POLITECNICO DI MILANO

Scuola di Ingegneria dell’Informazione

Master of Science in Automation Engineering

Analysis and Implementation of Model Predictive Control (MPC) in Chemical Plant

Supervisor: Prof. Riccardo SCATTOLINI

Master Graduation Thesis by:

Shahab Reza BESHARAT

Student Id. number 764488

Academic Year 2012/13

Abstract

Model based Predictive Control (MPC) is a form of control that has gained widespread acceptance in chemical industry due to its unique advantages compared to classic control methods. The main distinguishing features are the ability to efficiently control large scale interconnected systems and the inherent ability to cope with physical and other constraints of the controlled system. MPC controllers are designed on the basis of a dynamical model of the system that has to be controlled (i.e., the plant) and apply mathematical optimization techniques in order to obtain the optimal inputs to be applied to the plant. MPC acquires the current control action by solving, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant. In this thesis the focus is on linear MPC algorithms, i.e., MPC algorithms that can take the regulation problem. More specifically, the main aim of this thesis is the development of MPC algorithms in the environment of MATLAB and Simulink that can take input and output constraints into account and can guarantee stable behavior and acceptable performance. These aims are achieved by making improved algorithms for the construction of required matrixes and contributions to the constraints. On the level of stability, celebrated terminal state is enabled the pledge. We concentrate our attention on a plant with two reactors and a separator as a sample of chemical plant in order to enrich the experimental results. Several simulations on the model of this process show the improved properties of the obtained program.

Keywords:
Model predictive control; chemical plant; CSTR; optimal control; constraint; stability

Sommario

Il controllo predittivo, o MPC (Model Predictive Control) è una tecnica di sintesi di sistemi di controllo che ha riscosso un notevole successo nell'industria chimica per I vantaggi che può offrire rispetto a metodi classici, quali l'assegnamento degli autovalori o il controllo LQR. Le sue principali caratteristiche sono la possibilità di considerare sistemi con grandi dimensioni, tipicamente costituiti da sottosistemi interconnessi, a la capacità di tenere conto in modo esplicito di eventuali vincoli sulle variabili di processo o su altre variabili di interesse. I controllori MPC sono di tipo "model-based", cioè sono progettati a partire da un modello del processo sotto controllo, e si basano sulla formulazione di un opportuno problema di ottimizzazione da risolversi a ogni istante di campionamento per determinare il valore da imporre alle variabili di controllo. Più nello specifico, viene risolto un problema di controllo su orizzonte finito dove lo stato corrente è considerato come stato iniziale. La soluzione del problema di ottimo consiste nel determinare la sequenza di controlli ottimi da imporre, almeno teoricamente, lungo tutto l'intervallo considerato. Tuttavia, si implementa effettivamente soltanto il primo valore di questa sequenza e l'intera procedura è ripetuta al successivo istante di campionamento. In questo lavoro di Tesi si sviluppano in ambiente Matlab/Simulink due algoritmi MPC per sistemi lineari e se ne analizzano in dettaglio le caratteristiche e prestazioni. Entrambi gli algoritmi consentono di considerare vincoli sullo stato, sulle variabili di controllo e sulle uscite regolate, così da garantire stabilità e prestazioni. Per quanto riguarda la stabilità, è possibile introdurre nel problema di ottimizzazione un opportuno peso sullo stato finale raggiunto dal sistema al termine dell'orizzonte di predizione, vincolandolo anche ad appartenere a un dato insieme. Le caratteristiche di questi metodi MPC sono confrontate con riferimento a un processo, costituito da due reattori e da un separatore, in grado di rappresentare bene le problematiche tipiche di un impianto industriale. Nel lavoro sono riportate e commentate numerose simulazioni che consentono di mostrare le prestazioni ottenibili con questo approccio alla sintesi del controllore.

Keywords:
Model predictive control; chemical plant; CSTR; optimal control; constraint; stability

Acknowledgments

This thesis presents the results of the study carried out in the means of graduation on Master of Science in Automation engineering at Politecnico di Milano under the supervision of Prof. Riccardo Scattolini.

I would like to address a special word of gratitude to Prof. Scattolini; without whom the subject gathered in this thesis would have not been possible. Throughout my studies he took an active interest in my work and encouraged me to keep at it when I couldn’t see the light at the end of the tunnel. I will always be grateful for his investment of patience, time and effort in my work.

I would also like to take this opportunity to thank Prof. Sergio Bittanti and Prof. Paolo Rocco for allowing me to contribute in the IFAC Milano 2011 as a volunteer student. It was one of my greatest academicals experience.

A special thanks goes to Prof. Franco Zappa for his entire support to international students during the first days of university life in Milano and in the course “Electronic System”, at any time.

I am grateful to Prof. Marco Lovera and Prof. Alberto Leva for their astonishing classes in “Advanced and Multivariable Control” and “Automation of Energy systems” which were sources of inspiration for me.

During the two years I have studied on my Master program, I had the opportunity to spend enjoyable moments in the company of friends and colleagues. I would like to thank all of them.

I am eternally indebted to my wife Hananeh for her endless love and support and our parents for all their best wishes.

This thesis is dedicated to my wife.

Shahab R. Besharat
Milano, December, 2012

Contents

List of Figures

List of Tables

Notation

$\mathrm{a, b, c} \in \mathbb{R}^{\mathrm{n}}$	Lower case roman symbols denote scalar or vector variables
$\mathrm{A, B, C} \in \mathbb{R}^{\mathrm{n\times m}}$	Upper case roman symbols denote matrix variables
$\mathcal{A}, \mathcal{B} \in \mathbb{R}^{\mathrm{n\times m}}$	Upper case Unicode extended characters denote augmented matrices
$\mathbb{R},\ \mathbb{R}^{+}$	Set of real numbers and positive real numbers
$\coloneqq$	Assignment $\mathrm{a} \coloneqq \mathrm{b} \Leftrightarrow$ the value of *b* is assigned to variable *a*
$>, \geq$	(Strict) scalar inequality $a > b \Leftrightarrow a - b$ has strictly positive elements
$>, \geq$	(Strict) matrix inequality $A > 0 \Leftrightarrow A$ is strictly positive definite
$u(k) \in \mathbb{R}^{\mathrm{m}}$	*m*-dimensional input vector at discrete time k
$x(k) \in \mathbb{R}^{\mathrm{n}}$	*n*-dimensional state vector at discrete time k
$y(k) \in \mathbb{R}^{\mathrm{p}}$	*p*-dimensional output vector at discrete time k
$d(k) \in \mathbb{R}^{\mathrm{d}}$	*d*-dimensional disturbance vector at discrete time k
$\bar{u}, \bar{x}, \bar{y}$	Steady state values for inputs, states and outputs
y^0	Reference values for outputs or setpoint
N	Prediction horizon length
N_u	Control horizon length
A', A^T	Transpose of matrix *A*
$\|x\|_Q^2$	Weighted 2-norm of a vector: $\sqrt{x'Qx}$
min_x	Function minimization over *x*, optimal function value is returned

Motto:

“There is nothing more practical than a good theory.”

-Boltzmann

1 Introduction

In this age of globalization, the realization of production innovation and highly stable operation is the chief objective of the process industry. Obviously, modern advanced control plays an important role to achieve this target, but the key to success is the maximum utilization of PID control and conventional advanced control. It is obvious that the three central pillars of process control – PID control, conventional advanced control, and linear/nonlinear model predictive control – have been widely used and they have to be contributed toward increasing productivity.

Model predictive control (MPC) or receding horizon control (RHC) is a form of control in which the current control action is obtained by solving on-line, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant. This is its main difference from conventional control which uses a pre-computed control law.

Model predictive control is one of few suitable methods, and this fact makes it an important tool for the control engineer, particularly in the process industries where plants being controlled are sufficiently slow to permit its implementation. Other examples where model predictive control may be advantageously employed include unconstrained nonlinear plants, for which on-line computation of a control law usually requires the plant dynamics to possess a special structure, and time-varying plants.

1.1 Motivation

This thesis aims to reveal the principal of model predictive control, its formulation, historical issues and some practical tricks. Finally, by implementing this theory in a typical chemical process, as a plant consisting of two reactors and a separator in the Simulink environment, try to represent the advantages of this state of the art method. To motivate the idea of using MPC control, here are some fact according to the international surveys and researches.

The first fact is coming from the article “A survey of industrial model predictive control technology” by Quin and Badgwell (2003). According to their research, in Tables 1 and 2, where more than 4600 total MPC applications are reported, MPC technology can now be found in a wide variety of application areas. The largest single block of applications is in refining, which amounts to 67% of all classified applications. This is also one of the original application areas where MPC technology has a solid track record of success. A significant number of applications can also be found in petrochemicals and chemicals, although it has taken longer for MPC technology to break into these areas.

Significant growth areas include the chemicals, pulp and paper, food processing, aerospace and automotive industries.

Table 1: Summary of linear MPC applications by areas

Summary of linear MPC applications by areas (estimates based on vendor survey; estimates do not include applications by companies who have licensed vendor technology)[a]

Area	Aspen Technology	Honeywell Hi-Spec	Adersa[b]	Invensys	SGS[c]	Total
Refining	1200	480	280	25		1985
Petrochemicals	450	80	—	20		550
Chemicals	100	20	3	21		144
Pulp and paper	18	50	—	—		68
Air & Gas	—	10	—	—		10
Utility	—	10	—	4		14
Mining/Metallurgy	8	6	7	16		37
Food Processing	—	—	41	10		51
Polymer	17	—	—	—		17
Furnaces	—	—	42	3		45
Aerospace/Defense	—	—	13	—		13
Automotive	—	—	7	—		7
Unclassified	40	40	1045	26	450	1601
Total	1833	696	1438	125	450	4542
First App.	DMC:1985 IDCOM-M:1987 OPC:1987	PCT:1984 RMPCT:1991	IDCOM:1973 HIECON:1986	1984	1985	
Largest App.	603 × 283	225 × 85	—	31 × 12	—	

[a]The numbers reflect a snapshot survey conducted in mid-1999 and should not be read as static. A recent update by one vendor showed 80% increase in the number of applications.
[b]Adersa applications through January 1, 1996 are reported here. Since there are many embedded Adersa applications, it is difficult to accurately report their number or distribution. Adersa's product literature indicates over 1000 applications of PFC alone by January 1, 1996.
[c]The number of applications of SMOC includes in-house applications by Shell, which are unclassified. Therefore, only a total number is estimated here.

Table 1 shows that AspenTech and Honeywell Hi-Spec were highly focused in refining and petrochemicals, with a handful of applications in other areas. Adersa and Invensys apparently had a broader range of experience with applications in the food processing, mining/metallurgy, aerospace and automotive areas, among others. The applications reported by Adersa include a number of embedded PFC (predictive functional control) applications, so it is difficult to report their number or distribution. While only a total number was reported by SGS, this includes a number of in-house SMOC (Shell multivariable optimizing controller) applications by Shell, so the distribution is likely to be shifted towards refining and petrochemical applications.

The bottom of Table 1 lists the largest linear MPC applications to date by each vendor, in the form of (outputs)_(inputs). The numbers show a difference in philosophy that is a matter of some controversy.

AspenTech prefers to solve a large control problem with a single controller application whenever possible; they report an olefins application with 603 outputs and 283 inputs. Other vendors prefer to break the problem up into meaningful sub-processes.

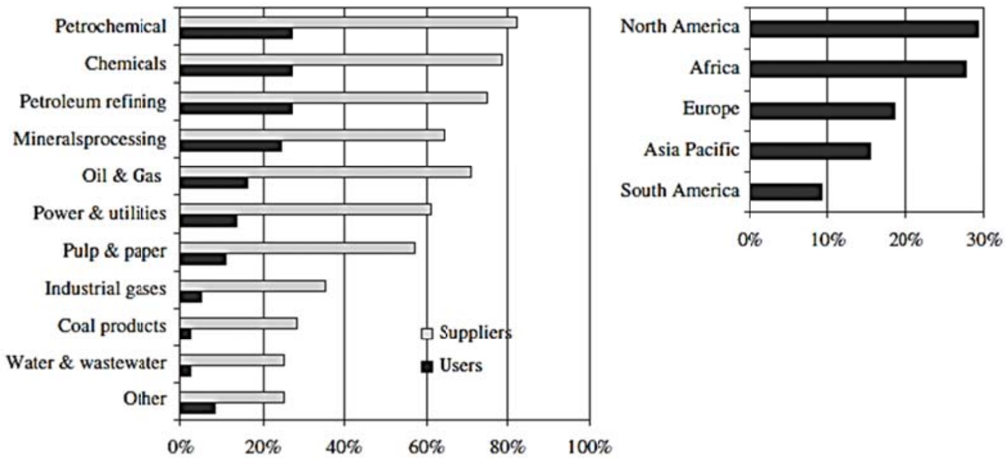
The nonlinear MPC (NMPC) applications reported in Table 2 are spread more evenly among a number of applications areas. Areas with the largest number of reported NMPC applications include chemicals, polymers, and air and gas processing. It has been observed that the size and scope of NMPC applications are typically much smaller than that of linear MPC applications. This is likely due to the computational complexity of NMPC algorithms.

Table 2: Summary of NMPC applications by areas

Summary of nonlinear MPC applications by areas (estimates based on vendor survey; estimates do not include applications by companies who have licensed vendor technology)

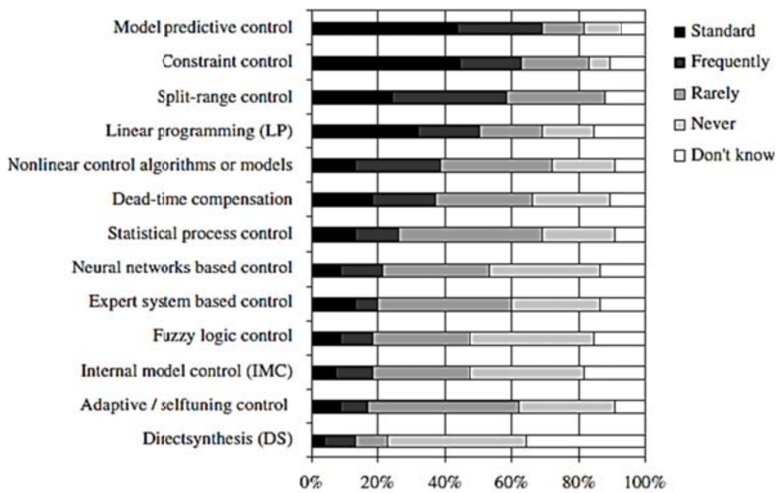
Area	Adersa	Aspen Technology	Continental Controls	DOT Products	Pavilion Technologies	Total
Air and Gas			18			18
Chemicals	2		15		5	22
Food Processing					9	9
Polymers		1		5	15	21
Pulp & Paper					1	1
Refining					13	13
Utilities		5	2			7
Unclassified	1		1			2
Total	3	6	36	5	43	93

In another work by Bauer and Craig (2007) "Economic assessment of advanced process control-A survey and framework" it has been indicated that In order to justify the cost associated with the introduction of new advanced process control (APC) technologies to a process, the benefits have to be quantified in economic terms and that paper reviews these methods and incorporates them in a framework for the economic evaluation of APC projects. Figure 1 shows the participants of APC survey by industrial and by continent and Figure 2 is about the industrial use of APC methods.



Participants of APC survey by industry and by continent (total: 66 participants). Several answers were allowed for industry sectors.

Figure 1: Participants of APC survey by industry and by continent



Industrial use of APC methods: survey results.

Figure 2: Industrial use of APC methods

Finally to conclude about this motivation, recent paper of Kanu and Ogawa (2010) in the Journal of Process Control as “The state of the art in chemical process control in Japan” is going to be referred. It has been noted that implementation of MPC, releases operators from most of the adjustment work they had to do in the past, because the optimal operating condition is automatically determined and maintained under disturbances. In addition, MPC makes it possible to maximize production rate by making the most use of the capability of the process and to minimize cost through energy conservation by moving the operating condition toward the control limit. Both the energy conservation and the productive capacity were improved by an average of 3 to 5% as the result of APC projects centered on MPC at Mitsubishi Chemical Corporation (MCC).

At the end of that paper, the MPC application for energy conservation and production maximization of the olefin unit at MCCMizushima plant is briefly explained which is the largest MPC application in the world, consisting of 283 manipulated variables and 603 controlled variables. The subsequent two commentaries are appreciated to declaim.

A skilled operator made the following comment on this MPC application: “We had operated the Ethylene fractionator in constant pressure mode for more than 20 years. I was speechless with surprise that we had made an enormous loss for many years, when I watched the MPC decreased the column pressure, improved the distillation efficiency, and maximized the production rate.” Another process control engineer said “I had misunderstood that setpoints were determined by operation section and process control section took the responsibility only for control. I realized MPC for the first time; it makes the most use of the capability of equipments, determines setpoints for economical operation, and maintains both controlled variables and manipulated variables close to the setpoints."

2 Dynamic model of chemical process

2.1 Introduction to process control

In recent years the performance requirements for process plants have become increasingly difficult to satisfy. Stronger competition, tougher environmental and safety regulations, and rapidly changing economic conditions have been key factors in tightening product quality specifications. A further complication is that modern plants have become more difficult to operate because of the trend toward complex and highly integrated processes. For such plants, it is difficult to prevent disturbances from propagating from one unit to other interconnected units.

In view of the increased emphasis placed on safe, efficient plant operation, it is only natural that the subject of process control has become increasingly important in recent years. Without computer-based process control systems it would be impossible to operate modern plants safely and profitably while satisfying product quality and environmental requirements. Thus, it is important for Automation engineers to have an overview about the process control.

2.2 Process dynamic

The term process dynamics refers to unsteady-state (or transient) process behavior. By contrast, most of the engineering may emphasize steady-state and equilibrium conditions in subjects as material and energy balances, thermodynamics, and transport phenomena. But process dynamics are also very important. Transient operation occurs during important situations such as start-ups and shutdowns, unusual process disturbances, and planned transitions from one product grade to another.

2.3 Process control

The primary objective of process control is to maintain a process at the desired operating conditions, safely and efficiently, while satisfying environmental and product quality requirements. The subject of process control is concerned with how to achieve these goals. In large-scale, integrated processing plants such as oil refineries or ethylene plants, thousands of process variables such as compositions, temperatures, and pressures are measured and must be controlled. Fortunately, large numbers of process variables (mainly flow rates) can usually be manipulated for this purpose. Feedback control systems compare measurements with their desired values and then adjust the manipulated variables accordingly.

The foundation of process control is process understanding. Thus, we continue with a basic question-What is a process? For our purposes, a brief definition is appropriate:

Process: The conversion of feed materials to products using chemical and physical operations. In practice, the term process tends to be used for both the processing operation and the processing equipment.

Note that this definition applies to three types of common processes: continuous, batch, and semibatch.

Following, we consider representative of continuous processes which is a main subject of this thesis. Here briefly summarize key control issues.

The process control problem has been characterized by identifying three important types of process variables.

• Controlled variables (CVs): The process variables that are controlled. The desired value of a controlled variable is referred to as its set point.

• Manipulated variables (MVs): The process-Variables that can be adjusted in order to keep the controlled variables at or near their set points. Typically, the manipulated variables are flow rates.

• Disturbance variables (DVs): Process variables that affect the controlled variables but cannot be manipulated. Disturbances generally are related to changes in the operating environment of the process, for example, its feed conditions or ambient temperature. Some disturbance variables can be measured on-line, but many cannot.

The specification of CVs, MVs, and DVs is a critical step in developing a control system. The selections should be based on process knowledge, experience, and control objectives.

2.4 The hierarchy of process control activities

As mentioned earlier, the chief objective of process control is to maintain a process at the desired operating conditions, safely and efficiently, while satisfying environmental and product quality requirements.

If we emphasized one process control activity and try to keeping controlled variables at specified set points, there are other important activities, also that we will now briefly describe.

In Fig.3 the process control activities are organized in the form of a hierarchy with required functions at the lower levels and desirable, but optional, functions at the higher levels. The time scale for each activity is shown on the left side of Fig.3. Note that the frequency of execution is much lower for the higher-level functions.

2.4.1 Measurement and actuation (Level 1)

Measurement devices (sensors and transmitters) and actuation equipment (for example, control valves) are used to measure process variables and implement the calculated control actions. These devices are interfaced to the control system. Clearly, the measurement and actuation functions are an indispensable part of any control system.

2.4.2 Safety and environmental/equipment protection (Level 2)

The Level 2 functions play a critical role by ensuring that the process is operating safely and satisfies environmental regulations. Generally, process safety relies on the principle of multiple protection layers that involve groupings of equipment and human actions. One layer includes process control functions, such as alarm management during abnormal situations, and safety instrumented systems for emergency shutdowns. The safety equipment operates independently of the regular instrumentation used for regulatory control in Level 3a.

2.4.3 Regulatory control (Level 3a)

As mentioned earlier, successful operation of a process requires that key process variables such as flow rates, temperatures, pressures, and compositions be operated at, or close to, their set points. This Level 3a activity, regulatory control, is achieved by applying standard feedback and feedforward control techniques. If the standard control techniques are not satisfactory, a variety of advanced control techniques are available.

2.4.4 Multivariable and constraint control (Level 3b)

Many difficult process control problems have two distinguishing characteristics: (i) significant interactions occur among key process variables, and (ii) inequality constraints exist for manipulated and controlled variables. The inequality constraints include upper and lower limits. Limits on controlled variables reflect equipment constraints and the operating objectives for the process.

The ability to operate a process close to a limiting constraint is an important objective for advanced process control. For many industrial processes, the optimum operating condition occurs at a constraint limit. For these situations, the set point should not be the constraint value because a process disturbance could force the controlled variable beyond the limit. Thus, the set point should be set conservatively, based on the ability of the control system to reduce the effects of disturbances.

The standard process control techniques of Level 3a may not be adequate for difficult control problems that have serious process interactions and inequality constraints. For these situations, the advanced control techniques of Level 3b, multivariable control and constraint control, should be considered. In particular, the model predictive control (MPC) strategy was developed to deal with both process interactions and inequality constraints. MPC is the main subject of this thesis report.

2.4.5 Real-time optimization (Level 4)

The optimum operating conditions for a plant are determined as part of the process design. But during plant operations, the optimum conditions can change frequently owing to changes in equipment availability, process disturbances, and economic conditions. Consequently, it can be very profitable to recalculate the optimum operating conditions on a regular basis. The new optimum conditions are then implemented as set points for controlled variables.

The Level 4 activities also include data analysis to ensure that the process model used in the RTO calculations is accurate for the current conditions.

2.4.6 Planning and scheduling (Level 5)

The highest level of the process control hierarchy is concerned with planning and scheduling operations for the entire plant. For continuous processes, the production rates of all products and intermediates must be planned and coordinated based on equipment constraints, storage capacity, sales projections, and the operation of other plants, sometimes on a global basis. Thus, planning and scheduling activities pose difficult optimization problems that are based on both engineering considerations and business projections.

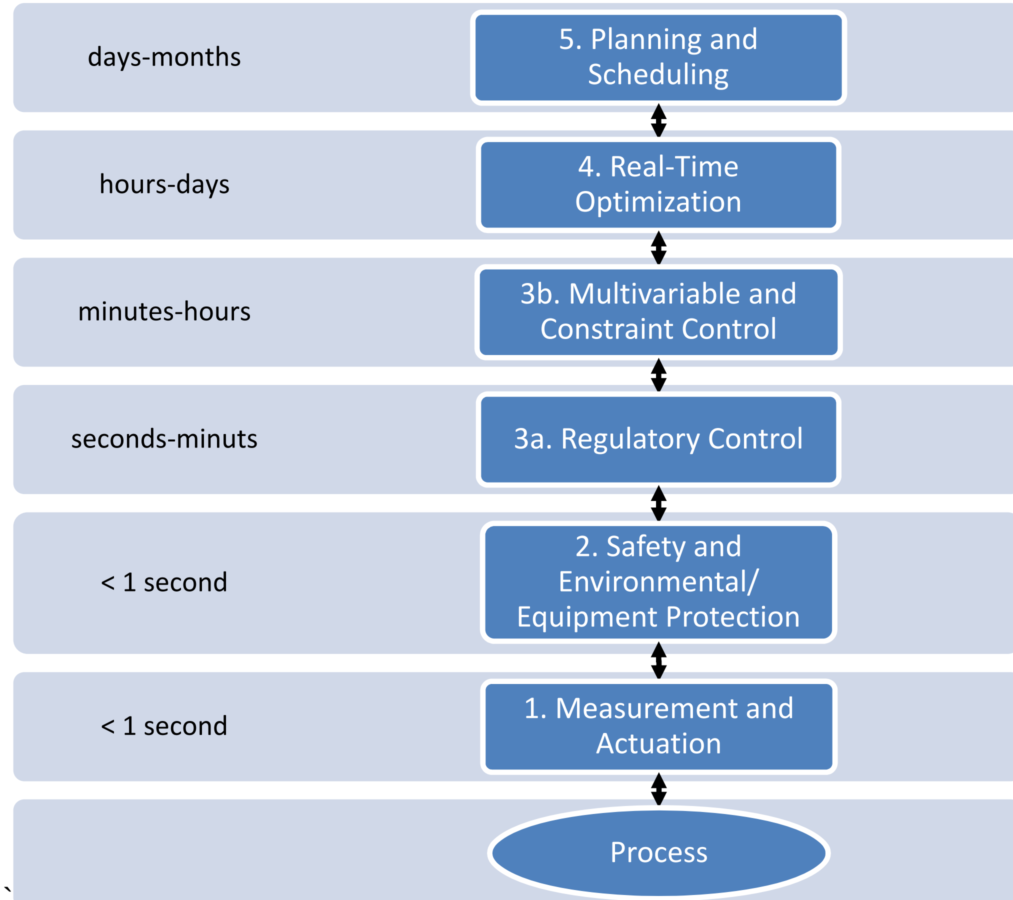


Figure 3: Hierarchy of process control activities

2.5 Continuous stirred tank reactor models

Continuous stirred-tank reactors have widespread application in industry and embody many features of other types of reactors. Chemical reactors are the most influential and therefore important units that a chemical engineer will encounter. To ensure the successful operation of a continuous stirred tank reactor (CSTR), it is necessary to understand their dynamic characteristics. A good understanding will ultimately enable effective control systems design. Consequently, a CSTR model provides a convenient way of illustrating modeling principles for chemical reactors.

