Titre: Title:	Dynamics, Control and Extremum Seeking of the Rectisol Process
Auteur: Author:	Mohammad Ghodratnama
Date:	2013
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Ghodratnama, M. (2013). Dynamics, Control and Extremum Seeking of the Rectisol Process [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1262/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/1262/
Directeurs de recherche: Advisors:	Michel Perrier
Programme: Program:	Génie chimique

UNIVERSITÉ DE MONTRÉAL

DYNAMICS, CONTROL AND EXTREMUM SEEKING OF THE RECTISOL PROCESS

MOHAMMAD GHODRATNAMA
DÉPARTEMENT DE GÉNIE CHIMIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE CHIMIQUE)
NOVEMBRE 2013

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

DYNAMICS, CONTROL AND EXTREMUM SEEKING OF THE RECTISOL PROCESS

présenté par : GHODRATNAMA Mohammad
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. Srinivasan Bala, Ph.D., président
M. Perrier Michel, Ph.D., membre et directeur de recherche
M. Abatzoglou Nicolas, Ph.D., membre

To My parents,
To whom I own everything I have. . .

ACKNOWLEDGEMENTS

Approaching the end of this period, I would like to thank those who helped realize the work done. First of all, It is with immense gratitude that I acknowledge the support of my supervisor, Professor Michel Perrier, for guiding me through out the path and for his patience and for showing me that it's not hard to follow your passion. I would also like to thank the members of his team and my colleagues, Guillaume for helping me out whenever I needed his help and encouraging me. Didac, Samin , Masood, Javier and Prof. Srinivasan for sharing their knowledge with me. I feel also responsible to thank Prof. Zhu who reminded me that the right way starts from the basics.

I also appreciate the cooperation of the department's informatics technician, Mehdi, and Prof. Francois Bertrand for allowing me to use the resources of URPEI to fulfill the requirements of this project.

I would also like to thank my former officemates at A-562, Ata, Majid, Ebrahim, Amirhosein, Richard,Patrice and Romain and my current officemates at URPEI, Julie, Bruno, Inci, Selim, Christopher, Benoit, Ebrahim and David for creating a friendly and warm environment.

I wish also to show my profound gratitude to my parents, my brother and sister, who where on my side and supported me during this time, and to as well for their encouragement and help.

I also wish to thank my friends at AECSP, who made my experience at Polytechnique fun, my friends and family in Iran all my friends in montreal,Shahab, Shervin, Helia, Milad, Amin, Amir, Danial,Tareq, Fanny and their son Tristan.

Since there's the rest of the page left, I would like to thank Metallica, Linkin park, Breaking Benjamin, Three days grace, 30 seconds to mars and Hoobastank for making great music that helped me significantly while running simulations.

RÉSUMÉ

Pendant la dernière décennie, les bioraffineries basées sur la gazéification ont fait l'objet de nombreuses études dans le cadre des efforts mondiaux visant à remplacer les combustibles fossiles qui produisent de l'énergie et des produits chimiques à valeur ajoutée. Une partie importante de ces bioraffineries est l'unité de purification des gaz de synthèse issus de l'oxydation partielle, qui enlève le CO_2 et l'H_2S. Un des procédés de purification considéré dans ces études est le Rectisol. Ce procédé est utilisé car il est plus environnemental et requièrt moins de coûts d'investissement et d'opération par rapport à d'autres procédés similaires.
Afin de faire l'étude dynamique de ce procédé, une simulation en régime permanent a, d'abord, été menée à l'aide du logiciel Aspen plus ®. Le comportement de ce modèle a été étudié et validé par rapport aux données trouvées dans la littérature. Des vannes de contrôle ont été placées dans les endroits nécessaires. Après avoir dimensionné les équipements, tels que les séparateurs, les vannes, les puisards de colonne et les condenseurs, les pressions ont été vérifiées pour que celles des courants entrants à l'équipement s'accordent avec la pression dans la zone d'entrée de l'équipement. Le modèle a été exporté en Aspen plus Dynamics et les effets des entrées de modèle et des perturbations ont été étudiés sur les variables de sorties.
Vu que la composition et les caractéristiques de la biomasse gazéifiée varient, la composition et la quantité d'impuretés du gaz produit changent aussi. Ceci crée alors des variations au niveau de la pureté du gaz de synthèse et des sous-produits de l'unité de purification du gaz. Dans une usine, il est important de garder les compositions de produits aussi constantes que possible afin de ne pas créer de perturbations dans les unités en aval. Pour surmonter ces variations, un schéma de recherche d'extremum adaptatif a été implanté. Il consiste à optimiser une fonction objectif quadratique des compositions de produit pour laquelle la relation entre les variables indépendantes et la fonction objectif est inconnue.
Pour que la recherche d'extremum soit bien efficace, une structure de contrôle régulateur sensible, à l'échelle de l'usine, est nécessaire. Les procédés de purification des gaz basés sur l'absorption ont tous un courant de recyclage du solvant, ce qui peut être problématique au niveau du contrôle des procédés. Une recherche a donc été menée sur les techniques de contrôle conventionnelles et avancées. Quatre stratégies potentielles de contrôle ont été mises en œuvre et leurs performances ont été analysées. Ces quatre stratégies sont : PI, MPC centralisé, MPC distribué et MPC décentralisé. La raison pour laquelle nous avons choisi des contrôleurs MPC est qu'ils peuvent envisager systématiquement, à la fois, les interactions entre les variables et les contraintes sur les entrées et sorties dans les calculs de contrôle.

Parmi les quatre stratégies, MPC distribué et MPC centralisé sont apparue comme les plus performante en terme de rejet de perturbations du flux d'entrée et de suivi des consignes. Pour ces deux stratégies, un schéma de recherche d'extremum adaptatif à plusier entreés a été conçu et appliqué et leurs performances ont été étudiées pour différentes fréquences de signal d'excitation. Les résultats ont montré que, pour la fréquence la plus performante, les deux combinaisons de structures d'optimisation et de contrôle ont un comportement identique. Pour finir, la combinaison de la recherche d'extremum adaptatif avec MPC distribué a été choisie comme structure d'optimisation et de contrôle pour l'usine de Rectisol étudiée. En effet, MPC distribué est moins sensible aux pannes et le contrôle de l'installation ne dépend pas d'un seul agent de contrôle. En conclusion, nous avons rendu le procédé Rectisol plus robuste aux perturbations sur la composition et le débit d'entrée afin que l'usine soit capable de garder ses compositions de produits les plus proches possibles des spécifications souhaitées.

ABSTRACT

Gasification based biorefineries have been studied in the past decade as part of a global effort to replace fossil fuels to produce energy and added value chemicals. An important part of these biorefineries is the acid gas removal units, that remove CO_2 and H_2S from the produced synthesis gas. One of the acid gas removal processes associated in these studies is Rectisol. Rectisol has been chosen since it's environmental friendly and requires a lower amount of operational and capital costs compared to its opponents.

To carry out a dynamic study of the process, as a first step, a steady-state simulation was carried out in Aspen Plus ®. The steady-state behavior of the columns were studied and validated based on data found in the literature. Control valves were placed in all the necessary places. After sizing the equipment, such as seperation drums, valves and column sumps, the pressures were varified, so that the pressure at the inlet of each equipment corresponds to incoming stream. Later on the model was exported to Aspen plus dynamics and the effect of different inputs and disturbances on the outputs were studied.

Due to the fact that the composition of the gasified biomass varies, the composition and the amount of impurities in the gasification gas also varies This creates variations in the purities of the syngas and byproducts of acid gas removal units. In any chemical plant it is important to keep compositions of products as constant as possible so that we don't create perturbations in downstream units. To overcome these variations an adaptive extremum control scheme was implemented that optimizes a quadratic objective function of product compositions, while the relation between the objective function and its independent variables is unknown.

For the adaptive extremum seeking control to be effective, a responsive plantwide regulatory control structure is required. Absorption based gas cleaning processes like Rectisol all have a recycle flow of solvent. This recycle flow can always be problematic from a process control point of view. Thus a search was conducted amongst the conventional and advanced control techniques. Four potential control strategies were implemented and their performance was analyzed. These four strategies were Multiloop PI, Centralized Model Predictive Control (MPC), Decentralized MPC and Distributed MPC. The reason we have chosen MPC is that these controllers can systematically consider process variable interactions and input and output constraints in their control calculations. Among the four, distributed and centralized MPC were found to be most effective in terms of rejecting input flow disturbances and tracking setpoints. Keeping this fact in mind a multivariable extremum-seeking scheme was designed and implemented on these two types of controllers and their performance was studied for different dither signal frequencies. The results showed that at the proper frequency, both combination of optimization and control structures have identical behavior.

At the end the combination of adaptive extremum seeking and Distributed MPC was chosen as the optimizing and control structure for the studied Rectisol plant, since Distributed MPC is more fault tolerant and the control of the plant will not depend on a single control agent.

In conclusion, Rectisol has been robustified to the composition and flowrate of the input and the plant is able to keep its product compositions as close as possible to the desired specifications.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIXES

LIST OF SYMBOLS AND ABREVIATIONS

List of Abbreviations

ARX	Auto Regressive with external input
COD	Constant Output Disturbance
DMC	Dynamic matrix control
DPS	Distributed Parameter system
FC	Flow Controller
FIR	Finite Impulse Response
FSR	Finite Step Response
GPC	Generalized Predictive control
IGCC	Integrated Gasificatiom combined Cycle
IMC	Internal Model Control
ISE	Integral of Squared Error
LC	Level Controller
LQG	Linear Quadratic Gaussian
LSS	Linear State Space
MIMO	Multi Input- Multi Output
MISO	Multi Input- Single Output
MPC	Model Predictive Control
NRMSE	Normalized root mean square error
PC	Pressure Controller
QDMC	Quadratic Dynamic Matrix control
RTO	RealTime Optimization
SVD	Singular Value Decomposition
TC	Temperature Controller
TF	Transfer function

List of Symbols

A, B, C	space state model parameters
a, b	constraint matrices
d	measured disturbance
F	state matrix in prediction generation
f	QP matrix

G	disturbance matrix in space state model
G_p	approximation of system model
H	QP matrix
J	MPC or LQG obejctive function
k	discrete time
k_c	proportional gain of PI
N_p	Prediction Horizon
N_c	Control Horizon
P	Pressure
Q	state or output weighting matrix
R	input weighting matrix
u_i	i^{th} Input of system
x	states of augmented system
x_{mi}	i^{th} state of system
y_i	i^{th} Output of system
$\hat{y}_i$	output of the identified model
Y	vector of predicted outputs
ΔU	predicted input sequence
$\Delta \bar{U}$	input sequence of other subsystems
Γ	disturbance matrix in prediction generation
λ	IMC tuning parameter
τ_I	Integral time of PID
τ_P	time constant of approximated model
θ	dead time of approximated model
Ψ	input of other subsystems matrix in prediction generation
Φ	input matrix in prediction generation

CHAPTER 1

INTRODUCTION

Work done in this thesis can be divided in two general sections, first the process engineering section which focuses on the design, steady-state and dynamic simulation of a Rectisol plant and second the control and real time optimization section. In this chapter we will briefly provide a background on these two subjects and describe the motivation and objectives behind our work. The structure of the thesis is also provided at the end of this chapter.

1.1 Motivation

Removal of acid gases, mainly CO_2 and sulphuric compounds such as H_2S and COS, is used for the purification of a tailgas, intermediate or final product gas stream. The purpose of this purification can be environmental and safety issues , operating constraints, or both. Acid gas removal processes are utilized in many industries such as the oil and gas, power plants and most recently in the gasification based biorefineries.
There are many reasons for removal of sour components in a gas, one is environmental regulations. For example in many regions there are regulations on the amount of carbon and sulfur in gas emissions of a power plant or chemical unit. Another reason can be the quality of a final product, for example if a plant is producing syn gas for combustion, the CO_2 is removed to increase the heating value of the gas and sulfur compounds have to be removed due to safety issues. Also another constraint is the requirement of downstream processes, for example the syn gas being sent to a Fischer Tropsch unit should not contain sulfur compounds to prevent their reaction with the catalyst and to protect the catalyst.
These processes consist of two main sections :

- The absorption section.
- The regeneration section.

1.1.1 Process configuration

The absorption section consists of one or two absorption columns where the gas is contacted with the lean solvent and impurities are absorbed. The regeneration section contains strippers and distillation columns where the rich solvent is stripped from the absorbed impurities. The configuration of the absorption and regeneration sections depends on the pressure, temperature and composition of the contaminated gas, the upstream and downstream processes and the use of the syngas, such as production of ammonia, methanol, or even combustion gas.

1.1.2 Absorption mechanisms

Depending on the absorption mechanism used for separation acid gas removal processes are divided into two groups :

- Chemical absorption.
- Physical absorption.

In chemical absorption the contaminants form a chemical bond with the solvent, while in physical absorption the contaminats are absorbed only based on their solubility in the solvent and no significant chemical reaction occurs. Examples of chemical absorption processes are Monoethanolamine (MEA), Diethanolamine (DEA), Methyl diethanolamine (MDEA), Diethyleneglycol (DEG) and Triethyleneglycol (TEG). As we see these solvent are all of a basic nature and thus react with the acid gas components to absorb it. Examples of physical absorption processes used for acid gas removal are Rectisol which uses methanol as solvent, Purisol with N-methyl-2-pyrrolidone (NMP) as solvent, Selexol with a mixture of dimethylether and polyethylene glycol. A third group of processes use a mixture of both physical and chemical solvents. Examples of these processes are Amisol and the Selefining process.

The process studied in this work is a physical absorption process named Rectisol. Compared to many similar processes, Rectisol is an economical and environmental friendly candidate for purification of gases produced by partial oxidation of carbon containing material. It is widely considered as part of many gasification based biorefinary schemes due to its design and operation flexibility, capability in removal of sulphuric compounds and CO_2 in ppm ranges, and its potential for energy integration. The advantages of Rectisol is that in this process the solvent does not foam in contact with the sour gas, the solvent is not corrosive and it can be easily regenerated by flashing at low pressures. But it also comes with a disadvantage which is the relatively high refrigeration energy requirement which leads to higher operating costs (Olajire, 2010).

Based on the area of application, Rectisol can have many configurations. If the absorption section consists only of one column, or in other words if the absorption of CO_2 and H_2S is done at the same time, the process is said to be single stage. But if the absorption of H_2S and CO_2 is done in two separate columns, the process is said to be two stage. Also if CO_2 and

H_2S are disposed of in the same stream the process is non-selective, and if CO_2 and sulfur compounds are disposed of in separate streams, the process is called selective (Ranke and Mohr, 1985).

The configuration we used in this work is two stage and selective. Figure 1.1 shows this configuration :

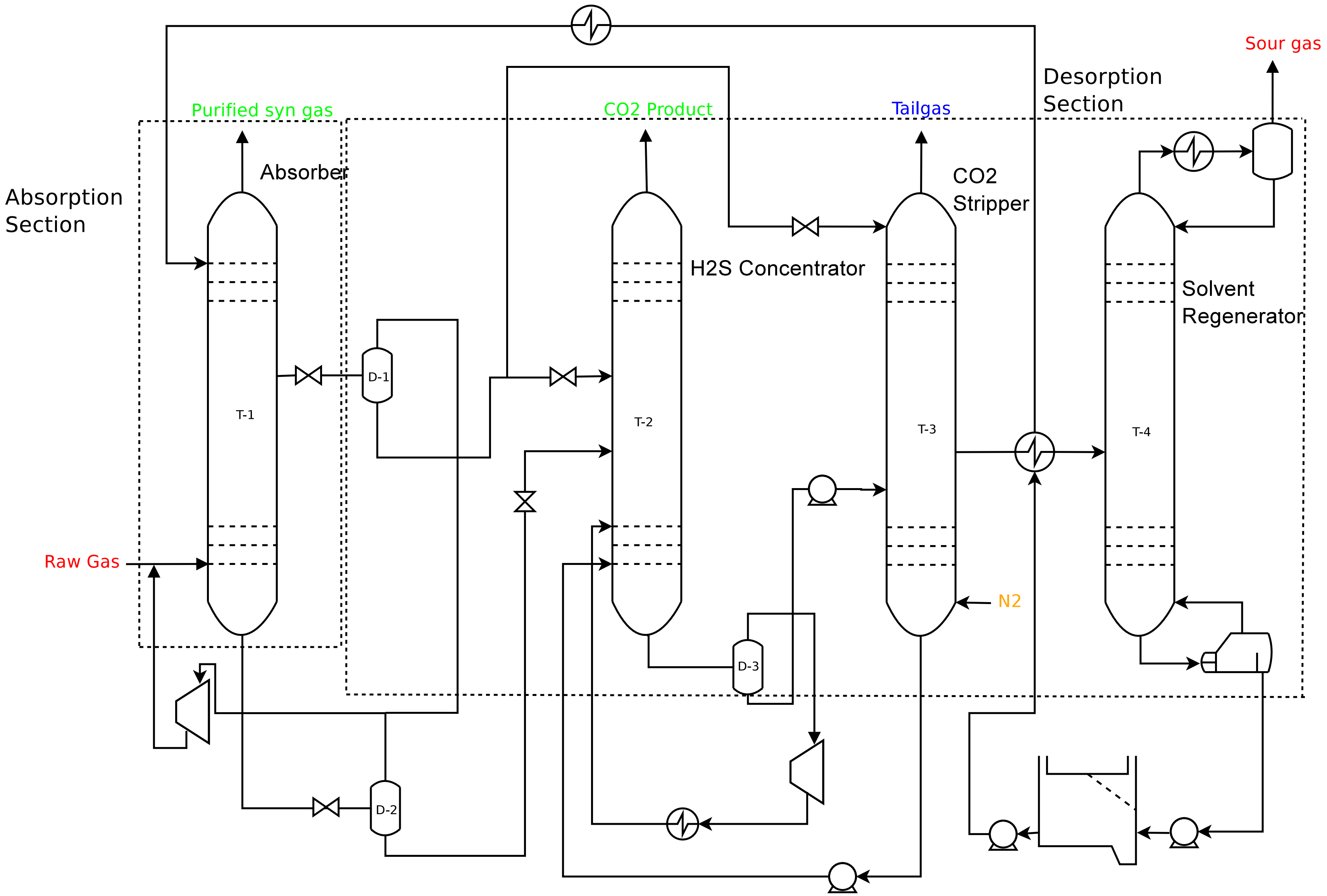


Figure 1.1 Rectisol configuration used

As we can see the syngas enters the bottom absorber (which actually consists of two columns on top of each other) where it is contacted with partially loaded solvent, mostly containing CO_2, to absorb the H_2S, and also CO_2 . The remaining amount of CO_2 is absorbed in the top section of the column in contact with the lean solvent. The solvent is withdrawn from the absorber at the end of each section, and flashed to remove the valuable components like H_2, CO and CH_4. While part of the liquids from the first section is sent to the third column, the remainder is sent to the second column. In the second column which is also called the "H_2S Concentrator" the rich solvent is heated up to a temperature where mainly CO_2 is stripped from the liquid. The gas product of the H_2S concentrator is 98 vol% CO_2. The pressure of the liquid product is dropped and it is flashed, the resulting gas stream is recycled back to column as stripping gas and the liquid phase is sent to the third column. The third column which is called the "CO_2 stripper" removes the remaining CO_2 from the rich solvent using a stream of pure nitrogen.The gas product of this column is approximately half CO_2 and half N_2. The liquid product is divided into two streams, while the smaller portion is recycled to the H_2S concentrator, the larger portion is sent the solvent regenerator. The solvent regenerator is a conventional distillation column, where the remaining sour components are removed from the solvent using distillation and the lean solvent is cooled and sent back to the absorber.

1.2 Problem definition

The problem assumed to be associated with the Rectisol process studied in this thesis is that the feed composition and flowrate to the Rectisol process may vary depending on the operating conditions of the gasifier, the biomass composition and many other parameters. In order for the process to react to these variations and keep the products as close as possible to the specified standards, adaptive extremum seeking control was used.

As seen in the previous section, the process is complex in terms of recycle streams. Such a process is multivariable and highly interactive and like many other chemical processes shows nonlinear behaviour. For the adaptive extremum seeking control to be functional and to reject measurable and unmeasurable disturbances and also for the process to adapt to new operating conditions a fast interactive regulatory control structure is required.

In this context a search for an implementable multivariable control structure was done and an adaptive extremum seeking scheme was designed that keeps the product specifications as close as possible to their standards.

1.3 Objectives

The general objective associated with this work is : **to implement adaptive extremum seeking control on the Rectisol process to optimize its performance in the presence of perturbations.** In this context the following specific objectives were declared to achieve our General objective :

– To develop a dynamic model for Rectisol

– To design and implement a plantwide regulatory control structure on the dynamic model

The hypothesis linked to our objectives is that adaptive extremum seeking can help improve the performance of the process especially in the presence of unmeasurable disturbances.

1.4 Thesis orginization

This thesis commences with a brief literature review (Chapter 2) on the Rectisol process and its simulation, a plantwide control solution to recycle processes, Linear MPC, Distributed Linear MPC and at the end adaptive extremum seeking control.

In Chapter 3 the methodological concepts used, will be described. The latter includes the development of a dynamic model in Aspen plus®Dynamics, design of a centralized and distributed MPC structure and design of a multi input adaptive extremum seeking scheme.

In chapter 4 the results of the regulatory control structure will be presented independently and compared using existing criteria and interpreted.Then the adaptive extremum seeking control layer will be implemented on some of these structures and their performance will be compared in terms of control and optimization.

In the final chapter (chapter 5) a conclusion will be made from the results presented in the previous chapter and the best regulation and optimization structure will be chosen.

CHAPTER 2

LITERATURE REVIEW

In this chapter we briefly review the literature on Rectisol, linear model predictive control, distributed MPC and adaptive extremum seeking.

2.1 Rectisol

Rectisol is known to be an economical process for acid gas removal of partial oxidation products(Weiss, 1988). As found in Ullmann's Encyclopedia of industrial chemistry, it has been cited by Ranke that the Rectisol process was at first invented by Lurgi in 1950 and later on further developed, in cooperation with Linde (Hiller *et al.*, 2000). In general it is used for the purification of partial oxidization gases, and has different configurations based on the purpose of its application. Ranke and Mohr (1985) classify different configurations of the process into two main classes, non-selective and selective. The selective systems have at least two sour gas products, one sulfur free CO_2 stream, and a sulfur stream which is fed to a Clause unit. The non-selective systems only have one sour gas stream containing both CO_2 and sulfur compounds. The standard Rectisol configuration is of the non selective type. Its flow diagram can be seen in figure 2.3. Ranke (1977) modified this flow diagram to create a selective system that has a high concentration CO_2 stream and a clause feed stream. Figure

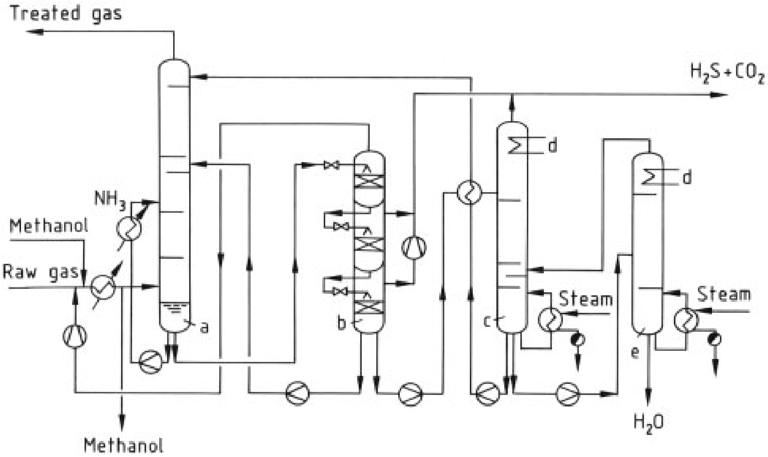


Figure 2.1 Standard Rectisol configuration (Hiller *et al.* (2000),p.96)

2.2 shows the proposed configuration.

