



Titre: Robustesse de la commande prédictive non-linéaire par rapport aux
Title: erreurs paramétriques

Auteur: Jean-Christophe Binette
Author:

Date: 2016

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Binette, J.-C. (2016). Robustesse de la commande prédictive non-linéaire par
Citation: rapport aux erreurs paramétriques [Master's thesis, École Polytechnique de
Montréal]. PolyPublie. <https://publications.polymtl.ca/2273/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/2273/>
PolyPublie URL:

**Directeurs de
recherche:** Balasubrahmanyam Srinivasan
Advisors:

Programme: Génie chimique
Program:

UNIVERSITÉ DE MONTRÉAL

ROBUSTESSE DE LA COMMANDE PRÉDICTIVE NON-LINÉAIRE PAR RAPPORT AUX
ERREURS PARAMÉTRIQUES

JEAN-CHRISTOPHE BINETTE

DÉPARTEMENT DE GÉNIE CHIMIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

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

AOÛT 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

ROBUSTESSE DE LA COMMANDE PRÉDICTIVE NON-LINÉAIRE PAR RAPPORT AUX
ERREURS PARAMÉTRIQUES

présenté par : BINETTE Jean-Christophe

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. PERRIER Michel, Ph. D., président

M. SRINIVASAN Bala, Ph. D., membre et directeur de recherche

Mme WOODWARD Lyne, Ph. D., membre

REMERCIEMENTS

Je tiens à remercier tous ceux qui, que ce soit directement ou indirectement, ont contribué à la réalisation de ce projet.

D'abord, mes remerciements vont au professeur Bala Srinivasan, mon directeur de maîtrise, qui m'a permis de réaliser ce projet. Sa patience et son encadrement me furent indispensables.

Merci aussi au professeur Michel Perrier qui, lorsque je l'ai approché, m'a encouragé à me lancer dans ce projet de maîtrise.

Un grand merci bien sûr au département de Génie Chimique au sein duquel j'ai réalisé ce projet.

J'aimerais aussi remercier le CRSNG pour le financement accordé à ce projet.

Enfin, un grand merci à ma famille, ma copine et mes amis, qui m'ont supporté et encouragé tout au long de ce projet.

RÉSUMÉ

La commande de procédé comporte diverse méthodes. Pour des problèmes complexes, certaines procédant par optimisation peuvent être appropriées, notamment la commande prédictive non-linéaire (NMPC pour *nonlinear model predictive control*). Ces méthodes de commande doivent performer, même en présence d'incertitudes paramétriques. Des méthodologies existent pour corriger les erreurs paramétriques (adaptation) ou réduire leur impact sur l'efficacité de la commande (méthodes robustes). Cependant, ces solutions ne sont pas toujours applicables, exigent des calculs supplémentaires et, surtout, sont plus conservatrices. Ce projet visait donc à déterminer dans quelles situations la méthode de base comporte une robustesse intrinsèque.

En effet, des articles publiés au cours des dernières années ont mis en évidence certains problèmes avec la NMPC appliquée à des procédés batch. Dans certaines simulations, l'optimisation hors ligne performait mieux que la NMPC, qui utilise pourtant des informations supplémentaires. Ce travail continue dans cette voie et plusieurs systèmes ont été simulés avec différentes combinaisons de coûts (économique ou de trajectoire), de contraintes (aucune, de trajectoire ou terminale) et de variations paramétriques. Pour ces simulations, les coûts de trajectoire quadratiques apportaient une robustesse accrue, la NMPC performant mieux avec ceux-ci que les optimisations hors ligne.

L'étape suivante fut d'analyser la NMPC d'un point de vue mathématique pour vérifier si ces résultats s'expliquaient. La première étape a démontré que, en présence d'erreurs paramétriques, avec ou sans bruit de procédé, la NMPC peut performer mieux ou moins bien que l'optimisation hors ligne, les deux situations étant permises par la méthode. La seconde étape de l'analyse a mené au résultat le plus important. En effet, une démonstration a indiqué que pour un coût de trajectoire quadratique, en présence d'erreurs paramétriques suffisamment petites, mais sans bruit de procédé, la NMPC donne de meilleurs résultats que l'optimisation hors ligne si les contraintes actives ne changent pas avec les variations paramétriques.

La dernière partie de ce travail fut de vérifier l'hypothèse de l'invariance des contraintes actives. Des optimisations ont été faites, comparant trois solutions optimales localement, composées de séquences d'arcs, avec différentes valeurs de paramètres et de contraintes. Cette analyse de la sensibilité a montré que, dans ce cas, la forme de la meilleure solution ne changeait pas avec les changements de paramètres, i.e. les contraintes actives restaient invariantes.

ABSTRACT

Process control has access to various strategies. When faced with complex problems, some of those using optimization, such as NMPC (nonlinear model predictive control), can be appropriate. Those methods must perform well, even with parametric errors. Strategies exist either to correct parametric errors (adaptation) or to reduce their impact on the controller's performance (robust strategies). However, these solutions are not always possible to implement, they require additional computational time and, most of all, are conservative. This project's main objective was therefore to find out whether there are cases where the base method has a natural robustness.

Papers published in the last few years showed some issues for the NMPC applied to a batch processes: offline optimization outperformed NMPC, which uses additional information. This work continues on this topic and various simulations were done for different combinations of cost (tracking or economic), constraints (none, trajectory or terminal) and parametric variations. The results obtained tended to show better robustness with trajectory costs, as NMPC always performed better in those cases than the offline optimization.

The next step was to analyse the NMPC from a mathematical point of view, in order to see if such results could be explained. The first part of the analysis showed that, with parametric errors and with or without process noise, NMPC can give better results than offline optimization as much as worse results. The second part gave the most important result: it proves that, for small enough parametric variations, with no process noise, NMPC used with a quadratic tracking cost will always give a better performance than the offline optimization if the active constraints are invariant with respect to parametric changes.

The last part of this work consisted in verifying the invariant constraints assumption. Optimizations were made on another system, comparing locally optimal solutions composed of sequences of arcs, with varied parameters and constraints. This sensitivity analysis showed that, for that case, the general shape of the best solution didn't change with parametric changes, i.e. the active constraints were invariant.

TABLE DES MATIÈRES

REMERCIEMENTS	III
RÉSUMÉ.....	IV
ABSTRACT	V
TABLE DES MATIÈRES	VI
LISTE DES TABLEAUX.....	VIII
LISTE DES FIGURES.....	IX
LISTE DES SIGLES ET ABRÉVIATIONS	X
LISTE DES ANNEXES.....	XI
CHAPITRE 1 INTRODUCTION.....	1
1.1 Commande de procédé.....	1
1.2 Incertitudes paramétriques et robustesse.....	2
1.3 Motivations de l'étude.....	3
1.4 Objectif général	4
1.5 Plan du mémoire.....	4
CHAPITRE 2 REVUE CRITIQUE DE LA LITTÉRATURE	5
2.1 Procédés batch et continus	5
2.2 Optimisation de procédés batch	6
2.2.1 Coût intégral et coût final.....	7
2.2.2 Incertitudes paramétriques	9
2.2.3 Optimisation robuste	10
2.3 Commande prédictive	11
2.3.1 Premiers développements.....	11
2.3.2 Horizon fuyant et horizon rétrécissant	12

2.3.3	Commande prédictive linéaire	13
2.3.4	Commande prédictive non-linéaire	13
2.3.5	Adaptation paramétrique	16
2.3.6	Commande prédictive non-linéaire robuste	17
2.3.7	Applications	18
2.4	Synthèse de la revue de la littérature	19
CHAPITRE 3 SYNTHÈSE DES ARTICLES		21
3.1	Premier article	21
3.1.1	Analyse de la variation	21
3.1.2	Exemple	23
3.2	Deuxième article	24
CHAPITRE 4 RÉSULTATS COMPLÉMENTAIRES ET DISCUSSION GÉNÉRALE.....		26
4.1	Résultats complémentaires à l'article 1	26
4.1.1	Simulations supplémentaires	26
4.2	Résultats complémentaires à l'article 2	32
4.3	Discussion générale	32
4.3.1	Hypothèses effectuées	32
4.3.2	Principaux résultats	34
CHAPITRE 5 CONCLUSION ET RECOMMANDATIONS		36
5.1	Conclusion	36
5.2	Recommandations	36
BIBLIOGRAPHIE		38
ANNEXES		42

LISTE DES TABLEAUX

Tableau 4.1: Comparaison des solutions hors ligne, avec ré-optimisation et optimales pour l'usine pour des coûts avec une trajectoire non réalisable	27
Tableau 4.2: Paramètres du modèle, limites d'opération et conditions initiales pour le réacteur d'isomérisation batch	28
Tableau 4.3: Comparaison des solutions hors ligne, avec ré-optimisation et optimales pour l'usine pour le réacteur d'isomérisation	29
Tableau 4.4: Paramètres du modèle, limites d'opération et conditions initiales pour le réacteur semi-batch isotherme avec contrainte de sécurité	31
Tableau 4.5: Comparaison des solutions hors ligne, avec ré-optimisation et optimales pour l'usine pour le réacteur semi-batch isotherme avec contrainte de sécurité	31
Tableau 4.6: Production de C pour les 3 solutions envisagées, considérant des changements de contrainte, de condition initiales ou de paramètres	32

LISTE DES FIGURES

Figure 4.1: a) Résultats pour un coût économique : Concentration finale en A en fonction de l'erreur sur α et sur k_{10} . b) Résultats pour un coût de trajectoire : Somme des carrés des écarts à la trajectoire en fonction de l'erreur sur α et sur k_{10}	29
--	----

LISTE DES SIGLES ET ABRÉVIATIONS

MPC	Model predictive control (Commande prédictive)
NMPC	Nonlinear model predictive control (Commande prédictive non-linéaire)
RTO	Real time optimization (Optimisation en temps réel)
d-RTO	dynamic-RTO (RTO dynamique)

LISTE DES ANNEXES

Annexe A – Article 1 : On the Use of Nonlinear Model Predictive Control without Parameter Adaptation for Batch Processes.....	42
Annexe B – Article 2 : On the various local solutions to a two-input dynamic optimization problem.....	53
Annexe C – Codes matlab écrits pour réaliser les différentes simulations	63

CHAPITRE 1 INTRODUCTION

Les travaux accomplis dans le cadre de ce mémoire s'inscrivent dans la recherche en commande et en optimisation de procédés. Les objectifs de ces recherches sont principalement d'assurer la sécurité, d'améliorer l'efficacité des méthodes de contrôle et, incidemment, l'efficacité des procédés. Dans le but de mieux comprendre le fonctionnement de certaines méthodes de commande procédant par optimisation, des simulations numériques et des analyses mathématiques ont été réalisées.

1.1 Commande de procédé

La commande de procédé est une discipline visant à amener et maintenir un procédé à un état désiré. Dans ce cadre, les variables d'état sont celles décrivant la situation dans laquelle se trouve un procédé. Parmi celles-ci, en génie chimique, on retrouve notamment les concentrations et les températures. Les variables d'entrée, ou variables manipulées, sont celles pouvant être ajustées par un actuateur. Par exemple, un débit d'alimentation à une unité peut être manipulé en ouvrant ou fermant une vanne. Enfin, les variables de sortie sont des fonctions des variables d'état, représentant des quantités pour lesquelles des valeurs précises peuvent être souhaitées pour des raisons économiques, de sécurité ou environnementales. Par ailleurs, une variable de sortie pour une opération peut agir comme variable d'entrée d'une autre opération. Ainsi, pour le contrôle d'une vanne, le débit dans la conduite correspondante est la variable de sortie et l'ouverture, la variable d'entrée. Ce même débit peut par la suite devenir une variable d'entrée pour le contrôle d'un réacteur.

Les stratégies de contrôle agissent sur les variables d'entrée en sélectionnant les valeurs qu'elles doivent prendre dans le but d'influencer l'évolution des variables d'état de manière à ce que la ou les variables de sortie atteignent les valeurs souhaitées. Cependant, les variables d'entrée, d'état et de sortie ne peuvent généralement pas prendre n'importe quelle valeur. Diverses contraintes imposent des limites à celles-ci. Ces contraintes peuvent provenir de différentes sources.

- Limitations techniques : il peut s'agir simplement de bornes sur un débit d'alimentation. Celui-ci ne peut être négatif et sa valeur maximale dépend des capacités des pompes, tuyaux et vannes impliqués. Il s'agit alors d'une contrainte sur une variable d'entrée.

- Normes environnementales : notamment, des normes sur la qualité de l'air imposent des limites sur l'impact qu'une usine peut avoir sur la concentration de polluant dans l'air ambiant. Ainsi, si un réacteur émet des rejets gazeux, une contrainte sur une variable de sortie dépendant de la température et de la concentration de polluant sera présente.
- Normes de sécurité : pour éviter des accidents, il est normal d'imposer certaines limites aux conditions d'opération de divers équipements. Par exemple, pour éviter qu'un réacteur atteigne une température trop élevée en cas de dysfonctionnement, une contrainte sur la concentration de réactif maximale peut être imposée.

Les contrôleurs classiques sont les PID (proportionnel, intégral et différentiel). Ceux-ci appliquent une correction à la variable manipulée en fonction de l'erreur, c'est-à-dire la différence entre la variable de sortie mesurée et la valeur désirée. La correction apportée par ces contrôleurs peut être proportionnelle à l'erreur (P), à l'intégrale de l'erreur (I) ou à la dérivée de l'erreur (D). Ces contrôleurs et leurs variantes (P, PI) sont les plus courants en raison de leur simplicité et de leur facilité d'implantation.

Cependant, cette simplicité implique certaines limites et, à l'occasion, des méthodes de contrôles plus complexes deviennent préférables. Dans ces situations, l'optimisation peut jouer un rôle majeur dans le contrôle de procédé. En mathématiques, l'optimisation consiste à trouver les valeurs maximales ou minimales d'une fonction objectif en déterminant les valeurs des variables indépendantes correspondantes, tout en tenant compte des diverses contraintes applicables. Comme les méthodes de contrôle visent à amener un système à des états désirés, il est naturel de vouloir déterminer quels sont les états idéaux et les meilleurs chemins pour s'y rendre. L'optimisation est donc un outil de choix, tant pour déterminer les états souhaités que les entrées qui y mèneront. La commande prédictive non-linéaire (NMPC) en est un bon exemple : celle-ci procède par optimisation successive alors que le système évolue dans le temps, utilisant les mesures faites en ligne pour adapter les conditions initiales à chaque fois.

1.2 Incertitudes paramétriques et robustesse

Les méthodes de contrôle doivent assurer le bon fonctionnement des procédés. Par exemple, la capacité de stabiliser un procédé est une caractéristique importante. Cela signifie que, si le système est dans le voisinage d'un point d'équilibre, alors il y restera. Le système est

asymptotiquement stable si, une fois à l'intérieur de ce voisinage, il finit par tendre vers le point d'équilibre et attractif s'il ne peut que se rapprocher du point d'équilibre. L'optimalité d'une méthode de contrôle est aussi très importante : en effet, il est souhaitable que le système évolue de la meilleure manière possible, afin d'obtenir les meilleures performances.

Une autre caractéristique majeure en contrôle est la robustesse. Plusieurs facteurs peuvent faire dévier l'évolution d'un procédé de ce qui est attendu. D'abord, des perturbations peuvent venir modifier les conditions d'opération. Ensuite, le bruit sur les mesures et sur le procédé lui-même peuvent poser problème. Les valeurs souhaitées pour les variables de sortie peuvent changer, entraînant naturellement un changement des conditions d'opération. Enfin, les modèles utilisés peuvent contenir des erreurs paramétriques, donnant des règles de contrôle imparfaitement adaptées au procédé réel. Une méthode de contrôle qui conserve les propriétés justifiant son utilisation (optimalité, stabilité, réduction de l'erreur, etc.) est dite robuste. Ce projet s'intéresse particulièrement à la robustesse du contrôle par optimisation.

1.3 Motivations de l'étude

Les problèmes liés aux incertitudes paramétriques sont bien connus. Toute méthode de contrôle doit demeurer utile en leur présence et conserver ses propriétés. L'utilisation de la NMPC est évidemment soumise aux mêmes nécessités. Kamaraju, Chiu et Srinivasan (2013) présente notamment un exemple comparant, entre autres choses, la performance d'une optimisation hors ligne et la NMPC. Or, dans certains cas, l'optimisation hors ligne avait mieux performé que la NMPC. Ce résultat, obtenu sur un procédé batch avec un coût économique, a entraîné le questionnement quant à la différence de robustesse potentiellement importante de la NMPC entre un coût économique et un coût de trajectoire. Quand la ré-optimisation de la NMPC est-elle bénéfique?

Ce problème n'est pas nouveau et diverses stratégies pour y pallier sont suggérées. L'adaptation des paramètres à l'aide des mesures en ligne, comme dans l'optimisation en temps réel dynamique (*dynamic real time optimization* ou dRTO) et les diverses versions robustes de la NMPC et de l'optimisation en général en sont des exemples. Cependant, les méthodes robustes peuvent augmenter le temps de calcul et être très conservatrices, i.e. sous-optimales, alors que l'implantation de l'adaptation peut être complexe, voire impossible, et nécessiter l'utilisation d'entrées sous-optimales.

Ce projet visait donc plutôt à déterminer dans quelles situations, pour la NMPC, un effort supplémentaire pour atteindre la robustesse est nécessaire et dans quelles situations la méthode de base présente une robustesse naturelle. Autrement dit, la sensibilité aux erreurs paramétriques de certaines stratégies de commande par optimisation est analysée.

1.4 Objectif général

Ce travail, basé sur des simulations numériques de procédés et des analyses mathématiques sur les méthodes de contrôle, a pour objectif d'évaluer la robustesse de la commande prédictive non-linéaire pour des coûts économiques ou de trajectoire, en présence de contraintes et d'erreurs sur les paramètres du modèle. La robustesse en soi est étudiée dans le premier article, tandis que le second article vise à valider une hypothèse utilisée dans le cadre de cette étude de robustesse.

L'hypothèse principale, étudiée dans le premier article, est que les coûts de trajectoire quadratiques apportent une robustesse supplémentaire à la commande prédictive non-linéaire, robustesse qui est absente pour les coûts économiques.

Une seconde hypothèse, présumée valide dans le premier article, est étudiée dans le deuxième. Selon celle-ci, les contraintes actives sont invariantes avec les variations paramétriques.

1.5 Plan du mémoire

Ce mémoire est rédigé sous la forme « par articles ». Deux articles rédigés dans le cadre de ce projet de maîtrise, soumis pour publication, y sont donc joints en annexe. En ce qui concerne le corps de ce mémoire, il est divisé en cinq parties, incluant cette introduction.

Le Chapitre 2 constitue la revue de la littérature autour des sujets centraux de ce projet : les procédés batch, leur optimisation et la commande prédictive non-linéaire. Le Chapitre 3 constitue une synthèse des articles produits dans le cadre de cette recherche. Les principaux résultats de chacun y sont présentés et la méthodologie y est décrite. Ensuite, le Chapitre 4 présente quelques résultats complémentaires pertinents ainsi qu'une brève discussion générale. Enfin, le Chapitre 5 conclut le tout en rappelant les principaux résultats et en faisant des recommandations pour de prochains projets.

CHAPITRE 2 REVUE CRITIQUE DE LA LITTÉRATURE

2.1 Procédés batch et continus

Les procédés de génie chimique peuvent être séparés en deux grandes catégories principales : continus et batch.

Les premiers, une fois démarrés, opèrent sans arrêt, donnant continuellement des produits et demandant une alimentation ininterrompue. Il n'y a pas d'interruption du procédé pour récupérer le produit, celui-ci sortant de façon automatisée et continue. Les procédés continus ne s'arrêtent que pour l'entretien des différentes unités, les urgences ou la mise hors-service d'une partie ou de la totalité de l'usine. En raison de leur très grande capacité de production, ceux-ci servent plus couramment pour les productions à grande échelle, comme on retrouve notamment dans le domaine pétrolier. (Bonvin, Srinivasan, & Hunkeler, 2006; Green & Perry, 2008)

Les procédés batch, quant à eux, consistent en une opération d'une durée limitée, souvent prédéfinie, au terme de laquelle le processus est complété et le produit récolté. L'opération doit être recommencée du début si une plus grande quantité de produit est désirée. On distingue les procédés semi-batch et batch, selon qu'il y a une alimentation au procédé ou non. Ces procédés sont utilisés dans des productions à plus petite échelle, notamment dans des industries spécialisées comme le domaine pharmaceutique. Ils permettent entre autres de contrôler les lots, ce qui peut être nécessaire pour des raisons de qualité et de sécurité. Ceci constitue un attrait important pour plusieurs industries pour lesquelles la qualité du produit prime sur la quantité. Par ailleurs, ils permettent une bonne flexibilité en opération. (Bonvin & al, 2006; Green & Perry, 2008; Mockus & Reklaitis, 1999; The Instrumentation, Systems, and Automation Society [ISA], 1995)

En ce qui concerne la commande, les deux types de procédés impliquent des approches et des objectifs différents. Pour les procédés continus, l'état est contrôlé afin de suivre une valeur cible ou une trajectoire, celle-ci étant déterminée de manière à atteindre un objectif économique à un niveau supérieur de l'usine ou pour contrôler l'entrée d'une opération subséquente. Ainsi, les changements de conditions d'opération pour un procédé continu sont généralement plutôt réduits. Les températures, débits d'alimentation et compositions, par exemple, peuvent varier, mais sur de longues périodes de temps et principalement en raison de changement désirés dans la production.

