N° d'ordre : 7966

# UFR SCIENTIFIQUE D'ORSAY

# THESE

## Présentée pour obtenir

## Le GRADE de DOCTEUR EN SCIENCES DE L'UNIVERSITE PARIS XI ORSAY

## Par


## Sorin Olaru


**Sujet :** CONTRIBUTION A L'ETUDE DE LA COMMANDE PREDICTIVE SOUS CONTRAINTES PAR APPROCHE GEOMETRIQUE

**Soutenue le 23 Septembre 2005 devant la Commission d'examen :**

| | |
|---|---|
| **M. Dominique BONVIN** | |
| **M. Patrick BOUCHER** | |
| **M. Luc DUGARD** | **Rapporteur** |
| **M. Didier DUMUR** | **Directeur de thèse** |
| **M Eric WALTER** | **Président** |
| **M. Vincent WERTZ** | **Rapporteur** |

*Pentru Mara,*
*… familiei mele*
*… memoriei bunicului meu, Leon Popescu*

**MOTTO: "... là où la loi les approuve tout simplement,**
**c'est par paresse d'esprit de ses auteurs;**
**mais chez nous la loi repose sur des raisons**
**plus belles et, comme je le disais, délicates à**
**débrouiller." PLATON, *Le banquet*.**

# Remerciements

Le travail lié à une thèse de doctorat peut être perçu de l'extérieur comme une construction individuelle ne nécessitant par conséquent que des ressources internes. Mon expérience me permet de témoigner du contraire, de sorte que mes trois années de thèse se sont effectuées avant tout grâce au soutien, à l'assistance ou tout simplement à la compagnie de plusieurs personnes auxquelles je dois ma reconnaissance.

Je pense tout d'abord à la personne qui a suivi tous mes travaux, en détail, avec une compétence et patience remarquables, mon directeur de thèse Didier Dumur, HDR et Professeur à Supélec. J'ai souvent mis sa disponibilité à l'épreuve, nos discussions très enrichissantes couvrant des thèmes scientifiques bien loin des limites de ce mémoire. Je lui suis reconnaissant de l'esprit synthétique et du professionnalisme qu'il a manifestés à l'égard de mes travaux et en général de la confiance qu'il a su m'inspirer, non seulement pendant la thèse, mais également lors de ces cinq dernières années.

Je remercie Monsieur Patrick Boucher, Professeur et Chef du Département Automatique de Supélec, de m'avoir accueilli au sein de ce laboratoire et d'avoir fait en sorte que la thèse se développe dans un cadre optimal. Au delà de ces aspects, je lui suis reconnaissant de la sympathie, de la disponibilité et de la confiance qu'il a su m'accorder à certains moments déterminants pour mon parcours scientifique. Je suis honoré de l'intérêt qu'il a toujours manifesté pour mes travaux ainsi que de sa présence dans mon jury de thèse.

Je n'oublie pas les personnes qui ont accepté de faire partie de mon jury de thèse : Monsieur Eric Walter, Directeur de recherche au CNRS, qui m'a fait l'honneur de présider ce jury, Monsieur Dominique Bonvin, Professeur à l'EPFL qui m'a honoré de sa présence. Je tiens à remercier tout particulièrement, pour le temps précieux qu'ils ont consacré à me relire, mes rapporteurs Monsieur Luc Dugard, Directeur de recherche au CNRS et Monsieur Vincent Wertz, Professeur à l'UCL, qui ont ainsi contribué, par leur attention, leurs remarques et interrogations, à l'amélioration de la qualité de mon mémoire de thèse. Qu'ils trouvent ici ma gratitude.

Je n'aurais peut-être jamais commencé une thèse si, tout au long de ma formation et bien au-delà, je n'avais eu la chance de me retrouver sous le parrainage du Professeur Ioan Dumitrache, qui a toujours su m'inciter à donner le meilleur de moi-même lors de mes activités de recherche. Je le remercie de toute l'attention qu'il a manifestée pour mon parcours.

Cette thèse a été une suite naturelle des deux stages de recherche proposés en coopération entre l'Université « Politehnica » de Bucarest et Supélec, je souhaite donc remercier les deux artisans de cet accord, les Professeurs Constantin Iliescu et Yves Tanguy.

Tout au long de ces trois années de thèse, mes travaux ont été également soutenus par la Direction de la Recherche et du Développement de la Commission Européenne, dans le prolongement du Prix Archimède obtenu en 2002. Ce support a représenté une base matérielle précieuse m'assurant une mobilité, pour laquelle je suis reconnaissant. Ce prix m'a montré par ailleurs que la recherche n'est pas ingrate, me donnant un passeport qui m'a permis de découvrir de très nombreuses figures scientifiques. Je remercie donc ici l'organisation, mais aussi les personnes, qui m'ont aidé directement à gérer le contrat relatif à ce prix, Christine Huguet et Geneviève Wilmont.

Pendant ces trois années, j'ai été entouré par des personnes qui ont manifesté un intérêt tout particulier pour mes travaux, dont les critiques se sont révélées très précieuses. La liste est longue mais je tiens à remercier plus spécialement Marius Zainea, Jean Thomas, Pedro Rodriguez, Simona Orosanu, mais aussi le groupe d'Automatique de Supélec Rennes.

La vie dans un laboratoire se partage avec d'autres stagiaires et doctorants, qui effectuent un séjour plus ou moins long. J'y ai pour ma part passé trois années, dont deux dans « le mythique » bureau C.3.15 (Assia, Marta, Bastien, Luca, Bilal, Nico peuvent confirmer) et j'ai vécu des moments mémorables à leur coté, ainsi qu'en la compagnie de Karla, Guyomar, Hulya, Cristina, Madalina, Loredana, Edna, Sébastien, Guillaume, Mohamed, Bouba, Christophe, Mihai, Bogdan, Thibault, Omar.

Je voudrais également remercier l'ensemble des professeurs du Département : Houria Siguerdidjane, Sihem Tebbani, Emmanuel Godoy, Gilles Duc, Martial Demerlé, Stéphane Font, Dominique Beauvois, et bien sûr Josiane Dartron pour toute sa bonne humeur, ainsi que Léon Marquet pour son efficacité lors de la résolution de mes problèmes techniques.

Effectuer une thèse loin de la société dans laquelle on a été formé est toujours un défi. Pendant cette épreuve, j'ai reçu heureusement de solides soutiens, et je remercie Farag, Joel, Luca, Bastien, Alex, Marius, Andreea, Tudor, Stefan, Gabi, Ana, Mugurel, Radu, Cristi, Marcelo, Gino.

Il sera difficile des trouver les mots pour remercier Mara, qui a pu me supporter, a su m'encourager et m'aider, jour après jour et au delà de ses travaux de thèse. Qu'elle trouve ici toute ma gratitude et mon amour.

Enfin, je veux rendre hommage à la mémoire de deux personnes : le Professeur Vlad Ionescu, qui a incarné la vocation pour une carrière académique et à qui je dois la découverte de l'Automatique ; mon grand-père, Leon Popescu, dont la qualité humaine reste un repère que j'essaierai de suivre.

# Avant-propos

Le travail présenté dans cette thèse a donné lieu à la publication d'un certain nombre d'articles dans des revues et à l'occasion de différents congrès internationaux avec actes.

**Articles de revues**

- S. OLARU, D. DUMUR, "Avoiding Constraints Redundancy in Predictive Control Optimization Routines", *IEEE Transaction on Automatic Control*, Vol. 50, no. 9, pp. 1459-1466, 2005.
- S. OLARU, D. DUMUR, "Feasibility of constrained generalized predictive control within invariant sets framework", *Australian Journal of Electrical and Electronics Engineering*, Vol. 2, No. 1, pp. 69-80, 2005.
- S. OLARU, D. DUMUR, "Constrained predictive control for position tracking of an induction motor", *Control Engineering and Applied Informatics*, Vol. 7, No. 3, ISSN 1454-8658, 2005.

**Conférences internationales avec actes**

- S. OLARU, D. DUMUR, "Feasibility analysis of constrained predictive control", *14th Computer Science and Control Systems International Conference*, Bucarest, Roumanie, 2-5 Juillet 2003.
- S. OLARU, D. DUMUR, "Feasibility of Constrained Generalized Predictive Control within Invariant Sets Framework", *5th IEEE Asian Control Conference ASCC'04*, Melbourne, Australie, 20-23 Juillet 2004. (Best Poster Prize)
- S. OLARU, D. DUMUR, "Some Feasibility Issues Related to Constrained Generalized Predictive Control", *1st International Conference on Informatics in Control, Automation and Robotics ICINCO'04*, Setubal, Portugal, 25-28 Août 2004.
- S. OLARU, D. DUMUR, "On Constrained Predictive Control On-line Optimization Routines", *10th International IEEE Conference on Methods and Models in Automation and Robotics, MMAR'04*, Miedzyzdroje, Pologne, 30 Août – 2 Septembre 2004.
- S. OLARU, D. DUMUR, "A Parameterized Polyhedra Approach for Explicit Constrained Predictive Control", *43rd IEEE Conference on Decision and Control*, Les Bahamas, 15-17 Décembre 2004.
- S. OLARU, D. DUMUR, "Constrained predictive control for position tracking of an induction motor", *15th Computer Science and Control Systems International Conference*, Bucarest, Roumanie, 25 – 27 Mai 2005.
- S. OLARU, D. DUMUR, "Explicit constrained model predictive control. The influence of constraints redundancy", *5th IEEE International Conference on Control and Automation*, Budapest, Hongrie, 26-29 Juin 2005.
- S. OLARU, D. DUMUR, "Constrained model predictive control. Influence of redundancy in the set of constraints, possible amelioration", *IMACS Conference*, Paris, France, 11-15 Juillet 2005.
- S. OLARU, D. DUMUR, "Robust GPC using parameterized polyhedra", *IEEE Conference on Control Applications*, Toronto, Canada, Août 2005.
- S. OLARU, D. DUMUR, "Analysis of MPC feasible domains using a parameterized polyhedra approach", *International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, Freudenstadt-Lauterbad, Allemagne, 26-30 Août 2005.
- S. OLARU, D. DUMUR, "Parameterized Polyhedra Approach for Robust Constrained Generalized Predictive Control", *11th International IEEE Conference on Methods and Models in Automation and Robotics, MMAR'05*, Miedzyzdroje, Pologne, Septembre 2005.

- S. OLARU, D. DUMUR, "A parameterized polyhedra approach for the explicit Robust Model Predictive Control", *2nd International Conference on Informatics in Control, Automation and Robotics ICINCO'05*, Barcelone, Espagne, 14-17 Septembre 2005.
- S. OLARU, D. DUMUR, "Compact explicit MPC scheme guarantee of feasibility by supervising the tracking trajectory", *44th IEEE Conference on Decision and Control and European Control Conference ECC*, Séville, Espagne, Décembre 2005.

**Conférences internationales sans actes**

- S. OLARU, D. DUMUR, "Parameterized polyhedra in constrained predictive control", SIAM Conference on Control and its Applications, New Orleans, USA, 11-14 Juillet 2005.

Ce travail a par ailleurs fait l'objet d'exposés lors de colloques sans actes.

- S. OLARU, D. DUMUR, "Faisabilité de lois de commande prédictive sous contraintes", GT Commande prédictive non linéaire, Juin 2003.
- S. OLARU, D. DUMUR, "On the constraints redundancy within model predictive control", IEEE Colloquium on Predictive Control, Sheffield, Avril 2005.

# Sommaire

**Notations et abréviations**

**Symboles**

| | |
|---|---|
| **p** | Le vecteur doit être considéré comme une séquence d'autres vecteurs (en général les séquences de vecteurs sont symbolisées par des symboles en caractère gras) |
| $\hat{y}(t+j)$ | La prédiction du signal $y(t)$ à l'instant $t+j$ connaissant l'information jusqu'à l'instant *t*. Cette notation est équivalente à la forme $\hat{y}(t+j\|t)$ |
| $\text{int}(D)$ | L'ensemble des points intérieurs à un ensemble *D*, frontières non comprises |
| $cl(P)$ | La fermeture du domaine *P* |
| $\arg\min f$ | L'argument correspondant à la valeur minimale de la fonction *f* |
| $\Re$ | L'ensemble des nombres réels |
| $P_1 \oplus P_2$ | La somme de Minkowski de deux ensembles |

**Acronymes**

| | |
|---|---|
| CARIMA | Controlled AutoRegressive Integrated Moving Average) |
| LMI | Inégalité Matricielle Affine (« Linear Matrix Inequality ») |
| LP | Programmation Linéaire (« Linear Programming ») |
| LTI | Système linéaire à temps invariant |
| mpPO | Problème d'optimisation multiparamétrique |
| mpLP | Programmation Linéaire multiparametrique (« Linear Programming ») |
| mpQP | Programmation Quadratique multiparamétrique (« multiparametric Quadratic Programming ») |
| QP | Programmation Quadratique (« Quadratic Programming ») |

# 1. Introduction

Les lois de commande constituant le sujet de cette thèse sont des lois prédictives sous contraintes, faisant partie de la famille des lois à base de modèle et qui se caractérisent par un problème d'optimisation visant à atteindre pour le système le meilleur comportement selon un certain critère.

On retrouve dans l'histoire de l'Automatique différentes approches abordant le rapport entre la conception de lois de commande performantes et la prise en compte de contraintes, parfois l'envisageant comme des problèmes séparés. La confrontation entre les buts de la régulation, le suivi de trajectoire et les demandes de robustesse d'une part, et la satisfaction des contraintes d'autre part, montre que les objectifs sont le plus souvent antagonistes, principalement lorsque les degrés de liberté de la commande sont réduits. C'est pour cette raison que la prise en compte des contraintes lors de l'étape de synthèse, en parallèle avec les critères de performance, est préférable pour assurer un comportement optimal pour tous les régimes de fonctionnement d'un système asservi.

A partir de cette observation, on peut voir l'association entre la commande prédictive et les contraintes comme un défi, cette thèse par ses contributions venant enrichir la théorie de la commande optimale avec contraintes. L'axe principal de recherche consiste à examiner le comportement du système sous contraintes, et principalement l'évolution du domaine représenté par les contraintes fonction de la dynamique du système, adressant les questions liées à la faisabilité, la stabilité, la robustesse et les aspects informatiques (formulation explicite, évaluation, performances).

## *1.1 Contexte*

Le travail de recherche concrétisé par ce mémoire se trouve en fait à l'intersection de deux domaines importants, la commande des systèmes sous contraintes et la commande prédictive, de sorte que le contexte peut être divisé en deux parties.

### 1.1.1 Commande sous contraintes

Les contraintes, les plus souvent définies dans le domaine temporel, interviennent sur toutes les parties d'un système de commande, des actionneurs aux capteurs, et en général comme contraintes sur les états du système à commander. Leur présence est source de problèmes parfois complexes, non seulement pour les praticiens de l'Automatique mais aussi pour le milieu académique, car ils constituent le type de non linéarité le plus classique affectant en pratique une loi de commande, avec des implications directes sur la stabilité, les performances et la sureté de fonctionnement du système en boucle fermée (parfois avec des conséquences tragiques [STE03]).

La première prise de conscience de l'importance des contraintes (dans les années 50) a eu comme résultat l'analyse de leurs influences sur les lois de commande optimales [Kal60] en plein essor à cette époque. Les premières solutions méthodologiques ont élaboré des techniques d'adaptation de la loi sans contrainte de type 'anti-windup'. Le succès pratique rencontré pour des problèmes ponctuels avait tout de même un caractère local et envisageait plutôt une diminution des écarts au régime à suivre. Les méthodes d'anti-windup sont restées très populaires parmi les praticiens, de sorte qu'il existe aujourd'hui des études les rassemblant dans une formulation unifiée [KCM94] qui permet leur utilisation d'une façon plus élaborée [GS03], [PVH96]. Elles demeurent une formule consacrée pour la prise en compte des contraintes même si leurs détracteurs reprochent leur façon d'agir a posteriori.