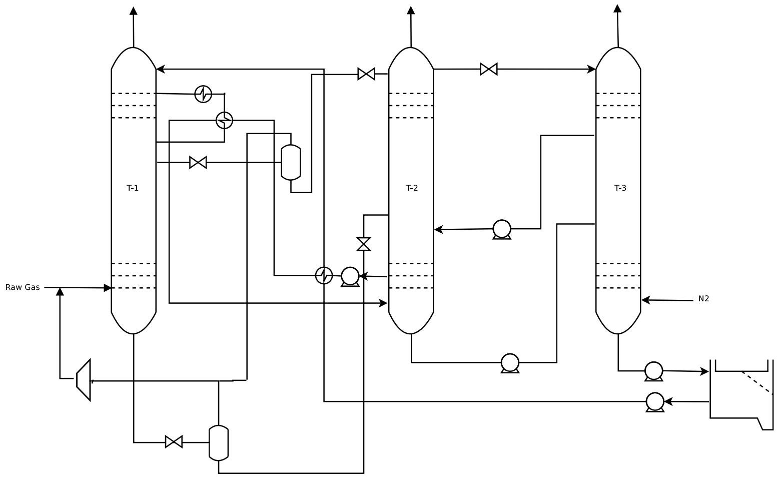


Figure 2.2 Selective Rectisol configuration (Ranke, 1977)

Ranke and Mohr (1985) have also compared the selective and non-selective configurations considering energy consumption and performance in different applications. They have also listed a few applications such as ammonia production and methanol production. They also studied the integration of different processes with Rectisol such as shift gas conversion, sulfur production and cryogenic separation.

The number of absorption columns can also be different, single stage Rectisol consists of only one wash column while two-stage Rectisol consists of two. Two-stage Rectisol is mostly used along with the shift conversion process in ammonia and methanol plants where the shift conversion is done between the two stages of the wash (Weiss, 1988).

Literature on simulation of Rectisol itself is very limited, and it has been mainly studied as part of another process and mainly in steady state. Preston (1981) has developed a steady state model of Rectisol using the Aspen® software. A non-selective configuration has been modeled. The Redlich-Kwong-Soave equation-of-state was found suitable and the coefficients were found using experimental data. The model was developed to obtain mass and energy balances and to predict the composition of clean product gas by varying different paramaters and operating conditions. The overall regenerated solvent recycle loop was not closed to obtain convergence.

As part of a Dynamic model for IGCC, only the absorption section of Rectisol was modelled using the Dymola software. Equilibrium stages where considered and it was assumed that the system obeys the ideal gas - Henry law. The steady state results of this simulation where validated by results from Aspen Plus and Chemcad (Heil *et al.*, 2009).

Many works has been done on dynamic modelling and simulation of other acid gas removal processes that are similar to Rectisol from a systems engineering point of view. The models are either created by combining first principle models of individual units or by using commercial software like Aspen®Plus Dynamics (Lin *et al.*, 2010; Harun *et al.*, 2012).

2.2 Plantwide control of processes with recycle

Design of a plantwide control procedure for cascaded unit operations without any recycle streams was developed over half a century ago and has been widely used in the industry ever since (Buckley, 1964). But when a process contains recycle streams these techniques might cause instabilities through what is called ”the snowball effect”. The dynamic behaviour of these systems have been studied and analyzed in detail by Luyben. In his work Luyben has used a Reactor/Separator example with a linear model (Luyben, 1993).

Later on Luyben (1994) proposes that in order to deal with the mass recycle loop , at least one of the flows in the in liquid recycle loop has to be flow controlled. The same concept can be applied on energy recycle loops. Once this is done conventional plantwide control procedures can be used. Tyreus and Luyben (1993) have also studied the snowball effect in a one reactor, two separators configuration with two recycle streams, where a second order reaction takes place in the reactor, and have once again reached the conclusion that the flow-rate should be fixed at-least in one stream of the recycle loop. It has been shown that fixing the flow rate in the recycle loop at the fresh feed inlet can be advantageous compared to other alternatives (Bildea and Dimian, 2003).

By analyzing Luyben’s structure we can see that it takes away the possibility of optimizing the plant’s production rate. We can also see that snowball effect can be dealt with by redoing the process design, in this case by increasing the reactor volume so that the larger volume of the reactor can damp the oscillation of the recycle loop . Larsson *et al.* (2003) pointed out that by defining an objective function and a set of active constraints we can develop a self optimizing control structure to regulate and optimize the plant’s performance. They later on point out that a MPC controller can explicitly handle the constraints. Seki and Naka (2008) have used Larsson’s self optimizing control structure as their regulatory layer and implemented a MPC as the supervisory layer to optimize the process economy.

The Tennessee Eastman process is another example of processes with a recycle stream. This process is nonlinear and open loop unstable. Amongst the first attempts to stabilize this process was the work of McAvoy and Ye (1994). They developed a multi loop PID structure using steady-state analysis, relative gain analysis, Niederlinski index and disturbance analysis. Ricker and Lee (1995) mention that Palavajjhala et al used PI controllers alongside with

Dynamic matrix control (DMC) to control this problem. The PI controllers give a partially open loop stable plant. Ricker formed a nonlinear model of the partially open loop stable plant and formed an MPC controller that linearizes the model at each time step. The Tennessee Eastman process is a benchmark for many control topics and a vast amount of literature can be found on this subject.

2.3 Linear Model Predictive Control

Roots of MPC can be traced back to Linear Quadratic Gaussian (LQG). In LQG for a linear time invariant discrete system in the form of :

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Gw(k) \\ y(k) &= Cx(k) + \xi(k) \end{aligned} \tag{2.1}$$

where $w(k)$ and $\xi(k)$ are the state disturbance and measurement noise, considered to be zero mean independent Gaussian noise, an objective function in the following form is formulated :

$$J = \sum_{i=1}^{\infty} x_i^T Q x_i + u_i^T R u_i \tag{2.2}$$

Q and R are tuneable weighting matrices. By replacing $u = -Kx(k)$ in equation 2.2 the objective function is minimised for K which is the gain. The optimization is done offline once, and the gain is implemented in the control loop. MPC is an extension of LQG in the sense that it controls the plant by optimizing a similar objective function, but the optimization is done at each time step for the current states of the system and it has a finite horizon (Qin and Badgwell, 2003).

In General MPC needs an internal model to generate the vector of predictions that represent the future dynamic behaviour of the plant. This model can be in the state space form, polynomial, or can even be a matrix of transfer functions. MPC also has an optimizer that tries to bring the vector of predictions generated by the internal model close to the reference trajectory by solving an optimization problem. This optimization problem can be constrained or unconstrained.

$$\min_{\Delta U} (Y - Y_{ref})^T Q (Y - Y_{ref}) + \Delta U^T R \Delta U \tag{2.3}$$

The first works on MPC were carried out by Cutler and Ramaker (1980). They used linear step response models and formed an unconstrained multivariable control algorithm which is called Dynamic Matrix Control. This algorithm is advantageous compared to a multiloop

PID structure since it considers the interactions between variables but it does not explicitly handle constraints. DMC is considered as first generation MPC. Garcia and Morshedi (1986) reformulated the DMC problem into a Quadratic programming problem that can find the plant's input by considering the constraints as part of the solution. This technique is called Quadratic Dynamic Matrix control (QDMC). QDMC is considered as second generation MPC.

Clarke *et al.* (1987) have used input-output models to find the predictions. They showed that as long as the input-output correlation is rich enough, the predictive controller formed by this model is able to control the system even in the presence of non-minimum phase behaviour, open loop instability or unknown dead-time. This technique is called Generalized Predictive Control. As promising as it seems, this technique cannot handle multivariable constrained systems well (Morari and Lee, 1999).

MPC has been formulated in the state space format (Morari, 1990). In this manner, many useful theories can be applied to it and it also facilitates the extension of MPC to more complex cases. State space MPC needs the value of the states to carry out the calculations. These values can either be measured from the plant (if possible) or be provided by a state estimator. Wang and Young (2006) have proposed a method where non-minimal state space models are formed using input output data or even using transfer functions. They also augmented the model with integrators to enable offset free tracking. Previously to handle modelling error and offset, at each sampling interval the error between the process output and the model prediction at that instant was calculated and was fedback to the controller as a constant disturbance over the prediction horizon (Constant Output Disturbance (COD)). Another idea is using a Kalman filter.

In the third generation MPC, new features are mainly use of state space models, an explicit description of disturbance models, the integration of a Kalman filter for state estimation and unmeasured disturbances and the introduction of soft and hard constraints to insure the feasibility of a solution by MPC. Examples of this generation are Shell multivariable optimizing control (SMOC), IDCOM-M by Setpoint Inc. and HIECON by Adersa . In terms of industrial application, MPC is pretty mature and has been applied in the industry for years. Aspen Technology Inc. and Honeywell are the two leaders in industrial MPC development. They have developed the fourth generation of MPC controllers which consider model uncertainty and enable multiple optimization levels (Qin and Badgwell, 2003). Table 2.1 shows a list of industrial linear MPC products.

Table 2.1 Industrial linear MPC products (Qin and Badgwell, 2003)

Company	Aspen Tech	Honeywell	Adersa	Invensys	SGS
Product	DMC-plus	RMPCT	PFC	Connois	SMOC
Model Type	FSR	ARX,TF	LSS,TF,ARX	ARX,FIR	LSS
Feedback	COD	COD	COD	COD	KF

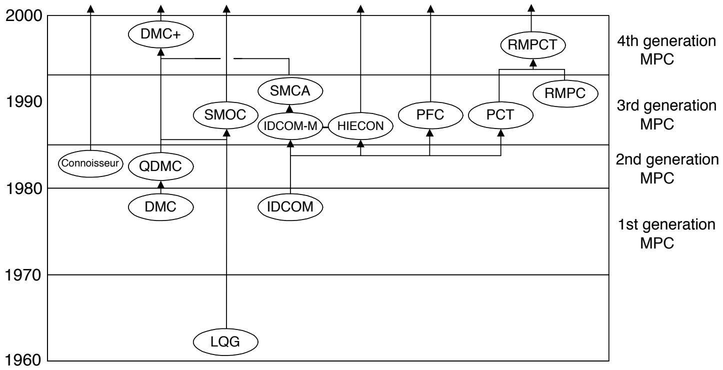


Figure 2.3 Evolution of industrial MPC technology (Qin and Badgwell, 2003)

2.4 Distributed Linear Model Predictive control

Centralized MPC control of a multivariable system comes with the of advantage of systematically considering all interactions between states and outputs. But if the system is large, the optimization problem can become too computationally demanding. Also the fact that the system relies on a single control agent can cause maintenance problems (Stewart *et al.*, 2010). A solution to this problem is decentralization of MPC. The model is decomposed into smaller subsystems and an optimization agent is assigned to each subsystem. The agents are completely independent of one another and there is no communication between them. The advantages that come with decentralization are easier implementation and modelling. The disadvantages are loss of performance in case of highly interactive systems and also in the presence of non-minimum phase transmission zeros (Cui and Jacobsen, 2002).

In general any type of communicating combination of MPC controllers can be seen as distributed MPC. Distributed MPC not only offers the flexibility and ease of implementation of decentralized MPC, but improves its performance by creating communication amongst the agents(Scattolini, 2009). The communication structure can be formed based on the topology of the plant. It is suggested that subsystems that interact with each other must have a communication link. In the case of chemical plants, if there is no recycle flow, it is just the neighbouring units that communicate, in the presence of a recycle flow all units inside the recycle loop must be linked (Rawlings and Stewart, 2008).

Depending on the number of information exchanges done in a sampling interval distributed MPC is divided into two groups :

- Non-iterative or independent, if information exchange is only done once during the sampling intervals.
- Iterative, if information exchange is done more than once during the sampling intervals.

Iterative algorithms can be more beneficial in the sense that the amount of information exchanged amongst local controllers is large (Scattolini, 2009). Another classification is based on the objective function used in the local controllers (Christofides *et al.*, 2013) :

- If all local controllers work together to optimize a global cost function the DMPC structure is called Cooperative.
- if each local controller solves it's own objective function, which is independent of the others it is called Non-cooperative.

An iterative, cooperating method is said to converge closely to the solution of a centralized method, what is called ”Pareto optimal solution” in game theory, whereas in non-iterative and independent algorithms, local controllers tend towards the ”Nash equilibrium” which may be unstable (Scattolini, 2009). In order to insure stability of non-iterative, independent algorithms for linear discrete systems, Camponogara *et al.* (2002) have added a constraint. They have studied the application of their proposed method to load frequency control of power systems. Alessio and Bemporad (2008) have also added a stability condition for when the communication between local controllers fails.

Stewart *et al.* (2010) have developed an iterative-cooperative DMPC algorithm for linear systems with decoupled or weakly coupled constraints. Venkat *et al.* (2005) have studied the stability and optimality of iterative-cooperative DMPC. They have also added a modification which insures that all intermediate iterates are stable. Mercangöz and Doyle (2007) have developed an iterative DMPC framework and applied it to the four tank system to control the level of the two bottom tanks. They compared their DMPC to centralized and completely decentralized controllers. The results showed that the DMPC's performance is significantly better than the decentralized controllers and very close to that of a centralized controller. Venkat *et al.* (2008) applied cooperative DMPC to automatic generation control. Our DMPC Structure is similar to that of Mercangöz and Doyle (2007), so it's non-cooperative leading to a Nash equilibrium despite its iterative nature.

2.5 Adaptive extremum seeking control and Realtime optimization

Adaptive extremum seeking is one of the subjects of adaptive control, where an input which optimizes an output is found, while the only knowledge required is the existence of an extremum. Tan *et al.* (2010) mention that adaptive extremum seeking was primarily presented by Leblanc to maximise the power transfer from an overhead electric transmission device to a tram car. Luxat and Lees (1971) have studied the stability of an adaptive extremum scheme using Lyapunov's second stability law. Wang and Krstic (2000) also studied the stability of classic extremum seeking and applied it to minimize limit cycle behaviour in the Van der Pol oscillator. Krstić (2000) has proposed adding a dynamic compensator to the integrator that improves the speed of the overall extremum seeking by accounting for the plant dynamics. Ariyur and Krstic (2002) have created a design algorithm based on common LTI control techniques and stability. They have extended their design procedure to multivariable systems.

Many applications of adaptive extremum seeking have been reported in the literature. Wang *et al.* (2000) have used adaptive extremum seeking to maximize pressure rise in an axial flow compressor with uncertain compressor characteristics. Schneider *et al.* (2000) have used adaptive extremum seeking to tune the controllers that stabilizes the thermoacoustic instabilities in combustion processes. Banaszuk *et al.* (2000) have used this technique to tune the phase shifting controller that is part of a control structure that reduces acoustic pressure oscillations in gas-turbine engines.

Nguang and Chen (2000) took advantage of the model free concept of adaptive extremum seeking and implemented it on a continuos fermentation process were the extremum seeking controller optimizes the biomass production rate using the feed substrate rate. Wang *et al.* (1999) have also used the model-free (steady-state) form of extremum seeking to maximise biomass production rate in bioreactors with Haldane and Monod kinetic models, two different non-linear models.

Other forms of extremum seeking have also been formed that require an explicit structure information of the objective function. (Zhang *et al.*, 2003; Titica *et al.*, 2003; Marcos *et al.*, 2004b,a). Zhang *et al.* (2003) claims that this approach insures global stability for non-linear systems since it's based on Lyapunov's theorem whereas the approach utilized by Wang *et al.* (1999) can only assure global stability for linear systems and in the non-linear case, it can only guarantee local stability. A parameter estimation algorithm was developed for the unknown parameters. The proposed approach was applied to a continuous stirred tank bioreactor with Monod's kinetic model with unknown parameters and was shown that as long as the dither signal respects a certain persistent excitation criteria the convergence is guaranteed.

Marcos *et al.* (2004b) have applied a similar technique with the difference that they used the Haldane's Kinetic model with unknown parameters, which has unstable steady states. The approaches mentioned up to here are all based on state feedback. An output feedback alternative was proposed and applied to a continuous stirred tank bioreactor with Monod's Kinetic model (Marcos *et al.*, 2004a). This methodology has also been applied to fed batch bioreactors with both of the kinetic models (Titica *et al.*, 2003; Cougnon *et al.*, 2011).

The principles of the methodology proposed by Zhang *et al.* (2003) have been applied to a non-isothermal continuous stirred tank where the Van de Vusse reaction occurs, to maximize the concentration of a product by manipulating the rate of heating and cooling (Guay *et al.*, 2005). The obtained extremum seeking structure was later on used to maximize an objective function of the reactor outlet concentrations through changing the jacket temperature, in a non-isothermal tubular reactor with the Williams-otto reaction, where the system is considered as a Distributed Parameter System (DPS). The kinetics were assumed unknown but a certain level of the systems structure information was used (Hudon *et al.*, 2005). Hudon *et al.* (2008) have extended this scheme to a case where input constraints are also considered in the optimization problem.

CHAPTER 3

Methodology

The methodology used in this thesis may be divided into three sections :

1. Development of a dynamic model for Rectisol
2. Design of a plantwide control strategy for disturbance rejection and setpoint tracking
3. Implementation of an Adaptive extremum seeking for realtime optimization

The steps taken in each section will be discussed in this chapter.

3.1 Development of a dynamic model

In order to create a dynamic model the Aspen Plus Dynamics software by AspenTech®was used, so a steady state model had to be developed in Aspen plus, due to the fact that the creation of a dynamic model in this manner is more simple. Later on the steady state model was modified and was exported to Aspen dynamics. This dynamic model will be treated in a blackbox manner.

3.1.1 Development of the steady state model

To create a steady state model, the steady state data and process flow diagram found in Larson *et al.* (2006) was used. Amongst the different process layouts introduced for Rectisol in this report, the more general case where both CO_2 and H_2S are removed was considered. In this layout the syngas enters the bottom of the column C1 where it is contacted with chilled methanol at a temperature of $-60°C$, there's a side-stream of methanol that is flashed and sent to C2 and C3. The bottom methanol stream is flashed and sent to C2 where it is heated just enough to strip a mostly CO_2 stream from it. The bottom liquid stream is flashed, and the gas released is recycled back to C2, the liquid stream is sent to C3 where the remaining CO_2 is stripped using a N_2 flow. It is then passed to C4 where using distillation the remaining CO_2 and H_2S are separated.

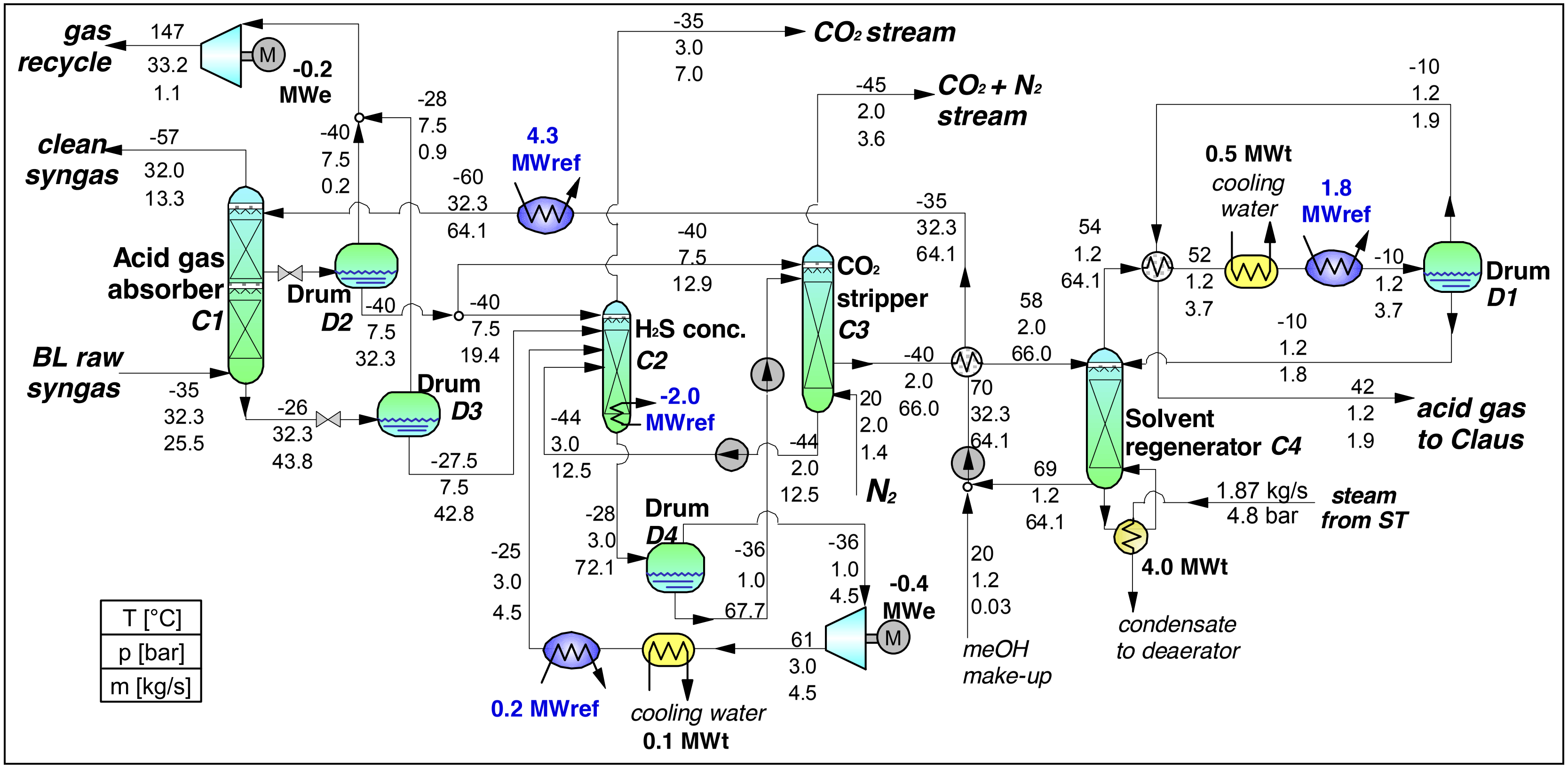


Figure 3.1 Process flow diagram of total CO_2 and H_2S removal Rectisol (Larson *et al.* (2006),p.43)

This layout was slightly modified by adding a compressor and a heat exchanger to the feed stream and by replacing the heat exchanger between columns C3 and C4 with a heater and cooler. Also valves are added to the flow diagram. All valves besides those indicated in red are control valves. The red valves are valves used to decrease pressure along the stream.

