POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Optimisation en temps réel avec projection sur contraintes actives : application à une usine de production du mononère[i.e monomère] d'acétate de vinyle
Auteur: Author:	Yanick Dumais
Date:	2006
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Dumais, Y. (2006). Optimisation en temps réel avec projection sur contraintes actives : application à une usine de production du mononère[i.e monomère] d'acétate de vinyle [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7612/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/7612/
Directeurs de recherche: Advisors:	Michel Perrier, & Balasubrahmanyan Srinivasan
Programme: Program:	Non spécifié

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

OPTIMISATION EN TEMPS RÉEL AVEC PROJECTION SUR CONTRAINTES ACTIVES : APPLICATION À UNE USINE DE PRODUCTION DU MONONÈRE D'ACÉTATE DE VINYLE

YANICK DUMAIS
DÉPARTEMENT DE GÉNIE CHIMIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE CHIMIQUE)
FÉVRIER 2006

© Yanick Dumais, 2006.

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-16777-9
Our file Notre référence
ISBN: 978-0-494-16777-9

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Université de Montréal

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

OPTIMISATION EN TEMPS RÉEL AVEC PROJECTION SUR CONTRAINTES ACTIVES : APPLICATION À UNE USINE DE PRODUCTION DU MONOMÈRE D'ACÉTATE DE VINYLE

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

M. KLVANA Danilo, Ph.D., président
M. PERRIER Michel, Ph.D., membre et directeur de recherche
M. SRINIVASAN Balasubrahmanyan, Ph.D., membre et codirecteur de recherche
M.GENDRON Sylvain, Ph.D., membre

Avant d'admettre l'absurde,

on épuise toutes les solutions.

Arthur Conan Doyle

Remerciements

Je veux tout d'abord remercier mon directeur de projet, Michel Perrier, pour ses conseils et son aide tout le long de mon cheminement dans mon projet. Je le remercie de m'avoir proposé et dirigé dans un sujet de recherche dans un domaine appliqué. Ensuite, je suis reconnaissant envers mon codirecteur, Bala Srinivasan, pour son aide, ses conseils, pour sa vision du projet et pour tout le temps qu'il m'a consacré.

Je remercie également toutes les personnes qui ont travaillé avec moi sur la simulation du monomère d'acétate de vinyle, pour m'avoir aidé à la développer et à mieux la comprendre.

Je veux aussi remercier tous mes confrères de travail, plus précisément Bertrand, pour ses plaisanteries et son aide pendant mes cours; Danielle, pour toutes les corrections et pour sa touche féminine qui a amélioré les conditions de vie dans le bureau; Moncef et Nicolas pour leurs longues discussions enrichissantes; Tommaso, pour les nombreuses bières et toutes les distractions qu'il m'a fournies Guillaume pour les nombreuses corrections et les activités physiques qui m'ont permis de garder un semblant de forme et Charles pour sa place dans le bureau.

Finalement, j'exprime ma gratitude envers tous mes amis et ma famille pour leur support et leur encouragement qui m'ont permis de terminer cette maîtrise.

Résumé

Le premier objectif de cette maîtrise est de créer un algorithme pour optimiser en temps réel une usine opérant sous contraintes. La procédure développée est l'optimisation avec projection sur contraintes actives. Elle est constituée d'une partie pour optimiser et d'une section pour la projection sur les contraintes. La première section est une fonction qui prend la plus grande valeur entre le gradient et un paramètre k pour déplacer le système. Lorsqu'une contrainte est rencontrée, la seconde section de l'algorithme entre en jeu. Cette méthode utilise une des variables de l'optimisation pour garder cette contrainte active. Une fois qu'un optimum pour la contrainte active est rencontré, la contrainte est relâchée pour vérifier si un autre chemin permet de poursuivre l'optimisation. Si une autre contrainte permet d'améliorer la fonction objectif, elle est à son tour considérée active. La convergence de cet algorithme est démontrée dans ce mémoire.

Le principal avantage de cette méthode est que si l'optimum est à la contrainte, il est possible de se maintenir exactement à ce point. Un second avantage est que sa vitesse d'évaluation du gradient et son application sont rapides. Le principal désavantage de cette méthode est qu'il y a uniquement une variable qui est manipulée à

la fois, ce qui peut nuire à la vitesse d'exécution si plusieurs variables d'entrées pour l'optimisation sont présentes.

Le second objectif de cette maîtrise est d'optimiser l'usine de production du monomère d'acétate de vinyle (MAV). L'optimisation dans le mode dynamique et statique est effectuée. En plus le pairage des variables et des contraintes active fait que les contraintes sont respectées. En second lieu, l'optimisation dynamique du procédé de production du MAV avec l'algorithme de projection sur contraintes actives fonctionne bien. Le fait d'associer une variable à une contrainte active par un contrôleur ramène le système à la contrainte rapidement.

L'optimisation statique et dynamique à une variable permet d'améliorer la fonction objectif, le débit de MAV produit, de l'ordre de 50 %. Les valeurs des variables manipulées sont exactement les mêmes en dynamique et en statique, mais la fonction objectif en mode dynamique est légèrement inférieure en raison de la dérive de la simulation. L'optimisation à deux variables en mode statique a produit une amélioration de 61.4 %, tandis que l'optimisation dynamique produit des résultats d'environ 50 %. La disparité entre les modes statique et dynamique s'explique par la dérive de la simulation. De plus, la marge de sécurité dans l'optimisation dynamique ajoute à cette différence. Finalement, l'optimisation statique à trois variables permet d'atteindre une amélioration de 81.6 % comparativement au point d'opération initial.

Abstract

The first objective of this thesis is the creation of an algorithm for the model-free real-time optimization of a constrained problem. The procedure proposed is extremum seeking with projection on active constraints. They are two sections in this algorithm: the first section is the optimization itself and the second is the hold of constraints that must be kept active. The optimization section takes the larger term between the gradient and a fixed criterion to move closer to the optimum, one variable at a time. The gradient is estimated whit the excitation provided from the pervious step. If a constraint is met, the second part of the algorithm starts where a variable of the optimization is used to keep this constraint active. When an optimum is found, the constraint is released and the system checks if another direction can improve the objective function. If this new direction is located on another constraint, the variable is used to hold this constraint. The optimization is completed when there are no other possible directions. The proof of the convergence of this algorithm is given in this thesis.

The principal advantage of this algorithm is that the exact optimum will be found if the optimum lies on the constraint. Another advantage of this algorithm is its speed in evaluating the gradient and its application. The principal disadvantage of this

procedure is that it uses only one variable at a time. This can reduce the speed of execution if many variables must be considered.

The second objective is the optimization of vinyl acetate production in a plant-wide computer simulation. Static and dynamic optimization were performed. The tuning between the manipulated variable and the constraint works well. The dynamic optimization of vinyl acetate production with the algorithm of extremum seeking with projection on active constraints gave good results. The association of the manipulated variable and the constrains with a controller works very well and rapidly.

The static and dynamic optimization with one manipulated variable result in an increase of 50 % of the objective function. In the context of vinyl acetate production, an increase of 0.417 kmol/min can be achieved by manipulating the evaporator pressure. The optimal value of the manipulated variables are the same in the dynamic case as in the static one, but the objective function in the dynamic optimization is smaller because of a simulation drift. The static optimization with two variables (evaporator pressure and reactor exit temperature) give an improvement of 61.4% and the dynamic optimization gives an improvement of approximately 50%. The difference between these two values is explained by drift of the dynamic simulation and the security margin on the constraints. Finally, the static optimization with three variables (evaporator pressure, reactor exit temperature and recirculation ratio of C_2H_6) gives an improvement of 81.6% compared to normal operating conditions.

Table des matières

Dédicace IV

Remerciements V

Résumé VI

Abstract VIII

Table des matières X

Liste des figures XIII

Liste des tableaux XVI

Liste des symboles XVII

Liste des annexes XXI

CHAPITRE 1 Introduction 1

1.1. Problématique 1

1.2. Objectifs 4

1.3. Organisation du mémoire 5

CHAPITRE 2 Revue de littérature 7

2.1. Concepts mathématiques 7

2.1.1. Formulation du problème d'optimisation sans contraintes 7

2.1.2. Formulation du problème d'optimisation avec contraintes 8

2.2. Types de contraintes 9

2.3. Méthodes d'optimisation proposées dans la littérature 10

2.3.1. IMC (Pomerleau et al., 2003 b). 10

2.3.2. Méthode des perturbations (Krstic et Wang, 2000). 13

2.3.3. Fonction barrière (Vassiliadis et Floudas 1997) 15

CHAPITRE 3 Méthodes développées 17

3.1. Optimisation sans contrainte 17

3.1.1. Formulation d'algorithme avec l'information du gradient. 17

3.1.2. Optimisation en temps réel avec estimation du gradient 18

3.1.3. Algorithme d'implantation en mode discret 22

3.2. Optimisation avec contraintes 25

3.2.1. Analyse de l'algorithme avec contraintes 25

3.2.2. Optimisation en temps réel avec estimation du gradient 29

3.2.2. Implantation de la projection utilisant deux niveaux de contrôleur. 30

3.2.3. Algorithme avec contrainte en modes discret 33

3.3. Comparaison entre les méthodes 38

3.3.1. Comparaison entre IMC et la méthode OTRPCA 38

3.3.2. Comparaison entre la méthode des perturbations et la méthode de projection sur contraintes actives 39

3.3.3. Fonction barrière et la méthode OTRPCA 40

CHAPITRE 4 Application et résultats 42

4.1. Description du procédé 42

4.2. Choix de la fonction objectif et des variables manipulées 44

4.2.1. Choix de la fonction objectif 44

4.2.2. Choix des variables manipulées 45

4.2.2.1. Type de variables 46

4.2.2.2. Plan factoriel 46

4.2.2.3. Méthode du gradient 51

4.3. Explication du choix de variables pour l'optimisation 53

4.4. Résultats d'optimisation sur le MAV 55

4.4.1. Optimisation statique 55

4.4.2. Optimisation dynamique 62

CHAPITRE 5 Conclusions et Recommandations 74

Conclusions 74

Recommandations 76

Bibliographie 78

Annexes 86

Liste des figures

Figure 2- 1 : Types de contraintes 10

Figure 2- 2 : Structure de l'algorithme d'optimisation IMC 12

Figure 2- 3 : Structure de l'« extremum seeking » de la méthode des perturbations. .. 14

Figure 3- 1 : Algorithme sans contrainte 23

Figure 3- 2 : Algorithme avec contrainte (1/2) 33

Figure 3- 3 : Algorithme avec contrainte (2/2) 34

Figure 3- 4 : Problème causé par l'ordre des variables (2 variables) 35

Figure 3- 5 : Solution au problème causé par l'ordre des variables (2 variables) 36

Figure 3- 6 : Problème causé par l'ordre des variables (3 variables) 36

Figure 3- 7 : Solution au problème causé par l'ordre des variables (2 variables) 37

Figure 4- 1 : Schéma de l'usine de production du MAV (Chen, Rong, 2003) 43

Figure 4- 2 : Variables manipulées 46

Figure 4- 3 : Valeur des gradients pour plusieurs points 52

Figure 4- 4 : Valeur moyenne des gradients 52

Figure 4- 5 : Optimisation à une variable. 57

Figure 4- 6 : Dérive de la fonction objectif. 64

Figure 4- 7 : Variable manipulée 5 pour l'optimisation à une variable 65

Figure 4- 8 : Fonction objectif pour l'optimisation à une variable 65

Figure 4- 9 : Solution au problème dynamique (2 variables) 66

Figure 4- 10 : Fonction objectif pour l'optimisation avec l'ordre 5-7 67

Figure 4- 11 : Variable manipulée 5 pour l'optimisation avec l'ordre 5-7 67

Figure 4- 12 : Variable manipulée 7 pour l'optimisation avec l'ordre 5-7 68

Figure 4- 13 : Contrainte du procédé pour l'optimisation avec l'ordre 5-7 68

Figure 4- 14 : Fonction objectif pour l'optimisation avec l'ordre 7-5 70

Figure 4- 15 : Variable manipulée 5 pour l'optimisation avec l'ordre 7-5 71

Figure 4- 16 : Variable manipulée 7 pour l'optimisation avec l'ordre 7-5 71

Figure 4- 17 : Contrainte 41 pour l'optimisation avec l'ordre 7-5 72

Figure B- 1 : Valeur normal en fonction des résidus 95

Figure B- 2 : Les résidus en fonction du numéro du cas 96

Figure B- 3 : Les résidus effacés en fonction des résidus 97

Figure B- 4 : Les résidus en fonction des valeurs prédites 98

Figure E- 1 : Fonction objectif pour l'ordre DT 102

Figure E- 2 : Taux de dilution pour l'ordre DT 103

Figure E- 3 : Température pour l'ordre DT 103

Figure E- 4 : Contrainte pour l'ordre DT 104

Figure E- 5 : Fonction objectif pour la méthode des perturbations 106

Figure E- 6 : Taux de dilution pour la méthode des perturbations 106

Figure E- 7 : Température pour la méthode des perturbations 107

Figure E- 8 : Contrainte pour la méthode des perturbations 107

Figure E- 9 : Méthode de projection avec du bruit 110

Figure E- 10 : Méthode de perturbation avec bruit 110

Liste des tableaux

Tableau 4.1 : Variables utilisées pour le plan factoriel 49

Tableau 4.2 : Liste des effets 49

Tableau 4.3 : Liste des effets confondus 50

Tableau 4.4 : Variable 5 gain 0.01 delta 0.001 57

Tableau 4.5 : Variables 5 et 7 Gain 0.01 Delta 0.001 58

Tableau 4.6 : Variables 5 et 7 avec glissement Gain 0.01 Delta 0.001. 59

Tableau 4.7 : Variables 5, 7 et 18 Gain 0.01 Delta 0.001 60

Tableau 4.8 : Variables 5, 7 et 18 avec relâchement de contrainte Gain 0.01 Delta 0.001 61

Tableau B- 1 : Listes des expériences 94

Liste des symboles

Acronymes

IMC	Internal Model Controller
MAV	Monomère d'acétate de vinyle
OTRPCA	Optimisation en temps réel avec projection sur les contraintes actives
ES	Recherche d'extremum "extremum seeking"

Alphabet romain

e	Vecteur de déplacement
F	Système renfermant les équations d'égalité.
G	Matrice des gain
g	Défini dans l'équation 3-8
H	Modèle du système
H_m	Modèle simplifié du système
h	État hybride
k_i	Valeurs propres
L	Lagrangien
M	Défini dans l'équation 3-8
m	Nombre de variables manipulées

P	Projection sur contraintes actives
P	Modèle des contraintes d'inégalité
P_m	Modèle simplifié des contraintes d'inégalité
P_{max}	Valeur maximale des contraintes d'inégalité
Q	Modèle des contraintes d'égalité
Q_{eq}	Valeurs des équations d'égalité
Q_m	Modèle simplifié des contraintes d'égalité
q_s	État discret indiquant la direction
S	Contient les équations des contraintes d'inégalité
T	Temps prédéterminé des changements
u	Entrées
V	Fonction de Lyapunov
x	Variables d'état

Alphabet grec

α	Gain
β	Gain d'adaptation
ε	Marge pour éviter un changement infini
ε_y	Erreur reliée au modèle
ε_v	Erreur reliée aux contraintes inégalité
ε_w	Erreur reliée aux contraintes égalité

ϕ	Fonction objectif doublement différentiable de façon continue; un scalaire pleuvant être minimisé
ϕ_u	$\frac{\partial \phi}{\partial u}$
ϕ_x	$\frac{\partial \phi}{\partial x}$
γ	Pas minimum (Dans le OTRPCA)
γ	Facteur qui donne la distance à la contrainte (Dans la fonction barrière)
λ	Multiplicateur de Lagrange pour les équations d'égalité
μ	Multiplicateur de Lagrange pour les équations d'inégalité

Indices

$a*$	Valeur optimale
a_u	a selon les entrées
a_x	a selon les états
a^T	Transposée de a
Δa	Différence de a
$\dot{a}$	Dérivée représentée par $\frac{da}{dt}$
a_0	Valeur initiale de a

$\bar{a}$	Contraintes actives
$\tilde{a}$	Minimisation du gradient
a^{+}	Pseudo-inverse
a^{-1}	Inverse

Liste des annexes

Annexe A : Le procédé 86

Annexe B : Résultats et analyses des résidus du plan factoriel 93

Annexe C : Propriété de la matrice P la matrice de projection 98

Annexe D : Les contrôleurs 99

Annexe E : Comparaison numérique de deux méthodes d'optimisation 100

CHAPITRE 1 Introduction

1.1. Problématique

L'augmentation de la puissance des ordinateurs et des outils de calcul au cours des dernières années facilite la création de simulations dynamiques de procédés chimiques dans leur totalité. Celles-ci permettent de mieux comprendre les interactions entre les différentes variables intervenant dans un procédé. Ces simulations permettent également d'appliquer des méthodes d'optimisation plus complexes pour simuler des usines complètes avec un risque nul et à très faible coût.

Il existe plusieurs types d'optimisation pouvant être effectués sur un procédé. Il y a tout d'abord les optimisations en mode hors ligne tel que celle de Newton, de quasi-Newton et celle de la sécante (Himmelblau, 1988). Ensuite, il y a plusieurs types de méthodes en ligne. La plus conventionnelle catégorie de méthodes d'optimisation en ligne consiste à réduire le gradient à zéro dans la direction désirée afin de se situer à un point optimum. Un résumé des différentes méthodes qui sont disponibles dans la littérature pour réduire le gradient est présenté dans l'article Sternby 1980.

Il existe une autre façon de classer les méthodes d'optimisation soit avec ou sans modèle. Un exemple d'optimisation sans modèle est l'optimisation avec le gradient. Une méthode se démarque davantage, la méthode des perturbations, qui est robuste au bruit (Krstic et Wang, 2000). La seconde stratégie d'optimisation consiste à utiliser un modèle du procédé. Ce modèle peut être très simple et même représenter uniquement une petite partie du procédé, comme dans la méthode où le point d'opération est estimé à l'aide d'une parabole pour ensuite minimiser le gradient de cette parabole. Dans d'autres cas, le modèle peut être très complexe et représenter le plus parfaitement possible le procédé de façon linéaire ou non. Un bon exemple de l'utilisation des modèles pour l'optimisation est celui réalisé sur le procédé Tennesse Eastman (Golshan et al., 2005). Finalement, une autre méthode d'optimisation est de combiner les deux techniques précédentes pour optimiser le procédé. Pour ce faire, elle utilise le gradient pour se déplacer vers l'optimum et ensuite adapter un modèle autour de ce point. Un bon exemple pour cette méthode est l'algorithme présenté par Guay et al. (2005).

Un autre aspect qui permet de différencier les méthodes d'optimisation est la prise en compte ou non par l'algorithme des contraintes. La majorité des méthodes d'optimisations ne se soucie pas des contraintes, mais une méthode peut être insérée afin de les considérer. La méthode la plus utilisée dans ce cas est la fonction barrière. Un travail récent sur ce sujet est présenté par Vassiliadis et Floudas (1997). La méthode IMC est une autre méthode utilisée pour considérer les contraintes (Desbiens et Shook,

2003, Pomerleau et al., 2003 b). Toutefois, l'utilisation de ces deux méthodes ne permet pas d'être exactement à l'optimum si celui-ci se situe sur une contrainte

Dans le cadre de ce mémoire, une nouvelle technique pour considérer les contraintes est présentée. C'est la méthode de la projection sur contraintes actives, qui est en réalité une optimisation conventionnelle de type recherche d'extremum ,"extremum seeking", ES. Par contre, lorsqu'une contrainte est atteinte, le point d'opération glisse sur cette contrainte. Cette méthode est décrite et comparée à d'autres méthodes présentées dans la littérature aux chapitres 2 et 3 de ce mémoire. La méthode développée ici peut se situer sur tous les points du domaine réalisable, donc elle peut être exactement sur l'optimum. En somme, cette méthode est une méthode en ligne sans modèle qui considère les contraintes.

Les méthodes d'optimisation sont un sujet très intéressant. Toutefois, il faut également se pencher sur les simulations des procédés, sur lesquelles s'appliquent les algorithmes d'optimisation. Une des premières simulations à l'échelle industrielle porte le nom de Tennesse Eastman Project. Elle a été réalisée par Downs (1993). Quelques années plus tard, d'autres simulations ont été présentées, soit : Shell Distillation Column (Banerjee et Pranob, 1995), monomère d'acétate de vinyle (Chen et Rong, 2003), Alstom Gaseifier (CHIN, 2003), Dimethyl Ether (VASBINDER, 2003). Une simulation a même été réalisée pour modéliser une usine de pâte et papier dans son

entier (Castro, 2004). À ce jour, une très grande quantité de simulations sont disponibles dans plusieurs domaines reliés au génie chimique.

La simulation retenue dans le cadre de ce mémoire est le procédé de synthèse du monomère d'acétate de vinyle (MAV), qui appartient au du groupe DuPont. Elle regroupe plusieurs opérations unitaires couramment utilisées dans le génie chimique. Cette simulation a été réalisée pour la compagnie Dupont par Luyben et Tyreus (1998), où une commande de type classique est implantée. Plus récemment, McAvoy et Chen (2003) ont réécrit cette simulation en langage C pour la rendre accessible au public comme banc d'essai afin d'appliquer des stratégies de contrôles et des méthodes d'optimisations perfectionnées. Naturellement, quelques changements sur les valeurs du procédé ont été réalisés afin de protéger la confidentialité des usines de DuPont.

1.2. Objectifs

Le premier objectif de ce travail est de développer, d'analyser et d'expérimenter un algorithme pour l'optimisation sous contraintes en ligne et sans modèle dans les modes statique et dynamique. Le second objectif est d'optimiser le procédé de production du monomère d'acétate de vinyle qui est une fonction non linéaire et multivariable.

Pour atteindre ce premier objectif, des descriptions et des comparaisons sont effectuées entre la méthode proposée et certaines méthodes suggérées dans la littérature. Ensuite, une analyse de la convergence est effectuée sur l'algorithme proposé. Pour réaliser le second objectif de cette maîtrise, un choix des variables influençant la production de MAV est réalisé. Ce tri est effectué selon deux méthodes : un plan factoriel et une comparaison des gradients. Ensuite, à l'aide de la méthode proposée, l'optimisation de ce procédé est réalisée en mode statique et dynamique.

1.3. Organisation du mémoire

Le chapitre 2 décrit quelques principes nécessaires pour l'optimisation en mode statique et dynamique. Deux méthodes utilisées dans la littérature pour l'optimisation sous contrainte sont également présentées soit la méthode IMC (internal model control) et la méthode de perturbation associée à la méthode des barrières logarithmiques.

Le chapitre 3 contient la description de l'algorithme proposé dans ce mémoire, soit la méthode d'optimisation en temps réel avec projection sur les contraintes actives (OTRPCA). Ensuite, le OTRPCA est comparé aux algorithmes présentés dans le chapitre 2.

Le chapitre 4 renferme la description du procédé où la méthode OTRPCA est implantée comme sujet d'étude, soit le procédé de production du MAV. Ensuite, une

description de la procédure pour le choix des variables utilisées lors de l'optimisation est présentée. Finalement, les données de l'optimisation du système statique et dynamique sont exposées et commentées.

Le chapitre 5 contient un résumé et une analyse des principaux résultats de cette maîtrise. Une liste de recommandations suit afin de proposer des travaux futurs sur l'OTRPCA et sur la simulation du procédé de production du MAV.

CHAPITRE 2 Revue de littérature

2.1. Concepts mathématiques

Avant de se diriger dans le cœur du problème, il est utile de présenter quelques concepts mathématiques. Ainsi, la formulation du problème d'optimisation sans contraintes et avec contraintes sont traitées dans la section 2.1.1 et 2.1.2.

2.1.1. Formulation du problème d'optimisation sans contraintes

Le problème d'optimisation sans contrainte peut être représenté par :

$$\min_{u} \phi(u); \tag{2.1}$$

où ϕ est doublement différentiable de façon continue et donne un scalaire qui peut être minimisé, u sont les entrées. u^* représente l'entrée qui obtient la valeur optimale de ϕ et ϕ^* est la valeur optimale de ϕ. La notion a_b représente $\frac{\partial a}{\partial b}$. La condition nécessaire d'optimalité peut s'écrire :

$$\phi_u(u^*) = 0; \tag{2.2}$$

2.1.2. Formulation du problème d'optimisation avec contraintes

Voici la formulation d'une optimisation avec contraintes d'égalité et d'inégalité :

$$\min_{u} \phi(x,u); \tag{2.3}$$

$$F(x,u)=0; \tag{2.4}$$

$$S(x,u)\leq 0; \tag{2.5}$$

où ϕ est la fonction scalaire qui doit être minimisée, x représente les variables supplémentaires, u représente les entrées, S contient les équations des contraintes d'inégalité et F renferme les équations d'égalité.