Une seconde étape de recherche accomplie dans le domaine des systèmes sous contraintes a débuté dans les années 90 et se poursuit actuellement ([HEN95], [TG97], [GSD04]), avec la prise en compte des contraintes lors de l'étape de synthèse. La recherche et ses résultats fructueux ont permis de nombreuses avancées dans des domaines associés à la problématique. Ainsi, les systèmes linéaires sous contraintes font appel à la théorie des ensembles invariants [BLA99], des systèmes linéaires par morceaux avec des répercutions sur la théorie des systèmes hybrides [BEM02] et de la stabilité par fonctions de Lyapunov non quadratiques. En général, ils ouvrent la voie à une possible adaptation des méthodes de l'Automatique linéaire aux systèmes non linéaires.

### 1.1.2 Commande prédictive

La commande prédictive des systèmes linéaires invariants dans le temps, restreinte au cas sans contraintes, ne requiert pas la résolution effective d'un problème d'optimisation en ligne, car le correcteur est à son tour linéaire invariant et sa description analytique peut être obtenue hors-ligne [BD96]. Un avantage primordial issu de cette caractéristique est que toute la théorie de la commande linéaire (sans doute la plus complète de l'Automatique) peut être utilisée pour l'analyse et la synthèse de la loi prédictive sans contraintes [GM82]. Une fois le correcteur élaboré, les pôles de la boucle ouverte, la stabilité ou l'erreur stationnaire sont faciles à déterminer. Clarke [CMT87] a montré que le réglage des performances d'une boucle de commande prédictive s'effectue de façon claire par le choix des paramètres de réglage de la loi prédictive, horizons de prédiction ou pondérations. Dans ce cas, choisir une méthode de commande plutôt qu'une autre est souvent lié au contexte [BGW90]. On peut donc conclure à ce stade, en l'absence de contraintes, qu'une méthodologie très complète de commande prédictive linéaire existe, s'appliquant aux systèmes à déphasage non minimal, avec retards, instables, multivariables, comme le soulignent les études effectuées par [BD96], [RIC93] ou [MOS95].

Le problème change cependant complètement lorsque des contraintes doivent être prises en compte. Les premiers essais se sont orientés vers l'ajustement du correcteur afin de réduire les effets désastreux de l'activation de contraintes (stratégie MPC couplée à un dispositif d'anti-windup). Ces méthodes sont en fait des techniques qui cachent le problème au lieu de le résoudre. Affronter le problème au contraire revient à considérer les contraintes depuis l'étape de synthèse, imposant ainsi leur présence au sein du problème d'optimisation constituant le fondement de la commande prédictive. Cette façon de voir la commande prédictive sous contraintes se heurte à certaines difficultés, comme la mise en forme des contraintes induisant une forme canonique utilisable par les programmes d'optimisation, mais surtout la stabilité et le choix des paramètres [MRRS00]. Fait aggravant, des configurations qui pouvaient être évitées dans le cas sans contraintes, comme l'infaisabilité, nécessitent maintenant une réponse ou au moins une technique permettant de récupérer les situations compromettant les objectifs de commande. Cette problématique est traitée in extenso dans des monographies comme celles de [CB03], [MAC02] et [ROS03] qui adressent également les questions de mise en œuvre. En effet, pour l'implémentation, les programmes d'optimisation en ligne proposés le plus souvent sont basés sur les méthodes d'ensemble actif ou du point intérieur. Le succès de leur application en pratique est certes indéniable, mais très souvent avec des restrictions aux systèmes lents, comme les processus chimiques. Rien d'étonnant en fait car les praticiens de ce domaine ont été parmi ceux qui ont soutenu avec acharnement la progression de la recherche en commande prédictive [QB97].

Enfin, des avancées au niveau de la mise en forme des problèmes de commande prédictive ont été obtenues vers la fin des années 90 [KBM96], permettant l'application des techniques de type LMI, devenues très populaires et qui offrent même aujourd'hui des solutions intéressantes avec un plus concernant la facilité de traitement du cas robuste.

## *1.2 Problématique (motivations de la thèse)*

La première remarque découlant de l'analyse précédente est que les formulations explicites pour les lois de commande sous contraintes étaient inexistantes. Il n'y avait pas d'autres issues que de raisonner en terme d'un problème d'optimisation à résoudre à chaque pas d'échantillonnage. La voie ouverte par les solutions explicites (apparues au début des années 2000 [BMDP02], [BOR03], [GSD04]) offre la possibilité d'approfondir l'étude des lois prédictives.

Les bases théoriques permettant d'aborder les problèmes de stabilité, de robustesse et de structure pour la commande prédictive ont été progressivement élaborées, permettant dès lors d'envisager une grande variété de problèmes dans l'industrie des processus. La commande prédictive demeure néanmoins une philosophie basée principalement sur l'optimisation, récoltant alors les limitations associées à ces problèmes d'optimisation. Trois éléments résumant la problématique traitée par la suite émergent alors :

- La difficulté liée au temps de calcul, qui diminue beaucoup le nombre de domaines concernés par des applications des lois prédictives, reste peut être le plus grand challenge pour les acteurs de ce domaine.
- La conception d'une loi de commande implique aussi l'analyse de sa structure avant son application sur le système réel. Dès lors que la commande prédictive se base sur des problèmes d'optimisation, résolus itérativement au cours de l'évolution de la dynamique du système, détenir une structure de loi de commande est un desideratum qui n'est malheureusement pas atteignable.
- Un troisième point important lié encore une fois au fait que la loi de commande est issue d'un problème d'optimisation résolu à chaque pas d'échantillonnage est la faisabilité. Très souvent l'association des contraintes avec la dynamique du système conduit à des situations limites pour lesquelles le module de commande devient impuissant face aux demandes très contraignantes.

## *1.3 Objectifs de la thèse*

Les travaux résumés dans ce mémoire adressent directement la problématique décrite précédemment. En ce qui concerne les principes de la commande prédictive, l'objectif est de garantir sa faisabilité, en plus des critères de performances et de robustesse. L'objectif prépondérant au niveau performances numériques est d'alléger la structure des problèmes d'optimisation à la base des lois de commande prédictive. Le mémoire propose donc des solutions en réponse à ces objectifs, en traitant en priorité l'ensemble des contraintes (en s'appuyant sur leur interconnexion et donc leur éventuelle redondance). En liaison avec le critère de coût associé, une telle approche permet d'élucider les problèmes de faisabilité et d'aller jusqu'à la synthèse des lois de commande.

Les deux premières problématiques énoncées au paragraphe 1.2 trouvent des réponses appropriées par l'intermédiaire de solutions explicites aux problèmes d'optimisation, et plus particulièrement aux problèmes d'optimisation multiparamétriques. Cette classe d'optimisation, caractérisée par le fait qu'un vecteur de paramètres influence à la fois la fonction de coût et le système de contraintes, a fait l'objet d'une attention forte de la part de plusieurs groupes de recherche [BMDP02]. Ceux-ci ont adapté des résultats déjà existants dans la théorie de la recherche opérationnelle ou en calcul numérique [GN72], [GAL94] (problèmes de sensibilité numérique des optimisations). L'objectif affiché dans nos travaux est d'obtenir des résultats semblables en s'appuyant sur des *techniques géométriques* qui permettent également de mettre en évidence les aspects liés à la *topologie des domaines faisables*.

En revenant à la comparaison entre le cas avec contraintes et la commande prédictive sans contraintes, mentionnons que le correcteur perd les propriétés de linéarité, et sa non linéarité induit un système bouclé non linéaire. Par conséquent, les outils linéaires ne sont plus appropriés pour l'analyse du système asservi (au niveau stabilité, robustesse …). Dans ce cadre, l'objectif consiste à examiner les lois prédictives résultantes dans le cas sous contraintes et trouver des outils permettant l'analyse des performances (en se basant sur des concepts comme l'invariance positive).

Enfin, les lois synthétisées selon les principes prédictifs sont appliquées à un problème de positionnement pour un système électromécanique. Des résultats antérieurs ont montré que les versions sans contraintes des régulateurs prédictifs induisaient un comportement approprié mais non généralisable lors de l'apparition de contraintes. L'objectif est alors de montrer que les structures développées permettent de satisfaire le fonctionnement sous contraintes.

## *1.4 Pourquoi une approche géométrique ?*

Sachant que l'objectif principal des travaux demeure l'analyse des contraintes, une approche géométrique est-elle justifiée ? Pour les systèmes linéaires (sujet de cette thèse), les contraintes généralement utilisées sont des contraintes linéaires ou des contraintes ellipsoïdales, le choix se faisant selon un compromis entre la simplicité et le conservatisme. Si les contraintes ellipsoïdales offrent des avantages lors de l'utilisation des méthodes de type LMI par exemple [KBM96], elles font l'objet de critiques liées à leur conservatisme. En effet, en plus du fait que les limitations physiques sont rarement exprimables sous forme d'ellipsoïdes, les opérations entre ensembles n'ont pas comme résultats des ellipsoïdes (voir par exemple l'intersection) et donc pour les techniques supposant des opérations itératives sur ces domaines, la conservatisme devient non maîtrisable.

Les contraintes exprimées sous forme d'inégalités et d'égalités linéaires, qui décrivent des ensembles polyédraux, ne souffrent pas de cet inconvénient, cet argument a plaidé pour le choix des contraintes linéaires. Un autre avantage est le fait que les contraintes non linéaires (décrivant des domaines convexes) peuvent toujours être approximées par des ensembles de contraintes linéaires.

Si l'on accepte ce choix des contraintes linéaires, leur emploi permet l'utilisation de résultats matures aux niveaux théorique et algorithmique [SCH86], qui font l'objet de ce que l'on appelle la géométrie informatique. Les résultats obtenus lors de ces travaux utilisent une double représentation des domaines polyédraux, sous la forme de contraintes mais aussi sous la forme de générateurs. Cette dualité donne accès aux détails topologiques mais aussi renseigne sur les degrés de liberté dont on dispose pour les problèmes d'optimisation.

L'un des aspects originaux des travaux présentés réside dans l'utilisation d'arguments géométriques pour l'analyse de l'espace faisable dans les problèmes d'optimisation liés à la commande prédictive, nécessitant en fait l'introduction du concept de polyèdres paramétrés [LW97]. Leur manipulation a fait l'objet d'études récentes, leur concrétisation au niveau algorithmique en procédures prêtes à l'emploi pour les applications en Automatique fait partie des défis que ce travail va relever.

Les points précédents justifient pleinement le recours à une approche géométrique pour étudier et représenter les lois de commande prédictive sous contraintes. Malgré tout, la critique rencontrée au sujet de cette approche géométrique est liée principalement à la complexité de la double représentation ou à la complexité des algorithmes associés. A cette critique deux réponses, la première liée au fait qu'une double description, une fois construite, offre des avantages informatiques lors de l'implémentation des opérations sur des objets polyédraux [WIL94]. La deuxième argumente que tous les aspects géométriques analysés concernent des programmes de calcul hors ligne, que dans ce cas le temps de calcul et de développement est beaucoup moins sensible et que par conséquent, une certaine liberté d'implantation existe.

## *1.5 Organisation de la thèse*

Cette thèse est divisée en sept chapitres (non compris cette introduction), qui développent les résultats originaux obtenus durant le travail de recherche. La structure du mémoire est conçue de façon à proposer les contributions sous une forme synthétique selon un enchaînement logique qui ne reflète pas l'élaboration chronologique des différentes solutions [1].

**Chapitre 2. La commande prédictive**

Ce chapitre se veut avant tout une introduction à la problématique de la commande prédictive sous contraintes. On y retrouve les principes généraux des stratégies de commande basées sur le principe de l'horizon glissant, introduisant les problèmes d'optimisation associés. Les conditions d'optimalité de la séquence de commande sont détaillées ainsi que les procédures d'optimisation itératives existantes permettant l'implantation effective.

**Chapitre 3. La géométrie de la commande prédictive sous contraintes**

Ce chapitre débute avec l'introduction des outils géométriques nécessaires à toutes les démarches envisagées. On obtient ainsi les premiers résultats liés à la caractérisation des domaines faisables pour les lois prédictives et leur paramétrisation en fonction de l'évolution de la dynamique du système. Du point de vue algorithmique, des résultats originaux concernant la redondance des ensembles de contraintes sont élaborés. Ils autoriseront la mise en place d'une formulation alternative au sein de laquelle les optimisations ne seront plus influencées par la redondance.

**Chapitre 4. Solution explicite**

Ce chapitre constitue le cœur du travail, détaillant la mise en œuvre de l'approche géométrique pour la construction des solutions explicites pour les problèmes d'optimisation quadratiques multiparamétriques et donc également pour la commande prédictive. Les contributions de ce chapitre sont surtout de nature algorithmique, développant des méthodologies pour la formulation des fonctions linéaires par morceaux à partir de l'expression des régions faisables à l'aide des polyèdres paramétrés.

**Chapitre 5. Faisabilité**

Le but de ce chapitre est de structurer des résultats qui permettront de garantir la faisabilité et au delà la stabilité de la boucle fermée corrigée par un régulateur prédictif. La démarche analyse bien sûr les paramètres de la loi prédictive (les horizons, la pondération) et fait appel le plus souvent aux contraintes terminales. Ces contraintes terminales aident à la définition d'ensembles positifs invariants par rapport à la dynamique du système, et servent à assurer des résultats sur la faisabilité des lois prédictives. Un autre aspect traité est lié à la récupération de situations d'infaisabilité par la description d'une structure de commande dans une formulation explicite garantissant la faisabilité pour les problèmes de suivi de trajectoire.

[1] En réalité, le travail de recherche sur la commande prédictive sous contraintes a débuté par l'évaluation des procédures en ligne (chapitre 2), révélant ainsi les problèmes de temps de calcul en ligne et pour certains cas des problèmes d'infaisabilité. Une attention particulière a alors été portée aux problèmes de faisabilité (chapitre 5) et la nécessité d'une description géométrique du domaine faisable s'est progressivement imposée (chapitre 3). Cette description géométrique a permis la construction de solutions explicites (chapitre 4) apportant d'une certaine manière une réponse au problème de temps de calcul en ligne. Dans la quête de procédures plus performantes, les améliorations liées aux traitements de la redondance se sont ensuite avérées bénéfiques (chapitres 2 et 3). Les essais réalisés dans le but d'appliquer ces résultats théoriques sur un système réel ont montré la nécessité de prise en compte de perturbations, ce qui a entraîné l'étude des lois prédictives robustes (chapitre 6). Le contenu du chapitre 7 rassemble finalement des résultats obtenus tout au long de la thèse.

**Chapitre 6. Commande prédictive robuste. Le cas sous contraintes.**

Ce chapitre se penche sur la commande prédictive sous contraintes en présence de perturbations, examinant alors la problématique des lois prédictives robustes. Les formulations permettant l'implémentation en tant que problèmes d'optimisation 'min-max' multiparamétriques sont tout d'abord détaillées, les arguments géométriques sont ensuite utilisés dans la continuité des chapitres précédents pour la construction des solutions explicites dans ce cas.

**Chapitre 7. Application de la commande prédictive sous contraintes pour un problème de positionnement**

Ce chapitre est dédié à l'application de la commande prédictive pour le positionnement d'un système électro-mécanique. Ainsi, sur le même banc d'essais incluant une machine asynchrone, et reprenant les solutions élaborées pendant les développements de ce mémoire, des lois de commande prédictives sont progressivement élaborées et testées, en commençant par la stratégie sans contraintes, puis avec contraintes égalité et en analysant ensuite les conséquences de la présence de contraintes générales sur la commande ou sur la sortie prédite.

**Chapitre 8. Conclusions**

Pour finir, ce dernier chapitre fait le bilan des solutions élaborées, dressent les conclusions par rapport aux objectifs fixés et proposent des perspectives dans la continuité de ce travail.