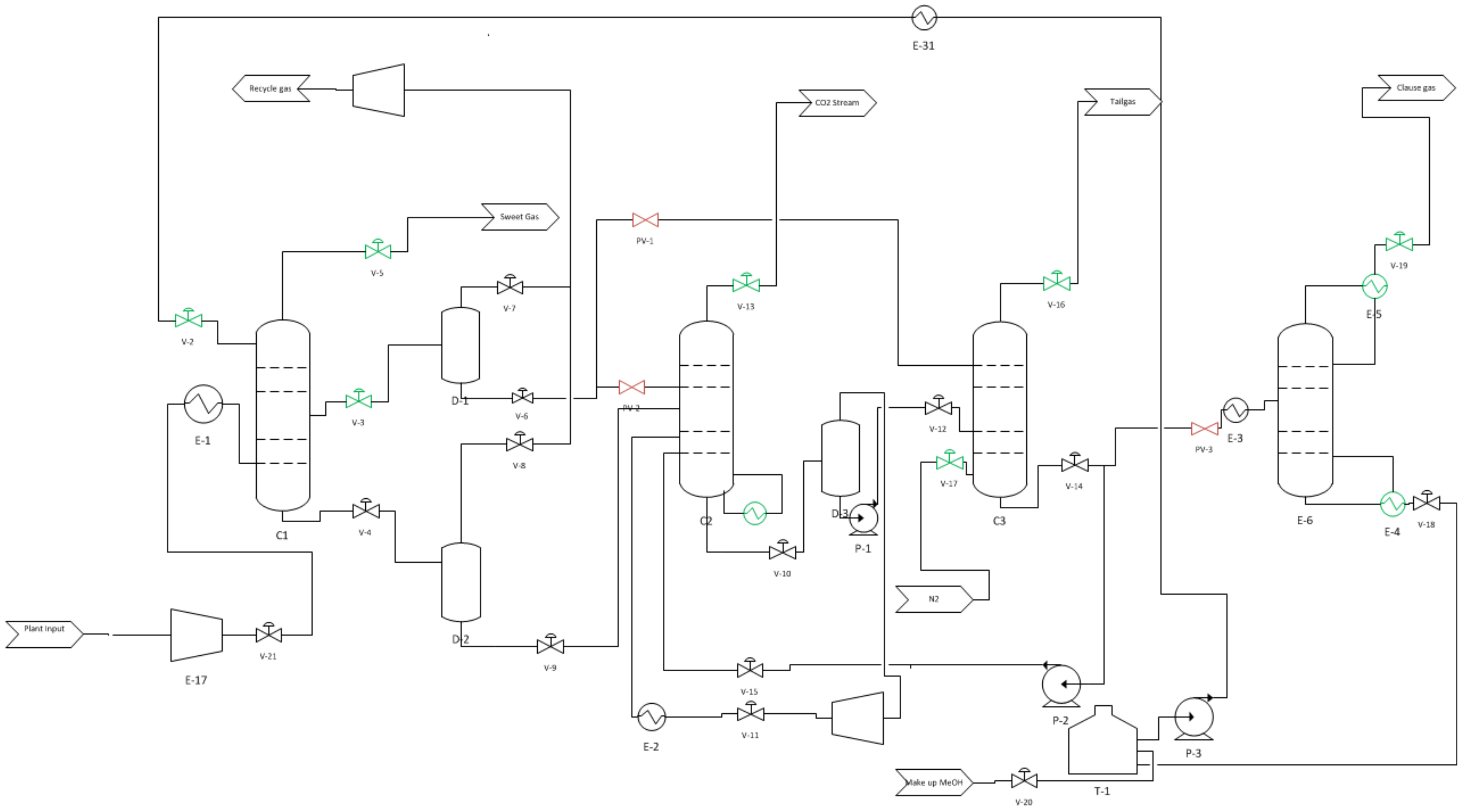


Figure 3.2 Process flow diagram of the Rectisol used

In order to carry out the steady state simulation, the Peng-Robinson thermodynamic package was used. This package was also used by Aspen in their Integrated Gasification combined Cycle (IGCC) example (AspenTech, July 2008), where the Rectisol process is used for acid gas removal. The feed composition was taken from Larson *et al.* (2006), which can be seen in the table 3.1.

Table 3.1 Feed Composition

Component	Molar Fraction
Ar	0.0101
CH_4	0.0206
CH_4O	0
CO	0.3609
CO_2	0.2095
COS	0.0002
H_2	0.3757
H_2S	0.0193
N_2	0.0037

The columns were based on the RadFrac (Equilibrium stages) model of Aspen since RateFrac (Rate based) cannot be used in dynamic mode. The absorber (C1) design was based also on information found in the Aspen Plus IGCC example. The number of stages was set to 30, and the Sum-rates algorithm was chosen as the convergence algorithm. Also a liquid side-stream was set at the 10th stage. The results were obtained from the absorber and were verified with Larson *et al.* (2006)'s results by comparing temperature, pressure, and compositions. It must be noted that the composition of the lean solvent stream was considered to be pure methanol as an initial guess. The valves and the flash drums were added with their relevant operational information.

In the next step the H_2S concentrator (C2) was added. The number of stages was considered to be 10 and the standard convergence method was used. As seen in Figure 3.2 the liquid stream leaving D-1 is split into two streams and one of these streams is sent to C2. This stream enters the column on the 2nd stage. As of the liquid stream leaving D2 , it enters the column on the 9th stage. As we can see in Figure 3.2, there are also two recycled streams, one coming from D3 and one coming from C3. These streams enter the column on stages 6 and 9, respectively. It must be noted that these stages were chosen in a way to have the smoothest composition and temperature profile in the columns. Since the composition of these streams were at first unknown, pure methanol was considered as an initial guess. After adding drum D3 and the compressor and cooler a next estimate could be found for the guessed composition of the respected stream. After a couple of trials and errors the cooler can be directly connected to the column.

Next, the CO_2 stripper was configured. The number of stages were set to 10, and the streams coming from drum D1 and D3 were introduced on stages 2 and 1, respectively. Also a stream of pure nitrogen was introduced at the bottom of the column. The stream leaving the bottom of C3 is split into two and is sent to C2. The same trial and error procedure, described above was applied.

The last column is the solvent regenerator (C4). This column was also considered to have 10 stages and the standard algorithm was used for convergence. The feed stream to column, coming from C3 was introduced on stage 7. The reboiler and condenser duties were set to values found in Larson *et al.* (2006) and slightly modified so that the temperature, pressure and flow rates match.

The total recycle loop was not closed because, firstly, the Aspen plus solvers are not able to solve the large problem created and by replacing the composition of the liquid inlet with the bottom composition of C4 several times we can get an adequately exact approximation. Secondly because an easier alternative is dynamic simulation which better serves our purpose (Luyben, 2006).

The finalized steady state model was analyzed. As mentioned absorption of H_2S ends in the bottom section of the absorber while the absorption of CO_2 ends in the top section (Figure 3.3). The evolution of CO_2 concentration in the H_2S concentrator was also studied, and as we see in Figure 3.4, in general the concentration of CO_2 in the gas phase increases from bottom to top showing that CO_2 is being removed from the liquid phase, leading to a higher concentration of H_2S.

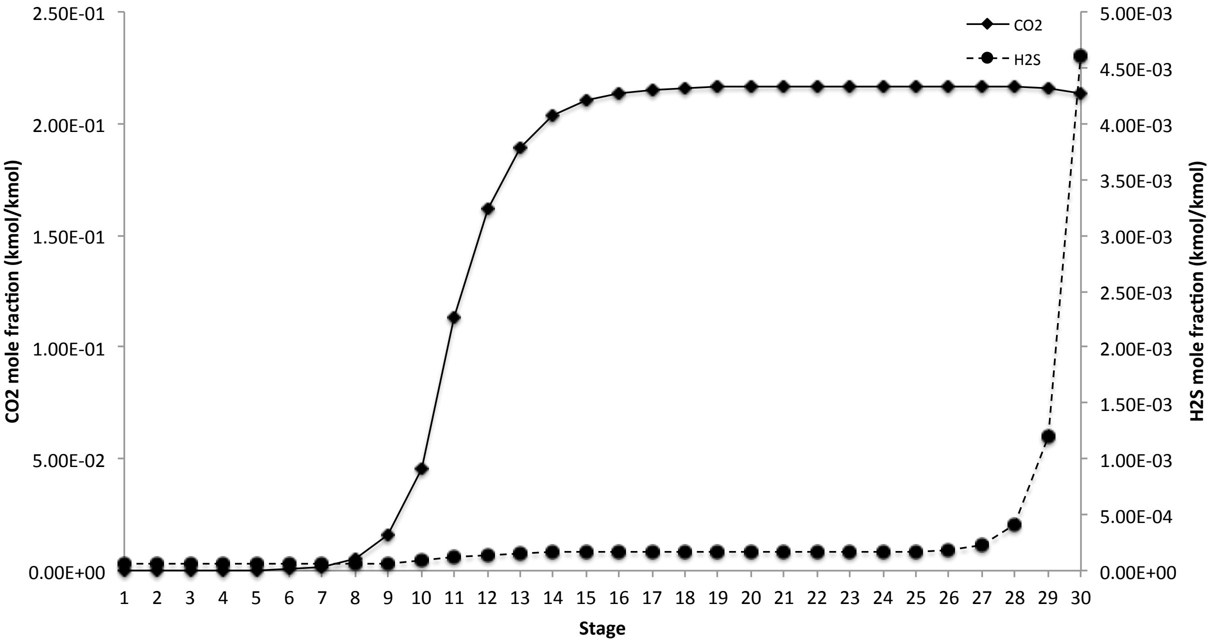


Figure 3.3 CO_2 and H_2S mole fraction profiles (gas phase) in the absorber

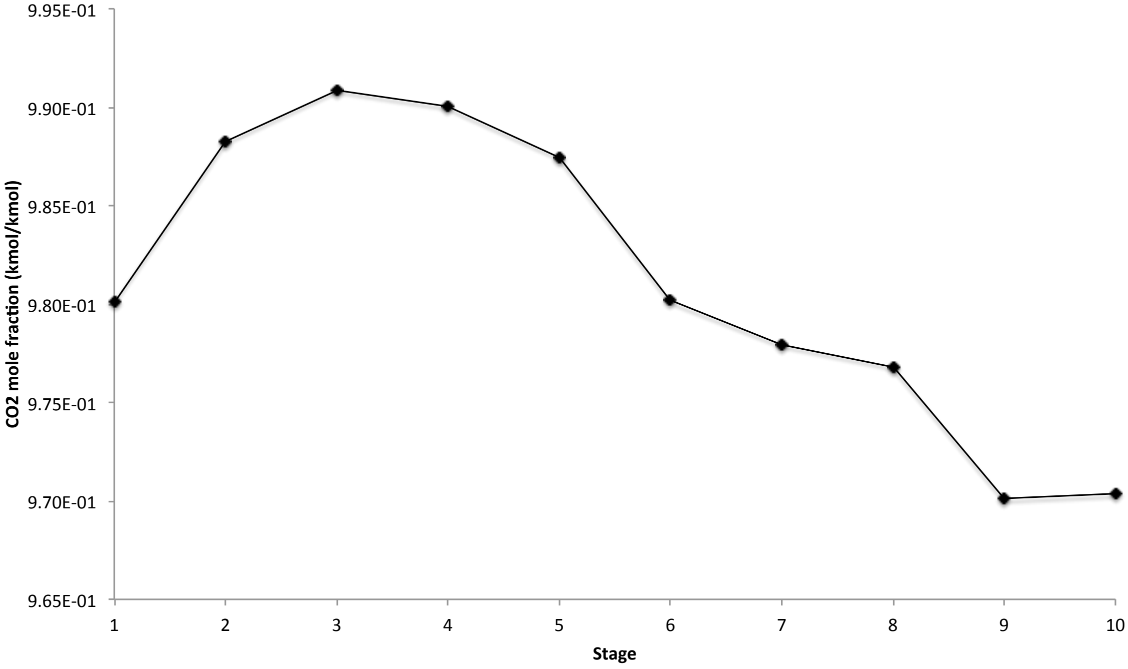


Figure 3.4 CO_2 mole fraction profile (gas phase) in the H_2S concentrator

The CO_2 concentration profile in the gas phase for the CO_2 stripper (Figure 3.5) shows that from bottom to top the amount of CO_2 in the phase increases, so it's concentration in the liquid phase decreases.

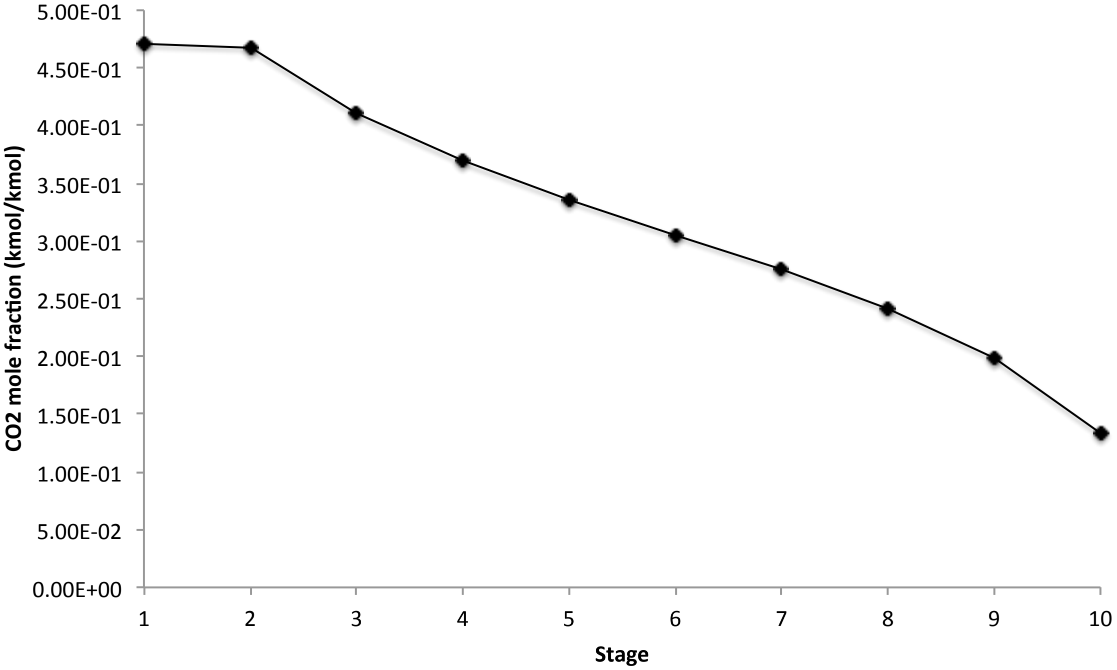


Figure 3.5 CO_2 mole fraction profile (gas phase) in the CO_2 stripper

Figure 3.6 shows the H_2S and CO_2 profile in the gas phase of the solvent regenerator.

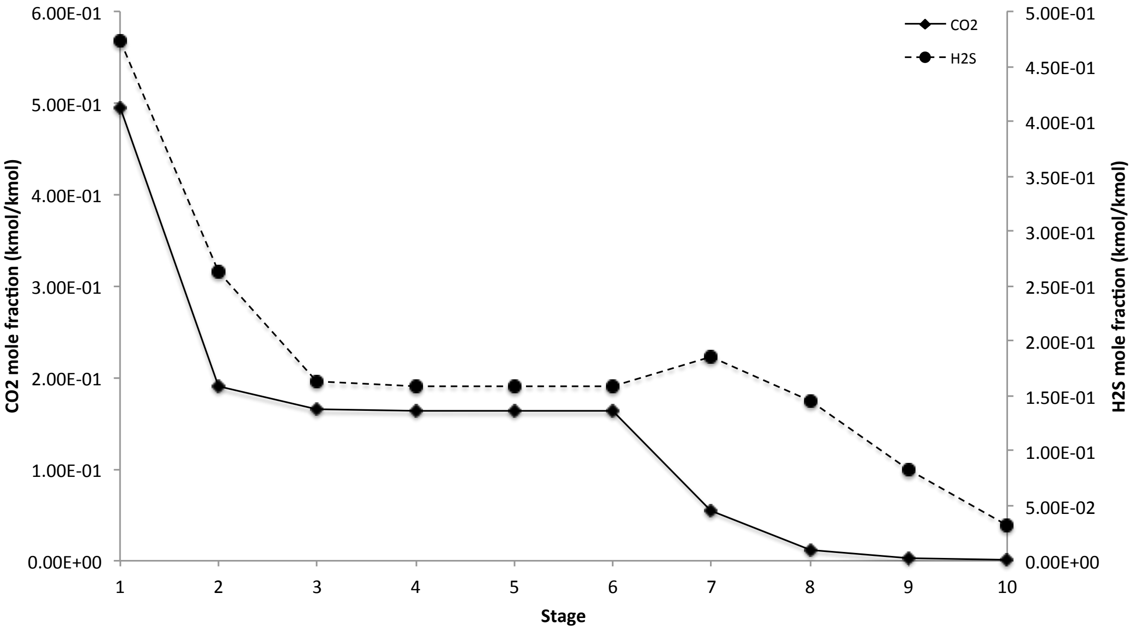


Figure 3.6 CO_2 and H_2S mole fraction profiles (gas phase) in the solvent regenerator

After obtaining a steady state model we must prepare it for the transition to a dynamic model. In order to do so all the vessels (drums and column sumps) must be sized and column pressure drops must be assigned.The vessel dimensions can be found in table 3.2 and the pressure drops in table 3.3. The model is then pressure driven in Aspen Plus to verify if there are any inconsistencies in pressures and exported to Aspen Dynamics environment.

Table 3.2 Vessel dimensions

Vessel	Diameter(m)	Length(m)
C1-Sump	2.35	5.2875
C2-Sump	2	5
C3-Sump	2	5
C4-Sump	2	4
C4-condenser	0.705	2.058
D1	2.35	5.2875
D2	2.82	6.34
D3	6.17	30.85
T1	9.135	45.675

Table 3.3 Pressure drop per tray

Vessel	ΔP (bar)
C1	0.016
C2	0.017
C3	0.017
C4	0.015

3.1.2 Forming the dynamic platform

The dynamic model consists of material and balances energy balances that are coupled by thermodynamic correlations. For example if we consider the flash drum D-1, molar balances can be written for each component :

$$\frac{dn_i}{dt} = Fz_i - Vy_i - Lx_i \tag{3.1}$$

where n is the number of moles, F the molar feed flow-rate, V and L the vapor and liquid molar flow-rates and and x_i, y_i and z_i are the corresponding mole fractions. x_i and y_i are related using the K-value which is a function of the temperature :

$$K(T) = \frac{y_i}{x_i} \tag{3.2}$$

The energy balance for this equipment is :

$$\frac{dH}{dt} = Fh_V - Vh_V - Lh_L + Q \tag{3.3}$$

where H is the accumulated enthalpy and h_F, h_V and h_L are the enthalpies of the feed, vapor and liquid streams. Q is the heat duty sent to the equipment. The enthalpies are calculated as a function of temperature using thermodynamic correlations. The combination of these equations gives a Differential Algebraic Equation system (DAE). The same principles are used for other equipments, meaning mass balances, an energy balance and thermodynamic equations.

The exported model from Aspen Plus already contains some pressure and level controllers. In fact when exporting the model pressure and level controllers were added to vessels as long as there is a valve directly on the gas or liquid stream. But if the model is run without adding any further controllers it well diverge since the level on D3 is not controlled. In order to develop a simple plant wide control configuration of PI controllers the procedure presented by Luyben was used (Luyben *et al.*, 1997; Luyben, 1993). The pressures and levels of all vessels were controlled. Also the output temperature of all heaters, coolers and flash drums were controlled as well the recycle flow-rate of the stream from C3 to C2. The remaining actuators (degrees of freedom) were used to control the temperatures on the stages of the columns. In order to choose what stage temperature to control the manipulated variables are subjected to a step change at steady state for each column and a matrix of steady gains for each tray temperature and the manipulated variables is formed. Using Singular Value Decomposition (SVD) the matrix is decomposed and the tray with the highest singular value is chosen as a controlled variable (Moore, 1993). A list of all the controllers used is provided in Table 3.4 and a PFD showing these controllers can be found in Figure 3.7. Something that is worth noting is that the method used for integration here was the Gear method with a relative tolerance of 0.0005.

Table 3.4 List of PID controllers

Controller tag	Process variable	Manipulated variable
PC-1	Top Pressure of C1	% opening of V-5
PC-2	Top Pressure of D1	% opening of V-7
PC-3	Top Pressure of D2	% opening of V-8
PC-4	Top Pressure of C2	% opening of V-13
PC-5	Top Pressure of D3	% opening of V-11
PC-6	Top Pressure of C3	% opening of V-16
PC-7	Top Pressure of C4	% opening of V-19
LC-1	level of C1-Sump	% opening of V-4
LC-2	Level of D1	% opening of V-6
LC-3	Level of D2	% opening of V-9
LC-4	Level of C2-Sump	% opening of V-10
LC-5	Level of D3	% opening of V-12
LC-6	Level of C3-Sump	% opening of V-14
LC-7	Level of C4-Sump	% opening of V-18
LC-8	Level of C4-Cond	Recycle flow rate (Kg/hr)
LC-9	Level of T1	% opening of V-20
TC-1	Temperature of 9th stage of C1	% opening of V-2
TC-2	Temperature of 11th stage of C1	% opening of V-3
TC-3	Temperature of D1	D1 Duty (GJ/hr)
TC-4	Temperature of D2	D2 Duty (GJ/hr)
TC-5	Temperature of 10th stage of C2	C2 reboiler Duty (GJ/hr)
TC-6	Output Temperature of E-2	E-2 Duty (GJ/hr)
TC-7	Temperature of 8th stage of C3	% opening of V-17
TC-8	Temperature of E-3	E-3 Duty (GJ/hr)
TC-9	Temperature of 2nd stage of C4	C4-cond Duty (GJ/hr)
TC-10	Temperature of 5th stage of C4	C4-reboiler Duty (GJ/hr)
TC-11	Output Temperature of E-31	E-31 Duty (GJ/hr)
FC-1	Mass flow rate of Recycle stream from C3 to C2	% opening of V-15

This model with PID controllers was used as our base case, and all the other control strategies developed later on were compared to it. In the PID case controllers were tuned using lambda-tuning, and in the other cases the PIDs were tuned using Ziegler-Nichols. It must be noted that the controllers PC-1, TC-1, TC-2, PC-4, TC-5, PC-6, TC-7, PC-7, TC-9, TC-10 were removed from the aspen dynamics file and created in a Simulink model that can communicate with Aspen dynamics using the Aspen Modeler block provided by AspenTech®. This was done so that all our controllers will use a unique file as their plant.