Le Lagrangien est défini par :

$$L(x,u,\lambda,\mu)=\phi(x,u)+\lambda^T F(x,u)+\mu^T S(x,u); \tag{2.6}$$

où $\lambda \neq 0$ est le multiplicateur de Lagrange des équations d'égalité, tandis que $\mu \geq 0$ est le multiplicateur de Lagrange des équations d'inégalité. Les conditions nécessaires d'optimalité avec le Lagrangien sont:

$$L_u=\phi_u+\lambda^T F_u+\mu^T S_u=0; \tag{2.7}$$

$$L_x=\phi_x+\lambda^T F_x+\mu^T S_x=0 \quad \Rightarrow \quad \lambda^T=-(\phi_x+\mu^T S_x)F_x^{-1}; \tag{2.8}$$

$$L_\lambda=F=0; \tag{2.9}$$

$$\mu^T S=0 \quad \Rightarrow \quad \mu_j=0 \quad ou \quad S_j=0; \tag{2.10}$$

Il est important de noter que les multiplicateurs de Lagrange μ sont différents de zéro lorsque la contrainte correspondante est active et est égale à zéro lorsqu'elle n'est pas active, donc $\mu^T S = 0$. Noter que $\mu^T \geq 0$, $S \leq 0$ et $\mu^T S = \sum_{j=1}^{\zeta} \mu_j S_j = 0$ s'applique à tous les termes $\mu_j S_j$ de la sommation qui doivent être égaux à zéro.

2.2. Types de contraintes

Deux types de contraintes dans les procédés sont présents dans la plupart des applications. Le premier type de contraintes sont les bornes qui représentent le maximum et le minimum des variables manipulées. Ces contraintes sont fonctions d'une seule variable. Elles sont faciles à considérer dans un algorithme d'optimisation. Il suffit de mettre un saturateur sur les variables pour que ces contraintes soient toujours respectées. La seconde catégorie de contraintes sont les contraintes d'états. Ces contraintes sont reliées aux bornes des valeurs de sorties. Elles peuvent varier selon plusieurs variables manipulées. Elles sont difficiles à tenir en compte lors d'une optimisation multivariable. La Figure 1.1 illustre les deux types de contraintes dans un espace à deux variables.

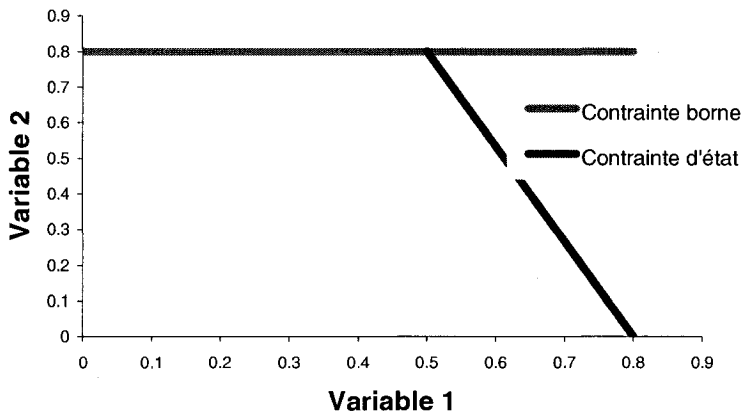

Figure 2- 1 : Types de contraintes

2.3. Méthodes d'optimisation proposées dans la littérature

Il reste maintenant plus qu'à s'intéresser aux méthodes d'optimisation proposées dans la littérature. Dans cette section, trois méthodes d'optimisation sont présentées, soit la méthode IMC, la méthode des perturbations et la fonction barrière. Ces descriptions se révèlent utile pour la comparaison entre la méthode OPRTPCA et ces méthodes présentées au chapitre 3.

2.3.1. IMC (Pomerleau et al., 2003 b).

Le système est défini :

$$Y=H(u). \tag{2.11}$$

L'optimisation est :

$$J = f(y,u). \tag{2.12}$$

Sous les contraintes suivantes :

$$P(u) \leq P_{max}; \tag{2.13}$$

$$Q(u)=Q_{eq}; \tag{2.14}$$

La méthode IMC est un algorithme d'optimisation statique. Elle permet d'optimiser un système fortement non linéaire à l'aide d'un modèle simplifié qui peut même être linéaire. Avec ce modèle simplifié, soit H_M, P_M et Q_M, il faut ensuite calculer la fonction objectif et les contraintes. Le système simplifié est alors optimisé. Un filtre est ajouté pour améliorer la robustesse de l'algorithme en le ralentissant. Ensuite le système, soit H, P et Q, est utilisé pour estimer les termes d'erreurs du système ε_y, ε_v et ε_w, comme le montre la Figure 2-2. L'erreur est ajoutée aux valeurs et le système simplifié est optimisé. Ensuite, des itérations sont effectuées jusqu'à ce que le régime permanent soit atteint.

Le calcul des fonctions objectif et des contraintes pour le pas suivant devient :

$$J_C(k)=f(Y_M(k)+ \varepsilon_y(k-1),U'(k)); \tag{2.15}$$

$$V_M(k) \leq V_{max} - \varepsilon_v(k-1); \tag{2.16}$$

$$W_M(k)=W_{eq} - \varepsilon_w(k-1). \tag{2.17}$$

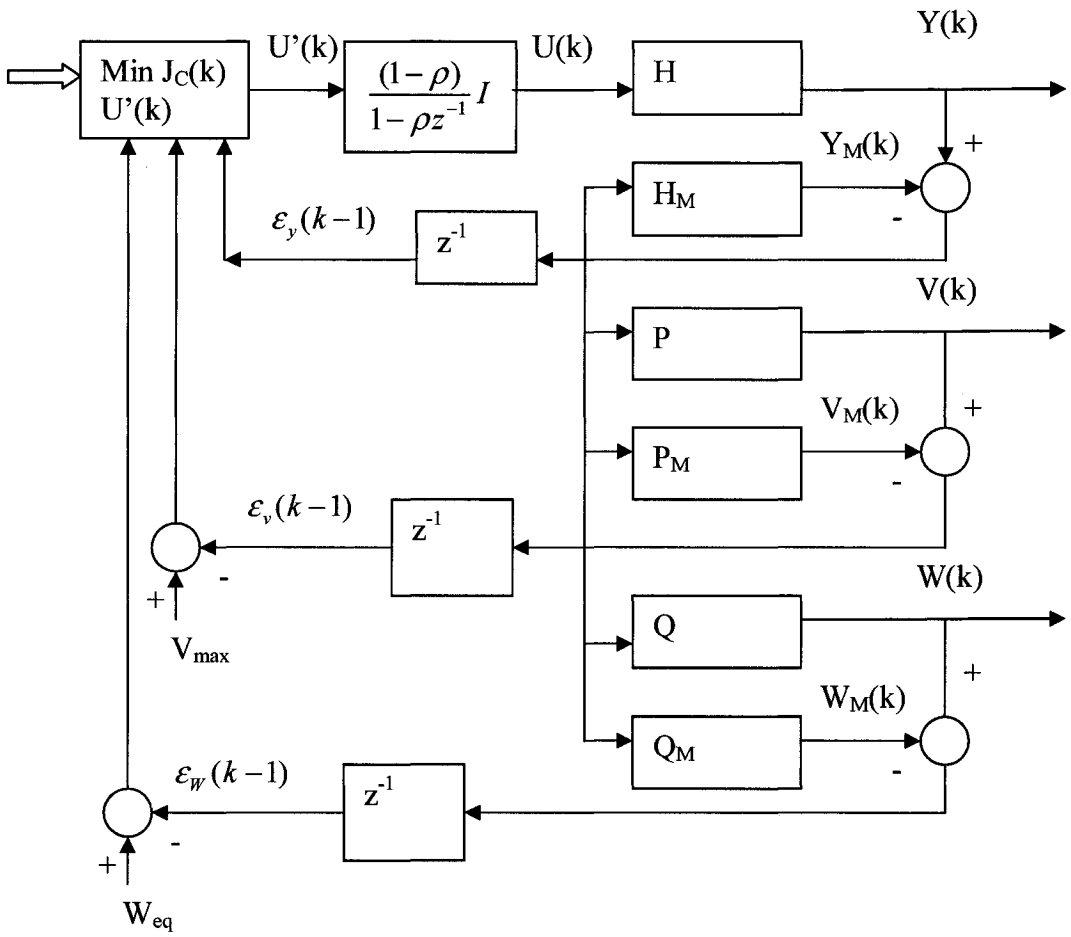

Figure 2- 2 : Structure de l'algorithme d'optimisation IMC
(Desbiens André, 2003)

Cette méthode possède néanmoins quelques inconvénients. Pour implanter cette méthode, un modèle du procédé doit être disponible. De plus, la solution obtenue est celle d'un modèle avec un réajustement. La valeur optimale du modèle obtenue par l'algorithme n'est donc pas exactement la même que celle du procédé.

2.3.2. Méthode des perturbations (Krstic et Wang, 2000).

Cette méthode «d'extremum seeking» trouve la valeur du gradient à l'aide de filtres et d'une excitation sinusoïdale (Voir la Figure 2-3). Plus précisément, une fonction sinus est ajoutée à l'entrée du système pour servir d'excitation. Cette perturbation fait alors osciller la variable de sortie. Le signal de sortie est ensuite envoyé dans un filtre «passe haut» qui laisse uniquement passer les oscillations causées par l'excitation du système. En d'autres mots, les données retenues sont uniquement les Δy et non la sommation de y et Δy. La fonction restante est donc une fonction sinusoïdale d'amplitude $\Delta y * \Delta u$ avec une moyenne de zéro car elle est centrée en zéro. Cette fonction est alors multipliée par la même fonction qui a servi à exciter le système. Ceci donne une fonction quadratique en sinus qui est toujours du même signe. Elle n'est donc plus centrée en zéro. Ce signal est ensuite dirigé dans un filtre « passe bas » pour obtenir la moyenne. La valeur à la sortie du filtre devient une approximation du gradient de la fonction qui est multipliée par l'amplitude de l'entrée au carré. La dernière étape consiste à utiliser cette valeur qui représente le gradient pour diriger le système au point optimal. Pour ce faire, cette valeur est multipliée par un gain et additionnée à l'entré du système, pour que celui-ci se déplace dans la direction ou le gradient est le plus petit, jusqu'à obtenir aucune variation de la sortie. Cette position correspond à la valeur minimale du système.

Cette méthode d'optimisation est facilement transposable au cas multivariable en utilisant les propriétés orthogonales des fonctions sinusoïdales. Pour ce faire, chacune des entrées est additionnée à une fonction de la forme a*sin(2*n*ω) où n représente le numéro de l'entrée. Après le filtre «passe haut », le signal de sortie est divisé en n branches. Chacune des branches est multipliée par la même fonction qui lui était associée à l'étape précédente. Comme le produit de chaque fonction est orthogonal à tous les autres, seulement la variation causée par cette entrée est non nulle. Le restant de chaque branche est similaire à celle monovariable.

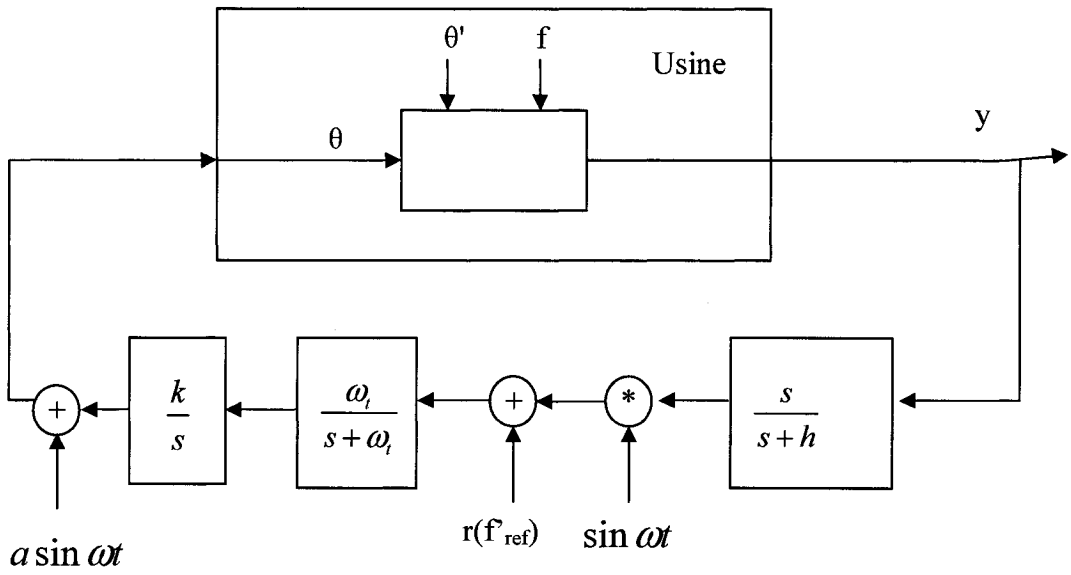

Figure 2- 3 : Structure de l'« extremum seeking » de la méthode des perturbations.

Cette méthode a plusieurs avantages. Cet algorithme est très robuste aux bruits, car ceux-ci sont éliminés à l'aide du filtre « passe bas » dans son cycle. Cette méthode ne perturbe pas le système de façon majeure lors de l'excitation, car elle est sous une

forme sinusoïdale avec une faible fréquence. L'algorithme d'optimisation multivariable est simple à définir. D'un autre côté, cette méthode possède deux inconvénients majeurs : la conception des filtres et la faible vitesse d'optimisation. La réalisation des filtres est complexe à cause du nombre important de ceux-ci dans l'algorithme et la précision dont ils doivent faire l'objet pour éliminer les fréquences non désirées. La vitesse d'optimisation, quant à elle, est calculée de cette manière : la fonction sinus qui sert de perturbation doit être au moins dix fois moins rapide que le procédé, l'estimation du gradient doit avoir lui aussi un facteur dix par rapport à la fonction sinusoïdale et finalement les calculs effectués avec ce gradient doivent aussi être dix fois moins rapide que l'estimation de ces gradients. Il en résulte un facteur 10^3 entre le système et la vitesse d'optimisation.

2.3.3. Fonction barrière (Vassiliadis et Floudas 1997)

Un désavantage majeur de la méthode de perturbation et de plusieurs autres méthodes est l'absence de considération pour les contraintes. Pour contourner ce problème, il suffit d'implanter à l'intérieur de ces algorithmes une méthode fréquemment utilisée dans la littérature, la fonction barrière. Elle consiste à transformer le problème d'optimisation sous contraintes en un ou plusieurs sous-problèmes sans contrainte. Pour ce faire, il faut ajouter à la fonction objectif un terme de correction, qui à l'intérieur des points réalisables, possède une valeur nulle et qui, à la contrainte, est infini. Le système ne peut donc pas franchir la contrainte. Une fonction comme celle-ci,

avec un saut de zéro à l'infini, est difficile à implanter. Des problèmes dans la résolution numérique surviennent également. Dans la littérature, plusieurs fonctions ont été proposées. La plus fréquemment utilisée est la barrière logarithmique qui se décrit comme suit :

Si la fonction à minimiser est :

$$V(x); \tag{2.18}$$

sous les contraintes :

$$ax \leq b. \tag{2.19}$$

Alors la nouvelle fonction à optimiser sans contrainte devient :

$$\gamma V(x) - \sum \log(b_i - a_i^T x). \tag{2.20}$$

Plus le système se rapproche d'une contrainte, plus le terme de la sommation devient important. Il empêche alors la fonction de continuer à diminuer. Le facteur γ permet de donner une prédominance soit à la fonction à optimiser ou à la contrainte, ce qui permet de s'éloigner plus ou moins des contraintes.

Le problème de l'application de cette méthode est l'ajout d'une fonction barrière pour toutes les contraintes du système. Cette procédure apporte souvent des inconvénients numériques au système. En plus, lorsque l'optimisation est discrète, la contrainte peut être dépassée entre deux intervalles. La fonction objectif peut donc être difficile à calculer. Le dernier désavantage est que le système ne se retrouve jamais sur la contrainte, il y a toujours un petit décalage avec la vraie valeur optimale.

CHAPITRE 3 Méthodes développées

Le chapitre trois décrit la méthode développée dans le cadre de ce projet de maîtrise, l' OTRPCA. Tout d'abord cette méthode est présentée pour un problème d'optimisation sans contrainte en mode continue pour prouver sa convergence dans ce cas simple. Ensuite l'algorithme discret est décrit. Pour continuer, la convergence de la méthode lors d'un problème contraint est présentée. Finalement, cette méthode est comparée aux algorithmes présentés au chapitre 2.

3.1. Optimisation sans contrainte

3.1.1. Formulation d'algorithme avec l'information du gradient.

Si l'information du gradient est disponible, un contrôleur intégral peut faire tendre le gradient vers zéro. Donc, l'optimisation en temps réel peut s'écrire :

$$\dot{u} = -\alpha \phi_u^T, \quad u(0) = u_0; \tag{3.1}$$

où u_0 est le point initial des variables d'entrées et α représente le gain.

Théorème 1. *(Convergence d'une optimisation en temps réel sans contrainte)* Dans un problème où il y a une seule valeur optimale avec un seul u pour ce point optimal, que $\phi_u(u^*)=0$, que $\phi_u(u)\neq 0, \forall u \neq u^*$ et que le contrôleur en temps réel est $\dot{u}=-\alpha\phi_u^T$, $u(0)=u_0$ pour toutes les valeurs $\alpha > 0$, alors le contrôleur converge asymptotiquement vers l'optimum.

Preuve: Prenons la fonction de Lyapunov suivante $V(u)=\phi(u)-\phi^*$, avec l'hypothèse que l'unicité de l'optimum, V est définie positif et est uniquement nulle à l'optimum. La dérivée en fonction du temps avec la dynamique $\dot{u}=-\alpha\phi_u^T$ est donnée par $\dot{V}=\phi_u \dot{u}=-\alpha\phi_u\phi_u^T$. Notez que $\dot{V}<0, \forall \quad \phi_u \neq 0$. Donc, la solution converge vers $\dot{V}=0$, ce qui est équivalent à dire que $\phi_u=0$. Comme il y a l'hypothèse d'un seul optimum, cet optimum unique est atteint.

3.1.2. Optimisation en temps réel avec estimation du gradient

Sachant que ϕ_u n'est pas connu, une estimation de ce paramètre est nécessaire. Dans ce projet, les différences finies sont utilisées, tel que $\phi_u \approx \frac{\Delta\phi}{\Delta u}$. L'idée est

d'utiliser le Δu généré par l'adaptation et non d'ajouter une excitation externe. En mode continu, on obtient donc : $\phi_u = \frac{\dot{y}}{\dot{u}}$. Ceci mène à la loi d'adaptation :

$$\dot{u} = -\alpha\phi_u = -\alpha\frac{\dot{\phi}}{\dot{u}}; \tag{3.2}$$

$$\dot{u}^2 = -\alpha\dot{\phi}; \tag{3.3}$$

$$\dot{u} = \sqrt{-\alpha\dot{\phi}}. \tag{3.4}$$

Deux inconvénients majeurs sont présents avec ce système. Premièrement, si $\dot{\phi}$ devient zéro, à cause du bruit, l'adaptation arrête peu importe si ϕ_u est égal ou non à zéro. En d'autres mots, la convergence est à $\dot{\phi} = 0$, $\dot{u} = 0$ mais pas nécessairement à $\phi_u = 0$. En second lieu, le signe de u n'est pas déterminé.

Pour régler le premier problème, un pas minimum, γ, est introduit, $\left|\dot{u}\right| = \max(\gamma, \sqrt{-\alpha\left|\dot{\phi}\right|})$. Cette formulation empêche la convergence à un point supérieur, mais cela empêche aussi u de converger. Par contre, si un petit γ est pris, la convergence parvient à une petite zone autour du point optimal, où elle peut être garantie. Pour régler le second problème, un état discret est défini : $q_s\{-1,1\}$. Cet état représente le signe de la variation. Cette valeur change à chaque fois que $\dot{\phi}$ devient

positif. Pour éviter un changement infini et aussi parce que l'estimation de la dérivée peut avoir un biais, une marge ε est introduite.

Avec les arguments pour le cas monovariable, la loi d'optimisation multivariable peut être formulée :

$$\dot{u} = -q_s(t)\max(\gamma, \sqrt{\alpha\left|\dot{\phi}\right|}), \quad u(0) = u_0; \tag{3.5}$$

$$q_s(t_+) = -q_s(t_-) \quad si \quad \dot{\phi}(t_-) \geq \varepsilon \quad q_s(0) = 1. \tag{3.6}$$

Les commutations de $e(t)$ se font dans les situations suivantes :

$$e(t) = \begin{cases} e_1 & si \quad t \in (t_0, t_0 + T) \\ e_2 & si \quad t \in (t_0 + T, t_0 + 2T) \\ \vdots & \vdots \qquad\qquad \vdots \\ e_m & si \quad t \in (t_0 + (m-1)T, t_0 + mT) \end{cases}. \tag{3.7}$$

La loi de commutation consiste à évaluer une variable à la fois et de changer de variable après un temps prédéterminé T. L'explication pour laquelle uniquement une variable est modifiée à la fois est que seulement une variable de sortie est disponible. Le gradient est obtenu par la dérivée de la variable de sortie sur la variable d'entrée. Un seul gradient peut donc être déterminé à la fois.

Théorème 2. *(Convergence de l'algorithme d'optimisation en temps réel par adaptation commutative.)* Posons le problème d'optimisation 3-6 et 3-7 avec un seul optimum ϕ^* à $u = u^*$ et $\phi_u(u^*) = 0$. Le contrôleur de l'équation 3-5 est utilisé. Pour $\gamma, \varepsilon, T > 0$, $\alpha \leq \frac{\gamma^2}{\varepsilon}$, le contrôleur mentionné converge vers une région autour de l'optimum où $\left|\phi_{u_i}\right| \leq \bar{\varepsilon}$, $\forall i$, où $\bar{\varepsilon} = \frac{\varepsilon}{\gamma} + (m-1) * \max_{i,j,i \neq j} \left\| \phi_{u_i u_j} \right\| \gamma T$.

Preuve : Considérons la fonction de Lyapunov $V = \phi - \phi^*$. Donc $\dot{V} = \dot{\phi}$, $\dot{\phi} < \varepsilon$ par la définition de la condition qui doit être satisfaite dans un état ou un autre. Si $\dot{\phi} < 0$ par le fait que V a une borne inférieure et qu'il y a un seul minimum, l'algorithme diminue la fonction objectif jusqu'à ce que le système atteigne $\dot{\phi} \geq 0$. Donc, la valeur finale de $\dot{\phi}$ pour n'importe quelles états discrets est $0 \leq \dot{\phi} \leq \varepsilon$.

Seulement aux commutations entre les états ($q_s = 1$ *et* $q_s = -1$), le signe de $\dot{u}_i$ change. Comme $\dot{\phi}(t_-) = \phi_{u_i} \dot{u}_i(t_-) = \varepsilon > 0$ avant l'instant de commutation, la valeur après la commutation est $\dot{\phi}(t_+) = \phi_{u_i} \dot{u}_i(t_+) = -\varepsilon < 0$. Donc, on peut conclure qu'avec cette loi de commutation, le système converge vers un intervalle où $-\varepsilon \leq \dot{\phi} \leq \varepsilon$.

Comme $\phi_{u_i} = e(t)\phi_u = \frac{\dot{\phi}}{\dot{u}_i} = \frac{\dot{\phi}}{e(t)\dot{u}}$ et $\left|e(t)\dot{u}\right| \geq \gamma$, la condition $\left|\dot{\phi}\right| \leq 0$ implique $\left|e(t)\phi_u\right| = \left|\phi_{u_i}\right| \leq \frac{\varepsilon}{\gamma}$, quand u_i est adapté $\left\|\phi_{u_i}\right\| \leq \frac{\varepsilon}{\gamma}$.

Quand $\alpha \leq \frac{\gamma^2}{\varepsilon}$, une fois que le système se situe dans une région près de l'optimum, $\dot{\phi} \leq \varepsilon$, $\max(\gamma, \sqrt{\alpha\left|\dot{\phi}\right|}) = \gamma$. Donc, quand u_j, $j \neq i$ est adapté, le changement maximal qui peut arriver à u_j durant une période d'adaptation est γT, $\left|\Delta u_j\right| \leq \gamma T$. Donc la variation maximale en ϕ_{u_i}, quand u_j est adaptée, est donnée par $\Delta\phi_{u_i} \leq \left\|\phi_{u_i u_j}\right\|_{\max} \Delta u_j \leq \left\|\phi_{u_i u_j}\right\|_{\max} \gamma T$, qui à son tour implique $\left\|\phi_{u_i}\right\| \leq \frac{\varepsilon}{\gamma} + \left\|\phi_{u_i u_j}\right\|_{\max} \gamma T$.