To describe the dynamic behavior of a CSTR, mass, component and energy balance equations must be developed. This requires an understanding of the functional expressions that describe chemical reaction. A reaction will create new components while simultaneously reducing reactant concentrations. The reaction may give off heat or my require energy to proceed. Hereafter the goal is a short introduction of equations and modeling, while entire specification and circumstances are available in any chemical dynamic references.

2.5.1 The mass balance

Rate of mass flow in – Rate of mass flow out = Rate of change of mass within system

Consider a well-mixed tank of liquid (Figure 4). The inlet stream flow is *Fin* with density *ρin*. The volume of the liquid in the tank is *V*, with constant density *ρ*. The flow leaving the tank is *F* with liquid density *ρ*.

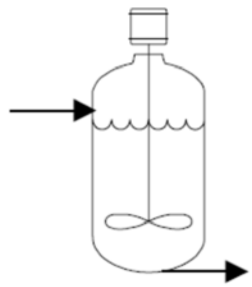


Figure 4: Mixed Tank of Liquid

Referring to the mass balance,

$$F_{in}\rho - F\rho = \frac{d(V\rho)}{dt}$$

For liquid systems mass balance equation normally can be simplified by making the assumption that liquid density is constant. Additionally as *V* = *Ah* thus,

$$F_{in} - F = A\frac{d(h)}{dt}$$

2.5.2 The component balance

To develop a realistic CSTR model, the change of individual species (or components) with respect to time must be considered. This is because individual components can appear / disappear because of reaction (remember that the overall mass of reactants and products will always stay the same). If there are *N* components *N – 1* component balances and an overall mass balance expression are required. Alternatively a component balance may be written for each species. A component balance for the *jth* chemical species is,

Rate of flow of jth component in – rate of flow of jth component out + rate of formation of jth component from chemical reactions = rate of change of jth component

2.5.3 Adding a chemical reaction to the stirred tank model

Assume that the reaction may be described as, *A* → *B*, i.e. component *A* reacts irreversibly to form component *B*. Further, assume that the reaction rate is 1st order. Therefore the rate of reaction with respect to C_A is modeled as,

$$-kC_A = \frac{d(C_A)}{dt}$$

The negative sign implies that C_A disappears because of reaction. The component balance differential equation is

$$F_{in}C_{Ain} - FC_A - kVC_A = \frac{d(VC_A)}{dt} = V\frac{d(C_A)}{dt}$$

2.5.4 The energy balance

Rate of energy flow in – rate of energy flow out + rate at which heat added due to reaction = rate of change of energy within system

The reaction ($A \rightarrow B$) is assumed to be exothermic. A cooling coil is used to remove any heat generated by reaction.

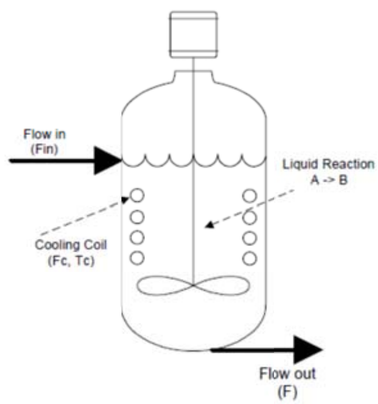


Figure 5: CSTR with cooling coil removing energy (*Q*)

The rate of heat removal by the cooling coil is Q. Fluid specific heat and density are assumed constant.

$$\frac{dE}{dt} = \frac{d}{dt}\left[MC_p(T - T_{datum})\right] = MC_p\frac{d}{dt}[T - T_{datum}] = MC_p\frac{dT}{dt} = V\rho C_p\frac{dT}{dt}$$

The effect of temperature on the reaction rate k is usually found to be exponential,

$$k = k_0 e^{-E/RT}$$

Where k_0 is a pre-exponential (or Arrhenius) factor, E is the activation energy, T is the reaction temperature and R is the gas law constant.

There are other considerations to complete the model of CSTR like, rate of heat transfer through a cooling coil / jacket and dynamics of the reactor wall which have been neglected for this short overview.

In summary, the dynamic model of the CSTR is nonlinear as a result of the many product terms and the exponential temperature dependence of k in above equation. Consequently, it must be solved by numerical integration techniques.

Additional species or chemical reactions may involve. If the reaction mechanism involved production of an intermediate species, $A \rightarrow B \rightarrow C$, then unsteady-state component balances for both A and B would be necessary, or balances for both A and B could be written. Information concerning the reaction mechanisms would also be required.

Reactions involving multiple species are described by high-order, highly coupled, nonlinear reaction models because several component balances must be written. Although the modeling task becomes much more complex, the same principles illustrated above can be extended and applied.

2.6 Case object

Real industrial chemical processes typically contain several reaction steps and multiple recycle streams. Most process synthesis, controllability, and flexibility studies in the literature have considered much simpler systems. In this report we present a simplified version of a real complex industrial process. This example illustrates many important characteristics of such systems like a complex flow sheet, significant interactions among units with recycle streams, and numerous byproduct / intermediate components. We think that this process would be utilized by researchers in the areas of process synthesis and process control as a test case for studying various techniques and approaches to problems in design and control.

Refer to a rich plant modeling, taken from the article “Cooperative distributed model predictive control” by Stewart, Venkat, Rawlings, Wright and Pannocchia (2010); we consider in this thesis report, a plant consisting of two reactors and a separator. A stream of pure reactant *A* is added to each reactor and converted to the product *B* by a first-order reaction as illustrated in Figure 6. The product is lost by a parallel first-order reaction to side product *C*. The distillate of the separator is split and partially redirected to the first reactor.

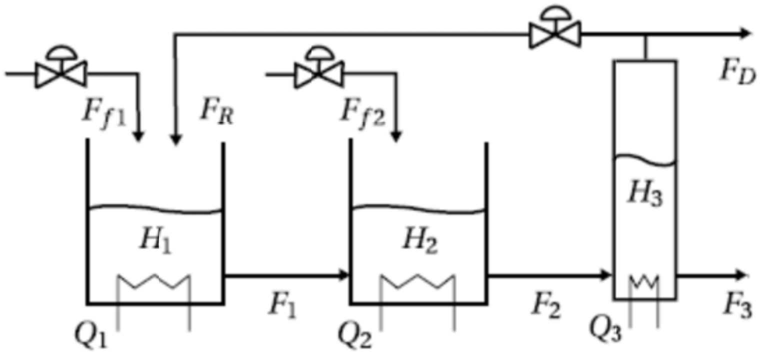


Figure 6: Two reactors in series with separator and recycle

The model for the plant is

$$\frac{dH_1}{dt} = \frac{1}{\rho A_1}(F_{f1} + F_R - F_1)$$

$$\frac{dx_{A1}}{dt} = \frac{1}{\rho A_1 H_1}\left(F_{f1}x_{A0} + F_R x_{AR} - F_1 x_{A1}\right) - k_{A1}x_{A1}$$

$$\frac{dx_{B1}}{dt} = \frac{1}{\rho A_1 H_1}(F_R x_{BR} + F_1 x_{B1}) + k_{A1}x_{A1} - k_{B1}x_{B1}$$

$$\frac{dT_1}{dt} = \frac{1}{\rho A_1 H_1}\left(F_{f1}T_0 + F_R T_R - F_1 T_1\right) - \frac{1}{C_p}(k_{A1}x_{A1}\Delta H_A + k_{B1}x_{B1}\Delta H_B) + \frac{Q_1}{\rho A_1 C_P H_1}$$

$$\frac{dH_2}{dt} = \frac{1}{\rho A_2}(F_{f2} + F_1 - F_2)$$

$$\frac{dx_{A2}}{dt} = \frac{1}{\rho A_2 H_2}\left(F_{f2}x_{A0} + F_1 x_{A1} - F_2 x_{A2}\right) - k_{A2}x_{A2}$$

$$\frac{dx_{B2}}{dt} = \frac{1}{\rho A_2 H_3}(F_1 x_{B1} + F_2 x_{B2}) + k_{A2}x_{A2} - k_{B2}x_{B2}$$

$$\frac{dT_2}{dt} = \frac{1}{\rho A_2 H_2}\left(F_{f2}T_0 + F_1 T_1 - F_2 T_2\right) - \frac{1}{C_P}(k_{A2}x_{A2}\Delta H_A + k_{B2}x_{B2}\Delta H_B) + \frac{Q_2}{\rho A_2 C_P H_2}$$

$$\frac{dH_3}{dt} = \frac{1}{\rho A_3}(F_2 - F_D - F_R - F_3)$$

$$\frac{dx_{A3}}{dt} = \frac{1}{\rho A_3 H_3}(F_2 x_{A2} - (F_D + F_R)x_{AR} - F_3 x_{A3})$$

$$\frac{dx_{B3}}{dt} = \frac{1}{\rho A_3 H_3}(F_2 x_{B2} - (F_D + F_R)x_{BR} - F_3 x_{B3})$$

$$\frac{dT_3}{dt} = \frac{1}{\rho A_3 H_3}(F_2 T_2 - (F_D + F_R)T_R - F_3 T_3) + \frac{Q_3}{\rho A_3 C_P H_3}$$

In which for all $i \in \mathbb{I}_{1:3}$

$$F_i = k_{vi}H_i \qquad k_{Ai} = k_A \exp\left(-\frac{E_A}{RT_i}\right) \qquad k_{Bi} = k_B \exp\left(-\frac{E_B}{RT_i}\right)$$

The recycle flow and weight percents satisfy

$$F_D = 0.01 F_R \qquad x_{AR} = \frac{\alpha_A\, x_{A3}}{\bar{x}_3} \qquad x_{BR} = \frac{\alpha_B\, x_{B3}}{\bar{x}_3}$$

$$\bar{x}_3 = \alpha_A\, x_{A3} + \alpha_B\, x_{B3} + \alpha_C\, x_{C3} \qquad x_{C3} = (1 - x_{A3} - x_{B3})$$

Table 3: Steady state and parameters

Parameter	Value	Units	Parameter	Value	Units
H_1	29.8	m	A_1	3	m^2
x_{A1}	0.542	wt(%)	A_2	3	m^2
x_{B1}	0.393	wt(%)	A_3	1	m^2
T_1	315	K	ρ	0.15	kg/m^3
H_2	30	m	C_P	25	kJ/kg K
x_{A2}	0.503	wt(%)	k_{v1}	2.5	kg/m s
x_{B2}	0.421	wt(%)	k_{v2}	2.5	kg/m s
T_2	315	K	k_{v3}	2.5	kg/m s
H_3	3.27	m	x_{A0}	1	wt(%)
x_{A3}	0.238	wt(%)	T_0	313	K
x_{B3}	0.570	wt(%)	k_A	0.02	1/s
T_3	315	K	k_B	0.018	1/s
F_{f1}	8.33	kg/s	E_A/R	-100	K
Q_1	10	kJ/s	E_B/R	-150	K
F_{f2}	0.5	kg/s	ΔH_A	-40	kJ/kg
Q_2	10	kJ/s	ΔH_B	-50	kJ/kg
F_R	66.2	kg/s	α_A	3.5	
Q_3	10	kJ/s	α_B	1.1	
T_R	318.56	K	α_C	0.5	

Chemical processes have been traditionally operated using linear controllers, although it is well recognized that a characteristic of chemical processes presenting a challenging control problem is the inherent nonlinearity of the process. Linear controllers can yield satisfactory performance, if the process is operated “close” to a nominal steady state or is fairly “linear”. Many times the process dynamic characteristics will change dramatically due to a large disturbance or due to significant setpoint changes from an on-line optimization routine.

Consequently, chemical manufacturing processes present many challenging control problems, including nonlinear dynamic behavior. Other common process characteristics that cause control difficulty for linear and nonlinear systems alike are:

- multivariable interactions between manipulated and controlled variables
- unmeasured state variables
- unmeasured and frequent disturbances
- high-order and distributed processes
- uncertain and time-varying parameters
- constraints on manipulated and state variables
- deadtime on inputs and measurements

3 Model Predictive Control

The only advanced control methodology which has made a significant impact on industrial control engineering is predictive control. It has so far been applied mainly in the petrochemical industry, but is currently being increasingly applied in other sectors of the process industry. The main reasons for its success in these applications are:

1. It handles multivariable control problems naturally.
2. It can take account of actuator limitations.
3. It allows operation closer to constraints (compared with conventional control), which frequently leads to more profitable operation. Remarkably short pay-back periods have been reported.
4. Control update rates are relatively low in these applications, so that there is plenty of time for the necessary on-line computations.

In addition to the “constraint-aware optimizing” variety of predictive control, there is an 'easy-to-tune, intuitive' variety, which puts less emphasis on constraints and optimization, but more emphasis on simplicity and speed of computation, and is particularly suitable for single-input, single-output (SISO) problems. This variety has been applied in high-bandwidth applications such as servomechanisms, as well as to relatively slow processes.

Model predictive control is an appropriately descriptive name for a class of model based control schemes that utilize a process model for two central tasks (i) explicit prediction of future process behavior, and (ii) computation of appropriate corrective control action required to drive the predicted output as close as possible to the desired target values. The overall objectives of an MPC may be summarized as:

- Prevent violations of input and output constraints.
- Drive some output variables to their optimal setpoints, while maintaining other outputs within specified ranges.
- Prevent excessive movement of the input variables.
- Control as many process variables as possible when a sensor or actuator is not available.

The ideas appearing to a greater or lesser degree, in all predictive controls are basically:

- dependence of the control law on predicted behavior,
- explicit use of models to predict the process output at future time instants,
- calculation of control sequence minimizing an objective function, and
- receding horizon strategy, i.e., updating of input and shifting of the horizon towards the future at each time instant.

Predictive control is intuitive and used in our daily activities like walking, driving, studying and so on. Think about the course of studying in a school. Basically one has to do a set of things:

• Predict: When one sets a target for a "desired" grade, one has to plan and work towards the target. It may be too early to consider the final target at beginning of a term. Instead, one should think a few days or a few weeks ahead and predict what performance may be achieved over this shorter time window. The target within the shorter time period can be, for example, certain "desired" grades in the assignment, quiz, etc.

• Plan: Compare the predicted performance with the shorter time target. If a difference is to be expected, for example, lower than the target, then additional efforts should be considered, subject to constraints of course, such as there are only 24 hours a day.

• Act: If it is expected that the additional efforts likely make one meet the target, then the additional efforts will be put into action. Although a set of the additional efforts, for today, tomorrow, and so on, has been planned days or weeks ahead, only the effort planned for today can actually be materialized today. In the next day, the procedure of prediction and planning is repeated, and a new set of efforts is determined. Thus the next day's action will be taken according to the new planning. This process proceeds continuously until the end of the term.

Other well-known daily-life analogies in MPC literature include crossing a road and playing chess. In chess, a good player predicts the game a few steps ahead based on the moves of the opponent, and plans a few future moves. However, only one move can actually be applied each time. Based on the following-up move of the opponent, a new set of predictions has to be made and a new set of future moves is determined as a result. This procedure is repeated throughout the game.

3.1 Historical issues on MPC in process control

When MPC was first advocated by Richalet, Rault, Testud and Papon (1976) for process control, several proposals for MPC had already been made, such as Lee and Markus (1967), and, even earlier, a proposal, by Propoi (1963), of a form of MPC, using linear programming, for linear systems with hard constraints on control. However, the early proponents of MPC for process control proceeded independently, addressing the needs and concerns of industry. Existing techniques for control design, such as linear quadratic control, were not widely used, perhaps because they were regarded as addressing inadequately the problems raised by constraints, nonlinearities and uncertainty. The applications envisaged were mainly in the petro-chemical and process industries, where economic considerations required operating points (determined by solving linear programs) situated on the boundary of the set of operating points satisfying all constraints. The dynamic controller therefore has to cope adequately with constraints that would otherwise be transgressed even with small disturbances. The plants were modeled in the early literature by step or impulse responses. These were easily understood by users and facilitated casting the optimal control and identification problems in a form suitable for existing software.

Thus, IDCOM (identification and command), the form of MPC proposed in Richalet et al. (1976,1978), employs a finite horizon pulse response (linear) model, a quadratic cost function, and input and output constraints. The model permits linear estimation, using least squares. The algorithm for solving the open-loop optimal control problem is a “dual” of the estimation algorithm. As in dynamic matrix control (DMC; Cutler & Ramaker, 1980; Prett & Gillette, 1980), which employs a step response model but is, in other respects, similar, the treatment of control and output constraints is ad hoc. This limitation was overcome in the second-generation program, quadratic dynamic matrix control (QDMC; Garcia & Morshedi, 1986) where quadratic programming is employed to solve exactly the constrained open-loop optimal control problem that results when the system is linear, the cost quadratic, and the control and state constraints are defined by linear inequalities. QDMC also permits, if required, temporary violation of some output constraints, effectively enlarging the set of states that can be satisfactorily controlled. The third generation of MPC technology, introduced about a decade ago, distinguishes between several levels of constraints (hard, soft, ranked), provides some mechanism to recover from an infeasible solution, addresses the issues resulting from a control structure that changes in real time, and allows for a wider range of process dynamics and controller specifications (Qin & Badgwell, 1997). In particular, the Shell multivariable optimizing control (SMOC) algorithm allows for state-space models, general disturbance models and state estimation via Kalman filtering (Marquis & Broustail, 1988). The history of the three generations of MPC technology, and the subsequent evolution of commercial MPC, is well described in the last reference. The substantial impact that this technology has had on industry is confirmed by the number of applications (probably exceeding 2000) that make it a multi-million dollar industry.

The industrial proponents of MPC did not address stability theoretically, but were obviously aware of its importance; their versions of MPC are not automatically stabilizing. However, by restricting attention to stable plants, and choosing a horizon large compared with the “settling” time of the plant, stability properties associated with an infinite horizon are achieved. Academic research, stimulated by the unparalleled success of MPC, commenced a theoretical investigation of stability. Because Lyapunov techniques were not employed initially, stability had to be addressed within the restrictive framework of linear analysis, confining attention to model predictive control of linear unconstrained systems. The original finite horizon formulation of the optimal control problem (without any modification to ensure stability) was employed. Researchers therefore studied the effect of control and cost horizons and cost parameters on stability when the system is linear, the cost is quadratic, and hard constraints are absent. A typical result establishes the existence of finite control and cost horizons such that the resultant model predictive controller is stabilizing.

3.2 Open loop optimal control problem

The system to be controlled is usually described, or approximated, by an ordinary differential equation, but since the control is normally piecewise constant, is usually modeled, in the MPC literature, by a difference equation.

Consider the system

$$x(k+1) = f(x(k), u(k)), \qquad x(k_0) = x_0$$

where $x(t) \in \mathbb{R}^n$and $u(t) \in \mathbb{R}^m$ are the state and input vector, respectively. We assume that the origin is an equilibrium point (*f(0, 0) = 0*).

We presumed the system to be specified in discrete time. One reason is that we are looking for solutions to engineering problems. In practice, the controller will always be implemented through a digital computer by sampling the variables of the system and transmitting the control action to the system at discrete time points. Another reason is that for the solution of the optimal control problems for discrete-time systems, we will be able to make ready use of powerful mathematical programming software. However, in many instances the discrete time model is an approximation of the continuous time model.

According to the optimal control theory, the problem is to minimize the performance index *J* (performance objective or cost function).

$$J(x(k_0), u(\cdot), k_0) = \sum_{i=k_0}^{\bar{k}-1} l(x(i), u(i)) + m(x(\bar{k}))$$

with respect to the future control sequence of input *u* from the time k_0 to $\bar{k} - 1$.

$$u(k_0), u(k_0 + 1), \ldots, u(\bar{k} - 1)$$

Generally the solution of the optimal control problem can be founded by solving the Hamilton-Jacobi-Bellman equation (HJB) as follow,

$$J^0(x, k) = \min_u\{l(x, u) + J^0(f(x, u), k + 1\}$$

with the boundary condition $J^0(x, \bar{k}) = m(x)$.

The plant model generally can be linearized around the operation point to yield the linearized plant formulation. By considering a linear system

$$x(k + 1) = Ax(k) + Bu(k)$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$ and are calculated as

$$A = (\frac{\partial f}{\partial x})_{x=\bar{x}, u=\bar{u}} \quad , \quad B = (\frac{\partial f}{\partial u})_{x=\bar{x}, u=\bar{u}}$$

such that $(x = \bar{x}, u = \bar{u})$ denotes the operating points.

Before introducing the quadratic cost function, it is restored to express some definitions. Expressions like $x'Qx$ and $u'Ru$, where *x*, *u* are vectors, *Q* and *R* are symmetric matrices and $x' = x^T$ indicate the transpose of vector *x*, are called quadratic forms, and are often written as $\|x\|_Q^2$ and $\|u\|_R^2$ respectively. They are just compact representations of certain quadratic functions in several variables.

Considering a linear system and a quadratic cost function

$$l(x,u) = x'Qx + u'Ru, \quad Q \geq 0, \qquad R > 0$$

$$m(x) = x'Sx, \quad S \geq 0$$

where *Q* and *S* are state weighting or penalty and *R* is input weighting.

The problem is turned into Linear Quadratic Control (LQ), where the HJB approach reduced to $J^0(x,k) = x'P(k)x$ with $P(\bar{k}) = S$.

The solution is Finite Horizon (FH) problem and completely defined by the control law

$$u(k) = -\underbrace{(R + (B'P(k+1)B)')^{-1}B'P(k+1)'A}_{K(k)}\, x(k)$$

where $P(k) = Q + A'P(k+1)A - A'P(k+1)B(R + B'P(k+1)B)^{-1}B'P(k+1)A$ with its boundary condition $P(\bar{k}) = S$.

Meant for continuous processes which are operating over a long time period, it would be interesting to solve the infinite horizon problem.

If we consider the Infinite Horizon (IH) cost function

$$J = \sum_{k=0}^{\infty} x'(k)Qx(k) + u'(k)Ru(k)\,, \quad Q \geq 0, \qquad R > 0$$

with the assumption of reachability of pair (*A,B*) and observablity of pair (*A,C*), where *Q*=*C'C*, then the optimal control law is $u(k) = -\bar{K}x(k)$ With

$$\bar{K} = (R + B'\bar{P}B)^{-1}B'\bar{P}A$$

such that $\bar{P}$ is the unique positive definite solution of the Riccati equation, which equal to

$$\bar{P} = A'\bar{P}A + Q - A'\,\bar{P}B\,(R + B'\bar{P}B)^{-1}B'\bar{P}A\ .$$

In order to consider the disturbances or unmeasurable states, Kalman predictor (KP) can be applied, where there are some equivalency of parameters in the formulation of LQ and KP. Moreover Linear Quadratic Gaussian (LQG) control can be considered where in the stochastic system, the disturbances and the initial state satisfy the assumption introduced for the KP, and the state is not measurable.