Pour les procédés batch, c'est l'état final qui importe le plus, bien que la sécurité et le respect des normes environnementales demeurent essentiels. Ceux-ci imposent ainsi des contraintes sur les conditions d'opération. Dans cette optique, afin d'atteindre les spécifications tout en respectant ces contraintes, les conditions d'opération doivent constamment être adaptées pour favoriser un état intéressant à la fin de la batch, plutôt que de stabiliser le système à un état en particulier qui ne change qu'occasionnellement. (Green & Perry, 2008)

Certains aspects du contrôle des procédés batch et continus seront développés plus particulièrement dans la discussion concernant la commande prédictive non-linéaire.

2.2 Optimisation de procédés batch

Dans le but d'améliorer la performance d'un procédé et d'en réduire les coûts, il est courant de bâtir un modèle phénoménologique du système. Celui-ci, en plus d'améliorer la compréhension d'un procédé, permet d'en calculer l'optimum en fonction du temps et des variables manipulées, souvent numériquement (Srinivasan, Palanki, & Bonvin, 2003). Un tel modèle dynamique peut être obtenu notamment par des équations de bilans de matières et d'énergie sur le système en régime transitoire.

Cependant, l'optimisation d'un procédé batch peut représenter un problème très complexe. En effet, plus il y a d'états et de variables manipulées, plus ceux-ci interagissent entre eux et avec les contraintes du système et plus le problème d'optimisation se complexifie. On retrouve abondamment de cas dans la littérature avec des problèmes de grandes dimensions (plusieurs variables d'entrée et d'état) résolus numériquement (Cervantes & Biegler, 1998; Kadam, Schlegel, Srinivasan, Bonvin, & Marquardt, 2007). Cependant, dans ces situations, l'interprétation du résultat obtenu n'est pas évidente. Dans d'autres cas, les problèmes sont de plus petites dimensions (une ou deux variables d'entrée, un à cinq états), ce qui rend plus aisée l'obtention d'une solution analytique et l'interprétation de celle-ci (Srinivasan & al, 2003).

De manière générale, le problème d'optimisation pour un procédé batch s'écrit comme suit :

$$\begin{aligned}
 \min_u J &= \phi(x(t_f), u(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t)) dt \\
 \text{s. t. } \dot{x} &= F(x(t), u(t)), \quad x(t_0) = x_0 \\
 S(x(t), u(t)) &\leq 0 \\
 T(x(t_f), u(t_f)) &\leq 0,
 \end{aligned} \tag{2.1}$$

où J est la fonction coût à minimiser, u les variables d'entrée, x les états, F les équations différentielles décrivant la dynamique du système, ϕ une fonction de coût terminal, L une fonction de coût intégral, t_0 et t_f représentant respectivement le temps de départ du procédé et le temps final. Enfin, S et T représentent respectivement les contraintes de chemin et les contraintes terminales.

En effet, les contraintes peuvent prendre deux formes distinctes. Une contrainte de chemin, ou de trajectoire, s'impose tout au long du procédé (Kadam & al, 2007, Pahija, Manenti, Mujtaba, & Rossi, 2014). Il peut s'agir de limites sur la variable manipulée (débit d'alimentation minimal et maximal par exemple) ou de contraintes sur des fonctions des états (contraintes de sécurité liées à la température, notamment). Les contraintes finales quant à elles ne s'imposent qu'à la fin du procédé ou de l'horizon de prédiction (voir section 2.3.2) (Bonvin & Srinivasan, 2013). La concentration d'un produit indésirable ou d'un intermédiaire, par exemple, peut être autorisée à atteindre des valeurs plus ou moins élevées pendant l'opération d'un procédé batch. Par contre, la valeur finale de cette concentration devra redescendre sous un certain seuil avant la fin de l'opération. Dans la section 2.3.4, il est aussi fait mention de l'utilisation d'une contrainte d'égalité terminale qui permet d'assurer la stabilité de la NMPC pour des procédés continus. Par ailleurs, qu'il s'agisse de contraintes de trajectoire ou terminales, celles-ci peuvent se présenter sous la forme d'inégalité ou d'égalité, comme dans le cas précédent.

2.2.1 Coût intégral et coût final

Le coût utilisé dans l'optimisation est généralement décrit par deux composantes : une intégrale et une finale. La première est liée au parcours pris par les états et les entrées du système tout au long du procédé, alors que la seconde correspond plutôt à une fonction des états finaux du système. La formulation du problème (2.1) présente bien ces deux composantes, avec ϕ pour la partie économique et L pour la partie intégrale :

$$J = \phi(x(t_f), u(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t)) dt. \quad (2.2)$$

Le coût intégral ou de trajectoire, plus souvent appliqué à un système en continu, peut s'exprimer notamment comme une différence entre une fonction de l'état du système tout au long du procédé et les valeurs souhaitées pour cette fonction.

Ce type de coût de trajectoire peut notamment s'exprimer sous la forme suivante ou son équivalent discrétisé :

$$\min_u J_{trajectoire} = \int_{t_0}^{t_f} L(x(t), u(t)) dt = \int_{t_0}^{t_f} \left((f(x(t)) - f_{des})^2 + (g(u(t)) - g_{des})^2 \right) dt. \quad (2.3)$$

Pahija et al (2014) présente notamment une simulation utilisant la NMPC couplée avec une optimisation dynamique en temps réel (dRTO ou *dynamic real time optimization*). Dans cette simulation, un coût de trajectoire est appliqué, composé de trois parties. La première concerne les états du système : il s'agit de la différence entre la trajectoire prédite par le modèle et la trajectoire désirée. La seconde partie fait entrer la différence entre la variable d'entrée et sa valeur souhaitée au régime permanent, forçant la variable d'entrée à ne pas trop s'éloigner de cette valeur. Enfin, la troisième partie impose un poids à la différence entre la variable d'entrée calculée à l'itération précédente et à l'itération active. Cette dernière partie permet notamment de réduire l'impact du bruit lié aux mesures en réduisant la variation de la variable d'entrée d'une itération à l'autre.

De manière générale, la trajectoire à suivre est obtenue à un niveau supérieur de commande, notamment par d'autres optimisations. De Souza, Odloak et Zanin (2010) explique bien dans son introduction comment, par exemple, une optimisation en temps réel (RTO ou *real-time optimization*) peut être utilisée à cet effet. La RTO optimise un problème statique représentant une usine. Les états optimaux sont ainsi déterminés pour l'ensemble du procédé. Ces informations sont ensuite transmises aux contrôleurs intermédiaires de chacune des unités, qui ont pour fonction de faire respecter les commandes calculées au niveau supérieur. De Souza & al (2010) présente par la suite une combinaison de la RTO et de la MPC au même niveau, plutôt que l'une au niveau supérieur et l'autre au niveau intermédiaire, puis une version simplifiée, qu'ils appliquent à la simulation d'une unité de craquage catalytique fluide.

Le coût final, plus utilisé pour les systèmes batch, peut s'exprimer comme une fonction de l'état et de l'entrée du système à la fin de la période de prédiction. En général, pour un système batch, cela correspond à la fin de l'opération. Le problème d'optimisation peut s'exprimer comme suit :

$$\min_u J = \phi(x(t_f), u(t_f)). \quad (2.4)$$

Srinivasan & al (2003) présente, entre autres choses, cinq procédés batch avec, pour chacun, un coût économique : le temps d'une batch, la concentration finale d'un produit ou la quantité finale d'un produit.

Enfin, le coût peut aussi comprendre les deux types de composantes. Par exemple, dans Mesbah, Streif, Findeisen et Braatz (2014), une simulation est faite pour un procédé de déposition d'un film mince. L'objectif comporte deux composantes : minimiser la rugosité à la fin du procédé tout en minimisant les variations autour de la valeur nominale. Maree & Imsland (2016) présente un algorithme impliquant les deux types de coût, dans ce cas appelés coût économique et coût de régulation. La fonction coût dans cet article implique un poids dynamique, assurant un coût économique optimal durant les phases transitoires et le suivi asymptotique des points de consigne. Cet article présente aussi l'exemple d'un réacteur CSTR auquel l'algorithme est appliqué.

L'ajout d'un poids terminal au coût du problème d'optimisation dans le but d'assurer la stabilité, comme expliqué à la section 2.3.4, constitue un autre exemple de situation où les deux types de composantes du coût sont présents.

Bref, le coût peut être formulé de sorte que ce soit la trajectoire qui importe, l'état final, ou une combinaison des deux.

2.2.2 Incertitudes paramétriques

Les solutions obtenues par des optimisations effectuées hors ligne peuvent être appliquées directement au procédé sans adaptations supplémentaires dans certains cas (Bonvin & al, 2006). Cependant, dans tous les procédés, des incertitudes quant à la précision des paramètres persistent toujours. Celles-ci peuvent provenir de différentes sources. Dans certaines situations, il peut être ardu, voire impossible, d'effectuer des mesures exactes permettant d'obtenir directement la valeur d'un paramètre. Dans certain cas, un paramètre considéré constant par le modèle choisi peut s'avérer être une approximation d'un paramètre ayant en réalité une certaine variabilité selon les conditions d'opération. Enfin, de simples erreurs paramétriques liées au changement d'échelle, de laboratoire à usine ou même d'usine pilote à usine, peuvent être difficiles à éliminer complètement. Ainsi, toute optimisation faite hors ligne peut souffrir de l'inexactitude du

modèle. Il est donc généralement plus approprié d'utiliser des mesures faites en ligne dans le cadre de l'optimisation pour palier à ces incertitudes à l'aide d'information nouvelle.

De nombreuses solutions existent pour corriger l'impact des incertitudes paramétriques, divisées en deux modes de pensée principaux. Le premier consiste à améliorer la robustesse de la méthode de contrôle, de façon à garantir la conservation de certaines caractéristiques. Certaines de ces méthodes sont présentées dans les sections Optimisation robuste (2.2.3) et Commande prédictive non-linéaire robuste (2.3.6). Le second mode de pensée consiste à utiliser en temps réel les mesures faites sur le procédé. Dans l'une des principales stratégies de ce groupe, les mesures servent à identifier les paramètres en ligne, de manière à adapter leur valeur dans la stratégie de contrôle alors même que le procédé est en fonction. Ce sujet est plus élaboré à la section 2.3.5.

2.2.3 Optimisation robuste

Il est possible, lorsqu'une optimisation est faite, de tenir compte des diverses incertitudes dans le processus, de manière à éviter qu'elles ne posent de réels problèmes. Ainsi, il faut trouver une solution qui demeure faisable et aussi optimale que possible. L'optimisation robuste est conçue dans cette optique. Le papier de Bertsimas, Brown et Caramanis (2011) concerne cette méthode d'optimisation. Le problème d'optimisation robuste y est formulé comme suit :

$$\begin{aligned} \min_u & \left(\max_{a_i} f_0(x, a_i) \right) \\ \text{s. t. } & f_i(x, a_i) \leq 0, \forall a_i \in A_i, \quad i = 1, \dots, m. \end{aligned} \quad (2.5)$$

Dans cette formulation, x représente un vecteur de variable de décision et a_i sont les incertitudes paramétriques. A_i représente l'ensemble des incertitudes. L'objectif du problème d'optimisation (2.5) est d'obtenir le coût minimal parmi toutes les solutions faisables pour toutes les réalisations possibles des incertitudes a_i dans A_i . Le \max à l'intérieur du problème indique que c'est la pire réalisation des incertitudes qui doit être minimisée. Notons que si A_i ne contient qu'une seule valeur, alors la contrainte correspondante ne contient aucune incertitude.

Les ensembles d'incertitudes peuvent se présenter sous diverses structures. La forme ellipsoïdale représente un certain intérêt à cause de la distribution normale (Bertsimas & Nemirovski, 1999; El Ghaoui & Lebret, 1997). Une forme polyédrale peut être vue comme un cas particulier de la forme ellipsoïdale. Bertsimas et al (2011) présente quelques autres structures pour l'incertitude,

avec dans chaque cas quelques résultats s'y appliquant, notamment concernant la taille des problèmes correspondants.

On retrouve quelques autres formes d'optimisation robuste : optimisation robuste quadratique, optimisation robuste semi-définie, programmation robuste géométrique, optimisation robuste discrète, ... Par ailleurs, lorsque l'évolution dans le temps et l'apport de nouvelles informations sont pris en compte plutôt que de considérer le problème statique (2.5), d'autres possibilités apparaissent.

L'optimisation robuste trouve des applications dans une grande variété de domaine, notamment la gestion de portfolio, la gestion de la chaîne d'approvisionnement et en ingénierie.

Le principal avantage de la formulation robuste (2.5) est qu'elle apporte une certaine assurance de faisabilité pour la solution obtenue. Le conservatisme de cette méthode peut être rebutant, mais certains cas présentent une flexibilité entre performance et robustesse. Par ailleurs, il n'est pas évident que le problème (2.5) puisse être résolu efficacement, ce qui peut constituer un autre inconvénient.

2.3 Commande prédictive

2.3.1 Premiers développements

L'origine de la commande prédictive remonte à la fin des années 1970, alors qu'on retrouve les premières publications concernant ce type de méthode de contrôle. La commande prédictive, ou MPC (de *model predictive control*) englobe plusieurs méthodes de contrôle consistant à minimiser une fonction objectif en utilisant un modèle du système à contrôler. (Camacho & Alba 1995; Henson, 1998; Morari & Lee, 1996)

Dans Morari & Lee (1999), les auteurs expliquent d'abord les premiers développements de la commande prédictive. Ils placent l'origine des idées menant à la MPC dans les années 1960. Il faudra toutefois attendre la publication de Richalet, Rault, Testud et Papon (1978) pour en observer les premiers développements. Ce dernier papier présente une méthode de contrôle utilisant la réponse à une impulsion pour représenter le système. Ce modèle approximatif est utilisé pour faire des prédictions en ligne, permettant d'ajuster les variables d'entrée conséquemment. Quelques autres développements suivirent, notamment la *dynamic matrix*

control (DMC) en 1979 (Cutler & Ramaker, 1979) et la *generalized predictive control* en 1987 (Clarke, Mohtadi, & Tuffs, 1987a, 1987b). L'étude théorique de ces deux premières méthodes de MPC mena progressivement au développement et à une compréhension de plus en plus poussées de la commande prédictive.

2.3.2 Horizon fuyant et horizon rétrécissant

Dans la commande prédictive, plusieurs temps sont utilisés. D'abord, il y a évidemment le pas de temps entre chaque optimisation. Ensuite, la période sur laquelle l'optimisation est effectuée dans la MPC, c'est-à-dire le temps jusqu'auquel la prédiction se rend, s'appelle horizon de prédiction. Enfin, la période de temps pendant laquelle la variable d'entrée peut être modifiée dans l'optimisation s'appelle horizon de commande. Cet horizon de commande peut prendre des valeurs entre un pas de temps et l'horizon de prédiction. Plus l'horizon de commande est grand par rapport au pas de temps, plus la taille du problème d'optimisation croît. Ainsi, un pas de temps trop petit ou un horizon trop grand peuvent rendre le problème trop long à résoudre.

Typiquement, deux types d'horizons de prédiction peuvent être utilisés, choisis selon le type de procédé à contrôler : un horizon fuyant ou un horizon rétrécissant.

L'horizon fuyant est utilisé pour les procédés continus. Au temps t_k , une fois l'optimisation complétée pour une prédiction jusqu'au temps t_p (l'horizon de prédiction), la commande obtenue est implémentée dans le procédé pour un certain temps t_c (l'horizon de commande) d'une durée équivalente à un pas de temps. Une fois cette période de temps écoulée, l'horizon de prédiction est repoussé d'un pas de temps, de sorte que la période de prédiction $t_p - t_k$ demeure la même à chaque nouvelle optimisation. Par exemple, Zavala et Biegler (2009) étudie une formulation de la NMPC avec un horizon fuyant sur un CSTR.

L'horizon rétrécissant quant à lui est utilisé pour les procédés batch. En effet, garder un temps de prédiction constant n'a pas de sens lorsque le temps d'opération restant diminue. Ici, l'horizon de prédiction t_p est donc fixe et correspond au temps final t_f . Ainsi, comme la commande obtenue à une étape est appliquée pour un pas de temps t_c , à chaque nouvelle optimisation, la durée de la prédiction diminue d'un temps t_c . L'horizon rétrécissant est utilisé par exemple dans Kamaraju et al (2013), pour le contrôle d'un réacteur d'isomérisation batch. De même, dans Mesbah et al

(2014), ce type d'horizon est utilisé dans une NMPC stochastique (décrite à la section 2.3.6) pour le contrôle des transformations polymorphiques dans un procédé de cristallisation batch.

Il faut noter que, dans les cas de MPC avec un horizon fuyant, la taille du problème d'optimisation reste constante, alors qu'elle diminue après chaque itération dans le cas d'un horizon rétrécissant.

2.3.3 Commande prédictive linéaire

Morari et Lee (1999) mentionne le modèle linéaire discret suivant pour la commande prédictive linéaire :

$$x(k+1) = Ax(k) + Bu(k), \quad x(0) = x_0, \quad (2.6)$$

où k représente le temps discrétisé.

À partir de ce modèle, la commande prédictive consiste à trouver les variables d'entrée donnant le minimum d'une fonction objectif, d'en implanter une partie pour un certain temps puis de refaire l'optimisation à partir des nouveaux états obtenus. Le problème d'optimisation peut, par exemple, prendre la forme suivante, pour laquelle des algorithmes de résolutions et des programmes commerciaux existent :

$$\begin{aligned} \min_u J = & x^T(p)P_0x(p) + \sum_{i=0}^{p-1} x^T(i)Qx(i) + \sum_{i=0}^{m-1} u^T(i)Ru(i) \\ \text{s. t. } & Ex + Fu \leq \psi, \end{aligned} \quad (2.7)$$

où p représente la longueur de l'horizon de prédiction et m la longueur de l'horizon de commande.

2.3.4 Commande prédictive non-linéaire

La NMPC est aussi parfois appelée NRHC, pour *nonlinear receding horizon control*, lorsque l'horizon est fuyant. Le terme NMPC sera utilisé dans ce texte.

La principale caractéristique de la NMPC est l'utilisation d'un modèle non-linéaire du système à contrôler, plutôt qu'un modèle linéaire comme on en trouvait dans les MPC développées auparavant.

Selon Kwon, Han et Ahn (2004), durant les années 1980, la NMPC a reçu peu d'attention en comparaison des méthodes linéaires. Cependant, depuis les années 1990, l'intérêt pour les méthodes non-linéaires a augmenté, principalement pour trois raisons. Premièrement, le milieu académique lui a prêté beaucoup d'attention. Deuxièmement, comme les modèles non-linéaires sont plus près de la réalité, ils sont plus appropriés pour représenter des opérations complexes. Troisièmement, la disponibilité d'ordinateurs ayant une puissance de calcul toujours plus grande a rendu la NMPC plus accessible et applicable en pratique. En effet, comme cette méthode requiert la résolution d'un problème d'optimisation non-linéaire, les calculs peuvent prendre un certain temps si les ordinateurs et les programmes utilisés ne sont pas assez performants.

La NMPC comporte certains intérêts (Green & Perry, 2008). D'abord, elle permet d'intégrer simplement les contraintes (sécurité, environnement, etc.) au problème d'optimisation, qu'elles s'appliquent aux états ou aux variables manipulées (Xi, Li, & Lin, 2013). De plus, elle tient naturellement compte des comportements particuliers d'un procédé non-linéaire. Par ailleurs, les changements de conditions d'opération, de contraintes ou autres peuvent facilement être intégrés en cours d'opération.

Par contre, la NMPC présentent aussi certains inconvénients (Green & Perry, 2008). D'abord, les calculs pour résoudre des problèmes d'optimisation non-linéaire peuvent être longs et complexes et, conséquemment, requièrent des ressources informatiques importantes. Ainsi, la NMPC ne peut être utilisée que pour des systèmes aux dynamiques suffisamment lentes pour les ordinateurs devant exécuter les calculs (Xi & al, 2013). En effet, il peut arriver pour certains systèmes que l'adaptation doive être modifiée rapidement, auquel cas de longs calculs peuvent être inopportuns. Ensuite, bien que pour des systèmes simples et connus, des modèles dynamiques relativement simples et petits peuvent être développés (Srinivasan & al, 2003), d'autres systèmes peuvent être complexes à modéliser et comporter beaucoup d'états et d'entrées (Cervantes & Biegler, 1998; Schlegel & Marquardt, 2006). De tels modèles peuvent être longs et complexes à développer. Enfin, d'un point de vue plus pratique, en raison de ses différences avec les méthodes de contrôle plus conventionnelles, le personnel d'usine peut avoir peu d'expérience au départ avec cette méthode de contrôle.

Comme son équivalent linéaire, la NMPC consiste à optimiser une fonction coût choisie selon les besoins du procédé à chaque nouveau pas de temps. Pour ce faire, des mesures sont effectuées sur le système. Celles-ci sont par la suite utilisées afin de déterminer les nouvelles conditions initiales à chaque itération.