Pour tous les $i \neq j, \left\|\phi_{u_i}\right\| \leq \frac{\varepsilon}{\gamma} + (m-1)\max_j \left\|\phi_{u_i u_j}\right\| \gamma T$.

3.1.3. Algorithme d'implantation en mode discret

Les équations 3-5 et 3-7 sont implantées en mode discret selon l'algorithme représenté à la Figure 3-1 :

```
% Étape principale

Initialisation
Tant que (1)
        Choisir la direction ou le sens
        Tant que (fonction objectif descend)
                Adaptation par gradient
                Calculer la fonction objectif
        fin
fin

% Description des étapes principales

Initialisation
k = 0 – nombre d'itération, i = 1 – la variable choisi,  d = 1 – direction
positive
Choisir u(0)
Simulation pour calculer la fonction objectif

Changer la direction ou le sens
      si d = 1,
                d = -1,
      autre
                d = 1, i = i +1
                if i > size(u), i = 1, end

      fin

Adaptation par gradient
        u (k+1) = u (k) + d*e_i*max(Δ, abs(k*(y_k-y_k-1)/ (u_k,i-u_k-1,i)))

Simulation pour calculer la fonction objectif
        y(k+1) = F(u(k+1))
```

Figure 3- 1 : Algorithme sans contrainte.

Cette méthode d'optimisation consiste à effectuer un premier pas de longueur delta, et de trouver si la fonction objectif diminue dans cette direction. Dans le cas où elle diminuerait, il faut prendre la plus grande valeur entre delta et le gradient de la fonction pour faire le pas suivant. Si la fonction objectif ne descend pas, alors il faut choisir la variable suivante et recommencer l'algorithme. Si toutes les variables ont été utilisées, il faut changer de direction et recommencer l'optimisation.

Ce modèle peut paraître un peu simpliste, mais il possède plusieurs avantages par rapport aux méthodes plus complexes d'ES. Tout d'abord, la méthode d'incrément, qui utilise le maximum entre un pas constant et la valeur obtenue avec le gradient permet un déplacement constant même si le gradient du système devient très petit. Par contre, le désavantage est de ne jamais trouver exactement le maximum. Toutefois à cet optimum, il faut exciter le système pour voir s'il y a un changement dans les conditions d'optimum. Un avantage de cette méthode d'adaptation est qu'elle permet de ne pas réagir aux bruits de la variable manipulée lorsque l'intervalle du pas est petit, contrairement à la méthode composée uniquement du gradient. En plus, elle permet un déplacement plus rapide dans le cas où le gradient devient très petit sur une longue plage.

Cette méthode d'extremum seeking est auto-existante, ce qui veut dire qu'aucune perturbation est apportée au système dans le seul but de le perturber. Les variations nécessaires pour l'identification sont causées par la variation précédente.

3.2. Optimisation avec contraintes

3.2.1. Analyse de l'algorithme avec contraintes

En partant de la définition du problème d'optimisation avec contraintes, posons que $\bar{S}$ représente les contraintes d'inégalité actives et $\bar{\mu}$ leurs multiplicateurs de Lagrange respectif. L'équation $L_u = 0$ peut être réécrit sous la forme :

$$L_u = g + \bar{\mu}^T M = 0. \quad g = \phi_u - \phi_x F_x^{-1} F_u, \quad M(\bar{S}) = \bar{S}_u - \bar{S}_x F_x^{-1} F_u. \tag{3.8}$$

Les multiplicateurs de Lagrange reliés aux contraintes actives peuvent alors s'écrire :

$$\bar{\mu}^T = -gM^+; \tag{3.9}$$

où M^+ est la pseudo-inverse de M. La définition de la pseudo inverse est $MM^+ = I$ et n'implique pas que $M^+M = I$. En substituant dans l'équation $L_u = 0$, elle devient :

$$L_u = gP = 0, \quad P(\bar{S}) = I - M^+(\bar{S})M(\bar{S}). \tag{3.10}$$

P représente la matrice de projection sur les contraintes actives. Les équations 2-7 et 2-10 peuvent être réécrites sous une forme où les termes reliés à la projection sur la contrainte et la minimisation sont séparés :

$$\text{Contraintes} \quad F = 0, \quad \bar{S} = 0; \tag{3.11}$$

$$\text{Gradients} \quad gP(\bar{S}) = 0. \tag{3.12}$$

Optimisation avec contraintes avec le gradient

La loi de commande pour une optimisation avec contraintes peut s'écrire comme suit :

$$\dot{u} = -\alpha P(\bar{S}) g^T, \quad u(0) = u_0. \tag{3.13}$$

L'optimisation en temps réel se compose en trois parties : (1) l'identification des contraintes actives et le maintien de ces contraintes actives à zéro, (2) pousser les autres variables vers un gradient égal à zéro et (3) l'activation et la désactivation des contraintes actives. Le troisième aspect relié à l'activation et la désactivation de la contrainte est crucial et donne une nature hybride à la loi de contrôle. L'activation se produit quand une contrainte S_j touche zéro pour devenir positive. Quand la contrainte est active, le multiplicateur de Lagrange est modifié à l'aide de l'équation 3-9 et la désactivation se produit quand μ_j touche zéro pour devenir négatif. Ces faits peuvent être formulés de la façon suivante. L'état hybride du système est déterminé par les contraintes actives. Il consiste en un vecteur h où les éléments peuvent prendre des valeurs de 1 ou 0. Le nombre de 1 dans h est déterminé par le nombre de contraintes actives. Donc, la règle de transition pour h est :

$$h_j(t_+) = \begin{cases} 1 \;\; si \;\; S_j(t) = 0 \;\; et \;\; S_j(t_-) < 0 \\ 0 \;\; si \;\; \bar{\mu}_j(t) = 0 \;\; et \;\; \bar{\mu}_j(t_-) > 0 \end{cases}. \tag{3.14}$$

où t_- et t_+ sont les temps juste après et juste avant la transition, maintenant la formulation $P(\bar{S})$ est une fonction discrète de l'état h.

Avec cette forme hybride, la stabilité de l'optimisation en temps réel avec contraintes peut être démontrée à l'aide du théorèmes suivants.

Théorème 3. *(Stabilité de l'optimisation en temps réel avec contrainte)* Posons que l'optimisation avec contraintes sous la forme présentée dans ce chapitre possède uniquement un ϕ^* à $u = u^*$ avec le contrôleur $\dot{u} = -\alpha P(\bar{S}) g^T$, $u(0) = u_0$ et que l'évolution h_j est donnée par l'équation 3.14. Si la condition initiale correspond à un point valable pour tous les $\alpha > 0$, un contrôleur de ce type converge asymptotiquement vers l'optimum.

Preuve : Prenons la fonction de Lyapunov suivante, $V(u) = (\phi - \phi^*) + \Gamma \sum_j \min(S_j(x(u), u), 0)$. En prenant Γ comme constante positive beaucoup plus grande que tous les multiplicateurs de Lagrange possibles, la fonction pénalité mène à l'optimum exact (Fletcher 1979). Avec l'hypothèse de l'unicité de l'optimum, V est défini positif et possède une valeur de 0 à l'optimum.

La solution reste toujours possible. Pour que la solution devienne infaisable, il faut que S_j soit positif. Pour ce faire, il doit traverser la valeur zéro mais si ceci se produit, h change et la nouvelle projection garde cette contrainte active à zéro. Pour montrer cela, $\dot{\bar{S}}$ se réécrit sous la forme suivante :

$$\dot{\bar{S}} = \bar{S}_u \dot{u} + \bar{S}_x \dot{x} = (\bar{S}_u - \bar{S}_x F_x^{-1} F_u)\dot{u} = M\dot{u} = -\alpha MPg = 0; \tag{3.15}$$

où $\dot{\bar{S}} = 0$ vient du fait que $MP = 0$ et que $\dot{F} = F_x \dot{x} + F_u \dot{u} = 0$.

La dérivé en fonction du temps de la fonction de Lyapunov devient :

$$\dot{V} = \phi_x \dot{x} + \phi_u \dot{u} + \Gamma \sum_{q_j=1} (S_{j_x} \dot{x} + S_{j_u} \dot{u}); \tag{3.16}$$

$$\dot{V} = (\phi_u - \phi_x F_x^{-1} F_u)\dot{u} + \Gamma \sum_{q_j=1} (S_{j_u} - S_{j_x} F_x^{-1} F_u)\dot{u}; \tag{3.17}$$

$$\dot{V} = -\alpha g P g^T - \alpha\Gamma \sum_{q_j=1} M_j P g^T \leq 0; \tag{3.18}$$

où M_j est la rangée de la matrice M qui correspond à la contrainte S_j. Comme $MP = 0$, $M_j P$ est aussi égale à zéro, le second terme est également zéro. Le premier terme est négatif, parce que P est semi-défini positif. Alors, $\dot{V}$ est plus petit que zéro. La solution converge vers $\dot{V} = 0$, parce que $gPg^T = gPP^T g^T = 0$ et cela s'explique par

$gP = 0$. Avec la supposition d'un seul optimum, cette hypothèse se produit uniquement à cet optimum. Toutefois, il faut faire attention à l'amplitude de la variable α.

3.2.2. Optimisation en temps réel avec estimation du gradient

La loi d'adaptation avec l'estimation de gradient et contrainte est la suivante :

$$\dot{u} = -Pq_s e(t)\max(\gamma\sqrt{\alpha\left|\dot{\phi}\right|}), \quad u(0) = u_0; \tag{3.19}$$

$$q_s(t_+) = -q_s(t_-), \quad si \quad \dot{\phi}(t_-) \geq \varepsilon, \quad q_s(0) = 1; \tag{3.20}$$

où la projection P est utilisée pour garder la contrainte active.

Théorème 4. *(Stabilité de l'optimisation en temps réel avec contrainte et estimation du gradient.)* Prenons le problème d'optimisation défini par l'équation 2.3 qui possède uniquement un optimum à $u = u^*$. Le contrôle défini à l'équation 3.19 est utilisé. Le point initial u_0 correspond à un point faisable pour γ, ε, $T > 0$, $\alpha \leq \frac{\gamma^2}{\varepsilon}$. Le contrôle va converger dans une région autour de l'optimum où $\left|\phi_{u_i}\right| \leq \bar{\varepsilon}$, $\forall i$, où $\bar{\varepsilon} = \frac{\varepsilon}{\gamma} + (m-1)\max_{i,j,i\neq j}\left\|\phi_{u_i u_j}\right\|\gamma T$.

Preuve : Considérons la fonction de Lyapunov suivante, $V(u) = (\phi - \phi^*) + \Gamma \sum_j \min(S_j(x(u),u),0)$, $\Gamma >> 0$, comme dans le Théorème 3.

Comme dans la preuve du Théorème 3, il peut être montré que $\bar{S} = 0$. La dérivé en fonction du temps de la fonction de Lyapunov avec la dynamique de l'équation 3.13 est donnée par :

$$\dot{V} = \dot{\phi} - \alpha\Gamma \sum_{q_j=1} M_j P q_s e \max(\gamma, \sqrt{\alpha \left| \dot{\phi} \right|}) \leq 0. \tag{3.21}$$

Comme dans la preuve du Théorème 3, le second terme est zéro. Comme $\dot{V} = \dot{\phi}$, la preuve du Théorème 2 peut être reprise pour conclure $\left| \phi_{u_i} \right| \leq \bar{\varepsilon}$.

3.2.2. Implantation de la projection utilisant deux niveaux de contrôleur.

La projection garde uniquement $\bar{S} = 0$ ce qui veut dire que toutes déviations de $\bar{S}$ dues à des perturbations ou à des couplages avec les autres variables manipulées doivent être corrigées. Ce qui revient à dire que la projection doit libérer autant de variables manipulées qu'il y a de contraintes actives, afin que ces contraintes puissent être gardées à zéro.

Les variables manipulées sont séparées en deux catégories : $\bar{u}$, les variables conservant les contraintes actives à zéro, (elles peuvent être choisies arbitrairement) et $\tilde{u}$, les variables libres. Les transformations s'écrivent : $\bar{u} = \bar{V} u$ et $\tilde{u} = \tilde{V} u$, où $V^T(S) = [\bar{V}^T \; \tilde{V}^T]$ est de rang complet.

Pour obtenir la loi d'adaptation, il faut regarder l'effet de la partition de $u = [\bar{u} \; \tilde{u}]$ sur $L_u = 0$:

$$L_{\bar{u}} = \bar{g} + \bar{\mu}^T \bar{M} = 0, \qquad \bar{g} = \phi_{\bar{u}} - \phi_x F_x^{-1} F_{\bar{u}}, \qquad \bar{M}(\bar{S}) = (\bar{S}_{\bar{u}} - \bar{S}_x F_x^{-1} F_{\bar{u}}); \tag{3.22}$$

$$L_{\tilde{u}} = \tilde{g} + \bar{\mu}^T \tilde{M} = 0, \qquad \tilde{g} = \phi_{\tilde{u}} - \phi_x F_x^{-1} F_{\tilde{u}}, \qquad \tilde{M}(\bar{S}) = (\bar{S}_{\tilde{u}} - \bar{S}_x F_x^{-1} F_{\tilde{u}}); \tag{3.23}$$

$$\bar{\mu}^T = -\bar{g} \bar{M}^{-1}, \qquad L_{\tilde{u}} = \tilde{g} - \bar{g} \bar{M}^{-1} \tilde{M} = 0. \tag{3.24}$$

Ce qui donne un contrôleur :

$$\dot{\bar{u}} = -\beta G^{-1} \bar{S}, \qquad \bar{u}(0) = \bar{u}_0; \tag{3.25}$$

$$\dot{\tilde{u}} = -\alpha(\tilde{g} - \bar{g} \bar{M}^{-1} \tilde{M}), \qquad \tilde{u}(0) = \tilde{u}_0. \tag{3.26}$$

où G^{-1} est l'inverse de la matrice gain entre $\bar{S}$, $\bar{u}$ et $\beta > 0$. β est le gain d'adaptation.

Théorème 5. *(Stabilité des contrôles sur les contraintes actives)* Posons $\bar{S}(\bar{u})$, une variable contrôlée, et $\bar{u}$, une variable manipulée, tel que $\bar{u}^*$ est la valeur qui porte $\bar{S}(\bar{u}^*)=0$. Prenons le contrôleur intégral, $\dot{\bar{u}}=-\beta G^{-1}\bar{S}$, $\bar{u}(0)=\bar{u}_0$. Si $\bar{S}^T G\Delta\bar{u}>\gamma\Delta\bar{u}^T\Delta\bar{u}$ avec $\Delta\bar{u}=\bar{u}-\bar{u}^*$, alors le contrôleur force $\bar{S}$ à zéro exponentiellement.

Preuve : Prenons la fonction de Lyapunov suivante $V=\frac{1}{2}\Delta\bar{u}^T G^T G\Delta\bar{u}$. V est bornée par $k_1\left\|\Delta\bar{u}\right\|^2\le V\le k_2\left\|\Delta\bar{u}\right\|^2$, où k_1 et k_2 sont les valeurs propres de $\frac{1}{2}G^T G$. La dérivée de V est $\dot{V}=-\beta\Delta\bar{u}^T G^T G\Delta\bar{u}=-\beta\Delta\bar{u}^T G^T\bar{S}<-\beta\gamma\Delta\bar{u}^T\Delta\bar{u}=-\beta\gamma\left\|\Delta\bar{u}\right\|^2\le 0$, à cause des bornes sur la dérivée du temps, le système est exponentiellement stable.

3.2.3. Algorithme avec contrainte en modes discret

```
% Initialisation

Tant que (1)
    Choisir la direction ou le sens et libérer la variable

    si (fonction objectif augmente) OU (Contraintes non actives non respectée)
        Remettre les anciennes contraintes actives
        Changer la direction ou le sens et libérer la variable
    fin
    Tant que (fonction objectif descend) ET (Contraintes non actives respectées)
        Adaptation par gradient avec projection pour les contraintes active
        Calculer la fonction objectif et les contraintes non actives
    fin
    si Contraintes non actives non respectées
        Ajouter la contrainte non respectées avec les contraintes actives
        Refaire le pairage si nécessaire
    fin
fin

% Description des étapes principale d'optimisation avec contraintes

Initialisation

k = 0 – nombre d'itération, i = 1 – la variable choisi, d = 1 – direction positive
Pas de contraintes actives - Sbar = {} – ensemble de contraintes actives, Stilde = S -
reste
Pas des variables liées aux contraintes actives - ubar = {} – ensemble de variables
liées aux contraintes actives, utilde = u – reste.
istore = {}; ParSstore = {} – Pas de mémoire de contraintes active
Choisir u(0)
Simulation pour calculer la fonction objectif et les contraintes non actives

Remettre la contrainte active
ubar = {ubar} + {istore}, utilde = {utilde} - {istore}
    Sbar = {Sbar} + {ParSstore}, Stilde = {Sbar} - { ParSstore}
```

Figure 3- 2 : Algorithme avec contrainte (1/2)

Enlever la contrainte active
ubar = {ubar} - {istore}, utilde = {utilde} + {istore}
Sbar = {Sbar} - {ParSstore}, Stilde = {Sbar} + { ParSstore}

Changer la direction ou le sens et libérer la variable
si d = 1,
d = -1,
autre
d = 1, i = i +1
if i > size(u), i = 1, end
if i in ubar (Libérer la variable si nécessaire)
istore = i, ParSstore = ParSi
Mémoriser pour remettre si la fonction objectif n'améliore pas,
ParSi est la contrainte qui est liée avec i.
Enlever la contrainte active
fin
fin
Ajouter la contrainte non respectée avec les contraintes actives
istore = i, ParSstore = j
Remettre la contrainte active

Adaptation par gradient avec projection pour garder les contraintes actives
utilde(k+1) = utilde(k) + $d*e_i*\max(\Delta, abs(k*(y_k-y_{k-1})/ (u_{k,i}-u_{k-1,i})))$
ubar(k+1) = ubar (k) + P* $d*e_i*\max(\Delta, abs(k*(y_k-y_{k-1})/ (u_{k,i}-u_{k-1,i})))$
(P est choisi de ne pas violer les contraintes Sbar)

Simulation pour calculer la fonction objectif et les contraintes non actives
y(k+1) = F(u(k+1)), Stilde(k+1) = Gtilde(u(k+1))

Refaire le pairage si nécessaire

Figure 3- 3 : Algorithme avec contrainte (2/2)

Dans cette section, l'algorithme est décrit à l'aide de plusieurs cas où il peut être employé. Cette méthode a pour but de faciliter la compréhension du lecteur. Celui-ci est également prié de consulter le chapitre 4, sur l'application et les résultats, pour avoir plus de détails sur les problèmes de l'algorithme.

Il faut tenir compte des contraintes lors d'une optimisation puisque l'algorithme peut rester pris à une contrainte. En effet, la contrainte peut bloquer toutes les directions et ainsi diriger l'algorithme vers un faux optimum. Voici une Figure 3-4 qui illustre ce qui peut se produire lors d'une optimisation possédant une contrainte avec un algorithme sans contrainte.

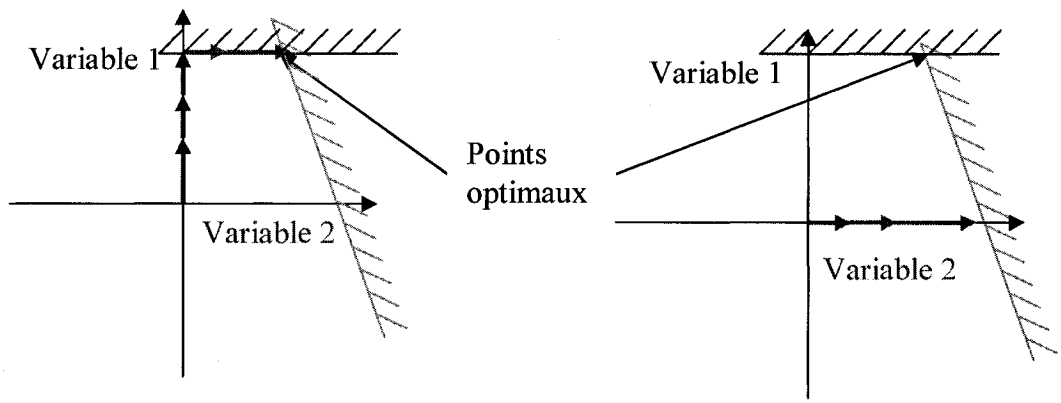

Figure 3- 4 : Problème causé par l'ordre des variables (2 variables)

Dans la Figure 3-4 qui représente l'optimisation bloquée, le système ne peut continuer l'optimisation car toutes les directions de la variable 1 ou 2 sont infaisables

ou bien elles conduisent à une augmentation de la fonction objectif. La solution envisagée dans ce mémoire pour régler ce problème est de faire un glissement sur la contrainte, comme le représente la Figure 3-5.

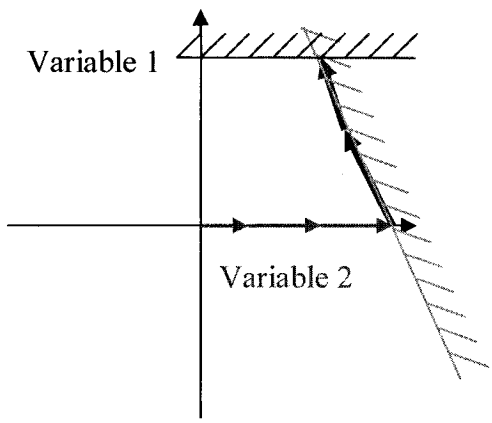

Figure 3- 5 : Solution au problème causé par l'ordre des variables (2 variables)

Un autre problème possible est que le système comporte plusieurs contraintes. L'utilisation de l'algorithme avec seulement un glissement peut capturer le système sur une mauvaise contrainte. La Figure 3-6 représente ce cas.

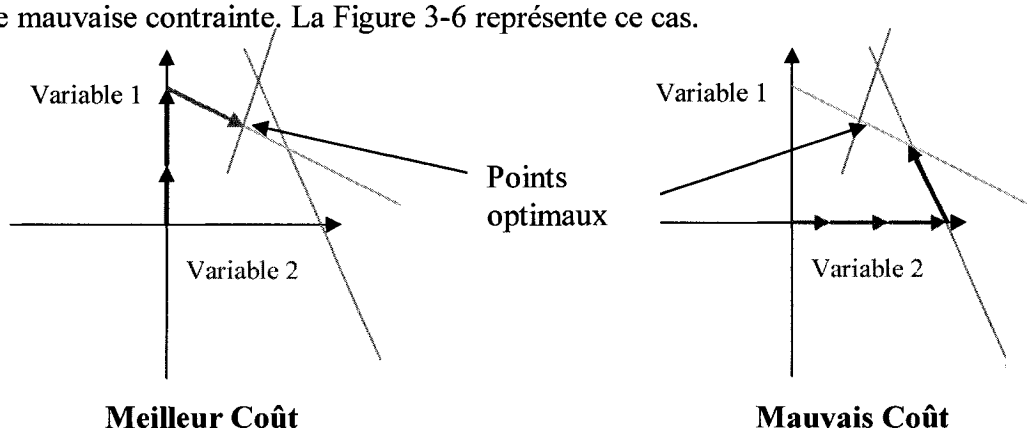

Figure 3- 6 : Problème causé par l'ordre des variables (3 variables)

Sur la Figure 3-6 droite , la première contrainte rencontrée n'est pas celle qui est active au point optimal, ceci empêche le système d'arriver au point optimal. La solution pour régler ce problème est que l'algorithme abandonne la contrainte lorsque ce point est atteint. Une alternative est de vérifier lorsque le système est bloqué si une nouvelle contrainte ou bien une direction sans contrainte peut améliorer l'objectif. La Figure 3-7 représente ce cas.

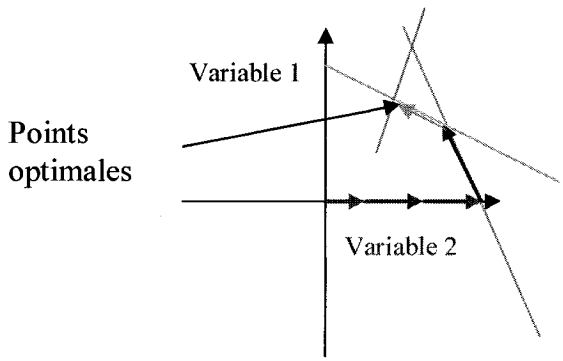

Figure 3- 7 : Solution au problème causé par l'ordre des variables (2 variables)

Bref l'algorithme proposé dans le cadre de ce mémoire implique que lorsqu'une contrainte est rencontrée, elle est considérée active pour la suite de l'optimisation. Pour ce faire, la variable manipulée de l'optimisation doit être couplée à la contrainte pour obtenir un degré de liberté permettant au système de suivre cette contrainte. Une fois qu'un optimum est détecté, la contrainte doit être abandonnée pour vérifier si une autre contrainte permet d'obtenir une valeur plus optimale de la fonction objectif ou même de continuer dans une direction sans contrainte.