3.3 Closed-loop _ open-loop analysis

Through considering the system as $x(k+1) = Ax(k) + Bu(k) \qquad x \in \mathbb{R}^n, u \in \mathbb{R}^m$

and modifying the performance index as

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

Where $Q = Q' \geq 0,\ R = R' > 0,\ S = S' \geq 0$ and outlining *N* as the so-called prediction horizon, the stated problem is formed to minimizing the above performance index *J*. Here we divide the general problem in three categories, and try to formulate the problem with some consideration on the stability:

1- IH-LQ:
2- FH optimal control
3- Receding Horizon (RH)

3.3.1 IH-LQ

According to the previous result by the infinite horizon cost function and with the assumption of reachability and observability, the optimal control law is

$$u^0(k) = -Kx(k)$$

where *K* is calculated as $K = (R + B'PB)^{-1}B'PA$

such that is obtained by finding the *P* as the unique positive definite solution of the algebraic Riccati equation (ARE).

$$P = A'PA + Q - A'\,PB\,(R + B'PB)^{-1}B'PA$$

Confined by these conditions for noted control law, the closed-loop system is asymptotically stable.

3.3.2 FH optimal control

The optimal solution is given by the state-feedback control law

$$u^0(k+i) = -K(i)x(k+i), \quad i = 0,1,\dots,N-1$$

Where *K(i)* is $K(i) = (R + B'P(i+1)B)^{-1}B'P(i+1)A$

which is obtained by finding the *P(i)* as the solution of the difference Riccati equation (DRE)

$$P(i) = A'P(i+1)A + Q - A'\,P(i+1)B\,(R + B'P(i+1)B)^{-1}B'P(i+1)A$$

with initial condition *P(N)=S.*

In order to find the open loop solution of FH optimal control, we recall the Lagrange equation

$$x(k+i) = A^i x(k) + \sum_{j=0}^{i-1} A^{i-j-1} B u(k+j), \;\; j > 0$$

And define the related matrices $X(k), U(k), \mathcal{A}, \mathcal{B}$.

$$X(k) = \begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+N-1) \\ x(k+N) \end{bmatrix} \qquad \mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N-1} \\ A^N \end{bmatrix} \qquad U(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-2) \\ u(k+N-1) \end{bmatrix}$$

$$\mathcal{B} = \begin{bmatrix} B & 0 & 0 & \dots & 0 & 0 \\ AB & B & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ A^{N-2}B & A^{N-3}B & A^{N-4}B & \dots & B & 0 \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \dots & AB & B \end{bmatrix}$$

where *0* is zero matrix.

Thus, the future state variables are given by following formula:

$$X(k) = \mathcal{A}x(k) + \mathcal{B}U(k)$$

Moreover by defining augmented weighting matrices $\mathcal{Q}, \mathcal{R}$ with the following construction,

$$\mathcal{Q} = \begin{bmatrix} Q & 0 & \dots & 0 & 0 \\ 0 & Q & & 0 & 0 \\ \vdots & & \ddots & \vdots & \\ 0 & 0 & \dots & Q & 0 \\ 0 & 0 & & 0 & S \end{bmatrix} \;\;, \; \mathcal{R} = \begin{bmatrix} R & 0 & \dots & 0 & 0 \\ 0 & R & & 0 & 0 \\ \vdots & & \ddots & \vdots & \\ 0 & 0 & \dots & R & 0 \\ 0 & 0 & & 0 & R \end{bmatrix}$$

and minimizing the modified performance index $\bar{J}$ with respect to *U(k),* the open loop solution is as follow:

$$u^0(k+i) = -\mathcal{K}(i)x(k), \quad i = 0,1,\dots,N-1$$

where $\mathcal{K}(i)$ depends on matrices $\mathcal{A}, \mathcal{B}, \mathcal{Q}\ and\ \mathcal{R}$ and is calculated as follows.

The new modified performance index is

$$\bar{J}(x(k), u(\cdot), k) = X'(k)\mathcal{Q}\,X(k) + U'(k)\mathcal{R}U(k)$$

where with respect to the original cost function, the terms $x'(k)Qx(k)$ has been ignored, since it does not depend on *U(k).* By minimizing this new performance index which can be rewrite in the form of quadratic function of *U(k)*, its minimum turns out to be $U^o(k) = -(\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})^{-1}\mathcal{B}'\mathcal{Q}\mathcal{A}x(k)$.

Thus, by letting $\mathcal{K}(i) = (\mathcal{B}'\mathcal{Q}\mathcal{B} + \mathcal{R})^{-1}\mathcal{B}'\mathcal{Q}\mathcal{A}$, $\mathcal{K}(i)$ is obtainable.

Note1: in the nominal case, the closed-loop and the open-loop solutions coincide.

Note2: if there are constraints on the control and/or state variables, the closed-loop solution is not available, while the open-loop one can be reformulated as a mathematical programming problem and can be easily solved by means of a QP (quadratic programming) method with reduced computational time.

3.3.3 RH problem

In all above cases open- or closed-loop solutions are time varying control laws, which are defined over a finite horizon. In order to obtain a time-invariant control law, the Receding Horizon has been defined in the way that at any time *k*, solve the optimization problem over the prediction horizon *[k,k+N]* and apply only the first input $u^o(k)$ of the optimal sequence $U^o(k)$. At time *k+1* repeat the optimization over the prediction horizon *[k+1,k+N+1]* again as shown in Figure 7.

Thus, intuitively, instead of making the horizon infinite we can get a similar behavior when we use a long, but finite horizon *N*, and repeat this optimization at each time step, in effect moving the horizon forward (moving horizon or receding horizon control).

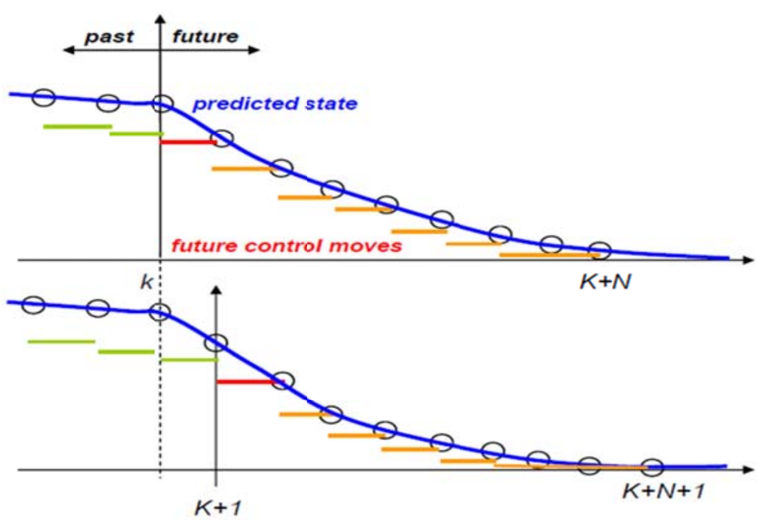


Figure 7: Receding horizon scheme

The RH principle allows one to obtain the state-feedback time-invariant control law.

$$u = \kappa_{RH}(x)$$

Intended for constrained systems, this control law is implicitly defined, while in the unconstrained case, it coincides with the first element of the open-loop solution and the first element of the closed-loop solution obtained by iterating the Riccati equation backwards from *P(N)=S.*

First element of the open-loop solution is:

$$u^0(k) = -\mathcal{K}(0)x(k)$$

First element of the closed-loop solution is:

$$u^0(k) = -K(0)x(k)\ , \quad K(0) = (R + B'P(1)B)^{-1}B'P(1)A$$

Thus the receding horizon solution formed as

$$u^0(k) = -\mathcal{K}(0)x(k) = -K(0)x(k)$$

Noted that, it is not a-priori guaranteed that the RH control law stabilizes the closed-loop. In some cases, stability may be achieved only with the large prediction horizon.

From this point upward, the formulation of the MPC can be modified according to the applications, nevertheless the principle of receding horizon endure intrinsic in all of these formulations. As a matter of fact that the subject of this report is setpoint tracking or in general terms, regulation problem to control some parameters within operation region in the chemical processes, the effect of introducing reference signals and disturbances with their considerations, are going to be discussed here after.

3.4 MPC formulation without integral action

Consider the system with disturbances

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

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m, y(k) \in \mathbb{R}^p, d(k) \in \mathbb{R}^d$ and

$A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$.

The new cost function which is penalizing the tracking error with respect to the reference signal y^o is

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

Again, we define the new matrices $Y(k), Y^0(k), D(k), \mathcal{A}_c, \mathcal{B}_c, \mathcal{M}_c$ with following constructions,

$$Y^0(k) = \begin{bmatrix} y^0(k+1) \\ y^0(k+2) \\ \vdots \\ y^0(k+N-1) \\ y^0(k+N) \end{bmatrix} , \; Y(k) = \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N-1) \\ y(k+N) \end{bmatrix} , \; \mathcal{A}_c = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N-1} \\ CA^N \end{bmatrix}$$

$$\mathcal{B}_c = \begin{bmatrix} CB & 0 & 0 & \dots & 0 & 0 \\ CAB & CB & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ CA^{N-2}B & CA^{N-3}B & CA^{N-4}B & \dots & CB & 0 \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \dots & CAB & CB \end{bmatrix}, \ D(k) = \begin{bmatrix} d(k) \\ d(k+1) \\ \vdots \\ d(k+N-1) \\ d(k+N) \end{bmatrix}$$

$$\mathcal{M}_c = \begin{bmatrix} CM & I & 0 & \dots & 0 & 0 & 0 \\ CAM & CM & I & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ CA^{N-2}M & CA^{N-3}M & CA^{N-4}M & \dots & CM & I & 0 \\ CA^{N-1}M & CA^{N-2}M & CA^{N-3}M & \dots & CAM & CM & I \end{bmatrix}$$

where *0* and *I* are zero and identity matrices.

Note here that matrices $\mathcal{A}_c, \mathcal{B}_c, \mathcal{M}_c$ are time independent and they can be computed offline.

Then, the future outputs (output predictions) are formed as

$$Y(k) = \mathcal{A}_c x(k) + \mathcal{B}_c U(k) + \mathcal{M}_c D(k)$$

By defining the future output, the problem is equivalent to minimize the modified cost function

$$\bar{J}(x(k), u(\cdot), k) = \left(Y^{\circ}(k) - Y(k)\right)' \mathcal{Q} \left(Y^{\circ}(k) - Y(k)\right) + U'(k)\mathcal{R}U(k)$$

In the unconstrained case, the optimal solution is

$$U^o(k) = (\mathcal{B}'_c\mathcal{Q}\mathcal{B}_c + \mathcal{R})^{-1}\mathcal{B}'_c\mathcal{Q}(Y^\circ(k) - \mathcal{A}_c x(k) - \mathcal{M}_c D(k))$$

which depends on the future reference signals $Y^o(k)$ and on the future disturbances *D(k).* The optimal solution arise the motivation to state that, the model predictive control can "anticipate" future reference variations or the effect of known disturbances.

Note1: there is not any integral action which has been forced in the feedback control law, therefore with assumption of providing the closed-loop stability, for constant reference signal, steady state zero error regulation cannot be achieved.

Note2: In all above considered cases, the state *x(k)* has been assumed to be measurable. Otherwise an observer can be utilized. Likewise to estimate the disturbance *d(k)* when it is unmeasurable.

Note3: When the future disturbance is unknown, it is a common practice to set $d(k+i) = d(k)\,, \quad i > 0$.

In the case of control regulation with the constant reference signals y^0, by assuming that there exists a pair $(\bar{x}, \bar{u})$ such that

$$\begin{cases} \bar{x} = A\bar{x} + B\bar{u} \\ y^0 = C\bar{x} \end{cases}$$

a more significant performance index has been fashioned as follow:

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

This performance index penalizes the control deviation with respect to the desired equilibrium point.

Note 4: this performance index does not penalize the state, subsequently in this case proper observability or detectability assumption is advisable.

3.5 MPC formulation with integral action

As the aim of setpoint tracking strategy is to driving the system to steady state zero error regulation for constant reference signal, it is also a common practice in algorithms based on impulse or step response models, to plug an integral action at the inputs like Figure 8-1.

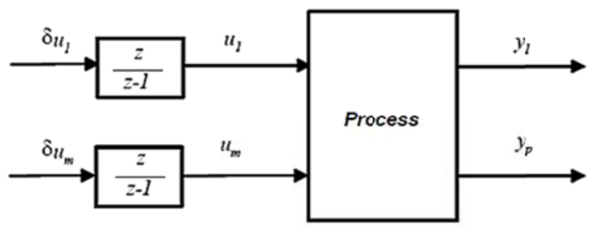


Figure 8-1: Plugging the integral action

As usual, the plant model expresses the plant state *x* in the terms of the values of input *u*. but the before mentioned cost function, penalizes changes in *δu*, rather than the input variable themselves. We shall see in the chapter 5 that the MPC algorithm will in fact produce the changes *δu*, rather than *u*. it is therefore convenient for many purposes to regard the controller as producing the signal *δu*, and the plant as having this signal as input. That is, it is often convenient to regard the discrete time integration from *δu* to *u* as being included in the plant dynamic.

There are several techniques to including this integration in a state-space model of the MPC plant and all of them involve augmenting the state vector. The first way, regarding the Figure 8-1, is in which the integrators can be formulated as:

$$\begin{cases} v(k+1) = v(k) + \delta u(k) \\ u(k) = v(k) + \delta u(k) \end{cases}$$

So the state-space form of the system plus integrators and by neglecting the disturbances is obtained:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \delta u(k)$$

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix}$$

While the Performance index with tracking error and control variation is

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

In unconstrained case, the RH control law is linear as follow

$$\delta u(k) = \mathcal{K}_y Y^o(k) - \mathcal{K}_x x(k) - \mathcal{K}_v v(k)$$

But in the block diagram view of this system, the integrator disappears due to the feedback term on signal *v*, as illustrated in Figure 8-2.

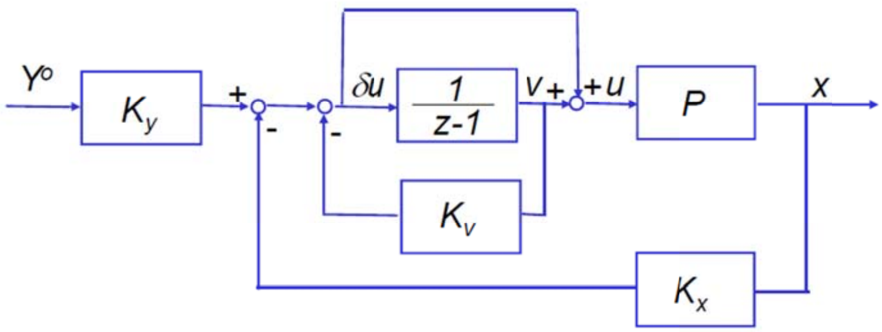


Figure 8-2: disappearing the integrator in augmented state block diagram

This problem can be avoided with an alternative formulation of MPC and exploiting again the enlarged system:

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

$$x(k+1) - x(k) = A(x(k) - x(k-1) + B(u(k) - u(k-1) \rightarrow \delta x(k+1)$$
$$= A\delta x(k) + B\delta u(k)$$

$$y(k+1) = Cx(k+1) \rightarrow y(k+1) - y^0 = Cx(k+1) - y^0$$
$$= CAx(k) + CBu(k) - y^0 - y(k) + y(k)$$

$$e(k+1) = y(k+1) - y^0$$

$$\begin{cases} \delta x(k+1) = A\delta x(k) + B\delta u(k) \\ e(k+1) = CA\delta x(k) + CB\delta u(k) + e(k) \end{cases}$$

$$\begin{bmatrix} \delta x(k+1) \\ e(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 \\ CA & I \end{bmatrix} \begin{bmatrix} \delta x(k) \\ e(k) \end{bmatrix} + \begin{bmatrix} B \\ CB \end{bmatrix} \delta u(k)$$

There is a hidden integral action in this model (indicated by I in enlarged state matrix).

Finally in unconstrained case, the control law takes the form

$$\delta u(k) = K_e e(k) - K_x \delta x(k)$$

which clearly has an integral action on the error signal *e(k),* and in steady-state conditions where *δx=0*, so that *δu=0* only for the condition of *e=0* as illustrated in Figure 9.

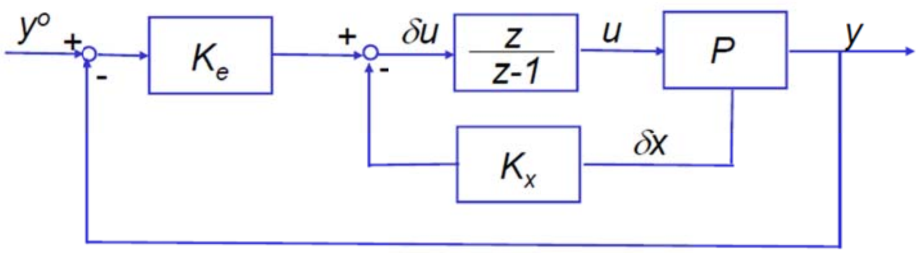


Figure 9: Integral action in augmented state block diagram

3.6 Extension to the basic formulation

The main MPC algorithms are characterized by a number of “tricks” which make them very different from a classical LQ algorithm. Some of these tricks are Control horizon, Minimum prediction horizon, Reference filtering, Filtering of disturbances and High level optimization. Among them, we just take a short view on control horizon which will be used in the generation of MPC algorithm of chemical plants.

If the prediction horizon *N* is sufficiently large, the number of optimization variables (or the future control increments) can make the optimization problem difficult to solve. For this reason, and to obtain a slower control action, it is often assumed that the control variables remain constant after $N_u<N$ time instants.

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

or

$$\delta u(k+i) = 0, \quad i = N_u, \dots, N-1$$

In this case, the cost function can be written as follow

$$J(x(k), u(\cdot), k) = \sum_{i=0}^{N-1} \|y^0(k+i) - y(k+i)\|_Q^2 + \sum_{i=0}^{N_u-1} \|\delta u(k+i)\|_R^2 + \|y^0(k+N) - y(k+N)\|_S^2$$

which is going to be used widely, by the advantages of easier optimization problem.

4 Stability

Early versions of MPC and generalized predictive control did not automatically ensure stability, thus requiring tuning. It is therefore not surprising that research in the 1990s devoted considerable attention to this topic. Indeed, concern for stability has been a major engine for generating different formulations of MPC. In time, differences between model predictive, generalized predictive, and receding horizon control became irrelevant; we therefore use MPC as a generic title in the consequence for that mode of control in which the current control action is determined by solving on-line an optimal control problem.

4.1 Stability analysis

Model predictive control of constrained systems is nonlinear necessitating the use of Lyapunov stability theory, that the value function (of a finite horizon optimal control problem) could be used as Lyapunov function to establish stability of receding horizon control of unconstrained systems when a terminal equality constraint is employed. These results extended that the value function as a Lyapunov function for establishing stability of model predictive control of time-varying, *constrained*, *nonlinear*, discrete-time systems (when a terminal equality constraint is employed); thereafter, the value function was almost universally employed as a natural Lyapunov function for stability analysis of model predictive control.

4.1.1 Definitions

While asymptotic convergence of the state *x(k)* in the form of $\lim_{k\to\infty} x(k) \to 0$ is a desirable property, it is generally not sufficient in practice. We would also like a system to stay in a small neighborhood of the origin when it is disturbed by a little. Formally this is expressed as Lyapunov stability.

Consider the system

$$x(k+1) = f\big(x(k)\big), \qquad x(0) = x_0$$

Where f is an arbitrary (discontinuous) function and $\bar{x}$ is equilibrium point if $f(\bar{x}) = \bar{x}$.

Letting $X^o \subseteq R^n$ an open neighborhood of $\bar{x}$, then $\bar{x}$ is stable, unstable, attractive, asymptotically stable and exponentially stable according to the following conditions:

Stable, if, for each $\varepsilon > 0$, there is $\delta > \delta(\varepsilon)$ such that

$$\|x_0 - \bar{x}\| \leq \delta \ \rightarrow \ \|x(k) - \bar{x}\| \leq \epsilon \text{ for any } k \geq 0.$$

Unstable, if not stable.

Attractive in X^o if

$$\lim_{k\to\infty}\|x(k)-\bar{x}\| = 0 \text{ for any } x_0 \in X^o.$$

Asymptotically stable in X^o , if it is stable and attractive in X^o.

Exponentially stable in X^o , if there exist $\theta \geq 0,\ \lambda \in (0,1)$ such that

$$\|x(k)-\bar{x}\| \leq \theta\|x_0-\bar{x}\|\, \lambda^k \text{ for any } k \geq 0.$$

The ε, δ requirement for stability definition takes a challenge-answer form. To demonstrate that the origin is stable, for any value of ε that a challenger may chose (however small), we must produce a value of δ such that a trajectory starting in a δ neighborhood of the origin will never leave the ε neighborhood of the origin.

Function $\varphi: \mathbb{R}_+ \to \mathbb{R}$ is a *K* function if it is continuous, strictly increasing with $\varphi(0) = 0$.

Usually to show Lyapunov stability of the origin for a particular system, one constructs a so called Lyapunov function, i.e., a function satisfying the conditions of the following theorem.

Let $X^o \subseteq R^n$be a positively invariant set for the system

$$x(k+1) = f\big(x(k)\big)$$

containing a neighborhood $\mathcal{N}$ of the equilibrium $\bar{x} = 0$.

Undertake that ω, ψ, r be the class K functions and assume that there exist a nonnegative scalar function $V: X^o \to \mathbb{R}_+,\ V(0) = 0$ such that

$$V(x) \geq \omega(\|x\|), \quad \forall x \in X^o$$

$$V(x) \geq \psi(\|x\|), \quad \forall x \in \mathcal{N}$$

$$\Delta V(x) \leq -r(\|x\|), \quad \forall x \in X^o$$

Then the origin is an asymptotically stable equilibrium in X^o. Moreover, if $\omega(\|x\|) \coloneqq a\|x\|^\sigma$, $\psi(\|x\|) \coloneqq b\|x\|^\sigma$, $r(\|x\|) \coloneqq c\|x\|^\sigma$ for some $a, b, c, \sigma > 0$ and $\mathcal{N} = X^o$, then the origin is exponentially stable in X^o.

We use the extension of the Lyapunov theory by considering non continuous Lyapunov functions (the cost function in constrained MPC control) and refer to Figure10.

4.1.2 RH and IH-LQ control

Considering linear system with measurable state

$$x(k+1) = Ax(k) + Bu(k)$$

and its performance index

$$J\big(x(k), u(\cdot)\big) = \sum_{i=0}^{N-1}\big(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2\big) + \|x(k+N)\|_S^2 .$$

According to RH and IH-LQ control in unconstrained case, some terminologies are outlining in the following.

Terminal cost: a cost penalizing non zero state at the end of prediction horizon with the choice $S = \bar{P}$ which is $\|x(k+N)\|_{\bar{P}}^2$.

Main control law: which is the RH solution $u^0(k) = -K(0)x(k)$.

Auxiliary control law: which is assumed to be used from the end of the prediction horizon onwards. $u(k) = -K_{LQ}x(k)$

By considering these assumptions, the resulting J is the classical IH-LQ cost function and with a suitable choice of the terminal cost, the RH control law guarantees stability in the unconstrained case.

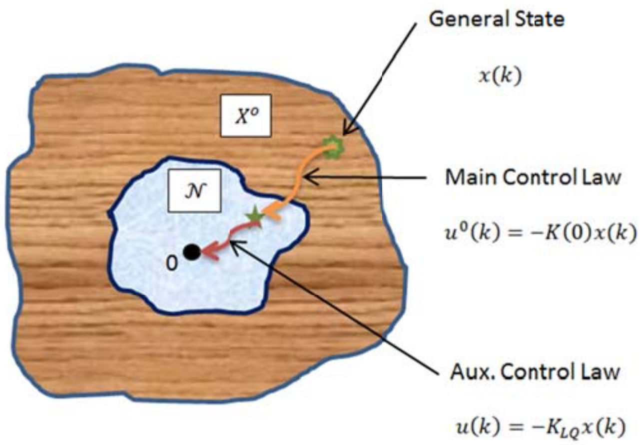


Figure 10: stabilized sets in the RH and IH-LQ unconstrained control

4.2 Stabilizing modifications

There are many methods for modifying the open-loop optimal control problem ($\mathcal{P}_N$) employed in model predictive control of constrained systems so that closed-loop stability could be guaranteed.

The modifications correspond, mainly for the terminal cost and the terminal constraint set.

4.2.1 Constrained IH-LQ control

Considering the following IH constrained problem

$$\mathcal{P}_\infty = \min J_\infty\big(x(k), u(\cdot)\big) = \sum_{i=0}^{\infty}\big(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2\ \big)$$

$$x(k+i) \in X \quad , \;\; i \geq 0$$

$$u(k+i) \in U \quad , \;\; i \geq 0$$

where *U* and *X* are closed sets containing the origin, *Q>0, R>0*.