Kamaraju et al (2013) présente entre autres bien la formulation de la NMPC. Au temps t_k , l'état est donné par x_k et le problème d'optimisation à résoudre peut s'écrire sous la forme générale suivante :

$$\begin{aligned} \min_u J &= \phi(x(t_p), u(t_p)) + \int_{t_k}^{t_p} L(x(t), u(t)) dt \\ \text{s. t. } \dot{x} &= F(x(t), u(t)), \quad x(t_k) = x_k \\ S(x(t), u(t)) &\leq 0 \\ T(x(t_p), u(t_p)) &\leq 0. \end{aligned} \tag{2.8}$$

Au temps t_k , le problème d'optimisation (2.8) est résolu numériquement. La commande optimale u ainsi obtenue est implémentée dans le procédé. Au bout d'un certain temps t_c , une nouvelle optimisation est effectuée sur (2.8), en utilisant les mesures faites sur le procédé pour déterminer les nouvelles conditions initiales. Ce processus est répété en boucle, sans arrêt pour les procédés continus ou jusqu'à la fin de l'opération pour les procédés batch. Ainsi, à chaque fois, seule la première partie de la variable d'entrée calculée est appliquée, le reste étant remplacé chaque fois par les nouvelles valeurs obtenues (Kwon & al, 2004). À noter que le problème (2.8) reprend exactement la formulation de (2.1). Ainsi, la commande prédictive non-linéaire se trouve être en fait une optimisation répétitive d'un procédé batch alors qu'il est en fonction.

En présence de bruit sur le procédé, mais sans erreur paramétrique, cette méthode atteint l'optimum du procédé dans la mesure permise par le pas de temps choisi. En effet, le bruit peut éloigner le procédé des valeurs prédites par le modèle, mais les optimisations répétitives de la NMPC assurent qu'un nouvel optimum sera trouvé à partir des valeurs modifiées.

De manière générale, il est important de pouvoir garantir la stabilité d'une méthode de contrôle. Pour la NMPC, lorsqu'appliquée à un procédé en continu, différentes façons d'assurer cette propriété ont été établies, présentant dans chaque cas une ou plusieurs conditions suffisantes pour atteindre cet objectif. La littérature concernant la stabilité de la commande prédictive est abondante, ce sujet se retrouvant au centre de plusieurs publications. Kwon et al (2004), entre

autres, présente un certain nombre de solutions permettant de garantir la stabilité de la NMPC. Celles-ci sont divisées en deux catégories, selon qu'elles imposent des conditions à la fin de l'horizon de prédiction ou non.

La première solution parmi celles imposant une condition terminale consiste à utiliser une certaine contrainte d'égalité à la fin de l'horizon de prédiction (Keerth & Gilbert, 1988). L'ajout d'un terme de pénalité terminal approprié dans le coût du problème d'optimisation peut aussi permettre d'atteindre l'objectif de stabilité (De Nicolao, Magni, & Scattolini, 1998). Une autre méthode, appelée *dual-mode* NMPC, consiste à imposer un ensemble de contraintes d'inégalité terminales (Chisci, Lombardi, & Mosca, 1996). Enfin, l'utilisation d'un critère min-max peut aussi suffire à garantir la stabilité (Fontes & Magni, 2003).

Une méthode utilisant conjointement une fonction de Lyapunov assignable (*control Lyapunov function* ou CLF) avec une stratégie de NMPC permet de considérer à la fois la stabilité et la performance, sans imposer de condition terminale au système (Primbs, Nevistic, & Doyle, 2000). Une autre méthode consiste à abandonner l'optimalité pour conserver certaines propriétés comme la stabilité. Une solution faisable est alors recherchée, plutôt qu'une solution optimale, en imposant une réduction du coût si possible.

Chacune de ces méthodes comportent des avantages et des inconvénients, tel que souligné dans. Kwon & al (2004). Les problèmes, selon le cas, peuvent être de l'ordre de la faisabilité ou de la sous-optimalité, par exemple. Dans le cas de l'utilisation d'une CLF, celle-ci doit être connue a priori, ce qui n'est pas toujours évident.

2.3.5 Adaptation paramétrique

Afin de réduire l'impact des incertitudes, l'utilisation en temps réel des mesures faites sur le procédé constitue une stratégie de choix. L'adaptation des paramètres consiste à prendre ces mesures pour identifier les paramètres en ligne, de manière à corriger leur valeur dans la stratégie de contrôle alors même que le procédé est en fonction. Certaines conditions sont toutefois essentielles pour rendre l'identification possible. Parmi celles-ci, on retrouve notamment la persistance de l'excitation (Sastry & Bodson, 1989; Yuan & Wang, 2011). Cette condition consiste à avoir suffisamment de variations adéquates sur les entrées et les sorties du système. En effet, si les changements effectués sur un système ne sont pas suffisants, les mesures obtenues

n'auront pas la variabilité nécessaire pour corriger des erreurs paramétriques. Par exemple, un système en régime permanent ou un système où les variations sont inférieures à la capacité de détection des instruments ne permettront l'adaptation d'aucun paramètre, puisqu'aucun changement dans les entrées et les sorties n'existera pour en permettre l'identification. D'autres types de variations ne permettront que l'identification d'un nombre limité de paramètres (Sasthy & Bodson, 1989). Ainsi, l'adaptation en ligne est possible sous certaines conditions, mais celles-ci peuvent nécessiter l'utilisation d'entrées sous-optimales, les entrées optimales ne remplissant typiquement pas la condition de persistance de l'excitation.

2.3.6 Commande prédictive non-linéaire robuste

La NMPC, telle que présentée jusqu'à maintenant, comporte des avantages, mais aussi des inconvénients, entre autres au niveau de la robustesse en présence d'incertitude sur les paramètres. Ainsi, il va de soi que de nombreuses études aient été faites afin de trouver des améliorations possibles à la NMPC qui, tout en conservant les caractéristiques la rendant intéressante, compenseraient ses défauts. Des méthodes dites robustes ont donc été développées pour contrecarrer la sensibilité aux erreurs paramétriques. On retrouve parmi celles-ci la NMPC min-max et la NMPC *multi-stage* et la NMPC stochastique.

La NMPC min-max consiste à résoudre le problème d'optimisation pour les pires scénarios d'incertitude des paramètres. Cependant, cette méthode est très conservatrice et clairement non-optimale (Bemporad & Morari, 1999; Lucia & al, 2013). Par ailleurs, Scokaert & Mayne, (1998) indique aussi que cette méthode peut mener à des infaisabilités. Ce-dernier article propose une amélioration à la méthode min-max en incluant une rétroaction, ce qui résout les problèmes de faisabilité et de conservatisme, mais augmente la taille du problème à résoudre au point de rendre la méthode inutilisable. Des solutions à ce dernier problème sont proposées dans la littérature, avec leurs avantages et inconvénients.

La NMPC multi-stage, présentée dans (Lucia, Finkler, & Engell, 2013), propose de prendre en compte explicitement que de nouvelles informations seront disponibles dans le futur. Pour ce faire, un arbre de scénarios discrets des possibles évolutions des incertitudes est utilisé. Cette méthode réduit le conservatisme par rapport à la méthode min-max. L'article étudie le cas d'un réacteur de polymérisation Chylla-Haase et arrive à de meilleurs résultats que la NMPC min-max tout en respectant les contraintes. Le principal défaut de la méthode est que la taille du problème

croît exponentiellement avec l'horizon de prédiction, le nombre d'incertitudes et les niveaux d'incertitudes pris en compte dans l'arbre de scénarios.

La NMPC stochastique (ou SMPC), proposée dans Mesbah et al (2014), s'applique quant à elle en présence d'un environnement avec des incertitudes stochastiques. Ainsi, le problème est considéré d'un point de vu probabiliste. L'intérêt de cette méthode provient des probabilités de réalisation extrêmement faibles de certains niveaux de perturbations considérés dans la NMPC min-max. En demandant la satisfaction des contraintes à un certain niveau de probabilité, plutôt qu'absolument, la SMPC permet de réduire le conservatisme par rapport au pire scénario possible et ainsi d'améliorer la performance. Mesbah et al (2014) propose une façon d'appliquer cette méthode et la teste sur un système de transformation polymorphique dans un procédé batch de cristallisation.

2.3.7 Applications

Les différentes méthodes de commande prédictive sont utilisées à travers le monde sur des milliers de procédés (Xi & al, 2013). Cependant, les applications de NMPC en particulier n'en représentent qu'une fraction. D'après Xi & al (2013), en 2003, la NMPC ne représentait que 2% du nombre d'application de son équivalent linéaire. Néanmoins, cette section présentera des cas d'applications de la MPC et de la NMPC.

Parmi les articles cités précédemment se trouve déjà plusieurs exemples de simulations faites pour des procédés utilisant la NMPC ou ses variations.

Dans Kamaraju & al (2013), des simulations sont faites pour un réacteur d'isomérisation batch avec un modèle assez simple. Lucia & al (2013) quant à eux font des tests pour le réacteur standard Chylla-Haase, un réacteur de polymérisation. Maree & Imsland (2016) présentent des tests numériques pour un réacteur CSTR d'hydrogénation de l'acétylène. Mesbah & al (2016) testent la commande prédictive stochastique et la NMPC sur un réacteur de cristallisation batch. Pahija & al (2014) effectuent des simulations pour un réacteur batch avec deux réactions parallèles. Enfin, dans Zavala & Biegler (2009), des simulations sont faites sur un réacteur CSTR.

De nombreuses autres applications de la NMPC se retrouvent dans la littérature. Dans Francisco, Vega et Skogestad (2015), la NMPC est appliquée à une plus grande échelle pour le contrôle du

procédé de traitement des boues activées d'une usine de traitement des eaux. Un modèle standard est utilisé pour faire les simulations, comportant deux réacteurs en anoxie et trois réacteurs aérés. Le système ainsi obtenu est de taille importante, comportant 13 états pour chaque bioréacteur. En raison des importants couplages des variables de ce procédé, la NMPC a été préférée aux traditionnels contrôleurs PI. La stabilité y est assurée par l'utilisation de contraintes terminales. La NMPC réussit à bien suivre les conditions de références en dépit d'importantes perturbations et des erreurs paramétriques. Revollar, Vega et Vilanova (2015) analyse l'efficacité de la NMPC sur la même unité, mais en utilisant cette fois un modèle simplifié en jumelant les réacteurs de même type, donnant un réacteur en anoxie et un réacteur aéré. Un coût économique est ajouté dans le cadre de cet article. Trois stratégies de contrôle sont comparées : la RTO+NMPC, la NMPC et le contrôle par PI. La performance de la RTO+NMPC s'est avérée meilleure pour certains aspects et moins bonne pour d'autres. En somme, celle-ci donnait un résultat intéressant lorsque la performance globale de l'usine était considérée. Toujours dans le domaine du traitement des eaux, Bello, Hamam et Djouani (2014) étudie une unité de coagulation des eaux usées. L'analyse de trois différents scénarios de 24 heures montre que la NMPC donne une performance satisfaisante, rejetant efficacement les perturbations et maintenant les variables de sortie constantes.

Gaida, Wolf, Bäck et Bongards (2012) étudient quant à eux l'application de la NMPC à une usine de biogaz, en se basant sur un modèle approuvé pour ce genre de procédé. Leur article démontre que la NMPC peut contrôler de façon optimale un procédé correspondant à ce modèle en utilisant les mesures sur le pH, la quantité de biogaz produit et les concentrations du biogaz en méthane et en CO₂.

Etchepareborda & Lolich (2007) appliquent la NMPC au contrôle de la puissance d'un réacteur nucléaire utilisé pour la recherche. Les simulations présentent d'excellentes performances, une bonne anticipation des perturbations et un bon rejet de celle-ci, évitant l'activation des systèmes de sûreté. De plus, des résultats préliminaires présentaient une faible sensibilité aux changements de paramètres.

2.4 Synthèse de la revue de la littérature

Bref, de nombreuses méthodes de contrôle par optimisation existent pour les procédés batch. Les incertitudes paramétriques peuvent influencer la qualité de leur performance. Pour cette raison,

plusieurs stratégies pour contrer l'effet des erreurs paramétriques sont développées, soit en augmentant la robustesse de la méthode ou en adaptant les paramètres. La suite de ce projet étudie la sensibilité aux incertitudes paramétriques de certaines méthodes de contrôle par optimisation, mettant en évidence des situations où une robustesse ou une adaptation supplémentaires seraient nécessaires et d'autres situations où la méthode de base possède une robustesse naturelle.

CHAPITRE 3 SYNTHÈSE DES ARTICLES

Dans le cadre de ce projet de maîtrise, deux publications ont été faites. La première est un article concernant spécifiquement la NMPC et constitue le cœur de ce projet de maîtrise. La seconde est une note de recherche faite en réponse à plusieurs commentaires reçus par les auteurs de Srinivasan et al (2003) au cours des années quant à l'optimisation d'un procédé batch. Cette note de recherche constitue notamment une analyse de sensibilité sur l'optimisation d'un procédé.

3.1 Premier article

Cet article, présenté à l'annexe A, comporte deux volets. Le premier, le plus important, consiste en une analyse de la variation de la formulation de la NMPC, qui aboutit à la présentation de deux propositions concernant la commande prédictive. Le second volet présente un exemple de la NMPC servant à représenter plus concrètement les différents aspects des deux propositions obtenues.

3.1.1 Analyse de la variation

D'abord, il faut rappeler la formulation de la NMPC pour un procédé batch avec un horizon de prédiction rétrécissant et un temps final t_f . Au temps t_k , les états x_k sont obtenus par les mesures et le problème d'optimisation est donné par

$$\begin{aligned}
 \min_{u[t_k, t_f]} J_k &= \phi(x(t_f)) + \int_{t_k}^{t_f} L(x(t), u(t)) dt \\
 \text{s. t. } \dot{x} &= F(x(t), u(t), \theta) + \dot{v}, & x(t_k) &= x_k \\
 S(x(t), u(t)) &\leq 0 \\
 T(x(t_f), u(t_f)) &\leq 0.
 \end{aligned} \tag{3.1}$$

où J est la fonction coût à minimiser, u les variables d'entrée, x les états, F les équations différentielles décrivant la dynamique du système, θ les paramètres du modèle, ϕ une fonction de coût terminal, L une fonction de coût intégral et S et T représentant respectivement les contraintes de chemin et les contraintes terminales. La variable v représente le bruit du procédé.

L'article étudie le cas où l'erreur paramétrique est non-nulle, alors que l'excitation pourrait être insuffisante pour une identification de paramètres.

En paramétrant de façon appropriée les états (X) et les entrées (U), puis en considérant les contraintes actives comme invariantes par rapport aux variations paramétriques, le problème (3.1) est réécrit sous la forme d'un problème statique non-linéaire :

$$\begin{aligned} \min_{U_k} J_k &= J(X_k, U_k, \theta) \\ X_k &= \Psi_k(U_k, \theta) + d_k, \end{aligned} \quad (3.2)$$

où d_k est la différence entre les mesures prédites et observées.

L'analyse de la variation qui suit est faite en supposant que les variations paramétriques sont suffisamment petites pour négliger les termes d'ordre supérieur. Ainsi, en présence de variations paramétriques et de perturbations sur le système, l'équation de variation ΔJ s'écrit :

$$\begin{aligned} \Delta J &= \frac{\partial J}{\partial U} \Delta U + \frac{\partial J}{\partial d} \Delta d + \frac{\partial J}{\partial \theta} \Delta \theta + \Delta U^T \frac{\partial^2 J}{\partial \theta \partial U} \Delta \theta + \Delta d^T \frac{\partial^2 J}{\partial \theta \partial d} \Delta \theta + \Delta U^T \frac{\partial^2 J}{\partial d \partial U} \Delta d + \frac{1}{2} \Delta U^T \frac{\partial^2 J}{\partial U^2} \Delta U \\ &\quad + \frac{1}{2} \Delta d^T \frac{\partial^2 J}{\partial d^2} \Delta d + \frac{1}{2} \Delta \theta^T \frac{\partial^2 J}{\partial \theta^2} \Delta \theta. \end{aligned} \quad (3.3)$$

En retirant de l'équation les termes ne pouvant être affectés par les manipulations et les termes nuls, le problème suivant est obtenu :

$$\min_{\Delta U} \widetilde{\Delta J} = \Delta U^T \frac{\partial^2 J}{\partial \theta \partial U} \Delta \theta + \Delta U^T \frac{\partial^2 J}{\partial d \partial U} \Delta d + \frac{1}{2} \Delta U^T \frac{\partial^2 J}{\partial U^2} \Delta U. \quad (3.4)$$

À partir de cette formulation, l'article élabore sur la formulation optimale de ΔU et sa formulation dans le cadre de la NMPC, considérant que celle-ci effectue une correction sans tenir compte des erreurs paramétriques. Le premier résultat ainsi obtenu montre que le coût du problème (3.4), lorsque le ΔU de la NMPC lui est appliqué, est donné par :

$$\widetilde{\Delta J}_{opt} = -\frac{1}{2} t_d^T t_d - t_d^T t_\theta, \quad (3.5)$$

où $t_d = \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} \left(\frac{\partial^2 J}{\partial d \partial U} \right) \Delta d$ et $t_\theta = \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} \left(\frac{\partial^2 J}{\partial \theta \partial U} \right) \Delta \theta$. Avec l'hypothèse standard selon lequel le Hessien est défini positif, la racine carrée existe.

L'article procède par la suite à deux propositions. La première requière quelques hypothèses. D'abord, les contraintes actives doivent être invariantes par rapport aux variations paramétriques. Ensuite, le problème d'optimisation dynamique (3.1) doit pouvoir être analysé en analysant le problème (3.2). Sur cette base, il est démontré que la ré-optimisation, telle que faite par la

NMPC, peut constituer une amélioration ou une détérioration par rapport à l'optimisation hors ligne. En effet, en présence d'erreurs paramétrique, avec ou sans bruit de procédé, les orientations de t_θ et t_d ne sont pas nécessairement les mêmes pour une fonction objectif générale, donc le signe du second terme de (3.5) reste indéterminé.

La seconde proposition utilise une hypothèse supplémentaire : les variations dans les états par rapport aux prédictions du modèle sont causées uniquement par les erreurs paramétriques. Alors, avec un coût de trajectoire quadratique de la forme $J = \frac{1}{2}(X_k - X_{ref})^T (X_k - X_{ref}) + \frac{1}{2}w(U_k - U_{ref})^T (U_k - U_{ref})$, il est démontré que, pour une trajectoire X_{ref} réalisable, la NMPC constitue une amélioration par rapport à l'optimisation hors ligne. En effet, la seconde proposition de l'article démontre que, pour des variations paramétrique suffisamment petites, $t_\theta = t_d$, donnant ainsi $\widetilde{\Delta J}_{opt} \leq 0$.

3.1.2 Exemple

L'article procède par la suite à un exemple visant à illustrer les résultats obtenus dans l'analyse. Pour ce faire, un réacteur batch à deux réactions, $A \rightarrow B$ and $A+B \rightarrow C$, inspiré de Kamaraju et al (2013), est étudié. La dynamique de ce système est donnée par

$$\begin{aligned}\dot{c}_A &= -k_1 c_A - k_2 c_A c_B, \\ \dot{c}_B &= k_1 c_A - k_2 c_A c_B,\end{aligned}\tag{3.6}$$

où c_X est la concentration en X et k_1 et k_2 les coefficients de réaction cinétique obtenus par l'équation d'Arrhenius :

$$k_i = k_{i0} \exp\left(-\frac{E_i}{RT}\right).\tag{3.7}$$

En utilisant la température modifiée suivante comme variable d'entrée

$$u = k_{10} \exp\left(-\frac{E_1}{RT}\right)\tag{3.8}$$

et considérant

$$\alpha = \frac{E_2}{E_1}, \quad \bar{k}_{10} = 1 \quad \text{et} \quad \bar{k}_{20} = k_{20} \left(\frac{1}{k_{10}}\right)^\alpha,\tag{3.9}$$

les coefficients cinétiques s'expriment

$$k_1 = \bar{k}_{10}u \text{ et } k_2 = \bar{k}_{20}u^\alpha. \quad (3.10)$$

Six cas ont été simulés : des coûts économiques et de trajectoire pour le système sans contrainte, avec contrainte de chemin et avec contrainte terminale. Des fonctions barrières ont été utilisées pour représenter les contraintes au niveau de la programmation. Les détails des différents cas sont présentés dans la section 4.1 de l'article. Les valeurs des paramètres sont présentées au Tableau 1 et les résultats au Tableau 2 du même article. Les codes utilisés pour générer les résultats sont présentés à l'Annexe C.

Les valeurs obtenues illustrent les différentes situations possibles : coût économique où la ré-optimisation améliore le résultat, coût économique où la ré-optimisation détériore le résultat et coût de trajectoire réalisable où la ré-optimisation améliore le résultat.