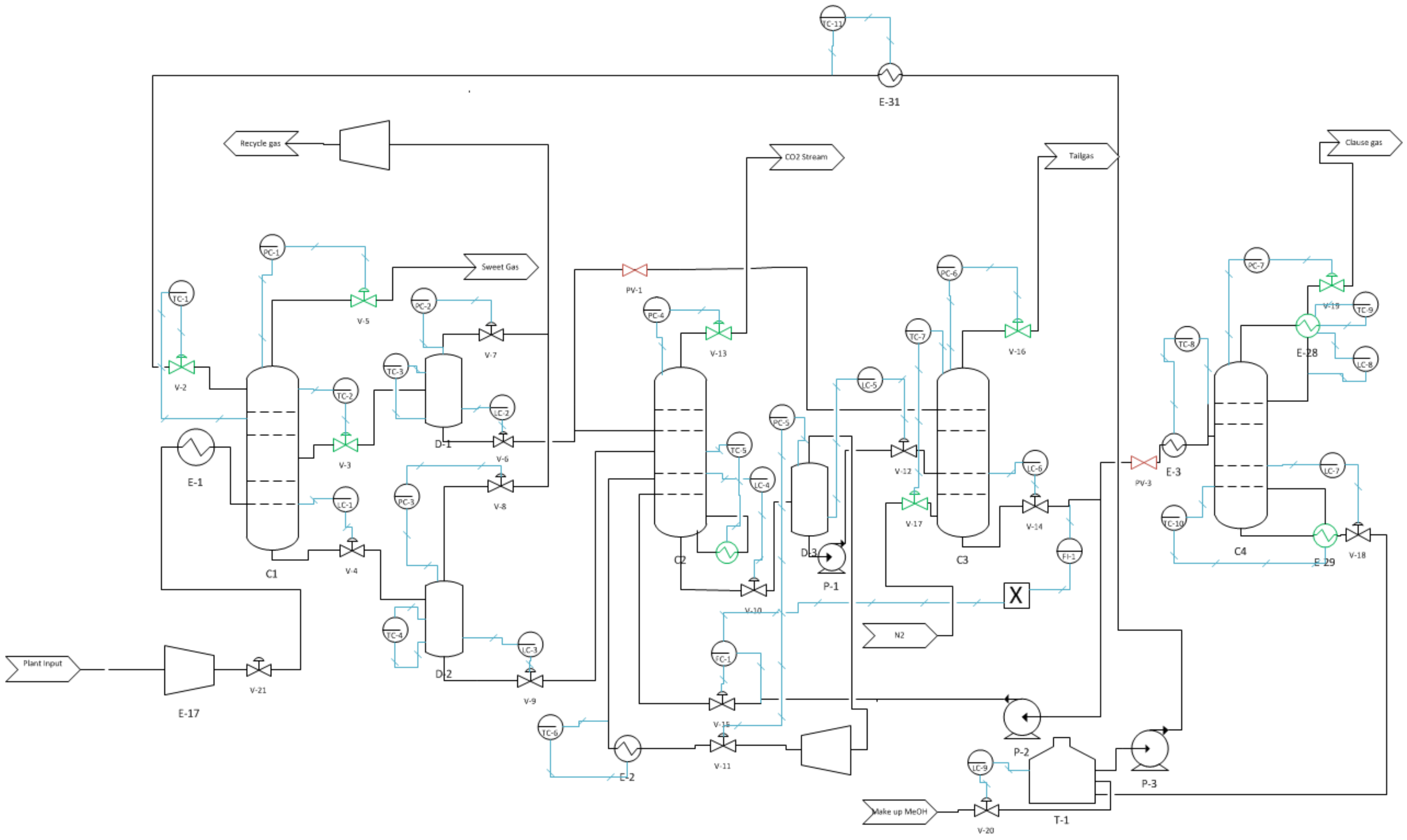


Figure 3.7 Process flow diagram of the rectisol with PID controllers

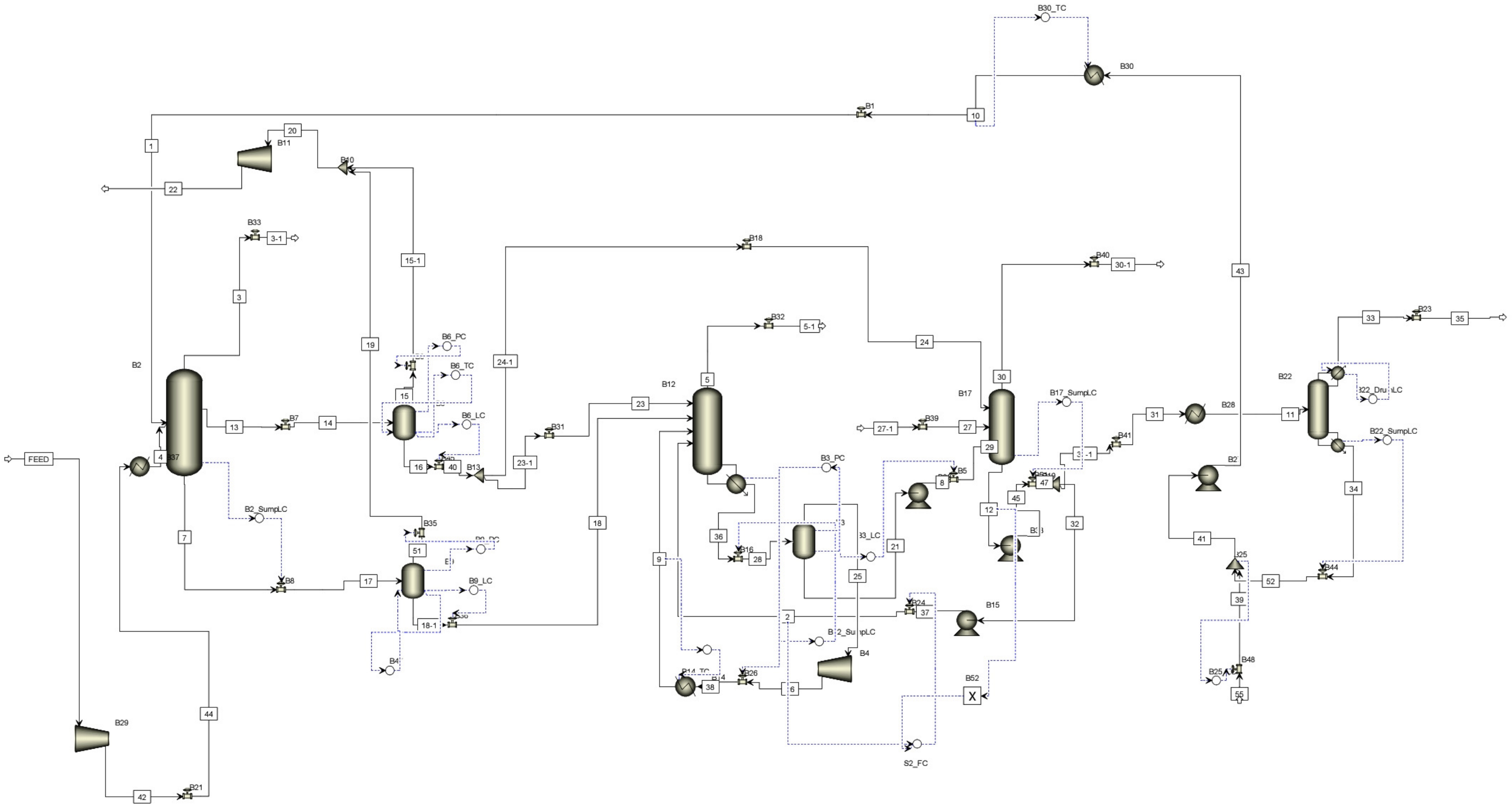


Figure 3.8 Schematic of the Aspen dynamics flowsheet

3.2 Identification of a linear model for MPC

To design a MPC controller , a discrete linear model of the plant is required. This model has to be able to find the dynamic relations between the considered inputs and outputs of the system. It must be noted that not all the process variables are considered in our MPC controller. Only those related to the controllers mentioned in the previous section are considered.

The inputs are subjected to a series of step changes. As can be seen in Figure 3.9 while one input is being manipulated the rest are kept constant. Later on the System Identification toolbox of Matlab® was used to find the needed model (Ljung, 1999).

Direct identification of a Multi Input- Multi Output (MIMO) is very time consuming and in the case where the number of inputs and outputs are slightly high it may be infeasible. So instead an indirect method was used. This means that multiple Multi Input- Single Output (MISO) models will be identified that map all inputs to each output, and then the MISO models were combined to form a MIMO model of the process.

There are many criteria used to determine the precision of an identified model, the one used in our work is Normalized Root Mean Square Error (NRMSE). NRMSE is calculated through Equation 3.4 .

$$NRMSE(y_i) = (1 - \frac{||\hat{y}_i - y_i||}{||y_i - mean(y_i)||})100 \tag{3.4}$$

where $\hat{y}_i$ corresponds to the simulated value of i^{th} output (from the identified model) and y_i the actual value of i^{th} output.

This identification method can lead to a composite model that can be helpful in forming the distributed model predictive control framework (Venkat *et al.*, 2007).

3.2.1 Identification of dynamic models

The data seen in Figure 3.9 were fed to the system identification toolbox as time domain data. The data series had a sampling time of 0.1 seconds. Another important point about the identification data is that all the data where subtracted from the values at the nominal point and the data used for identification was $\tilde{y}$ and $\tilde{u}$, where $\tilde{y} = y - \bar{y}$, $\tilde{u} = u - \bar{u}$. $\bar{y}$ and $\bar{u}$ are the nominal values.

Table 3.5 Inputs and Outputs of model

Output Number	Corresponding PV	Input Number	Corresponding MV
y_1	Top Pressure of C1	u_1	% opening of V-5
y_2	Temperature of 9th stage of C1	u_2	% opening of V-2
y_3	Temperature of 11th stage of C1	u_3	% opening of V-3
y_4	Top Pressure of C2	u_4	% opening of V-13
y_5	Temperature of 10th stage of C2	u_5	C2 reboiler Duty (GJ/hr)
y_6	Top Pressure of C3	u_6	% opening of V-16
y_7	Temperature of 7th stage of C3	u_7	% opening of V-17
y_8	Top Pressure of C4	u_8	% opening of V-19
y_9	Temperature of 2nd stage of C4	u_9	C4-cond Duty (GJ/hr)
y_{10}	Temperature of 5th stage of C4	u_{10}	C4-reboiler Duty (GJ/hr)

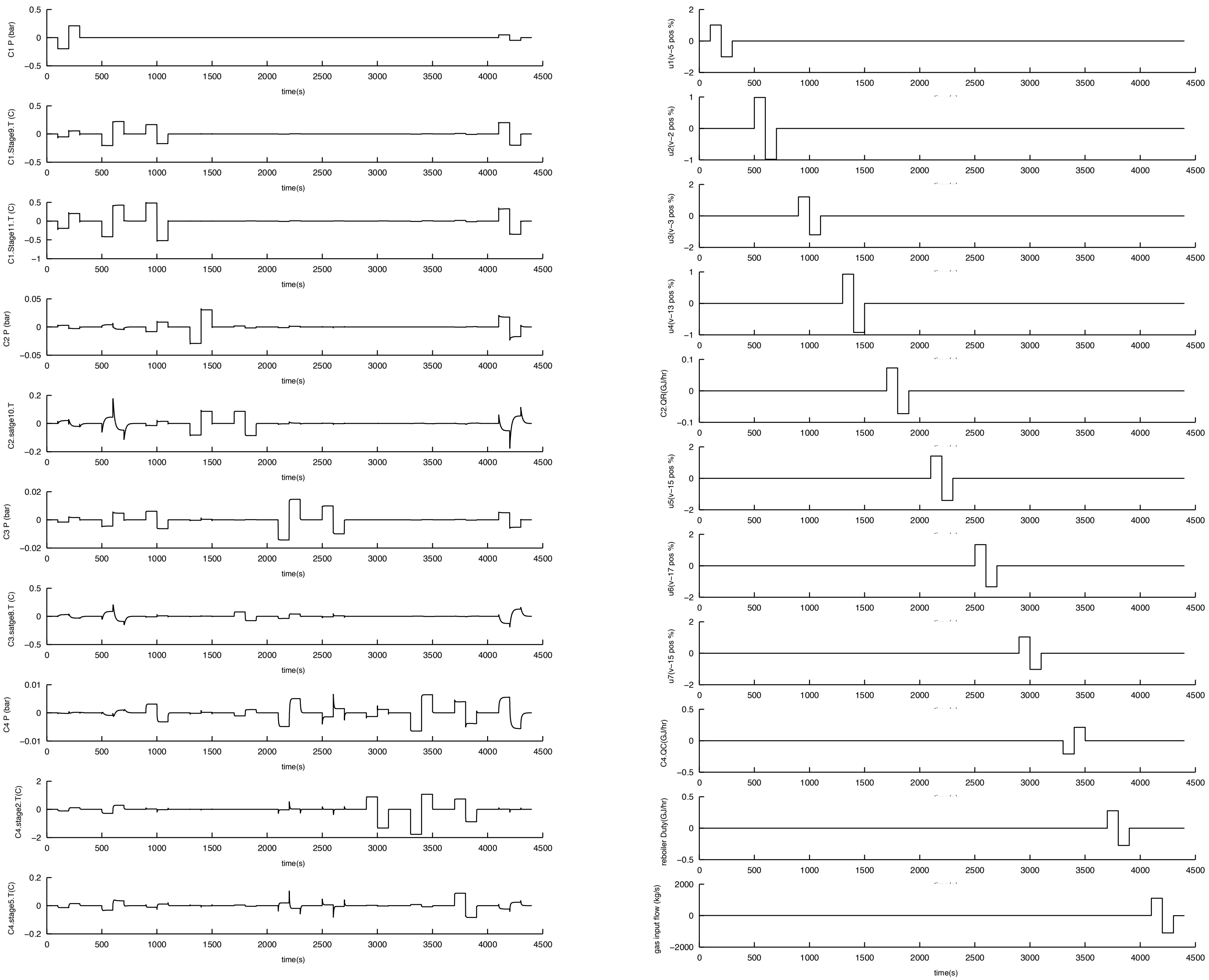


Figure 3.9 Identification signals

The identified models have the state space form, and were found using the N4SID (subspace identification) method (Ljung, 1999). These algorithms use QR and singular value decompositions ; therefore they have guaranteed convergence and numerical stability (Van Overschee and De Moor, 1996). The models have unit input delay and the N4weight and N4Horizon are chosen automatically. To identify the MISO models the entire input data and the data related to each output were imported to the toolbox. The order of the state space models are determined using the order selection tool, which plots the logarithm of the singular value versus the order of the model, the order after which the singular value decreases drastically is considered as the best choice. The model structure we're looking for is :

$$\begin{aligned} x_{mi}(k+1) &= A_{mi}x_{mi}(k) + B_{mi}u(k) + G_{mi}d(k) \\ y_{mi}(k) &= C_{mi}x_{mi}(k) \end{aligned} \tag{3.5}$$

where i represents the number of outputs and the corresponding model, x_i the states of the i^{th} model, d the measurable disturbance, A_i, B_i and C_i the system matrixes and G_i the measurable disturbance matrix. A model with this format can not be identified using the System Identification toolbox, so the model is modified to the following form :

$$\begin{aligned} x_{mi}(k+1) &= A_{mi}x_{mi}(k) + \begin{bmatrix} B_{mi} & G_{mi} \end{bmatrix} \begin{bmatrix} u(k) \\ d(k) \end{bmatrix} \\ y_{mi}(k) &= C_{mi}x_{mi}(k) \end{aligned} \tag{3.6}$$

This way the System Identification toolbox can be used to find the models.Table 3.6 shows the orders of the MISO models, the i^{th} model represents the model relating the y_{mi} to inputs. The identified MISO are combined to form a MIMO model in the following manner :

Table 3.6 Inputs and Outputs of model

Model number	Corresponding output	Order
1	Top Pressure of C1	4
2	Temperature of 9th stage of C1	3
3	Temperature of 11th stage of C1	4
4	Top Pressure of C2	4
5	Temperature of 10th stage of C2	3
6	Top Pressure of C3	4
7	Temperature of 7th stage of C3	3
8	Top Pressure of C4	4
9	Temperature of 2nd stage of C4	4
10	Temperature of 5th stage of C2	3

$$
\begin{aligned}
x_m(k+1) &= \begin{bmatrix} A_{m1} & 0 & \cdots & 0 \\ 0 & A_{m2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & A_{m10} \end{bmatrix} \begin{bmatrix} x_{m1}(k) \\ x_{m2}(k) \\ \vdots \\ x_{m10}(k) \end{bmatrix} + \begin{bmatrix} B_{m1} & B_{m2} & \cdots & B_{m10} \end{bmatrix} u(k) \\
&\quad + \begin{bmatrix} G_{m1} \\ G_{m2} \\ \vdots \\ G_{m10} \end{bmatrix} d(k) \\
y_m(k) &= \begin{bmatrix} C_{m1} & 0 & \cdots & 0 \\ 0 & C_{m2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & C_{m10} \end{bmatrix} \begin{bmatrix} x_{m1}(k) \\ x_{m2}(k) \\ \vdots \\ x_{m10}(k) \end{bmatrix}
\end{aligned}
\qquad (3.7)
$$

The NRMSE of the model output versus the identification data for each output can be seen in Table 3.7.

Table 3.7 Goodness of fit of model

Output	NRMSE %
y_1	94.16
y_2	94.62
y_3	93.70
y_4	95.29
y_5	79.26
y_6	94.93
y_7	69.99
y_8	91.07
y_9	79.01
y_{10}	92.53

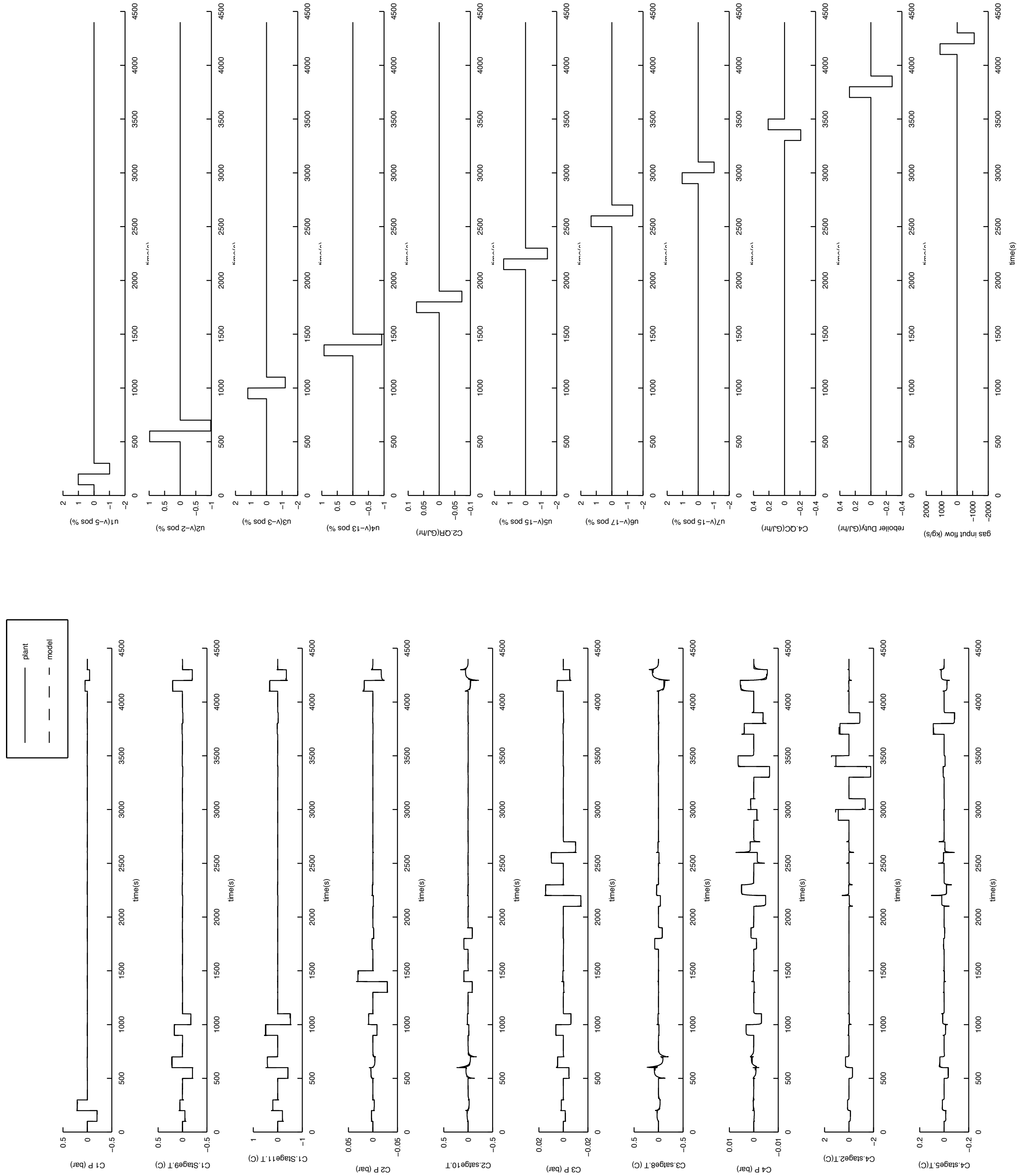


Figure 3.10 Model output vs identification data

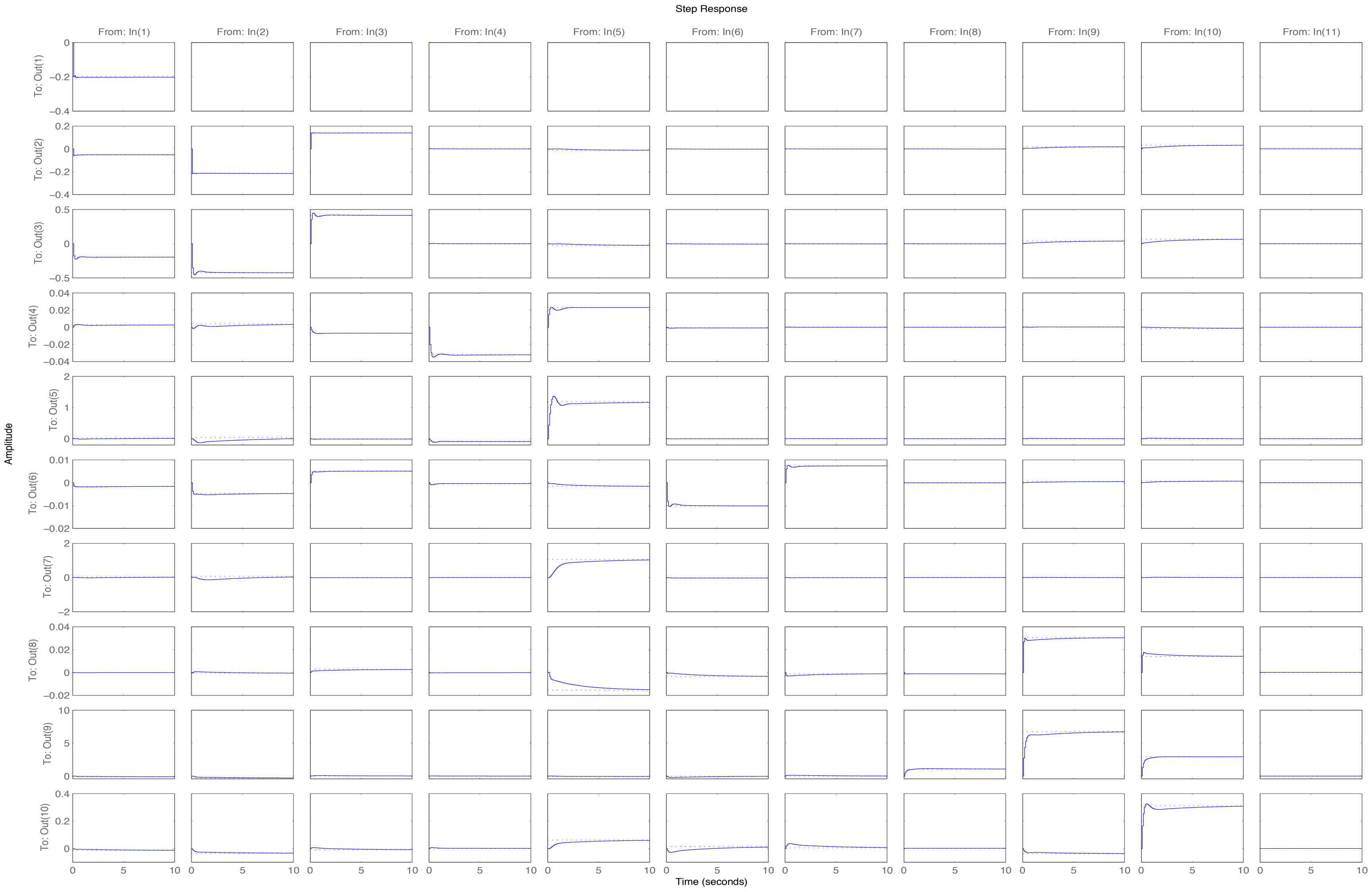


Figure 3.11 Step response of the identified model

3.2.2 Validation of the dynamic model

Now that a model has been identified, it has to be validated against a new series of data. So a new series of input signals was fed into the system and the results were gathered and compared to the model output. Table 3.8 showed the results of this validation.