## *1.6 Conventions*

Tout au long de ce mémoire, la présentation a été conçue de telle sorte que les éléments théoriques importants (propositions, théorèmes, …) soient numérotés et soulignés par une barre verticale gauche. Les résultats considérés comme originaux (soit au niveau théorique, soit au niveau algorithmique) sont encadrés. Les commentaires concernant des sujets proches des idées développées sont généralement adressés en notes de bas de page.

Des éléments permettant au lecteur de se familiariser avec la problématique détaillée des sujets traités sont ajoutés en Annexes. Des renvois à ces Annexes sont indiqués dans le mémoire aux emplacements appropriés.

# 2. La commande prédictive

Ce chapitre a pour objectif de rappeler dans un premier temps les principes généraux de la commande prédictive, puis d'analyser de façon plus détaillée pour des systèmes discrets invariants dans le temps les deux structures qui serviront de support aux développements futurs, la Commande Prédictive Généralisée (GPC) et la Commande Prédictive à base de modèle d'état (MPC).
Pour chacune de ces formulations, une analyse des degrés de liberté de la loi prédictive sera envisagée, ainsi que la représentation des solutions sous la forme d'une loi de commande linéaire lorsque cela est possible. Dans le cas contraire, une description du problème d'optimisation sera élaborée, directement exploitable pour un programme d'optimisation en-ligne.

Ce premier chapitre permettra également de définir les conditions d'optimalité d'une séquence de commande à chaque pas d'échantillonnage et de passer en revue les différents programmes d'optimisation adaptés à l'application de la commande prédictive en temps réel.

## *2.1 Principes de la commande prédictive*

Une description concise de cette famille de lois de commande peut être :

*L'algorithme prédictif prend en compte le comportement futur du système afin d'élaborer une commande permettant le meilleur suivi d'une trajectoire connue à l'avance.*

De façon plus détaillée, la stratégie envisagée a pour but de générer des valeurs de commande du système à piloter en tant que solutions de problèmes d'optimisation en ligne (temps réel). L'optimisation se base sur le modèle du processus et les valeurs réelles mesurées. L'intégration de mesures effectives lors de l'optimisation à chaque pas d'échantillonnage assure le bouclage de la structure de commande.

### 2.1.1 Concepts de base

Toutes les techniques proposées dans la littérature pendant presque trente années de recherche sur ce sujet ont comme base commune les idées suivantes :

- Utilisation d'un modèle du système pour construire la prédiction des signaux qui influencent les performances mais aussi la prédiction des signaux qui subissent des contraintes [2] ;
- Connaissance de la trajectoire à suivre sur un horizon au moins aussi long que l'horizon de prédiction ;
- Existence d'un critère quadratique portant en général sur l'écart entre la sortie prédite et la sortie future désirée, pondéré par l'effort de commande ;
- Existence d'un solveur élaborant en temps réel la solution optimale/sousoptimale/faisable tout en respectant les contraintes ;
- Application du premier élément de la séquence de commandes calculées ;
- Répétition de la procédure à la période d'échantillonnage suivante, selon le principe de l'horizon fuyant.

Le rôle du « modèle » est déterminant dans le succès de l'application de la commande prédictive, le modèle constitue le fondement de la loi de commande. Il faut souligner que la commande prédictive est souvent précédée d'un processus d'identification [KAN95] et que si les performances du système piloté ne sont pas satisfaisantes, il peut être nécessaire de revoir l'étape de modélisation pour élaborer un modèle ou une famille de modèles plus précis [ROS03]. [3]

Le principe de l'horizon fuyant constitue une procédure tout à fait originale qui distingue la commande prédictive des autres techniques de commande. L'idée est de fixer un horizon fini $N$, et en considérant l'état actuel comme état initial, d'optimiser une fonction de coût sur cet intervalle, tout en respectant les contraintes. Il en résulte une séquence optimale de $N$ commandes parmi lesquelles seule la première valeur sera effectivement appliquée. Au fur et à mesure que le temps avance, l'horizon de prédiction glisse et un nouveau problème

[2] Certains auteurs ont voulu souligner le fait que le modèle du système à commander constitue le point fondamental de la commande prédictive et par conséquent ont utilisé le plus souvent l'appellation « Commande prédictive à base de modèle » (MPC dans la terminologie anglaise) plutôt que « Commande prédictive ».
[3] Si seuls les paramètres du modèle doivent être modifiés et si ce processus est réalisé pendant le fonctionnement du système de commande, on peut faire une liaison entre la commande prédictive et la commande adaptative [SR96], [MOS95].

d'optimisation est à résoudre en considérant l'état du système mis à jour. En résumé, à chaque étape, il est nécessaire d'élaborer une séquence de commandes optimales en boucle ouverte, affinées systématiquement par l'arrivée de mesures présentes [4].

En plus de la dépendance envers un modèle et du principe de l'horizon glissant, une autre idée constitutive de la commande prédictive est l'exploitation explicite de la connaissance de la consigne à suivre dans le futur. Ceci implique que le schéma d'asservissement fait intervenir un effet d'anticipation en boucle fermée, avec comme conséquence principale le fait que les algorithmes prédictifs induisent des lois de commande non causales.

### 2.1.2 Schéma-bloc de fonctionnement

Le fonctionnement d'une loi de commande prédictive se décline en plusieurs actions qui doivent être validées simultanément. Tout d'abord, les performances du système asservi sont appréciées par la qualité du suivi de consigne. Dans le même temps, l'évolution des autres signaux directement liés à la dynamique doit être prédite pour assurer la satisfaction des contraintes spécifiées. Tous ces objectifs sont atteints en manipulant les signaux de commande sur un certain horizon, dit horizon de commande. A leur tour, les signaux de commande peuvent être soumis à des contraintes sur leurs amplitudes ou leurs incréments, ce qui supprime de fait un certain nombre de degrés de liberté.

Toutes ces décisions induisent une charge informatique pour l'algorithme prédictif, à comparer en terme de temps de calcul à la période d'échantillonnage du système. Formellement, la procédure s'apparente à un problème d'optimisation et peut être traitée par des programmes spécialisés afin d'obtenir la séquence optimale de commande. Cependant, le phénomène qui distingue la technique prédictive réside dans le fait que toute la procédure est réinitialisée (les fenêtres de prédiction glissent) au pas d'échantillonnage suivant. L'optimisation est alors confrontée à un nouveau contexte pour le système, le contexte étant déduit de l'évolution du système après application de la première partie de la séquence optimale au pas précédent.

**Remarque :** La loi prédictive est implémentée par une suite de problèmes d'optimisation. En règle générale, la structure de ce problème ne change pas avec l'évolution du système. Malgré tout, à chaque pas d'échantillonnage, les optimisations à résoudre dépendent tout particulièrement des caractéristiques instantanées du système à asservir. Il s'agit donc d'une paramétrisation du même problème d'optimisation. Cela revient à dire que la commande prédictive est implémentée par l'intermédiaire d'un problème d'optimisation multiparamétrique. Les paramètres sont en fait des *paramètres de contexte*, terme général désignant toutes les grandeurs caractérisant le système à un instant donné, comme les commandes passées, les sortie présentes et passées ou les consignes futures sur un horizon bien défini. Par exemple, toutes ces informations sont regroupées de façon compacte au sein de l'état du système, les paramètres de contexte sont alors le vecteur d'état du système. Comme la notion *d'état du système* est liée aux modèles par représentation d'état, par la suite le terme de *paramètres de contexte* sera conservé pour designer les paramètres qui influencent le problème d'optimisation dans le cadre prédictif (si le modèle à la base de la prédiction est un modèle d'état, alors les *paramètres de contexte* coïncident avec le *vecteur d'état*).

La figure 2.1 propose ainsi un schéma illustrant le principe de la philosophie prédictive. On peut également se ramener par abstraction et en isolant certains blocs à un schéma fonctionnel comme celui reproduit figure 2.2. L'avantage d'une telle description est l'évidence de l'existence de la rétroaction qui n'apparaît pas directement dans la définition de base des algorithmes prédictifs, mais qui intervient comme conséquence de l'intégration des mesures courantes dans le processus de décision.

De même, la figure 2.2 permet l'identification des blocs constitutifs par leur implémentation effective. Par exemple, le bloc qui procure une action anticipative de la loi prédictive est le bloc qui traite la consigne sur un horizon futur, caractérisé du point de vue pratique par le fait que son interaction avec la partie bouclée se fait seulement dans le sens d'envoi d'information. Pour son implémentation, il nécessite des modules qui permettent le stockage des signaux de consigne et une mise à jour régulière.

[4] Le fait que le principe de l'horizon glissant soit intimement lié à la façon d'agir de l'homme a été souvent évoqué dans la littérature, généralement en citant comme exemple des applications classiques comme celle d'un conducteur d'automobile. Le fait que la conquête d'un horizon infini passe par des mesures sur un horizon fini prend sa source dans les idées philosophiques universelles, par exemple en mathématique. Un numéro d'un important journal international dédié aux défis des contraintes en automatique choisissait comme 'motto' une citation inspirée de *Johann Wolfgang von Goethe (Gott, Gemut und Welt)* pouvant être vu comme une réflexion lointaine sur le paradigme d'horizon glissant :

*"If to the Infinite you want to stride*
*Just walk in Finite to every side"*

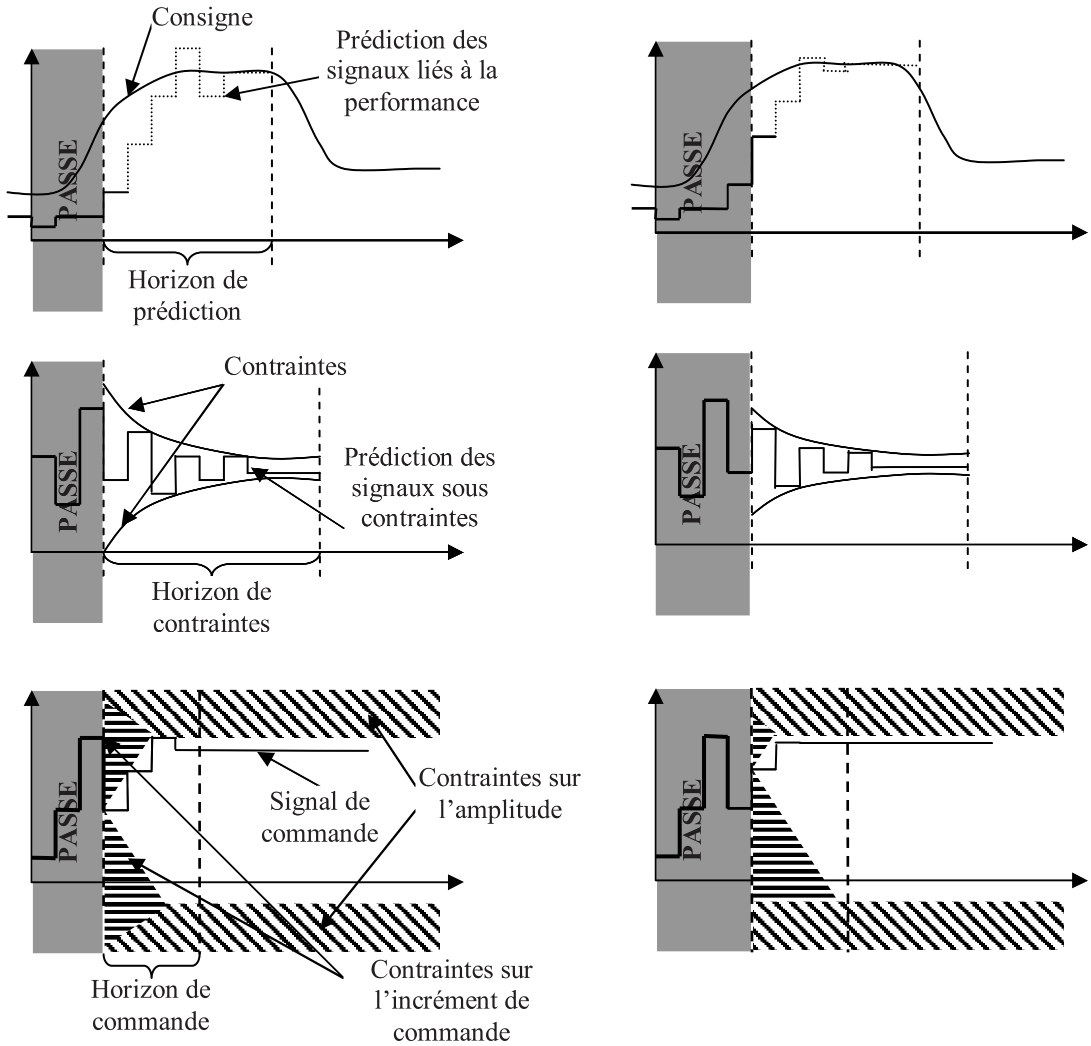



**Figure 2.1. Philosophie de la commande prédictive. A gauche : Résolution d'un problème de commande en boucle ouverte sur un horizon fini à l'instant *t*. A droite : Réitération de la même procédure pour le nouveau contexte du système à l'instant *t*+1 selon un horizon glissant.**

Le bloc fonctionnel qui regroupe les éléments nécessaires à la construction d'une séquence optimale à chaque pas d'échantillonnage reste le plus complexe. Il doit regrouper les ingrédients nécessaires pour la formulation d'un problème d'optimisation à l'instant $t$ dans le cadre prédictif :

- Le critère de coût, offrant un moyen de comparer différents scénarii de commande :

$$J(\{u_t, u_{t+1}, \ldots, u_{t+N-1}\}, x_t) = \sum_{k=0}^{N-1} L(x_{t+k}, u_{t+k}) + F(x_N) \tag{2.1}$$

avec $L(x_k, u)$ le coût instantané pour l'étape prédite, $N$ le nombre d'étapes prises en compte pour la prédiction et $F(x_N)$ le coût terminal. Le paramètre de contexte $x_t$ est construit tel qu'en se basant sur le modèle on ait :

$$x_{t+k} = \phi(x_t, u_t, u_{t+1}, \ldots, u_{t+k-1})$$

- Les contraintes de type égalité :

$$\begin{aligned}
&f_0(x_t, u_t) = 0;\\
&f_1(\{x_t, x_{t+1}\}, \{u_t, u_{t+1}\}) = 0;\\
&\ldots\\
&f_{N_c}(\{x_t, x_{t+1}, \ldots, x_{t+N_c}\}, \{u_t, u_{t+1}, \ldots, u_{t+N_c}\}) = 0;
\end{aligned} \tag{2.2}$$

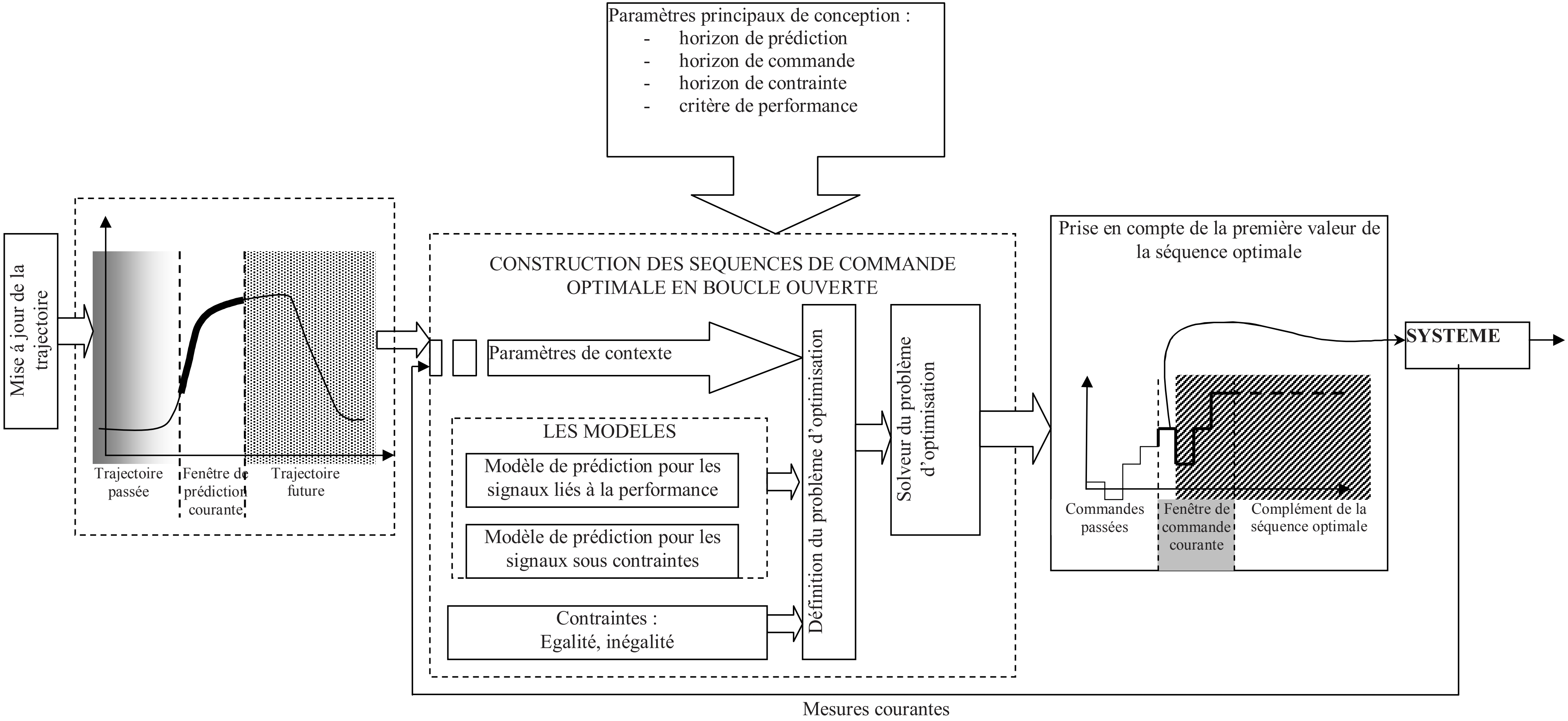



**Figure 2.2 : Schéma bloc d'une structure de commande prédictive.**

avec $N_c$ le nombre de pas pour lequel on demande la satisfaction des contraintes.