The solution of this problem cannot be computed with the HJB equation or with the open-loop solution in view of the infinite number of constraints to be considered.

However the solution of the stated optimization problem can be found by solving, with a sufficiently long prediction horizon *N* and with the RH strategy. Thus the FH optimal control problem is

$$\mathcal{P}_N = \min J_N\big(x(k), u(\cdot)\big) = \sum_{i=0}^{N-1}\big(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2\,\big) + \ \|x(k+N)\|_{\bar{P}}^2$$

$$x(k+i) \in X \quad , \quad i = 0, \dots, N-1$$

$$u(k+i) \in U \quad , \quad i = 0, \dots, N-1$$

It must be assumed that *x(k)* belongs to the positively invariant admissible set for $\mathcal{P}_\infty$, that is the set of states which can be satisfied by fulfilling the state and control constraints.

$$\bar{X} = \{x(k) | \exists u(\cdot) \in U : x(k+i) \in X, i \geq 0, and\ J_\infty^o < \infty\,\}$$

Define now the positively invariant admissible set $\bar{X}_{LQ}$ associated to the IH-LQ control law $u(k) = -K_{LQ}x(k)$,

$$x(k) \in \bar{X}_{LQ} \ \rightarrow \begin{cases} u(k+i) = -K_{LQ}x(k+i) \in U, & i \geq 0 \\ x(k+i) = (A - BK_{LQ})^i x(k) \in \bar{X}_{LQ}, & i \geq 0 \end{cases}$$

In order to compute $\bar{\mathrm{X}}_{\mathrm{LQ}}$, first note that $x'\bar{P}x - c = 0,\ c > 0$ is a level line of the Lyapunov function $V(x) = x'\bar{P}x$ for the closed-loop system with the IH-LQ control law. Therefore, in the unconstrained case,

$$X_c = \{x | x'\bar{P}x \leq c\}$$

is a positively invariant set for the closed-loop system with IH-LQ control. Now, if there is a prospect to find a set

$$\Gamma = \{x \middle| x \in X \; and \; u = -K_{LQ}x \in U\}$$

for any *c* such that $X_c \subseteq \Gamma$, it grades that X_c is the required set $\overline{\mathrm{X}}_{\mathrm{LQ}}$, which is illustrated in Figure 11.

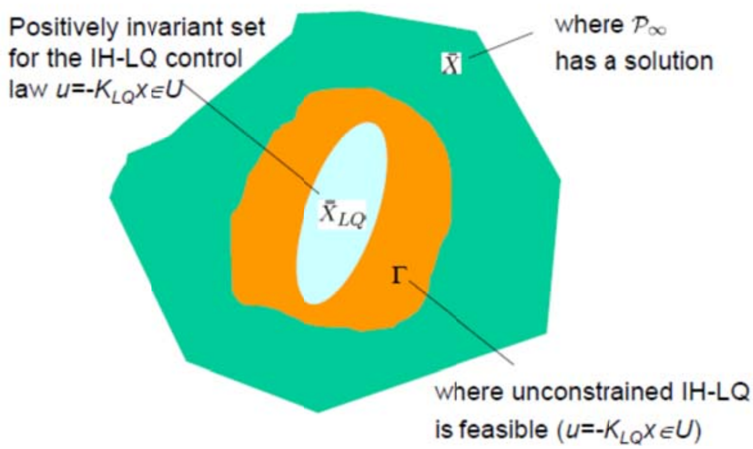


Figure 11: Stabilized sets in the constrained IH-LQ control

Finally it can be proven that by letting $x(k) \in \bar{X}$ and by given a set $\overline{\mathrm{X}}_{\mathrm{LQ}}$, there exists a (computable) sufficiently long prediction horizon *N* such that the solution of the associated problem $\mathcal{P}_N$ is such that

$$x(k+N) \in \bar{X}_{LQ} \; .$$

The computation of an upper bound of N can be performed with results available in the literature. Assuming that this value has been determined, in view of the dynamic programming approach, the solution of $\mathcal{P}_\infty$ coincides with the solution of $\mathcal{P}_N$. In fact, the terminal cost of $\mathcal{P}_N$ is the cost to go of the IH problem.

The discussion above reveals the presence of several ingredients that have been found useful in developing stabilizing model predictive controllers; these ingredients are a terminal cost, a terminal constraint set, and a local controller. There are more conditions in the relevant literature on these ingredients, which if satisfied, ensure that the model predictive controller is stabilizing.

4.3 Achievements on MPC stability

Research on stability of model predictive controlled systems has now reached a relatively mature stage. The important factors for stability have been isolated and employed to develop a range of model predictive controllers that are stabilizing and differ only in their choice of the three ingredients (i) terminal cost, (ii) terminal constraint set and (iii) local controller, that are common to most forms of model predictive control. These conditions are merely sufficient and several researchers are seeking relaxations.

5 Simulation in MATLAB and Simulink

5.1 Model of the system

According to the plant which consist of two reactors and a separator and their models which are described in Chapter 2, the plant has been divided in three subsystems and consequently three MATLAB functions have been constructed by utilizing differential equation in mfiles, relatively. These files are available in Appendix A.

5.2 Linearization

The model of the system has been described regarding the dynamic previously introduced, in differential equations. As it hold the nonlinear properties, first it must be converted to the linear model (Linearization continues/discrete).

How to compute the linearized model? There are many ways to linearize a nonlinear model and mostly work around the equilibrium point. One of the simplest ways in the environment of Simulink is by using the block “Timed-Based Linearization” which allows to numerically computing at the given time instant the linearized model of an overall nonlinear system. This block calls “linmod” or “dlinmod” to create a linear model for the system when the simulation clock reaches the time specified by the Linearization time parameter. No trimming is performed. The linear model is stored in the base workspace as a structure, along with information about the operating point at which the snapshot was taken. Multiple snapshots are appended to form an array of structures.

In order to use "Timed-Based Linearization" block, the open loop system must be built, define the inputs and define the additional input port, which correspond to the real input of linearized system. The same way to define the output of linearized model with output ports must be applied. Then the system must be executed for constant values. There have to be sure that at the time that Simulink wants to compute the linearized model, the system must be in steady state. Simulink automatically computes the linearized model, by giving small variation to each input and looking at the result on each output (numerically linearization technique). This approach is critical sometimes and need some precautions since for the unstable system, it can be very dangerous. However for stable system, there is a prior guarantee that after the linearization time, system remains in steady state position, and the result is reliable.

There are ways to simulate the model, like S-function and etc., but there is a simple way to use MATLAB functions in Simulink with a simple trick. This trick is schemed in Figure 12.

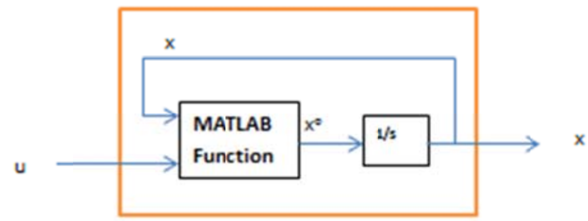


Figure 12: Simulate the model with MATLAB function

As the MATLAB function has been formed by the equation $x^o = f(x,u)$, it must be noted that when utilizing this method, the first argument is named '*u*' which is a pack of all the parameters (*u, x,* ...).

The result for the first subsystem (first reactor) is represented in Figure 13. Emphasize that the last output which is the flow to the next reactor, has a linear formulation.

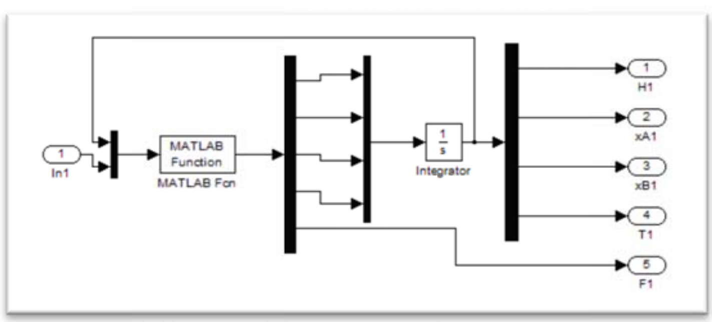


Figure 13: the Model of fist reactor

Regarding the hierarchical design in this project for better understanding and visibility structure, intermediate scheme as in Figure 14, has been created.

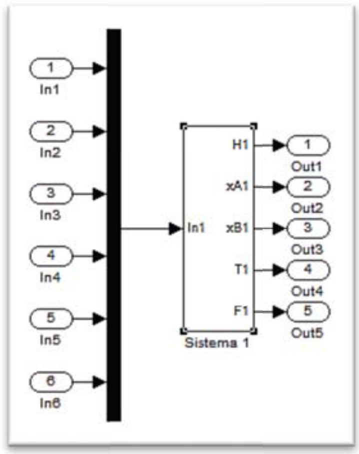


Figure 14: intermediate scheme for first reactor model

All equilibrium points and constant variables which are used in the models are available in Appendix A, which is defined according to the Table 3. These equilibrium points are aimed in linearization process, to obtain linear model in vicinity of these equilibrium points as a starting point. We can use other values for initialization, but since the model is nonlinear, there is no guarantee to reach the same steady state positions with different initial values.

Subsequently we run the open-loop system in Simulink with the simulation time greater than the linearization time. By looking at the output plots, we can see that all the states or outputs are in steady state condition after linearization time. By looking at the workspace, we can find (by whos) the structure with the name of open-loop Simulink model as sis1_Timed_Based_Linearization, with the number of variables and calculated matrices *A, B, C* and *D* of the linearized model. This method is surprisingly practical. By these matrices, the linearized model can be prepared. Therefore once we yield the linearized model, by means of off-line calculated matrices of the model, they can be exploited for controller design.

5.3 Required matrix construction

In order to use MPC formulation in MATLAB and in general framework, we need to construct some matrices. With the reference to the Chapter 3, about the formulation of MPC, the constructions of these matrices are coming in the following.

As the result of linearization procedure is obtained in the continuous mode, first we must convert the system matrices from continuous to discrete-time models with specified sampling time in order to obtain (*A, B, C*) matrices in discrete mode. The effect of disturbances can be introduced within construction of matrix *M*. by making this *M* matrix equal to zero, can simply neglect the disturbances. In next step, by using "DLQR" command in MATLAB, which is the Linear-quadratic (LQ) state-feedback regulator for discrete-time state-space system, the optimal gain matrix *K* can be calculated, such that the state-feedback law minimizes the quadratic cost function. In addition to the state-feedback gain *K*, DLQR returns the infinite horizon solution *S* of the associated discrete-time Riccati Equation and the closed-loop eigenvalues.

Subsequently, by defining the weighting matrices, prediction horizon, control horizon and finally constraint on inputs (manipulated variables) and outputs/states, relative matrices have been formed. To insure the stability, terminal constraint has been formed finally and been enforced to zero.

There is also one specific algorithm for reduced size of matrices in the case that the control horizon is less than prediction horizon, which can reduce the computations.

5.4 Optimization by Quadprog

The main advantages of this MPC algorithm is gained by using the "Quadprog" command in MATLAB for optimization which is the utmost significant part of each MPC algorithm. Quadratic programming is the problem of finding a vector that minimizes a quadratic function, possibly subject to linear constraints. These constraints can be equality constraints or inequality ones.

Basically in Quadprog we need to have the method for computing the performance index, subject to some constraints.

As the Quadprog in MATLAB has its own formulation, the required construction must be applied, as follow:

$$\min_x \frac{1}{2} x'Hx + f'x \text{ such that } \begin{cases} A.x \leq b \\ A_{eq}.x = b_{eq} \\ lb \leq x \leq ub \end{cases}$$

where the syntax is : $x = Quadprog(H, f, A, b, Aeq, beq)$

5.4.1 Optimization for controller without Integral action

Regarding the case of controller without integral action and referring to the introduced formula in Chapter 3, the following performance index is subject to some modification as per the project requirements.

$$\mathcal{P}_N = \min J_N = \sum_{i=0}^{N-1} \left(\|y^0 - y(k+i)\|_Q^2 \right) + \sum_{i=0}^{N_u-1} (\|u(k+i) - \bar{u}\|_R^2)$$

$$+ \ \|y^0 - y(k+N)\|_S^2$$

$$J_N = \left(y^0 - y(k)\right)' \mathcal{A'}_c \mathcal{Q} \mathcal{A}_c \left(y^0 - y(k)\right) + \left(y^0 - y(k)\right)' \underbrace{2\mathcal{A'}_c \mathcal{Q} \mathcal{B}_c}_{F} \Delta U(k)$$

$$+ \frac{1}{2} \Delta U'(k) \underbrace{2(\mathcal{B'}_c \mathcal{Q} \mathcal{B}_c + \mathcal{R})}_{H} \Delta U(k)$$

where the first term is not depend on U(k), so can be neglected in minimization, thus

$$\mathcal{P}_N = \min\{ \frac{1}{2} \Delta U'(k) H \ \Delta U(k) + \left(y^\circ - y(k)\right)' F' \ \Delta U(k) \}$$

where , $F = 2\mathcal{A'}_c \ \mathcal{Q} \ B_c$ and $H = 2(B'_c \ \mathcal{Q} \ B_c + \ \mathcal{R})$.

Note that in this case the term $\Delta U(k)$ is the input deviation from the equilibrium point.

$$\Delta u(k) = u(k) - u^0$$

5.4.2 Optimization for controller with Integral action

Refer to the pre-defined formula for the MPC with control action, and construction of matrices *F* and *H*, the performance index is developed.

$$\mathcal{P}_N = \min J_N = \sum_{i=0}^{N-1} \left(\|y^0 - y(k+i)\|_Q^2 \right) + \sum_{i=0}^{N_u-1} (\|\delta u(k+i)\|_R^2) + \|y^0 - y(k+N)\|_S^2$$

$$\mathcal{P}_N = \min\{ \frac{1}{2} \delta U'(k) H \, \delta U(k) + \left(y^\circ - y(k)\right)' F' \delta U(k)\}$$

Note that in this case the term $\delta U(k)$ is the input increment.

$$\delta u(k) = u(k) - u(k-1)$$

5.5 Constraints

MPC controller is nothing, but applying the first vector of Quadratic programming optimization as a control variable to the plant. By contributing different constraint for inputs, outputs and adding terminal constraint to this approach, different results may arise due to the theory of MPC control which has been discussed in Chapters 3 and 4.

Nearly every application imposes constraints; actuators are naturally limited in the force (or equivalent) they can apply, safety limits states such as temperature, pressure and velocity and efficiency often dictates steady-state operation close to the boundary of the set of permissible states. The prevalence of hard constraints is accompanied by a lack of control methods for handling them, despite a continuous demand from industry that has had, in their absence, to resort often to ad hoc methods

The formulations for constraint cases are divided in two portions. Inequality constraints are related to the saturation or high/low limits in input, or may some technological limits or safety issues on output or state. Equality constraint is cause of some guarantee for stability which may lead to computing terminal state or terminal constraint, all according to the subject of optimization in quadratic programming method. The general form of inequality constraint is:

$$GU(k) \leq W + Ex(k)$$

In controller without integral action, it turns to

$$G\Delta U(k) \le W + E\big(y^{\circ} - y(k)\big),\ \ u_{min} \le u \le u_{max}, y_{min} \le y \le y_{max}$$

and with integral action case, it becomes

$$G\delta U(k) \le W + E\big(y^{\circ} - y(k)\big),\ \ u_{min} \le u \le u_{max}, y_{min} \le y \le y_{max}\ .$$

To construct the inequality constraint in general form, we have

$$u_{min} \le u \le u_{max} \rightarrow \begin{cases} I.U \le U_{max} \\ -I.U \le -U_{min} \end{cases}$$

$$y_{min} \le y \le y_{max}\ ,(Y(k) = \mathcal{A}_c x(k) + \mathcal{B}_c U(k)) \rightarrow \begin{cases} \mathcal{B}_c.U \le Y_{max} + \mathcal{A}_c.x \\ -\mathcal{B}_c.U \le -Y_{min} - \mathcal{A}_c.x \end{cases}$$

which can be mixed together for the cases of input and output (state) constraint.

$$\begin{bmatrix} I \\ -I \\ \mathcal{B}_c \\ -\mathcal{B}_c \end{bmatrix}.U \le \begin{bmatrix} U_{max} \\ -U_{min} \\ Y_{max} \\ -Y_{min} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\mathcal{A}_c \\ \mathcal{A}_c \end{bmatrix}.x$$

Designed for equality constraint in general form, the result by forcing terminal constraint to zero is as follow

$$y(k+N) = 0\ in\ the\ last\ row\ of\ (Y(k) = \mathcal{A}_c x(k) + \mathcal{B}_c U(k))$$

$$0 = CA^N\, x(k) + [CA^{N-1}B \quad CA^{N-2}B \quad \dots \quad CAB \quad CB]U(k)$$

$$[CA^{N-1}B \quad CA^{N-2}B \quad \dots \quad CAB \quad CB]U(k) = CA^N\, x(k)$$

5.6 Controller design

5.6.1 Controller without integral action

According to the model of the plant consisting of two reactors and a separator, the output and input are denoted, respectively:

$$y = [H_1 \quad x_{A1} \quad x_{B1} \quad T_1 \quad H_2 \quad x_{A2} \quad x_{B2} \quad T_2 \quad H_3 \quad x_{A3} \quad x_{B3} \quad T_3]$$

$$u = \left[F_{f1} \quad Q_1 \quad F_{f2} \quad Q_2 \quad F_R \quad Q_3 \right]$$

Due to the centralized strategy of the controller and the aim of the regulation, and regarding the general formula for the MPC in Chapter 3, the following control diagram is achieved.

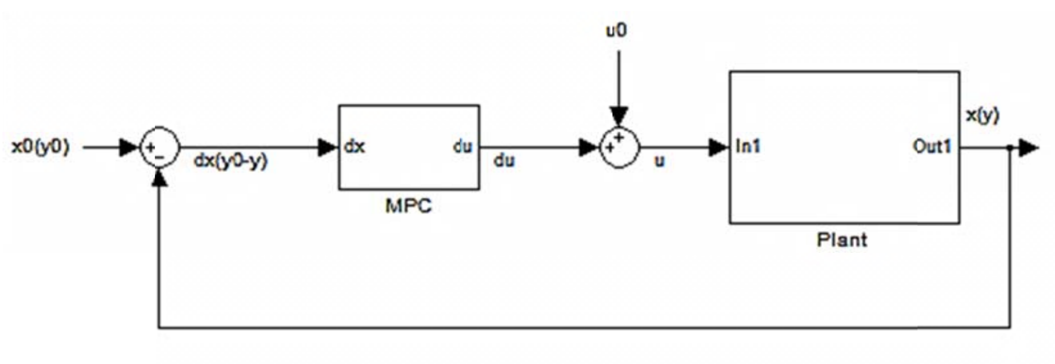


Figure 15: Control diagram of MPC without integral action

The MPC algorithm has been developed according to the discrete-time formulation in state-space, in this way we have discrete-time controller and continues time simulations.

In this controller, the states and outputs are equivalent. Due to the construction of MPC controller, inputs of the MPC are the error on the states or outputs, and its outputs are deviation of control variables from their steady state values. Therefore we need to accumulate them by their steady state values to derive the manipulated variables to the process. Controlled values as output or state return by feedback to calculate the error from the setpoint. More specification of this controller will be discussed on the Chapter 6 by some simulation results. Figure 16 illustrates the implementation of this controller in Simulink.

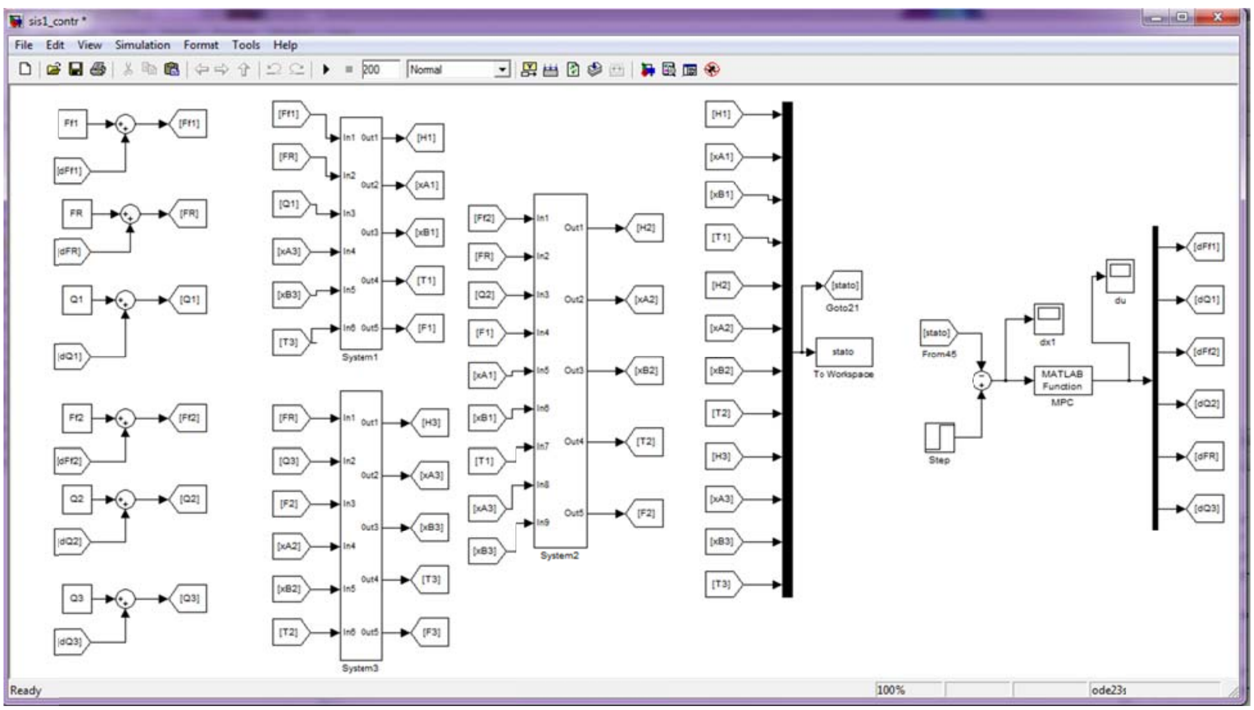


Figure 16: MPC controller without integral action in Simulink

As it has been presented by the control scheme, there is always a necessity of steady state values in the control loop, while if the equilibrium point of the system changes due to the operation region of the process in the time, these values must be modified in the structure, which make a disadvantages for this controller.

Another drawback of this controller is that since there is no integral action in the controller structure, zero steady state could not be guaranteed and may have a steady state error on the outputs. This problem motivates us to modify the controller in order to include the integral action, which is the subject of the following.

5.6.2 Controller with integral action

According to the formulation of MPC with integral action in Chapter 3, the technique promote by using the enlarged system. This matter changed the construction of the MPC controller with integral action with respect to its formulation. The general scheme of this controller represented in Fig. 17.

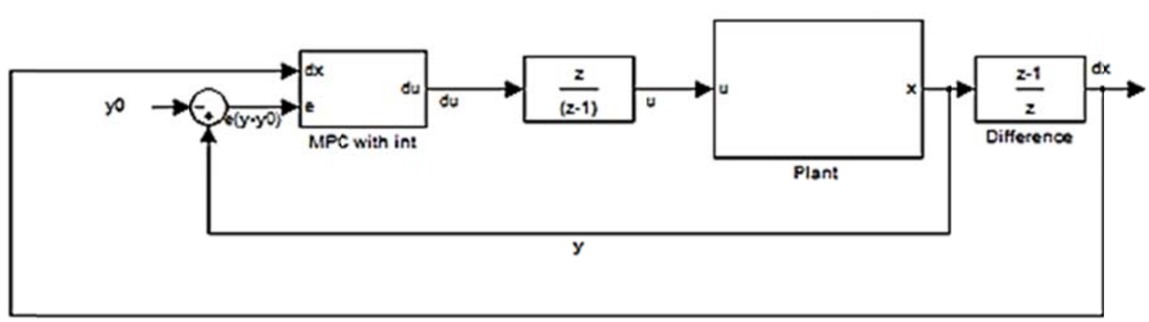


Figure 17: Control diagram of MPC with integral action

The formulation employs the increment of control variable and states. The inputs of this controller consist of deviation of states which come from the filtered feedback, and the error on the output of the system. More detail specification about this controller will be discussed on Chapter 6 by some simulation results. Figure 18 illustrates the implementation of this controller in Simulink.