Table 3.8 Validation of model

Output	NRMSE %
y_1	92.79
y_2	93.74
y_3	92.83
y_4	94.42
y_5	82.13
y_6	94.76
y_7	68.86
y_8	91.95
y_9	67.94
y_{10}	92.01

The observability and controllability matrices of this model were formed and found to be of rank 36, which is equal to the order of the overall model. This means that our model is controllable and that using this model and the measured outputs we can reconstruct the states. Since a proper model has been identified we can move on to the design of a MPC controller.

3.3 Centralized linear MPC design

MPC is based on minimization of an objective function which includes the prediction of output errors and manipulated variable variations subject to constraints. So a MPC controller can be described by Eq 3.8.

$$\begin{aligned}&\min_{\Delta U}(Y - Y_{ref})^T Q(Y - Y_{ref}) + \Delta U^T R\Delta U\\&st\\&a\Delta U \leq b\end{aligned} \tag{3.8}$$

where Y is the vector of predicted outputs,Y_{ref} setpoint, ΔU the input trajectory that minimizes the objective function and Q and R are weighting matrices. Now using the identified model we have to generate a Y in terms of ΔU. In order to do so we will rewrite Eq 3.7 in the following form :

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) + G_m d(k) \\ y_m(k) &= C_m x_m(k) \end{aligned} \tag{3.9}$$

This equation gives us the values of the states at $k+1$ based on values at instant k. We can rewrite Eq 3.9 to have the states at instant k based on instant $k-1$:

$$x_m(k) = A_m x_m(k-1) + B_m u(k-1) + G_m d(k-1) \tag{3.10}$$

Subtracting 3.10 from 3.9 leads to :

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) + G_m \Delta d(k) \tag{3.11}$$

where $\Delta x_m(k) = x_m(k) - x_m(k-1)$, $\Delta u(k) = u(k) - u(k-1)$ and $\Delta d(k) = d(k) - d(k-1)$. Now we have to relate Δx_m to y_m. This is done by augmenting the state variable and creating a new one such that $x(k) = \begin{bmatrix} \Delta x_m(k) \\ y_m(k) \end{bmatrix}$ (Wang, 2009). So we have :

$$\begin{aligned} y_m(k+1) - y_m(k) &= C_m(x_m(k+1) - x_m(k)) = C_m \Delta x_m(k+1) \\ &= C_m(A_m \Delta x_m(k) + B_m \Delta u(k) + G_m \Delta d(k)) \end{aligned} \tag{3.12}$$

We now have an augmented model :

$$\begin{aligned} \underbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y_m(k+1) \end{bmatrix}}_{\text{x(k+1)}} &= \underbrace{\begin{bmatrix} A_m & \emptyset \\ C_m A_m & I \end{bmatrix}}_{\text{A}} \underbrace{\begin{bmatrix} \Delta x_m(k) \\ y_m(k) \end{bmatrix}}_{\text{x(k)}} + \underbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}_{\text{B}} \Delta u(k) + \underbrace{\begin{bmatrix} G_m \\ C_m G_m \end{bmatrix}}_{\text{G}} \Delta d(k) \\ y(k) &= \underbrace{\begin{bmatrix} \emptyset & I \end{bmatrix}}_{\text{C}} \begin{bmatrix} \Delta x_m(k) \\ y_m(k) \end{bmatrix}. \end{aligned} \tag{3.13}$$

As we can see we have embedded an integrator into our model that will insure that we have no offset (Wang and Young, 2006). Using Eq 3.13 we will generate the vector of output predictions. To do so we need a prediction of the system states. It must be considered that from now on the when we refer to states it's the states in eq 3.13 that we have in mind. Before we proceed there are a few notations that should be introduced. First a prediction horizon (N_p) which is the number of intervals that the output is predicted on, and the control horizon (N_c) that is the number of intervals in which the control sequence is calculated. We know that :

$$\begin{aligned}
x(k+1|k) &= Ax(k) + B\Delta u(k) + G\Delta d(k)\\
x(k+2|k) &= Ax(k+1|k) + B\Delta u(k+1) = A^2x(k) + AB\Delta u(k) + AG\Delta d(k) + B\Delta u(k+1)\\
&\vdots\\
x(k+N_p|K) &= A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + A^{N_p-2}B\Delta u(k+1)\\
&\quad + \cdots + A^{N_p-N_c}B\Delta u(k+N_c-1) + A^{N_p-1}G\Delta d(k).
\end{aligned} \tag{3.14}$$

where $x(k+m|k)$ is the predicted state variable at instant $k+m$ based on states at k. It must be noted that $d(k)$ has no future value so it does not go further than instant k. Now that we have the prediction of states we generate the output values :

$$\begin{aligned}
y(k+1|k) &= CAx(k) + CB\Delta u(k) + CG\Delta d(k)\\
y(k+2|k) &= CA^2x(k) + CAB\Delta u(k) + CAG\Delta d(k) + CB\Delta u(k+1)\\
&\vdots\\
y(k+N_p|K) &= CA^{N_p}x(k) + CA^{N_p-1}B\Delta u(k) + CA^{N_p-2}B\Delta u(k+1)\\
&\quad + \cdots + CA^{N_p-N_c}B\Delta u(k+N_c-1) + CA^{N_p-1}G\Delta d(k).
\end{aligned} \tag{3.15}$$

Now we have to form the output prediction matrix. This is done using the following equation :

$$Y = Fx(k) + \Phi\Delta U + \Gamma\Delta d \tag{3.16}$$

where

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \Phi = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \cdots & CA^{N_p-N_c}B \end{bmatrix}; \Gamma = \begin{bmatrix} CG \\ CAG \\ CA^2G \\ \vdots \\ CA^{N_p-1}G \end{bmatrix}.$$

and

$$Y = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ y(k+3|k) \\ \vdots \\ y(k+N_p|k) \end{bmatrix}; \Delta U = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c+1) \end{bmatrix}$$

If n is the number of outputs and m is the number of outputs, then Y is a vector with a length of $n \times N_p$ and ΔU is a vector of length $m \times N_c$ (Wang, 2009). In our case where we have a multivariable model $y = \begin{bmatrix} y_1 y_2 \dots y_n \end{bmatrix}^T$ and $u = \begin{bmatrix} u_1 u_2 \dots u_m \end{bmatrix}^T$. Since our system is large the optimization problem is pretty time consuming. Instead of doing calculations on the whole prediction horizon, we skip a specific number of intervals. This way we have the same sampling interval and prediction horizon but we make the computation faster. This technique helps us because not all our outputs have the same response time and some are slower compared to others as seen in 3.11. We have used Quadratic Programming to solve this optimization problem, thus the constraints must have a form of $a\Delta U \leq b$. In our work we only considered constraints on the inputs and their variations.

$$\begin{aligned} &U_{min} \leq U(k+1) \leq U_{max} \\ &\Delta U_{min} \leq \Delta U \leq \Delta U_{max} \end{aligned} \tag{3.17}$$

As we can see our constraints also contain the absolute value of u_i, thus,U. So we have to reformulate them so that they are explicitly a function of ΔU. We know that if $U(k+1) \leq U_{max}$ then $U(k) + \Delta U \leq U_{max}$, Thus $\Delta U \leq U_{max} - U(k)$. So we can rewrite Eq 3.17 in the following form :

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

and in a matrix form :

$$\underbrace{\begin{bmatrix} I \\ -I \\ I \\ -I \end{bmatrix}}_{a} \Delta U \leq \underbrace{\begin{bmatrix} \Delta U_{max} \\ -\Delta U_{min} \\ U_{max} - U(k) \\ -U_{min} + U(k) \end{bmatrix}}_{b} \tag{3.18}$$

We will reformulate 3.8 into a Quadratic programming problem by replacing Y and rearranging the terms. We have :

$$\begin{aligned} &\Delta U = min \frac{1}{2} H^T \Delta U H + f^T \Delta U + c \\ &st \\ &a\Delta U \leq b \end{aligned} \tag{3.19}$$

where

$$
\begin{aligned}
H &= 2(\Phi^T Q \Phi + R) \\
f &= -2(\Phi^T Q (Y_{ref} - Fx - \Gamma \Delta d))
\end{aligned}
$$

We can see all the information required for solving the QP problem is obtained but the value of the states. Since our states do not have a physical meaning we can not measure them from the plant. We need a state estimator. Therefore, a Kalman Filter was used to provide realtime values of the states for the controller. A steady state Kalman filter was tuned using the kalman command of Matlab®, with a small Q and a large R. This way the filter is based more on the measured values than the model itself or in other words provides state values that correspond to the measured outputs of the plant. As we know the Kalman filter state update equation is :

$$
\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K(y(k) - C\hat{x}(k)) \tag{3.20}
$$

where $\hat{x}$ is the estimated state, K the filter gain and A,B and C the model matrices. By putting the Centralized MPC, Kalman filter and the plant in loop, we can close the control loop. The MPC controller was coded using a S-function. The plant is the Aspen dynamics model linked to Simulink® using the Aspen Modeler block provided by AspenTech®. Since the plant, the controller and Kalman filter have different sampling intervals, zero order holds were placed at the inputs and outputs of the plant. Figures 3.12 to 3.14 show the simulink layout of the centralized MPC simulation and its related subsystems.

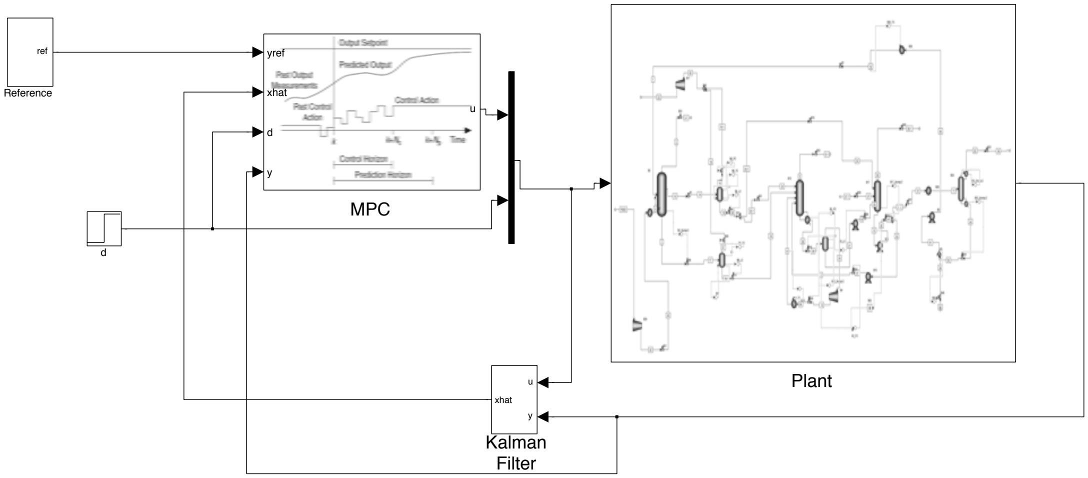


Figure 3.12 Simulink layout of Centralized MPC

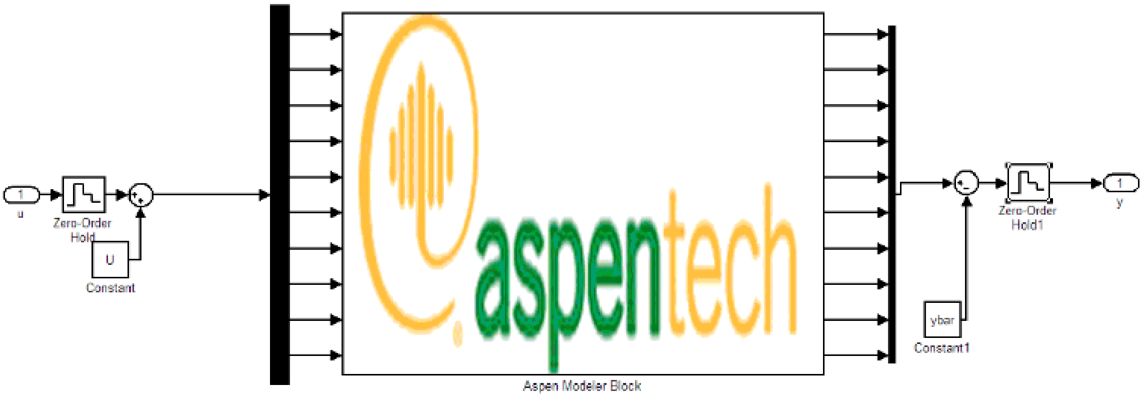


Figure 3.13 Simulink layout of centralized MPC, plant subsystem

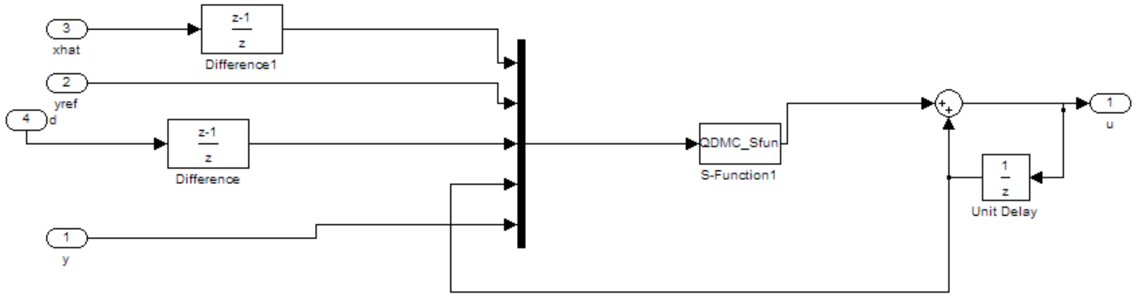


Figure 3.14 Simulink layout of centralized MPC, controller subsystem

3.4 Distributed MPC Design

In order to break down our centralized MPC controller, first the model of the system has to be decomposed into local subsystems. This was done by considering each column as a subsystem and thus forming four. The identified model has a composite form (Venkat *et al.*, 2007) which helps with the decomposition. The pressures and temperatures of each column were put together forming a subsystem. All models consider the effect of all inputs and the measured disturbance on their outputs. Therefore equation 3.7 is broken down in the following form :

$$\begin{aligned} x_{m1}(k+1) &= A_{m1}x_{m1}(k) + B_{m1}u_1(k) + G_{m1}d(k) + \bar{B}_{m1}\bar{u}_1(k) \\ y_{m1}(k) &= C_{m1}x_{m1}(k) \\ x_{m2}(k+2) &= A_{m2}x_{m2}(k) + B_{m2}u_{m2}(k) + G_{m2}d(k) + \bar{B}_{m2}\bar{u}_{m2}(k) \\ y_{m2}(k) &= C_{m2}x_{m2}(k) \\ x_{m3}(k+3) &= A_{m3}x_{m3}(k) + B_{m3}u_{m3}(k) + G_{m3}d(k) + \bar{B}_{m3}\bar{u}_{m3}(k) \\ y_{m3}(k) &= C_{m3}x_{m3}(k) \\ x_{m4}(k+4) &= A_{m4}x_{m4}(k) + B_{m4}u_{m4}(k) + G_{m4}d(k) + \bar{B}_{m4}\bar{u}_{m4}(k) \\ y_{m4}(k) &= C_{m4}x_{m4}(k) \end{aligned} \tag{3.21}$$

where $\bar{u}_i$ represents all the inputs not manipulated by the subsystem. Table 3.9 shows the corresponding inputs and outputs of each subsystem. Just as seen in the previous section through equations 3.9 to 3.16 the models were all augmented to the following form :

$$\begin{aligned} Y_i &= F_i x_i(k) + \Phi_i \Delta U_i + \Gamma_i \Delta d + \Psi_i \Delta \bar{U}_i \\ i &= 1 \dots 4 \end{aligned} \tag{3.22}$$

Table 3.9 Inputs and Outputs of model

Subsystem # (i)	Outputs	Inputs
1	Top Pressure of C1(y_1)	% opening of V-5 (u_1)
	Temperature of 9th stage of C1(y_2)	% opening of V-2 (u_2)
	Temperature of 11th stage of C1(y_3)	% opening of V-3(u_3)
2	Top Pressure of C2(y_4)	% opening of V-13(u_4)
	Temperature of 10th stage of C2(y_5)	% C2 reboiler Duty (GJ/hr)(u_5)
2	Top Pressure of C3(y_6)	% opening of V-16(u_6)
	Temperature of 7th stage of C3(y_7)	% opening of V-17(u_7)
4	Top Pressure of C4 (y_8)	% opening of V-19(u_8)
	Temperature of 2nd stage of C4(y_9)	C4-cond Duty (GJ/hr)(u_9)
	Temperature of 5th stage of C4(y_{10})	C4-reboiler Duty (GJ/hr)(u_{10})

where F_i,Φ_i and Γ_i are defined in the same manner as in equation 3.16, and

$$\Psi_i = \begin{bmatrix} C_i\bar{B}_i & 0 & 0 & \cdots & 0 \\ C_iA_i\bar{B}_i & C_i\bar{B}_i & 0 & \cdots & 0 \\ C_iA_i^2\bar{B}_i & C_iA_i\bar{B}_i & C_i\bar{B}_i & \cdots & 0 \\ \vdots & & & & \\ C_iA_i^{N_p-1}\bar{B}_i & C_iA_i^{N_p-2}\bar{B}_i & C_iA_i^{N_p-3}\bar{B}_i & \cdots & C_iA_i^{N_p-N_c}\bar{B}_i \end{bmatrix}$$

All the controllers here have the same N_p and N_c. $N_c \geq 2$ is a must in order for the algorithm to have a good performance and without this condition the communication would be somewhat pointless. Ψ_i has the same dimensions as Φ_i. Now the QP problems can be formed. Each problem has its own constraints which are defined as in the previous section, in other words our DMPC subproblems have a structure and formulation very similar to the centralized problem .

$$\begin{aligned} &\Delta U_i = min\frac{1}{2}H_i^T\Delta U_i H_i + f_i^T\Delta U_i \\ &st \\ &a_i\Delta U_i \leq b_i \\ &i = 1\ldots 4 \end{aligned} \tag{3.23}$$

where

$$\begin{aligned} H &= 2(\Phi_i^T Q_i \Phi_i + R_i) \\ f &= -2(\Phi_i^T Q_i(Y_{refi} - F_i x_i - \Gamma_i\Delta d - \Psi_i\Delta\bar{U}_i)) \end{aligned}$$

Rawlings and Stewart (2008) have suggested that for a process with total recycle the information flow diagram between controllers should be similar to Figure 3.15, but the information flow in our DMPC is better represented by Figure 3.16 :

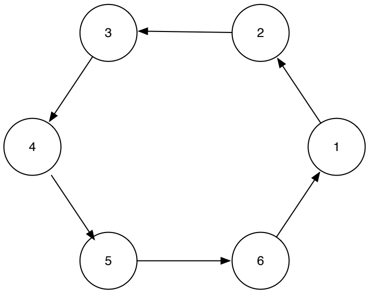


Figure 3.15 Information flow suggested by Rawlings and Stewart (2008)

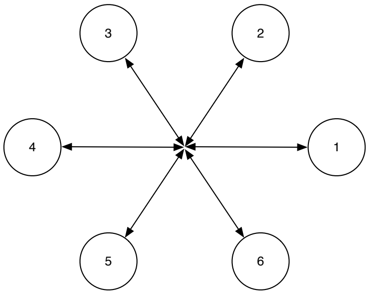


Figure 3.16 Information flow of our DMPC

As we can see these problems are not completely independent of each other compared to Decentralized MPC. The four optimization of each MPC subcontroller were solved in an iterative manner, but they do not cooperate with each other in solving a global objective function.Our DMPC is of a iterative, non-cooperative class, which acts by the following algorithm :

1. At each sampling instant (every 0.1 seconds) the changes in states, disturbance and inputs and also the absolute value of process outputs are measured.
2. $\Delta\bar{U}$ is set to $\begin{bmatrix} \Delta u \\ \vdots \\ 0 \end{bmatrix}_{1\times mN_c}$ and the proper elements are sent to each subcontroller.
3. for $i = 1$ to N (N is a tuneable parameter which we have chosen 10).
   (a) Each subproblem (subcontroller) is solved (equation 3.23).

(b) $\Delta\bar{U} = \begin{bmatrix} \Delta u_1 \\ \vdots \\ \Delta u_4 \end{bmatrix}$ and sent to other subcontrollers.

(c) return to (a) unless $i = N$.

4. The first series of ΔU is sent to the plant as a control signal.

Further details of this method and the code can be found in the appendix.

3.5 Design of the adaptive extremum seeking loop

After finding the proper primary layer of control configuration or the regulatory layer, based on our objectives we were to design and tune an adaptive extremum seeking scheme that optimizes an objective function of the system. In this framework we decided to define an objective function that quantifies the quality of the product streams, or in other words that shows the deviation of these product streams from their standards. the chosen objective function is :

$$\begin{aligned} J =& a_1(x_{\mathrm{H_2S\ in\ Clean\ gas}} - x^{standard}_{\mathrm{H_2S\ in\ Clean\ gas}})^2 \\ &+ a_2(x_{\mathrm{CO_2\ in\ CO_2\ stream}} - x^{standard}_{\mathrm{CO_2\ in\ CO_2\ stream}})^2 \\ &+ a_3(x_{\mathrm{H_2S\ in\ clause\ gas\ stream}} - x^{standard}_{\mathrm{H_2S\ in\ clause\ gas\ stream}})^2 \end{aligned} \tag{3.24}$$

The variables we chose to manipulate to optimize our objective function were setpoints on y_2, y_5 and y_{10}. Since we have three manipulated variables we will have a schematic in the form of Figure 3.17. As seen in Figure 3.17 the extremum seeking method used is multivariable static mapping. The overall system has three time scales : the fastest which is the plant and it's controllers, the medium which is the perturbation signal or dieter signal and the slowest which is the high pass filter. As we can see in Figure 3.17 the setpoints are perturbed by the dither signal, therefore perturbing the objective function. The high pass filter washes out or removes the low frequency part of J, or in simple words the part of the system that is not affected by the dither. After demodulation the integrator's estimation gives us the gradient, which optimizes the objective. This method finds local optimums , which is why we have chosen a quadratic objective (Ariyur and Krstic, 2003).