- les contraintes de type inégalité :

$$g_0(x_t, u_t) \leq 0;$$
$$\ldots$$
$$g_{N_c}(\{x_t, x_{t+1}, \ldots, x_{t+N_c}\}, \{u_t, u_{t+1}, \ldots, u_{t+N_c}\}) \leq 0 \tag{2.3}$$

La structure d'optimisation est liée au choix du modèle et des contraintes ; l'implémentation quant à elle se caractérise par la méthodologie utilisée pour construire la solution optimale (algorithme, mémoire nécessaire, logique combinatoire dédiée ou puissance de calcul requise). Les paramètres de réglage de la commande prédictive modifient exclusivement cette partie du schéma de commande.

Le bloc permettant de ne délivrer que la première partie de la séquence optimale pour application effective sur le système est l'image symbolique de l'application du principe de l'horizon glissant, par le fait que toute la suite de la séquence est abandonnée. Son implémentation ne nécessite pas d'efforts matériels mais ce bloc constitue le dernier maillon pouvant modifier la commande effectivement appliquée. Son implantation peut donc être renforcée par des mécanismes de sûreté évitant des comportements aberrants dus à l'infaisabilité du problème d'optimisation par exemple.

**Remarque :** La loi prédictive dispose d'une famille de signaux de commande beaucoup plus riche que les lois préétablies en boucle ouverte. Il faut imaginer par exemple les lois de commande avec horizon 1, pour lesquelles la loi en boucle ouverte correspond à un échelon d'une certaine amplitude. La loi prédictive a quant à elle la liberté de choisir parmi toutes les familles de commande par le fait que le principe de l'horizon glissant lui permet d'envisager des modifications de l'action de commande à chaque pas d'échantillonnage.

### 2.1.3 Paramètres de réglage

Pour les lois de commande classiques (de type PID par exemple), on retrouve comme paramètres de réglage l'amplification, le temps d'action intégrale, le temps d'action dérivée, ou des paramètres liés aux caractéristiques fréquentielles comme la bande passante, les marges de stabilité … Dans le cas des lois prédictives, les paramètres sont d'une toute autre nature de par la formulation temporelle des objectifs de commande.

Enumérer tous les paramètres de réglage n'est pas chose aisée à cause de la multitude des approches prédictives existantes. Malgré tout, les paramètres de synthèse que l'on retrouve comme dénominateur commun de la majorité des stratégies existantes sont :

- les horizons :
  - de contrainte – qui décrivent explicitement les intervalles sur lesquels on prend en compte les restrictions physiques ou de performances ;
  - de prédiction – sur lesquels les performances du système asservi sont évaluées ;
  - de commande – qui expriment la liberté autorisée à chaque pas d'échantillonnage pour construire des séquences optimales en boucle ouverte ;
- les pondérations intervenant dans le critère de performance :
  - sur les erreurs ;
  - sur l'effort de commande ;
  - sur l'évolution de la dynamique du système ;
- les paramètres terminaux
  - contraintes terminales ;
  - coûts terminaux.

Il a été montré précédemment qu'une loi prédictive est construite sur la solution d'un problème d'optimisation à chaque pas d'échantillonnage. Parmi les différentes approches évoquées dans la littérature sur la commande prédictive figurent également des lois de commande basées sur la modification de la structure (horizons) du problème d'optimisation lors de l'implémentation effective [CM96]. Toutefois, ce type de stratégie, qui offre certes des renseignements théoriques, s'avère peu convaincant pratiquement, principalement à cause des problèmes induits par ces modifications de la structure du problème d'optimisation en temps réel. En laissant donc ces stratégies de côté, il est possible d'examiner l'influence des paramètres de réglage d'une loi prédictive sur la structure et la complexité du problème d'optimisation. Les points suivants s'avèrent alors significatifs pour la suite des travaux :

- le nombre d'arguments – il est affecté exclusivement par l'horizon de commande choisi ;
- la structure de la fonction de coût – c'est la partie sur laquelle intervient la majorité des paramètres. On retrouve l'horizon de prédiction, les coûts terminaux et bien évidement toutes les pondérations définies en liaison avec le critère ;
- le nombre de contraintes et leur complexité – l'horizon de contrainte joue pour l'ensemble des contraintes un rôle déterminant, mais on retrouve aussi les contraintes terminales. Il est important de noter que le réglage des paramètres liés à l'ensemble des contraintes est très sensible par les implications qu'ils induisent vis-à-vis de la faisabilité et de la stabilité du système en boucle fermée (voir chapitre 5).

Mis à part ces aspects qui mettent en relief les éléments constitutifs de la commande prédictive, la synthèse des lois prédictives (principalement en présence de contraintes) doit accorder une attention toute particulière aux aspects liés au temps de calcul de l'algorithme. Ils sont généralement dépendants du temps nécessaire à la construction de la séquence optimale. Il est toujours nécessaire de comparer ce temps de calcul à la période d'échantillonnage qui caractérise l'application. De ce point de vue, on peut voir le processus d'optimisation comme un programme qui à son tour doit subir des contraintes. Les plus naturelles sont exprimées en temps de calcul, en nombre d'itérations permises, ou en précision autorisée pour les programmes itératifs. Dans leur plus grande majorité, les problèmes d'optimisation avec contraintes mixtes (égalités et inégalités) qui font appel aux solveurs en-ligne doivent accepter un certain degré de sous-optimalité. Ceci peut donc être vu comme un paramètre de réglage, même s'il ne fait pas partie théoriquement de la construction de la loi de commande.

**Remarque :** Certaines implémentations prédictives, conscientes de la charge informatique dont elles doivent faire face, se contentent d'appliquer des commandes contenues dans la dernière séquence disponible jusqu'au moment où la phase d'optimisation délivre une nouvelle solution.

## *2.2 Commande Prédictive Généralisée (GPC)*

Parmi toutes les méthodes prédictives reprenant bien sûr les principes exposés précédemment, la Commande Prédictive Généralisée [CMT87] est peut-être celle qui a connu le plus grand nombre d'applications [SKH01] et qui demeure une référence [BD96] dans le cas de la commande prédictive des systèmes monovariables. C'est pourquoi ce paragraphe propose une description des idées principales de cette stratégie.

### 2.2.1 Le cas sans contrainte

La commande GPC se différentie des autres algorithmes prédictifs par deux caractéristiques majeures :

- Même si toute représentation demeure admissible, elle utilise le plus souvent pour la prédiction du comportement un modèle entrée/sortie par fonction de transfert de type CARIMA (Controlled AutoRegressive Integrated Moving Average) :

$$A(q^{-1})y_t = B(q^{-1})u_{t-1} + C(q^{-1})\frac{\xi_t}{\Delta(q^{-1})} \tag{2.4}$$

  où $u$, $y$ représentent l'entrée et la sortie du système à commander, $\xi_t$ est un signal aléatoire centré non corrélé avec l'entrée, $A$ et $B$ sont des polynômes en l'opérateur de retard $q^{-1}$ de degrés $n_a$ et $n_b$ respectivement, et $\Delta(q^{-1}) = 1 - q^{-1}$ est l'opérateur différence. $C(q^{-1})$ est un polynôme en l'opérateur retard, lié aux perturbations et par la suite, sans une connaissance supplémentaire sur la nature des perturbations, il sera choisi égal à 1 (sa valeur n'influe pas par ailleurs sur le comportement en suivi de trajectoire, il peut jouer un rôle en rejet de perturbation).

- Le critère de performance est représenté par une fonction de coût quadratique considérant l'erreur de poursuite et l'effort de commande sur un horizon glissant de la forme :

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \left[w_{t+j} - \hat{y}_{t+j}\right]^2 + \lambda \sum_{j=1}^{N_u} \left[\Delta u_{t+j-1}\right]^2 \tag{2.5}$$

  où $\hat{y}_{t+j}$ représente la prédiction optimale à l'instant $t+j$ connaissant les paramètres de contexte à l'instant présent $t$. $N_1, N_2$ caractérisent le début et la fin de la fenêtre de prédiction sur la sortie. $N_u$

est l'horizon de prédiction sur la commande, $\lambda > 0$ un facteur de pondération sur l'effort de commande et enfin $w$ est la consigne à suivre supposée connue sur l'horizon de prédiction.

**Remarque :** En règle générale, l'horizon de prédiction sur la sortie est supérieur à l'horizon de commande et pour la cohérence de la prédiction il est supposé qu'à la fin de l'horizon de commande :

$$\Delta u_{t+j} = 0,\ j \geq N_u \tag{2.6}$$

**Remarque :** Le modèle CARIMA fait intervenir les incréments de commande et non la commande effective. Cet aspect permet d'imposer au final une action intégrale au sein du régulateur et assure par conséquent une erreur statique nulle pour des consignes et perturbations constantes. L'aspect incrémental du modèle se retrouve aussi dans le critère par la présence de $\Delta u$.

En se basant sur le modèle mentionné en (2.4) et en appliquant les idées de modélisation présentées par Clarke et ses coauteurs [CMT87], un prédicteur peut être construit sous la forme :

$$y_{t+j} = \underbrace{F_j(q^{-1})y_t + H_j(q^{-1})\Delta u_{t-1}}_{\text{réponse libre}} + \underbrace{G_j(q^{-1})\Delta u_{t+j-1} + J_j(q^{-1})\xi_{t+j}}_{\text{réponse forcée}} \tag{2.7}$$

avec $F_j, G_j, H_j, J_j$ polynômes solutions uniques des équations diophantiennes suivantes :

$$\begin{aligned} &\Delta(q^{-1})A(q^{-1})J_j(q^{-1}) + q^{-j}F_j(q^{-1}) = 1 \\ &G_j(q^{-1}) + q^{-j}H_j(q^{-1}) = B(q^{-1})J_j(q^{-1}) \end{aligned} \tag{2.8}$$

Le prédicteur optimal déduit de la considération que la meilleure estimée du signal perturbateur dans le futur est égale à sa moyenne, nulle ici, prend la forme :

$$\hat{y}_{t+j} = \underbrace{F_j(q^{-1})y_t + H_j(q^{-1})\Delta u_{t-1}}_{\text{réponse libre}} + \underbrace{G_j(q^{-1})\Delta u_{t+j-1}}_{\text{réponse forcée}} \tag{2.9}$$

En écrivant ces polynômes sous forme matricielle :

$$\grave{\mathbf{I}}\mathbf{H} = \begin{bmatrix} H_{N_1}(1) \cdots H_{N_1}(n_b - 1) \\ \vdots \qquad\qquad \vdots \\ H_{N_2}(1) \cdots H_{N_2}(n_b - 1) \end{bmatrix} ;\ \mathbf{IF} = \begin{bmatrix} F_{N_1}(1) \cdots F_{N_1}(n_a) \\ \vdots \qquad\qquad \vdots \\ F_{N_2}(1) \cdots F_{N_2}(n_a) \end{bmatrix} \tag{2.10}$$

avec $n_a$ le degré de $A$ et $n_b$ le degré de $B$, la réponse libre peut être écrite sous la forme :

$$\mathbf{l} = \mathbf{IH}\ \mathbf{\Delta u}_{\mathbf{past}} + \mathbf{IF}\ \mathbf{y}_{\mathbf{past}} \tag{2.11}$$

où : $\mathbf{\Delta u}_{\mathbf{past}} = \begin{bmatrix}\Delta u_{t-1} & \Delta u_{t-2} & \cdots & \Delta u_{t-n_b-1}\end{bmatrix}^{\mathrm{T}}$ et $\mathbf{y}_{\mathbf{past}} = \begin{bmatrix}y_t & y_{t-1} & \cdots & y_{t-n_a-1}\end{bmatrix}^{\mathrm{T}}$

De façon similaire, la réponse forcée peut être réécrite sous forme matricielle :

$$\mathbf{G} = \begin{bmatrix} g_{N_1} & g_{N_1-1} & \cdots & 0 \\ g_{N_1+1} & g_{N_1} & \cdots & 0 \\ \vdots & \cdots & \cdots & 0 \\ g_{N_u} & \cdots & \cdots & g_1 \\ \vdots & \cdots & \cdots & \vdots \\ g_{N_2} & g_{N_2-1} & \cdots & g_{N_2-N_u+1} \end{bmatrix} \in \Re^{(N_2-N_1+1)\times N_u},\ g_0 = 0 \tag{2.12}$$

avec les coefficients $g_k$ issus des polynômes $G_j$ correspondant à ceux de la réponse indicielle du modèle. Les contraintes (2.6) sont prises en compte dans (2.12), car la matrice $\mathbf{G}$ a une taille limitée à $N_u$ colonnes.