Le principal avantage de cette méthode est que si l'optimum est à la contrainte, il est possible d'être exactement à ce point. Un second avantage provient de la rapidité à évaluer le gradient et à l'appliquer. Deux désavantages à cette méthode existent. Le premier est qu'il y a uniquement une variable qui est manipulée à la fois, ce qui peut nuire à la vitesse d'exécution s'il a plusieurs variables d'entrées pour l'optimisation. Le second est le comportement sur le bruit des variables observées. Pour enrayer ce problème, un filtre peut être placé sur chaque sortie, quoique ceci peut se révéler complexe.

3.3. Comparaison entre les méthodes

Après avoir présenté la méthode OTRPCA dans les sections précédentes et effectué la description des méthodes d'optimisation au chapitre 2, nous sommes rendu à l'étape de comparer ces méthodes. Dans l'annexe E, un exemple est présenté pour la comparaison numérique des différentes méthodes.

3.3.1. Comparaison entre IMC et la méthode OTRPCA

La première différence entre la méthode OTRPCA et la méthode IMC est la présence d'un modèle dans l'optimisation IMC. Pour des problèmes concernant des procédés aussi complexes qu'une usine, il est difficile et coûteux d'obtenir de bons

modèles. De plus, ceux-ci peuvent devenir rapidement désuets ou même inutilisables en raison d'une modification dans le procédé. Les méthodes d'optimisation sans modèle peuvent, quant à elles, s'adapter au changement.

Une autre différence est que l'optimisation avec la méthode IMC ne permet pas d'être à la valeur optimale du procédé. En effet, c'est la valeur du modèle qui est considérée et qui est ensuite ajustée. De plus, cette comparaison s'effectue aussi sur le traitement de la contrainte, tandis que dans la méthode développée, c'est la contrainte qui est considérée active. Ainsi, le point optimal est exactement celui de l'optimum.

Un avantage de la méthode IMC est la quantité plus faible d'informations nécessaires pour optimiser le procédé. Car en se basant sur le modèle, on obtient une approximation de l'optimum et qui doit ensuite être réajusté, tandis que des informations sur le procédé doivent être continuellement fournies à la méthode OTRPCA pour calculer le gradient.

3.3.2. Comparaison entre la méthode des perturbations et la méthode de projection sur contraintes actives

Avec la méthode des perturbations, plusieurs filtres sont à définir et des gains, à ajuster, tandis qu'avec la méthode OTRPCA, des gains et un delta doivent être

spécifiés. Le delta est facile à estimer alors que les filtres peuvent être plus compliqués à dimensionner si le nombre de variables manipulées est importants.

La méthode d'optimisation avec perturbation est très lente comme il a été expliqué plus tôt, tandis que la méthode OTRPCA prend des mesures de gradient rapidement.

Un avantage de cette méthode est qu'elle est très résistante au bruit, tandis que la méthode OTRPCA est probablement résistante dans la section avec le delta mais pas dans la section qui considère la fonction du gradient. La plus grande influence du bruit survient lorsque le gradient se rapproche de la dimension du bruit. Pour pallier ce problème, un delta plus grand doit être utilisé.

Un avantage de la méthode des perturbations comparativement à la méthode de glissement est qu'elle permet de faire varier toutes les variables en même temps plutôt que de les faire varier l'une après l'autre.

3.3.3. Fonction barrière et la méthode OTRPCA

Pour la fonction barrière, des problèmes numériques peuvent survenir lors de son application. Ils sont causés par le changement rapide de la fonction objectif. De plus, si le système discret est considéré, un pas d'itération trop grand peut amener le

système de l'autre côté de la contrainte. Dans certains cas, la solution est inexistante, ce qui entraîne des complications majeures au niveau de l'algorithme. Pour la méthode du glissement de contrainte, la contrainte d'inégalité devient une contrainte d'égalité, ce qui n'est pas contraignant.

Une autre différence est que la méthode des barrières ne permet pas d'être exactement sur la contrainte si le point optimal est situé à cet emplacement. Un facteur de convergence doit être ajouté pour déplacer le point près de la contrainte. La méthode de glissement de contrainte, quant à elle, peut se situer sur toutes les contraintes. Elle peut donc atteindre tous les points dans son domaine réalisable.

En somme, les diverses comparaisons effectuées précédemment, nous poussent à choisir la méthode de glissement de contrainte pour effectuer le traitement du procédé utilisé dans le cadre de cette maîtrise. Toutefois, avant de présenter les résultats et l'analyse, il ne reste qu'à décrire le procédé utilisé dans le cadre de ce projet. Ceci est réalisé au chapitre suivant.

CHAPITRE 4 Application et résultats

4.1. Description du procédé

Comme mentionné précédemment, le système étudié au cours de ce projet est le procédé de production monomère d'acétate de vinyle (MAV) (Chen, 2003). Le procédé est représenté par une simulation d'un système contenant 26 variables manipulées, 43 variables mesurées et 246 variables d'états. Le pairage effectué par McAvoy est utilisé dans ce projet, mais quelques modifications sur des réglages de contrôleur ont été réalisées. Le lecteur est invité à consulter l'annexe A pour connaître les valeurs précises des paramètres des contrôleurs.

Le procédé est représenté par un réacteur et un important système de purification. Il contient quelques boucles de recirculation et un système d'échangeur de chaleur intégré. Quatre produits sont introduits dans le système : de l'éthylène (C_2H_4), de l'oxygène (O_2), de l'acide acétique (HAc) et finalement l'inerte qui est de l'éthane (C_2H_6). Trois substances sont générées par le système soit le produit désiré, du monomère d'acétate de vinyle (MAV) et des produits secondaires, de l'eau (H_2O) et du gaz carbonique (CO_2). Le schéma du procédé est donné à la Figure 4-1.

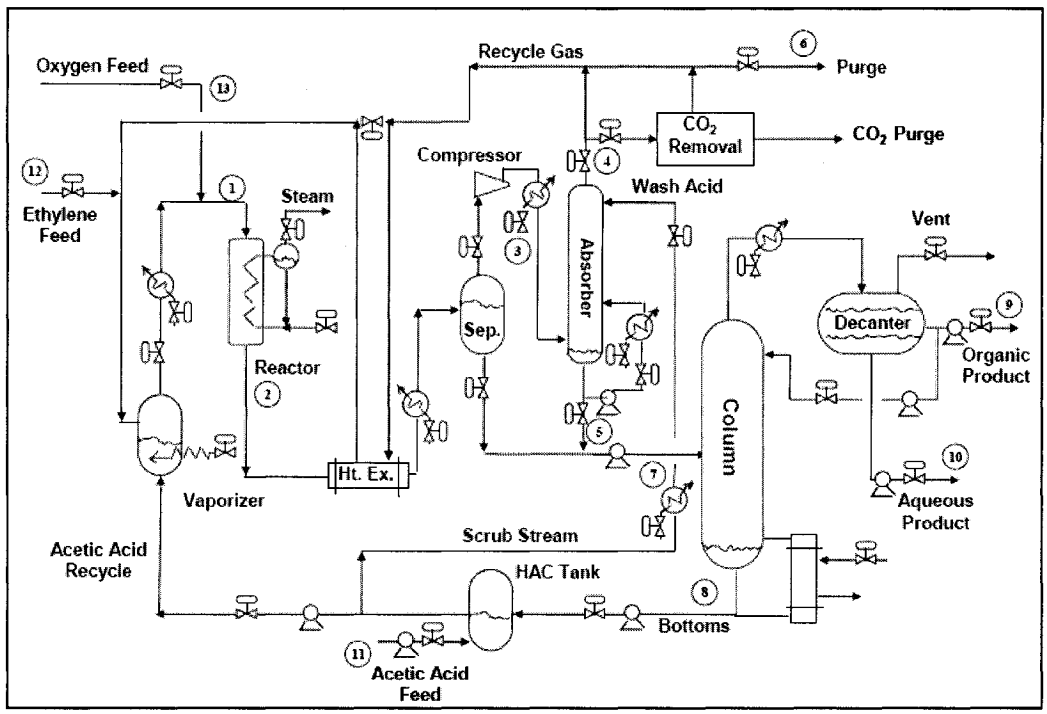

Figure 4- 1 : Schéma de l'usine de production du MAV (Chen, Rong, 2003)

Le réacteur

Le réacteur est de type piston et il contient des tubes remplis d'un catalyseur métallique sur un support de silice. L'énergie est enlevée du réacteur lors de l'évaporation de l'eau dans la chemise de celui-ci. La vapeur d'eau quitte le réacteur sous forme saturée. Le réacteur est modélisé en un système distribué de 10 sections dans la direction axiale. Deux réactions irréversibles et exothermiques se produisent à l'intérieur des tubes du réacteur. Tout d'abord, la réaction qui produit le MAV :

$$C_2H_4 + CH_3COOH + \frac{1}{2}O_2 \longrightarrow CH_2 = CHOCOCH_3 + H_2O; \qquad (4.1)$$

à une vitesse de réaction de :

$$r1 = 0.1036 \times \exp(-\frac{3674}{T}) \times \frac{p_O p_E p_A (1+1.7 p_w)}{(1+0.583 p_O (1+1.7 p_W))(1+6.8 p_A)} \quad (4.2)$$

et la réaction secondaire :

$$C_2H_4 + 3O_2 \longrightarrow 2CO_2 + 2H_2O. \quad (4.3)$$

à une vitesse de réaction de :

$$r2 = 1.9365 \times 10^5 \times \exp(-\frac{10,116}{T}) \times \frac{p_O (1+0.68 p_W)}{1+0.76 p_O (1+0.68 p_W)} \quad (4.4)$$

où p représente la pression partielle, O l'oxygène, E l'éthylène, A l'acide ascétique et W l'eau.

Le système de purification est composé de trois séparateurs, d'une colonne d'absorption, d'une colonne à distiller azéotropique et d'un décanteur. Un réservoir pour récupérer l'acide acétique utilisé dans le procédé et un vaporisateur pour évaporer l'acide acétique sont également présents. Finalement, un compresseur pour séparer les éléments qui peuvent être condensés s'y retrouve aussi. Une description plus détaillée de tous ces composants est disponible dans l'annexe A.

4.2. Choix de la fonction objectif et des variables manipulées

4.2.1. Choix de la fonction objectif

La fonction objectif pour l'optimisation est le débit du MAV produit. Cette valeur est calculée en multipliant le débit du courant de sortie et la concentration de MAV dans ce même courant. Au niveau de la simulation, cela est représenté par le produit de la variable observée 17, le débit du produit organique, et la variable 24, la concentration de MAV dans le courant du produit organique. La valeur au point d'opération de la fonction objectif est de 0.787 kmol/min.

4.2.2. Choix des variables manipulées

Naturellement, l'optimisation peut être effectuée à l'aide des 26 variables du procédé, ce qui permet d'obtenir la valeur maximale possible pour le procédé. Le problème, c'est qu'une optimisation à l'aide d'autant de variables se révèle fastidieuse, longue et souvent sans véritable augmentation de la fonction objectif. On peut penser que le principe de Pareto peut s'applique. Donc, pour ce problème, le fait d'étudier moins de 5 variables peut suffire à atteindre un rendement adéquat avec un effort raisonnable. La plus grande difficulté consiste à faire un choix éclairé pour identifier ces variables. Pour ce faire deux méthodes sont utilisées : 1) un plan factoriel et 2) une méthode du gradient. Finalement, une bonne compréhension du procédé permet également d'aiguiller vers un choix judicieux des variables à utiliser pour l'optimisation.

4.2.2.1. Type de variables

A première vue dans un problème comme celui-ci, il peut sembler évident que les variables utilisées pour faire varier le système sont les variables manipulées. Ces variables sont prises en considération par le système à chaque pas de temps pour faire varier les états du système. Le problème est qu'une fois ces valeurs fixées, le système n'est plus contrôlé. Ainsi si ces variables n'ont pas de valeurs stables, le système diverge et aucun moyen n'est disponible pour le corriger. Alors, les variables pour l'optimisation ne doivent pas être les variables manipulées mais elles doivent donner une valeur à respecter tout en laissant les variables manipulées libres de corriger les perturbations qui peuvent arriver. Les variables qui correspondent à cette description sont les points de consigne du contrôleur. Elles sont donc utilisées dans ce projet.

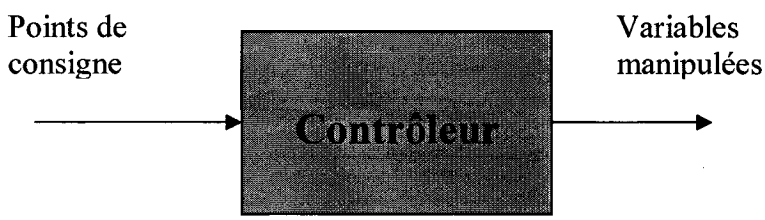

Figure 4- 2 : Variables manipulées

4.2.2.2. Plan factoriel

La méthode des plans factoriels est couramment utilisée pour sélectionner les variables les plus influentes sur une fonction sortie. Cette méthode consiste à faire une

suite d'expériences avec des valeurs prédéterminées pour chaque variable, afin que l'analyse des données soit simplifiée et que l'effet de chaque variable soit déterminé. Il existe plusieurs types de plan factoriel qui peuvent servir soit à tamiser les facteurs, soit à les modéliser. L'objectif des plans factoriels dans ce projet est de tamiser les variables afin de limiter l'optimisation aux variables les plus critiques de la fonction objectif. Les modèles de type 2^{k-p} sont les plus utilisés pour accomplir cet objectif, où k représente le nombre de variables utilisées dans le plan et p, le degré de fractionnement.

Rappel théorique sur les plans factoriels 2^k

Voici la solution obtenue d'un plan factoriel avec 3 variables :

$$Y=B_0+B_1X_1+B_2X_2+B_3X_3+B_{12}X_1X_2+B_{13}X_1X_3+B_{23}X_3X_3+B_{123}X_1X_2X_3; \quad (4.5)$$

B_0 : l'effet moyen

B_1 B_2 B_3 : les effets principaux

B_{12} B_{13} B_{23} :les effets secondaires ou d'interactions

B_{123} : l'effet d'interaction triple

X : Les valeurs des variables normalisées entre 1 et –1

Y : Fonction objectif

Les effets principaux sont généralement plus significatifs que les effets secondaires et les effets secondaires sont souvent plus significatifs que l'interaction

triple. En général, les effets principaux et secondaires sont uniquement considérés lors d'analyse statistique.

La solution des plans factoriels donne, si le facteur p est à zéro, la famille des plans factoriels complets. L'avantage de cette famille consiste en la possibilité de bien définir tous les effets du plan, tandis que le désavantage est que le nombre d'expériences qui croit exponentiellement avec le nombre de variables. Il est possible de diminuer le nombre d'expériences nécessaires pour identifier les effets du système en confondant certaines des interactions d'ordre élevé avec des effets principaux. Comme les effets d'ordre élevé sont généralement peu significatifs en comparaison des effets principaux, cette hypothèse est acceptable. Cette famille de plan, que sont les plans factoriels partiels, est représentée par la formule 2^{k-p}. Une description plus approfondie sur les plans factoriels est disponible dans l'ouvrage *Design and Analysis of Experiments*, (MONTGOMERY, 2005)

Pour l'analyse du choix des variables pour le MAV, il est avantageux de diminuer le nombre de variables à analyser, car le nombre d'expériences avec 26 variables manipulées est très élevé même avec des plans factoriels partiels. Tout d'abord, les variables de consigne pour les niveaux de remplissage (réservoir, réacteur, colonne) ont été retirées de l'analyse car ceux-ci n'influencent pas énormément la production de MAV. Ensuite, les variables qui affectent le système de purification ont aussi été enlevées, car le système a peu de perte en MAV. Les effets de ces dernières

variables sur le réacteur par les boucles de recirculation sont également jugées négligeables. Voici une liste des variables utilisées pour l'analyse statistique:

Tableau 4.1 : Variables utilisées pour le plan factoriel

# variable	nom	Intervalle McAvoy		Intervalles utilisés
1	Alimentation d'oxygène	0 – 2.268	kmol/min	-0.6 – 0.3
2	Alimentation d'éthylène	0 – 7.56	kmol/min	-0.1 – 0.5
5	Pression dans le vaporisateur	0 – 200	psia	0.175 – 0.7
7	Température du réacteur	0 – 200	C	0.45 – 0.595
17	% de gaz carbonique dans la purge	0 – 50	%	-0.976 – -0.95
18	% d'éthane dans la purge	0 – 100	%	-0.51 – 0.2

Les intervalles utilisés présentés au Tableau 4.1 sont les valeurs normalisées entre 1 et –1 d'après les valeurs maximales données par Chen et McAvoy (2003). De moins grandes plages que celles de McAvoy sont utilisées car il y a une interaction entre les variables qui empêchent le respect de certaines contraintes.

Une liste complète des expérimentations est présentée à l'annexe B.

Tableau 4.2 : Liste des effets

Factor	Effect	Std.Err.	t(2)	p	-95,% Cnf.Limt	+95,% Cnf.Limt	Coeff.	Std.Err. Coeff.	-95,% Cnf.Limt	+95,% Cnf.Limt
	Effect Estimates; Var.:Y; R-sqr=,99986; Adj:,99895 (PLANFONTIONNETU.STA) 2**(6-2) design; MS Residual=,0000562 DV: Y									
Mean/Interc.	0,710807	0,001873	379,4258	0,000007	0,702747	0,718867	0,710807	0,001873	0,702747	0,718867
(1)A	-0,100232	0,003747	-26,7518	0,001394	-0,116353	-0,084111	-0,050116	0,001873	-0,058177	-0,042056
(2)B	0,007295	0,003747	1,9470	0,190913	-0,008826	0,023416	0,003647	0,001873	-0,004413	0,011708
(3)C	0,352577	0,003747	94,1022	0,000113	0,336456	0,368698	0,176289	0,001873	0,168228	0,184349
(4)D	0,145580	0,003747	38,8551	0,000662	0,129459	0,161701	0,072790	0,001873	0,064730	0,080851
(5)E	-0,007305	0,003747	-1,9498	0,190511	-0,023426	0,008816	-0,003653	0,001873	-0,011713	0,004408
(6)F	-0,199818	0,003747	-53,3310	0,000351	-0,215939	-0,183697	-0,099909	0,001873	-0,107969	-0,091848
1 by 2	-0,000760	0,003747	-0,2028	0,858072	-0,016881	0,015361	-0,000380	0,001873	-0,008440	0,007681
1 by 3	-0,039857	0,003747	-10,6378	0,008721	-0,055978	-0,023736	-0,019929	0,001873	-0,027989	-0,011868
1 by 4	-0,011411	0,003747	-3,0456	0,093014	-0,027532	0,004710	-0,005705	0,001873	-0,013766	0,002355
1 by 5	-0,013797	0,003747	-3,6824	0,066475	-0,029918	0,002324	-0,006899	0,001873	-0,014959	0,001162
1 by 6	-0,001658	0,003747	-0,4426	0,701306	-0,017779	0,014463	-0,000829	0,001873	-0,008890	0,007231
2 by 4	-0,046743	0,003747	-12,4757	0,006364	-0,062864	-0,030622	-0,023372	0,001873	-0,031432	-0,015311
2 by 6	0,030560	0,003747	8,1564	0,014701	0,014439	0,046681	0,015280	0,001873	0,007219	0,023340

Tableau 4.3 : Liste des effets confondus

Factor	Confounding of Effects (PLANFONTIONNETU.STA	
	Alias 1	Alias 2
(1)A		
(2)B		
(3)C		
(4)D		
(5)E		
(6)F		
1 by 2	3*5	
1 by 3	2*5	
1 by 4	5*6	
1 by 5	2*3	4*6
1 by 6	4*5	
2 by 4	3*6	
2 by 6	3*4	

Pour qu'une variable soit significative, le critère P doit être inférieur à un certain seuil. Le seuil le plus souvent utilisé est 0.05. Donc, selon ce critère, les variables A C D et F sont les variables les plus significatives sur les effets principaux. Pour les effets secondaires, les effets 1-3 2-4 et 2-6 sont significatifs. Avant d'analyser ce que représente ces effets secondaires, il faut regarder avec quels autres effets secondaires, ils sont confondus. Les effets 2-4 et 2-6 sont respectivement confondus avec les effets 3-6 et 3-4. Donc, les trois effets secondaires les plus significatifs sont le produit de la variable la plus significative, soit la variable 3 avec les autres variables significatives.

En conclusion, selon l'analyse du plan factoriel, les variables les plus influentes sur la production du MAV sont A, C, D, et F. Ces lettres représentent respectivement les variables 1, 5, 7 et 18 du procédé.

4.2.2.3. Méthode du gradient

Dans le cas du plan factoriel, un nombre limité de variables ont été examinées. Il serait intéressant d'utiliser une méthode pour s'assurer que toutes les variables soient dans le plan factoriel. De cette façon aucune variable significative ne peut être oubliée. Une façon simple d'estimer l'intensité du gradient est de prendre uniquement le gradient au point d'opération initial. Mais, cette méthode n'est pas bonne en général, car le système est souvent non linéaire. Ainsi, le gradient d'une certaine variable manipulée peut changer de valeur rapidement selon le point de fonctionnement, tandis qu'une autre variable peut avoir un gradient constant sur une plus grande plage d'opération. Malgré le fait que le gradient soit inférieur au point d'origine pour la seconde valeur, elle peut avoir une plus grande influence sur la fonction objectif. Pour corriger ce problème, la moyenne de plusieurs points est utilisée pour calculer les gradients dans l'espace réalisable du procédé.

La Figure 4-3 représente les gradients des 26 variables manipulées pour des points d'opération aléatoires dans le domaine du procédé. La Figure 4-4 représente, quant à elle la moyenne des gradients.

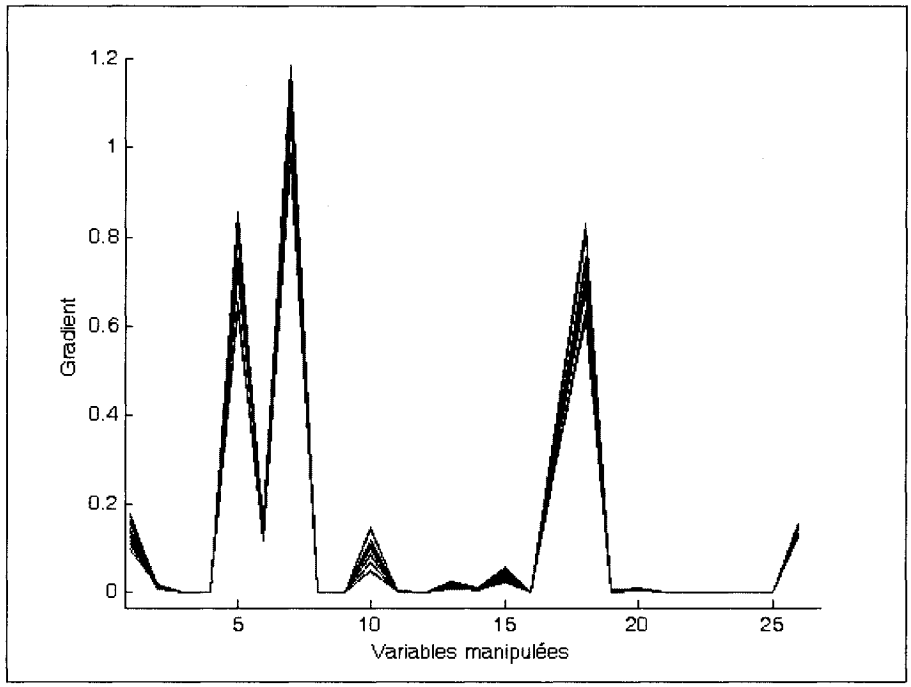

Figure 4- 3 : Valeur des gradients pour plusieurs points

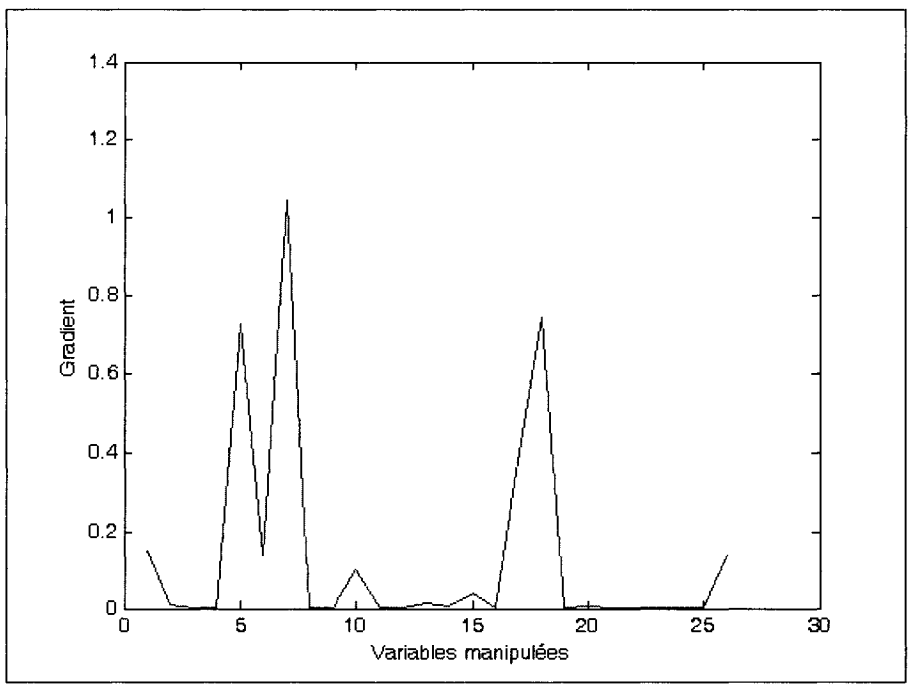

Figure 4- 4 : Valeur moyenne des gradients

Selon la méthode du gradient, les variables les plus influentes sur la production de MAV sont les variables 5, 7 et 18: la pression dans le vaporisateur, la température dans le réacteur et le pourcentage d'éthane dans le courant de recirculation (débit de purge).