3.2 Deuxième article

Le second article produit dans le cadre de ce projet de maîtrise, présenté à l'annexe B, visait à étudier la robustesse de différentes solutions composées d'une série d'arcs. Le procédé étudié, présenté dans Srinivasan & al (2003), consiste en un réacteur semi-batch non-isotherme avec des réactions en série ($A + B \rightarrow C \rightarrow D$) et une contrainte de chaleur. La température du réacteur et le taux d'alimentation sont les variables d'entrée. La dynamique du système est donnée par

$$\dot{c}_A = -k_1 c_A c_B - \frac{u}{V} c_A, \quad (3.11)$$

$$\dot{c}_B = -k_1 c_A c_B + \frac{u}{V} (c_{Bin} - c_B), \quad (3.12)$$

$$\dot{c}_C = k_1 c_A c_B - k_2 c_C - \frac{u}{V} c_C, \quad (3.13)$$

$$\dot{V} = u, \quad (3.14)$$

où c_X est la concentration en X, c_{Bin} la concentration en B dans l'alimentation, k_i le coefficient cinétique pour la réaction i , u le débit d'alimentation en B, et V le volume dans le réacteur. Les coefficients cinétiques sont obtenus par l'équation d'Arrhenius :

$$k_i = k_{i0} \exp\left(-\frac{E_i}{RT}\right), \quad (3.15)$$

où k_{i0} est le facteur pre-exponentiel, E_i l'énergie d'activation, R la constante des gaz parfaits et T la température.

Le problème d'optimisation vise à maximiser la production de C :

$$\begin{aligned}
 \max_{u,T} J &= c_C(t_f)V(t_f) \\
 \text{s. t. } T_{\min} &\leq T(t) \leq T_{\max} \\
 u_{\min} &\leq u(t) \leq u_{\max} \\
 V(t) &\leq V_{\max} \\
 (-\Delta H_1)k_1c_A(t)c_B(t)V(t) &+ (-\Delta H_2)k_2c_C(t)V(t) \leq q_{rx,\max}.
 \end{aligned} \tag{3.16}$$

Les détails des valeurs des paramètres sont présentés au Tableau 1 de l'Annexe B.

Le débit d'alimentation est soit à l'une de ses bornes (u_{\min} ou u_{\max}) ou déterminé par la contrainte de chaleur (u_{path}) selon l'équation tirée de Srinivasan et al (2003).

La température est soit à l'une de ses bornes (T_{\min} ou T_{\max}), définie par la contrainte de chaleur (T_{path}) ou définie par une expression du compromis entre la production et la consommation de C (T_{sens}). Les paramètres présentés au Tableau 1 de l'Annexe B ont été ajustés pour que la contrainte de chaleur soit active au départ pour $T(0) = T_{\max}$, sans quoi un arc supplémentaire serait nécessaire.

L'article étudie trois solutions possibles composées de trois à cinq arcs. Les temps de passage d'un arc à l'autre sont déterminés soit par l'arrivée d'une entrée à l'une de ses bornes, soit par optimisation. Les trois variantes étudiées sont présentées au Tableau 2 de l'Annexe B, la première étant celle présentée dans l'article de Srinivasan et al (2003).

Des simulations sont d'abord complétées pour des situations avec quelques variations de contraintes, puis, dans la section suivante, avec des variations paramétriques afin de vérifier la sensibilité des solutions proposées vis-à-vis des incertitudes paramétriques. Dans ce dernier cas, des arcs supplémentaires au départ des procédés ont parfois été nécessaires, puisque, avec un changement de paramètres, la contrainte de chaleur n'était plus atteinte. Les résultats obtenus sont présentés aux Tableaux 3 et 4 de l'Annexe B.

Le principal résultat observé dans cette note est que, peu importe les variations paramétriques, la forme générale de la meilleure solution (i.e. la séquence d'arcs) ne change pas. Autrement dit, l'ensemble des contraintes actives est invariant, ce qui vient supporter l'une des hypothèses de l'article 1. Ainsi, même si le coût optimal n'est pas robuste, les contraintes actives le sont.

CHAPITRE 4 RÉSULTATS COMPLÉMENTAIRES ET DISCUSSION GÉNÉRALE

Pour des raisons de concision, les deux articles présentés en annexe comportent seulement une partie des résultats obtenus par simulation. En effet, plus de simulations ont été faites que celles s’y retrouvant et d’autres système ont été simulés. Certains de ces autres résultats, pertinents à la discussion, sont donc ajoutés ici.

Une discussion générale concernant l’ensemble des résultats et certains aspects des analyses faites dans les deux articles suit.

4.1 Résultats complémentaires à l’article 1

4.1.1 Simulations supplémentaires

Un certain nombre de résultats de simulations sont présentés dans l’article 1 pour des cas de coût économique et de coût de trajectoire pour un système en particulier. Cependant, d’autres simulations pertinentes ont été faites sur ce système, de même que sur d’autres systèmes. Une partie de ces résultats supplémentaires est présentée ici.

4.1.1.1 Réacteur à deux réactions (Système de l’article 1)

Pour les simulations présentées dans l’article 1, les trajectoires avaient été spécifiquement choisies de manière à ce qu’elles soient réalisables, afin de représenter la conclusion de l’analyse faite sur la NMPC. D’autres simulations avaient été faites, mais pour des trajectoires non réalisables. Les résultats correspondants sont présentés au Tableau 4.1. Ceux-ci sont pertinents, car ils montrent bien que, lorsque l’hypothèse de trajectoire réalisable n’est pas respectée, la preuve de la proposition 2 de l’article 1 ne tient plus.

Par ailleurs, d’autres variations paramétriques ont été faites, mais ne sont pas présentées puisqu’elles donnaient des résultats semblables et n’apportent rien de plus à la discussion.

Tableau 4.1: Comparaison des solutions hors ligne, avec ré-optimisation et optimales pour l'usine pour des coûts avec une trajectoire non réalisable

Cas	Erreur paramétrique	Coût		
		Hors ligne	Ré-optimisation	Optimum de l'usine
Sans contrainte	--	0,82	0,82	0,82
	\bar{k}_{10} : -20%; α : -20%	1,77	1,71	1,62
	\bar{k}_{10} : -20%; α : +20%	1,90	1,93	1,89
Contrainte terminale	--	0,94	0,94	0,94
	\bar{k}_{10} : -20%; α : -20%	4,04	3,39	1,78
	\bar{k}_{10} : -20%; α : +20%	2,28	2,34	2,22
Contrainte de chemin	--	0,96	0,96	0,96
	\bar{k}_{10} : -20%; α : -20%	1,73	1,72	1,65
	\bar{k}_{10} : -20%; α : +20%	2,3011	2,2967	2,2962

4.1.1.2 Réacteur d'isomérisation batch

Le premier procédé simulé dans le cadre de ce projet de maîtrise fut un réacteur d'isomérisation batch avec la réaction à l'équilibre $A \leftrightarrow B$. Du A est présent dans le réacteur au début de la batch et le produit désiré est le B.

Comme mentionné à la section 1.3, ce sont certains résultats présentés pour ce procédé dans Kamaraju & al (2013) qui ont inspiré ce projet de maîtrise. En effet, des cas y sont présentés montrant la détérioration du coût lors de la ré-optimisation comparativement à l'optimisation hors ligne. C'est ce qui a entraîné le questionnement quant à l'existence de circonstances entraînant plus de robustesse intrinsèque à la NMPC. L'observation de la Figure 4.1 a confirmé l'intention d'arriver aux résultats présentés à l'article de l'annexe A.

D'un bilan de matière, le modèle dynamique suivant est obtenu (Kamaraju & al, 2013) :

$$\dot{c}_A = -k_1 c_A + k_2 (c_{A0} - c_A) \quad (4.1)$$

où c_A est la concentration en A, c_{A0} la concentration initiale en A, k_1 et k_2 les coefficients cinétiques de réaction. Les valeurs des différents paramètres sont présentées au Tableau 4.2.

De la même manière que dans Kamaraju & al (2013), on obtient les équations suivantes :

$$k_i = k_{i0} \exp\left(-\frac{E_i}{RT}\right) \quad (4.2)$$

$$u = k_{10} \exp\left(-\frac{E_1}{RT}\right) \quad (4.3)$$

$$\alpha = \frac{E_2}{E_1}, \quad \bar{k}_{10} = 1 \text{ et } \bar{k}_{20} = k_{20} \left(\frac{1}{k_{10}}\right)^\alpha \quad (4.4)$$

$$k_1 = \bar{k}_{10}u \text{ et } k_2 = \bar{k}_{20}u^\alpha. \quad (4.5)$$

Considérant un objectif de maximisation de la concentration finale en B, sachant que, pour ce procédé, $c_B = c_{A0} - c_A$, le problème d'optimisation pour ce système peut s'écrire comme suit :

$$\begin{aligned} \min_{u(t)} J &= c_A(t_f) \\ \text{s. t. } &(4.1), \end{aligned} \quad (4.6)$$

où t_f est le temps final d'une batch.

Pour les simulations avec un coût de trajectoire, une valeur de 80% de la trajectoire optimale pour la concentration en B a été choisie, puis traduite en concentration en A, sachant que, pour ce procédé, $c_B = c_{A0} - c_A$. Le coût s'écrit donc :

$$\min_{u(t)} J = \left(c_A - c_{Aref}\right)^T \left(c_A - c_{Aref}\right). \quad (4.7)$$

Tableau 4.2: Paramètres du modèle, limites d'opération et conditions initiales pour le réacteur d'isomérisation batch

Paramètre	Valeur	Unités
c_{A0}	5	mol/L
k_{10}	5×10^3	hr-1
k_{20}	7×10^{16}	
E_1	2×10^4	J/mol
E_2	1×10^5	J/mol
R	8.314	J/mol.K
A	5	--
\bar{k}_{10}	1	--
\bar{k}_{20}	0,0224	--
t_f	1	h

Par ailleurs, pour ce procédé, des simulations pour beaucoup de variations de paramètres ont été faites. Les figures 4.1 a) et b) montrent les coûts obtenus en fonction des erreurs paramétriques. Sur 4.1 a), les courbes se croisent, donnant à certains moments un meilleur résultat avec l'optimisation hors ligne et à d'autres un meilleur résultat avec la NMPC, tel qu'attendu pour un

coût économique. Sur la figure 4.1 b), par contre, la NMPC donne toujours de meilleur résultat que l'optimisation hors ligne, avec un coût de trajectoire quadratique. C'est ce graphique qui a poussé à vérifier s'il y avait une raison mathématique justifiant la robustesse de la NMPC avec un coût quadratique, ce qui a été accompli dans l'article 1. Quelques résultats individuels sont présentés au Tableau 4.3.

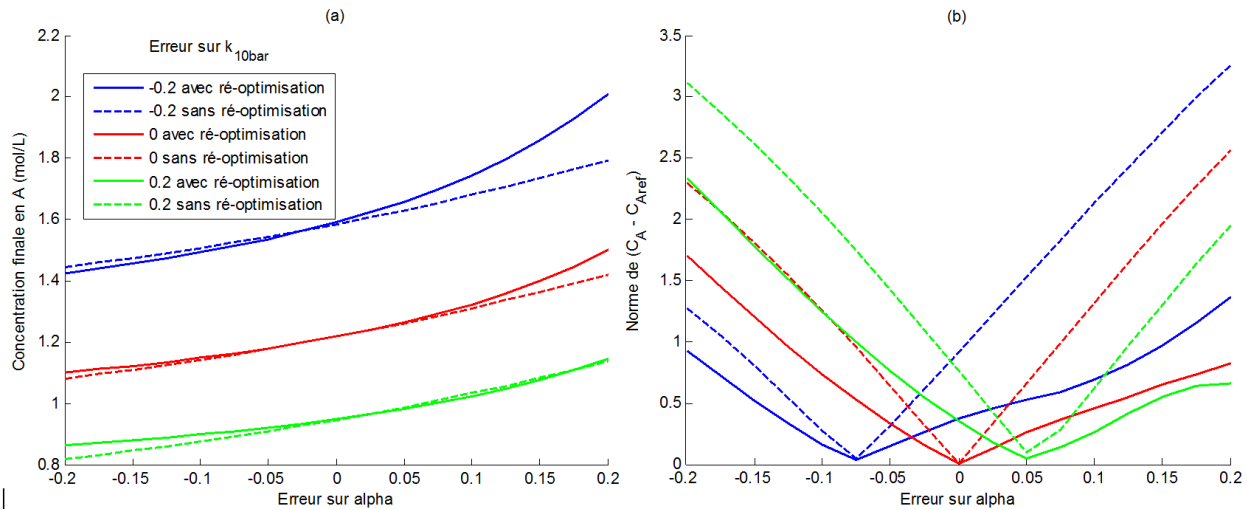


Figure 4.1: a) Résultats pour un coût économique : Concentration finale en A en fonction de l'erreur sur α et sur \bar{k}_{10} .

b) Résultats pour un coût de trajectoire : Somme des carrés des écarts à la trajectoire en fonction de l'erreur sur α et sur \bar{k}_{10} .

Tableau 4.3: Comparaison des solutions hors ligne, avec ré-optimisation et optimales pour l'usine pour le réacteur d'isomérisation

Cas	Erreur paramétrique	Coût		
		Hors ligne	NMPC	Optimum de l'usine
Coût économique (minimiser $c_A(t_f)$)	--	1,21	1,21	1,21
	\bar{k}_{10} : -20%; α : -20%	1,44	1,42	1,34
	\bar{k}_{10} : -20%; α : +20%	1,79	2,08	1,73
Coût de trajectoire quadratique	--	0	0	0
	\bar{k}_{10} : -20%; α : -20%	0,56	0,39	0
	\bar{k}_{10} : -20%; α : +20%	1,94	0,94	0,26

4.1.1.3 Réacteur semi-batch isotherme avec une contrainte de sécurité

Un réacteur semi-batch isotherme avec une contrainte de sécurité, tel que présenté dans Srinivasan & al (2003), a aussi été simulé dans le cadre de ce projet de maîtrise. La contrainte de sécurité s'applique à la température maximale pouvant être atteinte en cas de défaillance du système de refroidissement. Cette température, appelée T_{cf} , dépend des concentrations de réactifs restant dans le réacteur. Cette température se calcule comme suit :

$$T_{cf}(t) = T(t) + \min(c_A(t), c_B(t)) \frac{-\Delta H}{\rho c_p}, \quad (4.8)$$

où c_X représente la concentration en X, T la température, ΔH l'enthalpie de réaction, ρ la masse volumique et c_p la chaleur massique.

Le modèle dynamique suivant est obtenu par bilan de matière :

$$\dot{c}_A = -k c_A c_B - \frac{u}{V} c_A, \quad c_A(0) = c_{A0} \quad (4.9)$$

$$\dot{c}_B = -k c_A c_B - \frac{u}{V} (c_{Bin} - c_B), \quad c_B(0) = c_{B0} \quad (4.10)$$

$$\dot{V} = u, \quad V(0) = V_0, \quad (4.11)$$

où c_{Bin} est la concentration en B dans l'alimentation au réacteur, V le volume occupé dans le réacteur et u le débit d'alimentation. Les valeurs utilisées sont présentées au Tableau 4.4 et les résultats obtenus pour ce système au Tableau 4.5.

L'objectif choisi pour ce système étant de maximiser la quantité finale de C, le problème d'optimisation peut s'écrire comme suit :

$$\begin{aligned} \min_{u(t)} J &= n_A(t_f) \\ \text{s. t. } &(4.9), (4.10), (4.11) \\ &u_{min} \leq u(t) \leq u_{max} \\ &T_{cf}(t) \leq T_{cfmax} \\ &V(t) \leq V_{max} \end{aligned} \quad (4.12)$$

Par ailleurs, des simulations ont été faites en remplaçant les contraintes de température et de volume par une contrainte sur la concentration finale en B, comme suit :

$$c_B(t_f) \leq c_{Bfmax}. \quad (4.13)$$

Comme dans l'article de l'annexe A, les contraintes sont représentées sous forme de fonctions barrières additionnées au coût pour les fins de simulations numériques :

$$b(c) = \begin{cases} -\gamma \log(c - c_{max}), & c > \beta c_{min} \\ \frac{\gamma(c - \beta c_{max})}{(1 - \beta)c_{max}}, & c \leq \beta c_{min} \end{cases} \quad (4.14)$$

où $b(c)$ est une fonction barrière pour la contrainte $c - c_{max} \leq 0$.

Tableau 4.4: Paramètres du modèle, limites d'opération et conditions initiales pour le réacteur semi-batch isotherme avec contrainte de sécurité

Paramètres	Valeurs	Unités
k	0,0482	L/mol.hr
T	70	°C
ΔH	-60000	J/mol
ρ	900	g/L
c_p	4,2	J/g.K
c_{Bin}	2	mol/L
u_{min}	0	L/h
u_{max}	0,1	L/h
$T_{cf_{max}}$	80	°C
V_{max}	1	L
$c_{Bf_{max}}$	0,5	Mol/L
V_0	0,7	L
c_{A0}	2	mol/L
c_{B0}	0,63	mol/L

Tableau 4.5: Comparaison des solutions hors ligne, avec ré-optimisation et optimales pour l'usine pour le réacteur semi-batch isotherme avec contrainte de sécurité

Cas	Erreur paramétrique	Coût		
		Hors ligne	NMPC	Optimum de l'usine
Contrainte terminale	--	0,71	0,71	0,71
	c_{Bin} : -20%; k : -20%	0,85	0,90	0,83
	c_{Bin} : +20%; k : -20%	1,04	0,95	0,81
	c_{Bin} : +20%; k : +20%	0,64	0,62	0,60
Contrainte de chemin	--	0,80	0,80	0,80
	c_{Bin} : -20%; k : -20%	0,92	0,91	0,91
	c_{Bin} : +20%; k : -20%	1,56	0,95	0,90
	c_{Bin} : +20%; k : +20%	0,83	0,74	0,73

4.2 Résultats complémentaires à l'article 2

Plusieurs résultats sont présentés aux Tableaux 3 et 4 de l'article 2. Il est fait mention de simulations supplémentaires effectuées pour différentes variations de paramètres. Ces résultats sont compilés dans le tableau qui suit.

Tableau 4.6: Production de C pour les 3 solutions envisagées, considérant des changements de contrainte, de condition initiales ou de paramètres

Changement	Production de C		
	Solution 1 T_{\max} -3 arcs	Solution 2 T_{\max} -5 arcs	Solution 3 u_{\max} -5 arcs
--	2.0158	2.0499	2.0529
k_{10} : -10%	1.9558	1.9913	1.9948
E_1 : -2%	2.0417	2.0729	2.0762
E_2 : -5%	1.9608	1.9655	1.9709
$q_{rx,\max}$: +10%	2.0657	2.0964	2.0997
$q_{rx,\max}$: +15%	2.0485	2.1131	2.1164
T_{\max} : -2%	2.0162	2.0492	2.0521
T_{\max} : +2%	2.0155	2.0496	2.0529
ΔV_{\max} + 5%	2.0528	2.0882	2.0914
C_{A0} : -10%	1.9425	1.9768	1.9807
C_{A0} : +10%	2.0835	2.1077	2.111
C_{B0} : -10%	1.9658	1.9977	2.0008
C_{B0} : +10%	2.0691	2.0976	2.1008
V_0 : -3%	2.199	2.2256	2.2305
V_0 : +4%	1.6751	1.6932	1.6954

4.3 Discussion générale

4.3.1 Hypothèses effectuées

Dans l'article de l'annexe A, quelques résultats ont été présentés concernant la sensibilité aux erreurs paramétriques de la NMPC. Pour y arriver, les hypothèses suivantes sont considérées :

- Contraintes actives invariantes;
- Variations paramétriques suffisamment petites pour négliger les termes d'ordre supérieur;
- Hessien défini positif.

Or, certaines hypothèses supplémentaires ont été implicitement faites. Celles-ci sont expliquées dans les sections qui suivent.

4.3.1.1 Horizon de commande correspondant à l'horizon de prédiction

D'abord, l'horizon de commande pris dans chaque simulation correspondait à l'horizon de prédiction. Cependant, l'horizon de commande peut être bien plus court. Ainsi, dans certains cas, la NMPC n'optimise que pour une valeur de la variable d'entrée, supposant que celle-ci demeurera constante. Au pas de temps suivant, cette valeur est recalculée de la même manière puis modifiée dans le procédé.

Cette façon de faire est beaucoup plus robuste, puisque moins de risques sont pris. En effet, lorsque l'horizon de commande correspond à l'horizon de prédiction, la NMPC obtient des variables d'entrée beaucoup plus agressives, considérant que la situation sera corrigée plus tard. À l'opposé, quand l'horizon de commande est plus court, la NMPC obtient une entrée plus conservatrice, puisqu'elle devra, dans le cadre de l'optimisation, suffire à respecter toutes les contraintes tout en obtenant le meilleur coût possible.

Pour un procédé continu, il est généralement souhaité qu'un régime permanent soit atteint après un certain temps. La variable d'entrée devra alors prendre une valeur constante, ce qui justifie l'utilisation d'un horizon de commande plus court que l'horizon de prédiction. Pour un procédé batch, l'atteinte d'un régime permanent n'est pas recherchée. Pour cette raison, la formulation avec un horizon de commande égal à l'horizon de prédiction paraît plus appropriée. Celle-ci a donc été choisie dans les simulations.