Le vecteur des prédictions $\hat{\mathbf{y}} = \begin{bmatrix}\hat{y}_{t+N_1} & \cdots & \hat{y}_{t+N_2}\end{bmatrix}^{\mathrm{T}}$ s'écrit sous forme vectorielle en fonction de la séquence des incréments futurs $\mathbf{k}_u = \begin{bmatrix}\Delta u_t & \cdots & \Delta u_{t+N_u-1}\end{bmatrix}^{\mathrm{T}}$ :

$$\hat{\mathbf{y}} = \mathbf{G}\ \mathbf{k_u} + \mathbf{l} \tag{2.13}$$

Avec le vecteur des consignes futures $\mathbf{w} = \left[w_{t+N_1} \ \cdots \ w_{t+N_2}\right]^{\mathrm{T}}$, le critère (2.5) s'écrit :

$$\begin{aligned} J &= (\mathbf{G}\,\mathbf{k_u} + \mathbf{l} - \mathbf{w})^{\mathrm{T}}(\mathbf{G}\,\mathbf{k_u} + \mathbf{l} - \mathbf{w}) + \lambda\,\mathbf{k_u^{\mathrm{T}}}\,\mathbf{k_u} \\ &= \mathbf{k_u^{\mathrm{T}}}(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\,\mathbf{I}_{N_u \times N_u})\mathbf{k_u} + 2(\mathbf{l} - \mathbf{w})^{\mathrm{T}}\mathbf{G}\,\mathbf{k_u} + (\mathbf{l} - \mathbf{w})^{\mathrm{T}}(\mathbf{l} - \mathbf{w}) \end{aligned} \tag{2.14}$$

La problématique de la commande prédictive revient donc à retrouver l'argument optimal pour le critère :

$$\min_{\mathbf{k_u}} \mathbf{k_u^{\mathrm{T}}}(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\,\mathbf{I}_{N_u \times N_u})\mathbf{k_u} + 2(\mathbf{l} - \mathbf{w})^{\mathrm{T}}\mathbf{G}\,\mathbf{k_u} \tag{2.15}$$

ou plus explicitement :

$$\min_{\mathbf{k_u}} \mathbf{k_u^{\mathrm{T}}}(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\,\mathbf{I}_{N_u \times N_u})\mathbf{k_u} + 2\,\mathbf{p}_t^{\mathrm{T}}\left[\mathbf{IH}^{\mathrm{T}} \ \ \mathbf{IF}^{\mathrm{T}} \ \ -\mathbf{I}_{(N_2-N_1+1)\times(N_2-N_1+1)}\right]^{\mathrm{T}}\mathbf{G}\mathbf{k_u} \tag{2.16}$$

avec $\mathbf{p}_t = \left[\Delta u_{t-1} \ \cdots \ \Delta u_{t-n_b-1} \ \ y_t \ \cdots \ y_{t-n_a-1} \ \ w_{t+N_1} \ \cdots \ w_{t+N_2}\right]^{\mathrm{T}}$ le vecteur des *paramètres de contexte* pour le problème d'optimisation (2.16). Comme il a été mentionné, ce vecteur des paramètres de contexte est le seul qui modifie, par sa mise à jour, le problème d'optimisation donnant la séquence de commande optimale.

Une première conclusion pour la formulation GPC dérivé de l'équation (2.16) est que **la fonction de coût GPC possède une forme quadratique multiparamétrique** :

$$J^*(\mathbf{k_u}) = \min_{\mathbf{k_u}}\left\{\mathbf{k_u^{\mathrm{T}}}\,\mathbf{H}\,\mathbf{k_u} + 2\,\mathbf{p}_t^{\mathrm{T}}\,\mathbf{F}\,\mathbf{k_u}\right\} \tag{2.17}$$

En revenant au problème (2.16), si aucune contrainte n'est ajoutée, l'optimum peut être obtenu analytiquement. Puisque la matrice $\mathbf{H}$ est définie positive (par construction $\lambda > 0$), la solution unique est donnée par :

$$\mathbf{k_u^*} = \underbrace{-(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\,\mathbf{I}_{N_u \times N_u})^{-1}\mathbf{G}^{\mathrm{T}}}_{\mathbf{M}}\left[\mathbf{IH}^{\mathrm{T}} \ \ \ \mathbf{IF}^{\mathrm{T}} \ \ -\mathbf{I}_{(N_2-N_1+1)\times(N_2-N_1+1)}\right]\mathbf{p}_t \tag{2.18}$$

Selon le principe de l'horizon glissant, la première composante de $\mathbf{k_u^*}$, à savoir $\Delta u_t$, est réellement appliquée. Si l'on note $\mathbf{m}$ la première ligne de la matrice $\mathbf{M} = (\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\,\mathbf{I}_{N_u \times N_u})^{-1}\mathbf{G}^{\mathrm{T}}$, il vient :

$$\Delta u_t = -\mathbf{m}\left[\mathbf{IH}\,\mathbf{\Delta u}_{\mathbf{past}} + \mathbf{IF}\,\mathbf{y}_{\mathbf{past}} - \mathbf{w}\right] \tag{2.19}$$

qui constitue en fait un régulateur linéaire représentable sous forme polynomiale RST (Figure 2.3) :

$$S(q^{-1})\Delta u_t = -R(q^{-1})y_t + T(q)\,w_t \tag{2.20}$$

$$\text{avec :} \quad \begin{array}{ll} S(q^{-1}) = (1 + \mathbf{m}\,\mathbf{IH}\,q^{-1}) & \text{degré}\left[S(q^{-1})\right] = \text{degré}\left[B(q^{-1})\right] \\ R(q^{-1}) = \mathbf{m}\,\mathbf{IF} & \text{degré}\left[R(q^{-1})\right] = \text{degré}\left[A(q^{-1})\right] \\ T(q) = \mathbf{m}\left[q^{N_1} \ \cdots \ q^{N_2}\right]^{\mathrm{T}} & \text{degré}\left[T(q)\right] = N_2 \end{array} \tag{2.21}$$

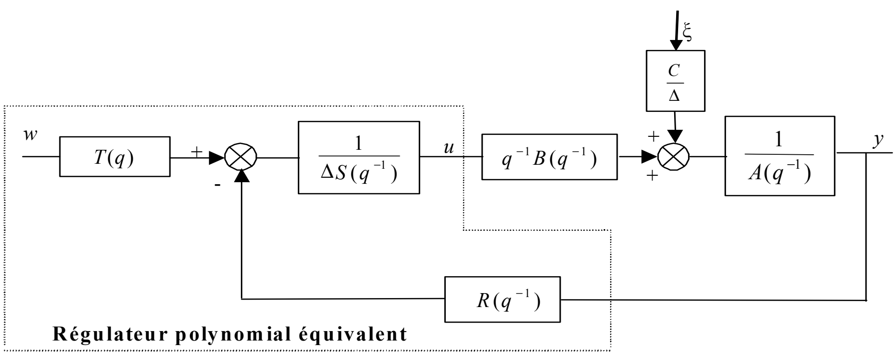



**Figure 2.3 : Régulateur polynomial équivalent de la loi GPC sans contrainte.**

On obtient donc pour le cas sans contrainte une loi programmable par une simple équation aux différences, ce qui évite une optimisation en ligne (toujours coûteuse). Les avantages d'une telle formulation compacte polynomiale sont décrits de façon détaillée dans la littérature, par exemple [BD96].

### 2.2.2 Les contraintes

Le critère (2.16) est quadratique en $\mathbf{k_u}$, mais en réalité les solutions possibles de cette optimisation sont restreintes par l'existence des contraintes provenant de limitations portant sur l'amplitude de la commande, la vitesse de réaction de l'actionneur, le signal de sortie ou autres signaux critiques pour le fonctionnement du système, mais aussi des contraintes terminales permettant de renforcer la stabilité [EBD96].

Les contraintes qui interviennent dans la formulation GPC peuvent être décrites par l'intermédiaire d'un formalisme unifié [SC94b]. Sachant que les incréments des commandes futures sont les variables qui subissent finalement les contraintes, il est intéressant d'exprimer toutes les limitations en se basant sur des modèles de structure similaire à la relation (2.4) afin de faciliter la construction des prédictions des signaux sous contraintes :

$$\begin{cases} A_\gamma(q^{-1})\,\gamma_t = B_\gamma(q^{-1})\,u_{t-1} \\ \gamma_{\min} \le \hat{\gamma}_{t+j} \le \gamma_{\max}, \quad N_{c1} \le j \le N_{c2} \end{cases} \tag{2.22}$$

avec $A_\gamma, B_\gamma$ des polynômes (de degrés $n_a^\gamma$ et $n_b^\gamma$) en l'opérateur retard $q^{-1}$, $\gamma_t$ le signal contraint, $\gamma_{\min}, \gamma_{\max}$ les limitations à respecter. $N_{c1}$ et $N_{c2}$ représentent l'horizon de contrainte et peuvent être différents des horizons de prédiction ($N_1, N_2$) ou de l'horizon de commande choisi, $N_u$.

En utilisant $(A_\gamma, B_\gamma)$, il devient possible de construire les prédictions des signaux contraints $\hat{\gamma}_{t+j}$ comme la superposition d'une réponse libre et d'une réponse forcée :

$$\hat{\gamma}_{t+j} = \underbrace{F_{\gamma_j}(q^{-1})\gamma_t + H_{\gamma_j}(q^{-1})\Delta u_{t-1}}_{l_\gamma=\text{réponse libre}} + \underbrace{G_{\gamma_j}(q^{-1})\Delta u_{t+j-1}}_{\text{réponse forcée}} \tag{2.23}$$

ou sous forme vectorielle :

$$\hat{\gamma}_{t+j} = \mathbf{F}_{\gamma_j}\boldsymbol{\gamma}_{\mathbf{past}} + \mathbf{H}_{\gamma_j}\boldsymbol{\Delta}\mathbf{u}_{\mathbf{past}} + \mathbf{G}_{\gamma_j}\mathbf{k_u} \tag{2.24}$$

avec $\mathbf{k}_u = \begin{bmatrix}\Delta u_t & \cdots & \Delta u_{t+N_u-1}\end{bmatrix}^{\mathrm{T}}$, $\boldsymbol{\Delta}\mathbf{u}_{\mathbf{past}} = \begin{bmatrix}\Delta u_{t-1} & \Delta u_{t-2} & \cdots & \Delta u_{t-n_b^\gamma-1}\end{bmatrix}^{\mathrm{T}}$ et $\boldsymbol{\gamma}_{\mathbf{past}} = \begin{bmatrix}\gamma_t & \gamma_{t-1} & \cdots & \gamma_{t-n_a^\gamma-1}\end{bmatrix}^{\mathrm{T}}$.

Les vecteurs $\mathbf{F}_{\gamma_j}$, $\mathbf{H}_{\gamma_j}$, $\mathbf{G}_{\gamma_j}$ contiennent les coefficients des polynômes $F_{\gamma_j}(q^{-1})$, $H_{\gamma_j}(q^{-1})$, $G_{\gamma_j}(q^{-1})$ solutions d'équations diophantiennes similaires à celles données par la relation (2.8).

Afin d'illustrer le mécanisme de construction des contraintes pour un problème GPC, les paragraphes suivants proposent quelques exemples illustratifs de contraintes très souvent rencontrées en pratique.

### 2.2.3 Contraintes sur l'incrément de commande

On s'intéresse ici aux contraintes de la forme :

$$\Delta u_{\min} \le u_{t+j} - u_{t+j-1} \le \Delta u_{\max}, N_{c1} \le j \le N_{c2}$$

Le modèle de prédiction pour le signal contraint peut se construire à partir de (2.22) avec $A_\gamma = 1, B_\gamma = \Delta$ :

$$\begin{cases} \gamma_t = \Delta u_t; \\ \Delta u_{\min} \le \hat{\gamma}_{t+j} \le \Delta u_{\max}, \quad N_{c1} \le j \le N_{c2} \end{cases} \tag{2.25}$$

Les solutions des équations diophantiennes sont alors :

$$F_{\gamma_j}(q^{-1}) = 0, H_{\gamma_j}(q^{-1}) = 0, G_{\gamma_j}(q^{-1}) = q^{j-1}$$

soit, en les regroupant sous forme matricielle comme en (2.10) et (2.12) :

$$\mathbf{G}_{\gamma} = \mathbf{I}_{N_u \times N_u}, \mathbf{IF}_{\gamma} = \mathbf{0}, \mathbf{IH}_{\gamma} = \mathbf{0}$$

l'expression matricielle pour ce type de contraintes :

$$\mathbf{1}\Delta u_{\min} \leq \mathbf{G}_{\gamma}\mathbf{k}_{\mathbf{u}} \leq \mathbf{1}\Delta u_{\max} \tag{2.26}$$

avec $\mathbf{1} \in \Re^{N_u \times 1}$ le vecteur composés d'éléments unitaires.

### 2.2.4 Contraintes sur l'amplitude de commande

On s'intéresse ici aux contraintes de la forme :

$$u_{\min} \leq u_{t+j} \leq u_{\max}, \quad N_{c1} \leq j \leq N_{c2}$$

Le modèle de prédiction pour le signal contraint peut se construire à partir de (2.22) avec $A_{\gamma} = 1, B_{\gamma} = 1$ :

$$\begin{cases} \gamma_t = u_t; \\ u_{\min} \leq \hat{\gamma}_{t+j} \leq u_{\max}, \quad N_{c1} \leq j \leq N_{c2} \end{cases} \tag{2.27}$$

Les solutions des équations diophantiennes sont alors :

$$H_{\gamma_j} = 0, F_{\gamma_j} = 1, G_{\gamma_j} = 1 + q^{-1} + q^{-2} + \cdots + q^{-j+1}$$

soit, en les regroupant sous forme matricielle :

$$\mathbf{G}_{\gamma} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}, \quad \mathbf{IF}_{\gamma} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{IH}_{\gamma} = \mathbf{0}$$

l'expression matricielle pour ce type de contraintes :

$$\mathbf{1}u_{\min} \leq \begin{bmatrix} \mathbf{IF}_{\gamma} & \mathbf{G}_{\gamma} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_{past} \\ \mathbf{k}_{u} \end{bmatrix} \leq \mathbf{1}u_{\max}; \boldsymbol{\gamma}_{past} = \mathbf{u}(t-1) \tag{2.28}$$

### 2.2.5 Contraintes sur la sortie

On s'intéresse ici aux contraintes de la forme :

$$y_{\min} \leq y_{t+j} \leq y_{\max}, \; N_{c1} \leq j \leq N_{c2}$$

Les prédictions pour le signal contraint sont construites avec le même modèle utilisé lors de la construction du critère, donc $A_{\gamma} = A, B_{\gamma} = B$ :

$$\begin{cases} A(q^{-1})\gamma_t = B(q^{-1})u_{t-1}; \\ y_{\min} \leq \hat{\gamma}_{t+j} \leq y_{\max}, \quad N_{c1} \leq j \leq N_{c2} \end{cases} \tag{2.29}$$

Les solutions des équations diophantiennes ont été déjà décrites par (2.10) et (2.12) de sorte que la formulation compacte des contraintes sur la sortie est donnée par :

$$y_{\min} \leq \begin{bmatrix} \mathbf{IF}_{\gamma} & \mathbf{IH}_{\gamma} & \mathbf{G}_{\gamma} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_{past} \\ \Delta\mathbf{u}_{past} \\ \mathbf{k}_{u} \end{bmatrix} \leq y_{\max}; \boldsymbol{\gamma}_{past} = \mathbf{y}_{past} \tag{2.30}$$

### 2.2.6 Contraintes sur le dépassement

On s'intéresse ici aux contraintes de la forme :

$$y_{t+j} \leq w_{t+j}, N_{c1} \leq j \leq N_{c2}$$

Il s'agit encore une fois de contraintes sur la sortie, et donc le modèle est donné par $A_\gamma = A, B_\gamma = B$, avec la particularité que la consigne à suivre intervient également :

$$\begin{cases} \gamma_t = y_t; \\ \hat{\gamma}_{t+j} \leq w_{t+j}, \quad N_{c1} \leq j \leq N_{c2} \end{cases} \tag{2.31}$$

Les solutions des équations diophantiennes ont été déjà décrites par (2.10) et (2.12) de sorte que la formulation compacte des contraintes dans ce cas est donnée par :

$$\left[\mathbf{IF}_\gamma \;\; \mathbf{IH}_\gamma \;\; \mathbf{G}_\gamma \;\; -\mathbf{I}_{(N_{c2}-N_{c1}+1)\times(N_{c2}-N_{c1}+1)}\right]\begin{bmatrix} \mathbf{y}_{past} \\ \Delta\mathbf{u}_{past} \\ \mathbf{k}_u \\ \mathbf{w} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} w_{t+N_{c1}} \\ \vdots \\ w_{t+N_{c2}} \end{bmatrix} \tag{2.32}$$