Bref, les deux méthodes utilisées (plan factoriel et gradient) font ressortir les mêmes variables significatives du procédé. Le plan factoriel fait aussi ressortir la variable 1 qui par contre est la moins significative des variables sélectionnées. Pour l'optimisation, uniquement les variables 5, 7 et 18 sont utilisées.

Malheureusement, les deux méthodes utilisées n'indiquent pas bien les variables qui peuvent avoir le plus d'impact sur la fonction objectif, car elles ne considèrent pas les valeurs des variables au point d'opération. En d'autres mots, même si une variable possède une très grande influence, si le point d'opération est très près de la valeur optimale, cette variable n'apporte pas une grande amélioration à la fonction objectif. Donc, ces méthodes ne prédisent pas l'importance relative des variables. Elles peuvent uniquement aider à sélectionner ces variables.

4.3. Explication du choix de variables pour l'optimisation.

Selon les analyses effectuées, trois variables ont une influence importante, soit les variables 5, 7 et 18, la pression dans le vaporisateur, la température dans le réacteur

et le pourcentage d'éthane dans le courant de recirculation. Essayons maintenant de comprendre cette influence en regardant le procédé.

Une augmentation de la pression totale dans le vaporisateur causée par l'augmentation de la température accroît la pression partielle de l'acide acétique et de l'éthylène dans le réacteur. Selon la cinétique des équations 4-2 et 4-4, une augmentation de ces pressions partielles augmente la vitesse de réaction de production du MAV dans le réacteur, donc pour le même temps de séjour une plus grande quantité de MAV est produite. Cette quantité additionnelle de produit est acheminée au système de purification et ensuite au courant de sortie du système. Ceci cause une augmentation de la quantité finale du produit.

Selon l'équation de Arrhenius, une augmentation de la température cause une augmentation exponentielle de la vitesse des réactions. Dans ce cas précis, si l'accroissement de la vitesse de la réaction désirable est supérieur à celui de la réaction indésirable, une augmentation de la vitesse globale est possible. Une augmentation de la température dans le réacteur provoque un accroissement des vitesses, ce qui a pour conséquence d'augmenter la conversion dans le réacteur. Cette conversion supérieure produit une plus grande quantité MAV à la sortie du procédé.

L'éthane dans le système représente l'inerte. Une augmentation de l'éthane à une pression constante diminue la concentration des réactifs ce qui cause une

diminution de la conversion du réacteur, donc une diminution de MAV produit. Donc une diminution de la quantité d'éthane cause une augmentation de MAV. Cet effet inverse à bien été constaté à l'aide de la simulation.

Fait à noter, ces trois variables sont toutes des variables qui affectent le réacteur. Ceci s'explique par le fait qu'uniquement à cet endroit, il y a production de MAV. Les autres unités du procédé permettent la à purification. Les variables des étapes de purification risquent peu d'améliorer la production, car les pertes de MAV sont faibles au cours de ces étapes. Cependant, un changement dans le système de purification peut causer un effet sur le réacteur à cause des boucles de recirculation. Cet effet peut causer une augmentation de la production de MAV.

Connaissant les variables critiques du procédé, la prochaine étape consiste à effectuer l'optimisation du procédé.

4.4. Résultats d'optimisation sur le MAV

4.4.1. Optimisation statique

Dans les procédés du génie chimique, un algorithme d'optimisation doit absolument considérer plusieurs contraintes pour représenter la réalité. Pour illustrer cette situation, un algorithme sans contrainte est d'abord implanté au système, l'étape

A. Lorsque l'algorithme sans contrainte ne donne pas la même valeur, la projection sur contrainte active est implantée, l'étape B. Finalement, l'étape C est implantée quand le glissement sur contrainte ne permet pas d'atteindre des valeurs égales pour tous les ordres des variables manipulées pour l'optimisation. Cette étape est le relâchement des contraintes. L'étape C représente l'algorithme développé.

Description du logiciel statique

Un programme a été créé pour l'optimisation statique. Les variables manipulées et les états du système sont fournis au programme pour calculer les variables de sortie en régime permanent. Comme seulement des points statiques sont observés, les contrôleurs n'ont aucun impact. Ce programme s'exécute très rapidement. Lorsque le point d'opération est situé près d'une contrainte, plusieurs points peuvent être évalués pour connaître exactement la contrainte. Un autre avantage est l'absence de dépendance avec le point précédent. Si un point est pris hors contrainte, il suffit de revenir à la valeur précédente. Le système ne risque donc pas de diverger, car pour lui il n'a jamais dépassé les contraintes.

Optimisation statique à une variable

L'optimisation est effectuée avec la variable 5, soit la pression dans le vaporisateur. C'est l'algorithme sans contrainte (étape A) qui est implanté. Au cours de

l'optimisation, cette variable tend à sa borne supérieure, ce qui cause l'arrêt de l'optimisation. Un algorithme qui tient en compte les contraintes obtiendrait le même point optimal sauf qu'il oscillerait autour de ce point à cause de l'excitation.

Tableau 4.4 : Variable 5 gain 0.01 delta 0.001.

Variable	Débit (kmol/min)	Valeur	Contrainte
5	1.2048	0.8	Borne

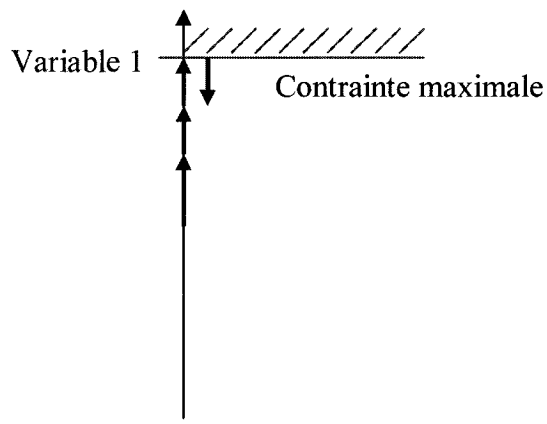

Figure 4- 5 : Optimisation à une variable.

Optimisation statique à deux variables

Cette optimisation est effectuée avec la variable 5, la pression dans le vaporisateur et la variable 7, la température dans le réacteur. Encore une fois c'est l'optimisation sans contrainte qui est effectuée (étape A). L'ordre de traitement des variables influence la valeur de la fonction objectif (Voir Tableau 4.5), ce problème est

représenté sur la Figure 3-4. Le problème survient lorsque le système se trouve pris sur une contrainte d'état et que les gradients des variables poussent le système sur ces contraintes. Le programme obtient alors un faux optimum. Cette situation est représentée à la Figure 3-7 du chapitre 3 sur les méthodes développées.

Tableau 4.5 : Variables 5 et 7 Gain 0.01 Delta 0.001.

ordre	Débit (kmol/min)	Valeur 5	Valeur 7	contrainte 5	contrainte 7
5 7	1.2698	0.8	6.46E-01	borne	Var. man. 19
7 5	0.9161	0.361	6.52E-01		Var. man. 19

La solution envisagée pour régler le problème précédent est de faire un glissement sur la contrainte (étape B). Ceci revient à mettre la contrainte active et de laisser une variable libre pour que le système puisse satisfaire cette contrainte. Les autres variables du système se déplacent et la variable libérée s'ajuste pour garder la contrainte active. Ce traitement est illustré à la Figure 3-5 au le chapitre 3. Comme le démontre les données du Tableau 4.6, lorsque le glissement de contrainte est appliqué au système, les mêmes points optimaux sont obtenus peu importe l'ordre du traitement des variables d'optimisation. En conclusion, un système avec une contrainte d'état peut être optimisé avec l'algorithme d'optimisation sans contrainte en ajoutant une étape de glissement sur les contraintes.

Tableau 4.6 : Variables 5 et 7 avec glissement Gain 0.01 Delta 0.001.

ordre	Débit (kmol/min)	Variable 5	Variable 7	contrainte 5	contrainte 7
5 7	1.27	0.8	0.647	Borne	Var. man. 19
7 5	1.27	0.8	0.647	Borne	Var. man. 19

Optimisation statique à trois variables

Pour l'optimisation à trois variables (étape B), la variable 18, le pourcentage d'éthane dans le courant de recirculation, est ajoutée aux deux variables traitées dans la section précédente. Une fois encore, une différence de la valeur de la fonction objectif est présente selon l'ordre des variables utilisées, et ce, malgré le glissement sur les contraintes. Le problème est la première contrainte rencontrée qui est considéré active, même si elle n'est pas active pour l'optimum. Comme il n'y a pas de relâchement de contrainte, un faux optimum est créé. Les résultats obtenus montrent que la contrainte 45 (borne supérieur de la variable manipulée 19) ne doit pas être active à l'optimum. Les parcours d'optimisation croissent cette contrainte et ont pour effet de la garder active. Il se crée donc un mauvais optimum (Voir Tableau 4.7). C'est le cas des multiples contraintes du chapitre 3 de la méthode développée. La Figure 3-7 représente le problème.

Tableau 4.7 : Variables 5, 7 et 18 Gain 0.01 Delta 0.001

ordre	Débit (kmol/min)	Valeur 5	Valeur 7	Valeur 18	Contrainte 5	Contrainte 7	Contrainte 18
5 7 18	1.397	0.8	0.645	-0.641	Borne	Var. man. 19	Var. man. 20
5 18 7	1.397	0.8	0.645	-0.642	Borne	Var. man. 19	Var. man. 20
7 5 18	1.397	0.8	0.64	-0.641	Borne	Var. man. 19	Var. man. 20
7 18 5	1.394	0.642	0.646	-0.8	Var. man. 20	Var. man. 19	Borne
18 5 7	1.429	0.8	0.562	-0.8	Borne	Var. man. 20	Borne
18 7 5	1.394	0.642	0.646	-0.8	Var. man. 20	Var. man. 19	Borne

La solution au problème rencontré consiste en un relâchement pour s'assurer que la contrainte active soit satisfaite au point optimum (étape C). Pour ce faire, une fois la contrainte rencontrée, elle est considérée active durant l'optimisation de toutes les autres contraintes. Lorsque c'est de nouveau à son tour d'être optimisée, celle-ci est relâchée et des itérations sont effectuées pour s'assurer que le gradient tend vers la direction de la contrainte. Si une autre contrainte est rencontrée, cette nouvelle contrainte est considérée active. Dans le cas contraire, la première redevient active. La Figure 3-8 au chapitre précédent représente ce cas.

Selon les résultats obtenus au le Tableau 4.8, tous les chemins se dirigent vers le même optimum. Bref, cette méthode permet d'optimiser un système avec plusieurs contraintes d'états.

Tableau 4.8 : Variables 5, 7 et 18 avec relâchement de contrainte Gain 0.01 Delta 0.001

ordre	Débit (kmol/min)	Valeur 5	Valeur 7	Valeur 18	Contrainte 5	Contrainte 7	Contrainte 18
5 7 18	1.429	0.8	0.562	-0.8	borne	Var. man. 20	borne
5 18 7	1.429	0.8	0.562	-0.8	borne	Var. man. 20	borne
7 5 18	1.429	0.8	0.562	-0.8	borne	Var. man. 20	borne
7 18 5	1.429	0.8	0.562	-0.8	borne	Var. man. 20	borne
18 5 7	1.429	0.8	0.562	-0.8	borne	Var. man. 20	borne
18 7 5	1.429	0.8	0.562	-0.8	borne	Var. man. 20	borne

Conclusion optimisation statique

Pour l'optimisation statique, la nécessité de chaque étape d'optimisation avec contrainte a été démontrée. Cette dernière méthode consiste à appliquer l'algorithme statique jusqu'à ce qu'une contrainte soit atteinte. Si cette contrainte est une borne, l'algorithme considère cette contrainte comme un saturateur de cette variable. L'optimisation avec une variable est un exemple de ce principe. Deuxièmement, si cette contrainte est un état alors l'algorithme fait un glissement sur cette contrainte. La nécessité de cette dernière étape à été montrée avec l'optimisation à deux variables. Ensuite, l'algorithme effectue une optimisation avec toutes les variables. Lorsque c'est de nouveau le tour de la variable qui est associée à la contrainte active, celle-ci est relâchée pour vérifier si une autre contrainte est plus appropriée pour poursuivre

l'optimisation ou même si le système doit continuer sans contrainte active. L'obligation de cette dernière étape est abordée dans l'optimisation à trois variables.

L'optimisation à une variable a permis d'augmenter la fonction objectif à une valeur de 1.20 kmol/min. Ceci correspond à une amélioration de 53.1 %. L'optimisation atteint la borne de la variable manipulée. Lors de l'optimisation à deux variables, la fonction objectif atteint 1,27 kmol/min ou 61.4 % d'amélioration. La variable 5 atteint sa borne supérieure, tandis que la variable 7 est jumelée à la contrainte sur la variable manipulée 19 du système. Finalement, l'optimisation à trois variables permet d'atteindre une valeur de 1.43 kmol/min, ce qui représente une amélioration de 81.6 % comparativement au point d'opération initial.

4.4.2. Optimisation dynamique

L'optimisation dynamique représente mieux la réalité, car il ne suffit pas de regarder uniquement le comportement du système en régime permanent. Il faut également considérer l'état transitoire. En considérant les états transitoires, les contrôleurs prennent leur importance. Il est important de noter que les contrôleurs ne doivent pas être uniquement conçus pour leur point d'opération, mais ils doivent être également stables sur toute la plage d'opération. Pour ce faire, un compromis sur l'agressivité des contrôleurs doit être réalisé pour favoriser la robustesse. Le lecteur est invité à consulter les contrôleurs en Annexe D.

L'optimisation dynamique réalisée dans ce projet est une optimisation en régime quasi-permanent. Ce qui veut dire qu'après chaque changement, un certain temps doit s'écouler pour que l'état transitoire se termine et que le système devienne en régime permanent. Dans le cas de la simulation du MAV, un temps de 300 minutes est nécessaire pour que le système tende vers un régime permanent. Il est impossible de pouvoir connaître exactement une contrainte d'état, car l'influence des variables manipulées sur cette contrainte n'est pas connue. Il est impossible de faire plusieurs itérations pour identifier précisément la valeur des variables contrôlées à la contrainte en raison du temps pour revenir au régime permanent, car chaque itération ajout 300 minutes à la simulation. De plus, lorsque le système dépasse une contrainte, il peut devenir instable et il n'est pas toujours possible de revenir dans le domaine stable.

Légère dérive du système du MVA

La Figure 4-6 montre une légère dérive dans l'ensemble de la simulation de production du monomère d'acétate de vinyle. Cette légère dérive est causée par la boucle de recirculation des inertes qui possède une dynamique très lente. Les contrôleurs accélèrent énormément cette boucle. Toutefois, cette légère dérive reste toujours présente. Pour l'optimisation, cette dérive est prise en compte lors des calculs des différences entre les valeurs de la fonction objectif.

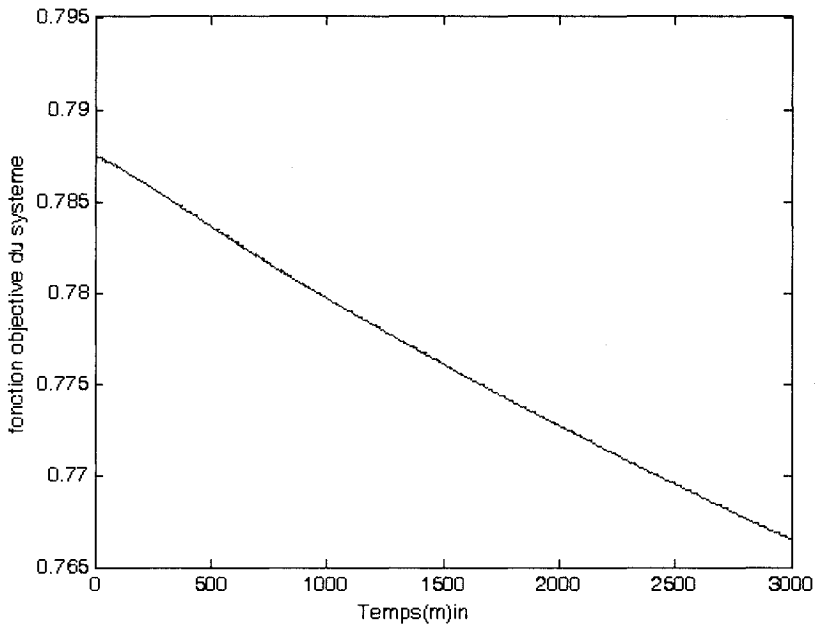

Figure 4- 6 : Dérive de la fonction objectif.

Optimisation dynamique à une variable

L'optimisation dynamique est effectuée d'abord avec la variable 5. Le profil de la variable manipulée est très semblable au profil obtenu en statique. Elle commence à sa valeur d'opération, puis elle augmente jusqu'à sa borne maximale et finalement elle se met à osciller autour de cette valeur. À la Figure 4-8, la fonction objectif maximale atteinte est de 1.18 comparativement à 1.20 en statique. Cette différence est probablement causée par la légère dérive de la simulation.

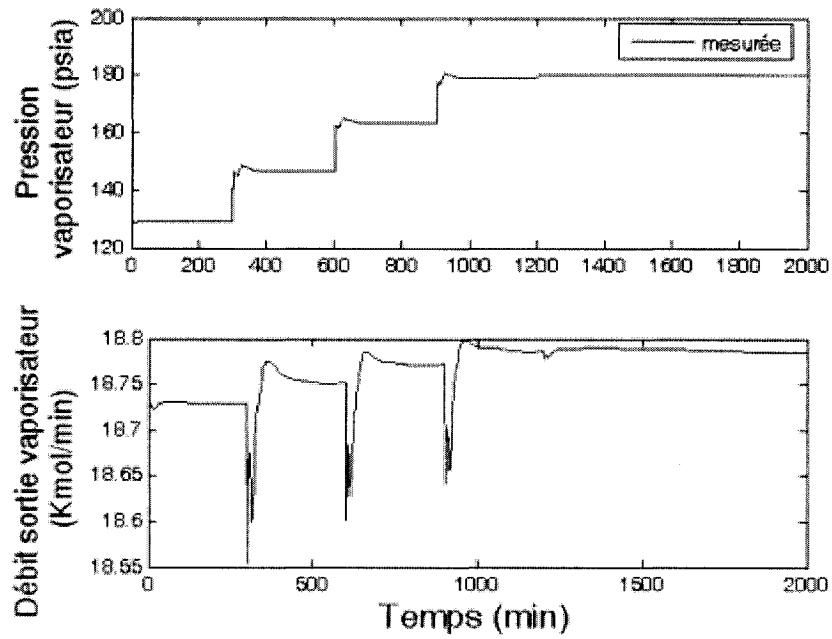

Figure 4- 7 : Variable manipulée 5 pour l'optimisation à une variable

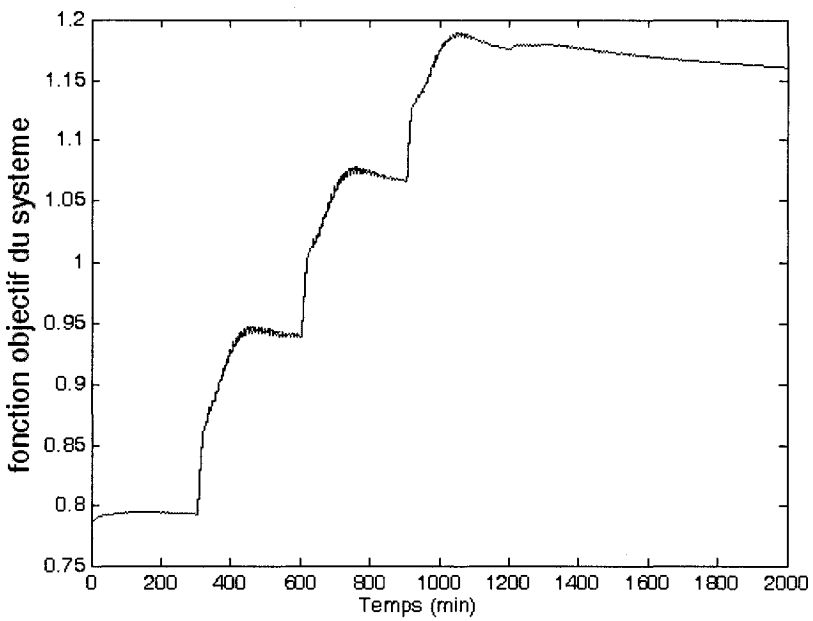

Figure 4- 8 : Fonction objectif pour l'optimisation à une variable

Optimisation dynamique à 2 variables

Dans un cas dynamique, il est impossible de connaître exactement la contrainte d'état. Il faut trouver une façon de s'assurer que le système revient à la contrainte après chaque changement. La méthode utilisée est d'insérer un contrôleur entre la contrainte et une variable mesurée pour que le système revienne à la contrainte. Un second problème avec la dynamique est que le système ne doit jamais se trouver au-delà de la contrainte. Pour régler ce problème, une contrainte parallèle à la véritable contrainte avec une certaine marge de sécurité est utilisée pour l'optimisation. Il est possible de dépasser cette nouvelle contrainte sans nuire à la stabilité du système. Voici, la Figure 4-9 qui représente ce nouvel algorithme.

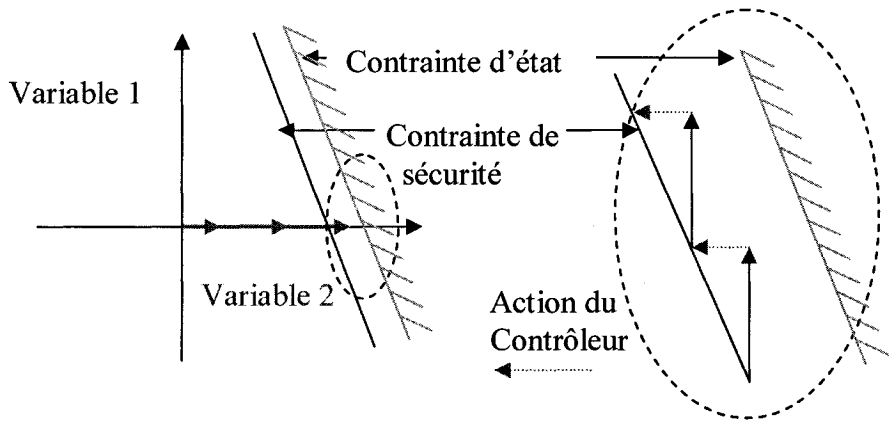

Figure 4- 9 : Solution au problème dynamique (2 variables)

L'optimisation dynamique à deux variables est effectuée avec les variables 5 et 7. Ils sont de 0.2 pour la variable 5 et de 0.01 pour la 7. Les résultats de l'optimisation avec l'ordre 5 et 7 sont présentés dans les Figures 4-10 à 4-11.