4.3.1.2 États mesurables

Dans un procédé, de nombreuses mesures peuvent être effectuées sur différentes variables : température, pression, niveau, etc. Cependant, certaines variables peuvent être impossibles à mesurer. Or, la NMPC requiert les valeurs des états à chaque pas de temps afin de fonctionner. Advenant que ceux-ci ne soient pas mesurables, il faut donc utiliser des observateurs, i.e. des méthodes permettant de les estimer. Ceux-ci se basent sur un modèle du système afin d'estimer la valeur de la variable requise. Comme pour la commande prédictive, le modèle utilisé par l'observateur est sujet aux erreurs paramétriques.

Dans le cadre de ce projet, les démonstrations et simulations ont été faites en supposant que les états sont mesurables, ce qui peut constituer une simplification de la réalité selon le procédé étudié.

4.3.1.3 Variables d'entrée modifiables instantanément

Dans les simulations effectuées, il a été supposé que la variable d'entrée est parfaitement manipulable. En effet, dans l'article 1, elle change par paliers et, dans l'article 2, elle suit parfaitement les courbes sélectionnées. Or, la température se trouve être une variable d'entrée dans la plupart des simulations faites dans ce projet. En général, cette variable est elle-même contrôlée par un débit ou une température dans une chemise de refroidissement. Comme la température possède généralement une dynamique plus lente que les autres composantes des procédés présentés, elle ne peut en général suivre exactement les courbes présentées. L'inclusion d'un bilan d'énergie dans les modèles des systèmes aurait permis de prendre en compte la dynamique de la température.

4.3.2 Principaux résultats

L'article 1 présente deux propositions, celles-ci constituant le cœur de la recherche effectuée dans le cadre de ce projet.

La première proposition explique mathématiquement les problèmes déjà observés que l'application simple de la NMPC peut entraîner en présence d'incertitudes paramétriques, c'est-à-dire la dégradation de la performance par rapport à l'optimisation hors ligne. Cela justifie l'intérêt des diverses méthodes pour parer aux incertitudes présentées dans le Chapitre 2 : NMPC robuste, stochastique, adaptation des paramètres, etc.

La seconde proposition, basée sur l'hypothèse additionnelle que les variations dans les états ne sont causées que par les erreurs paramétriques, apporte un élément important en précisant que, pour un coût de trajectoire quadratique, la NMPC améliore toujours la performance par rapport à une optimisation hors ligne. Ainsi, le coût de trajectoire quadratique rend la NMPC moins sensible aux erreurs paramétriques. Ce résultat est intéressant, car il peut arriver que l'utilisation d'une méthode plus complexe s'avère impossible pour diverses raisons, notamment concernant la taille du problème et la possibilité de résoudre celui-ci.

Le second article permet de valider dans une certaine mesure une hypothèse du premier article. Plusieurs séquences d'arcs servant de solution à l'optimisation d'un procédé batch y sont comparées, afin de déterminer laquelle donne la meilleure performance. Une analyse de sensibilité est faite, en faisant varier les contraintes et les paramètres. Dans tous les cas, la même séquence d'arcs s'est avérée la meilleure. Certains cas présentaient le besoin d'un arc initial, mais dans l'ensemble, la meilleure solution présentait peu de sensibilité au changement de paramètre. Ainsi, la forme générale de la solution optimale ne changeait pas, supportant l'hypothèse d'invariance des contraintes active du premier article.

CHAPITRE 5 CONCLUSION ET RECOMMANDATIONS

5.1 Conclusion

L'objectif au cœur de ce projet consistait à analyser la sensibilité de la commande par optimisation vis-à-vis des erreurs paramétriques, notamment dans le cadre de la NMPC. Des résultats de simulations montrant que, pour des coûts économiques, la NMPC pouvait performer à l'occasion moins bien qu'une optimisation hors ligne ont entraîné cette étude.

Les travaux effectués dans le cadre de ce projet ont procédé en deux étapes. D'abord, des simulations supplémentaires avec différents systèmes, coûts et contraintes ont été effectuées, mettant en évidence les problèmes de sensibilités des coûts économiques, mais aussi une apparente robustesse des coûts de trajectoire. Ensuite, une analyse mathématique a été effectuée sur la NMPC, menant à deux résultats. Ainsi, une première démonstration a expliqué qu'il était normal et attendu que la NMPC donne à l'occasion une moins bonne performance qu'une optimisation hors ligne. La seconde démonstration a porté le résultat le plus intéressant. En effet, en acceptant quelques hypothèses, notamment qu'il y a des erreurs paramétriques mais pas de bruit de procédé, il est montré que la NMPC avec un coût de trajectoire quadratique réalisable donnera une meilleure performance qu'une optimisation hors ligne. Autrement dit, le coût de trajectoire quadratique confère à la NMPC une robustesse naturelle absente pour un coût général.

Enfin, une analyse a été faite en parallèle sur la sensibilité de séquences d'arcs servant de solution pour l'optimisation d'un procédé batch. Dans tous les cas, la même séquence s'est avérée la meilleure, montrant la faible sensibilité de la trajectoire optimale aux variations paramétriques, c'est-à-dire que les contraintes actives demeuraient constantes.

5.2 Recommandations

Afin de pousser plus loin les résultats obtenus dans le cadre de ce projet, certaines investigations supplémentaires seraient pertinentes. Entre autres, la conclusion quant à la robustesse de la NMPC avec un coût de trajectoire quadratique ne peut être généralisée à une situation comportant à la fois des erreurs paramétriques et des bruits de procédés et de mesures. Il serait donc pertinent d'analyser plus en profondeur ces cas plus près de la réalité. S'il n'est pas possible de garantir la robustesse de la NMPC en présence de bruit de procédé et d'erreur paramétrique, est-il possible

d'établir que, pour certains rapports entre ceux-ci, la robustesse demeure? Par exemple, si le bruit de procédé est très faible par rapport aux erreurs paramétriques, intuitivement, il semble normal que la conclusion tienne.

De même, seule la situation où les états sont mesurables a été considérée. Des analyses supplémentaires, incluant des états non-mesurables et des observateurs, permettraient de pousser plus loin les résultats obtenus.

Par ailleurs, toujours concernant la NMPC, ce projet n'a montré de robustesse supplémentaire que pour le coût de trajectoire quadratique. Il est démontré qu'une telle conclusion est impossible pour un coût général. Cependant, est-il possible que d'autres formes de coûts présentent des caractéristiques semblables?

Si l'analyse mathématique de ces situations supplémentaires s'avère trop complexe, il demeurerait intéressant de voir pour celles-ci les résultats de simulations.

BIBLIOGRAPHIE

- 1- Instrumentation, Systems, and Automation Society (1995). Batch Control Part 1: Models and Terminology, Norme ANSI/ISA–88.01–1995. Research Triangle Park, North Carolina : Instrument Society of America
- 2- Bello O., Hamam Y., Djouani K. (2014) Nonlinear Model Predictive Control of a coagulation chemical dosing unit for water treatment plants. Communication présentée à la 19th World Congress The International Federation of Automatic Control, Cape Town, South Africa (p. 370-376). Tiré de <http://www.sciencedirect.com/science/article/pii/S1474667016416423>
- 3- Bemporad A., Morari M. (1999) Robust Model Predictive Control: A Survey. Lecture Notes in Control and Information Sciences. Volume 245
- 4- Ben-Tal A., Nemirovski A. (1999) Robust solutions of uncertain linear programs. *Operations Research Letters*. 25(1),1-13.
- 5- Bertsimas D., Brown D.B., Caramanis C. (2011) Theory and Applications of Robust Optimization. *Society for Industrial and Applied Mathematics*. 53 (3), 464–501
- 6- Bonvin D., Srinivasan B. (2013) On the role of the necessary conditions of optimality in structuring dynamic real-time optimization schemes. *Computers and Chemical Engineering* 51, 172– 180
- 7- Bonvin D., Srinivasan B., Hunkeler D. (2006) Control and Optimization of Batch Processes – Improvement of Process Operation in the production of specialty chemicals. *IEEE Control Systems Magazine*
- 8- Camacho E.F., Alba C.B. (1995) Model Predictive Control. (2^e edition)
- 9- Canale, M., Fagiano, L., Signorile, M. C. (2012) Nonlinear model predictive control from data: a set membership approach. *International Journal of Robust and Nonlinear Control*. 24. 123-139. doi:10.1002/rnc.2878
- 10- Cervantes A., Biegler L.T. (1998) Large-scale DAE optimization using a simultaneous NLP formulation. *AIChE J.*, 44 (5), 1038–1050. doi:10.1002/aic.690440505
- 11- Chisci L., Lombardi A., Mosca E. (1996) Dual receding horizon control of constrained discrete time systems. *European Journal of Control*, 2, 278-285. doi:10.1016/S0947-3580(96)70052-3
- 12-Clarke D.W., Mohtadi C., Tuffs P.S. (1987a). Generalized predictive control—Part I: the basic algorithm. *Automatica*, 23, 137–148. doi:10.1016/0005-1098(87)90087-2
- 13- Clarke D.W., Mohtadi C., Tuffs P.S. (1987b). Generalized predictive control—Part II: extensions and interpretations. *Automatica*, 23, 149–160. doi:10.1016/0005-1098(87)90088-4
- 14- Cutler C. R., Ramaker B. L. (1979). *Dynamic matrix control—a computer control algorithm*. AIChE 86th National Meeting. Houston,TX.
- 15- De Nicolao G., Magni L., Scattolini R. (1998) Stabilizing receding-horizon control of nonlinear time-varying systems. *IEEE Transactions on Automatic Control*, 43(7). Tiré de <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=701133>
- 16- De Souza G., Odloak D. Zanin A.C. (2010) Real time optimization (RTO) with model predictive control (MPC). *Computers and Chemical Engineering* 34 (12), 1999-2006. Tiré de <http://www.sciencedirect.com/science/article/pii/S0098135410002498>

- 17- Edgar T.F., Smith C.L., Shinskey F.G., Gassman G.W., Waite A.W.R., McAvoy T.J., Seborg D.E. (2008) Section 8 Process Control. Dans Green D.W., Perry R.H (édit.) *Perry's Chemical Engineer Handbook* (8^e éd. P. 881-979) doi:10.1036/0071422943
- 18- El Ghaoui L., Lebret H. (1997) Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Analysis and Applications*, 18(4),1035-1064. Tiré de <https://people.eecs.berkeley.edu/~elghaoui/Pubs/rob-ls.pdf>
- 19- Etchepareborda A., Lolich J. (2007) Research reactor power controller design using an output feedback nonlinear receding horizon control method. *Nuclear Engineering and Design* 237, 268–276. Tiré de <http://www.sciencedirect.com/science/article/pii/S0029549306002731>
- 20- Findeisen R., Allgöwer F., Beigler L.T. (2007) Assessment and future directions of nonlinear model predictive control. Berlin (Allemagne). Springer. Tiré de <http://lib.mylibrary.com/Open.aspx?id=106602>
- 21- Fontes F.A.C.C., Magni L. (2003) Min-max model predictive control of nonlinear systems using discontinuous feedbacks. *IEEE Transactions On Automatic Control* 48(10), 1750-1755. Tiré de <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1235378>
- 22- Francisco M., Vega P., Skogestad S. (2015) *Nonlinear offset free MPC for self-optimizing control in wastewater treatment plants*. Communication à la 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romania. doi:10.1109/ICSTCC.2015.7321325
- 23- Gaida D., Wolf C., Bäck T., Bongards M. (2012) *Nonlinear Model Predictive Substrate Feed Control of Biogas Plants*. Communication à la 20th Mediterranean Conference on Control & Automation (MED) Barcelona, Espagne. Tiré de <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6265712>
- 24- Grancharova, A. (2012) Nonlinear Model Predictive Control. Dans Explicit nonlinear model predictive control: Theory and application. (P.39-69). doi:10.1007/978-3-642-28780-0
- 25- Grüner, L., Pannek, J. (2011) *Nonlinear Model Predictive Control : Theory and Algorithm*. doi : 10.1007/978-0-85729-501-9
- 26- He, D.F., Ji, H.B., Zheng, T. (2010) On robustness of constrained non-linear H-infinite predictive controllers with disturbances. *International Journal of System Science*.41(2) 203-212. doi:10.1080/00207720903045742
- 27-He, D.F., Ji, H.B., Yu, L. (2013) Constructive robust model predictive control for constrained non-linear systems with disturbances. *IET Control Theory and Applications*,7(15) 1869-1876. doi:10.1049/iet-cta.2012.0841
- 28- Henson M. A. (1998) Nonlinear model predictive control: current status and future directions. *Computers and Chemical Engineering* 23 (2) 187-202. doi:10.1016/S0098-1354(98)00260-9
- 29-Kadam J.V., Schlegel M., Srinivasan B., Bonvin D., Marquardt W. (2007) Dynamic optimization in the presence of uncertainty: From off-line nominal solution to measurement-based implementation. *Journal of Process Control*, 17 (5) 389–398. Tiré de <http://www.sciencedirect.com/science/article/pii/S095915240600076X>
- 30-Kamaraju V.K., Chiu M.S., Srinivasan B. (2013) Reformulating real-time optimal feedback based on model uncertainty. *IFAC Proceedings Volumes*, 10 (32) 756-761. doi:10.3182/20131218-3-IN-2045.00084

- 31- Keerth S.S., Gilbert E.G. (1988) Optimal Infinite-Horizon Feedback Laws for a General Class of Constrained Discrete-Time Systems: Stability and Moving-Horizon Approximations. *Journal of Optimization Theory and Applications*, 57 (2). Tiré de [http://web.eecs.umich.edu/~grizzle/GilbertFest/\(57\).pdf](http://web.eecs.umich.edu/~grizzle/GilbertFest/(57).pdf)
- 32- Kwon W.H., Han S., Ahn C. K. (2004) Advances in Nonlinear Predictive Control: A Survey on Stability and Optimality. *International Journal of Control, Automation, and Systems*, 2 (1). Tiré de https://www.researchgate.net/publication/239572661_Advances_in_Nonlinear_Predictive_Control_A_Survey_on_Stability_and_Optimality
- 33- Lucia S., Finkler T., Engell S. (2013) Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23 (9) 1306-1319. Tiré de <http://www.sciencedirect.com/science/article/pii/S0959152413001686>
- 34- Magni, L., Raimondo, D.M., Allgöwer, F. (2009) *Nonlinear model predictive control: towards new challenging applications*. Berlin (Allemagne). Springer. doi:10.1007/978-3-642-01094-1
- 35- Maree J.P., Imsland L. (2016) Combined economic and regulatory predictive control. *Automatica*, 69, 342–347. Tiré de <http://www.sciencedirect.com/science/article/pii/S000510981500535X>
- 36- Mesbah A., Streif S., Findeisen R., Braatz R.D. (2014) *Stochastic nonlinear model predictive control with probabilistic constraints*. Communication à la American Control Conference (ACC). Tiré de <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6858851>
- 37- Mockus L., Reklaitis G.V. (1999) Continuous Time Representation Approach to Batch and Continuous Process Scheduling. *Ind. Eng. Chem. Res.*, 38 (1), 197-203. doi:10.1021/ie970311r
- 38- Morari M., Lee J. H. (1999) Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23 (4-5) 667–682. Tiré de <http://www.sciencedirect.com/science/article/pii/S0098135498003019>
- 39- Morrisson, D. (2010). Addressing the Myths of Model Predictive Control (MPC). Consulté le 31 mars 2014. Tiré de <http://www.automation.com/automation-news/article/addressing-the-myths-of-model-predictive-control-mpc>
- 40- Pahija E., Manenti F., Mujtaba I.M., Rossi F. (2014) Assessment of control techniques for the dynamic optimization of (semi-)batch reactors. *Computers and Chemical Engineering*, 66, 269–275. Tiré de <http://www.sciencedirect.com/science/article/pii/S0098135414000544>
- 41- Primbs J., Nevistic V., Doyle J. (2000) A receding horizon generalization of pointwise min-norm controllers. *IEEE Transactions on Automatic Control*, 45 (5), 898-909. Tiré de <http://authors.library.caltech.edu/2348/1/PRlieetac00.pdf>
- 42- Revollar S., Vega P. Vilanova R. (2015) *Economic optimization of wastewater treatment plants using Non Linear Model Predictive Control*. Communication à la 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romania. doi:10.1109/ICSTCC.2015.7321356
- Richalet J., Rault A., Testud J.L., Papon J. (1978) Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14 (5) 413-428. doi:10.1016/0005-1098(78)90001-8
- 43- Santos, L.O., Dewasme, L., Coutinho, D., Vande Wouwer, A. (2012) Nonlinear model predictive control of fed-batch cultures of micro-organisms exhibiting overflow

- metabolism: Assessment and robustness. *Computers and Chemical Engineering*, 39, 143-151. Tiré de <http://www.sciencedirect.com/science/article/pii/S0098135411003474>
- 44- Sastry S., Bodson M. (1989) *Adaptive Control Stability, Convergence, and Robustness*. Englewood Cliff, New Jersey, Prentice Hall.
- 45- Schlegel M., Marquardt W. (2006) Adaptive switching structure selection for the solution of dynamic optimization problems. *Ind. Eng. Chem. Res.*, 45 (24), 8083–8094. doi:10.1021/ie060496e
- 46- Scokaert P., Mayne D., (1998) Min–max feedback model predictive control for constrained linear systems, *IEEE Transactions on Automatic Control* 43 (8) 1136–1142. Tiré de <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=704989>
- 47- Srinivasan B., Palanki S., Bonvin D. (2003) Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Computers and Chemical Engineering*, 27 (1), 1-26. Tiré de <http://www.sciencedirect.com/science/article/pii/S0098135402001163>
- 48- Srinivasan, B., Palanki, S., Bonvin, D. (2003) Dynamic optimization of batch processes II. Role of measurements in handling uncertainty. *Computers & Chemical Engineering*, 27 (1), 27-44. Tiré de <http://www.sciencedirect.com/science/article/pii/S0098135402001175>
- 49- Xi Y.G., Li D.W., Lin S. (2013) Model Predictive Control -- Status and Challenges. *Acta Automatica Sinica*, 39 (3), 222-236. doi:10.1016/S1874-1029(13)60024-5
- 50- Yuan, C., Wang, C. (2011) *Persistency of Excitation and Performance of Deterministic Learning*. Communications à la 30th Chinese Control Conference, Yantai, China. Tiré de <http://www.sciencedirect.com/science/article/pii/S0167691111001770>
- 51- Zavala V.M., Biegler L.T. (2009) The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45 (1), 86–93. Tiré de <http://www.sciencedirect.com/science/article/pii/S0005109808004196>

ANNEXE A – ARTICLE 1 : ON THE USE OF NONLINEAR MODEL PREDICTIVE CONTROL WITHOUT PARAMETER ADAPTATION FOR BATCH PROCESSES

Revue : Processes 2016, 4 (27)

Article

On the Use of Nonlinear Model Predictive Control without Parameter Adaptation for Batch Processes

Jean-Christophe Binette * and Bala Srinivasan

Département de Génie Chimique, École Polytechnique Montréal, C.P.6079 Succ., Centre-Ville Montréal, Montréal, H3C 3A7, QC, Canada; bala.srinivasan@polymtl.ca

* Correspondence: jean-christophe.binette@polymtl.ca; Tel.: +514-340-4711

Academic Editor: name

Received: date; Accepted: date; Published: date

Abstract: Optimization techniques are typically used to improve economic performance of batch processes, while meeting product and environmental specifications and safety constraints. Offline methods suffer from the parameters of the model being inaccurate, while re-identification of the parameters may not be possible due to the absence of persistency of excitation. Thus, a practical solution is the Nonlinear Model Predictive Control (NMPC) without parameter adaptation, where the measured states serve as new initial conditions for the re-optimization problem with a diminishing horizon. In such schemes, it is clear that the optimum cannot be reached due to plant-model mismatch. However, this paper goes one step further in showing that such re-optimization could in certain cases, especially with an economic cost, lead to results worse than the offline optimal input. On the other hand, in absence of process noise, for small parametric variations, if the cost function corresponds to tracking a feasible trajectory, re-optimization always improves performance. This shows inherent robustness associated with the tracking cost. A batch reactor example presents and analyzes the different cases. Re-optimizing led to worse results in some cases with an economical cost function, while no such problem occurred while working with a tracking cost.

Keywords: process optimization; batch processes; process control; constrained optimization; sensitivity; real-time optimization

1. Introduction

Batch processes are widely used in specialty industries, such as pharmaceuticals, due to their flexibility in operation. As opposed to continuous processes, their operating conditions vary with time, in order to meet the specifications and respect safety and environmental constraints. Additionally, in order to improve process operation efficiency, reduce cost, numerical optimization based on phenomenological models is used to obtain the time-varying schedule [1].

However, using an optimum, computed off-line, suffers from the problem of the model not exactly representing the reality. Very often, it is hard to get a precise model due to the lack of quality or quantity in the experimental data. In addition, in many cases, parameters are estimated from lab experiments, and thus are not very accurate when scaled-up to industrial processes.

To address this problem, use of measurements in the framework of optimization is recommended [2,3]. The idea is to repeatedly re-optimize, changing the optimization problem appropriately using the information obtained from measurements. The initial conditions of the optimization problem are adapted based on the current measurements. In addition, it is also possible to identify the parameters of the system from the measurements and update them. Thus, two main categories need to be distinguished, though there is a bit of inconsistency in the nomenclature reported in the literature. If only the initial conditions are updated, the schemes are referred to as Model Predictive Control (MPC) [4–11], while Dynamic Real Time Optimization (D-RTO) schemes incorporate adaptation of both initial conditions and parameters [12].