### 2.2.7 Contraintes terminales de type égalité

On s'intéresse ici [CS91] aux contraintes de la forme :

$$y_{t+j} = w_{t+N_2}, \quad N_2 < N_{c1} \leq j \leq N_{c2}$$

Les prédictions des sorties sont toujours construites à partir du modèle nominal $A_\gamma = A, B_\gamma = B$, en étendant cependant la portée de la prédiction au delà de l'horizon de prédiction nécessaire pour le critère de coût ($N_2 < N_{c1} \leq N_{c2}$) :

$$\begin{cases} \gamma_t = y_t; \\ \hat{\gamma}_{t+j} = w_{N_2}, \quad N_2 < N_{c1} \leq j \leq N_{c2} \end{cases} \tag{2.33}$$

Les prédictions sont construites à partir des polynômes $F_{\gamma_j}(q^{-1}), H_{\gamma_j}(q^{-1}), G_{\gamma_j}(q^{-1})$ de façon similaire à (2.10) et (2.12), mais pour $j > N_2$. Par exemple la matrice $\mathbf{G}_\gamma$ pour $N_2 < j$ s'exprime par :

$$\mathbf{G}_\gamma = \begin{bmatrix} g_{N_{c1}} & \cdots & g_{N_{c1}-N_u+1} \\ \vdots & \vdots & \vdots \\ g_{N_{c2}} & \cdots & g_{N_{c2}-N_u+1} \end{bmatrix} \in \Re^{(N_{c2}-N_{c1}+1)\times N_u}$$

Avec ces éléments, les contraintes de type égalité à la fin de l'intervalle de prédiction se reformulent par :

$$\left[\mathbf{IF}_\gamma \;\; \mathbf{IH}_\gamma \;\; \mathbf{G}_\gamma \;\; -\mathbf{J}\right]\begin{bmatrix} \mathbf{y}_{past} \\ \Delta\mathbf{u}_{past} \\ \mathbf{k}_u \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \; \mathbf{J} = \begin{bmatrix} \mathbf{0} \in \Re^{(N_{c2}-N_{c1}+1)\times(N_{c2}-N_{c1})} & \begin{matrix} 1 \\ \vdots \\ 1 \end{matrix} \end{bmatrix}, \; \mathbf{w} = \begin{bmatrix} w_{t+N_{c1}} \\ \vdots \\ w_{t+N_{c2}} \end{bmatrix} \tag{2.34}$$

### 2.2.8 Autres types de contraintes

En plus des contraintes décrites précédemment, d'autres contraintes peuvent être rencontrées lors des applications, comme par exemple imposer un comportement monotone pour éviter les oscillations, ou encore imposer des contraintes agissant sur le comportement des systèmes à déphasage non minimal. En conclusion de ces développements sur les formulations des contraintes, il faut souligner que le but est d'exprimer les limitations en termes de degrés de liberté sur l'expression sur lesquelles elles agissent, soit ici $\mathbf{k}_u$.

Toutes les limitations ont été représentées par des contraintes linéaires, comme on peut le constater au travers des relations (2.26), (2.28), (2.30), (2.32) et (2.34). Parmi les différentes formes que ces contraintes peuvent prendre, il est toujours possible d'identifier la présence du vecteur correspondant à l'argument du problème d'optimisation, $\mathbf{k}_u$. En particularisant ce vecteur qui demeure l'objectif du processus de décision, l'ensemble de toutes les contraintes peut s'écrire sous forme compacte :

$$\text{Contraintes GPC :} \quad \begin{cases} \mathbf{A}_{in}\;\mathbf{k}_u \leq \mathbf{B}_{in}\,\mathbf{p}_t + \mathbf{b}_{in} \\ \mathbf{A}_{eq}\;\mathbf{k}_u = \mathbf{B}_{eq}\,\mathbf{p}_t + \mathbf{b}_{eq} \end{cases} \tag{2.35}$$

Cette formulation est très utile du point de vue pratique, mais la présence du vecteur des paramètres de contexte – $\mathbf{p}_t$ – mérite quelques développements. Dans la formulation (2.35), ce vecteur regroupe en fait une séquence finie des commandes passées, des consignes futures mais aussi des valeurs présentes et passées des signaux contraints. Formellement, il s'exprime par :

$$\mathbf{p}_t = \left[ \left\{ \gamma_{past} \right\}^{\mathrm{T}} \ \ \Delta\mathbf{u}_{past}^{\mathrm{T}} \ \ \mathbf{w}^{\mathrm{T}} \right]^{\mathrm{T}}$$

avec $\left\{ \gamma_{past} \right\}$ regroupant l'ensemble de toutes les séquences passées des signaux sous contraintes nécessaires pour la construction des prédictions $\hat{\gamma}$.

Finalement, en regroupant la formulation du critère de performance (2.17) avec l'ensemble des contraintes à satisfaire pendant le processus de prédiction, on obtient le problème de commande en boucle ouverte que la loi GPC doit résoudre à chaque pas d'échantillonnage en vue d'une application selon le principe de l'horizon glissant :

$$\begin{array}{l} \min\limits_{\mathbf{k}_u} J_t = \dfrac{1}{2}\mathbf{k}_u^{\mathrm{T}} \, \mathbf{H}\,\mathbf{k}_u + \mathbf{p}_t^{\mathrm{T}}\mathbf{F}\,\mathbf{k}_u \\ \text{sous les contraintes :} \begin{cases} \mathbf{A}_{in} \ \mathbf{k}_u \le \mathbf{B}_{in}\,\mathbf{p}_t + \mathbf{b}_{in} \\ \mathbf{A}_{eq} \ \mathbf{k}_u = \mathbf{B}_{eq}\,\mathbf{p}_t + \mathbf{b}_{eq} \end{cases} \end{array} \tag{2.36}$$

Ce type de critère fait partie de la classe des problèmes de programmation non-linéaire, plus spécifiquement de la catégorie des problèmes de programmation quadratique. Le fait que la fonction de coût et les contraintes dépendent d'un vecteur des paramètres $\mathbf{p}_t$ induit que toute implémentation de la loi GPC avec contraintes doit envisager la construction de la solution de ce problème de programmation quadratique multiparamétrique à chaque pas d'échantillonnage.

## *2.3 MPC – commande prédictive avec modèle d'état*

L'approche prédictive la plus proche de la théorie standard pour les systèmes linéaires est certainement celle qui considère un modèle par représentation d'état :

$$x_{t+1} = Ax_t + Bu_t, \quad t \in Z^+ \tag{2.37}$$

avec $x_t \in \Re^n$ le vecteur d'état, $u_t \in \Re^m$ le vecteur de commande à l'instant *t*, *A* et *B* les matrices de dimensions correspondantes. Il est supposé que la paire $(A, B)$ est stabilisable. A chaque pas d'échantillonnage, l'état courant (que l'on suppose mesurable) $x_t = x_{t|t}$ peut être utilisé pour la recherche de la séquence de commande optimale en boucle ouverte :

$$\mathbf{k}_u^* = \left[ u_{t|t}^{\mathrm{T}} \ \cdots \ u_{t+N_u-1|t}^{\mathrm{T}} \right]^{\mathrm{T}}$$

solution du problème d'optimisation suivant :

$$\mathbf{k}_u^* = \arg\min_{\mathbf{k}_u} x_{t+N|t}^{\mathrm{T}} P\, x_{t+N|t} + \sum_{k=1}^{N-1} \left\{ x_{t+k|t}^{\mathrm{T}} Q x_{t+k|t} + u_{t+k|t}^{\mathrm{T}} R\, u_{t+k|t} \right\} \tag{2.38}$$

tout en respectant les contraintes imposées par la dynamique du modèle considéré, les contraintes fonctionnelles et autres contraintes terminales :

$$\begin{cases} x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t} & k \ge 0 \\ Cx_{t+k|t} + Du_{t+k|t} \le \gamma & 1 \le k \le N_c \\ u_{t+k|t} = Kx_{t+k|t} & N_u \le k \le N-1 \end{cases} \tag{2.39}$$

où $Q = Q^{\mathrm{T}} \ge 0$ et $R = R^{\mathrm{T}} > 0$ sont les matrices de pondération et $P = P^{\mathrm{T}} \ge 0$ définit le coût terminal. Il est supposé que $(Q^{1/2}, A)$ est détectable. Les contraintes de type inégalité sont décrites par $\gamma \in \Re^q, C \in \Re^{q \times n}, D \in \Re^{q \times m}$ qui représentent toutes les limitations sur les combinaisons linéaires des états et/ou commandes. Le vecteur $\gamma \ge 0$ est défini tel que l'ensemble faisable contienne l'origine. Si l'on fait abstraction de ces particularités liées au modèle, on retrouve les éléments caractéristiques de la commande prédictive : l'horizon de prédiction compris entre $t+1$ et $t+N$, l'horizon de commande entre $t$ et $t+N_u-1$ et l'horizon de contrainte entre $t+1$ et $t+N_c$. A chaque pas, ces horizons se déplacent selon le principe de l'horizon glissant. Les pondération $Q, R$ et $P$ constituent les paramètres de réglage au niveau de la fonction de coût. Concernant les contraintes sur la structure de commande à la fin de l'horizon de prédiction, selon le système à commander, le gain $K$ peut prendre différentes valeurs. Un choix classique reste cependant le gain $K_{LQR}$ issu du problème de régulation quadratique à horizon infini sans contraintes, construit à partir de l'équation algébrique discrète de Riccati :

$$\begin{aligned} K_{LQR} &= (R+B^T PB)^{-1} B^T PA \\ P &= Q + A^T PA - A^T PB(R+B^T PB)^{-1} B^T PA \end{aligned} \tag{2.40}$$

En revenant au problème d'optimisation (2.38)-(2.39), on constate la présence d'une somme de termes quadratiques et d'un ensemble fini de contraintes. On peut alors essayer de séparer les arguments afin d'obtenir une formulation directement exploitable par un programme d'optimisation. En effet, en compactant les états prédits en un seul vecteur :

$$\mathbf{x} = \begin{bmatrix} x_{t+1|t}^{\mathrm{T}} & \cdots & x_{t+N_2|t}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

et en notant $x_{t|t} = x$, le problème d'optimisation (2.38)-(2.39) devient :

$$\begin{aligned} &\arg\min_{\mathbf{k}_u} x^T Qx + \mathbf{x}^T \overline{Q}\mathbf{x} + \mathbf{k}_u^T \overline{R}\mathbf{k}_u \\ &\text{sous les contraintes :} \begin{cases} \overline{D}\mathbf{k}_u + \widetilde{C}x + \overline{C}x \le \overline{\gamma} \\ J\mathbf{k}_u = \overline{K}x \end{cases} \end{aligned} \tag{2.41}$$

pour lequel on a considéré :

$$\begin{aligned} \overline{Q} &= diag\begin{bmatrix} Q & \cdots & Q & P \end{bmatrix} \in \Re^{Nn\times Nn} \\ \overline{R} &= diag\begin{bmatrix} R & \cdots & R \end{bmatrix} \in \Re^{Nm\times Nm} \\ \overline{D} &= diag\begin{bmatrix} D & \cdots & D \end{bmatrix} \in \Re^{N_c q\times N_c n} \end{aligned} \tag{2.42}$$

$$J = \begin{bmatrix} \underbrace{0 \quad I_{(N-N_u)m}}_{Nm} \end{bmatrix}; \overline{K} = \begin{bmatrix} 0 & diag\begin{bmatrix} \underbrace{K \quad \cdots \quad K}_{(N-N_u)\ fois} \end{bmatrix} \end{bmatrix} \in \Re^{(N-N_u)m\times(N-N_1+1)m}$$

$$\overline{C} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & C & 0 \end{bmatrix} \in \Re^{N_c q\times Nn}; \overline{\gamma} = \begin{bmatrix} \gamma \\ \gamma \\ \vdots \\ \gamma \end{bmatrix} \in \Re^{N_c q\times 1}; \widetilde{C} = \begin{bmatrix} C \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \Re^{N_c q\times n}$$

On peut aussi réécrire l'évolution décrite par (2.37) sur l'horizon de prédiction sous forme matricielle :

$$\mathbf{x} = \Phi x + \Gamma \mathbf{k}_u \tag{2.43}$$

avec :

$$\Phi = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}; \Gamma = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \tag{2.44}$$

En utilisant l'expression (2.44) et la formulation (2.41), on obtient alors :

$$\arg\min_{\mathbf{k}_u} 0{,}5\,\mathbf{k}_u^{\mathrm{T}} H\mathbf{k}_u + x^{\mathrm{T}} F\mathbf{k}_u + x^{\mathrm{T}} Gx$$
$$\text{sous les contraintes :} \begin{cases} A_{in}\mathbf{k}_u \le B_{in}x + b_{in} \\ A_{eq}\mathbf{k}_u = B_{eq}x \end{cases} \tag{2.45}$$

où l'on a noté :

$$G = Q + \Phi^{\mathrm{T}}\overline{Q}\,\Phi;\ F = \Gamma^{\mathrm{T}}\overline{Q}\,\Phi;\ H = \overline{R} + \Gamma^{\mathrm{T}}\overline{Q}\,\Gamma$$

$$A_{in} = \overline{D} + \overline{C}\Gamma \in \Re^{r\times Nm};\ B_{in} = \widetilde{E} + \overline{E}\Phi \in \Re^{r\times n};\ b_{in} = \overline{\gamma} \in \Re^{r};\ r = Nq$$

$$A_{eq} = J - \overline{K}\Gamma \in \Re^{s\times Nm};\ B_{eq} = \overline{K}\Phi \in \Re^{s\times n};\ s = (N - N_u)q;\ p = N\,m$$

Dans la formulation (2.45), l'argument du problème d'optimisation est $\mathbf{k}_u$ de sorte que la présence du terme $x^{\mathrm{T}}Gx$ ne modifie pas l'argument optimal. Ce terme n'apparaîtra plus par la suite.

Par ailleurs, à l'ensemble des contraintes (2.39) s'ajoute un jeu de contraintes terminales forçant l'état à la fin de l'horizon de prédiction à atteindre un certain ensemble polyèdral :

$$x_{t+N} \in X_N = \left\{ x \in \Re^n \,\middle|\, L_{in}x \le l_{in};\ L_{eq}x = l_{eq} \right\} \tag{2.46}$$

Ce type de contraintes peut être facilement ajouté à la formulation (2.45) comme un ensemble de contraintes mixtes paramétrées.

En conclusion, dans le cas général, l'algorithme MPC doit être en mesure de résoudre un problème d'optimisation multiparamétrique, pour lequel le paramètre de contexte est l'état $x$ du système :

$$\arg\min_{\mathbf{k}_u} \frac{1}{2}\mathbf{k}_u^{\mathrm{T}} H\mathbf{k}_u + x^{\mathrm{T}} F\,\mathbf{k}_u$$
$$\text{sous les contraintes :} \begin{cases} A_{in}\mathbf{k}_u \le B_{in}x + b_{in} \\ A_{eq}\mathbf{k}_u = B_{eq}x \end{cases} \tag{2.47}$$

**Remarque :** Si l'ensemble des contraintes dans la formulation (2.47) est vide, on considère alors un problème MPC sans contraintes pour lequel la solution analytique unique est immédiate, sachant que la matrice $H$ est définie positive. La séquence de commandes optimales est alors :

$$\mathbf{k}_u = -H^{-1}F^{\mathrm{T}}x \tag{2.48}$$

et, en exprimant la seule commande courante qui sera effectivement appliquée, on déduit la loi linéaire :

$$u_{t|t} = \begin{bmatrix} I_{n\times n} & \underbrace{0_{n\times n} \ \cdots \ 0_{n\times n}}_{N-1} \end{bmatrix} H^{-1}F^{\mathrm{T}}x = K_{MPC}\ x \tag{2.49}$$

évitant ainsi l'utilisation d'un solveur en ligne (Figure 2.4) et n'étant pas plus complexe du point de vue implémentation qu'une loi de commande classique par retour d'état. Son expression est tout à fait similaire à celle de la loi RST trouvée pour le cas GPC (2.20).