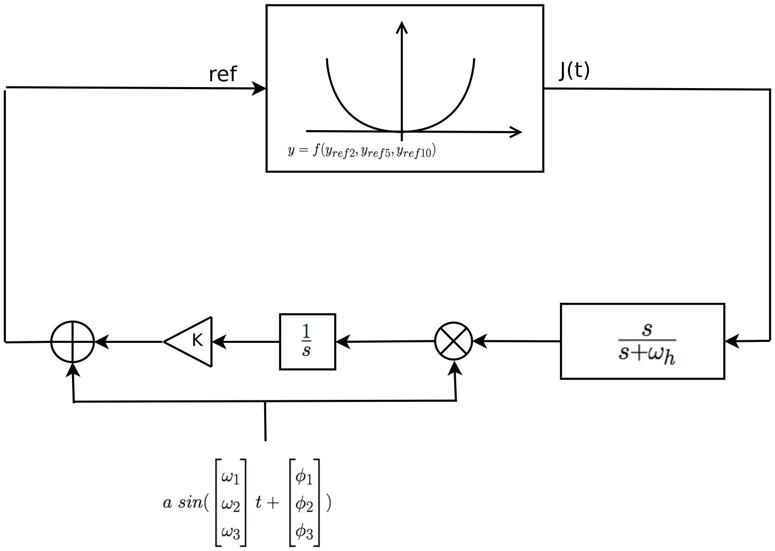

ref
J(t)
$y = f(y_{ref2}, y_{ref5}, y_{ref10})$
K
$\frac{1}{s}$
$\frac{s}{s+\omega_h}$
$a\ sin(\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} t + \begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\end{bmatrix})$

Figure 3.17 Adaptive extremum seeking principle

The equivalent of Figure 3.17 in simulink is given in Fig 3.18

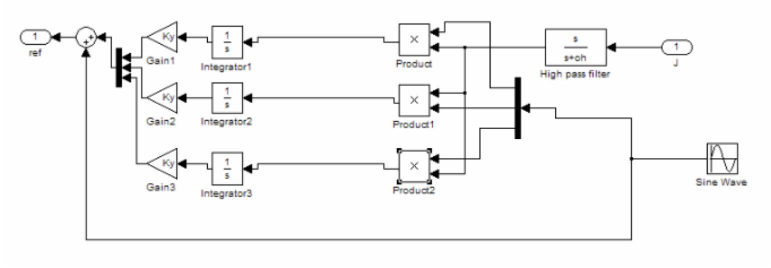
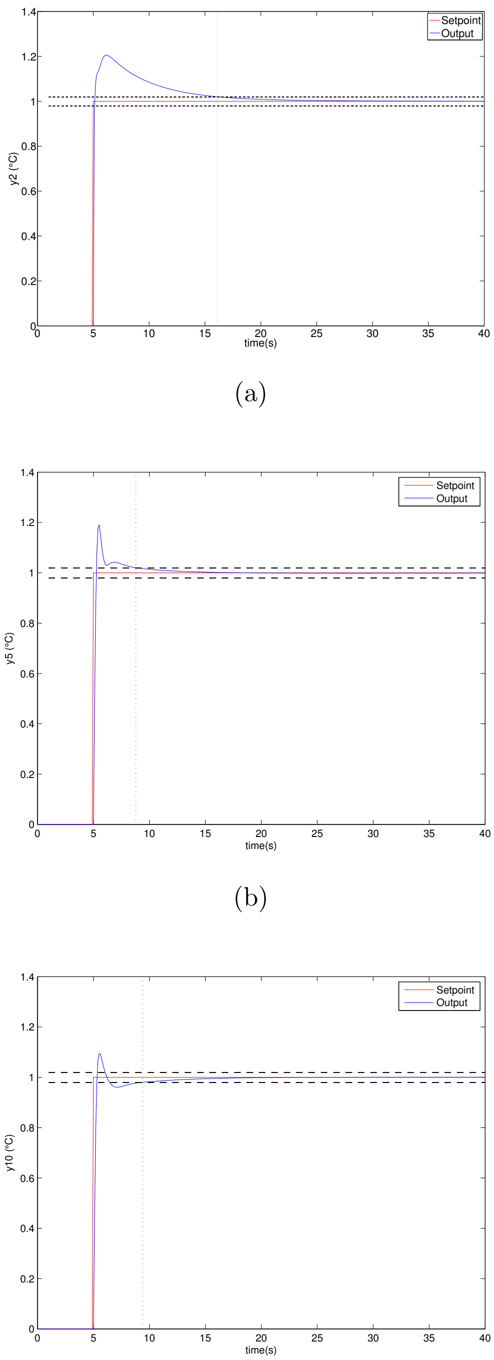


Figure 3.18 Adaptive extremum seeking in simulink

To tune the adaptive extremum seeking structure, the amplitude of the dither signal was chosen in a way to obtain a small steady state error. The components of the dither signal have a phase angle of 0 ,$\frac{\pi}{2}$ and π respectively. The high pass filter was tuned so that it is in a slower time scale compared to the plant with its controllers. The gains were set to a value corresponding to the effect of their related input on the objective function (Ariyur and Krstic, 2003). The frequencies were chosen corresponding to the settling time of the related output of the system. Figure 3.19 shows the closed loop time response of plant with a centralized MPC for the setpoints considered in our extremum seeking scheme.



(a)

(b)

(c)

Figure 3.19 Settling time for the closed loop system for a) 2^{nd}, b) 5^{th}, c) 10^{th} output of the system with Centralized MPC

The frequency of the dither signal for the 5^{th} and the 10^{th} setpoint are chosen proportional to the frequency of the 2^{nd} output by the inverse of their settling times.

The combination of adaptive extremum seeking and our regulatory controllers is shown in Figure 3.20.

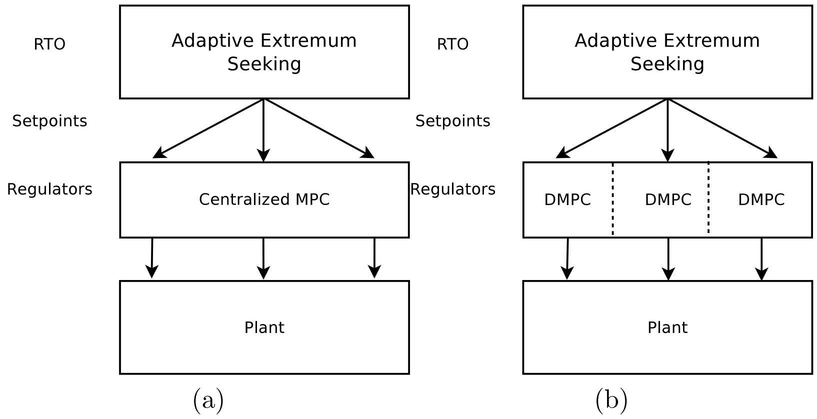


Figure 3.20 Combination of adaptive extremum seeking with a) Centralized MPC and b) Distributed MPC

Since the simulations with adaptive extremum seeking need more time and memory to run, and due to fact that the Aspen plus dynamics can only run in a 32 bit windows environment, the simulations where run using a linear model obtained from the Control Design Interface of Aspen plus Dynamics.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter the preparation of simulations and of multiloop PI, Centralized MPC, Decentralized MPC and Distributed MPC are presented and compared. Afterwards the adaptive extremum seeking scheme is applied to some of the control structures and results are obtained and examined.

4.1 Tuning of different control schemes

4.1.1 Multiloop PI tuning

As mentioned in the previous chapter a series of PI controllers were removed form the Aspen plus dynamics file in order to obtain a uniform dynamic platform for all simulations. The controllers in both Simulink and Aspen plus dynamics were tuned using the Internal Model Control (IMC) interpretation of PID (lambda tuning). The input output relations were approximated by a first order plus dead time model and the following formulas were used for tuning :

$$G_p i = \frac{Y_i(s)}{U_i(s)} = \frac{k_p}{\tau_p s + 1} e^{-\theta s} \tag{4.1}$$

The parameters of this model were used in the following equations to tune an ideal PI controller :

$$k_c = \frac{\tau_p + \frac{\theta}{2}}{k_p(\theta + \lambda)} \tag{4.2}$$

$$\tau_I = \tau_p + \frac{\theta}{2} \tag{4.3}$$

The controllers obtained are continous so in order to implement them they were discretized using the Tustin method. Table 4.1 shows the discrete controller parameters obtained. Controller i corresponds to the i^{th} input and output and the sampling interval for all is 0.1 seconds.

Table 4.1 Tuning of PI controllers

i	Flowsheet tag	k_p	τ_i
1	PC-1	-3.39 (%/bar)	16.9(s)
2	TC-1	-0.0395(%/(°C)	0.051(s)
3	TC-2	0.162(%/°C)	0.223(s)
4	PC-4	-0.46 (%/bar)	0.148(s)
5	TC-5	0.229 (Gj/hr/°C)	1.35(s)
6	PC-6	-8.33(%/bar)	3.39(s)
7	TC-7	-2.46(%/°C)	0.051(s)
8	PC-7	-8.29 (%/bar)	0.051(s)
9	TC-9	0.00176(Gj/hr/°C)	0.355(s)
10	TC-10	1.82(Gj/hr/°C)	5.68 (s)

PI controllers with this tuning were implemented. The simulations were carried out for two different cases, one where the controller rejects a 500Kg/s input feed step increase at instant 20 seconds, and other for tracking the setpoints.

4.1.2 Centralized MPC tuning

A centralized MPC controller was designed in Chapter 3. This controller runs alongside with a series of PI controllers tuned by the Ziegler-Nichols method. Tuning of the MPC controller was done done using the linear model and fine-tuned using the actual plant later on. The tuning matrices of this controller are as follows :

$$Q_i = \begin{pmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{pmatrix}$$

$$R_i = \begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}$$

These matrices are adjusted to the required size based on N_p and N_c to be compatible with Eq 3.8. The prediction horizon is 500 and the control horizon is set to 2.

$$Q = \begin{bmatrix} Q_i & 0 & \cdots & 0 \\ 0 & Q_i & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & Q_i \end{bmatrix}_{nN_p \times nN_p} ; R = \begin{bmatrix} R_i & 0 & \cdots & 0 \\ 0 & R_i & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & R_i \end{bmatrix}_{mN_c \times mN_c}$$

4.1.3 Distributed and Decentralized MPC tuning

The DMPC designed in Chapter 3, also runs along with PI controllers tuned in Ziegler Nichols. The subcontrollers were tuned using the the same weights of centralized MPC for outputs. The output and input weight matrices are as follows :

$$Q_{1i} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 10 \end{pmatrix} ; Q_{2i} = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} ; Q_{3i} = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} ; Q_{4i} = \begin{pmatrix} 50 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}.$$

$$R_{1i} = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} ; R_{2i} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} ; R_{3i} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} ; R_{4i} = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}.$$

The prediction and control horizons are exactly the same as the centralized controller and the number of iterations for DMPC in one sampling interval was determined by simulation with the complete linear model.

4.2 Comparison of controller performance

As a criteria to compare the results of our different controllers The Integral of Squared Error (ISE) criterion was used.

$$ISE = \int_0^t (y_{ref}(t) - y(t))^2 dt \tag{4.4}$$

Another criteria considered was the integral of the the centralized MPC objective function as shown below :

$$\int_0^t (y_{ref}(t) - y(t))^T Q (y_{ref}(t) - y(t)) + \Delta u(t)^T R \Delta u(t) dt \tag{4.5}$$

As mentioned before all controllers were simulated in two main themes : disturbance rejection and setpoint tracking. The following tables and Figures 4.1 and 4.2 show the results with the chosen criteria :

Table 4.2 Results for Disturbance rejection

Criteria	Multiloop PI	MPC cent	DMPC	Decent MPC
$ISE(y_1)$	3.401E-3	9.889E-5	7.805E-5	6.695E-5
$ISE(y_2)$	1.801E-2	1.240E-3	9.570E-4	8.424E-4
$ISE(y_3)$	2.484E-2	5.726E-3	3.894E-3	3.740E-3
$ISE(y_4)$	5.102E-4	1.040E-4	2.834E-6	4.932E-6
$ISE(y_5)$	1.438E-3	2.422E-3	2.644E-4	2.507E-4
$ISE(y_6)$	3.415E-3	1.033E-4	3.252E-5	1.441E-4
$ISE(y_7)$	8.154E-4	3.525E-4	8.510E-4	5.989E-3
$ISE(y_8)$	1.372E-4	1.016E-5	1.651E-4	1.783E-4
$ISE(y_9)$	6.574E+1	2.417E-4	9.911E-4	1.692E-3
$ISE(y_{10})$	4.522E-3	2.426E-5	6.353E-4	8.321E-4
$\sum ISE(y_i)$	6.580E+1	1.032E-2	7.872E-3	1.374E-2
Integral of MPC objective function	6.582E+2	1.498E-1	1.062E-1	1.671E-1

Table 4.3 Results for Set point tracking

ISE	Multiloop PI	MPC cent	DMPC	Decent
$ISE(y_1)$	7.138E+0	6.462E-1	7.556E-1	7.557E-1
$ISE(y_2)$	6.811E+0	1.163E+0	2.636E-1	2.526E-1
$ISE(y_3)$	1.114E+1	1.882E+0	5.842E-1	5.571E-1
$ISE(y_4)$	8.387E-2	3.332E-1	6.746E-3	8.856E-3
$ISE(y_5)$	4.265E+0	4.494E+0	7.010E-1	5.434E-1
$ISE(y_6)$	8.406 E+0	1.395E-1	4.359E-1	8.116E-1
$ISE(y_7)$	5.162E+0	1.970E+0	4.574E+0	3.031E+1
$ISE(y_8)$	5.884E-1	5.464E-2	5.868E-1	9.231E-1
$ISE(y_9)$	8.616E+4	2.400E+0	8.072E+0	2.810E+0
$ISE(y_10)$	4.156E+1	3.970E-1	3.291E+0	6.791E+0
$\sum ISE(y_i)$	8.625E+5	1.348E+1	1.927E+1	6.905E+1
Integral of MPC objective function	8.627E+5	1.801E+2	2.323E+2	7.534E+2

The results show that all predictive controllers have a much better performance compared to the Multiloop PI controllers, mainly due the fact that a feedforward disturbance model is easily integrated in these controllers. For setpoint tracking, the centralized MPC shows the best performance and the DMPC has a performance very close to it. Also we see that adding communication and an iterative nature to a decentralize MPC can improve its performance. Table 4.5 shows the loss of precision for the controllers based on the performance for setpoint tracking of centralized MPC using the following equation :

$$\frac{\int J_C dt - \int J_{CMPC} dt}{\int J_{CMPC} dt} \tag{4.6}$$

Table 4.4 Comparison of loss of precision for setpoint tracking

Controller	Loss of precision
Multiloop PI	4789.7
Centralized MPC	0
DMPC	0.29
Decentralized MPC	3.18

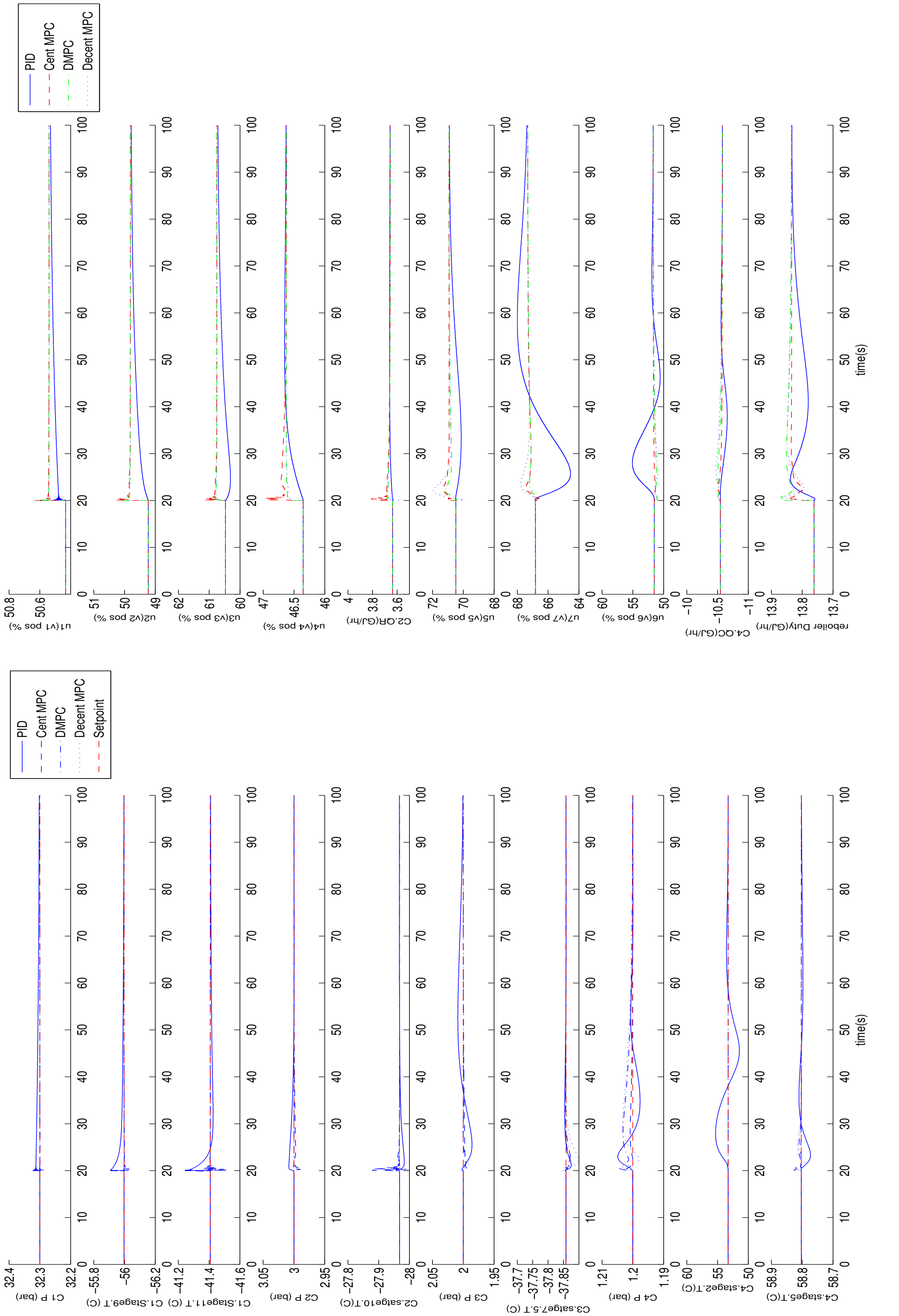


Figure 4.1 Performance of controllers for disturbance rejection

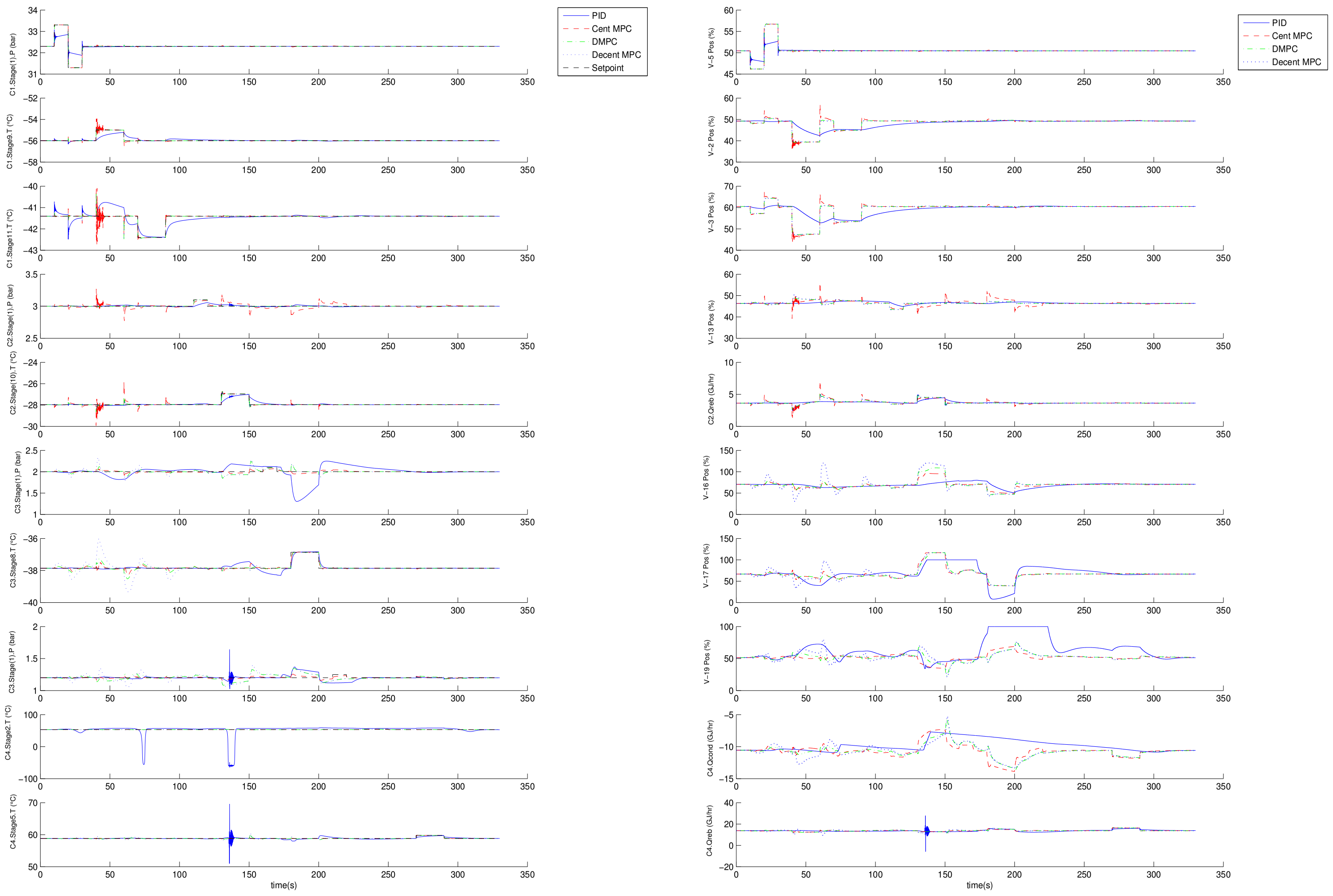


Figure 4.2 Performance of controllers for setpoint tracking

Both the numerical and graphical results show that PI controllers show a poor performance compared to all MPC schemes, in both disturbance rejection and setpoint tracking. Even in some cases the PI controllers become wind up at the constraints of the manipulated variables, while the predictive controllers are not even close to the constraints. this can be related to a couple of reasons. First the tuning of the PI controllers, which is not meant for fast responses unlike the MPCs. Another reason is that our PI controllers have no antiwind up strategy. Tests were also done with anti-windup PID tuned by Ziegler Nichols, but the performance was still significantly poor compared to MPCs. At last PIDs do not consider the interaction amongst process variables. This can be solved by adding a decoupler, but then, the problem is that considering input constraints with decouplers is complicated.