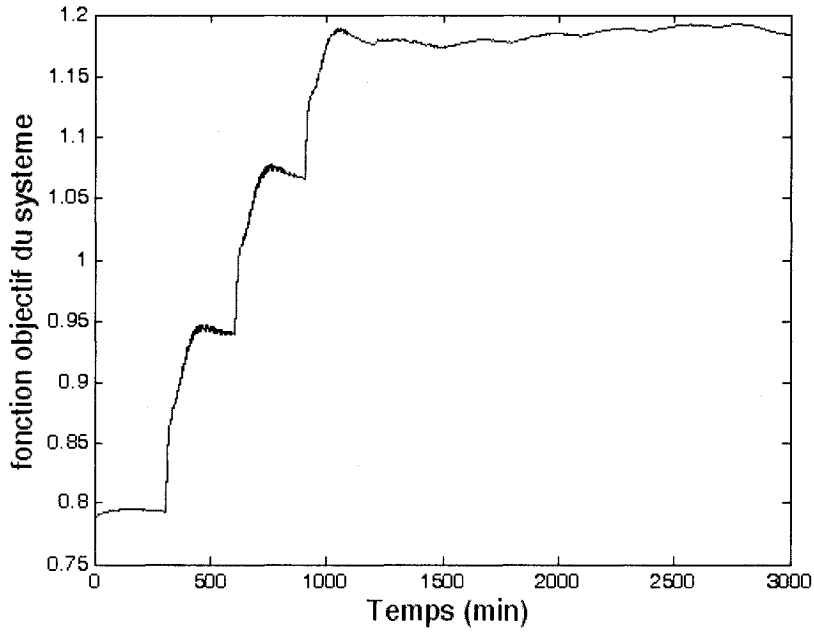

Figure 4- 10 : Fonction objectif pour l'optimisation avec l'ordre 5-7

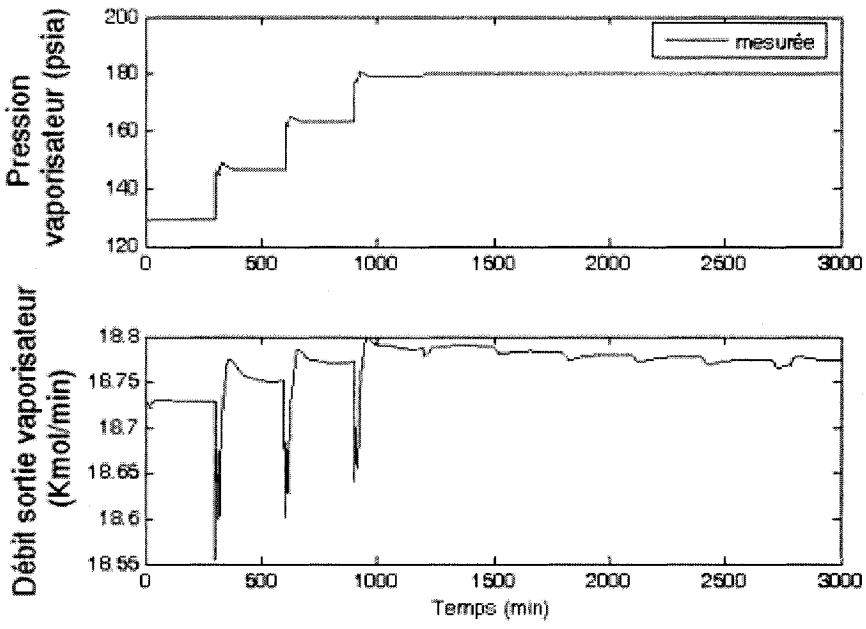

Figure 4- 11 : Variable manipulée 5 pour l'optimisation avec l'ordre 5-7

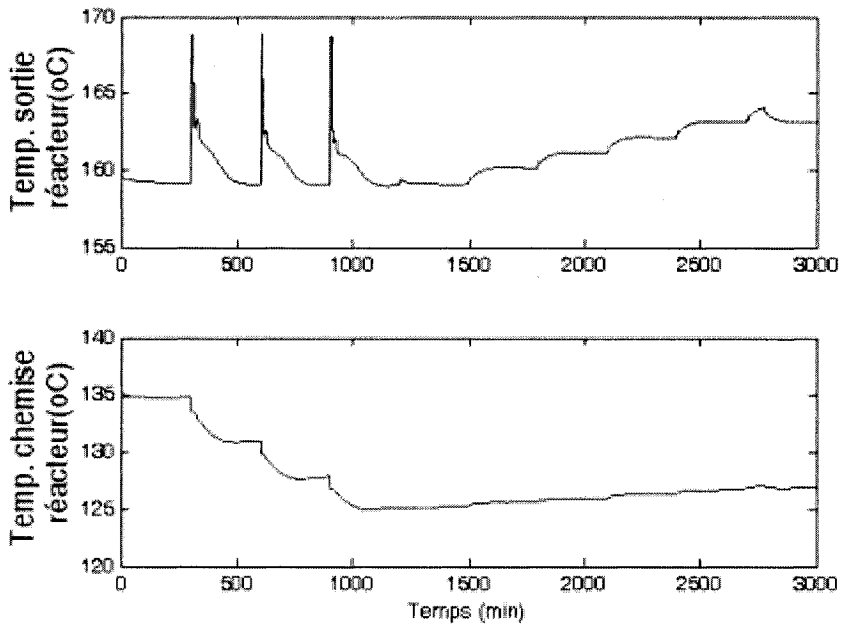

Figure 4- 12 : Variable manipulée 7 pour l'optimisation avec l'ordre 5-7

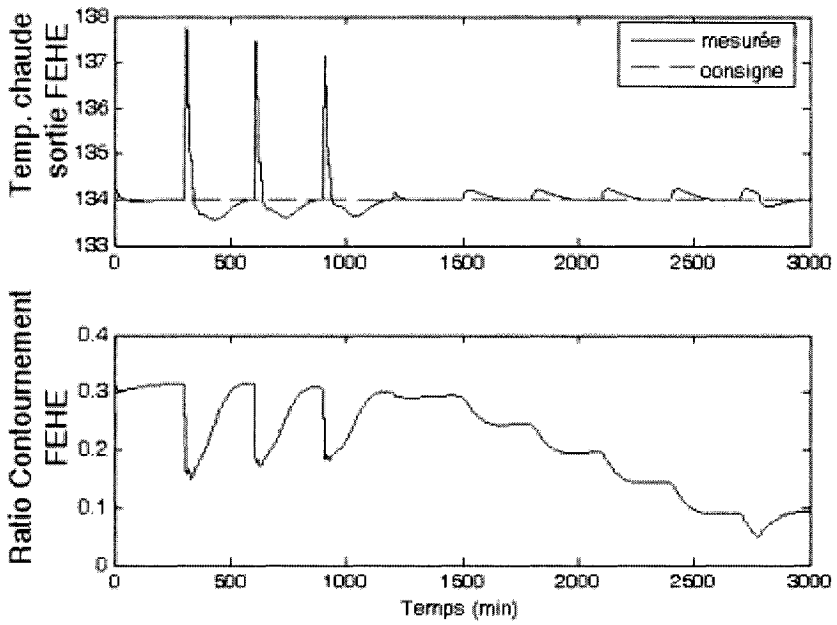

Figure 4- 13 : Contrainte du procédé pour l'optimisation avec l'ordre 5-7

La contrainte est uniquement atteinte à la fin de l'optimisation de la seconde variable comme lors de l'optimisation statique présentée dans section précédente. Les valeurs des variables et de la fonction objectif sont semblables à celle de l'optimisation statique. Les principales différences sont probablement causées par la légère dérive de la simulation et la marge de sécurité ajoutée sur la contrainte.

Lors de l'optimisation avec l'ordre des variables 7 et 5, la contrainte d'état a encore une fois été atteinte, mais une projection exacte sur la contrainte n'est pas possible en dynamique. Il est impossible de connaître exactement la contrainte car pour ce faire, un très grand nombre d'évaluations de la contrainte est nécessaire. Ce ci cause un temps d'attente important (300 minutes à chaque itération). En plus, il est risqué d'être sur une contrainte en dynamique car le comportement autour de cette contrainte peut être compliqué. Il ne faut pas que la contrainte soit dépassée durant les étapes transitoires, ce qui peut causer des instabilités. La solution pour régler ce problème est de déterminer une autre contrainte parallèle à la vraie mais avec une marge de sécurité suffisante. Dans ce cas, un léger dépassement de cette nouvelle contrainte de sécurité n'est pas catastrophique pour le système.

Le second problème est de connaître la valeur des variables manipulées pour être à la contrainte précisément. La solution proposée est de faire un contrôleur entre la contrainte comme point de consigne et la variable manipulée comme variable contrôlée. Pour tous les changements des autres variables manipulées, le système repositionne la

variable manipulée pour être à la contrainte en régime permanent. La Figure 3-6 représente cette procédure. Le gain de l'algorithme de la variable 5 doit être diminué une fois que la contrainte est atteinte pour assurer la stabilité du système. Ce gain a une valeur de 0.07. Les résultats de l'optimisation dynamique des variables 5 et 7 dans l'ordre 7 et 5 sont présentés dans les Figure 4-14 à 4-17.

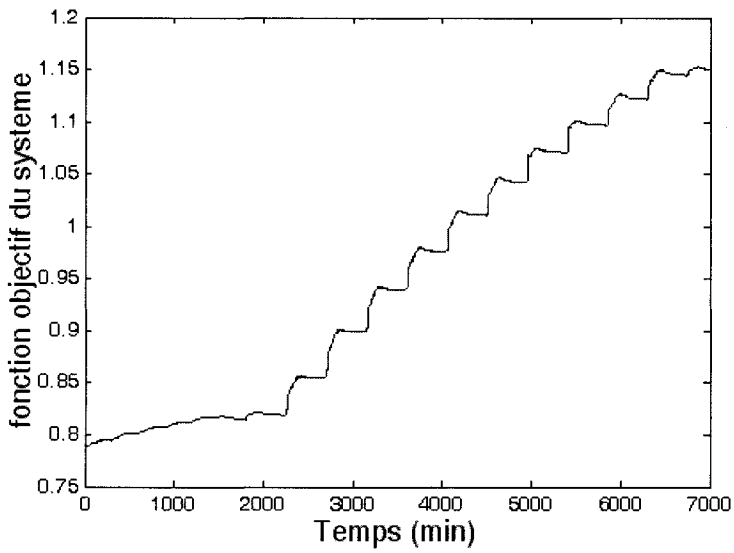

Figure 4- 14 : Fonction objectif pour l'optimisation avec l'ordre 7-5

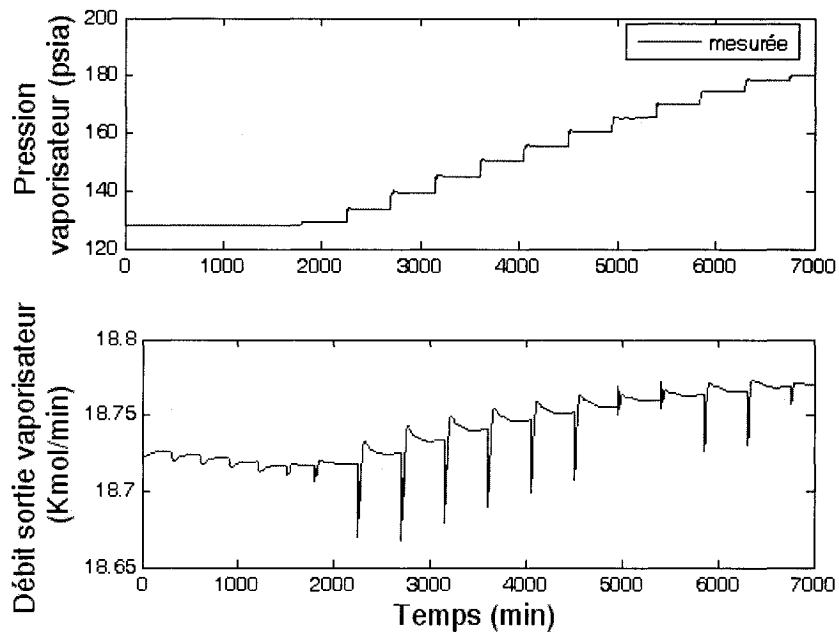

Figure 4- 15 : Variable manipulée 5 pour l'optimisation avec l'ordre 7-5

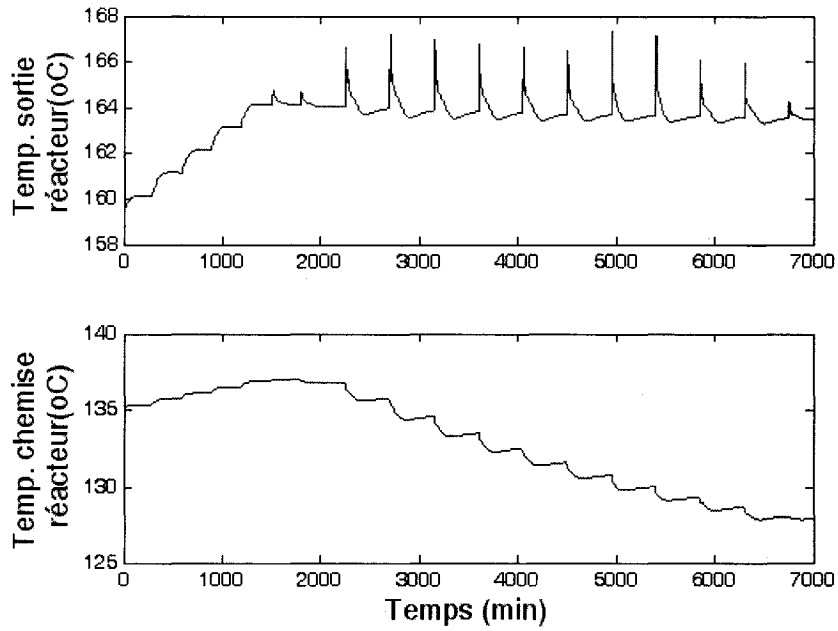

Figure 4- 16 : Variable manipulée 7 pour l'optimisation avec l'ordre 7-5

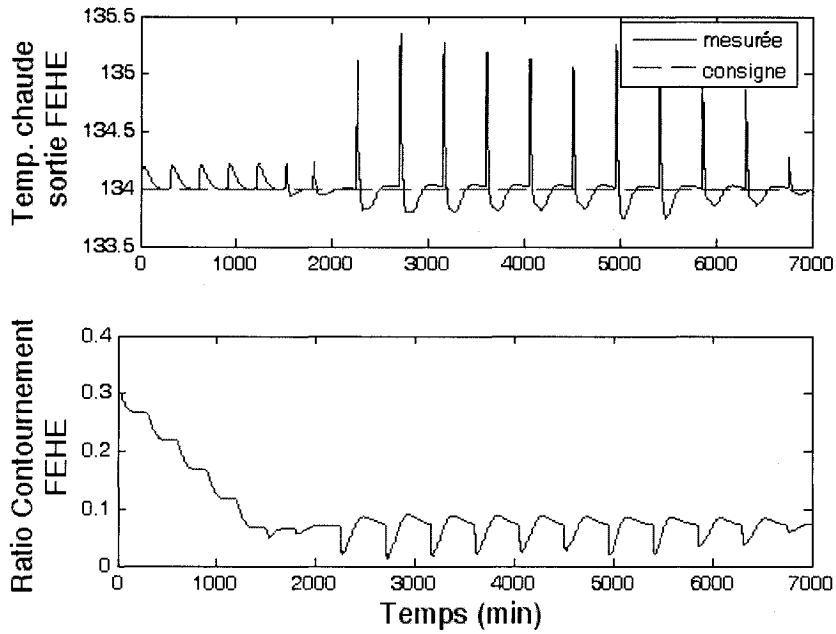

Figure 4- 17 : Contrainte 41 pour l'optimisation avec l'ordre 7-5

Contournement

La Figure 4-16 indique que le système glisse bien sur la contrainte, si on considère seulement les points avant chaque changement de variables manipulées. La Figure 4-15 montre qu'à chaque pas de la variable 5, la variable 7 diminue pour laisser une marge de manœuvre afin de respecter la contrainte. Les valeurs des variables manipulées sont presque identiques aux valeurs trouvées en statique. La légère différence est probablement causée par la marge de sécurité laissée près de la contrainte. La fonction objectif est légèrement inférieure à celle en statique, cet écart est probablement causé par la petite dérive du système.

Conclusion de l'optimisation dynamique

Lors de l'optimisation dynamique du procédé de production du MAV, l'algorithme de projection sur contrainte donne de bonnes performances. Le fait d'associer une variable à une contrainte active par un contrôleur donne des résultats très rapidement pour le système à la contrainte. Le gain de l'algorithme lorsque la contrainte devient active a été diminué.

Lors de l'optimisation à une variable, une amélioration de 48.7% est atteinte. La différence entre cette valeur et celle statique est causée par la dérive de la simulation. L'amélioration de la fonction objectif est de 49.9% pour l'ordre 5-7 et de 46.1% pour l'ordre 7-5. La différence entre ces valeurs peut être expliquée par le fait que la seconde optimisation dure beaucoup plus longtemps. La dérive de la simulation est alors plus importante. Une différence entre les valeurs obtenues en statique peut aussi être expliquée par la dérive et la marge de sécurité sur la contrainte. Toutes les valeurs pour les variables manipulées pour l'optimisation sont les mêmes.

CHAPITRE 5 Conclusions et Recommandations

Conclusions

Un algorithme pour optimiser les problèmes avec contraintes a été présenté. Sa convergence a été démontrée et elle a été comparée avec d'autres méthodes. Cette procédure est l'optimisation en temps réel avec projection sur contraintes actives. Elle est constituée d'une section pour optimiser et une section pour la projection sur les contraintes. La première section est une fonction qui prend la plus grande valeur entre le gradient et une valeur fixe pour déplacer le système. Une fois qu'une contrainte est rencontrée, la seconde section de l'algorithme entre en jeu. Cette méthode utilise une des variables de l'optimisation pour garder cette contrainte active. Une fois qu'un point stagnant est rencontré, la contrainte est relâchée pour vérifier si un autre chemin permet de poursuivre l'optimisation. Si une autre contrainte permet d'améliorer la fonction objectif, elle est à son tour considérée active.

La comparaison avec deux autres méthodes, IMC et la méthode des perturbations associé à une fonction barrière, a permis de faire ressortir les principaux avantages et inconvénients de cette méthode. Le principal avantage de cette méthode

est que si l'optimum se situe à la contrainte, il est possible d'être exactement à ce point. Un second avantage est que sa vitesse d'évaluation du gradient et son application est rapide. Le principal désavantage de cette méthode est qu'il y a uniquement une variable qui est manipulée à la fois, ce qui peut nuire à la vitesse d'exécution, si plusieurs variables d'entrées pour l'optimisation sont présentes.

Pour l'optimisation statique, il a été montré que chaque étape de l'algorithme est nécessaire. Peu importe l'ordre des variables d'optimisation, le même résultat est atteint. Le pairage des variables et des contraintes actives fonctionne bien. En second lieu, l'optimisation dynamique du procédé de production du MAV avec l'algorithme de projection sur contrainte a bien fonctionné. Le fait d'associer une variable à une contrainte active par un contrôleur mène le système à la contrainte rapidement. Le gain de l'algorithme lorsque la contrainte devient active a été diminué.

L'optimisation statique à une variable a permis d'améliorer la fonction objectif de 53.1 %, tandis que l'optimisation dynamique à une variable donne une amélioration de 48.7%. La différence entre les valeurs est causée par la dérive de simulation. En effet, les valeurs des variables manipulées sont exactement les mêmes et elles correspondent à la valeur de la borne de ces variables. L'optimisation à deux variables en mode statique a produit une amélioration de 61.4 %, tandis que l'optimisation dynamique obtient deux résultats différents selon l'ordre des variables. Ces rendements sont de 49.9% pour l'ordre 5-7 et de 46.1% pour l'ordre 7-5. La disparité entre les deux

valeurs dynamiques est causée par la différence de durée de la simulation, car la dérive est plus importante si la durée est plus longue. L'écart entre les modes statique et dynamique est aussi expliqué par la dérive, mais également par la marge de sécurité dans l'optimisation dynamique. Finalement, l'optimisation statique à trois variables a permis d'atteindre une amélioration de 81.6 % comparativement au point d'opération. Tandis qu'en mode dynamique, l'apport de la variable 7 est inexistant, car la marge de sécurité pour la contrainte et la dérive de la simulation anéantissent l'amélioration de cette variable. Donc, si une optimisation est effectuée sur ce système, la variable 5 apporte une amélioration satisfaisante.

Recommandations

Maintenant que l'optimisation avec l'algorithme développé a été effectuée sur le procédé de production du monomère d'acétate, il serait intéressant d'effectuer l'optimisation de ce procédé avec la méthode de perturbation et la méthode IMC, afin de comparer de façon numérique ces méthodes dans un procédé complexe.

Maintenant que le fonctionnement de l'algorithme avec projection sur contraintes actives est démontré, il serait intéressant d'appliquer le principe de glissement sur contrainte active avec une différente méthode pour estiment le gradient. Par exemple, la méthode de perturbation peut être utilisée. Ceci aurait pour effet de faire varier toutes les variables à la fois, ce qui est un désavantage de notre méthode. Le

problème va être la vitesse d'estimation du gradient. Donc, une méthode qui estime rapidement plusieurs gradients peut être appliquée à la méthode de glissement sur contrainte. Bref, la méthode de projection sur contraintes actives peut être appliqué à un grand nombre d'algorithmes qui ne considèrent pas les contraintes pour augmenter leur efficacités. En d'autres mots, la méthode de glissement sur contraintes actives pourrait remplacer la fonction barrière dans une grande quantité d'algorithmes déjà existants.

Bibliographie

ARIYUR, Kartik B,. 2003. *Real-time optimization by extremum-seeking control*, Wiley-Interscience. New Jersey. 236p.

ARIYUR, Kartik B., KRSTIC, Miroslav, 2003 b. «Slope seeking: a generalization of extremum seeking». *International Journal of Adaptive Control and Signal Processing*. 18, 1-22

BANERJEE, Pranob. SHAH, Sirish L. NIU, Shaohua. FISHER, D. Grant. 1995. «Identification of dynamic models for the shell benchmark problem. ». *Journal of Process Control*. 5-2. 85-97

BAZARAA M.S, Shetty C.M. 1979. «Nonlinear Programming Theory and Algorithms». John Wiley & Son. New York. 560 p.

BELL, M.L. SARGENT, R.W.H. 2000. «Optimal control of inequality constrained DAE systems». Computers and Chemical Engineering. 24. 2385–2404

BELLIVEAU, Caroline. 2004. *Contrôle avancé d'une unité de production d'acétate de vinyle*. PFE. École Polytechnique de Montréal. 47 p.

BEQUETTE, B. WAYNE. 2003. *Process Control.* Prentice Hall. New Jersey. 769 p.

BIEGLER, Lorenz T. GROSSMANN, Ignacio. 2004 «Retrospective on optimization.». *Computers and Chemical Engineering.* 28-8. 1169-1192.

BORELLI, Francesco. 2005. «Book Reviews Constrained Optimal Control of Linear and Hybrid Systems». *IEEE Transactions on Automatic Control.* 50-7. 1069-1070

CASTRO, J.J. 2004. «A pulp mill benchmark problem for control : Application of plantwide control design». *Journal of Process Control.* 14. 329–347

CHEN, Rong, MCAVOY, Thomas, Kedar, DAVE. 2003. «A nonlinear dynamic model of a vinyl acetate process. ». *Ind. Eng. Chem. Res.* 42. 4478–4487.

CHIN, C.S. 2003. «Control of the alstom gaseifier benchmark problem using *h*2 methodology. ». *Journal of Process Control.* 13 .759–768.

DAVISON, E.J., CHADHA, K.J. 1972. «On the control of a large chemical plant by using modal analysis. ». *Automatica.* 8. 263–273.

DEHAAN, Darryk. GUAY, Martin. 2005. «Extremum-seeking control of state-constrained nonlinear systems. ». *Automatica.* 41. 1567-1574

DESBIENS A. SHOOK A.A. 2003. «IMC-Optimization of a Direct Reduced Iron Phenomenological Simulator ». *4th International Conference on Control and Automation.* Montréal. Canada. 446-450.

DOUCET J. 2004. *Design et optimisation d'une unité de synthèse d'acétate de vinyle.* PFE. École Polytechnique de Montréal. 64p.

DOWNS, J. 1993. «A plantwide industrial process control problem.». *Computers Chemical Engineering.* 17 . 245

EDGAR, Thomas F. HIMMELBLAU, David Mautner. 1988. *Optimization of chemical processes.* McGraw-Hill. New York. 652 p.

FLETCHER, Roger. 1979. *Practical methods of optimization.* John Wiley & Son. New York. 436 p.

FOSS, A. 1973. «Critique of chemical process control theory. ». *AIChE.* 19. 209–214.