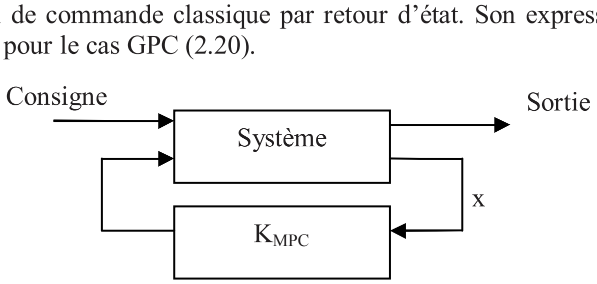



**Figure 2.4 : Schéma d'implémentation d'une loi MPC sans contraintes.**

**Remarque :** Le modèle (2.37) se base sur une expression classique des représentations d'état. Comme on a pu le constater, dans le cas GPC, le modèle CARIMA permettait aux lois prédictives de travailler avec des incréments de commande, induisant un intégrateur dans le schéma de régulation. Le même principe peut être utilisé avec un modèle d'état incrémental, basé sur un état augmenté ([CB03], [ROS03]) :

$$\bar{x} = \begin{bmatrix} x_t^{\mathrm{T}} & u_{t-1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

incluant un intégrateur et pour lequel tous les développements précédents restent valables. Dans la suite, on considérera que l'argument du problème d'optimisation liée à la loi prédictive est une séquence d'incréments futurs de la commande :

$$\mathbf{k}_u = \begin{bmatrix} \Delta u_{t|t}^{\mathrm{T}} & \cdots & \Delta u_{t+N_u-1|t}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \quad \rightarrow \quad \mathbf{k}_u = \begin{bmatrix} u_{t|t}^{\mathrm{T}} & \cdots & u_{t+N_u-1|t}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{2.50}$$

## 2.4 Conditions d'optimalité

Partant du problème (2.36) avec $x$ paramètre de contexte :

$$\begin{aligned} &\arg\min_{\mathbf{k}_u} 0{,}5\,\mathbf{k}_u^{\mathrm{T}} H \mathbf{k}_u + x^{\mathrm{T}} F\, \mathbf{k}_u \\ &\text{sous les contraintes} : \begin{cases} A_{in}\mathbf{k}_u \le B_{in}x + b_{in} \\ A_{eq}\mathbf{k}_u = B_{eq}x + b_{eq} \end{cases} \end{aligned} \tag{2.51}$$

et de la supposition que le vecteur de paramètres de contexte noté ici $x$ est mesurable, l'objectif de ce paragraphe consiste à se demander quelles sont les conditions d'optimalité pour une séquence de commande $[\mathbf{k}_u(1) \cdots \mathbf{k}_u(r)]$. Pour cela, on considère le gradient de la fonction objectif $J$ et son Hessien :

$$\begin{aligned} \nabla J(\mathbf{k}_u) &= \begin{bmatrix} \dfrac{\partial J}{\partial \mathbf{k}_u(1)} & \cdots & \dfrac{\partial J}{\partial \mathbf{k}_u(r)} \end{bmatrix}^T = H\mathbf{k}_u + F^T x; \\ \nabla^2 J(\mathbf{k}_u) &= \begin{bmatrix} \dfrac{\partial^2 J}{\partial \mathbf{k}_u(1)^2} & \cdots & \dfrac{\partial^2 J}{\partial \mathbf{k}_u(1)\partial \mathbf{k}_u(r)} \\ \vdots & & \vdots \\ \dfrac{\partial^2 J}{\partial \mathbf{k}_u(1)\partial \mathbf{k}_u(r)} & \cdots & \dfrac{\partial^2 J}{\partial \mathbf{k}_u(r)^2} \end{bmatrix} = H \end{aligned} \tag{2.52}$$

La classe de problème qui nous intéresse (2.51) possède l'avantage que la structure de la fonction de coût est connue, ses valeurs étant également calculables, de même pour son gradient. Un autre aspect important est que le degré de satisfaction des contraintes est facile à évaluer. En effet, une caractéristique des problèmes d'optimisation liés à la commande prédictive est leur appartenance à la catégorie de problèmes d'optimisation convexe car la fonction objectif et l'ensemble faisable sont convexes [BSS93].

### 2.4.1 Cas sans contraintes

Pour le cas sans contraintes, tous les points $\mathbf{k}_u \in \Re^p$ sont admissibles et donc les conditions d'optimalité sont exprimées exclusivement en analysant les propriétés de la fonction objectif $J$ (La Figure 2.5 propose un exemple de fonction objectif quadratique). Ainsi pour un point candidat, on peut directement vérifier :

*Les conditions nécessaires d'optimalité :*

- Conditions du premier ordre : $\nabla J(\mathbf{k}_u) = H\mathbf{k}_u + F^T x = 0$ (2.53)
- Conditions de deuxième ordre : $\nabla^2 J(\mathbf{k}_u) = H \ge 0$ (2.54)

*Les conditions suffisantes d'optimalité :*

- Conditions du premier ordre : $\nabla J(\mathbf{k}_u) = H\mathbf{k}_u + F^T x = 0$ (2.55)
- Conditions de deuxième ordre : $\nabla^2 J(\mathbf{k}_u) = H > 0$ (2.56)

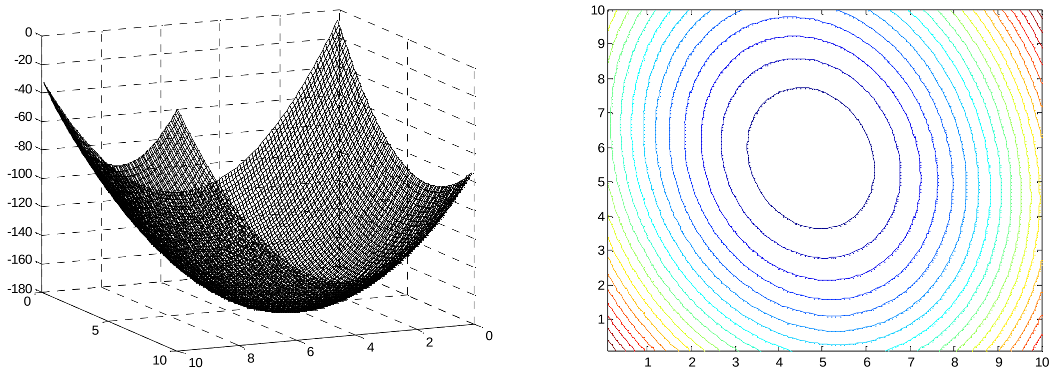


**Figure 2.5 : Fonction de coût quadratique (à gauche) et courbes de niveau (à droite).**

Même si l'absence de contraintes est un phénomène plutôt rare lors d'applications réelles, la solution sans contraintes ne manque pas d'intérêt car les algorithmes d'optimisation déterminent souvent la solution d'un problème d'optimisation avec contraintes en construisant itérativement une séquence de solutions pour des problèmes sans contraintes.

Comme on peut le constater à partir des relations (2.53) à (2.56), l'optimum est identifié par le point stationnaire satisfaisant :

$$H\mathbf{k}_u + F^T x = 0 \Rightarrow \mathbf{k}_u = -H^{-1}F^T x \tag{2.57}$$

Notons que lors d'implémentations pratiques, il est préférable d'éviter l'inversion matricielle apparaissant dans (2.57), qui pourrait s'avérer désastreuse si la matrice $\Gamma$ dans (2.45) est mal conditionnée, induisant des répercussions au niveau de la construction du Hessien $H$, en privilégiant une méthode de type « moindres carrés ». La même observation est valable pour la construction de la loi GPC dans le cas sans contraintes (2.19) pour laquelle le mauvais conditionnement de la matrice $G$ dans (2.13) (et en conséquence du Hessien dans (2.19)) doit être pris en compte. Ainsi, dans [MAC02], il est montré que pour la commande prédictive sans contraintes, la séquence des commandes optimales peut être directement décrite comme la solution au sens des « moindres carrés » d'un système d'équations non carré résolu à l'aide de l'algorithme QR.

**Remarque :** Du point de vue implantation temps réel, l'expression de la solution (2.57) sous la forme particulière $\mathbf{k}_u = -Kx$ avec $K = H^{-1}F^T$ est indispensable. Elle permet la programmation d'une loi polynomiale RST dans le cas GPC ou d'une loi linéaire par retour d'état pour le cas MPC, les deux impliquant un temps de calcul en temps réel extrêmement faible.

### 2.4.2 Cas avec contraintes égalité

Il s'agit ici de la minimisation d'une fonction de coût de la forme :

$$J(\mathbf{k}_u) = 0{,}5\,\mathbf{k}_u^{\mathrm{T}} H \mathbf{k}_u + x^{\mathrm{T}} F \mathbf{k}_u \tag{2.58}$$

pour laquelle un ensemble de contraintes égalité est défini :

$$A_{eq}\mathbf{k}_u = B_{eq}x + b_{eq} \tag{2.59}$$

On fait l'hypothèse pour la suite que l'ensemble des contraintes est compatible pour toute valeur possible du vecteur des paramètres (il est supposé qu'il existe au moins un point solution de (2.59) pour chaque $x$). On supposera également que le rang de la matrice $A_{eq}$ est maximal par ligne (dans le cas contraire, une procédure d'élimination des contraintes linéaires dépendantes peut être mise en place).

Du point de vue du problème d'optimisation, l'apparition de contraintes égalité équivaut à supprimer un certain nombre de degrés de liberté égal au nombre de lignes dans (2.59), noté $s$. En conséquence, le problème d'optimisation peut être réduit à un problème équivalent de dimension $p-s$ (où $p$ est la dimension de $\mathbf{k}_u$). Un exemple de fonction quadratique avec contraintes égalité est donné Figure 2.6.

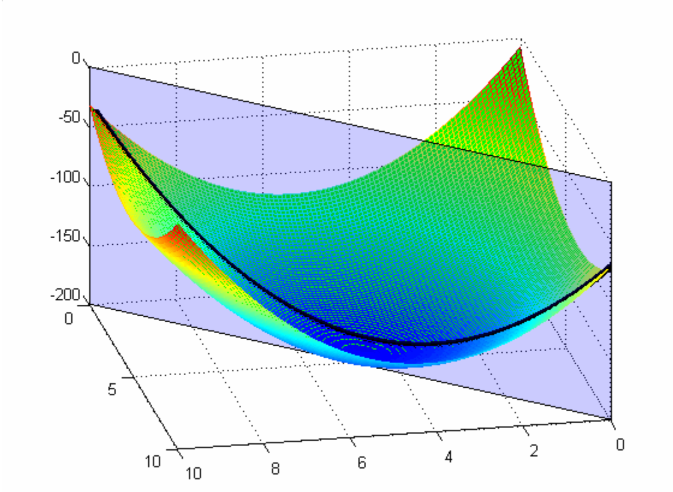



**Figure 2.6 : Fonction de coût quadratique avec contraintes égalité.**

Pour définir les conditions d'optimalité, il est nécessaire d'identifier les directions d'optimisation admissibles. Une telle direction admissible $\mathbf{z}$ relie deux points admissibles $\mathbf{k}_u^1, \mathbf{k}_u^2$ :

$$\begin{aligned} A_{eq}\mathbf{k}_u^1 &= b_{eq} + B_{eq}x; \\ A_{eq}\mathbf{k}_u^2 &= b_{eq} + B_{eq}x \end{aligned} \tag{2.60}$$

et vérifie donc :

$$A_{eq}(\mathbf{k}_u^1 - \mathbf{k}_u^2) = A_{eq}\mathbf{z} = 0 \tag{2.61}$$

L'ensemble des vecteurs satisfaisant (2.61) constitue un sous-espace de l'espace défini par les vecteurs $\mathbf{k}_u$. Soit $Z$ une matrice dont les colonnes forment une base pour ce sous-espace. On a donc :

$$A_{eq}Z = 0 \tag{2.62}$$

et toute direction admissible $\mathbf{z}$ est un combinaison linéaire des colonnes de $Z$. En contrepartie, on peut aussi définir une base pour le sous-espace décrit par les lignes de $A_{eq}$ (les $s$ contraintes indépendantes). Soit $Y$ la matrice qui regroupe cette base. La matrice $Y$ n'est pas unique et une façon de la choisir est par exemple $Y = A_{eq}^T$. Puisque $Y$ et $Z$ représentent des sous-espaces complémentaires, un vecteur $\mathbf{k}_u$ peut être écrit sous la forme :

$$\mathbf{k}_u = Y\mathbf{k}_u^Y + Z\mathbf{k}_u^Z \tag{2.63}$$

A partir de (2.63) on peut déduire :

$$A_{eq}\mathbf{k}_u = A_{eq}(Y\mathbf{k}_u^Y + Z\mathbf{k}_u^Z) = b_{eq} + B_{eq}x \Rightarrow A_{eq}Y\mathbf{k}_u^Y = (b_{eq} + B_{eq}x) \Rightarrow \mathbf{k}_u^Y = (A_{eq}Y)^{-1}(b_{eq} + B_{eq}x) \tag{2.64}$$

et en conséquence détenir un point initial admissible :

$$\mathbf{k}_u^{init} = Y(A_{eq}Y)^{-1}(b_{eq} + B_{eq}x)$$

En revanche, toutes les solutions possibles pour (2.59) sont décrites par :

$$\mathbf{k}_u = \mathbf{k}_u^{init} + Z\mathbf{k}_u^Z = Y(A_{eq}Y)^{-1}(b_{eq} + B_{eq}x) + Z\mathbf{k}_u^Z \tag{2.65}$$

et donc pour retrouver la solution optimale pour (2.58)-(2.59), il reste la liberté de travailler sur $\mathbf{k}_u^Z$.

Pour caractériser les conditions d'optimalité, il faut donc être conscient que l'on manipule un problème sans contraintes avec la liberté dans l'espace du gradient projeté. Une condition nécessaire est alors que le gradient projeté de $J$ au point $\mathbf{k}_u^*$ satisfasse :

$$Z^T\nabla J(\mathbf{k}_u) = 0 \tag{2.66}$$

ce qui du fait de (2.62) implique que le gradient $\nabla J(\mathbf{k}_u)$ soit une combinaison linéaire de ligne de $A_{eq}$, soit :

$$\nabla J(\mathbf{k}_u) + A_{eq}^T \mu = 0 \tag{2.67}$$

où le vecteur $\mu$ représente le vecteur de multiplicateurs Lagrange, unique si les lignes de $A_{eq}$ sont linéairement indépendantes. D'une façon similaire, on peut retrouver le Hessien projeté pour exprimer les conditions d'optimalité.