Among the MPC controllers, the centralized scheme has the best performance, since it has full control over all interactions, but it is also more costly and very risky in terms of maintenance. Distributed MPC has a performance insignificantly worse compared to centralized MPC but it's easier to implement and maintain. Decentralized MPC is better than multi loop PI, but we have to consider that by adding communication and iteration to it we will significantly improve its performance by replacing decentralization with distribution. In the case of DMPC in a worst case scenario where communication fails we will still be able to control the process with a Decentralized scheme.

4.3 Adaptive extremum seeking

Adaptive extremum seeking was applied to centralized and distributed MPC at different frequency based on what was discussed in Chapter 3 with the following tuning :

Table 4.5 Tuning of adaptive extremum seeking parameters

Parameter	Value
K_1	30
K_2	10
K_3	30
high pass filter frequency	3×10^{-4}
a_1	0.005
a_2	0.05
a_3	0.005

where K_i is the gain and a_i is the amplitude for each component. As mentioned before, if the frequency of the setpoint on y_2 is chosen to be ω, the frequency for y_5 and y_{10} will be 2.9053ω and 2.5458ω, respectively. Figure 4.3 shows the results of adaptive extremum seeking with Centralized MPC for three different frequencies.

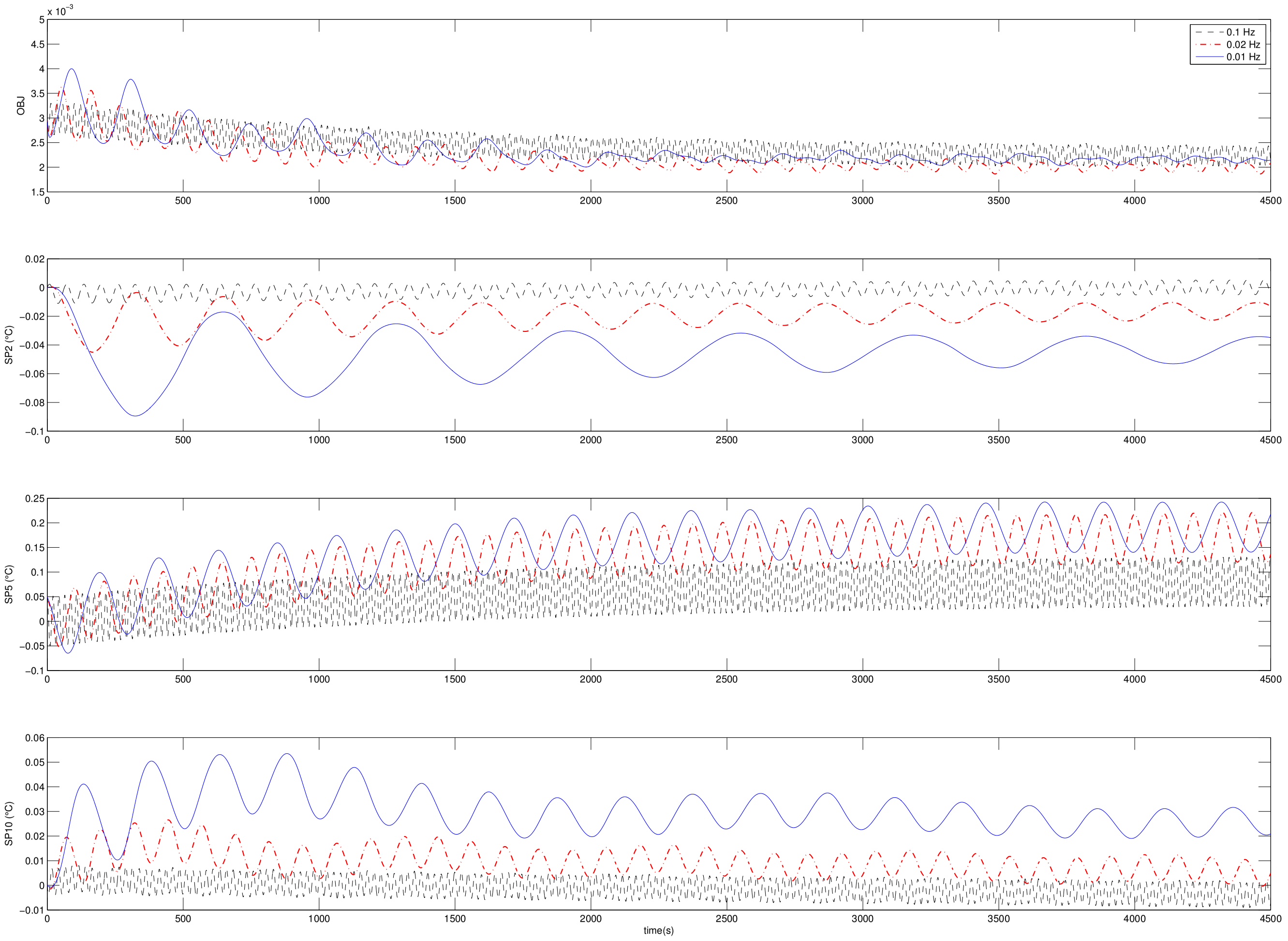


Figure 4.3 Performance of adaptive extremum seeking with centralized MPC

As we can see, the best performance is achieved through a frequency of 0.02 Hz. At 0.01 Hz we do not reach smaller values and at 0.1 Hz we have a similar case. The reason can be related to the fact that in order to have a good estimate of our gradient we need an optimum frequency.
The same frequencies were applied to Distributed MPC. The results are shown in Figure 4.4. For frequencies of 0.01 and 0.02 Hz we have identical results to centralized MPC, but for 0.1 Hz the objective function value oscillates around its initial value. This can be related to fact that DMPC is not able to follow the setpoints generated for the 10^{th} output. This is shown in Figure 4.5.

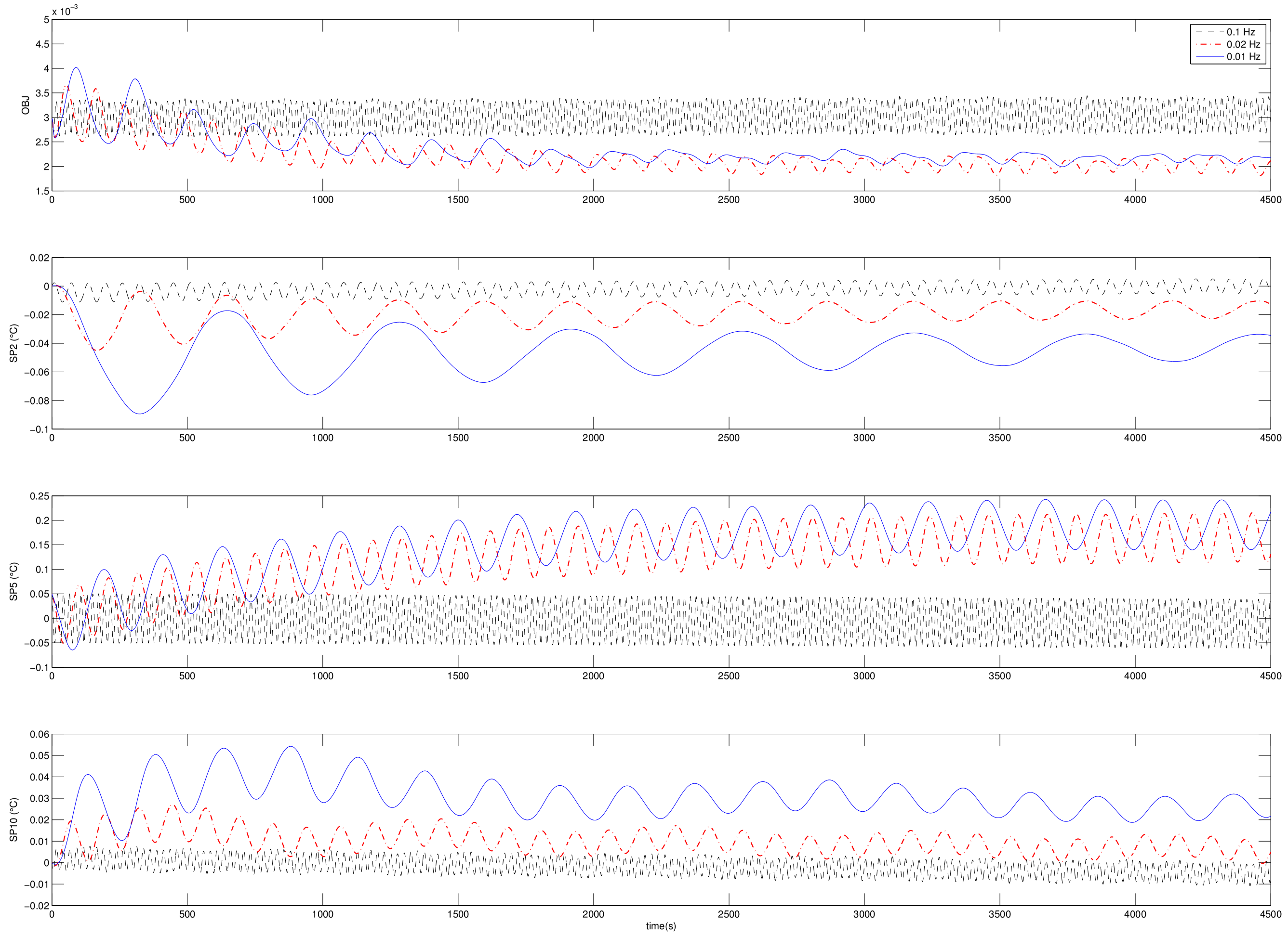


Figure 4.4 Performance of adaptive extremum seeking with Distributed MPC

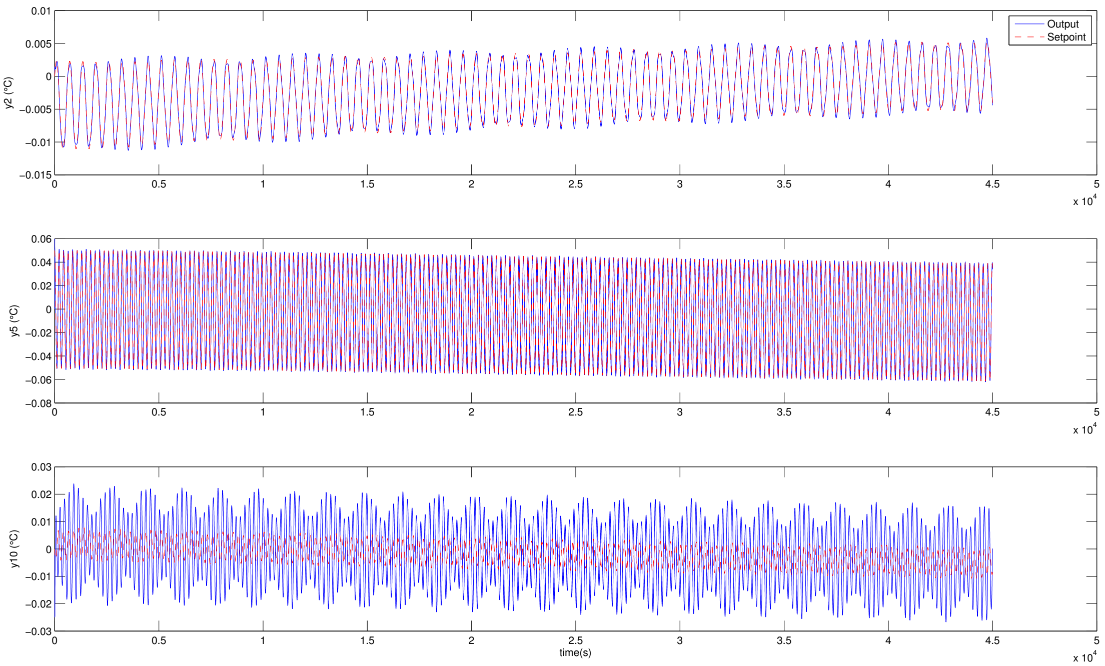
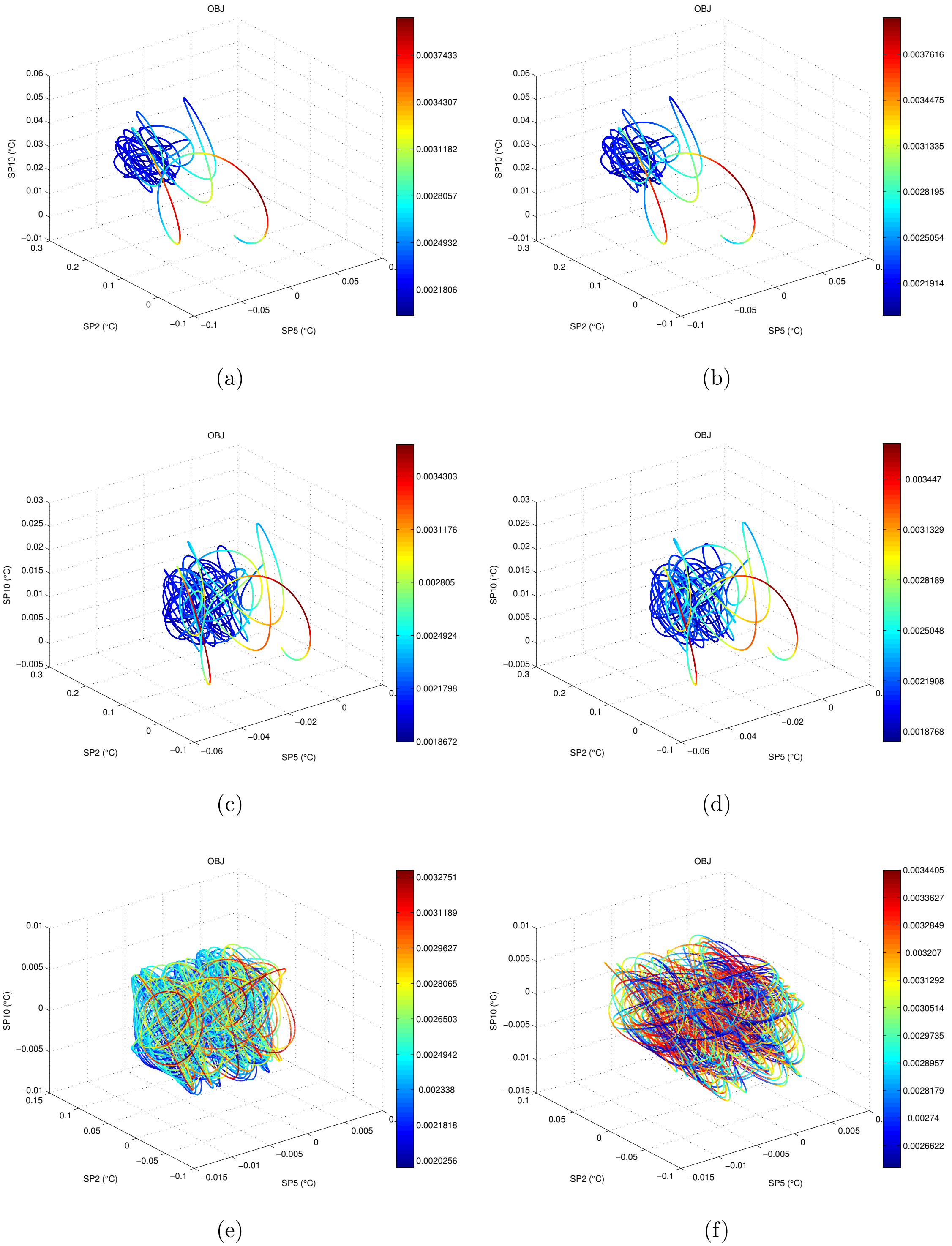


Figure 4.5 Tracking performance of distributed MPC for setpoints generated by adaptive extremum seeking

The trajectories of setpoints generated by adaptive extremum seeking have been plotted in 3D (Figure 4.6). As we can see the behaviour of the system for the controllers is exactly identical at 0.01 and 0.02 Hz, but as explainbed aboved, they perform differently at 0.1 Hz.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.6 Setpoint trajectories for a) CMPC at 0.01 Hz, b) DMPC at 0.01 Hz, c) CMPC at 0.02 Hz, d) DMPC at 0.02 Hz, e) CMPC at 0.1 Hz and f) DMPC at 0.1 Hz

4.4 Summary of results

Four different plantwide control structures were presented. Centralized MPC and distributed MPC were found to be better performing. Adaptive extremum seeking was applied to these two controllers and it was found that for a proper frequency the two control structures perform identically.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The general objective of this thesis was to apply adaptive extremum seeking to the Rectisol process to improve its performance in the presence of measured and unmeasured disturbances related to upstream gasification unit. To achieve this objective, three steps were defined, obtaining a dynamic model of Rectisol, finding a responsive regulatory control structure and implementation of adaptive extremum seeking.

The Aspen Plus dynamic software was used to develop a dynamic model of Rectisol based on information found in the literature. This information was based on steady state performance, and the steady state behaviour was used to validate our model. The dynamic model was used to evaluate our different control structures.

Four plantwide control structures were applied, multiloop PI, centralized MPC, distributed MPC and decentralized MPC. The four control strategies were simulated using a combination of Matlab, Simulink and Aspen Plus dynamic software. The results of our simulations were compared for setpoint tracking and disturbance rejection using measures of perofrmance, such as ISE, and showed that the centralized and distributed MPC controllers are able to attenuate measured disturbances and follow desired setpoints with a performance superior to that of decentralized MPC and Multiloop PI.

To fulfill our main objective, a multivariable static mapping adaptive extremum seeking control scheme was developed. The objective function to be minimized was a quadratic sum of the difference of the three gas products from their standards. This scheme was applied to centralized and distributed MPC and were simulated. The results showed that at a proper dither signal frequency both controllers showed identical performance and capability to minimize our objective function. This also means that if ever any unmodelled or unmeasurable perturbation affects the plant, adaptive extremum seeking will correct its assigned setpoints to keep the product quality as close as possible to their desired standards. At last we have been able to achieve our objective and also take advantage of Distributed MPC and to overcome the limitations of centralized MPC.

5.2 Future works and recommendations

To continue the work done in this thesis the following unexplored topics can be studied :

1. Validation of dynamic model for Rectisol : The dynamic model of Rectisol can be validated by using industrial data, to make sure that it represents the dynamic behaviour as closely as possible .
2. Improvement of Distributed MPC : Our Distributed MPC structure is of non cooperative nature. It is recommended to develop and evaluate a cooperative DMPC structure. Also the distribution parameters here were chosen in a manner that guarantees performance without considering calculation time. It is recommended to optimize these parameters with respect to computation time.
3. Improvement of the integration of Simulink and Aspen dynamics for cosimulation : The main problem with the interface provided by AspenTech is the limitation to 32 bit operating systems, leading to the inability of performing very long simulations. It is recommended to find a solution to this problem by creating an interface for 64 bit operating systems.
4. Implementation of the suggested control and optimization configuration on a real plant.

References

ALESSIO, A. and BEMPORAD, A. (2008). Stability conditions for decentralized model predictive control under packet drop communication. *American Control Conference, 2008.* IEEE, p. 3577–3582.

ARIYUR, K. and KRSTIC, M. (2002). Analysis and design of multivariable extremum seeking. *American Control Conference, 2002. Proceedings of the 2002.* vol. 4, p. 2903–2908 vol.4.

ARIYUR, K. B. and KRSTIC, M. (2003). *Real-time optimization by extremum-seeking control.* Wiley-Interscience.

ASPENTECH (July 2008). Aspen plus igcc model.

BANASZUK, A., ZHANG, Y. and JACOBSON, C. (2000). Adaptive control of combustion instability using extremum-seeking. *American Control Conference, 2000. Proceedings of the 2000.* vol. 1, p. 416–422 vol.1.

BILDEA, C. S. and DIMIAN, A. C. (2003). Fixing flow rates in recycle systems : Luyben's rule revisited. *Industrial & Engineering Chemistry Research*, 42, 4578–4585.

BUCKLEY, P. (1964). *Techniques of process control.* Wiley.

CAMPONOGARA, E., JIA, D., KROGH, B. and TALUKDAR, S. (2002). Distributed model predictive control. *Control Systems, IEEE*, 22, 44–52.

CHRISTOFIDES, P. D., SCATTOLINI, R., MUÑOZ DE LA PEÑA, D. and LIU, J. (2013). Distributed model predictive control : A tutorial review and future research directions. *Computers & Chemical Engineering*, 51, 21–41.

CLARKE, D. W., MOHTADI, C. and TUFFS, P. (1987). Generalized predictive control—part i. the basic algorithm. *Automatica*, 23, 137–148.

COUGNON, P., DOCHAIN, D., GUAY, M. and PERRIER, M. (2011). On-line optimization of fedbatch bioreactors by adaptive extremum seeking control. *Journal of Process Control*, 21, 1526–1532.

CUI, H. and JACOBSEN, E. W. (2002). Performance limitations in decentralized control. *Journal of Process Control*, 12, 485–494.

CUTLER, C. R. and RAMAKER, B. (1980). Dynamic matrix control-a computer control algorithm. *Proceedings of the joint automatic control conference.* American Automatic Control Council Piscataway, NJ, vol. 1, p. Wp5–B.

GARCIA, C. E. and MORSHEDI, A. (1986). Quadratic programming solution of dynamic matrix control (qdmc). *Chemical Engineering Communications*, 46, 73–87.

GUAY, M., DOCHAIN, D. and PERRIER, M. (2005). Adaptive extremum-seeking control of nonisothermal continuous stirred tank reactors. *Chemical engineering science*, 60, 3671–3681.

HARUN, N., NITTAYA, T., DOUGLAS, P. L., CROISET, E. and RICARDEZ-SANDOVAL, L. A. (2012). Dynamic simulation of MEA absorption process for CO_2 capture from power plants. *International Journal of Greenhouse Gas Control*, 10, 295 – 309.

HEIL, S., BRUNHUBER, C., LINK, K., KITTEL, J. and MEYER, B. (2009). Dynamic modelling of co2-removal units for an igcc power plant. *Proceedings 7th modelica conference.* The Modelica Association, p. p77–85.

HILLER, H., REIMERT, R., MARSCHNER, F., RENNER, H.-J., BOLL, W., SUPP, E., BREJC, M., LIEBNER, W., SCHAUB, G., HOCHGESAND, G., HIGMAN, C., KALTEIER, P., MÜLLER, W.-D., KRIEBEL, M., SCHLICHTING, H., TANZ, H., STÖNNER, H.-M., KLEIN, H., HILSEBEIN, W., GRONEMANN, V., ZWIEFELHOFER, U., ALBRECHT, J., COWPER, C. J. and DRIESEN, H. E. (2000). *Gas Production*, Wiley-VCH Verlag GmbH and Co. KGaA. p. 95–99.