GEORGAKIS, C.,PRICE, R. 1993. «Plantwide regulatory control design procedure using a tiered framework. ». *Ind. Eng. Chem. Res.* 32. 2693–2705.

GOLSHAN, M., BOZORGMEHRY BOOZARJOMEHRY, R. PISHVAIE, M.R. 2005. «A new approach to real time optimization of the Tennessee Eastman challenge problem». *Chemical Engineering Journal. 112. 33-44*

GROSSMANN, Ignacio, BIEGLER, Lorenz T., 2004, «Part II. Future perspective on optimization». *Computers and Chemical Engineering*, 28-8. 1193-1218

GUAY, M. DOCHAIN, D. PERRIER, M. 2005. «Adaptive extremum-seeking control of nonisothermal continuous stirred tank reactors ». *Chemical Engineering Science*. 60. 3671-3681

HUDON, Nicolas. 2004. *Initiation à Matlab. URCPC, École Polytechnique de Montréal.* 49 p.

KLATT, K-U. ENGEL, S. 1998 «Gain scheduling trajectory control of a continuous stired tank reactor». *Computers & Chemical Engineering*, 22-4-5. 491-502

KRSTIC, Miroslav. 2000. «Performance Improvement and Limitations in Extremum Seeking Control». *Systems & Control Letters*. 39. 313-326

KRSTIC, Miroslav. WANG, Hsin-Hsiung. 2000 b. «Stability of extremum seeking feedback for general nonlinear dynamic systems, *Automatica*, 36, 595-601

LARSSON, Truls. SKOGESTAD, Sigurd. 2000. «Plantwide control—A review and a new design procedure». *Modeling, Identification and Control. 21*-4. 209–240

LEMIEUX, M. 2004. *Design et optimisation d'une stratégie de contrôle d'une unité de purification du MAV.* PFE. École Polytechnique de Montréal. 68 p.

LUYBEN, W. TYRÉUS, B LUYBEN M. 1997 «Plantwide control design procedure». *AIChE* . 43. 3161-3174

MACIEJOWSKI, Jan Marian. 2001. *Predictive Control with Constraints.* Prentice Hall. England. 331 p.

MASON, R. L., GUNST, R. F., HESS, J. L. 2003. *Statistical design and analysis of experiments : with applications to engineering and science.* 2nd ed. Wiley-Interscience. New Jersey. 728 p.

MCAVOY, Thomas J., 1983. *Interaction analysis : principles and applications.* Research Triangle Park, : Instrument Society of America. USA. 245 p.

MCAVOY, T. 1994. «Base control for the Tennessee Eastman problem. ». *Comput. Chem.Eng.*, 19. 383–413.

MCAVOY, T. 1999. « Synthesis of plantwide control systems using optimization. ». *Ind. Eng. Chem. Res.* 38. 2984–2994.

MCAVOY T., CHEN, R. 2003. «Plantwide system design : Methodology and application to the vinyl acetate process. ». *Ind. Eng. Chem.* 42. 4753– 4771.

MONTGOMERY, Douglas. C 2005. *Design and Analysis of Experiments*, 6th ed., John Wiley & Sons. New York. 704 p.

OGUNNAIKE, Babatunde Avodeji. 1994. *Process dynamics, modeling, and control*. Oxford University Press. New York. 1260p.

POMERLEAU, Dominique. DESBIENS, Andre. BARTON, Geoff. W. 2003. «Real time optimization of an extrusion cooking process using a first principles model.». *IEEE Conference on Control Applications – Proceedings*. 1. 712-717

POMERLEAU, Dominique. DESBIENS, André. HODOUIN, Daniel. 2003 b. « Optimization of a Simulated Iron-Oxide Pellets Induration Furnace.». *11th Mediterranean Conference on Control and Automation.* Rhodes. Grèce. T7-047

RAKOTE, B. L.. 1981. *Factorial design.* Wiley. New York. 209 p.

ROTEA, Mario A. 2000. «Analysis of Multivariable Extremum Seeking Algorithms». *Proceedings of the American Control Conference.* Chicago. Illinois. 433-437

SKOGESTAD, S. 2000. «Plantwide control : The search for self-optimizing control structure. ». *Journal of Process Control*, 10. 487–507.

KOGESTAD, S. 2004. «Control structure design for complete chemical plants. ». *Journal of Process Control.* 28. 219–234.

STEPHANOPOULOS, G. 1983. «Synthesis of control systems for chemical plants - a challenge for creativity. ». *Comput. Chem. Eng.* 7. 331–365.

STERNBY, J. 1980 «Extremum control systems—an area for adaptive control ? ». *1980 Proceedings of the joint American control conference.* San-Francisco. CA. USA. WA2-A/12

TYREUS, M., LUYBEN, L. 1998. «An industrial design/control study for the vinyl acetate monomer process. ». *Computers Chemical Engineering*. 22. 867–877.

VASBINDER, E.M., HOO, K.A. 2003. «Decision-based approach to plantwide control structure synthesis. ». *Ind. Eng. Chem. Res.*, 42. 4586–4598

VASSILIADIS, V.S, FLOUDAS, C.A. 1997 «The modified barrier function approach for large-scale optimization». *Computer Chem Enging.* 21. 855-874

WANG, Hsin-Hsiung. 1998. *Bifurcation Control and Extremum Seeking for Aeroengine Compressors and Bioreactors.* Ph.D. Thesis, University of Maryland, USA

WILLIAMS, T.J., OTTO, R.E. 1960. «A generalized chemical model for the investigation of computer control. ». *AAIEE Trans. part 1 (Comm. and Electronics)*. 79. 458–473

Annexes

Annexe A : Le procédé

Le réacteur

Voici une liste des principales hypothèses sur le réacteur qui ont été posées dans la réalisation du fichier Matlab :

1- L'hypothèse du réacteur piston est considérée, alors il n'y a pas de différence radiale de température, de concentration et de vitesse. La diffusion axiale est négligeable par rapport au débit. L'énergie potentielle et cinétique sont considérées négligeables dans les calculs du bilan de chaleur.

2- Le transfert de masse et de chaleur entre le catalyseur et le milieu sont très rapides, donc les températures et les concentrations dans ces deux phases sont toujours identiques.

3- La perte de pression est linéaire en fonction de la position sur le tube et elle est invariante avec le temps.

4- La température de la paroi de la chemise est uniforme et elle sert de variable manipulée. La dynamique de la vapeur dans la chemise n'est donc pas modélisée. La valeur de la température de cette paroi au point d'opération est de 159.17^0C.

Échangeur de chaleur

L'échangeur de chaleur est intégré, ce qui veut dire que le courant chaud est utilisé pour réchauffer le froid en se refroidissant lui-même, ce qui représente une économie d'énergie. Le courant de sortie du réacteur se refroidi en préchauffant le courant qui se dirige vers le vaporisateur. La dynamique de l'échangeur de chaleur n'a pas été modélisée dans le fichier Matlab car la dynamique n'a pas d'effets significatifs sur le restant du procédé. Seulement un petit délai de temps est ajouté à la sortie de l'échangeur pour représenter sa dynamique. Il y a une seule variable manipulée dans l'échangeur de chaleur, la valve de contournement. Le point de consigne de l'échangeur est la température de sortie du courant qui se dirige vers le réacteur, mais qui passe dans l'échangeur. Cette température est de 134^0C. Un échangeur de chaleur conventionnel est ajouté à la suite de celui-ci pour diminuer la température du courant qui se dirige vers le séparateur.

Séparateur

Dans le programme Matlab, le séparateur est modélisé comme un condenseur partiel. À tous les pas de temps, un équilibre flash en régime permanent est calculé pour obtenir le débit et les compositions des courants liquides et gazeux. La température flash est estimée en ajoutant 5 degrés à la température de la chemise qui est posé uniforme. Les courants liquides et vapeurs sont considérés parfaitement séparés à la

sortie. La température de sortie du gaz est posée constante et la température de liquide est calculée avec un bilan thermique.

Il y a trois variables manipulées dans le séparateur soit les deux débits de sorties et la température de la chemise. Le volume utile du séparateur est de $4m^3$ ce qui correspond à 50 % du volume total. La pression et la température de la phase liquide au point d'opération du séparateur sont de 84.25 psia et de 40^0C.

Compresseur

La pression augmente tout au long du compresseur. Cette différence de pression est calculée en multipliant le coefficient de compression à la densité du courant d'entrée. La température de sortie est estimée à l'aide de l'hypothèse de compression isentropique. Le compresseur est suivi par un système de refroidissement. La variable manipulée est la quantité d'énergie soutirée au courant dans ce système de refroidissement. La température de sortie au point d'opération est de 80 ^{0}C.

Colonne d'absorption

La colonne d'absorption est composé de deux sections distinctes. La section du haut est composé de six étages théoriques. Son courant liquide provient du réservoir d'acide acétique, tandis que celui vapeur vient de la partie inférieure. L'unité du bas de

la colonne est composée de deux étages théoriques. Le courant d'entrée de liquide provient en partie de la section supérieure et le reste vient du courant de recirculation de cette même colonne. Le débit d'entré du gaz de cette section arrive du compresseur. La pression est supposée uniforme dans toute la colonne et le point d'opération est de 128 psia.

La dynamique de la phase gazeuse est négligée. La phase liquide et la phase gazeuse de chaque étage ne sont pas en équilibre. Le transfert de masse entre les deux phases est calculé en fonction des compostions des phases et de la température. Une quantité maximale de transfert est posée à la moitié de la concentration du composé dans le courant d'entré de la vapeur.

Le transfert d'énergie entre les deux phases est calculé en fonction des températures des phases et d'une constante de transfert de chaleur. Durant les étapes entre étages, les bilans totaux de masse de composition et d'énergie fait sur la phase vapeur servent à calculer le débit, la concentration et la température de sortie du gaz, tandis que les bilans sur la phase liquide sont utilisés pour déterminer la dynamique dans la colonne d'absorption.

Dans cette colonne, il y a trois variables manipulées : le débit de sortie du liquide, le débit d'alimentation de l'acide acétique et le débit de recirculation. Au point d'opération, le niveau du liquide est de $0.25m^3$ soit 50 % du volume utilisable. En plus,

il y a deux refroidisseurs qui sont associés à cette colonne, ils sont sur le courant de recirculation et sur celui d'alimentation d'acide acétique. La quantité d'énergie soutirée dans chacun de ces refroidisseurs est une variable manipulée. Au point d'opération la température de sortie de ces refroidisseurs est de 25 ^{0}C.

Système pour enlever le CO_2

Ce système est un système conventionnel d'enlèvement du CO_2. Il y a une variable manipulée qui est le courant d'alimentation de CO_2. Au point d'opération, la fraction molaire de CO_2 dans le courant de recirculation est de 0.73%.

Système pour enlever les gaz

Ce système n'est pas représenté sur le schéma du procédé. Il sert d'assurance pour qu'aucun gaz ne puisse entrer dans la colonne à distillée azéotropique, donc il est situé juste avant celle-ci. Il y a deux courants qui se rendent à ce système. Il y a le courant liquide du séparateur et le courant liquide de la colonne d'absorption. Ce système est un système de séparation idéal. Il retourne le courant gazeux composé de l'O_2, du CO_2, du C_2H_4 et du C_2H_6 à l'alimentation du compresseur. Le courant liquide composé de l'acide acétique, du MAV et de l'eau est alimentée à la colonne à distiller.

Colonne à distiller azéotropique

Il y a 20 plateaux théoriques dans la colonne. Il est supposé qu'il y a uniquement une phase liquide dans la colonne et qu'elle est homogène. Un profil de la pression dans la colonne est posé. Le point de bulle est utilisé pour calculer la température et la composition à chaque étage, puis le bilan énergétique est utilisé pour estimer le débit de la phase gazeuse inter étage. Les coefficients utilisés dans le décanteur sont supposés constants et indépendants de la température. Il y a aussi une hypothèse sur le fait que la température des deux phases est identique.

Il y a six variables manipulées dans cette colonne. Ces variables sont : le débit de reflux, la quantité d'énergie au rebouilleur, l'énergie au condenseur, le débit de la phase organique, le débit de la phase aqueuse et le débit en bas de colonne. Au point d'opération le volume utilisé en bas de colonne est de 2.33 m^3 ce qui représente 50 % du volume utilisable. Le volume de liquide organique et aqueux sont de 0.85 m^3 soit 50 % du volume possible. Finalement, la température dans le décanteur est de 45.84 ^{0}C.

Réservoir d'acide acétique

Le réservoir d'acide acétique sert à mélanger le débit d'acide acétique recyclé au débit d'acide acétique d'alimentation. Les deux débits d'alimentations du réservoir sont des variables manipulées.

Le Vaporisateur

Le vaporisateur sert à évaporer l'acide acétique. Le volume utile du vaporisateur est de 2.8 m^3 ce qui correspond à 70 % du volume total. Il y a deux courants d'acide acétique. Le premier est le courant de recirculation de l'acide acétique provenant de la colonne azéotropique. Cette substance est sous forme liquide. Le second effluent est le débit d'entrée de l'acide acétique qui sert à maintenir la quantité de ce réactif constant. Il y a aussi une autre entrée dans le vaporisateur, c'est le courant d'alimentation de l'éthylène qui est mélangé avec le courant de recirculation des produits qui n'ont pas réagit, donc ce courant transporte de petite quantité des autres produits du procédé ce qui fait que le vaporisateur a aussi la fonction d'assurer un bon mélange pour l'approvisionnement du réacteur.

Le vaporisateur a un système de chauffage pour fournir l'énergie nécessaire au maintient de son niveau. En plus, le vaporisateur est suivit d'un échangeur de chaleur pour augmenter la température du courant pour satisfaire les conditions d'entrées au réacteur. La température du courant sortant de cet échangeur doit être de 150 ^{0}C.

Annexe B : Résultats et analyses des résidus du plan factoriel

Tableau B- 1 : Listes des expériences

Design: 2**(6-2) design
(planfactotot.sta)

	A	B	C	D	E	F	Y
1	-1	-1	-1	-1	-1	-1	0.57091012
2	1	-1	-1	-1	1	-1	0.51508871
3	-1	1	-1	-1	1	1	0.43667016
4	1	1	-1	-1	-1	1	0.3853633
5	-1	-1	1	-1	1	1	0.73651729
6	1	-1	1	-1	-1	1	0.62147523
7	-1	1	1	-1	-1	-1	0.98561278
8	1	1	1	-1	1	-1	0.85249731
9	-1	-1	-1	1	-1	1	0.54395526
10	1	-1	-1	1	1	1	0.46593497
11	-1	1	-1	1	1	-1	0.70728798
12	1	1	-1	1	-1	-1	0.65093578
13	-1	-1	1	1	1	-1	1.17620102
14	1	-1	1	1	-1	-1	1.02719391
15	-1	1	1	1	-1	1	0.93023113
16	1	1	1	1	1	1	0.76703696

Analyse des résidus pour le plan factoriel

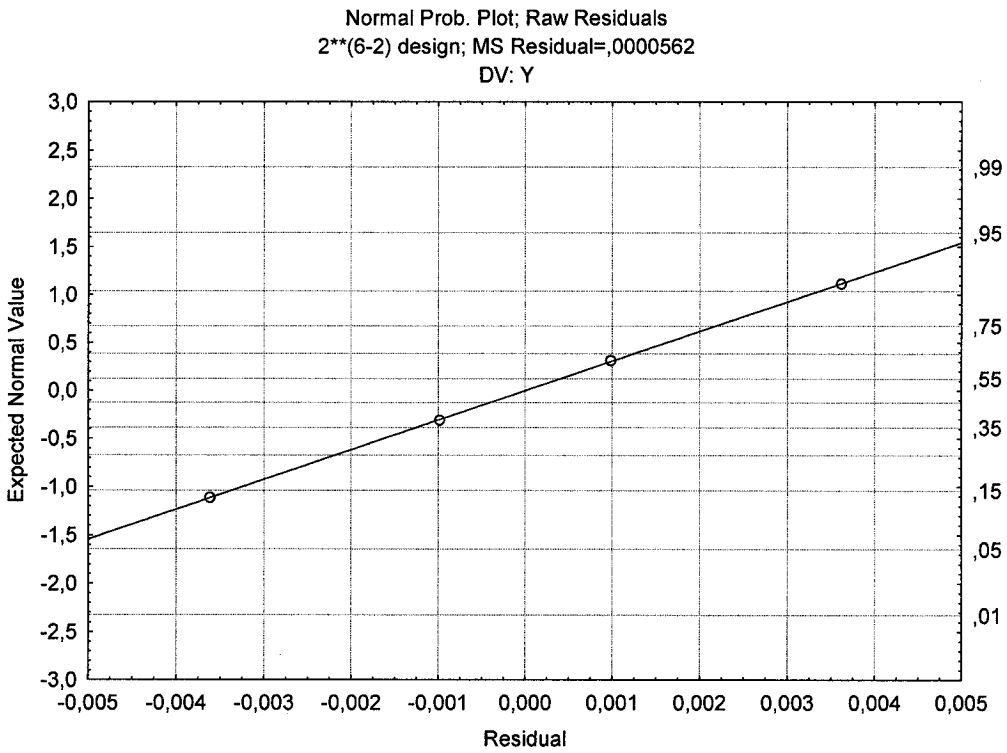

Figure B- 1 : Valeur normal en fonction des résidus

Selon le graphique précédent l'hypothèse gaussienne est respectée car il y a une droite dans le centre de la courbe.

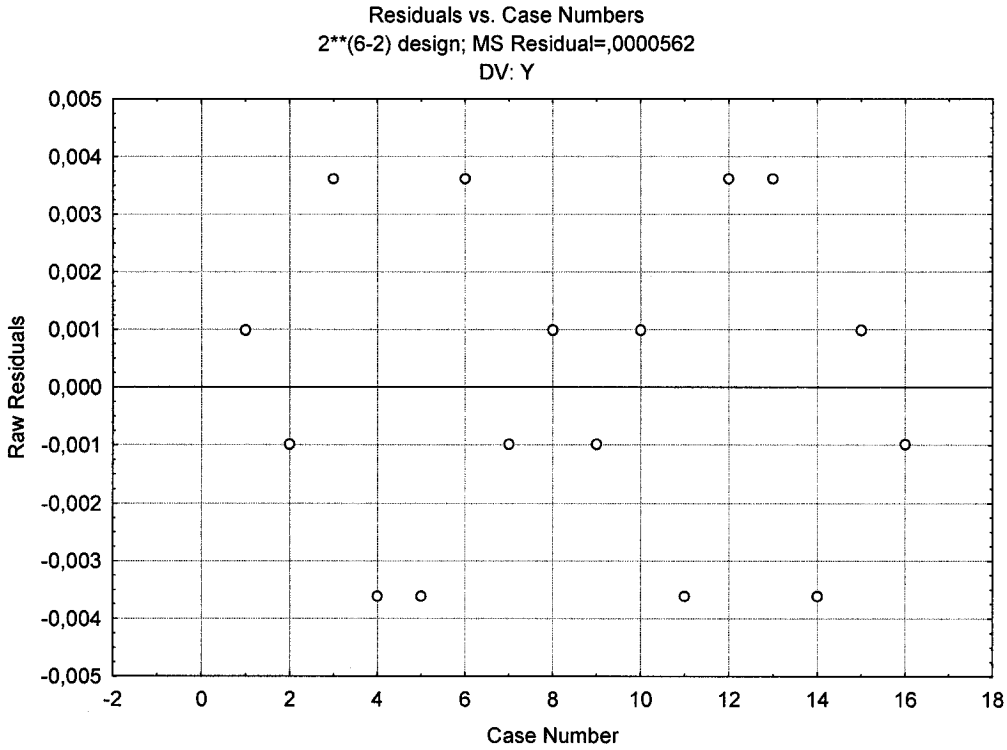

Figure B- 2 : Les résidus en fonction du numéro du cas

Ce graphique représente l'indépendance des échantillons, car les points forment un nuage.

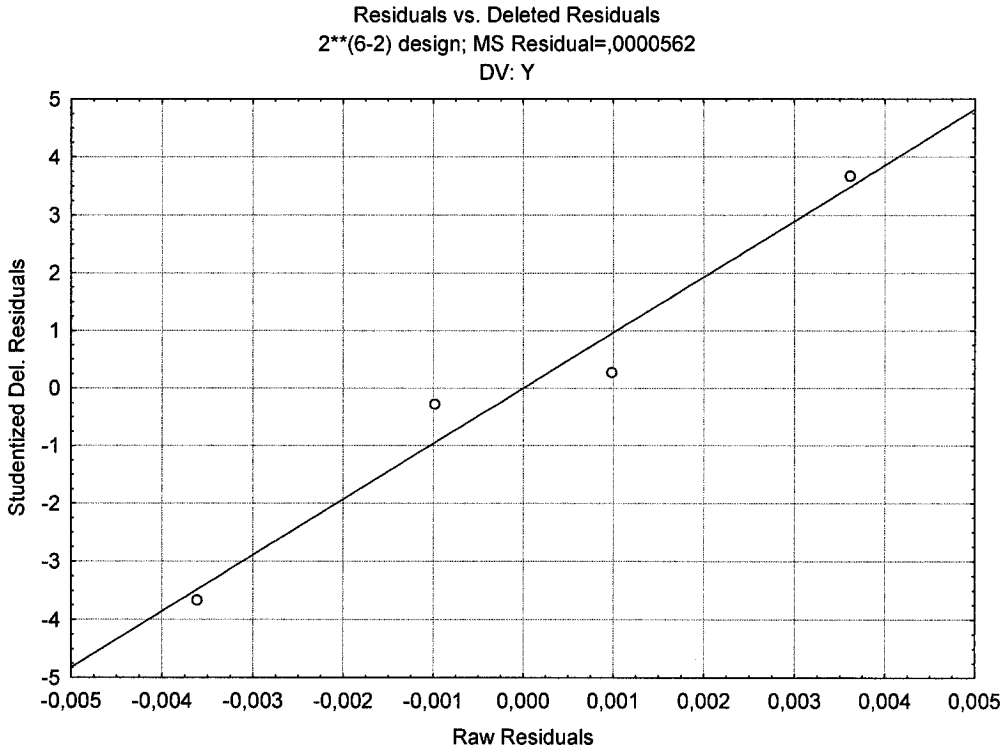

Figure B- 3 : Les résidus effacés en fonction des résidus

Cette figure montre qu'aucune valeur n'a un poids très grand sur les résultats.

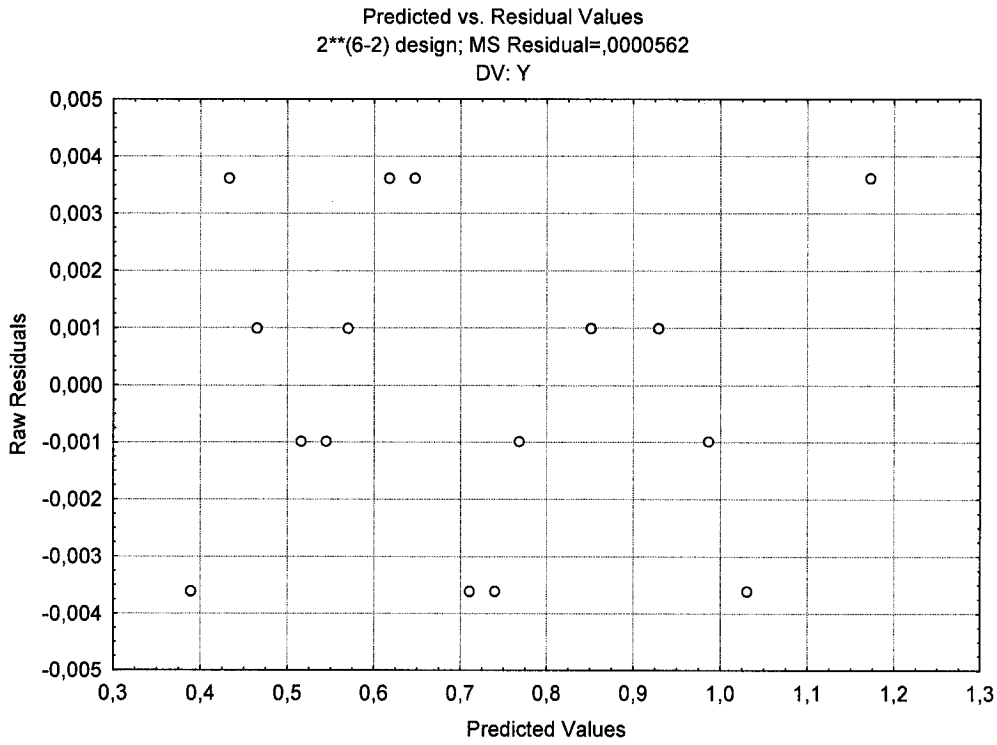

Figure B- 4 : Les résidus en fonction des valeurs prédites

Ce graphique représente l'uniformité de la variance, car c'est plus ou moins une bande horizontale.