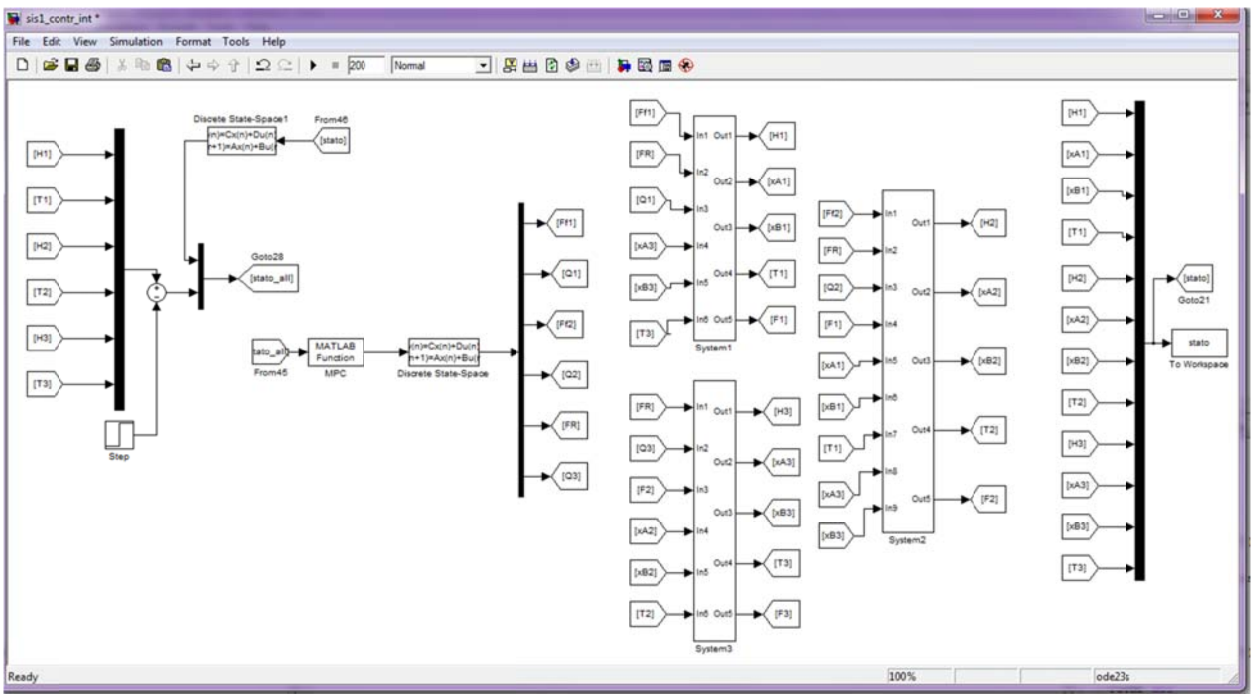


Figure 18: MPC controller with integral action in Simulink

According to the modifications with respect to the controller without integral action, there are some adjustments in the construction of related matrices for integral action, where is indicated in Appendix A.

6 Experimental results

In this thesis report, two methods for implementing the MPC controller are discussed and related formula and considerations have been disposed. The first approach is the controller without integral action and the second one is by concerning the integral action. In order to assess the control performance and stability issue of the model predictive controller structured in the previous chapters, we perform the simulations and try to conclude with an overview of the different controllers with enforcing assortments of parameters, compared here. To recognize the advantages and disadvantages of each method, the plots of control and state variables are depicted in this chapter.

As the problem arises in the multivariable space, the stream of graphs is settled to illustrate the contribution of changed parameter in the entire system, or in some case by associating two selected variables. In point of fact, in the methodology of centralized controller, there is not any independent loop in the system and all the loops are coupled. For this intention the construction of any decoupling and modifying the problem to the different SISO cases like decentralized control approach is intolerable. Contained by this tactic, better recognition of a plant as a unique system has been achieved.

6.1 Results for the MPC controller without integral action

To commencement the analysis of the plant, the constrained MPC algorithm without any Integral action is presumed. Resembling a general control tuning analysis method, we apply the step to a setpoint and illustrate the result in that output trajectory and the effect of this parameter changing, in the other system parameters as well.

6.1.1 Simulation 1: tuning the level of first reactor

The leading term in the first simulation is the Level of reactor1 (*H1*), where the step of 5 units in its setpoint makes the following Figures 19, 20, 21. The weighting rates for all inputs are equal to *0.00001*, and for outputs is one. The prediction horizon and control horizon are both equal to *10*. Table 4 indicates the essential parameters in this simulation.

Table 4: Simulation1 parameters

Parameter	value
Sampling time (Tc) [s]	0.25
Prediction Horizon (N)	10
Control Horizon (Nu)	10
Input Weighting rate(R)	All equal to 0.0001
Output Weighting rate(Q)	All equal to 1
Input Min. Constraint ($Umin$)	All equal to -*inf.*
Input Max. Constraint ($Umax$)	All equal to *inf.*
Output Min. Constraint ($Ymin$)	All equal to -*inf.*
Output Max. Constraint ($Ymax$)	All equal to *inf.*

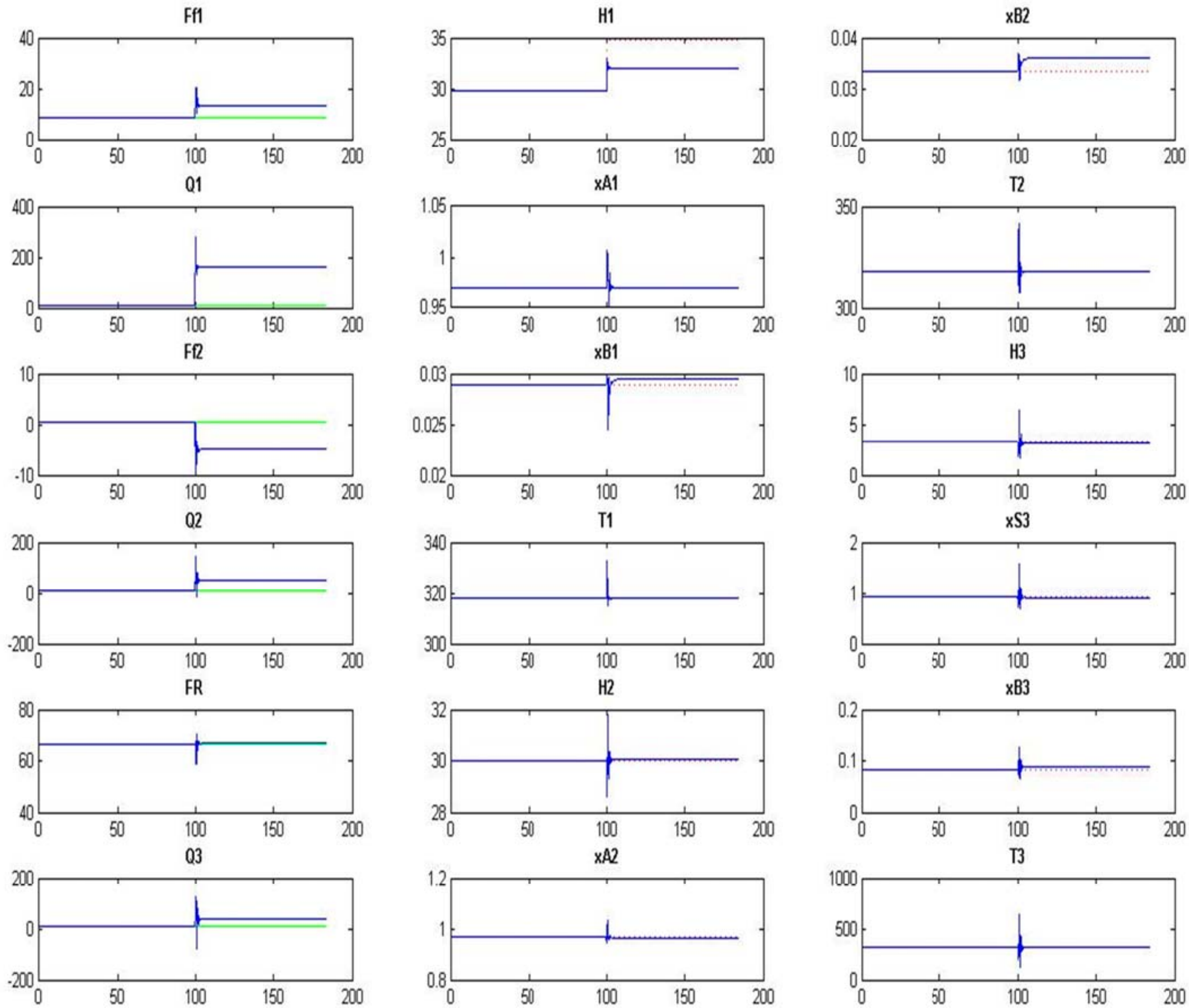
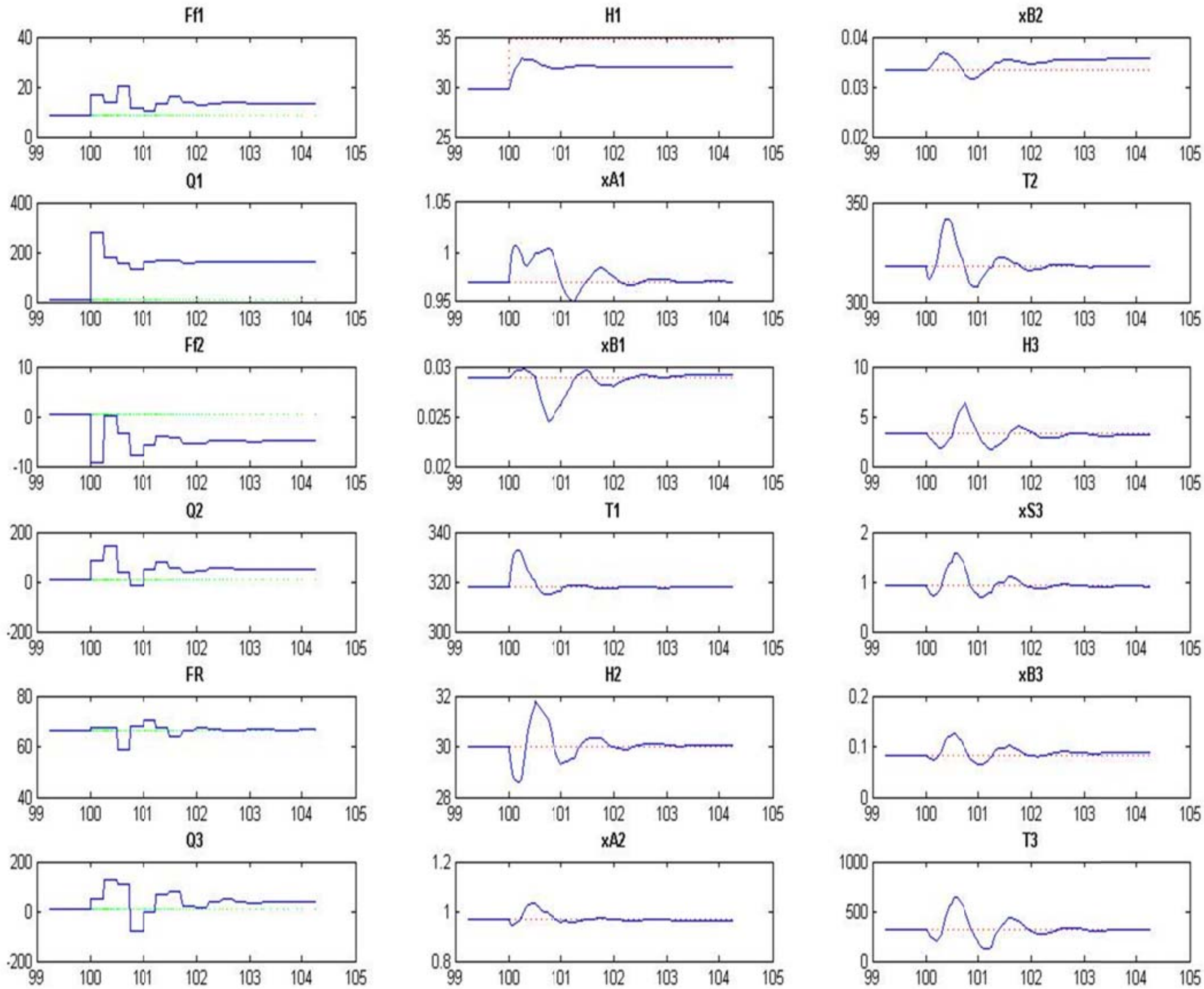


Figure 19 illustrate the tuning on *H1* in the entire plant variables. Since the step is applied in the time *100*, the zoomed plot has been shown in Figure 20.



According to this figure, the change of the setpoint in the level of first reactor, make the sudden fluctuations in all the control variables (*Ff1,Q1,Ff2,Q2,FR,Q3*) according to the weighting of that variable in the cost function. The realization of centralized controller in the frame of model predictive control is revealed here.

Since the physical system possess some dynamics which lead to a propagation delay from the first reactor to the next one and then separator, in the classical control, related inputs for further reactor and separator must be introduced with a delay, while the substance of predictive model alter all the variations in the whole system as it is anticipated. Output tracking is correspondingly under the investigation, while the related weighting of the output in the cost function must be respected. As it has been noted in chapter 3, this type of controller cannot make the steady state error equal to zero, and therefore some steady state error in the outputs are observable, however entire outputs are prosper to follow respected setpoints. The perturbation of output *H1* by make of the change to its setpoint is illustrated in Figure 21. The output attempts to regulate and reach to the steady state condition within 4 seconds, with no overshoot, no unexpected oscillation and just a steady state error.

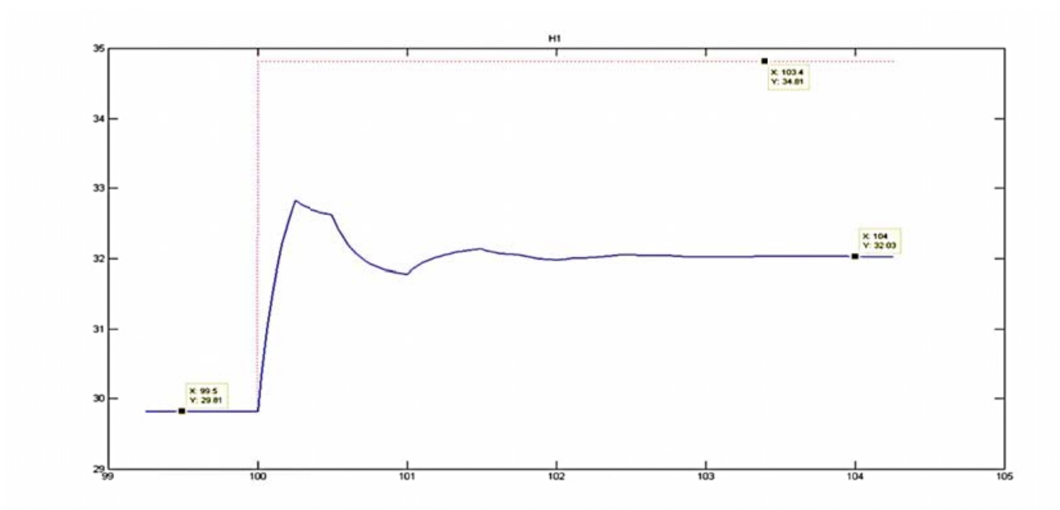


Figure 21: Response of *H1* to the setpont change

In the following, and to get some idea about weighting and horizons, it specifically focused on the two inputs (*Ff1* and *Ff2*) and two output (*H1* and *H2*). To have a proper view, the normalized values with respect to the equilibrium values are going to be used. From the above conditions, we have:

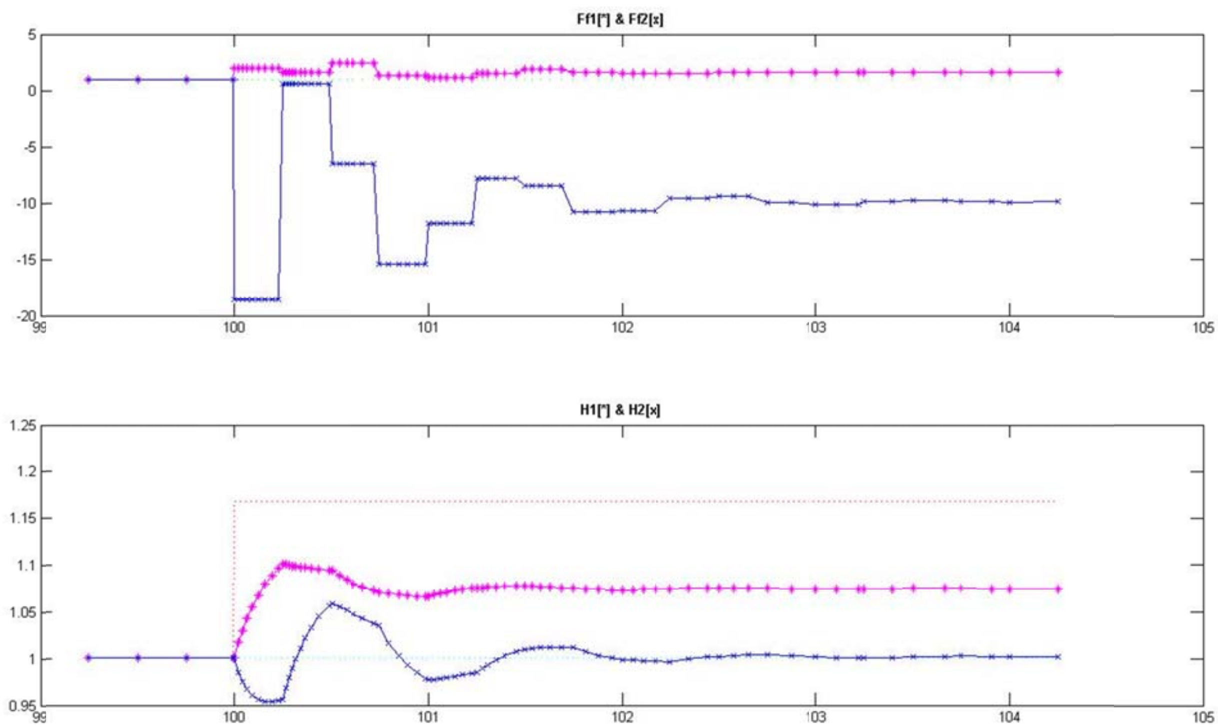


Figure 22: Result of *H1* setpoint change in *Ff1, Ff2, H1* and *H2* (normalized)

6.1.2 Simulation 2: input weighting effect

Through increasing the weighting rate of the first input among the others by the value of *1.1*, and with respect to the parameters in Table 5, the following result has been achieved in Figure 23.

Table 5: Simulation2 parameters

Parameter	value
Sampling time (Tc) [s]	0.25
Prediction Horizon (N)	10
Control Horizon (Nu)	10
Input Weighting rate(R)	*R(Ff1)* equal to 1.1 The rest equal to 0.0001
Output Weighting rate(Q)	All equal to 1
Input Min. Constraint ($Umin$)	All equal to -*inf.*
Input Max. Constraint ($Umax$)	All equal to *inf.*
Output Min. Constraint ($Ymin$)	All equal to -*inf.*
Output Max. Constraint ($Ymax$)	All equal to *inf.*

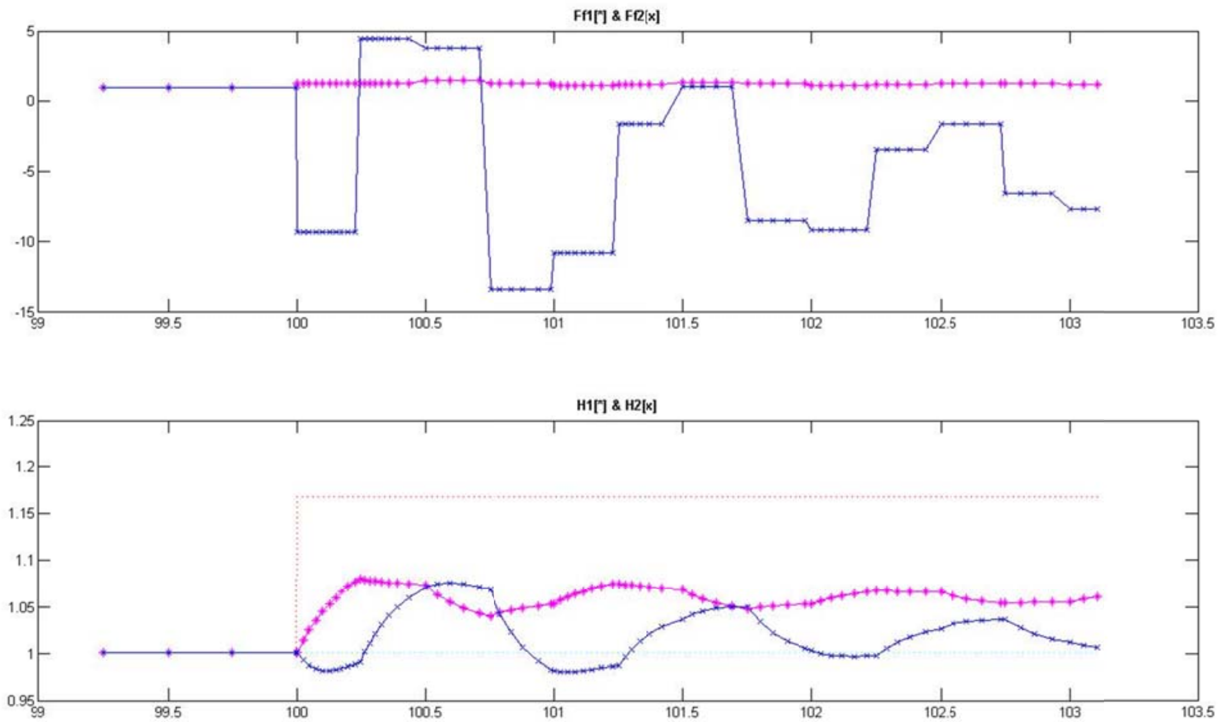


Figure 23: Result of input weighting with *H1* setpoint change in *Ff1, Ff2, H1* and *H2* (normalized)

The smoother variation of Input variable *Ff1* and quicker settling in its equilibrium value is indicated, while the output *H1* experience more variation and oscillation, by consequence of imposing more weighting rate on the input *Ff1*. Due to the emphasizing on this special manipulated variable, other variable settling times are increased, to support the flatter change in *Ff1*.

6.1.3 Simulation 3: output weighting effect

To following the trend, the weighting rate of controlled variable *H1* has been changed and increased to *10* and the result with respect to Table 6 is available in Figure 24.

Table 6: Simulation3 parameters

Parameter	value
Sampling time (Tc) [s]	0.25
Prediction Horizon (N)	10
Control Horizon (Nu)	10
Input Weighting rate(R)	All equal to 0.0001
Output Weighting rate(Q)	*Q(H1)* equal to 10 The rest equal to 1
Input Min. Constraint (*Umin*)	All equal to -*inf.*
Input Max. Constraint (*Umax*)	All equal to *inf.*
Output Min. Constraint (*Ymin*)	All equal to -*inf.*
Output Max. Constraint (*Ymax*)	All equal to *inf.*

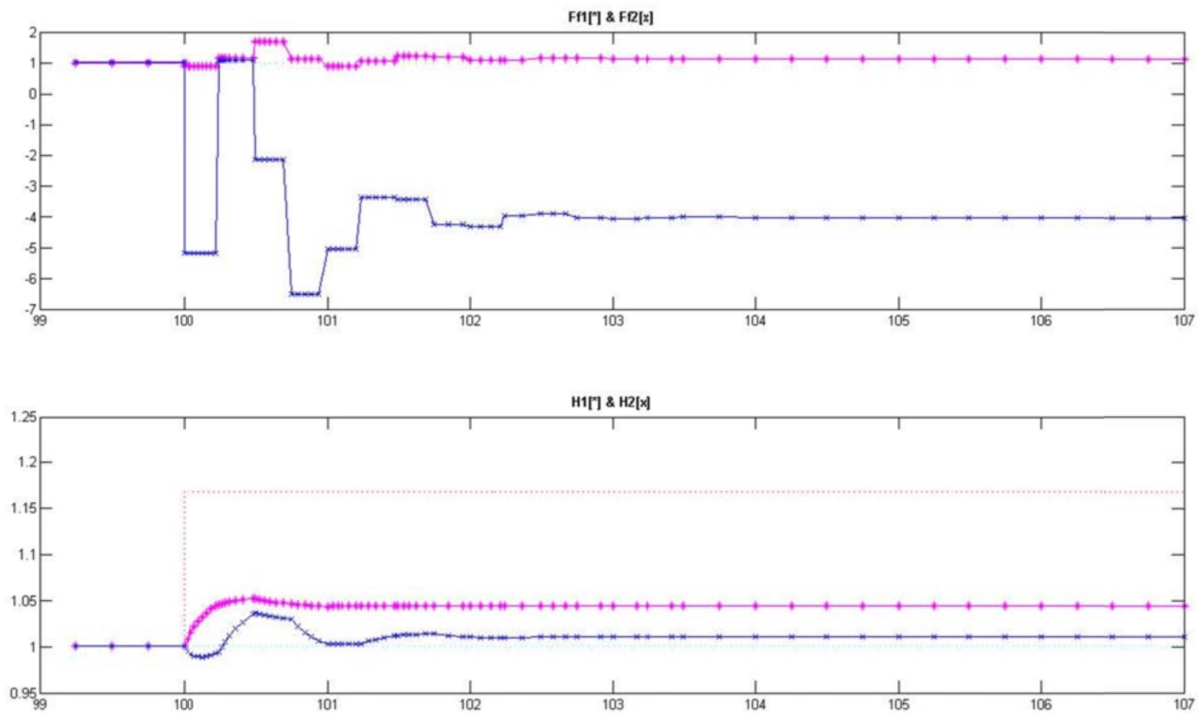


Figure 24: Result of output weighting with *H1* setpoint change in *Ff1, Ff2, H1* and *H2* (normalized)

The controlled output *H1* with higher weighting rate, settled in steady state very quickly, while there is some steady state error which is due to the unavailability of integral action in the MPC controller.