MPC schemes incorporate feedback by re-optimization, when computation is not prohibitive [4–7]. In this case, the model is not adapted, while a new optimum is computed from the initial conditions obtained from current measurements. Most real systems are better represented by a nonlinear model [8,9] and using Nonlinear Model Predictive Control (NMPC) is more appropriate [10,11].

In D-RTO, the parameters of the model are also adapted. The major problem with the adaptation of model parameters is the persistency of excitation. The optimal input is typically not persistently exciting, and adding an excitation for the purpose of identification would cause sub-optimality [13]. Thus, in short, D-RTO is very difficult to implement except in special cases [14,15].

NMPC schemes do not get to the optimum due to plant-model mismatch, while D-RTO is not practical to implement. An intermediary solution is the robust NMPC reported in the literature [16,17]. The most known is the min-max method, that considers the worst-case scenario for optimization [18]. This method, however, is very conservative and clearly not optimal. Other methods such as the multi-stage NMPC [18] seek a compromise between conservatism and optimality. Stochastic NMPC [19] considers a probabilistic setting for the parameter uncertainties, and seeks an optimum in a stochastic sense.

The current study takes a different approach and explores the pertinence of re-optimizing with adapted initial conditions without adapting the model (NMPC) in the case of batch processes optimization with parametric errors. The main question asked is: “Given that the true optimum will not be reached due to plant-model parameter mismatch, is re-optimizing worthwhile? Will there be an improvement compared to simple implementation of the off-line optimal solution?” It is shown that NMPC re-optimization may deteriorate the performance, especially with an economic cost function. On the other hand, no such effect is present when the cost function is a squared error of the deviation from a desired trajectory feasible for the plant and the active constraints are invariant. In the absence of process noise, the tracking objective shows robustness and repeated optimization can be used even when the model is subject to small parametric errors. This paper, thus, highlights the difference in robustness between the economic and tracking objectives.

This paper first presents the basics of NMPC. Then, an analysis points out why re-optimizing without parameter adaption can give worse results. A demonstration showing that such situation does not arise for a quadratic tracking cost follows. Finally, an example is used to illustrate the different possible situations.

2. Problem Formulation—Model Predictive Control without Parameter Adaptation

Model Predictive control consists of repeatedly optimizing a given cost function based on a model of the system, using the state information obtained from the measurements. Two types of formulations are found in the literature—the receding horizon [20], typically used for continuous processes, and the

diminishing horizon [21], used for batch processes. In this paper, the diminishing horizon for a batch process with fixed final time t_f will be studied. Thus, at a given time t_k , the state obtained from the measurements is x_k , and the optimization problem is given as follows:

$$\begin{aligned} \min_{u[t_k, t_f]} J_k &= \phi(x(t_f)) + \int_{t_k}^{t_f} L(x(t), u(t)) dt \\ \text{s. t. } \dot{x} &= F(x(t), u(t), \theta) + \dot{v}, x(t_k) = x_k \\ S(x, u) &\leq 0 \\ T(x(t_f)) &\leq 0, \end{aligned} \quad (1)$$

where J is the function to minimize, u the input variable, x the states, F the equations describing the system dynamics, v the process noise, ϕ a function representing the terminal cost evaluated, L the integral cost function, θ the parameters and S and T respectively the path and terminal constraints. The initial conditions are obtained from the measured values, x_k . \dot{x} and \dot{v} represent, respectively, the differentiated states and noise.

The above formulation gets to the optimum, to the extent allowed by the sampling, when there is process noise but no parametric errors. The process noise would move the states away from the predicted value, but the repetition of the optimization assures that an optimum is found even from the deviated value.

Contrarily, this paper would consider the case where the functional form is assumed to be correct, but the parameters θ are unknown, so the error in the parameters $\Delta\theta = \theta - \theta_{real}$ is non-zero. This would also cause a variation in the states, but it might not be sufficient to simply optimize from the new states and the wrong parameters. Additionally, the excitation present in the system might not be sufficient to identify them online. In this work, the influence of such a parametric error on the operation of the NMPC would be studied.

3. Variational Analysis of Model Predictive Control without Parameter Adaptation

Let us consider an appropriate input and state parameterization (e.g., piecewise constant), where the parameterized input vector U and the parameterized states X will be used. Additionally, assume that the active constraints are invariant with respect to parametric variations and so become additional algebraic equations. These algebraic equations reduce the dimension of the search space. Let U_k represent the reduced vector of inputs from time t_k until t_f , and the states during this time interval is represented by X_k . The dynamic relationships can be written in a nonlinear static form and the dynamic optimization problem becomes the following static nonlinear programming problem:

$$\begin{aligned} \min_{U_k} J_k &= J(X_k, U_k, \theta) \\ X_k &= \Psi_k(U_k, \theta) + d_k, \end{aligned} \quad (2)$$

where d_k is the difference between the predicted and observed measurements, caused by process noise and parametric variations.

In what follows, variational analysis will be carried out assuming that the parametric variations are “small”. Thus, higher order terms will be neglected. Thus, the results obtained are valid for “small” parametric variations. In presence of parametric uncertainties and disturbances to the system, the variation equation ΔJ can then be written as a second order development:

$$\begin{aligned} \Delta J &= \frac{\partial J}{\partial U} \Delta U + \frac{\partial J}{\partial d} \Delta d + \frac{\partial J}{\partial \theta} \Delta \theta + \Delta U^T \frac{\partial^2 J}{\partial \theta \partial U} \Delta \theta + \Delta d^T \frac{\partial^2 J}{\partial \theta \partial d} \Delta \theta + \Delta U^T \frac{\partial^2 J}{\partial d \partial U} \Delta d + \frac{1}{2} \Delta U^T \frac{\partial^2 J}{\partial U^2} \Delta U \\ &\quad + \frac{1}{2} \Delta d^T \frac{\partial^2 J}{\partial d^2} \Delta d + \frac{1}{2} \Delta \theta^T \frac{\partial^2 J}{\partial \theta^2} \Delta \theta. \end{aligned} \quad (3)$$

In this equation, certain terms are constant since Δd and $\Delta \theta$ cannot be affected by manipulation on the process. Furthermore, the first term is zero by definition. Removing these terms and renaming the modifiable terms as $\widetilde{\Delta J}$, the equation becomes the following:

$$\min_{\Delta U} \widetilde{\Delta J} = \Delta U^T \frac{\partial^2 J}{\partial \theta \partial U} \Delta \theta + \Delta U^T \frac{\partial^2 J}{\partial d \partial U} \Delta d + \frac{1}{2} \Delta U^T \frac{\partial^2 J}{\partial U^2} \Delta U. \quad (4)$$

The necessary condition for the variational optimization can be obtained by differentiating it with respect to the input and equating to zero. The following equation is obtained:

$$\frac{\partial \widetilde{\Delta J}}{\partial U} = \frac{\partial^2 J}{\partial \theta \partial U} \Delta \theta + \frac{\partial^2 J}{\partial d \partial U} \Delta d + \frac{\partial^2 J}{\partial U^2} \Delta U = 0. \quad (5)$$

The optimal input can be calculated as:

$$\Delta U_{opt} = - \left(\frac{\partial^2 J}{\partial U^2} \right)^{-1} \left(\frac{\partial^2 J}{\partial \theta \partial U} \Delta \theta + \frac{\partial^2 J}{\partial d \partial U} \Delta d \right). \quad (6)$$

Define:

$$t_d = \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} \left(\frac{\partial^2 J}{\partial d \partial U} \right) \Delta d, \quad (7)$$

$$t_\theta = \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} \left(\frac{\partial^2 J}{\partial \theta \partial U} \right) \Delta \theta, \quad (8)$$

which are mathematical constructs that represent the parts of (6) that correspond to Δd and $\Delta \theta$ respectively. Under the standard assumption that the Hessian is positive definite, the square root exists. The units of t_θ and t_d are the same as $J^{-0.5}$ and so it is difficult to find a physical interpretation.

This paper considers the case where the parameters are not adapted principally due the absence of persistency of excitation. It is well known that the optimum cannot be reached in such a case. The following proposition goes one step further to show that it might even be harmful to re-optimize under certain circumstances.

Proposition 1. *Consider the repeated dynamic optimization problem (1) solved using the corresponding static nonlinear programming problem (2). Let the variations in the measured states be caused by both parametric variations and process noise. Furthermore, assume that the active constraints are invariant with respect to parametric variations. If the correction is only based on state measurements and the parameters are not adapted, then re-optimization will be worse than the offline solution when the terms t_θ and t_d point in opposing directions, satisfying $t_d^T t_\theta \leq -\frac{1}{2} t_d^T t_d$.*

Proof. If only Δd is measured and corrected, then:

$$\Delta U_{opt} = - \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} t_d \quad (9)$$

and:

$$\widetilde{\Delta J}_{opt} = -\frac{1}{2} t_d^T t_d - t_d^T t_\theta. \quad (10)$$

Obviously, $-\frac{1}{2} t_d^T t_d \leq 0$ while $-t_d^T t_\theta$ is sign indefinite. If t_d and t_θ point in the same direction, i.e., $\angle(t_d, t_\theta) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ rad, then $\widetilde{\Delta J}_{opt} \leq 0$. However, if they point on different directions, $\widetilde{\Delta J}_{opt}$ could still be

negative as long as the first term $(-\frac{1}{2}t_d^T t_d)$ dominates. Yet, if $t_d^T t_\theta \leq -\frac{1}{2}t_d^T t_d$, $\widetilde{\Delta J}_{\text{opt}}$ is positive, making the offline solution better than the re-optimization.

In the absence of process noise, $\Delta d = \frac{\partial \Psi}{\partial \theta} \Delta \theta$, and, thus, the terms t_θ and t_d can be written as:

$$t_d = \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} \left(\frac{\partial^2 J}{\partial d \partial U} \right) \frac{\partial \Psi}{\partial \theta} \Delta \theta, \quad (11)$$

$$t_\theta = \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} \left(\frac{\partial^2 J}{\partial \theta \partial U} \right) \Delta \theta. \quad (12)$$

Since for a general cost function (such as an economic objective) there is no relationship between these two terms, the result from Proposition 1 holds. However, it will be shown in the following proposition that, for a tracking cost with a trajectory feasible for the plant, re-optimization is beneficial even if the parameters are not adapted. This, in other words, expresses the inherent robustness associated with a tracking cost function.

Proposition 2. *Consider the repeated dynamic optimization problem (1) solved using the corresponding static nonlinear programming problem (2) with the tracking cost $J = \frac{1}{2}(X_k - X_{ref})^T (X_k - X_{ref}) + \frac{1}{2}w(U_k - U_{ref})^T (U_k - U_{ref})$, with X_{ref} being a trajectory feasible for the plant with U_{ref} being the corresponding input and w the weight for the input variations. Let the variations in the measured states be caused by parametric variations only. Furthermore, assume that the active constraints are invariant with respect to parametric variations. If the correction is only based on state measurements and the parameters are not adapted, then, for small enough parametric variations, $t_\theta = t_d$, and the re-optimization will be better or equal to the offline solution, i.e., $\widetilde{\Delta J}_{\text{opt}} \leq 0$.*

Proof. If the variation in the state is caused only by parametric uncertainties, then $\Delta d = \frac{\partial \Psi}{\partial \theta} \Delta \theta$. The partial derivatives for this case are given by:

$$\frac{\partial J}{\partial U} = (X - X_{ref})^T \frac{\partial \Psi}{\partial U} + w(U_k - U_{ref})^T, \quad (13)$$

$$\frac{\partial^2 J}{\partial \theta \partial U} = (X - X_{ref})^T \frac{\partial^2 \Psi}{\partial \theta \partial U} + \left(\frac{\partial \Psi}{\partial U} \right)^T \frac{\partial \Psi}{\partial \theta}, \quad (14)$$

$$\frac{\partial J}{\partial d} = (X - X_{ref})^T, \quad (15)$$

$$\frac{\partial^2 J}{\partial d \partial U} = \left(\frac{\partial \Psi}{\partial U} \right)^T. \quad (16)$$

With these, the two terms t_θ and t_d can be written as

$$t_d = \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} \left(\frac{\partial \Psi}{\partial U} \right)^T \frac{\partial \Psi}{\partial \theta} \Delta \theta, \quad (17)$$

$$t_\theta = \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} (X - X_{ref})^T \frac{\partial^2 \Psi}{\partial \theta \partial U} \Delta \theta + \left(\frac{\partial^2 J}{\partial U^2} \right)^{-\frac{1}{2}} \left(\frac{\partial \Psi}{\partial U} \right)^T \frac{\partial \Psi}{\partial \theta} \Delta \theta. \quad (18)$$

At the optimum, since X_{ref} is assumed to be feasible for the plant, $(X - X_{ref}) = 0$. Outside the optimum, $(X - X_{ref})$ grows with $\Delta \theta$ and the first term in Equation (18) becomes proportional to $\Delta \theta^2$. For small enough parametric variations, this term can be neglected. Then, t_θ and t_d are the same and this gives:

$$\widetilde{J}_{opt} = -\frac{1}{2}t_d^T t_d - t_d^T t_\theta = -\frac{3}{2}t_d^T t_d \leq 0. \quad (19)$$

In this case, the re-optimization is always better than the offline solution.

Such a robustness result cannot be established when process noise is present. Inclusion of process noise would cause $\Delta d = \frac{\partial \Psi}{\partial \theta} \Delta \theta + v$, which would lead to an additional term in Equation (17). This in turn prevents t_θ from being equal to t_d , which could eventually lead to a potential degradation in performance. Thus, robustness can only be established mathematically for a trajectory cost without process noise.

4. Results and Discussion

4.1. Illustrative Example

To illustrate the importance of parametric errors on NMPC, six different cases will be treated. The first three will be with economical cost, while the last three will have a trajectory to follow. In both situations, cases with terminal constraint, path constraint and no constraints will be done. Barrier functions will be used to treat the constraints.

For each case, a batch reactor with two reactions is studied (inspired from Reference [12]): $A \rightarrow B$ and $A + B \rightarrow C$. From a mass balance, the following model is derived for the system:

$$\begin{aligned} \dot{c}_A &= -k_1 c_A - k_2 c_A c_B, \\ \dot{c}_B &= k_1 c_A - k_2 c_A c_B, \end{aligned} \quad (20)$$

where c_X is the concentration of X (mol/L) and k_1 and k_2 are the kinetic reaction coefficient (h^{-1}), which are obtained using the Arrhenius equation:

$$k_i = k_{i0} \exp\left(-\frac{E_i}{RT}\right). \quad (21)$$

Using the following scaled temperature as the input parameter:

$$u = k_{10} \exp\left(-\frac{E_1}{RT}\right) \quad (22)$$

and considering:

$$\alpha = \frac{E_2}{E_1}, \bar{k}_{10} = 1 \text{ and } \bar{k}_{20} = k_{20} \left(\frac{1}{k_{10}}\right)^\alpha, \quad (23)$$

the kinetic coefficient are expressed as:

$$k_1 = \bar{k}_{10} u \text{ and } k_2 = \bar{k}_{20} u^\alpha. \quad (24)$$

The nominal values of all the parameters, as well as the constraint values, for these simulations are given in Table 1. For each case, the parameters with errors will be α and k_{10} .

Table 1. Models parameters, operating bounds, and initial conditions for Cases 1 to 6.

Parameter	Value	Units
c_{A0}	5	mol/L
c_{B0}	0	mol/L
k_{10}	5×10^3	h^{-1}
k_{20}	7×10^{16}	-
E_1	2×10^4	J/mol
E_2	1×10^5	J/mol

R	8.314	J/mol.K
α	5	-
\bar{k}_{10}	1	-
\bar{k}_{20}	0.0224	-
t_f	2	H
u_{min}	1.25	-
c_{Afmax}	0.1	mol/L
γ	0.001	-
β	0.999	-

Case 1: Unconstrained system with economical cost

The objective of the three first cases is to maximize the final concentration of B. In the first case, there are no constraints on the system, which gives the following optimization problem:

$$\max_{u(t)} J = c_B(t_f). \quad (25)$$

Case 2: System with terminal constraint and economical cost

The objective is to maximize the final concentration of B, in this case with a constraint on the final concentration of A:

$$\begin{aligned} \max_{u(t)} J &= c_B(t_f) \\ \text{s.t. } c_A(t_f) &\leq c_{Amax}. \end{aligned} \quad (26)$$

The optimization is subject to a terminal constraint on c_A . The terminal constraint is included in the numerical optimization using the following barrier function, where $b(c)$ is a barrier function for the constraint $x - x_{max} \leq 0$:

$$b(c) = \begin{cases} -\gamma \log(c - c_{max}), & c > \beta c_{min} \\ \frac{\gamma(c - \beta c_{max})}{(1 - \beta)c_{max}}, & c \leq \beta c_{min} \end{cases}. \quad (27)$$

Case 3: System with path constraint and economical cost

The objective is to maximize the final concentration of B, in this case with a lower bound on the input parameter:

$$\begin{aligned} \max_{u(t)} J &= c_B(t_f) \\ \text{s.t. } u &\geq u_{min}. \end{aligned} \quad (28)$$

The optimization is subject to a path constraint on u . The path constraint is included in the numerical optimization once again using a barrier function.

Case 4: Unconstrained system with trajectory cost

The objective of the three last cases is to minimize the difference between a trajectory and the concentration of B. In this case, there are no constraints on the system, which gives the following optimization problem:

$$\min_{u(t)} J = \int_{t_k}^{t_f} \left(c_B(t) - c_{Bref}(t) \right)^T \left(c_B(t) - c_{Bref}(t) \right) dt, \quad (29)$$

where $c_{B_{ref}}$ is the trajectory to follow. For the three tracking cases, $c_{B_{ref}}$ is a path following 90% of the maximal production (model optimum). Additionally, in those three cases, inputs are not penalized, mainly because no measurement noise was considered.

Case 5: System with terminal constraint and trajectory cost

The objective is to minimize the difference between a trajectory and the concentration of B, in this case with a constraint on the final concentration of A:

$$\begin{aligned} \min_{u(t)} J &= \int_{t_k}^{t_f} \left(c_B(t) - c_{B_{ref}}(t) \right)^T \left(c_B(t) - c_{B_{ref}}(t) \right) dt \\ \text{s. t. } c_A(t_f) &\leq c_{A_{max}}. \end{aligned} \quad (30)$$

The optimization is subject to a terminal constraint on c_A . The terminal constraint is included in the numerical optimization using a barrier function. $c_{B_{ref}}$ is the trajectory to follow and not a function of time.

Case 6: System with path constraint and trajectory cost

The objective is to minimize the difference between a trajectory and the concentration of B, in this case with a lower bound on the input parameter:

$$\begin{aligned} \min_{u(t)} J &= \int_{t_k}^{t_f} \left(c_B(t) - c_{B_{ref}}(t) \right)^T \left(c_B(t) - c_{B_{ref}}(t) \right) dt \\ \text{s. t. } u &\geq u_{min}. \end{aligned} \quad (31)$$

The optimization is subject to a path constraint on u . The path constraint is included in the numerical optimization once again using a barrier function. $c_{B_{set}}$ is the trajectory to follow and not a function of time.

4.2. Results

The terminal cost obtained for each simulation is shown in Table 2. The simulations in which the feedback re-optimization ended giving worse result than just using the offline optimization are indicated in bold. The parametric errors considered are all $\pm 20\%$ except for Case 3. This particular case was harder to optimize and a greater parametric error was required for the feedback's impact to surpass the optimization difficulties.

Table 2. Comparison of offline, re-optimization and plant optimum solutions for the six cases with parametric errors. Cost is maximized for Cases 1–3 and minimized for Cases 4–6.

Case	Parametric Error	Cost		
		Offline	Re-Optimization	Plant Optimum
1. Unconstrained, Economic cost	--	4.03	4.03	4.03
	\bar{k}_{10} : -20%; α : -20%	3.70	3.71	3.72
	\bar{k}_{10} : -20%; α : +20%	3.691	3.686	3.697
2. Terminal constraint Economic cost	--	3.71	3.71	3.71
	\bar{k}_{10} : -20%; α : -20%	1.29	2.03	3.35
	\bar{k}_{10} : -20%; α : +20%	3.01	2.93	3.06
3. Path constraints Economic cost	--	3.80	3.80	3.80
	\bar{k}_{10} : -50%; α : -50%	3.079	3.079	3.17
	\bar{k}_{10} : -50%; α : +50%	2.39	2.37	2.40
4. Unconstrained Trajectory cost	--	0.00	0.00	0.00
	\bar{k}_{10} : -20%; α : -20%	1.16	0.303	0.03

	$\bar{k}_{10}: -20\%; \alpha: +20\%$	1.10	0.30	0.00
5. Terminal constraint	--	0.00	0.00	0.00
Trajectory cost	$\bar{k}_{10}: -20\%; \alpha: -20\%$	2.84	1.49	0.03
	$\bar{k}_{10}: -20\%; \alpha: +20\%$	1.80	1.38	0.36
6. Path constraint	--	0.21	0.21	0.21
Trajectory cost	$\bar{k}_{10}: -20\%; \alpha: -20\%$	0.20	0.16	0.02
	$\bar{k}_{10}: -20\%; \alpha: +20\%$	2.060	1.48	0.47

Note that scenarios where re-optimization is worse than offline solution only occur with economical costs. Additional simulations have been made with different parametric errors, all leading to this same observation. All trajectory-tracking problems with a trajectory feasible for the plant always lead to the re-optimization being better. However, if a path more demanding than the maximal production was chosen for $c_{B_{set}}$, i.e., not a feasible trajectory, then the tracking problem suffers the same difficulties as the economical cost.

