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation, *Int. J. Robust Nonlinear Control* 18 (2008) 862-875
- [6] L. Magni, R. Scattolini, *Advanced and multivariable control*, 1st ed. Bologna: Pitagora Editrice Bologna; 2014