6.1.4 Simulation 4: prediction and control horizons

Another parameter which is quite important in the field of predictive controllers is prediction horizon and control horizon. By laying *N=6* and *Nu=1* and respected parameter in Table 7, the resulted plot is in Figure 25.

Table 7: Simulation4 parameters

Parameter	value
Sampling time (Tc) [s]	0.25
Prediction Horizon (N)	6
Control Horizon (Nu)	1
Input Weighting rate(R)	All equal to 0.0001
Output Weighting rate(Q)	All equal to 1
Input Min. Constraint ($Umin$)	All equal to -*inf.*
Input Max. Constraint ($Umax$)	All equal to *inf.*
Output Min. Constraint ($Ymin$)	All equal to -*inf.*
Output Max. Constraint ($Ymax$)	All equal to *inf.*

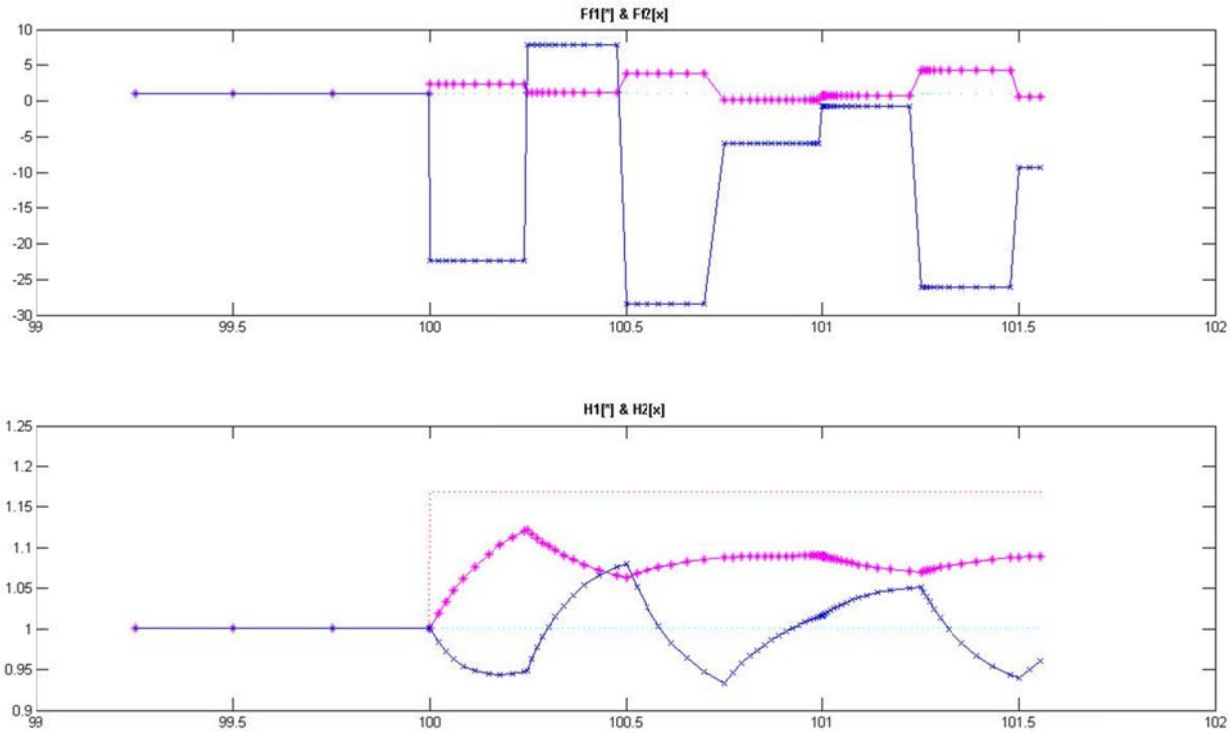


Figure 25: Result of horizons with *H1* setpoint change in *Ff1, Ff2, H1* and *H2* (normalized)

In this case, due to the slighter horizons of prediction, the settling time increased dramatically and variables experience more oscillation, which is quite predictable. However, less optimization effort requires for computation with reduced horizons.

6.1.5 Simulation 5: disturbance effect

To experience some disturbances and evaluate the response of the plant with the MPC controller, an initial perturbation has been forced to an input variable which can be presumed as an initial disturbance or the actuator disturbance. For this determination, drive the manipulated variable *Ff1* with a step of 4 units after time *100* and squared the system responses in Figure 26. All the parameters are referred to Table 4.

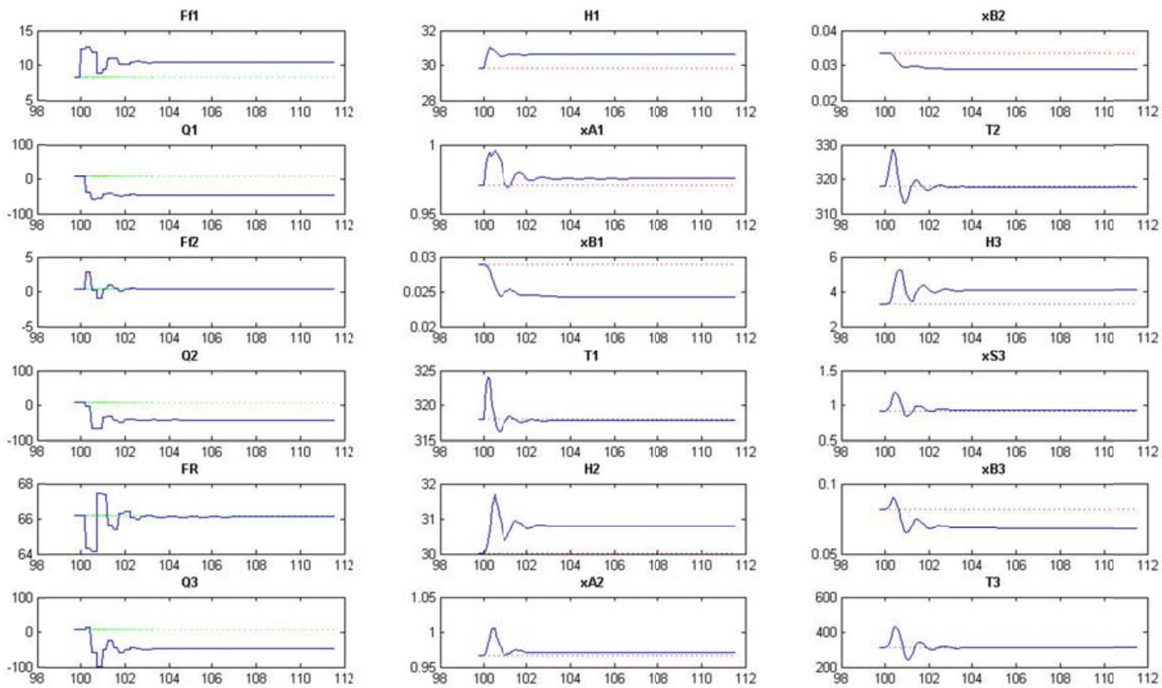


Figure 26: Result of disturbance in the system

According to Figure 26, the system is started from its steady state condition and triggered by step disturbance in the manipulated variable, where the system can reject the disturbance, however since there is no integral action in the construction of the MPC controller, the steady state errors have been observed in some controlled variables.

6.1.6 Simulation 6: input constraint

Since one of the advantages of MPC algorithm is its ability to conclude the limitation on process inputs or outputs/states, here are some experimental results by forcing the constraint to the system.

We start from the first manipulated variable *Ff1* and set the constraint terms maximum to *15* and minimum to zero. Other parameters are available in Table 8. The following is the result in Figure 27.

Table 8: Simulation6 parameters

Parameter	value
Sampling time (Tc) [s]	0.25
Prediction Horizon (N)	10
Control Horizon (Nu)	10
Input Weighting rate(R)	All equal to 0.0001
Output Weighting rate(Q)	All equal to 1
Input Min. Constraint ($Umin$)	*Umin(Ff1)=0* rest equal to -*inf.*
Input Max. Constraint ($Umax$)	*Umax(Ff1)=15* rest equal to *inf.*
Output Min. Constraint ($Ymin$)	All equal to -*inf.*
Output Max. Constraint ($Ymax$)	All equal to *inf.*

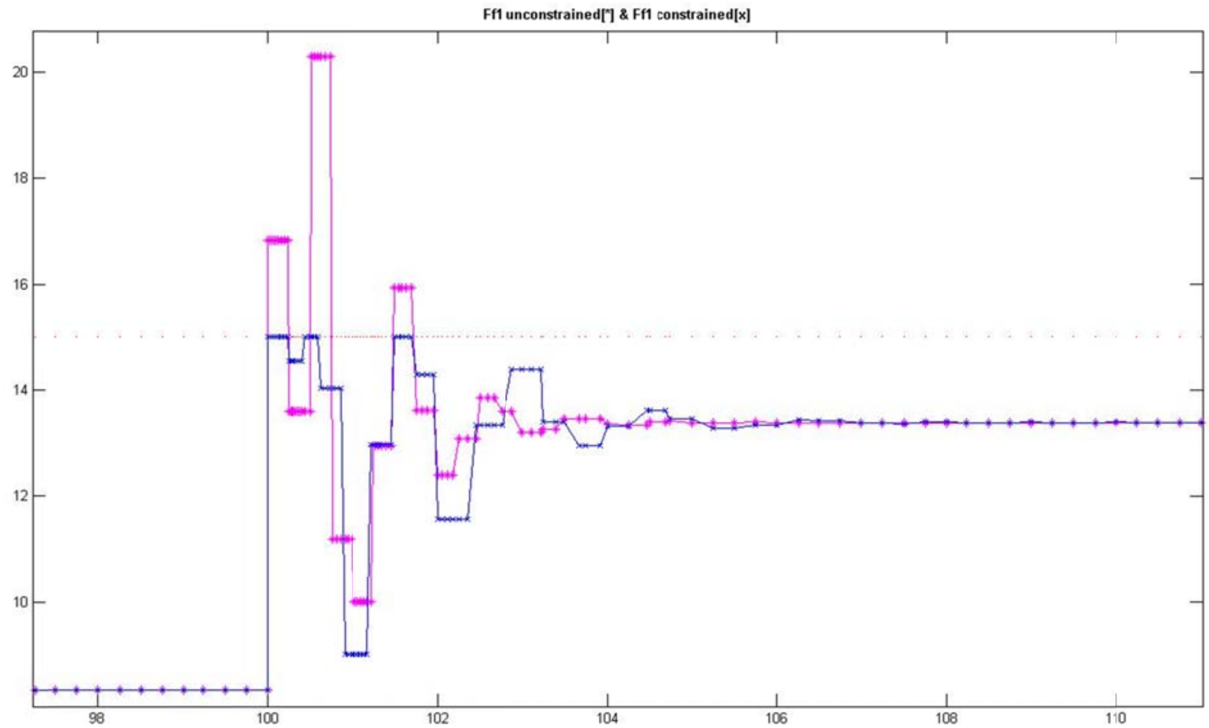


Figure 27: Result of constraint for *Ff1* with H1 setpoint change

So, in comparison with the unconstraint case, as illustrate in Figure 27, this variable is fluctuates in the determined range.

6.1.7 Simulation 7: output constraint

Next step is illustrating the output constraint and its effect in the system variables. In order to derive the constraint, the temperature of first reactor (*T1*) is considered with the maximum value set to *325*. The result is presented in Figure 28 with the comparison to the unconstrained case. Related parameters are referred to Table 9.

Table 9: Simulation7 parameters

Parameter	value
Sampling time (*Tc*) [*s*]	0.25
Prediction Horizon (*N*)	10
Control Horizon (*Nu*)	10
Input Weighting rate(*R*)	All equal to 0.0001
Output Weighting rate(*Q*)	All equal to 1
Input Min. Constraint (*Umin*)	All equal to -*inf.*
Input Max. Constraint (*Umax*)	All equal to *inf.*
Output Min. Constraint (*Ymin*)	All equal to -*inf.*
Output Max. Constraint (*Ymax*)	*Ymax(T1)=325* rest equal to *inf.*

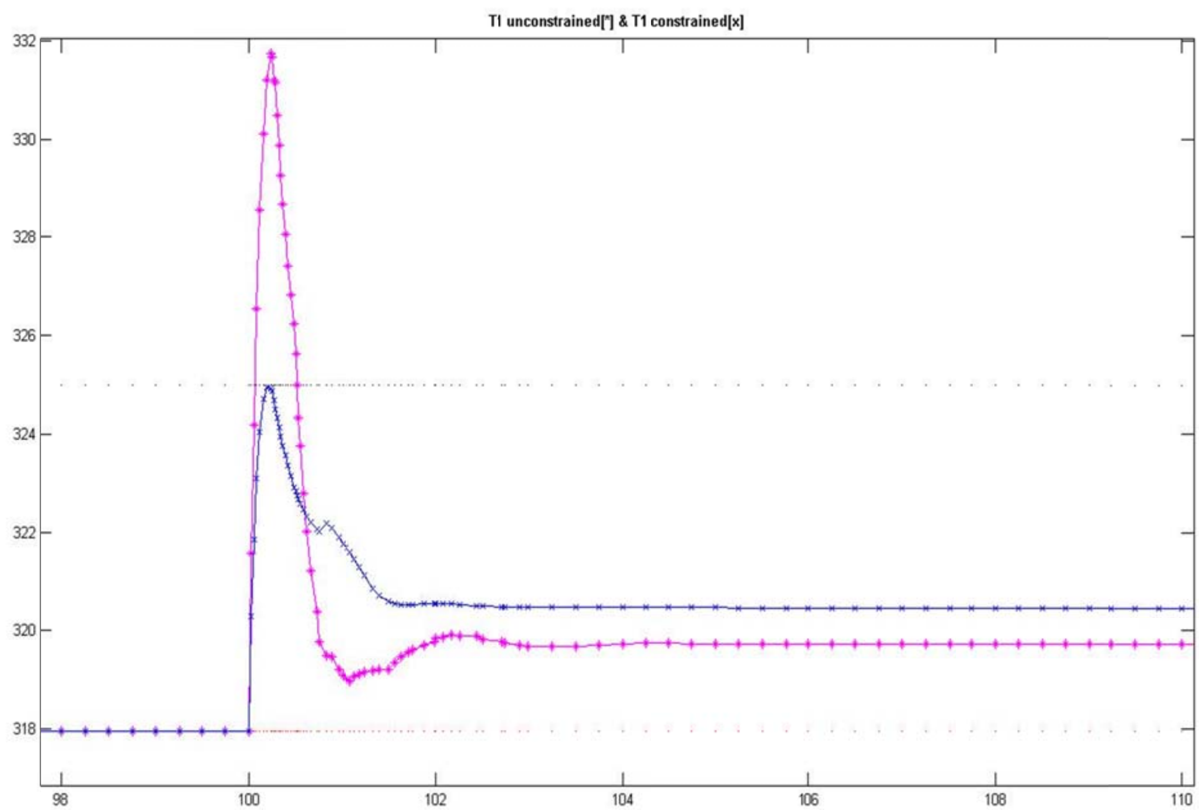


Figure 28: Result of constraint for *T1* with *H1* setpoint change

6.1.8 Conclusion on MPC controller without integral action

Within these experiment graphs of the MPC controller without integral action, the anticipated results have been achieved. The output tuning with new setpoint worked in the proper way and the controller make the system stable after due times, however there are some steady state errors according to the non-availability of the integral action.

Regarding the prediction and control horizons, we can declare that input horizon can be shorter than prediction (output) horizon, by losing some degree of freedom. Hence, loss of performance and decreased computational time in the way of smaller Quadratic Program, are resulted, which is a tradeoff for design the controller. However in this case, since the constraints are checked up to prediction horizon, feasibility is maintained.

Input disturbances can be rejected perfectly, within operating region.

Concerning the weight ratio for input and output, it can be concluded that the larger ratio of output by input weighting, the more aggressive the controller is.

Physical constraints can apply to the controller within MPC method and controller experience less aggressiveness, when small control deviations have been chosen.

6.2 Results for MPC controller with integral action

Resembling the case of controller without integral action, here we start analysis of the plant with the constrained MPC algorithm and Integral action. As a general control tuning analysis method, we apply the step to the setpoint and illustrate the result in that output trajectory and the influence of this parameter changing, in the complete system variables, as well.

Noted that MPC controller with the integral action aimed the augmented system in state space, thus the controller is driven by the state deviation plus output variation from the setpoint. Consequently, segregation of outputs through states is conceivable. To employ this benefit, the outputs are defined as [*H1, T1, H2, T2, H3, T3*] while the states are remained as [*H1, xA1, xB1, T1, H2, xA2, xB2 T2, H3, xA3, xB3, T3*]

6.2.1 Simulation 8: tuning the level of first reactor

The leading term in this simulation is the Level of reactor1 (*H1*), where the step of *5* units in the setpoint makes the following Figures 29, 30, 31. All the significant parameters are depicted in Table 10.

Table 10: Simulation8 parameters

Parameter	value
Sampling time (Tc) [s]	0.25
Prediction Horizon (N)	10
Control Horizon (Nu)	10
Input Weighting rate(R)	All equal to 0.0001
Output Weighting rate(Q)	All equal to 1
Input Min. Constraint ($Umin$)	All equal to -*inf.*
Input Max. Constraint ($Umax$)	All equal to *inf.*
Output Min. Constraint ($Ymin$)	All equal to -*inf.*
Output Max. Constraint ($Ymax$)	All equal to *inf.*

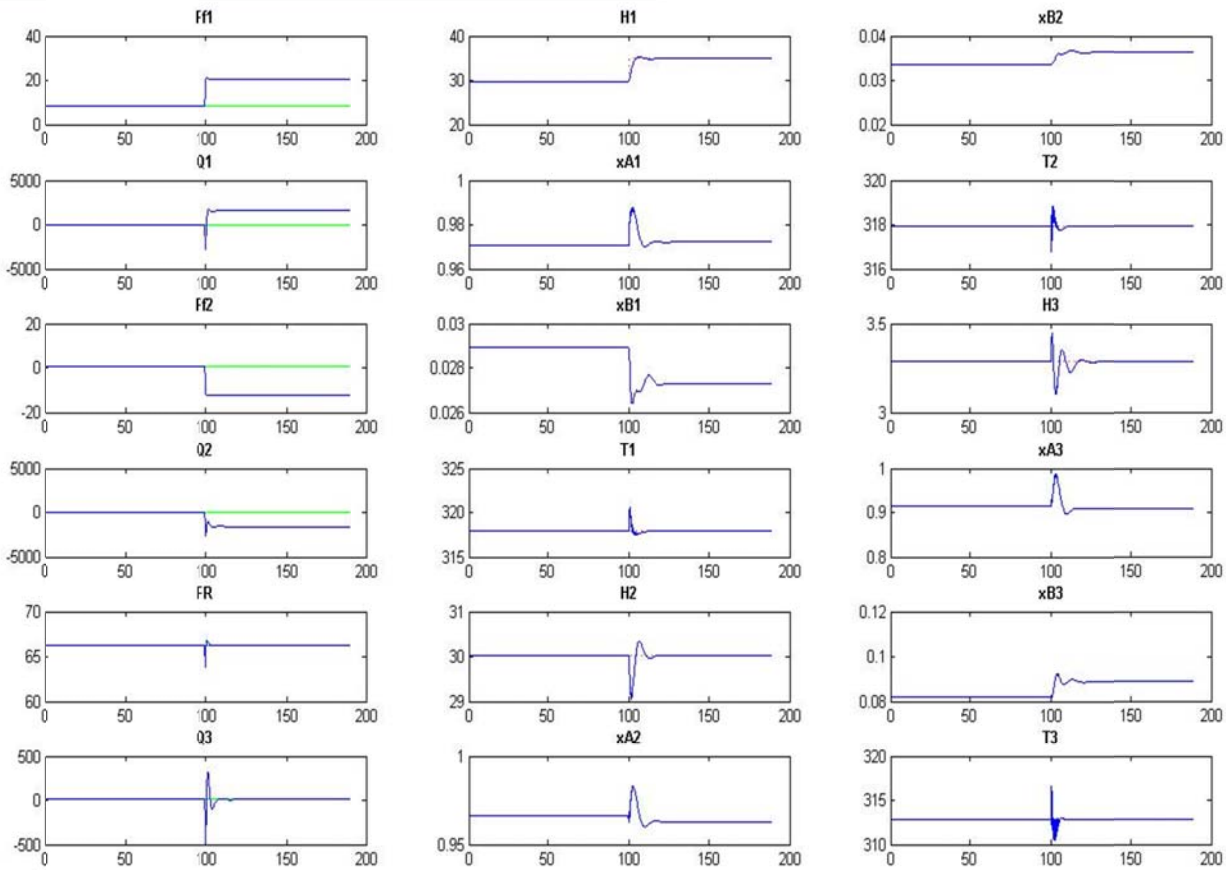


Figure 29: Result of *H1* setpoint change in the system with integrator

Figure 29 illustrate the tuning on *H1* in the entire system. As the step is applied in the time *100*, the zoomed plot has been shown in Figure 30.

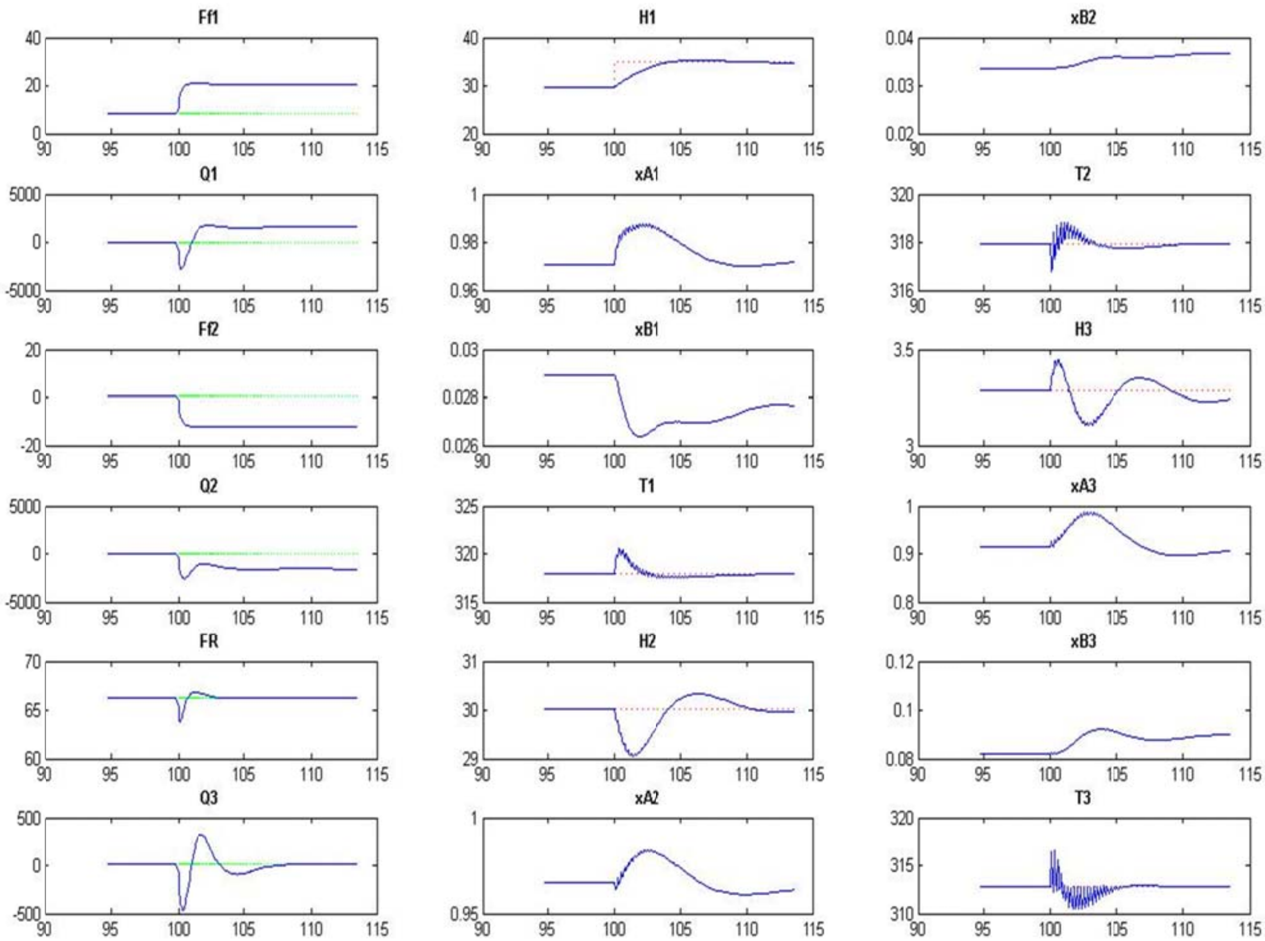


Figure 30: Result of *H1* setpoint change in the system with integrator-zoomed

According to Figure 30, the change of the setpoint in the level of first reactor, make the abrupt changes in all the manipulated variables (*Ff1,Q1,Ff2,Q2,FR,Q3*) with respect to the weighting of that variable in the cost function. Output tracking is correspondingly under the investigation, while the related weighting of the output in the cost function must be respected. As it has been noted in chapter 3, this type of controller can make the steady state error equal to zero, due to the constructed integral action. Since the outputs are different from the states, the remaining states (*xA* and *xB*) experience some variation and settle down after proper times, to the steady state values. The perturbation of output *H1* by dint of the change to its setpoint is illustrated in Figure 31. The output tracks the setpoint and reaches the steady state condition in less than *20* seconds with small overshoot and no unexpected oscillation. The settling time has been increased with respect to the controller without integral action.