The simulation for a +20% error on α and -20% on \bar{k}_{10} in Case 1 is shown on Figure 1. It shows how re-optimization is actually worse than the offline solution. The figure clearly shows that the input is being pulled away from its optimal value with each re-optimization.

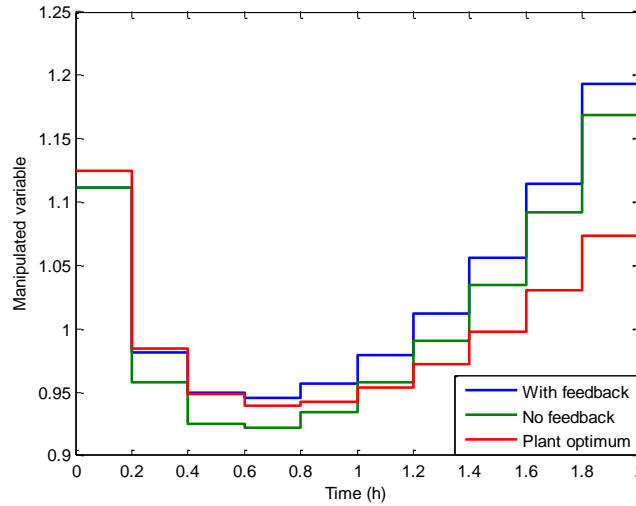


Figure 1. Input for Case 1, with perturbation in α and \bar{k}_{10} .

5. Conclusions

Optimization is frequently used on processes, whether it is offline or online in a control method, such as NMPC. In this paper, the impact of using NMPC in presence of parametric errors is studied. An analysis of the mathematical formulation of NMPC has shown that situations can occur where online optimization could lead to results worse than the offline one. The example studied presented this case in particular. It was seen that deterioration of the performance occurred only for an economical cost, while online optimization always helped with the tracking cost. A theoretical analysis has been performed and supports this result, showing that, for a quadratic tracking cost, online re-optimization will improve performance with small parametric uncertainties.

Acknowledgment: The authors would like to thank the NSERC (Natural Sciences and Engineering Research Council of Canada) for supporting their research.

Author Contributions: J.C.B. and B.S. conceived and designed the experiments; J.C.B. performed the simulations; J.C.B. and B.S. analyzed the data; J.C.B. and B.S. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MPC	Model predictive control
NMPC	Nonlinear MPC
D-RTO	Dynamic real-time optimization

References

1. Peters, N.; Guay, M.; DeHaan, D. Real-time dynamic optimization of batch systems. *J. Process Control* **2007**, *17*, 261–271.
2. Srinivasan, B.; Palanki, S.; Bonvin, D. Dynamic optimization of batch processes—I. Characterization of the nominal solution. *Comput. Chem. Eng.* **2003**, *27*, 1–26.
3. Bonvin, D.; Srinivasan, B.; Hunkeler, D. Control and optimization of batch processes—Improvement of process operation in the production of specialty chemicals. *IEEE Control Syst. Mag.* **2006**, *26*, 34–45.
4. De Souza, G.; Odloak, D.; Zanin, A.C. Real time optimization (RTO) with model predictive control (MPC). *Comput. Chem. Eng.* **2010**, *34*, 1999–2006.
5. Diehl, M.; Bock, H.G.; Schlöder, J.P.; Findeisen, R.; Nagy, Z.; Allgöwer, F. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J. Process Control* **2002**, *12*, 577–585.
6. Lu, J. Challenging control problems and emerging technologies in enterprise optimization. *Control Eng. Pract.* **2003**, *11*, 847–858.
7. Würth, L.; Hannemann, R.; Marquardt, W. Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization. *J. Process Control* **2009**, *19*, 1277–1288.
8. Aravind, P.; Valluvan, M.; Saranya, M. Simulation based modeling and implementation of adaptive control technique for Non Linear process tank. *Int. J. Comput. Appl.* **2013**, *68*, 16.
9. Dostal, P.; Bobal, V.; Vojtesek, J.; Chramcov, B. Adaptive control of nonlinear processes using two methods of parameter estimation. *WSEAS Trans. Syst.* **2014**, *13*, 292–301.
10. Paz Suárez, L.A.; Georgieva, P.; de Feye Azevedo, S. Nonlinear MPC for fed-batch multiple stages sugar crystallization. *Chem. Eng. Res. Des.* **2011**, *89*, 753–767.
11. Revollar, S.; Vega, P.; Vilanova, R. Economic optimization of wastewater treatment plants using Non Linear model predictive control. In Proceedings of the 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romania, 14–16 October 2015.
12. Kamaraju, V.K.; Chiu, M.S.; Srinivasan, B. Reformulating real-time optimal feedback based on model uncertainty. In Proceedings of the 10th IFAC Symposium on Dynamics and Control of Process Systems, Mumbai, India, 18–20 December 2013; pp. 756–761.
13. Yuan, C.; Wang, C. Persistency of excitation and performance of deterministic learning. In Proceedings of the 30th Chinese Control Conference, Yantai, China, 22–24 July 2011.
14. Pahija, E.; Manenti, F.; Mujtaba, I.M.; Rossi, F. Assessment of control techniques for the dynamic optimization of (semi-)batch reactors. *Comput. Chem. Eng.* **2014**, *66*, 269–275.
15. Jamaludin, M.Z.; Swartz, C.L.E. A bilevel programming formulation for dynamic real-time optimization. *IFAC-PapersOnLine* **2015**, *48*, 906–911.
16. Nagy, Z.K.; Braatz, R.D. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *J. Process Control* **2004**, *14*, 411–422.

17. Kadam, J.V.; Schlegel, M.; Srinivasan, B.; Bonvin, D.; Marquardt, W. Dynamic optimization in the presence of uncertainty: From off-line nominal solution to measurement-based implementation. *J. Process Control* **2007**, *17*, 389–398.
18. Lucia, S.; Finkler, T.; Engell, S. Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *J. Process Control* **2013**, *23*, 1306–1319.
19. Mesbah, A.; Streif, S.; Findeisen, R.; Braatz, R.D. Stochastic nonlinear model predictive control with probabilistic constraints. In Proceedings of the 2014 American Control Conference (ACC), Portland, OR, USA, 4–6 June 2014.
20. Morari, M.; Lee, J.H. Model predictive control: Past, present and future. *Comput. Chem. Eng.* **1999**, *23*, 667–682.
21. Nagy, Z.K.; Allgöwer, F. A nonlinear model predictive control approach for robust end-point property control of a thin-film deposition process. *Int. J. Robust Nonlinear Control* **2007**, *17*, 1600–1613.



© 2016 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).

ANNEXE B – ARTICLE 2 : ON THE VARIOUS LOCAL SOLUTIONS TO A TWO-INPUT DYNAMIC OPTIMIZATION PROBLEM

J.C. Binette^a, B. Srinivasan^a, D. Bonvin^b

^aDépartement de Génie Chimique, École Polytechnique Montréal, C.P.6079 Succ., Centre-Ville Montréal, Que., Canada H3C 3A7

^bÉcole Polytechnique Fédérale de Lausanne, EPFL – Station 9, CH-1015 Lausanne, Switzerland

Revue : Computer and Chemical Engineering, 95 (2016), 71–74

Abstract

Solving a multi-input dynamic optimization of a batch processes is a complex problem involving interactions between input variables and constraints over time. The problem gets more difficult due to the presence of local solutions that have almost the same cost but widely varying structures. This paper studies various local optimal solutions for a non-isothermal semi-batch reactor with the feed rate and temperature as inputs and a heat removal constraint. Three solution patterns were studied, all consisting in meeting the heat removal constraint for the first part and seeking the compromise between the main and side reactions in the later part. A sensitivity analysis shows that the best solution pattern among those studied does not change with variations in parameters or initial conditions.

1. Introduction

The solutions to multi-input dynamic optimization problems can be fairly intricate because of the interactions between the inputs and the constraints over time. There are many examples of this in the literature, most of which are problems with 2-7 inputs and 20-50 differential equations (Cervantes, Biegler, 1998; Schlegel, 2006). These problems are typically solved numerically (Kadam et al., 2007; Skogestad, 2000; Würth et al., 2011) and, as a result, there is no physical interpretation of the solution in terms of the types and sequence of arcs.

In contrast, Srinivasan et al. (2003) used an analytical approach to solve a simple two-input problem. The inputs are either constrained or sensitivity-seeking arcs. The solution presented in that paper corresponds to a local optimum with only three arcs. It illustrates well the analytical method and the interaction between the two inputs and the constraints. However, in recent years, the authors have been confronted with alternative numerical solutions that are “supposed to be better” than the one proposed in that paper. Some of the solutions are indeed better, while others give a better cost because they slightly violate the path constraints due to numerical inaccuracies.

This note compares these alternative optimal solutions by computing them *analytically*. The analytical approach has the advantage of not violating the constraints. Furthermore, a sensitivity analysis is presented to show whether the solution could easily change from one form to another when parameters change.

2. Problem formulation

The two-input optimization problem considered in Srinivasan et al. (2003) is briefly recalled. It consists of a non-isothermal semi-batch reactor with the series reactions $A + B \rightarrow C \rightarrow D$ and a heat-removal constraint. The reactor temperature and the feed rate of B are considered as manipulated variables. The system is described by the following equations:

$$\dot{c}_A = -k_1 c_A c_B - \frac{u}{V} c_A \quad (1)$$

$$\dot{c}_B = -k_1 c_A c_B + \frac{u}{V} (c_{Bin} - c_B) \quad (2)$$

$$\dot{c}_C = k_1 c_A c_B - k_2 c_C - \frac{u}{V} c_C \quad (3)$$

$$\dot{V} = u \quad (4)$$

where c_X is the concentration of the species X, c_{Bin} the concentration of B in the feed, k_i the kinetic parameter for the i^{th} reaction, u the feed rate of B, and V the volume of the reactor. The kinetic parameters follow Arrhenius law:

$$k_i = k_{i0} \exp\left(-\frac{E_i}{RT}\right)$$

where k_{i0} is the pre-exponential factor, E_i the activation energy, R the gas constant, and T the temperature.

The optimization problem maximizes the production of the desired product C:

$$\begin{aligned} \max_{u,T} \quad & J = c_C(t_f)V(t_f) \\ \text{s.t.} \quad & \text{Equation 1 – 4} \\ & T_{min} \leq T(t) \leq T_{max} \\ & u_{min} \leq u(t) \leq u_{max} \\ & V(t) \leq V_{max} \\ & (-\Delta H_1)k_1c_A(t)c_B(t)V(t) + (-\Delta H_2)k_2c_C(t)V(t) \leq q_{rx,max} \end{aligned} \tag{5}$$

Numerical values are given in Table 1.

Parameter	Value	Units
k_{10}	4	L/(mol h)
k_{20}	800	1/h
E_1	6e3	J/mol
E_2	20e3	J/mol
R	8.31	J/(mol K)
ΔH_1	-30e3	J/mol
ΔH_2	-10e3	J/mol
u_{min}	0	L/h
u_{max}	1	L/h
T_{min}	20	°C
T_{max}	50	°C
V_{max}	1.1	L
$q_{rx,max}$	1.5e5	W
c_{A0}	10	mol/L

c_{B0}	1.1685	mol/L
c_{C0}	0	mol/L
V_0	1	L
c_{Bin}	20	mol/L
t_f	0.5	h

Table 1: Parameters, bounds and initial conditions

The only possible options for the feed-rate input are u_{min} , u_{max} , and $u_{path}(t)$, that is, the feed rate is either at its bounds or determined by the heat-removal constraint. The analytical expression for $u_{path}(t)$ is provided in the original article:

$$u_{path} = V \frac{\Delta H_1 k_1^2 c_A c_B (c_A + c_B) + \Delta H_2 k_2 (k_1 c_A c_B - k_2 c_C)}{\Delta H_1 k_1 c_A (c_{Bin} - c_B)} - \frac{\dot{T}V}{RT} \frac{\Delta H_1 E_1 k_1 c_A c_B + \Delta H_2 E_2 k_2 c_C}{\Delta H_1 k_1 c_A (c_{Bin} - c_B)}$$

The possible options for the temperature input are T_{min} , T_{max} , $T_{path}(t)$, and $T_{sens}(t)$. This means that the temperature is either at its bounds, determined by the heat-removal constraint, or following a profile that expresses the compromise between the main reaction producing C and the side reaction consuming it. The temperature determined by the heat-removal constraint can be calculated from the previous equation:

$$\dot{T}_{path} = - \left(u - V \frac{\Delta H_1 k_1^2 c_A c_B (c_A + c_B) + \Delta H_2 k_2 (k_1 c_A c_B - k_2 c_C)}{\Delta H_1 k_1 c_A (c_{Bin} - c_B)} \right) \frac{RT}{V} \frac{\Delta H_1 k_1 c_A (c_{Bin} - c_B)}{\Delta H_1 E_1 k_1 c_A c_B + \Delta H_2 E_2 k_2 c_C}$$

The analytical expression for the sensitivity-seeking arc is provided in the original article:

$$\dot{T}_{sens} = \frac{-RT^2 k_1 c_A c_B}{E_2 c_C} - \frac{RT^2 (c_{Bin} - c_B) u}{c_B (E_1 - E_2) V}$$

Finally, note that the parameters have been adjusted so that the heat-removal constraint is active initially for $T(0) = T_{max}$. Otherwise, additional arc(s) might be necessary to reach $q_{rx,max}$ as quickly as possible.

3. Three local solutions to the problem

The analytical solution method is used in this article, where a sequence of arcs is determined by physical reasoning along with mathematical analysis of the optimization problem. The switching times are considered as free parameters obtained using numerical optimization. The main advantage of the analytical approach is that there is no constraint violation caused by sampling and the optimization problem is numerically inexpensive. Each arc, by construction, satisfies the necessary conditions of optimality within the specified time interval. However, the choice of the sequence of arcs is crucial to satisfy the necessary conditions at initial and final time. So, an additional step of verifying the necessary/sufficient conditions is required to validate the solution and guarantee that it is at least a local optimum.

All investigated sequence of arcs meet the heat-removal constraint for the first part of the batch and stop the feed once the maximum volume has been reached. As explained in the original article, without constraint, the optimal solution would simply consist in initially filling up the reactor with B and following a sensitivity-seeking temperature profile. The three investigated solutions are described next and the types and sequence of arcs are summarized in Table 2. They all correspond to increasing the time interval during which the heat removal constraint is active.

Solution 1: Three arcs, start with T_{max} . In the solution presented in the original paper, the heat-removal constraint is met initially by adjusting the feed rate $u_{path}(t)$. The temperature is set initially at T_{max} and then switch to the sensitivity-seeking arc $T_{sens}(t)$ at the switching time t_1 . Note that the temperature $T_{sens}(t)$ is still used after the maximum volume has been reached at time t_2 . Upon knowledge of the sequence of arcs and the aforementioned expressions for $u_{path}(t)$ and $T_{sens}(t)$, this problem has a single decision variable, namely, the switching time t_1 .

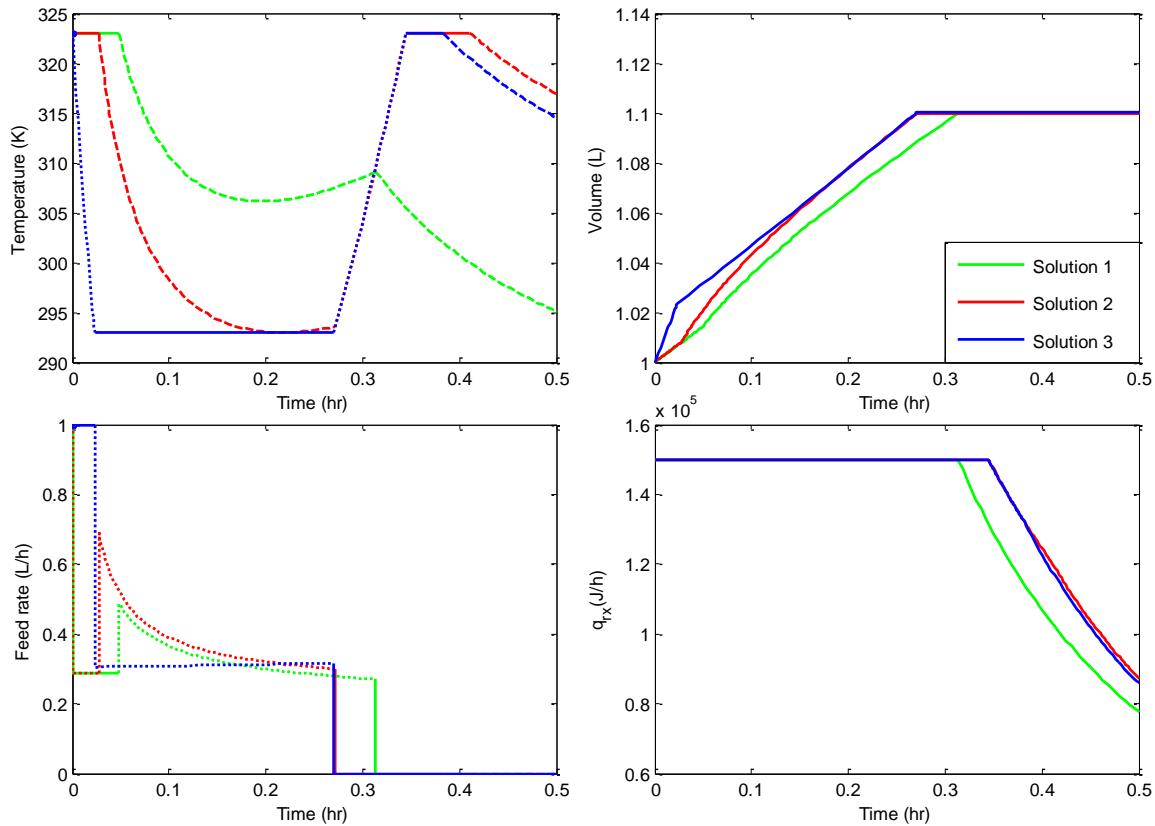
Solution 2: Five arcs, start with T_{max} . The first variant analyzed in this note consists in staying on the heat-removal constraint a little longer by also using the temperature to meet the path constraint. The option of using $T_{path}(t)$ was not explored in the original paper. The temperature is increased to follow the constraint and, once it hits T_{max} , it stays there until the numerical optimization decides that it is better to change to a sensitivity-seeking arc. Eventually, the temperature follows a sensitivity-seeking arc to reduce the consumption of the desired product. This gives the first 5-arc solution described in Table 2 with two decision variables, namely, the switching time t_1 between the first and second arcs, and the switching time t_4 between the fourth and fifth arcs.

Solution 3: Five arcs, start with u_{max} . The second variant consists in using u_{max} initially and thus the temperature $T_{path}(t)$ to follow the heat-removal constraint. The temperature reaches T_{min} quickly, and from there on, it becomes necessary to use the feed rate $u_{path}(t)$ to satisfy the heat-removal constraint. Once the volume attains its maximum, the same logic as above is used for the last three arcs to give the second 5-arc solution described in Table 2, with only one decision variable, namely, the switching time t_4 between the fourth and the fifth arcs.

Solution	Inputs	t_0 to t_1	t_1 to t_2	t_2 to t_3	t_3 to t_4	t_4 to t_f
Solution 1 3 arcs, T_{max}	u	u_{path}	u_{path}	u_{min}	--	
	T	T_{max}	T_{sens}	T_{sens}		
Solution 2 5 arcs, T_{max}	u	u_{path}	u_{path}	u_{min}	u_{min}	u_{min}
	T	T_{max}	T_{sens}	T_{path}	T_{max}	T_{sens}
Solution 3 5 arcs, u_{max}	u	u_{max}	u_{path}	u_{min}	u_{min}	u_{min}
	T	T_{path}	T_{min}	T_{path}	T_{max}	T_{sens}

Table 2: Types and sequence of arcs for the three considered solutions

Figure 1 presents the input trajectories for the three solutions corresponding to the nominal case that uses the parameters given in Table 1. It also presents the heat evolution $q_{rx}(t)$, where it can be seen that the system follows this constraint longer in the two 5-arc solutions, which results in a higher production of the desired product. All simulations were performed using MATLAB.



Figure

1: Temperature, feed-rate, volume, and heat-removal profiles for the nominal case

Constraint changes	Amount of C produced (mol)		
	Solution 1 3 arcs, T_{max}	Solution 2 5 arcs, T_{max}	Solution 3 5 arcs, u_{max}
--	2.0158	2.0499	2.0529
$q_{rx,max} + 15\%$	2.0485	2.1131	2.1164
$T_{max} + 2\%$	2.0155	2.0496	2.0529
$\Delta V_{max} + 5\%$	2.0528	2.0882	2.0914

Table 3: Production of C for the 3 solutions upon considering constraint changes

4. Sensitivity of the solutions

This section investigates the effect of constraint variations on the three solutions. The types and sequence of arcs get modified slightly for the two reasons explained below. The various cases are presented in Table 3 and the corresponding time profiles in Figure 2.