Annexe C : Propriété de la matrice P la matrice de projection

Cette matrice est définie de la section 3-2-3.

$P^T = P$

Preuve : $P^T = I - M^T(M^+)^T = I - M^T((MM^T)^{-T}M)$. Comme MM^T est symétrique son inverse est aussi symétrique. Donc, $P^T = I - (M^T(MM^T)^{-1})M = I - M^+M = P$.

$P^2 = P$

Preuve : $P^2 = I - 2M^+M + M^+MM^+M$, Comme $MM^+ = I$,

$$P^2 = I - 2M^+M + M^+M = P.$$

P est définie semi-positif

Preuve : Comme $P = P^2 = P^TP$, donc $x^TPx = x^TP^TPx = y^Ty$ où $y = Px$. Comme $y^Ty \geq 0, \forall y$, alors $x^TPx \geq 0, \forall x$, donc P est définie semi-positif

Annexe D : Les contrôleurs

```
k1=1;
tau1=10;
k2=0.3;
tau2=20;
k3=2;%(controleur P)
k4=-0.1 ;
tau4=30;
k5 = -0.1;
tau5=10;
k6=1;
tau6=5;
k7=0.5;
tau7=10;
k8=-5; %(controleur P)
k9=-5;
tau9=20;
k11=-1;
tau11=5;
k12=-5; %(controleur P)
tau12=10000;
k14=-1;
tau14=5;
k16=-1;
tau16=1;
k17=-1;  %(controleur P)
k18=-1; %(controleur P)
k19=5;
tau19=10;
k20=-0.5;
tau20=60;
k21=20;
tau21=100;
k22=-.25 ;
tau22=5;
k23=-1; % controleur p
k24=-1; % controleur p
k25=-1; % controleur p
```

Annexe E : Comparaison numérique de deux méthodes d'optimisation

Description du procédé

La simulation utilisée pour comparer les deux méthodes est une modification du réacteur Van de Vusse. Tout d'abord, la troisième réaction, soit la réaction concurrente pour le réactif A, a été soutirée du système. Une contrainte a été ajoutée au système. Cette contrainte est représentée par le taux maximal de transfert de chaleur qui peut être enlevé du système (1000 KJ/h). Il y a deux variables manipulées dans ce système. Le taux de dilution au réacteur est la première variable manipulée et la seconde est la température dans le réacteur. La fonction objectif de ce problème est la concentration du produit B à la sortie du réacteur que l'on veut maximiser.

Les réactions irréversibles se déroulent dans un réacteur CSTR. Les deux réactions sont sous la forme suivante :

$$A \xrightarrow{k1} B \xrightarrow{k2} C \qquad \text{(E-1)}$$

Optimisation en temps réel avec projection sur contraintes actives

Cette méthode a été appliquée deux fois au réacteur, une fois en débutant avec chacune des variables manipulées. La variable manipulée qui a été utilisée pour la projection sur la contrainte active est la température, car cette variable a une plus grande influence sur la contrainte.

La figure E-1 représente la fonction objectif, les figures E-2 et E-3 représentent l'évolution des variables manipulées et finalement le comportement près de la contrainte est illustré dans la figure E-4.

Ordre : Taux de dilution suivi de Température

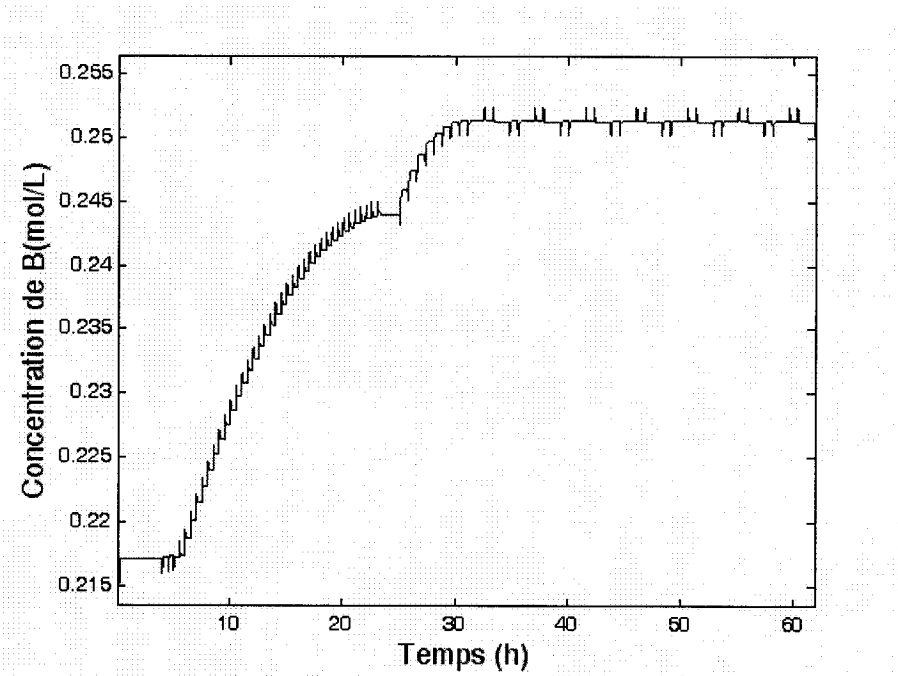

Figure E- 1 : Fonction objectif pour l'ordre DT

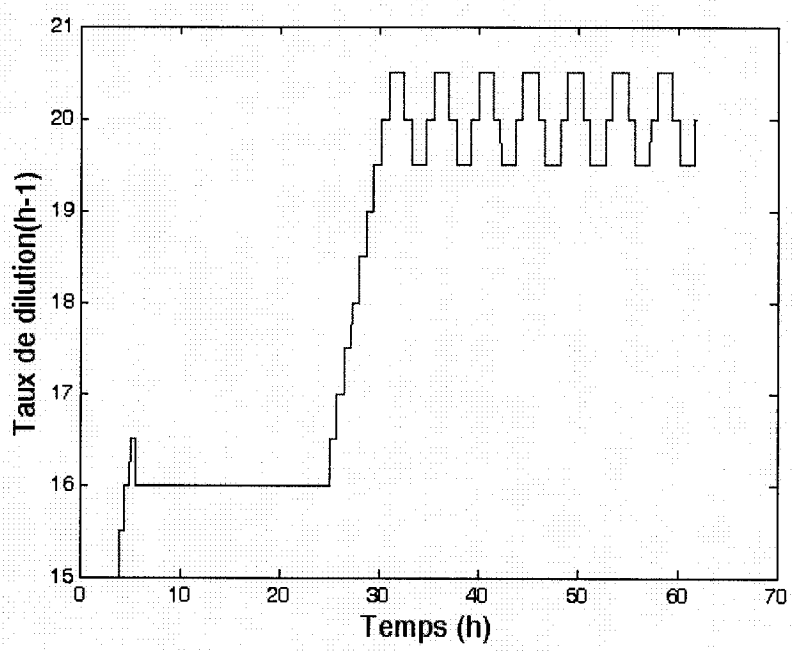

Figure E- 2 : Taux de dilution pour l'ordre DT

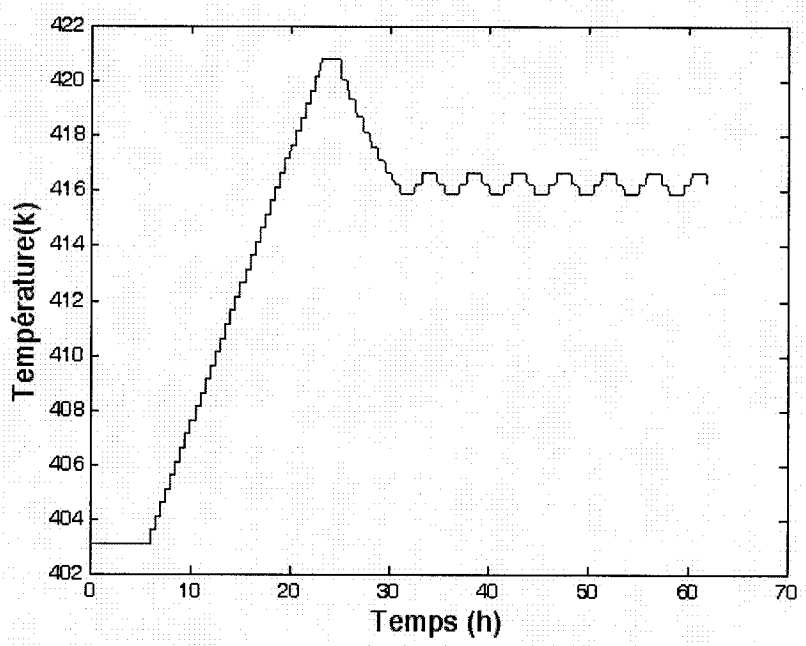

Figure E- 3 : Température pour l'ordre DT

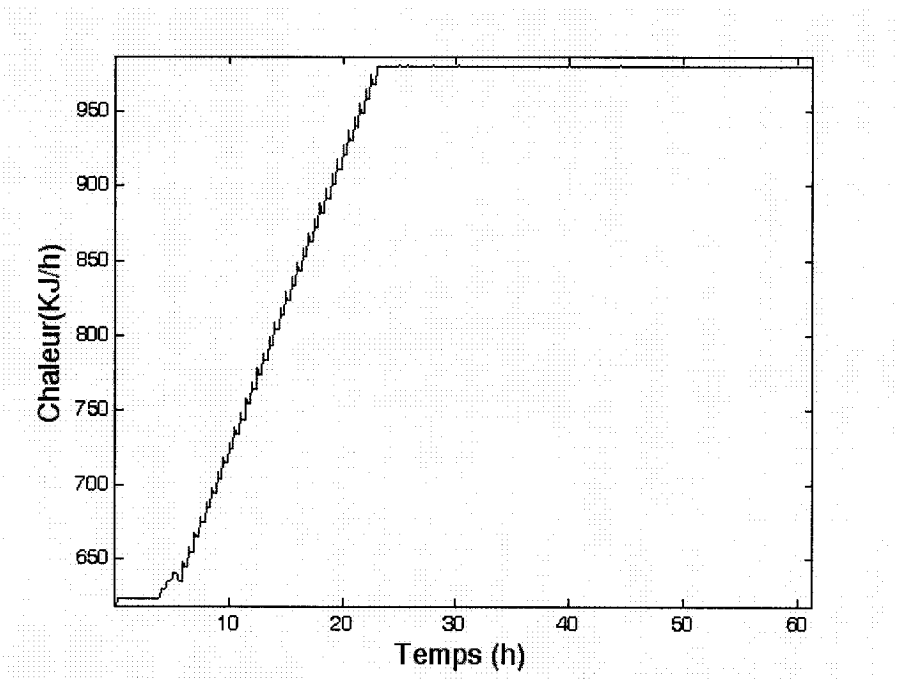

Figure E- 4 : Contrainte pour l'ordre DT

Ordre : Température suivi du taux de dilution

L'optimisation avec cet ordre des variables manipulées, donne des résultats très semblables à la précédente. L'optimum détecté est le même pour les deux problèmes d'optimisation. La seule différence c'est le parcourt avant la projection sur la contrainte active.

Discussion sur l'application de la méthode d'optimisation en temps réel avec projection sur contraintes actives.

Dans les deux problèmes d'optimisation, la première variable manipulée apporte le système à l'optimum mono variable. Ensuite la seconde variable est optimisée jusqu'à ce que la contrainte soit rencontrée. Alors la contrainte est considérée active. L'optimisation avec le taux de dilution apporte le système au point optimal, pendant que le contrôleur entre la température et la contrainte s'assure que la contrainte est respectée. La fonction objectif est de 0.2525 mol/h. Bref, la méthode d'optimisation en temps réel avec projection sur contraintes actives fonctionne bien dans cet exemple.

La méthode des perturbations et la fonction barrière.

Comme la méthode des perturbations ne considère pas les contraintes, la fonction barrière a été ajoutée à la fonction objectif de la méthode des perturbations.

Discussion sur l'application de la méthode des perturbations.

Les deux variables manipulées sont modifiées en même temps, afin d'optimiser le procédé. Ceci est représenté dans les figures E-6 et E-7 qui illustre les variables manipulées. La fonction barrière permet au système de bien de respecter la contrainte, voir la figure E-8. Finalement, la méthode des perturbations atteint une valeur maximale moyenne de 0.2495 mol/h. Le trajet de la fonction objectif est représenté à la figure E-5.

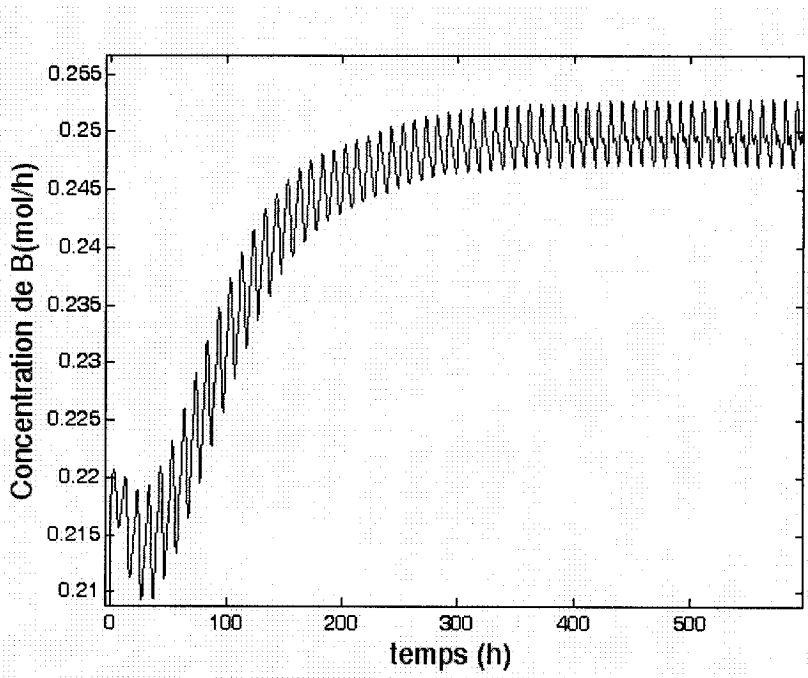

Figure E- 5 : Fonction objectif pour la méthode des perturbations

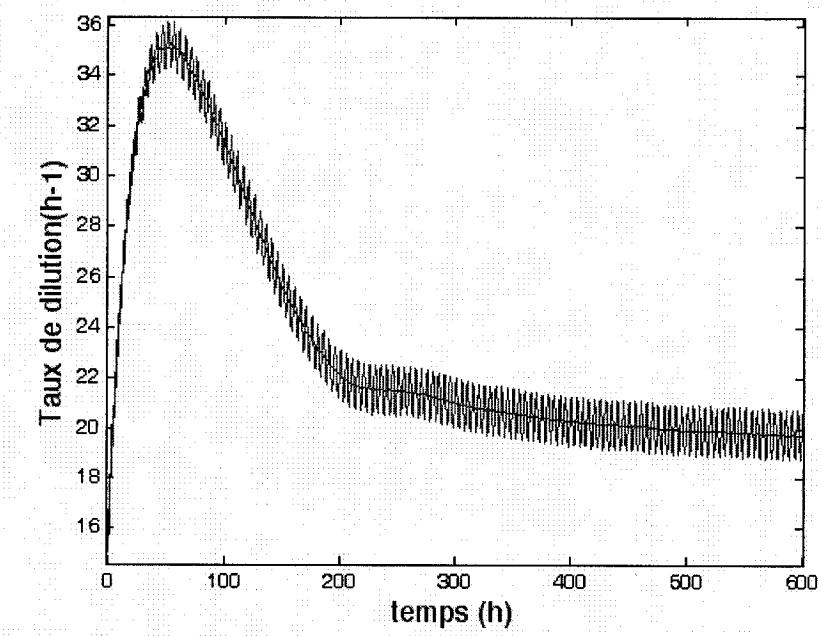

Figure E- 6 : Taux de dilution pour la méthode des perturbations

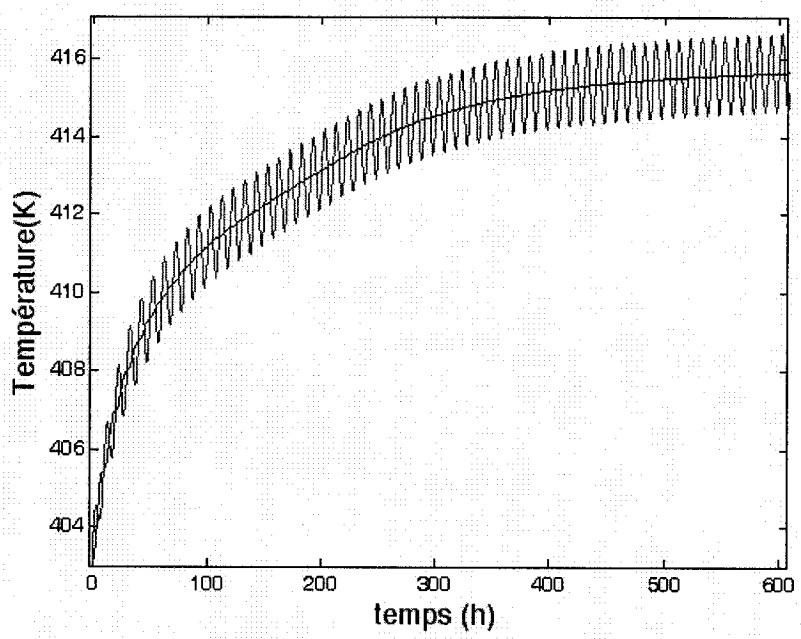

Figure E- 7 : Température pour la méthode des perturbations

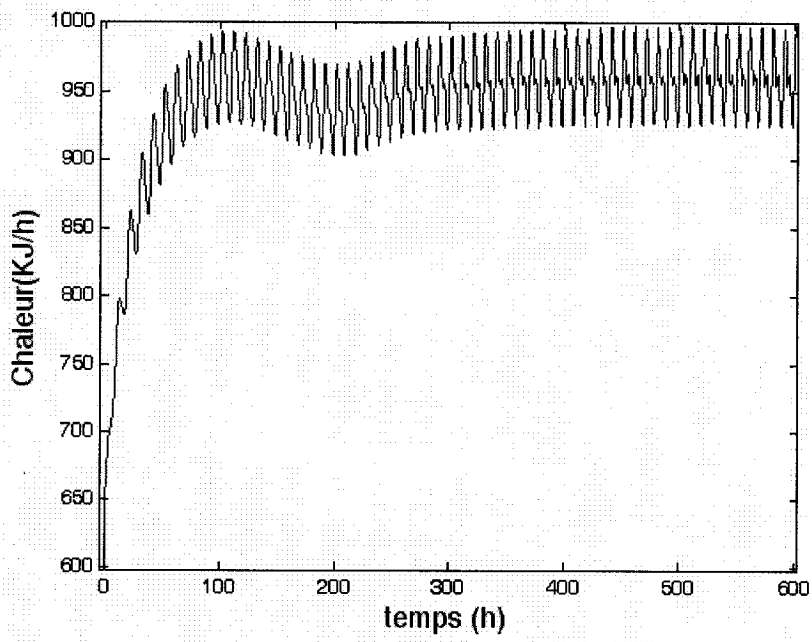

Figure E- 8 : Contrainte pour la méthode des perturbations

Comparaison entre les différentes méthodes d'optimisation.

Les deux méthodes optimisation trouvent le même point optimal. La méthode OTRPCA est plus près de la contrainte ce qui donne une fonction objectif légèrement supérieure. Cette distance est causée par l'amplitude de l'excitation externe dans la méthode des perturbations.

Le temps d'optimisation est très différent entre les deux méthodes d'optimisation. La méthode d'optimisation en temps réel avec projection sur contraintes actives atteint le l'optimum en 29 heures, tandis que la méthode des perturbations se rend à l'optimum en 600 heures. Donc, la méthode des perturbations prend 20 fois plus de temps, en plus l'agressivité de la méthode OTRPCA pourrait être augmentée, car le temps que la méthode laisse pour que le régime permanent soit atteint est grand.

L'inconvénient d'utiliser une variable manipulée à la fois dans la méthode OTRPCA est représenté sur la figure E-3. Lors de l'optimisation de la température le point optimal est dépassé. Une fois la contrainte atteinte la température revient à sa valeur pour avoir la concentration de B optimale. Le détour effectué lors de l'optimisation de la température représente une perte de temps dans l'optimisation. Ici

cette perte est faible, mais dans d'autres problèmes cette valeur pourrait être significative.

Un bruit a été appliqué à la concentration de Cb lors de la solution des problèmes d'optimisation (Figure E-9 et E-10). Pour la méthode avec projection le gain de l'algorithme et le delta ont du être ajustés, tandis que dans la méthode des perturbations le gain de la fonction barrière a du être modifié. La projection sur la contrainte active atteint une meilleure fonction objective, ce qui peut être expliqué par le fait que le contrôleur entre la température et la chaleur soutirée permet d'être plus près de la contrainte en corrigeant les perturbations rapidement. Lors de l'application de la méthode avec projection, la section avec le relâchement de la contrainte n'était pas active, ce qui pourrait entraîner des problèmes d'utilisation de cette méthode avec du bruit.

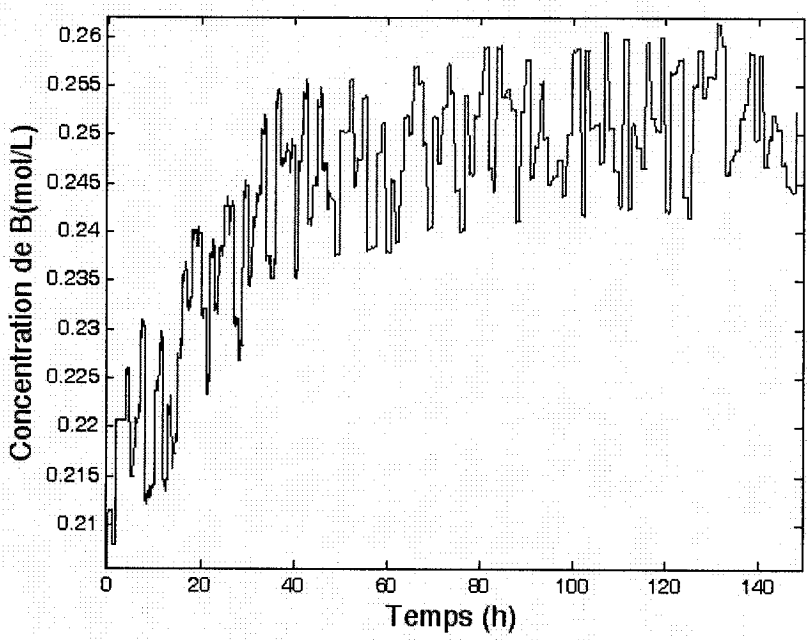

Figure E- 9 : Méthode de projection avec du bruit

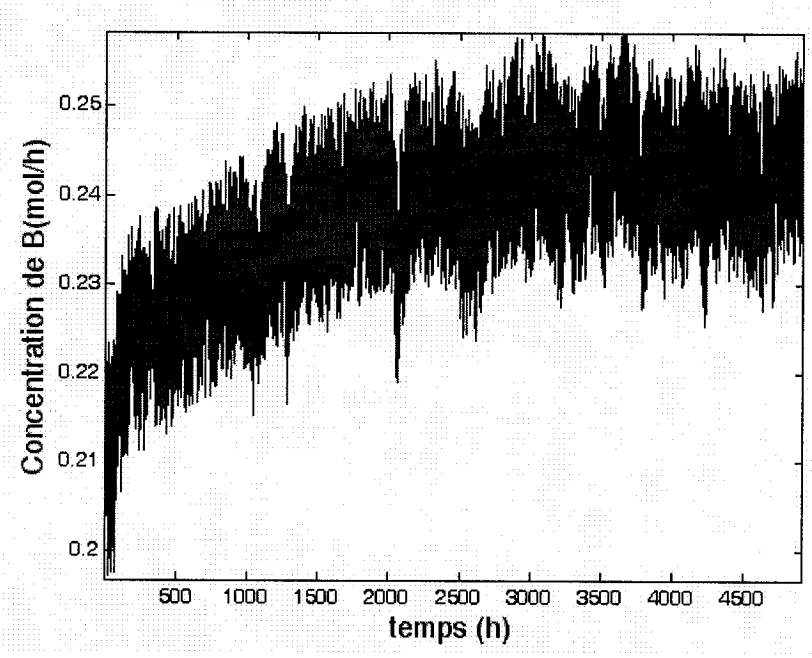

Figure E- 10 : Méthode de perturbation avec bruit