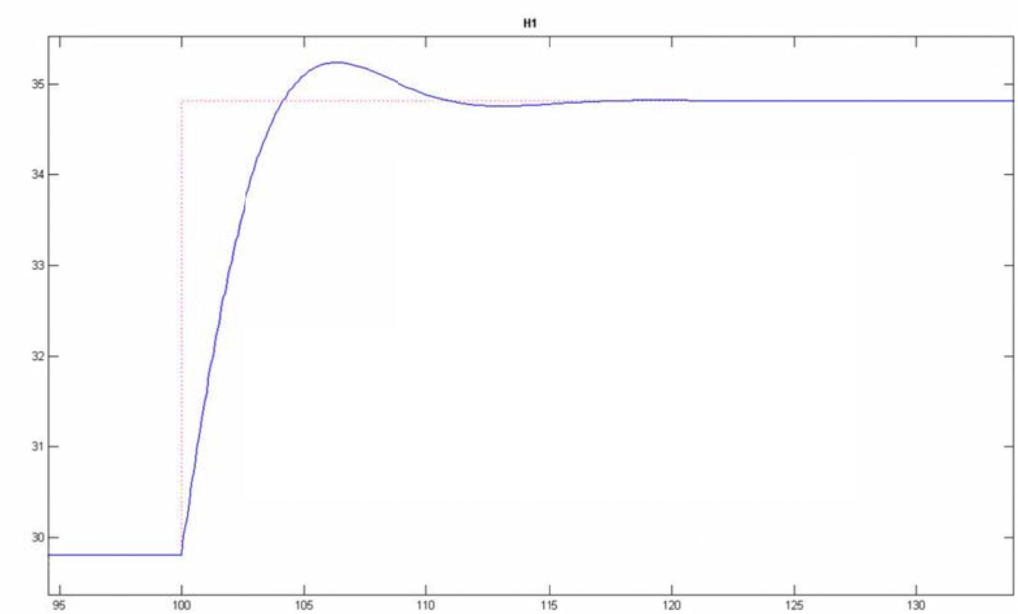


Figure 31: Response of *H1* to the setpoint change with integrator

In the following, and to investigate about weighting and horizons, the same as controller without integrator, it specifically focused on the two inputs (*Ff1* and *Ff2*) and two outputs (*H1* and *H2*). To have a proper view, the normalized values with respect to the equilibrium values are going to be used. From the above conditions we have:

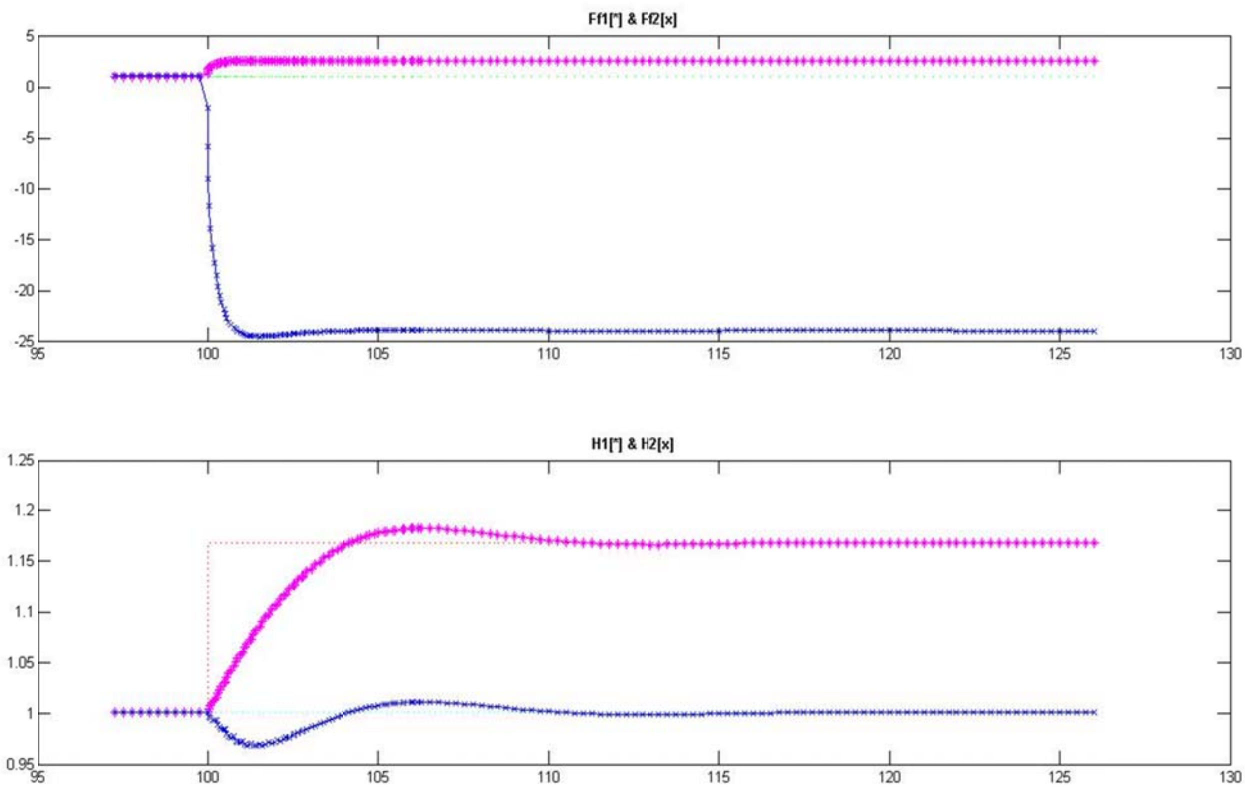


Figure 32: Result of *H1* setpoint change in *Ff1, Ff2, H1* and *H2* (normalized) with integrator

6.2.2 Simulation 9: input weighting effect

Through increasing the weighting rate of the first input among the others by the value of *10* and with respect to the parameters in Table 11, the following result has been achieved in Figure 33.

Table 11: Simulation9 parameters

Parameter	value
Sampling time (*Tc*) [*s*]	0.25
Prediction Horizon (*N*)	10
Control Horizon (*Nu*)	10
Input Weighting rate(*R*)	*R(Ff1)* equal to 10 The rest equal to 0.0001
Output Weighting rate(*Q*)	All equal to 1
Input Min. Constraint (*Umin*)	All equal to -*inf.*
Input Max. Constraint (*Umax*)	All equal to *inf.*
Output Min. Constraint (*Ymin*)	All equal to -*inf.*
Output Max. Constraint (*Ymax*)	All equal to *inf.*

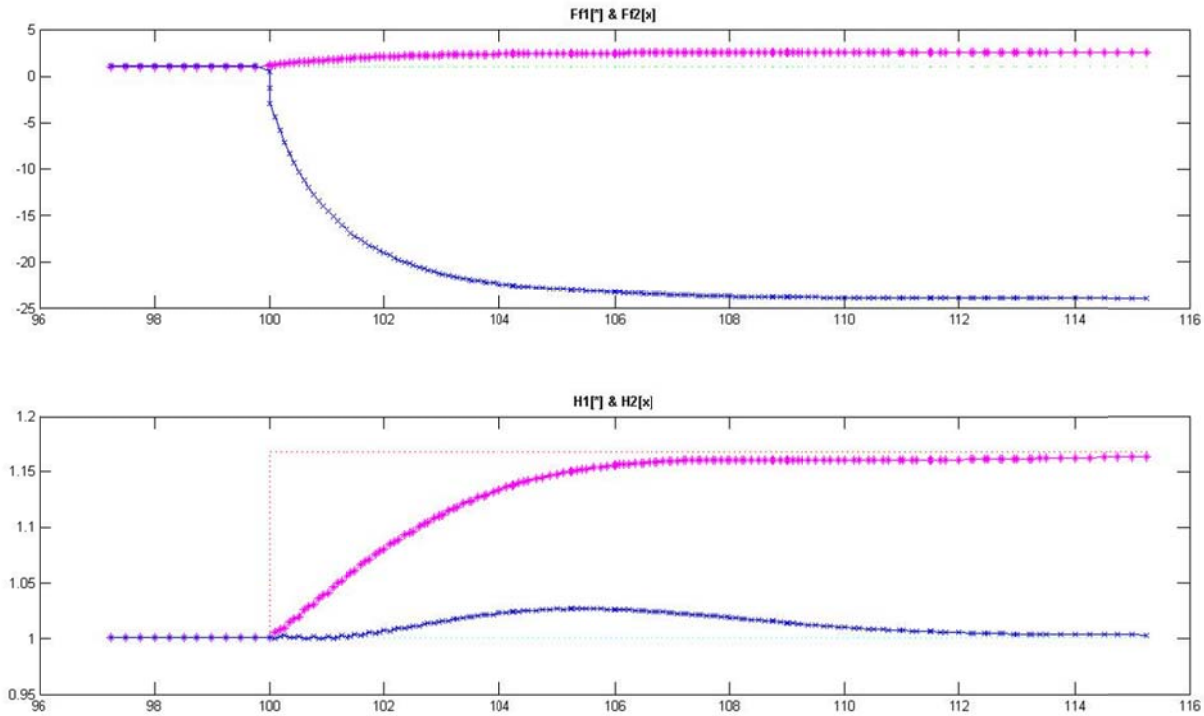


Figure 33: Result of input weighting with *H1* setpoint change in *Ff1, Ff2, H1* and *H2* (normalized) with integrator

It indicates the smoother variation of Input variable *Ff1* and faster settling in its steady state value, while the output *H1* experience more time to settle down, by consequence of imposing more weighting rate on the input *Ff1*. Due to the emphasizing on this special manipulated variable, other variable settling times are increased, to support the flatter change in *Ff1*.

6.2.3 Simulation 10: output weighting effect

To following the trend, the weighting rate of controlled variable *H1* has been increased and changed to *10* and the result with respect to Table 12 is available is in Figure 34.

Table 12: Simulation10 parameters

Parameter	value
Sampling time (*Tc*) [*s*]	0.25
Prediction Horizon (*N*)	10
Control Horizon (*Nu*)	10
Input Weighting rate(*R*)	All equal to 0.0001
Output Weighting rate(*Q*)	*Q(H1)* equal to 10 The rest equal to 1
Input Min. Constraint (*Umin*)	All equal to -*inf.*
Input Max. Constraint (*Umax*)	All equal to *inf.*
Output Min. Constraint (*Ymin*)	All equal to -*inf.*
Output Max. Constraint (*Ymax*)	All equal to *inf.*

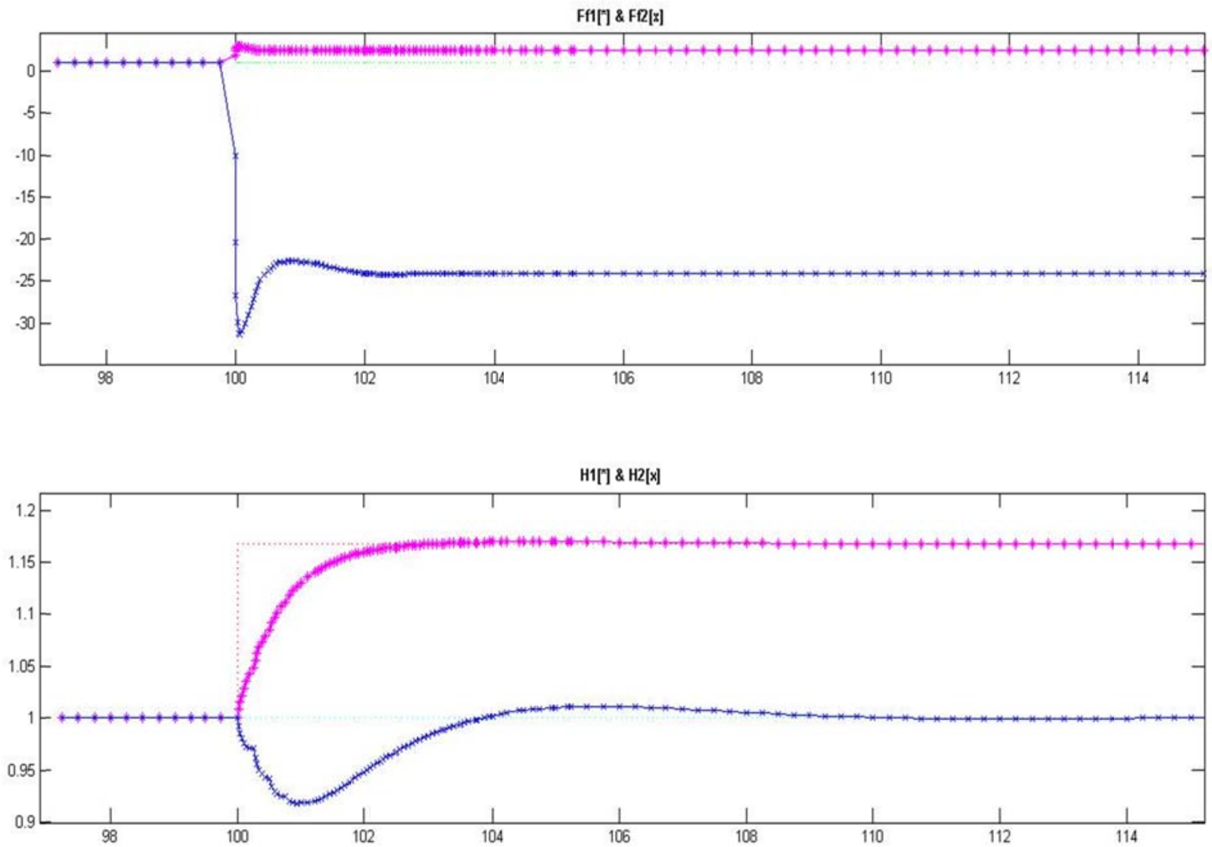


Figure 34: Result of output weighting with *H1* setpoint change in *Ff1, Ff2, H1* and *H2* (normalized) with integrator

The controlled output *H1* with higher weighting rate, settled in the steady state very quickly, which greater weighting has been dedicated to it.

6.2.4 Simulation 11: prediction and control horizons

Another parameter which is quite important in the field of predictive controllers is prediction horizon and control horizon. By laying $N=6$ and $Nu=1$ and respected parameters in Table 13, the resulted plot is in Figure 35.

Table 13: Simulation11 parameters

Parameter	value
Sampling time (Tc) [s]	0.25
Prediction Horizon (N)	6
Control Horizon (Nu)	1
Input Weighting rate(R)	All equal to 0.0001
Output Weighting rate(Q)	All equal to 1
Input Min. Constraint ($Umin$)	All equal to -*inf.*
Input Max. Constraint ($Umax$)	All equal to *inf.*
Output Min. Constraint ($Ymin$)	All equal to -*inf.*
Output Max. Constraint ($Ymax$)	All equal to *inf.*

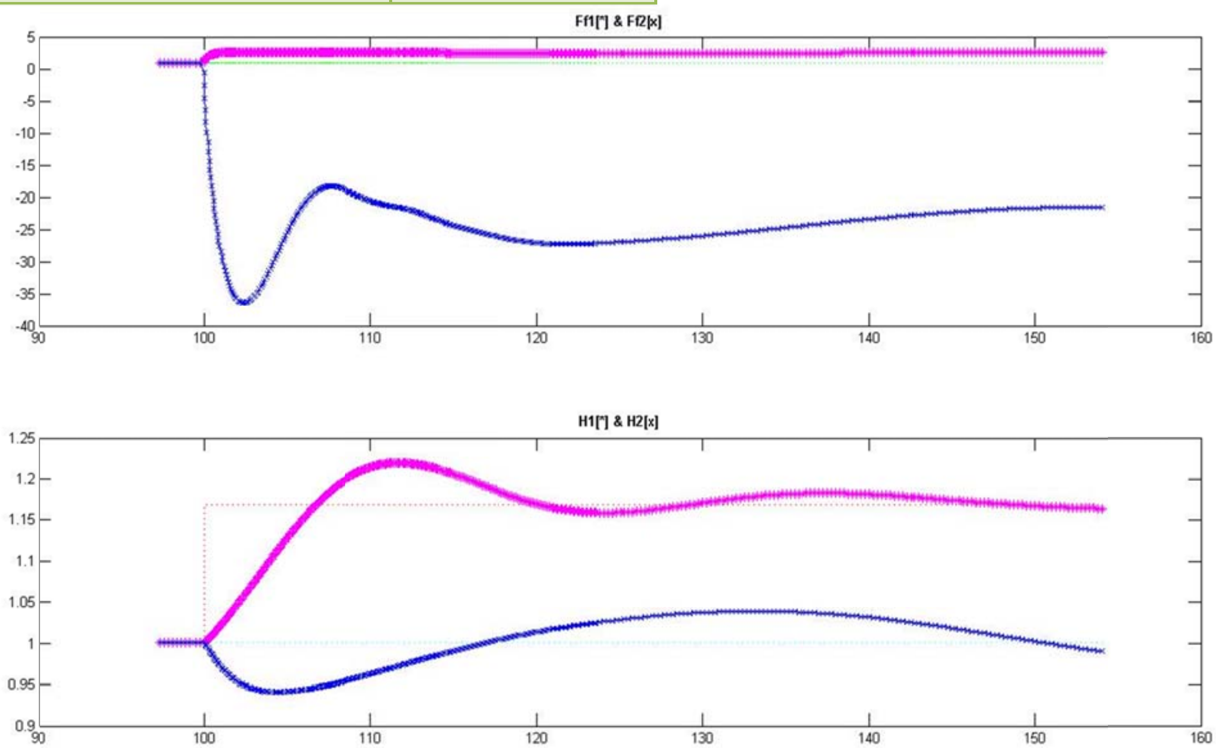


Figure 35: Result of horizons with *H1* setpoint change in *Ff1, Ff2, H1* and *H2* (normalized) with integrator

In this case, due to the slighter horizons for prediction, the settling time increased dramatically and variables experience more oscillation, which is quite predictable.

However, less optimization effort needs for such computation with reduced horizons.

6.2.5 Simulation 12: disturbance effect

To experience the effect of the disturbances, and evaluate the response of the plant with MPC controller, an initial perturbation has been forced to the input variable which can be presumed as an initial disturbance or the actuator disturbance. For this purpose, drive the manipulated variable *Ff1* with a step of *4* units after time *100* and squared the system responses in figure 36. All the parameters are referred to Table 10.

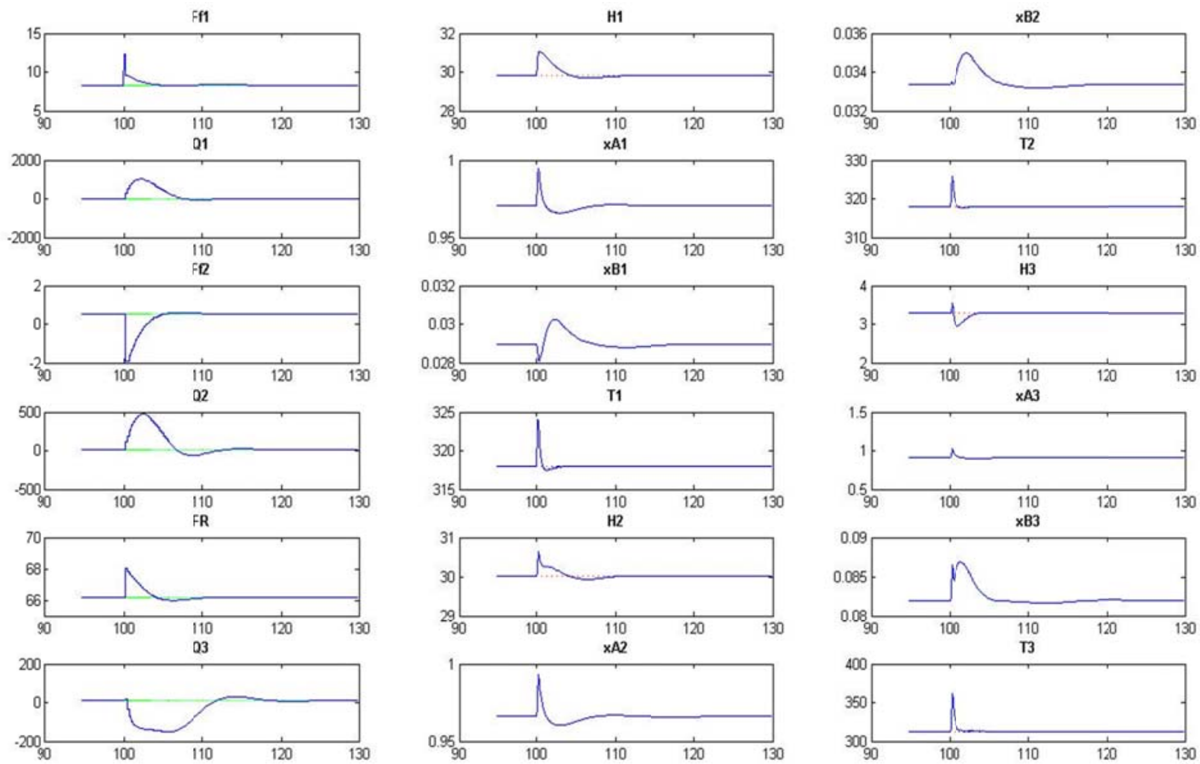


Figure 36: Result of disturbance in the system with integrator

According to Figure 36, the system started from its steady state condition and triggered by a step disturbance in the input variable, where the system can reject the disturbance and since there is an integral action in the construction of the MPC controller, the zero steady state errors have been observed in controlled variables.

6.2.6 Simulation 13: input constraint

Since one of the advantages of MPC algorithm is its ability to conclude the constraint on process inputs or outputs/states, here are some experimental results by forcing the constraint to the system.

We start from the first manipulated variable *dFf1* which is the variation of input *Ff1*, according to the formula in the chapter 3.5 for MPC controller with integral action, and set the constraint maximum to one deviation unit. The following is the result in figure 37. Other parameters are available in Table 13.