1. Recall that the nominal parameters in Table 1 were chosen such that, with the initial temperature T_{max} , optimal operation starts directly on the heat-removal constraint. However, with constraint variations, this is no longer the case. If the produced heat q_{rx} is less than the constraint value, an additional initial arc with $u = u_{max}$ and $T = T_{max}$ needs to be added (see Fig 2, $q_{rx,max} + 15\%$). On the other hand, if the heat generated by the reaction at $T = T_{max}$ is more than $q_{rx,max}$, the system needs to start with $T(0) < T_{max}$ in order to satisfy the heat-removal constraint. In Solution 3, the system could start with $u_{max} - T_{path}$ (see Fig 2, $T_{max} + 2\%$). However, Solutions 1 and 2 require an additional arc $u_{min} - T_{path}$ in order to get to T_{max} (see Fig 2, $T_{max} + 2\%$).
2. Depending on the importance of the side reaction, it might be necessary to start the last $u_{min} - T_{sens}$ arc much earlier, thereby completely eliminating the $u_{min} - T_{max}$ arc. In this case, Solutions 2 and 3 would have only 4 arcs (see Fig 2, $T_{max} + 2\%$).

Variations of parameters and initial conditions were also performed as follows: k_{10} (-10%), k_{20} , E_1 (-2%), E_2 (-5%), $q_{rx,max}$ (+10%, +15%), T_{max} (+2%), T_{min} , ΔV_{max} (+5%), C_{A0} (-10%, +10%), C_{B0} (-10%, +10%), C_{C0} , C_{Bin} , U_{max} and, V_0 (-3%, +4%). In all cases, Solution 3 was better than Solution 2, which was in turn better than the initial 3-arc solution. This shows, in some sense, the robustness of the input structure with respect to parametric variations. Three of these cases are presented in Table 4.

Parametric changes	Amount of C produced (mol)		
	Solution 1 3 arcs, T_{max}	Solution 2 5 arcs, T_{max}	Solution 3 5 arcs, u_{max}
C_{A0} : -10%	1.9425	1.9768	1.9807
V_0 : +4%	1.6751	1.6932	1.6954
E_1 : -2%	2.0417	2.0729	2.0762

Table 4: Production of C for the 3 solutions upon considering parametric changes

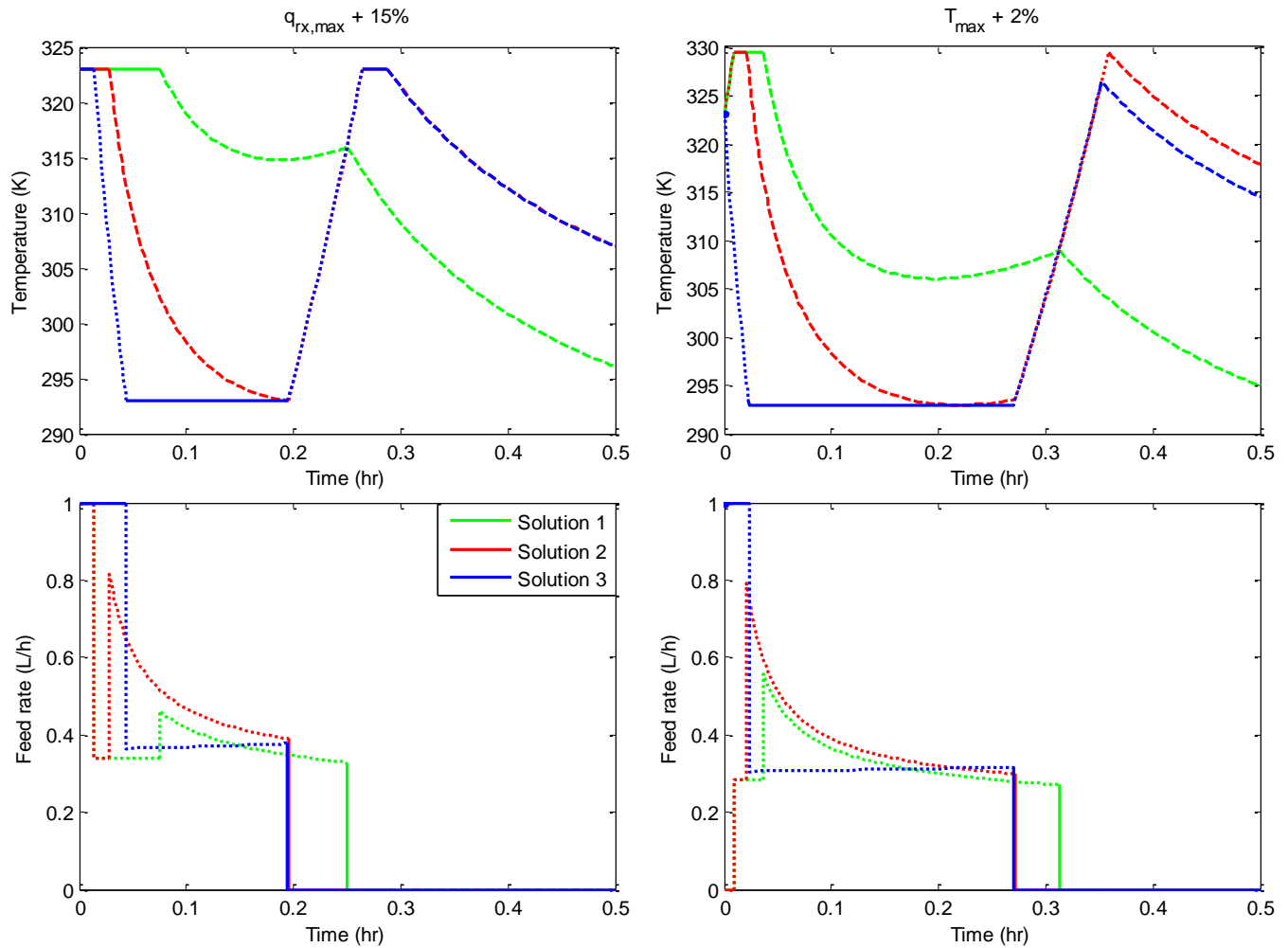


Figure 2: Temperature and feed-rate profiles upon parameter variations

5. Conclusion

Nonlinear dynamic optimization problems might have many local solutions. In this note, local solutions involving different types and sequences of arcs were studied. The analytical approach was favored as it guarantees satisfaction of all constraints, and it allows including intuition about the system in order to find solutions that are physically meaningful. Some of the structures provided better solutions than others. Although it cannot be readily generalized, the types and sequence of arcs seem to be robust with respect to minor changes in parameters, constraints and initial conditions.

Acknowledgement

The first two authors would like to thank the NSERC for supporting their research.

References

- 1- Cervantes A, Biegler LT. Large-scale DAE optimization using a simultaneous NLP formulation. *AIChE J.*, 44 (5), 1038–1050 (1998).
- 2- Schlegel M, Marquardt W. Adaptive switching structure selection for the solution of dynamic optimization problems. *Ind. Eng. Chem. Res.*, 45 (24), 8083–8094 (2006).
- 3- Kadam JV, Schlegel M, Srinivasan B, Bonvin D, Marquardt W. Dynamic optimization in the presence of uncertainty: From off-line nominal solution to measurement-based implementation. *J. of Process Control*, 17, 389–398 (2007).
- 4- Skogestad S. Plantwide control: The search for the self-optimizing control structure. *J. of Process Control*, 10, 487-507 (2000).
- 5- Würth L, Hannemann R, Marquardt W. A two-layer architecture for economically optimal process control and operation. *J. of Process Control*, 21, 311–321 (2011).
- 6- Srinivasan B, Palanki S, Bonvin D. Dynamic optimization of batch processes I. Characterization of the nominal solution. *Comput. and Chem. Eng.*, 27, 1-26 (2003).

ANNEXE C – CODES MATLAB ÉCRITS POUR RÉALISER LES DIFFÉRENTES SIMULATIONS

Exemple de code pour les simulations (réacteur à deux réactions)

Pour chaque système, ce code a été utilisé, avec les modifications appropriées.

```
% Sélection du type de contrainte
type = '6'; % 1 et 4: finale, 2 et 5: trajectoire, 3 et 6: sans contrainte

% Erreur sur les paramètres du modèle [delta-k1_0bar delta-k2_0bar delta-alpha]
delta_système = [-0.2 0.5 -0.2]; % Le modèle est constant et connu, l'erreur est sur le système
delta_modele = [0 0 0];

% Définition de période de temps modélisée
dTsamp = 0.2; % h
Tend = 2; % h
Tsamp = 0:dTsamp:Tend; % h

% Consigne (La consigne est charge par le programme)
load consigneopt09

% Options pour l'optimisation
options = optimset('Display','Iter','TolFun',1e-9,'MaxFunEvals',5000,'MaxIter',2000);
intoptions = odeset('RelTol',1e-6);

% Conditions initiales
Ca0 = 5; % mol/L
Cb0 = 0; % mol/L
x0 = [Ca0,Cb0];
x_ini = x0;

% La valeur initiale pour l'optimisation est charge par le programme
load u_set09_no_peak
u_act0 = u_new;

% Initialisation des vecteurs états pour les trois cas (nominal, avec réoptimisation et sans réoptimisation)
x_nom = x0;
```

```

x_ar          = x0;
x_sr          = x0;
% Horizon de commande
N              = length(Tsamp);

% Optimisation sans erreur
[u_new,FVAL]   =
fminunc('cost_batch',u_act0,options,x_ini,0,Tsamp(end),dTsamp,N+1,delta_systeme,type,Cb_set);
u_nom         = u_new;

%% Simulation avec le u optimal sans erreur
for k = 1:length(Tsamp)-1
    Tspan      = [(k-1)*dTsamp,k*dTsamp];

    [~,x]      = ode45('systeme',Tspan,x_ini,intoptions,u_new(k),delta_systeme);

    x_nom(k+1,:) = x(end,:);
    x_ini       = x(end,:);
end

% Réinitialisation de Ca_ini pour la boucle avec erreur
x_ini         = x0;

% Simulation avec erreur
for j = 1:length(Tsamp)-1

    % Optimisation pour trouver u
    [u_new,FVAL] = fminunc('cost_batch',u_act0,options,x_ini,Tsamp(j), Tsamp(end),dTsamp,N-j,
delta_modele,type,Cb_set);

    % Prochaine valeur initiale
    u_act0      = u_new(2:end);
    u_ar(j,1)   = u_new(1);

    % Modélisation sans réoptimisation
    if(j==1)
        x1      = x0;
        u_sr     = u_new;
    end
end

```

```

for k = 1:length(Tsamp)-1
    Tspan = [(k-1)*dTsamp,k*dTsamp];

    [~,x] = ode45('systeme',Tspan,x1,[],u_new(k),delta_systeme);

    x_sr(k+1,:) = x(end,:);
    x1 = x(end,:);
end
end
u_sr_act(:,j) = [u_ar;u_act0];
cost_sr(j,1) = cost_batch(u_sr_act(:,j) ,x0,0,Tend,dTsamp,N,delta_systeme,type,Cb_set);

Tspan = [(j-1)*dTsamp,j*dTsamp];

[t,x] = ode45('systeme',Tspan,x_ini,[],u_new(1),delta_systeme);

x_ar(j+1,:) = x(end,:);
x_ini = x(end,:);
end

%%
cost_nom = cost_batch(u_nom,x0,0,Tend,dTsamp,N,delta_systeme,type,Cb_set);
cost_ar = cost_batch(u_ar ,x0,0,Tend,dTsamp,N,delta_systeme,type,Cb_set);

```

Exemple de fonctions coût

```
function [ cost ] = cost_batch( uact, x0, Tini, Tend, dTsamp, N, delta, type, Cb_set)

% Définition de période de temps modélisée
Tsamp      = Tini:dTsamp:Tend;
intoptions = odeset('RelTol',1e-5);

x_ini      = x0;

% Modélisation
for k = 1:length(Tsamp)-1

    % Définition de la période de temps pour l'itération en cours
    Tspan    = [(k-1)*dTsamp,k*dTsamp];

    [~,x]     = ode45('systeme',Tspan,x_ini,intoptions,uact(k),delta);

    Ca(k+1,1) = x(end,1);
    Cb(k+1,1) = x(end,2);
    x_ini     = x(end,:);
end

Caf      = Ca(end);
Cbf      = Cb(end);

% Contrainte finale, coût économique (fonction barrière)
if(type == '1')
    Cafmax = 0.1;
    alpha  = 0.001;
    beta    = 0.999;

    if(Caf < beta*Cafmax)
        cost = -Cbf - alpha*log(Cafmax - Caf);
    else
        cost = -Cbf + 1/(1-beta)/Cafmax*alpha*(Caf-beta*Cafmax);
    end
end
```

```

% Contrainte de chemin, coût économique (fonction barrière)
if(type == '2')
    u_min = 1.25;
    alpha      = 0.0001;
    beta       = 0.9999;
    cost = -Cb_f;
    for k=1:length(Tsamp)-1
        if(uact(k) > beta*u_min)
            cost = cost - alpha*log(uact(k) - u_min);
        else
            cost = cost + 1/(1-beta)/u_min*alpha*(-uact(k)+beta*u_min);
        end
    end
end

% Sans contrainte, coût économique (fonction barrière)
if(type == '3')
    cost = -Cb_f;
end

% Contrainte finale, coût de tracking (fonction barrière)
if(type == '4')
    cost = norm(Cb(2:end) - Cb_set(length(Cb_set)-length(Cb)+2:end));

    Cafmax = 0.1;
    alpha   = 0.0001;
    beta    = 0.9999;

    if(Caf < beta*Cafmax)
        cost = cost - alpha*log(Cafmax - Caf);
    else
        cost = cost + 1/(1-beta)/Cafmax*alpha*(Caf-beta*Cafmax);
    end
end

% Contrainte de chemin, coût de tracking (fonction barrière)
if(type == '5')
    cost = norm(Cb(2:end) - Cb_set(length(Cb_set)-length(Cb)+2:end));

    u_min = 1.25;

```

```

alpha          = 0.0001;
beta           = 0.9999;

for k=1:length(Tsamp)-1
    if(uact(k) > beta*u_min)
        cost      = cost - alpha*log(uact(k) - u_min);
    else
        cost      = cost + 1/(1-beta)/u_min*alpha*(-uact(k)+beta*u_min);
    end
end
end

% Sans contrainte, coût de tracking (fonction barrière)
if(type == '6')
    cost          = norm(Cb(2:end) - Cb_set(length(Cb_set)-length(Cb)+2:end));
end

end

```


Modèles des systèmes

Réacteur à deux réactions

```
function dxdt=systeme(t,x,dummy,u,delta)

% États
Ca=x(1);
Cb=x(2);

% Paramètres
Ca0      = 5;                % mol/L
k1_0     = 5e3;              % h^-1
k2_0     = 7e16;             % h^-1
E1       = 2e4;              % J/mol
E2       = 1e5;              % J/mol
R        = 8.314;            % J/mol/K

alpha    = (E2/E1)           * (1+delta(3)); % Adimensionnel
k1_0bar  = 1                 * (1+delta(1)); % Adimensionnel
k2_0bar  = (k2_0 * (1/k1_0)^(E2/E1)) * (1+delta(2)); % Adimensionnel

k1       = k1_0bar * u;      % h^-1
k2       = k2_0bar * u^alpha; % h^-1

% Équation différentielle du système
dCadtdt = -k1 * Ca - k2 * Ca*Cb; % mol/L/h
dCbdt    =  k1 * Ca - k2 * Ca*Cb;

dxdt     = [dCadtdt; dCbdt];

end
```

Réacteur à réactions parallèles

```
function dxdt=systeme(t,x,dummy,u,delta)

% États
Ca      = x(1);    % mol/L
Cb      = x(2);    % mol/L
V       = x(3);    % L

% Paramètres
k1      = 0.053;    % L/(mol.min)
k2      = 0.128;    % L/(mol.min)
Cbin    = 5;        % mol/L

% Introduction de l'erreur sur k et Ca0
k1      = k1*(1 + delta(1));
k2      = k2*(1 + delta(2));

% Équation différentielle du système
```

```

dCadt = -k1*Ca*Cb - u/V *Ca;
dCbdt = -k1*Ca*Cb - 2*k2*Cb^2 + u/V*(Cbin-Cb);
dVdt = u;

dxdt = [dCadt;dCbdt;dVdt];
end

```

Réacteur A-B sans contrainte

```

function dxdt=systeme(t,x,dummy,u,delta)

% États
Ca=x(1);

% Paramètres
Ca0 = 5; % mol/L
k1_0 = 5e3; % h^-1
k2_0 = 7e16; % h^-1
E1 = 2e4; % J/mol
E2 = 1e5; % J/mol
R = 8.314; % J/mol/K

alpha = (E2/E1) * (1+delta(3)); % Adimensionnel
k1_0bar = 1 * (1+delta(1)); % Adimensionnel
k2_0bar = (k2_0 * (1/k1_0)^(E2/E1)) * (1+delta(2)); % Adimensionnel

k1 = k1_0bar * u; % h^-1
k2 = k2_0bar * u^alpha; % h^-1

% Équation différentielle du système
dCadt = -k1 * Ca + k2 * (Ca0-Ca); % mol/L/h

dxdt = dCadt;

end

```

Réacteur avec contrainte de sécurité

```

function dxdt=systeme(t,x,dummy,u,delta)

% États
na = x(1); % mol
nb = x(2); % mol
nc = x(3); % mol
V = x(4); % L

% Paramètres
k = 0.0482; % L/(mol.h)
Cbin = 2; % mol/L

% Introduction de l'erreur sur k et Ca0

```

```

k      = k      *(1 + delta(2));
Cbin   = Cbin   *(1+delta(1));

Ca      = na/V;    %mol/L
Cb      = nb/V;    %mol/L

% Équation différentielle du système
dnadt   = -k*Ca*Cb*V;
dnbdt   = -k*Ca*Cb*V + u*Cbin;
dncdt   = k*Ca*Cb*V;
dVdt    = u;

dxdt    = [dnadt;dnbdt;dncdt;dVdt];
end

```

Réacteur semi-batch non-isotherme avec réaction en série contrainte de chaleur

```

function [dxdt,u] = systeme(t,x,utype,Ttype,delta)

% États
Ca=x(1);
Cb=x(2);
Cc=x(3);
V =x(4);
T =x(5);

Cbin    = 20      *(1+delta(7));

k1_0    = 4       *(1+delta(1));           % h^-1
k2_0    = 800     *(1+delta(2));           % h^-1
E1      = 6e3     *(1+delta(3));           % J/mol
E2      = 20e3    *(1+delta(4));           % J/mol
R       = 8.31;      % J/mol/K

deltaH1 = -3e4;
deltaH2 = -1e4;

k1      = k1_0*exp(-E1/R/T);
k2      = k2_0*exp(-E2/R/T);

u_min   = 0;
u_max   = 1      *(1+delta(8));

Tmax    = (50+273)*(1+delta(14));
Tmin    = (20+273)*(1+delta(5));

% u_path = A-dTdt*B; dTsensdt = C-u*D;
% où
A       = V*(((-deltaH1)*k1^2*Ca*Cb*(Ca+Cb) - (-deltaH2)*k2*(k1*Ca*Cb -
k2*Cc))/((-deltaH1)*k1*Ca*(Cbin-Cb)));
B       = V/(R*T^2)*(((-deltaH1)*E1*k1*Ca*Cb + (-deltaH2)*E2*k2*Cc)/((-
deltaH1)*k1*Ca*(Cbin-Cb)));

```

```

C      = R*T^2*k1*Ca*Cb / (E2*Cc);
D      = R*T^2*(Cbin-Cb) / (Cb*(E1-E2)*V);

switch utype
case 'min'
    u = u_min;
    switch Ttype
    case 'min'
        dTdt = 0;
    case 'max'
        dTdt = 0;
    case 'path'
        dTdt = -(u-A)/B;
    case 'sens'
        dTdt = -C - u*D;
    end
case 'max'
    u = u_max;
    switch Ttype
    case 'min'
        dTdt = 0;
    case 'max'
        dTdt = 0;
    case 'path'
        dTdt = -(u-A)/B;
    case 'sens'
        dTdt = -C - u*D;
    end
case 'path'
    switch Ttype
    case 'min'
        dTdt = 0;
        u = A - dTdt*B;
    case 'max'
        dTdt = 0;
        u = A - dTdt*B;
    case 'sens'
        u = (A+C*B) / (1-D*B);
        u = max(min(u,u_max),u_min);
        dTdt = -C - u*D;
    end
end

% Équation différentielle du système
dCadT = -k1*Ca*Cb - u*Ca/V; % mol/L/h
dCbdt = -k1*Ca*Cb + u*(Cbin-Cb)/V;
dCcDt = k1*Ca*Cb - k2*Cc - u*Cc/V;
dVdt = u;

dxdt = [dCadT; dCbdt; dCcDt; dVdt; dTdt];

end

```