HUDON, N., GUAY, M., PERRIER, M. and DOCHAIN, D. (2008). Adaptive extremum-seeking control of convection-reaction distributed reactor with limited actuation. *Computers & Chemical Engineering*, 32, 2994–3001.

HUDON, N., PERRIER, M., GUAY, M. and DOCHAIN, D. (2005). Adaptive extremum seeking control of a non-isothermal tubular reactor with unknown kinetics. *Computers & Chemical Engineering*, 29, 839–849.

KRSTIĆ, M. (2000). Performance improvement and limitations in extremum seeking control. *Systems & Control Letters*, 39, 313–326.

LARSON, E. D., CONSONNI, S., KATOFSKY, R. E., CONSULTING, N., BURLINGTON, I., IISA, K. and FREDERICK JR, W. J. (2006). A cost-benefit assessment of gasification-based biorefining in the kraft pulp and paper industry.

LARSSON, T., GOVATSMARK, M. S., SKOGESTAD, S. and YU, C. C. (2003). Control structure selection for reactor, separator, and recycle processes. *Industrial & Engineering Chemistry Research*, 42, 1225–1234.

LIN, Y.-J., PAN, T.-H., WONG, D. S.-H., JANG, S.-S., CHI, Y.-W. and YEH, C.-H. (2010). Plantwide control of CO_2 capture by absorption and stripping using monoethanolamine solution. *Industrial & Engineering Chemistry Research*, 50, 1338–1345.

LJUNG, L. (1999). *System identification.* Wiley Online Library, second edition.

LUXAT, J. C. and LEES, L. H. (1971). Stability of peak-holding control systems. *Industrial Electronics and Control Instrumentation, IEEE Transactions on*, IECI-18, 11–15.

LUYBEN, M. L., TYREUS, B. D. and LUYBEN, W. L. (1997). Plantwide control design procedure. *AICHE journal*, 43, 3161–3174.

LUYBEN, W. L. (1993). Dynamics and control of recycle systems. 1. simple open-loop and closed-loop systems. *Industrial & Engineering Chemistry Research*, 32, 466–475.

LUYBEN, W. L. (1994). Snowball effects in reactor/separator processes with recycle. *Industrial & Engineering Chemistry Research*, 33, 299–305.

LUYBEN, W. L. (2006). *Distillation design and control using Aspen simulation.* Wiley-AIChE.

MARCOS, N., GUAY, M. and DOCHAIN, D. (2004a). Output feedback adaptive extremum seeking control of a continuous stirred tank bioreactor with monod's kinetics. *Journal of Process Control*, 14, 807–818.

MARCOS, N., GUAY, M., DOCHAIN, D. and ZHANG, T. (2004b). Adaptive extremum-seeking control of a continuous stirred tank bioreactor with haldane's kinetics. *Journal of Process Control*, 14, 317–328.

MCAVOY, T. and YE, N. (1994). Base control for the Tennessee Eastman problem. *Computers & Chemical Engineering*, 18, 383–413.

MERCANGÖZ, M. and DOYLE, F. J. (2007). Distributed model predictive control of an experimental four-tank system. *Journal of Process Control*, 17, 297–308.

MOORE, C. (1993). Selection of controlled and manipulated variables. W. Luyben, editor, *Practical Distillation Control*, Springer US. p. 140–177.

MORARI, M. (1990). Model predictive control : Multivariable control technique of choice in the 1990s ? *In Advances in Model-based Predictive Control.* Oxford University Press Inc, p. p22–37.

MORARI, M. and LEE, J. (1999). Model predictive control : past, present and future. *Computers & Chemical Engineering*, 23, 667–682.

NGUANG, S. K. and CHEN, X. D. (2000). Extremum seeking scheme for continuous fermentation processes described by an unstructured fermentation model. *Bioprocess Engineering*, 23, 417–420.

OLAJIRE, A. A. (2010). CO_2 capture and separation technologies for end-of-pipe applications–a review. *Energy*, 35, 2610–2628.

PRESTON, R. A. (1981). A computer model of the rectisol process using the aspen simulator.

QIN, S. J. and BADGWELL, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11, 733–764.

RANKE, G. (1977). Process and apparatus for the production of hydrogen and carbon dioxide. US Patent 4,050,909.

RANKE, G. and MOHR, V. (1985). The rectisol wash : new developments in acid gas removal from synthesis gas. *Acid and Sour Gas Treating Processes*, 80–111.

RAWLINGS, J. B. and STEWART, B. T. (2008). Coordinating multiple optimization-based controllers : New opportunities and challenges. *Journal of Process Control*, 18, 839–845.

RICKER, N. and LEE, J. (1995). Nonlinear model predictive control of the tennessee eastman challenge process. *Computers & Chemical Engineering*, 19, 961 – 981.

SCATTOLINI, R. (2009). Architectures for distributed and hierarchical model predictive control ,Äì a review. *Journal of Process Control*, 19, 723 – 731.

SCHNEIDER, G., ARIYUR, K. and KRSTIC, M. (2000). Tuning of a combustion controller by extremum seeking : a simulation study. *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on.* vol. 5, p. 5219–5223 vol.5.

SEKI, H. and NAKA, Y. (2008). Optimizing control of cstr/distillation column processes with one material recycle. *Industrial & Engineering Chemistry Research*, 47, 8741–8753.

STEWART, B. T., VENKAT, A. N., RAWLINGS, J. B., WRIGHT, S. J. and PANNOCCHIA, G. (2010). Cooperative distributed model predictive control. *Systems & Control Letters*, 59, 460–469.

TAN, Y., MOASE, W., MANZIE, C., NESIC, D. and MAREELS, I. M. Y. (2010). Extremum seeking from 1922 to 2010. *Control Conference (CCC), 2010 29th Chinese.* p. 14–26.

TITICA, M., DOCHAIN, D. and GUAY, M. (2003). Adaptive extremum seeking control of fed-batch bioreactors. *European Journal of Control*, 9, 618–631.

TYREUS, B. D. and LUYBEN, W. L. (1993). Dynamics and control of recycle systems. 4. ternary systems with one or two recycle streams. *Industrial & Engineering Chemistry Research*, 32, 1154–1162.

VAN OVERSCHEE, P. and DE MOOR, B. (1996). *Subspace identification for linear systems : theory, implementation, applications.* Kluwer academic publishers.

VENKAT, A., RAWLINGS, J. and WRIGHT, S. (2005). Stability and optimality of distributed model predictive control. *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on.* p. 6680–6685.

VENKAT, A. N., HISKENS, I. A., RAWLINGS, J. B. and WRIGHT, S. J. (2008). Distributed mpc strategies with application to power system automatic generation control. *Control Systems Technology, IEEE Transactions on*, 16, 1192–1206.

VENKAT, A. N., RAWLINGS, J. B. and WRIGHT, S. J. (2007). Distributed model predictive control of large-scale systems. *Assessment and Future Directions of Nonlinear Model Predictive Control*, Springer. p. 591–605.

WANG, H.-H. and KRSTIC, M. (2000). Extremum seeking for limit cycle minimization. *Automatic Control, IEEE Transactions on*, 45, 2432–2436.

WANG, H.-H., KRSTIC, M. and BASTIN, G. (1999). Optimizing bioreactors by extremum seeking. *International Journal of Adaptive Control and Signal Processing*, 13, 651–669.

WANG, H.-H., YEUNG, S. and KRSTIC, M. (2000). Experimental application of extremum seeking on an axial-flow compressor. *Control Systems Technology, IEEE Transactions on*, 8, 300–309.

WANG, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB®*. Springer Verlag.

WANG, L. and YOUNG, P. C. (2006). An improved structure for model predictive control using non-minimal state space realisation. *Journal of Process Control*, 16, 355 – 371.

WEISS, H. (1988). Rectisol wash for purification of partial oxidation gases. *Gas Separation & Purification*, 2, 171–176.

ZHANG, T., GUAY, M. and DOCHAIN, D. (2003). Adaptive extremum seeking control of continuous stirred-tank bioreactors. *AIChE journal*, 49, 113–123.

APPENDIX A

Distributed MPC matlab code

```
function dunew = DQDMC_mf(dxhat,yref,dd,uold,ybar)
%2/26/2013 9:09 AM
Np=500;
dunew=zeros(10,1);

yref1=yref(1:3);
yref2=yref(4:5);
yref3=yref(6:7);
yref4=yref(8:10);

dxhat1=dxhat(1:12);
dxhat2=dxhat(13:19);
dxhat3=dxhat(20:26);
dxhat4=dxhat(27:36);



uold1=uold(1:3);
uold2=uold(4:5);
uold3=uold(6:7);
uold4=uold(8:10);

persistent dunewk
if isempty(dunewk)
    dunewk=zeros(10,2);
end
dd=[dunewk;[dd(11) dd(11)]];

dd1=dd(4:11,:);
dd1=[dd1(:,1); dd1(:,2)];
```

```
dd2=[dd(1:3,:);dd(6:11,:)];dd2=[dd2(:,1); dd2(:,2)];
dd3=[dd(1:5,:);dd(8:11,:)];dd3=[dd3(:,1); dd3(:,2)];
dd4=[dd(1:7,:);dd(11,:)];dd4=[dd4(:,1); dd4(:,2)];
i=0;

ybar1=ybar(1:3);
ybar2=ybar(4:5);
ybar3=ybar(6:7);
ybar4=ybar(8:10);

%j1=[];
ddkong=zeros(10,1);


for j=1:10


    dunew1k=dqmc1(dxhat1,yref1,dd1,uold1,ybar1);
    dunew2k=dqmc2(dxhat2,yref2,dd2,uold2,ybar2);
    dunew3k=dqmc3(dxhat3,yref3,dd3,uold3,ybar3);
    dunew4k=dqmc4(dxhat4,yref4,dd4,uold4,ybar4);
    dunewk=[[dunew1k(1:3); dunew2k(1:2); dunew3k(1:2); dunew4k(1:3)] ...
    [dunew1k(4:6) dunew2k(3:4); dunew3k(3:4); dunew4k(4:6)]];
    dd=[dunewk;[dd(11) dd(11)]];


    dd1=dd(4:11,:);dd1=[dd1(:,1); dd1(:,2)];
    dd2=[dd(1:3,:);dd(6:11,:)];dd2=[dd2(:,1); dd2(:,2)];
    dd3=[dd(1:5,:);dd(8:11,:)];dd3=[dd3(:,1); dd3(:,2)];
    dd4=[dd(1:7,:);dd(11,:)];dd4=[dd4(:,1); dd4(:,2)];
    j2=norm(abs(dunewk(:,1)-ddkong));

    ddkong=dunewk(:,1);
    i=i+1;
```

```
end


dunew=ddkong;

end

%-----------------------------------------------------%
function dunew1  = dqmc1(dxhat,yref,dd,uold,ybar)
Num_y=3;
Np=500;
U=[50.3566685765720;100;77.5900919718307;22.1166376165571;1.89321771047563;...
0.574850448861118;44.7494717139745;-15.9088120280391;18.7553295034155;]';
Nc=2;
Num_u=3;
rs=rs_gen(yref,Np,Num_y);
dxhataug=[dxhat;ybar];
% eml.extrinsic('quadprog')
% dunew=double(zeros(18,1));
%%%%%%%%%%constraints%%%%%%%
dUmax=[100 75 100];
dUmin=-dUmax;
Umax=[50 50 50];
Umin=[-50 -25 -50];
%%%%%model%%%%%%%
Amt = [0.122953849808169,-0.946070996513701,-0.109278736566159,...]
Bmt = [-29.4327374071590,-0.0519540422363396,-0.000238618690118114,...];
Cmt = [0.0222915493767768,-0.0457691502076677,0.205340068384083,,...];
Dm=[0,0,0,0,0,0,0,0,0,0,0,...];
%Dm=zeros(9,10);
Am=Amt(1:12,1:12);
Gm=Bmt(1:12,4:11);
Bm=Bmt(1:12,1:3);
Cm=Cmt(1:3,1:12);

%%%%%%%%%%%%%% Control calculation%%%%%%%%
```

```
persistent F phi wy wdu kessi
if isempty(F)||isempty(phi)||isempty(wy)||isempty(wdu)||isempty(kessi)
[A,B,C,G]=dmcaug(Am,Bm,Cm,Gm,Num_y);
 F=F_gen(A,C,Np);
 wy1=[1 2 1];
wdu1=0.1*[1 1 1]; [wy wdu]= w_gen(wy1,wdu1,Np,Num_u,Nc,Num_y);
phi=phi_gen(A,B,C,Np,Nc);
kessi=phi_gen(A,G,C,Np,Nc);

end

[H,f,a,b]=qdmcform(F,phi,wy,wdu,dUmin,dUmax,Umin,Umax,uold,rs,dxhataug,dd,kessi);
dunewa=QDMC(H,f,a,b);
dunew1=dunewa;
end

%-----------------------------------------------------%
function dunew2  = dqmc2(dxhat,yref,dd,uold,ybar)
Np=500;
Num_y=2;
U=[50.3566685765720;100;77.5900919718307;22.1166376165571;1.89321771047563;...
0.574850448861118;44.7494717139745;-15.9088120280391;18.7553295034155;]';
Nc=2;
Num_u=2;
rs=rs_gen(yref,Np,Num_y);
dxhataug=[dxhat;ybar];
% eml.extrinsic('quadprog')
% dunew=double(zeros(18,1));
%%%%%%%%%constraints%%%%%%%
dUmax=[100 10];
dUmin=-dUmax;
Umax=[ 50 Inf ];
Umin=[ -50 -Inf ];
%%%%%model%%%%%%%
Amt = [0.122953849808169,-0.946070996513701,-0.109278736566159,...]
Bmt = [-29.4327374071590,-0.0519540422363396,-0.000238618690118114,...];
```

```
Cmt = [0.0222915493767768,-0.0457691502076677,0.205340068384083,,...];
Dm=[0,0,0,0,0,0,0,0,0,0,0,...];
%Dm=zeros(9,10);
Am=Amt(13:19,13:19);
Gm=Bmt(13:19,[1:3 6:11]);
Bm=Bmt(13:19,4:5);
Cm=Cmt(4:5,13:19);

%%%%%%%%%%%%%% Control calculation%%%%%%%%
persistent F phi wy wdu kessi
if isempty(F)||isempty(phi)||isempty(wy)||isempty(wdu)||isempty(kessi)
[A,B,C,G]=dmcaug(Am,Bm,Cm,Gm,Num_y);
 F=F_gen(A,C,Np);
 wy1=[1 1];
wdu1=0.1*[1 1]; [wy wdu]= w_gen(wy1,wdu1,Np,Num_u,Nc,Num_y);
phi=phi_gen(A,B,C,Np,Nc);
kessi=phi_gen(A,G,C,Np,Nc);

end

[H,f,a,b]=qdmcform(F,phi,wy,wdu,dUmin,dUmax,Umin,Umax,uold,rs,dxhataug,dd,kessi);
dunewa=QDMC(H,f,a,b);
dunew2=dunewa;
end

%----------------------------------------------%
function dunew3=dqmc3(dxhat,yref,dd,uold,ybar)
Np=500;
Num_y=2;
U=[50.3566685765720;100;77.5900919718307;22.1166376165571;1.89321771047563;...
0.574850448861118;44.7494717139745;-15.9088120280391;18.7553295034155;]';
Nc=2;
Num_u=2;
rs=rs_gen(yref,Np,Num_y);
dxhataug=[dxhat;ybar];
% eml.extrinsic('quadprog')
% dunew=double(zeros(18,1));
%%%%%%%%%%constraints%%%%%%%
dUmax=[ 100 100 ];
dUmin=-dUmax;
Umax=[ 50 50 ];
Umin=[ -50 -50 ];
%%%%%model%%%%%%%
Amt = [0.122953849808169,-0.946070996513701,-0.109278736566159,...]
Bmt = [-29.4327374071590,-0.0519540422363396,-0.000238618690118114,...];
Cmt = [0.0222915493767768,-0.0457691502076677,0.205340068384083,...];
Dm=[0,0,0,0,0,0,0,0,0,0,0,...];
%Dm=zeros(9,10);
Am=Amt(20:26,20:26);
Gm=Bmt(20:26,[1:5 8:11]);
Bm=Bmt(20:26,6:7);
Cm=Cmt(6:7,20:26);

%%%%%%%%%%%%%% Control calculation%%%%%%%%
persistent F phi wy wdu kessi
if isempty(F)||isempty(phi)||isempty(wy)||isempty(wdu)||isempty(kessi)
[A,B,C,G]=dmcaug(Am,Bm,Cm,Gm,Num_y);
 F=F_gen(A,C,Np);
 wy1=[1 1];
wdu1=0.1*[1 1 ]; [wy wdu]= w_gen(wy1,wdu1,Np,Num_u,Nc,Num_y);
phi=phi_gen(A,B,C,Np,Nc);
kessi=phi_gen(A,G,C,Np,Nc);

end

[H,f,a,b]=qdmcform(F,phi,wy,wdu,dUmin,dUmax,Umin,Umax,uold,rs,dxhataug,dd,kessi);
dunewa=QDMC(H,f,a,b);
dunew3=dunewa;
end

%---------------------------------------------------%
function dunew4=dqmc4(dxhat,yref,dd,uold,ybar)
```

```
Np=500;
Num_y=3;
U=[50.3566685765720;100;77.5900919718307;22.1166376165571;1.89321771047563;...
0.574850448861118;44.7494717139745;-15.9088120280391;18.7553295034155;]';
Nc=2;
Num_u=3;
rs=rs_gen(yref,Np,Num_y);
dxhataug=[dxhat;ybar];
% eml.extrinsic('quadprog')
% dunew=double(zeros(18,1));
%%%%%%%%%%constraints%%%%%%%
dUmax=[100 50 50];
dUmin=-dUmax;
Umax=[50 Inf Inf];
Umin=[-50 -Inf -Inf];
%%%%%model%%%%%%%
Amt = [0.122953849808169,-0.946070996513701,-0.109278736566159,...]
Bmt = [-29.4327374071590,-0.0519540422363396,-0.000238618690118114,...];
Cmt = [0.0222915493767768,-0.0457691502076677,0.205340068384083,...];
Dm=[0,0,0,0,0,0,0,0,0,0,0,...];
%Dm=zeros(9,10);
Am=Amt(27:36,27:36);
Gm=Bmt(27:36,[1:7 11]);
Bm=Bmt(27:36,8:10);
Cm=Cmt(8:10,27:36);

%%%%%%%%%%%%%% Control calculation%%%%%%%%
persistent F phi wy wdu kessi
if isempty(F)||isempty(phi)||isempty(wy)||isempty(wdu)||isempty(kessi)
[A,B,C,G]=dmcaug(Am,Bm,Cm,Gm,Num_y);
 F=F_gen(A,C,Np);
 wy1=[5 1 1];
wdu1=0.1*[1 1 5]; [wy wdu]= w_gen(wy1,wdu1,Np,Num_u,Nc,Num_y);
phi=phi_gen(A,B,C,Np,Nc);
kessi=phi_gen(A,G,C,Np,Nc);
```

```
end

[H,f,a,b]=qdmcform(F,phi,wy,wdu,dUmin,dUmax,Umin,Umax,uold,rs,dxhataug,dd,kessi);
dunewa=QDMC(H,f,a,b);
dunew4=dunewa;
end

%%%%%%%%%kessi%%%%%%%%%%%%
function kessi=kessi_gen(A,C,G,Np)
[n,m]=size(C*G);
kessi=zeros(Np/10*n,m);
  for i=1:10:Np
        kessi((i-1)/10*n+1:(i-1)/10*n+n,1:m)=C*A^(i-1)*G;
  end
end
%%%%%%%%%%%%%%wy%%%%%%%%%%
function [wy wdu]=w_gen(A,B,Np,Num_u,Nc,Num_y)
    wy=zeros(Num_y*Np/10,Num_y*Np/10);
    for i=1:10:Np
        for j=1:Num_y
            wy((i-1)/10*Num_y+j,(i-1)/10*Num_y+j)=A(j);
        end
    end
wdu=diag([B B]);
end
%%%%%%%%%%%%F_gen%%%%%%%%%%%
function F=F_gen(A,C,Np)
[n,m]=size(C*A);
F=zeros(n*Np/10,m);
for i=1:10:Np
    F((i-1)/10*n+1:(i-1)/10*n+n,:)=C*A^i;
end
end
%%%%%%%%%%%%%%%%%%%phi_gen%%%%%%%%%%%%%%%%
function phi=phi_gen(A,B,C,Np,Nc)
```

```
[n,m]=size(C*B);

phi=zeros(Np/10*n,Nc*m);


  for i=1:10:Np
       phi((i-1)/10*n+1:(i-1)/10*n+n,1:m)=C*A^(i-1)*B;
  end

  for j=2:Nc
  phi(:,(j-1)*m+1:(j-1)*m+m)=[zeros((j-1)*n,m); phi(1:(Np/10-(j-1))*n,1:m)];
  end

end
%%%%%%%%%%%%model Augmentation%%%%%%%%%%%%
function [A,B,C,G]=dmcaug(Am,Bm,Cm,Gm,Num_y)
n1=length(Am);
A=[Am zeros(n1,Num_y); Cm*Am eye(Num_y)];
B=[Bm;Cm*Bm];
C=[zeros(Num_y,n1) eye(Num_y)];
G=[Gm; Cm*Gm];

end
%%%%%%%%RS%%%%%%%%%
function rs=rs_gen(yref,Np,Num_y)
rs=zeros(Num_y*Np/10,1);
for i=1:10:Np
    rs ((i-1)/10*Num_y+1:(i-1)/10*Num_y+Num_y)=yref;
end
end
%%%%%%%%%%%%%%%%%%%%%Quadratic
%%%%%%%%%%%%%%%%%%%%%Formulation%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [H,f,a,b]=qdmcform(F,phi,wy,wdu,dUmin,dUmax,Umin,Umax,uold,rs,xhat,dd,kesi)
Num_u=length(uold);

H=2*(phi'*wy*phi+wdu);
```

```
f=-2.*(phi'*wy*(rs-F*xhat-kesi*dd));
a=[eye(Num_u) zeros(Num_u,Num_u);zeros(Num_u,Num_u) eye(Num_u); -eye(Num_u)...
zeros(Num_u,Num_u);zeros(Num_u,Num_u) -eye(Num_u);...
eye(Num_u) zeros(Num_u,Num_u);eye(Num_u) eye(Num_u,Num_u);...
-eye(Num_u) zeros(Num_u,Num_u);-eye(Num_u) -eye(Num_u,Num_u) ];
b=[dUmax';dUmax';  -dUmin';-dUmin'; Umax'-uold;Umax'-uold;...
 -Umin'+uold; -Umin'+uold];

end
```

```
%%%%%%%%Optimization%%%%%%%%%%
function dunew1=QDMC(H,f,a,b)
options=optimset('Algorithm','active-set','Display','off');
[dunew1,J]= quadprog(H,f,a,b,[],[],[],[],[],options);
end
```