Table 14: Simulation13 parameters

Parameter	value
Sampling time (Tc) [s]	0.25
Prediction Horizon (N)	10
Control Horizon (Nu)	10
Input Weighting rate(R)	All equal to 0.0001
Output Weighting rate(Q)	All equal to 1
Input Min. Constraint (d$Umin$)	All equal to -*inf.*
Input Max. Constraint (d$Umax$)	*dUmax(Ff1)=1* rest equal to *inf.*
Output Min. Constraint ($Ymin$)	All equal to -*inf.*
Output Max. Constraint ($Ymax$)	All equal to *inf.*

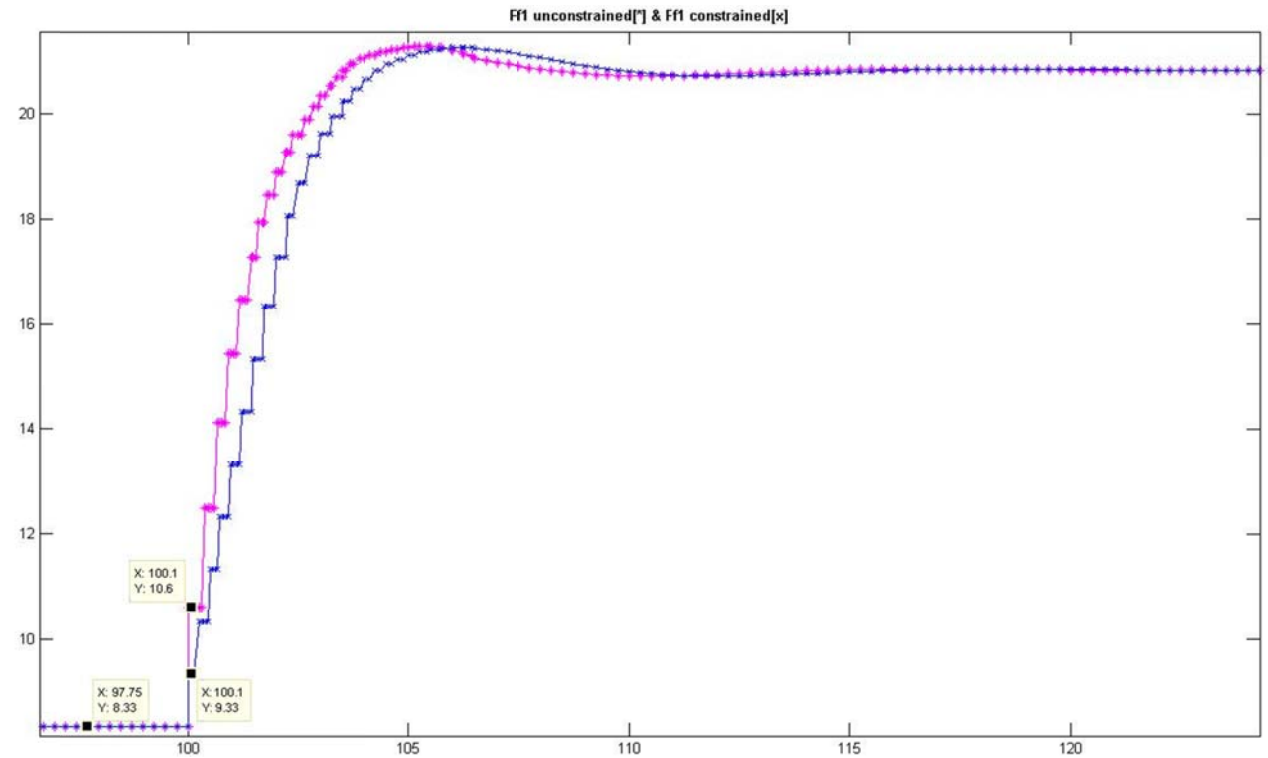


Figure 37: Result of constraint for dFf1 with *H1* setpoint change with integrator

where is illustrated that in the constrained case, the variation of *Ff1* in each step, has been limited by its maximum deviation constraint. This variation constraint, sometimes named as "skew rate", is mostly useful, when the actuator has specified properties like a pneumatic valve with some limits of opening or closing step.

6.2.7 Simulation 14: output constraint

Final step is illustrating the output constraint and its effect in the system variables. In order to derive the constraint, the level of second reactor (*H2*) is considered with the maximum value of *30.5* units. The result is shown in Figure 38 with the comparison to the unconstrained case. Related parameters are referred to Table 15.

Table 15: Simulation14 parameters

Parameter	value
Sampling time (*Tc*) [*s*]	0.25
Prediction Horizon (*N*)	10
Control Horizon (*Nu*)	10
Input Weighting rate(*R*)	All equal to 0.0001
Output Weighting rate(*Q*)	All equal to 1
Input Min. Constraint (*Umin*)	All equal to -*inf.*
Input Max. Constraint (*Umax*)	All equal to *inf.*
Output Min. Constraint (*Ymin*)	All equal to -*inf.*
Output Max. Constraint (*Ymax*)	*Ymax(H2)=30.5* rest equal to *inf.*

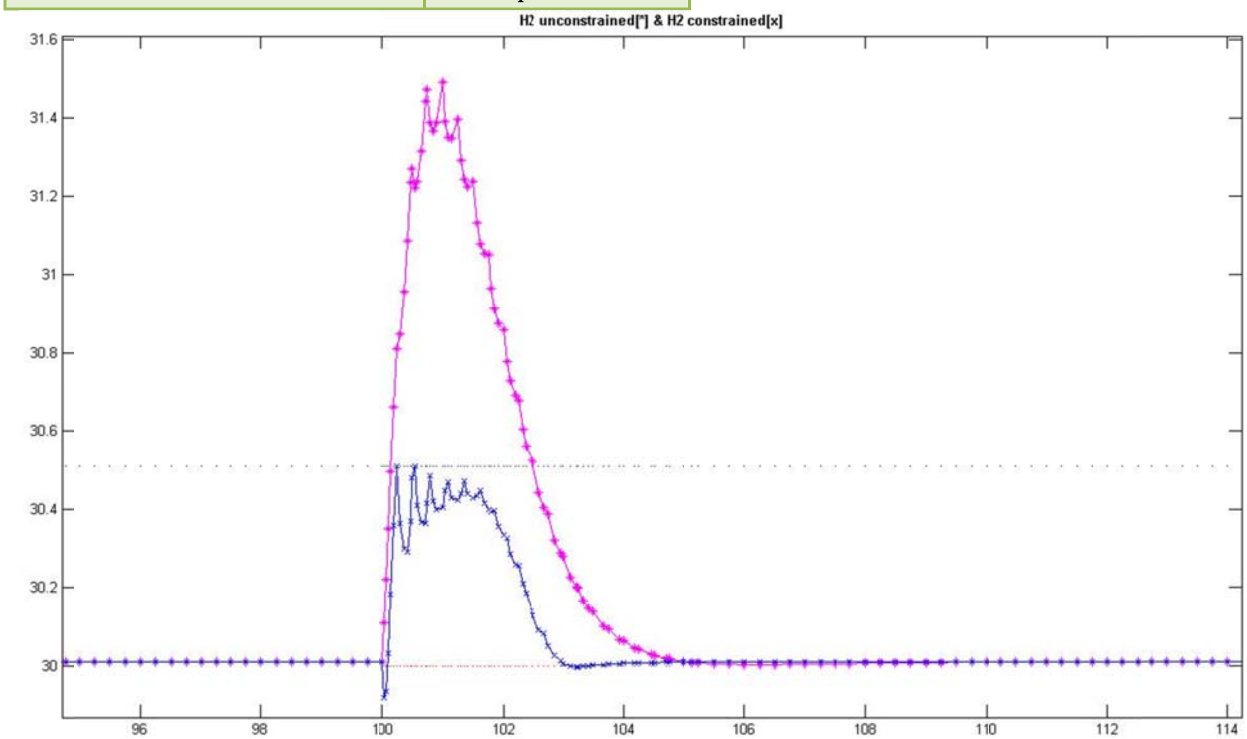


Figure 38: Result of constraint for H2 with *H1* setpoint change with integrator

6.2.8 Conclusion on MPC controller with integral action

Within these experiment results of the MPC controller with integral action, the anticipated results have been achieved. The output tuning with new setpoint worked in the proper way and the controller make the system stable after proper times which is settled on the setpoint value due to the availability of integral action.

Regarding the prediction and control horizons, we can assert that input horizon can be shorter than prediction (output) horizon, by losing some degree of freedom. Hence, loss of performance and decreased computational time with smaller Quadratic Program, are resulted, which is a tradeoff for design the controller. However in this case also, since the constrained are checked up to prediction horizon, feasibility is maintained. Larger control horizon makes the controller to perform better but increases the complexity. Smaller the prediction horizon derives the controller to be more aggressive. On the other hand, control variable deviation constraint can set the controller less aggressive.

Input disturbances can be rejected perfectly, within the operating region.

Concerning the weight ratio for input and output, it can be concluded that the larger ratio of the output by input weighting (*Q/R*), make the controller more aggressive.

As the ability to operate a process close to a limiting constraint is an important objective for advanced process control, for many industrial processes, the optimum operating condition occurs at a constraint limit. For these situations, the set point should not be the constraint value because a process disturbance could force the controlled variable beyond the limit. Thus, the set point should be set conservatively, based on the ability of the control system to reduce the effects of disturbances.

Appendix A

We here present the MATLAB codes for all part of program to enrich the content and been referred within the thesis report. Codes for modeling the subsystems is presented only for the first subsystem, while can be duplicated for the second and third subsystem. MPC algorithm, referring to the body of the report, has been generated through the approaches for controller without integral action and with integral action.

A.1 MATLAB codes of first subsystem model

```
% states
%
H1=u(1);
xA1=u(2);
xB1=u(3);
T1=u(4);
%
% inputs
%
Ff1=u(5);
FR=u(6);
Q1=u(7);
%
% inputs from other states
%
xA3=u(8);
xB3=u(9);
T3=u(10);
%
F1=kv1*H1;
x3s=alfaA*xA3+alfaB*xB3+alfaC*(1-xA3-xB3);
xAR=(alfaA*xA3)/x3s;
xBR=(alfaB*xB3)/x3s;
kA1=kA*exp(-EAR/T1);
kB1=kB*exp(-EBR/T1);
%
dx(1)=(1/(rho*A1))*(Ff1+FR-F1);
dx(2)=(1/(rho*A1*H1))*(Ff1*xA0+FR*xAR-F1*xA1)-kA1*xA1;
dx(3)=(1/(rho*A1*H1))*(FR*xBR-F1*xB1)+kA1*xA1-kB1*xB1;
```

```
dx(4)=(1/(rho*A1*H1))*(Ff1*T0+FR*TR-F1*T1)-
(1/Cp)*(kA1*xA1*DHA+kB1*xB1*DHB)+Q1/(rho*A1*Cp*H1);
dx(5)=F1;
```

A.2 MATLAB codes for preparation of MPC parameters without integral action

```
Tc=.25;
[A,B]=c2d(AA,BB,Tc);
C=CC;
[n,m]=size(B);
[p,n]=size(C);
M=zeros(n,1);
d=1;
%
R=diag([1.10  0.000010  0.000010  0.000010  0.000010  0.000010]);
QY=diag([10  1  1  1  1  1  1  1  1  1  1  1]);
Q=C'*QY*C;
[K,P,E] = DLQR(A,B,Q,R);
S=P;
SY=eye(p);
%
umin=[-1000000 ; -1000000;  -1000000 ; -1000000 ; -1000000 ; -1000000];
umax=[1000000 ; 1000000 ; 1000000 ; 1000000 ; 1000000 ; 1000000];
%
ymin=[-1000000  ;  -1000000  ;  -1000000  ;-1000000;  -1000000  ;  -1000000;  -
1000000; -1000000; -1000000; -1000000; -1000000; -1000000];
ymax=[1000000  ;   1000000;   1000000;    1000000;   1000000  ;   1000000;
1000000; 1000000; 1000000; 1000000; 1000000; 1000000 ];
%
Np=10;
Nu=10;
%
% construction of matrices for quadprog of output
%
H=2*(CBT'*QTY*CBT+RT);
D=zeros(Np*d,1);
```

A.3 MATLAB codes for construction of matrixes without integral action

```
% construct the matrices for predictive control of linear systems
% A,B : matrix of the system (possibly extended first with integrators)
% Q, S, QY, SY, R : weight matrices of state, final state, output, final
output, control
% Np, Nu : horizons of prediction and control
%  umin,umax,  dumin,dumax  :  limits  on  the  values  ??of  control  and
variation
% of control
[n,n]=size(A);
```

```
[n,m]=size(B);
[p,n]=size(C);
[n,d]=size(M);
%
% construction of the matrices AT, BT, QT, RT, MT, CAT,CBT,QTY, CMT
%
AT=[];
CAT=[];
for i=1:Np
 AT=[AT;A^i];
 CAT=[CAT;C*A^i];
end
BT=[];
MT=[];
CBT=[];
CMT=[];
for i=1:Np
    TE=[];
    CTE=[];
    ME=[];
    CME=[];
    for j=1:i
        TE=[A^(j-1)*B TE];
        CTE=[C*A^(j-1)*B CTE];
        ME=[A^(j-1)*M ME];
        CME=[C*A^(j-1)*M CME];
    end
    TE=[TE zeros(n,(Np-i)*m)];
    CTE=[CTE zeros(p,(Np-i)*m)];
    BT=[BT;TE];
    CBT=[CBT;CTE];
    ME=[ME zeros(n,(Np-i)*d)];
    CME=[CME zeros(p,(Np-i)*d)];
    MT=[MT;ME];
    CMT=[CMT;CME];
end

QT=[];
QTY=[];
for i=1:Np-1
    QT=[QT;zeros(n,(i-1)*n) Q zeros(n,(Np-i)*n)];
    QTY=[QTY;zeros(p,(i-1)*p) QY zeros(p,(Np-i)*p)];
end
QT=[QT;zeros(n,(Np-1)*n) S];
QTY=[QTY;zeros(p,(Np-1)*p) SY];
RT=[];
for i=1:Np
    RT=[RT;zeros(m,(i-1)*m) R zeros(m,(Np-i)*m)];
end
%
% reduction of the matrices BT and RT if Nu<Np
```

```
%
if Np>Nu
    for i=1:Np-Nu
        BT(:,(Nu-1)*m+1:Nu*m)=BT(:,(Nu-1)*m+1:Nu*m)+BT(:,(Nu-
1+i)*m+1:(Nu+i)*m);
        CBT(:,(Nu-1)*m+1:Nu*m)=CBT(:,(Nu-1)*m+1:Nu*m)+CBT(:,(Nu-
1+i)*m+1:(Nu+i)*m);
    end
    BT=BT(:,1:Nu*m);
    CBT=CBT(:,1:Nu*m);
    RT=RT(1:Nu*m,1:Nu*m);
    for i=1:Np-Nu
        RT((Nu-1)*m+1:Nu*m,(Nu-1)*m+1:Nu*m)=RT((Nu-1)*m+1:Nu*m,(Nu-
1)*m+1:Nu*m)+R;
    end
end
%
% construction of control constraint matrix
%
Avinc=eye(Nu*m);
Avinc=[Avinc;-eye(Nu*m)];
Avinc=[Avinc;CBT];
Avinc=[Avinc;-CBT];
%
Bvinc=[];
for i=1:Nu
    Bvinc=[Bvinc;umax];
end
for i=1:Nu
    Bvinc=[Bvinc;-umin];
end
for i=1:Np
    Bvinc=[Bvinc;ymax];
end
for i=1:Np
    Bvinc=[Bvinc;-ymin];
end
%
Cvinc=zeros(Nu*m,p);
Cvinc=[Cvinc;zeros(Nu*m,p)];
Cvinc=[Cvinc;CAT];
Cvinc=[Cvinc;-CAT];
% matrices for Terminal constraint
%
Afin=A^Np;
Bfin=BT(end-n+1:end,:);
Mfin=MT(end-n+1:end,:);
```

A.4 MATLAB codes for MPC function without integral action

```
x0=u(1:12);
f=2*(CAT*x0)'*QTY*CBT;
b=Bvinc+Cvinc*x0;
AF=Bfin;
BF=-Afin*x0;
 USOL=QUADPROG(H,f,Avinc,Bvinc,AF,BF);
dx=USOL(1:m);
```

A.5 MATLAB codes for preparation of MPC parameters with integral action

```
Tc=.25;
[A,B]=c2d(AA,BB,Tc);

CI=[1 0 0 0 0 0 0 0 0 0 0 0;0 0 0 1 0 0 0 0 0 0 0 0;...
    0 0 0 0 1 0 0 0 0 0 0 0;0 0 0 0 0 0 0 1 0 0 0 0;...
    0 0 0 0 0 0 0 0 1 0 0 0;0 0 0 0 0 0 0 0 0 0 0 1];
%
A=[A zeros(12,6);CI*A eye(6)];
B=[B;CI*B];
C=eye(18);
[n,m]=size(B);
[p,n]=size(C);
M=zeros(n,1);
d=1;
%
R=diag([10.10  0.000010  0.000010  0.000010  0.000010  0.000010]);
QY=diag([10  1  1  1  1  1  1  1  1  1  1  1  10  1  1  1  1  1]);
Q=QY;
[K,P,E] = DLQR(A,B,QY,R);
S=P;
SY=eye(p);
%
umin=[-1000000 ; -1000000;  -1000000 ; -1000000 ; -1000000 ; -1000000];
umax=[1000000 ; 1000000 ; 1000000 ; 1000000 ; 1000000 ; 1000000];
ymin=[-1000000  ;  -1000000  ;  -1000000  ;-1000000;  -1000000  ;  -1000000;  -
1000000;  -1000000;  -1000000;  -1000000;  -1000000;  -1000000;  -1000000;  -
1000000; -1000000; -1000000; -1000000; -1000000];
ymax=[1000000  ;   1000000;   1000000;    1000000;   1000000  ;   1000000;
1000000; 1000000; 1000000; 1000000; 1000000; 1000000; 1000000; 1000000 ;
1000000; 1000000; 1000000; 1000000 ];
%
Np=10;
Nu=10;
%
% construction of matrices for quadprog of output
%
H=2*(CBT'*QTY*CBT+RT);
```

```
D=zeros(Np*d,1);
```

A.6 MATLAB codes for construction of matrixes with integral action

```
% construct the matrices for predictive control of linear systems
% A,B : matrix of the system (possibly extended first with integrators)
% Q, S, QY, SY, R : weight matrices of state, final state, output, final
output, control
% Np, Nu : horizons of prediction and control
% umin,umax, dumin,dumax : limits on the values ??of control and
variation
% of control

[n,n]=size(A);
[n,m]=size(B);
[p,n]=size(C);
[n,d]=size(M);
%
% construction of the matrices AT, BT, QT, RT, MT, CAT,CBT,QTY, CMT
%
AT=[];
CAT=[];
for i=1:Np
 AT=[AT;A^i];
 CAT=[CAT;C*A^i];
end
BT=[];
MT=[];
CBT=[];
CMT=[];
for i=1:Np
    TE=[];
    CTE=[];
    ME=[];
    CME=[];
    for j=1:i
        TE=[A^(j-1)*B TE];
        CTE=[C*A^(j-1)*B CTE];
        ME=[A^(j-1)*M ME];
        CME=[C*A^(j-1)*M CME];
    end
    TE=[TE zeros(n,(Np-i)*m)];
    CTE=[CTE zeros(p,(Np-i)*m)];
    BT=[BT;TE];
    CBT=[CBT;CTE];
    ME=[ME zeros(n,(Np-i)*d)];
    CME=[CME zeros(p,(Np-i)*d)];
    MT=[MT;ME];
    CMT=[CMT;CME];
```

```
end

QT=[];
QTY=[];
for i=1:Np-1
    QT=[QT;zeros(n,(i-1)*n) Q zeros(n,(Np-i)*n)];
    QTY=[QTY;zeros(p,(i-1)*p) QY zeros(p,(Np-i)*p)];
end
QT=[QT;zeros(n,(Np-1)*n) S];
QTY=[QTY;zeros(p,(Np-1)*p) SY];
RT=[];
for i=1:Np
    RT=[RT;zeros(m,(i-1)*m) R zeros(m,(Np-i)*m)];
end
%
% reduction of the matrices BT and RT if Nu<Np
%
if Np>Nu
    for i=1:Np-Nu
        BT(:,(Nu-1)*m+1:Nu*m)=BT(:,(Nu-1)*m+1:Nu*m)+BT(:,(Nu-
1+i)*m+1:(Nu+i)*m);
        CBT(:,(Nu-1)*m+1:Nu*m)=CBT(:,(Nu-1)*m+1:Nu*m)+CBT(:,(Nu-
1+i)*m+1:(Nu+i)*m);
    end
    BT=BT(:,1:Nu*m);
    CBT=CBT(:,1:Nu*m);
    RT=RT(1:Nu*m,1:Nu*m);
    for i=1:Np-Nu
        RT((Nu-1)*m+1:Nu*m,(Nu-1)*m+1:Nu*m)=RT((Nu-1)*m+1:Nu*m,(Nu-
1)*m+1:Nu*m)+R;
    end
end
%
% construction of control constraint matrix
%
Avinc=eye(Nu*m);
Avinc=[Avinc;-eye(Nu*m)];
Avinc=[Avinc;CBT];
Avinc=[Avinc;-CBT];
%
Bvinc=[];
for i=1:Nu
    Bvinc=[Bvinc;umax];
end
for i=1:Nu
    Bvinc=[Bvinc;-umin];
end
for i=1:Np
    Bvinc=[Bvinc;ymax];
end
for i=1:Np
```

```
    Bvinc=[Bvinc;-ymin];
end
%
Cvinc=zeros(Nu*m,p);
Cvinc=[Cvinc;zeros(Nu*m,p)];
Cvinc=[Cvinc;-CAT];
Cvinc=[Cvinc;CAT];
% matrices for Terminal constraint
%
Afin=A^Np;
Bfin=BT(end-n+1:end,:);
Mfin=MT(end-n+1:end,:);
```

A.7 MATLAB codes for MPC function with integral action

```
x0=u(1:18);
f=2*(CAT*x0)'*QTY*CBT;
b=Bvinc+Cvinc*x0;
AF=Bfin;
BF=-Afin*x0;
USOL=QUADPROG(H,f,Avinc,b,AF,BF);
dx=USOL(1:m);
```

Bibliography

[1] J. M. Maciejowski: “Predictive Control with Constraints”, *Prentice Hall*, 2001.

[2] R. Scattolini, “Lecture Notes for SIDRA Summer School 2011 on Model Predictive Control”*, ftp://ftp.elet.polimi.it/outgoing/Riccardo.Scattolini /Bertinoro2011_controllo_predittivo/*

[3] C.E. Garcia, D.M. Prett, M. Morari: “Model predictive control: theory and practice – a survey”, *Automatica*, Vol. 25, n. 3, pp. 335-348, 1989.

[4] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O. Scokaert: “Constrained model predictive control: stability and optimality”, *Automatica*, Vol. 36, pp. 789-814, 2000.

[5] B.T. Stewart, A.N. Venkat, J. B. Rawlings, S. J. Wright, G. Pannocchia: “Cooperative distributed model predictive control “, Systems & Control Letters 59 (2010) 460_469.

[6] D.E. Seborg, T.F. Edgar, D.A. Mellichamp: “Process dynamics and control”, *John Wiley & Sons*, 2004.

[7] M.J. Willis: “Continuous stirred tank reactor models”, *Dept. of Chemical and Process Eng., Uni. of Newcastle*, 2000.

[8] S.J. Qin, T.A. Badgwell: “A survey of industrial model predictive control technology”, *Control Engineering Practice*, Vol. 11, n. 7, pp. 733-764.

[9] M. Bauer, I.K. Craig: “Economic assessment of advanced process control – A survey and framework”, *Journal of Process Control*, Vol. 18, n. 1, pp. 2-18, 2008.

[10] M. Kano, M. Ogawa: “The state of the art in chemical process control in Japan: Good practice and questionnaire survey”, Journal of Process Control 20 (2010) 969–982.

[11] J.A. Rossiter: “Model based predictive control: a practical approach”, *CRC Press*, 2003.

[12] M. Morari, J. H. Lee: “Model predictive control: past, present and future”, *Computers and Chemical Engineering*, Vol. 23, n. 4-5, pp. 667-682, 1999.

[13] J.A. Richalet, A. Rault, J.D. Testud, J. Papon: “Model predictive heuristic control: applications to industrial processes”, *Automatica*, Vol. 14, pp. 413-428, 1978.

[14] M.L. Luyben, W.L. Luybed: “Design and Control of a Complex Process Involving Two Reaction Steps, Three Distillation Columns, and Two Recycle Streams” , *Ind. Eng. Chem. Res*. 1996,34, 3885-3898.

[15] B. Wayne: “Nonlinear Control of Chemical Processes: A Review”, *Ind. Eng. Chem. Res*. 1991, 30, 1391-1413.

[16] P.B. Sistu, B. Wayne: “Nonlinear Predictive Control of Uncertain Processes: Application to a CSTR”, *AIChE Journal*, 1991, Vol. 37, No. 11.