*Conditions nécessaires d'optimalité :*

- Contraintes égalité : $A_{eq}\mathbf{k}_u^* = b_{eq} + B_{eq}x$ (2.68)
- Conditions du premier ordre : $Z^T \nabla J(\mathbf{k}_u) = Z^T (H\mathbf{k}_u + F^T x) = 0$ (2.69)
- Conditions de deuxième ordre : $Z^T \nabla^2 J(\mathbf{k}_u) Z = Z^T HZ \geq 0$ (2.70)

*Conditions suffisantes d'optimalité :*

- Contraintes égalité : $A_{eq}\mathbf{k}_u^* = b_{eq} + B_{eq}x$ (2.71)
- Conditions du premier ordre : $Z^T \nabla J(\mathbf{k}_u) = Z^T (H\mathbf{k}_u + F^T x) = 0$ (2.72)
- Conditions de deuxième ordre : $Z^T \nabla^2 J(\mathbf{k}_u) Z = Z^T HZ > 0$ (2.73)

Si le Hessien est défini positif, sa projection sera aussi définie positive et en utilisant l'expression des solutions possibles en termes des directions admissibles (2.65) on peut écrire l'optimum sous la forme :

$$\begin{aligned}\mathbf{k}_u^* &= Y(A_{eq}Y)^{-1}(b_{eq} + B_{eq}x) - Z(Z^T HZ)^{-1} Z^T (HY(A_{eq}Y)^{-1}(b_{eq} + B_{eq}x) + F^T x) = \\ &= (Y(A_{eq}Y)^{-1} - Z(Z^T HZ)^{-1} Z^T HY(A_{eq}Y)^{-1}) b_{eq} \\ &\qquad + (Y(A_{eq}Y)^{-1} B_{eq} - Z(Z^T HZ)^{-1} Z^T (HY(A_{eq}Y)^{-1} B_{eq} + F^T))x\end{aligned} \tag{2.74}$$

L'utilisation de l'expression (2.74) pour la construction effective de la valeur optimale est fortement contre-indiquée car les matrices impliquées sont mal conditionnées. De façon pratique, la solution unique de (2.58) peut être obtenue en utilisant une approche plus performante numériquement qui part des conditions :

$$\begin{cases} H\mathbf{k}_u + A_{eq}^T \mu = -F^T x \\ A_{eq}\mathbf{k}_u = b_{eq} + B_{eq}x \end{cases} \tag{2.75}$$

Ces équations peuvent être écrites sous forme matricielle :

$$\begin{bmatrix} H & A_{eq}^T \\ A_{eq} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{k}_u \\ \mu \end{bmatrix} = \begin{bmatrix} -F^T x \\ b_{eq} + B_{eq}x \end{bmatrix} \tag{2.76}$$

la matrice du membre de gauche étant supposée non singulière de rang maximal par ligne (ce qui est souvent le cas pour les problème de commande prédictive car le Hessien est défini positif). Elle peut être ensuite factorisée sous la forme **LU** :

$$\begin{bmatrix} H & A_{eq}^T \\ A_{eq} & 0 \end{bmatrix} = \mathbf{LU} \tag{2.77}$$

pour faciliter la résolution par élimination gaussienne d'un premier problème :

$$\mathbf{L}y = \begin{bmatrix} -F^T x \\ b_{eq} + B_{eq}x \end{bmatrix} \tag{2.78}$$

Sa solution sera ensuite utilisée pour obtenir la solution finale $\mathbf{k}_u^*$ après une deuxième résolution :

$$\mathbf{U}\begin{bmatrix}\mathbf{k}_u \\ \mu\end{bmatrix} = y \tag{2.79}$$

**Remarque :** Sachant que la solution optimale peut être exprimée sous forme explicite, les techniques prédictives avec contraintes égalité peuvent profiter d'une implémentation par une loi linéaire de faible complexité.

### 2.4.3 Cas avec contraintes inégalité

Il s'agit à présent de la minimisation d'une fonction de coût du type :

$$J(\mathbf{k}_u) = 0{,}5\,\mathbf{k}_u^{\mathrm{T}} H\,\mathbf{k}_u + x^{\mathrm{T}} F\,\mathbf{k}_u \tag{2.80}$$

pour lequel le vecteur $\mathbf{k}_u$ peut prendre des valeur à l'intérieur d'un domaine décrit par l'ensemble des inégalités :

$$A_{in}\mathbf{k}_u \le B_{in}x + b_{in} \tag{2.81}$$

En comparaison avec le cas des contraintes égalité, pour lequel le nombre de contraintes est limité par le rang de la matrice $A_{eq}$, dans le cas présent il n'y a pas de restriction sur le nombre des contraintes inégalité. Un exemple graphique d'une fonction quadratique et des limitations imposées par les contraintes inégalité est donné Figure 2.7a.

**Remarque :** Pendant de nombreuses années, des méthodes naïves traitant les problèmes d'optimisation avec contraintes inégalité pour la commande prédictive ont été utilisées principalement pour les applications impliquant des saturations de la commande. Il s'agissait essentiellement de remplacer les composantes de l'optimum sans contrainte violant les contraintes par leurs limites. Des tels choix s'avèrent parfois très éloignés (Figure 2.7b) du véritable optimum avec contraintes. Ces techniques sont de plus inutilisables pour des contraintes plus complexes (par exemple sur la sortie).

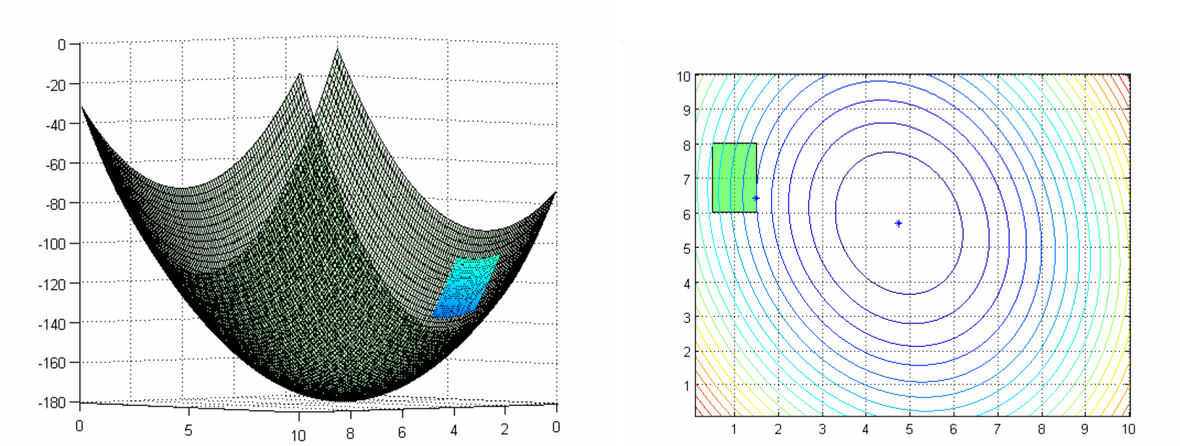



**Figure 2.7 : Fonction de coût quadratique et limitation du domaine par des contraintes inégalité (à droite). Argument de l'optimum sans contraintes et de l'optimum avec contraintes.**

Analysons alors pour une certaine valeur du vecteur de paramètres de contexte $x^0$ les conditions pour qu'un certain point $\mathbf{k}_u$ puisse se trouver relativement à une sortie $a_{in_i}^T\mathbf{k}_u \le b_{in_i} + B_{in_i}x^0$ dans l'ensemble (2.81) :

- Si $a_{in_i}^T\mathbf{k}_u = b_{in_i} + B_{in_i}\,x^0$ la contrainte est dite active (ou saturée) ;
- Si $a_{in_i}^T\mathbf{k}_u < b_{in_i} + B_{in_i}\,x^0$ la contrainte est dite inactive ;
- Si $a_{in_i}^T\mathbf{k}_u > b_{in_i} + B_{in_i}\,x^0$ la contrainte est dite non satisfaite au point $\mathbf{k}_u$.

Si une contrainte est inactive il est possible de se déplacer dans n'importe quelle direction tandis que cette contrainte reste satisfaite. En revanche, pour une contrainte saturée, les directions de déplacement admissible sont restreintes.

**Définition :** Le cône de directions faisables pour un point $\mathbf{k}_u$ est décrit par :

$$D(\mathbf{k}_u, x) = \{d | d \neq 0, \exists \delta > 0 \; t.q. \; \forall \lambda \in (0, \delta) \; A_{in}(\mathbf{k}_u + \lambda d) \leq b_{in} + B_{in} x\} \tag{2.82}$$

Tout vecteur $d \in D(\mathbf{k}_u, x)$ est dit une *direction faisable*■

**Définition :** Pour la fonction de coût $J(\mathbf{k}_u, x)$, le cône de directions de descente est décrit par :

$$F(\mathbf{k}_u, x) = \{d | \exists \delta > 0 \; t.q. \; \forall \lambda \in (0, \delta) \; J(\mathbf{k}_u + \lambda d, x) < J(\mathbf{k}_u, x)\} \tag{2.83}$$

Tout vecteur $d \in F(\mathbf{k}_u, x)$ est dit une *direction de descente*■

En se basant sur le fait que si $\nabla J(\mathbf{k}_u^*)^{\mathrm{T}} d < 0$ alors $d$ est une direction de descente, un résultat (non crucial pour la suite des développements mais réutilisé dans l'Annexe 1) décrivant les conditions d'optimalité est donné par le théorème suivant [BSS93].

> **Théorème 2.1:** Si $\mathbf{k}_u^*$ est une solution optimale pour la minimisation de (2.80) sous les contraintes (2.81) pour $x = x_0$, alors $F_0(\mathbf{k}_u^*, x_0) \cap D(\mathbf{k}_u^*, x_0) = \varnothing$ avec $F_0(\mathbf{k}_u^*, x) = \{d | \nabla J(\mathbf{k}_u^*, x_0)^T d < 0\}$ et $D(\mathbf{k}_u^*, x_0)$ comme en (2.82). Réciproquement, si $F_0(\mathbf{k}_u^*, x_0) \cap D(\mathbf{k}_u^*, x_0) = \varnothing$ et s'il existe un $\varepsilon$ voisinage - $V_\varepsilon(\mathbf{k}_u^*), \varepsilon > 0$, tel que $d = (\mathbf{k}_u^* - \mathbf{k}_u) \in D(\mathbf{k}_u^*, x_0)$ pour tout $\mathbf{k}_u^* \in S(x_0) \cap V_\varepsilon(\mathbf{k}_u^*)$ et $S(x_0) = \{\mathbf{k}_u | A_{in}\mathbf{k}_u \leq b_{in} + B_{in} x_0\}$, alors $\mathbf{k}_u^*$ est l'optimum pour (2.80)-(2.81) ■

Pour détailler les conditions d'optimalité, on peut partir de l'observation que, pour un certain point admissible, il existe des directions faisables qui s'éloignent des contraintes actives. Si $\mathbf{k}_u^*$ représente l'optimum, il faut garantir que, pour ce point, une telle direction n'est pas une direction de descente. Ainsi, il faut déterminer une condition pour garantir que $\nabla J(\mathbf{k}_u^*)^T d \geq 0$ pour toute direction $d$ avec $\overline{A}_{in} d \leq 0$ ou $\overline{A}_{in}$ correspond au sous-ensemble de contraintes saturées par $\mathbf{k}_u^*$. On sait que pour un point optimal, les conditions nécessaires du premier ordre imposent que le gradient soit une combinaison linéaire de lignes de $\overline{A}_{in}$ :

$$\nabla J(\mathbf{k}_u^*) = -\overline{A}_{in}^T * \mu^* \tag{2.84}$$

avec $\mu$ le vecteur des multiplicateurs Lagrange [5]. Les conditions d'optimalité recherchées doivent tenir compte du fait que :

$$\nabla J(\mathbf{k}_u^*)^T d = -\mu^{*T} \overline{A}_{in} d \geq 0 \tag{2.85}$$

En sachant que $\overline{A}_{in} d \leq 0$, il est évident que (2.85) est uniquement satisfait si tous les multiplicateurs sont non-negatifs :

$$\mu^* \geq 0$$

Cette relation permet d'exprimer les conditions d'optimalité.

*Conditions nécessaires d'optimalité :*

- Contraintes : $A_{in}\mathbf{k}_u^* \leq b_{in} + B_{in} x$ parmi lesquelles $\overline{A}_{in}\mathbf{k}_u^* = \overline{b}_{in} + \overline{B}_{in} x$ (2.86)
- Conditions du premier ordre : $\nabla J(\mathbf{k}_u^*) = -\overline{A}_{in}^T * \mu^*$ (2.87)
- Multiplicateurs de Lagrange : $\mu^* \geq 0$ (2.88)
- Conditions du deuxième ordre : $\overline{Z}^T \nabla^2 J(\mathbf{k}_u^*) \overline{Z} = \overline{Z}^T H \overline{Z} \geq 0$ (2.89)

[5] Il est possible de définir des multiplicateurs de Lagrange à valeurs nulles pour les contraintes inactives de telle sorte que (2.84) soit réécrite sous la forme : $\nabla J(\mathbf{k}_u^*) = -A_{in}^T * \mu^*$.

*Conditions suffisantes d'optimalité :*

- Contraintes : $A_{in}\mathbf{k}_u^* \le b_{in} + B_{in}x$ avec $\overline{A}_{in}\mathbf{k}_u^* = \overline{b}_{in} + \overline{B}_{in}x$ (2.90)
- Conditions du premier ordre : $\nabla J(\mathbf{k}_u^*) = -\overline{A}_{in}^T * \mu^*$ (2.91)
- Multiplicateurs de Lagrange : $\mu^* \ge 0$ (2.92)
- Conditions du deuxième ordre : $\overline{Z}^T \nabla^2 J(\mathbf{k}_u^*)\overline{Z} = \overline{Z}^T H\overline{Z} > 0$ (2.93)

### 2.4.4 Cas avec contraintes mixtes

En réunifiant tous les problèmes traités précédemment, on retrouve la formulation qui constituait l'objectif des développements sur les conditions d'optimalité :

$$\begin{aligned} &\arg\min_{\mathbf{k}_u} 0{,}5\,\mathbf{k}_u^{\mathrm{T}} H\mathbf{k}_u + x^{\mathrm{T}} F\,\mathbf{k}_u \\ &\text{sous les contraintes} : \begin{cases} A_{in}\mathbf{k}_u \le B_{in}x + b_{in} \\ A_{eq}\mathbf{k}_u = B_{eq}x + b_{eq} \end{cases} \end{aligned} \tag{2.94}$$

Mentionnons avant tout que la complexité des problèmes de programmation quadratique de ce type dépend largement de la nature des matrices qui composent le Hessien [6]. La commande prédictive bénéficie d'un avantage important, à savoir le fait que les problèmes QP associés sont caractérisés par un Hessien défini positif grâce à la construction de la fonction de coût (2.16) et (2.38) et en conséquence les conditions d'optimalité du deuxième ordre sont vérifiées.
Les conditions d'optimalité du problème (2.94) sont résumées par les conditions Karush-Kuhn-Tucker (KKT) [7] :

- La faisabilité primale $\begin{aligned} A_{in}\mathbf{k}_u^* &\le B_{in}x + b_{in} \\ A_{eq}\mathbf{k}_u^* &= B_{eq}x + b_{eq} \end{aligned}$
- La faisabilité duale $\begin{aligned} &H\mathbf{k}_u^* + F^T x + A_{eq}^T\lambda + A_{in}^T\mu = 0 \\ &\mu \ge 0 \end{aligned}$
- La relaxation complémentaire – (*complementary slackness*) $\mu^T (A_{in}\mathbf{k}_u^* - B_{in}x - b_{in}) = 0$

## *2.5 Résolution de la commande prédictive par des programmes d'optimisation en ligne*

On a vu aux paragraphes 2.2 et 2.3 que la loi de commande prédictive implique la résolution d'un problème de programmation quadratique différent à chaque pas d'échantillonnage. Au paragraphe 2.4, les conditions d'optimalité pour une solution candidate ont été décrites. On examine désormais ici les algorithmes effectifs pour la résolution d'un tel problème.

**Définition :** On considère qu'un algorithme de commande prédictive se base sur des méthodes *en ligne* si l'implémentation prend en compte les mesures courantes des paramètres de contexte et fait appel à des programmes itératifs pour résoudre des problèmes classiques de programmation quadratique :

$$\begin{aligned} &\arg\min_{\mathbf{k}_u} 0{,}5\,\mathbf{k}_u^{\mathrm{T}} H\mathbf{k}_u + \bar{f}^{\mathrm{T}}\,\mathbf{k}_u \\ &\text{sous les contraintes} : \begin{cases} A_{in}\mathbf{k}_u \le \overline{b}_{in} \\ A_{eq}\mathbf{k}_u = \overline{b}_{eq} \end{cases} \end{aligned} \tag{2.95}$$

[6] Il est fait mention dans la littérature ([BSS93]) que le fait que le Hessien possède une valeur propre négative transforme le problème QP en un problème « NP-difficile ».

[7] Le théorème 2.1 propose des conditions d'optimalité qui, à l'aide de la qualification de contraintes d'Abadie et du lemme de Farkas [BSS93], trouvent une expression compacte dans les conditions d'optimalité KKT. Ce développement est standard pour la littérature concernant les problèmes de programmation non-linéaire.

avec $\bar{f} = F^T p_t ; \bar{b}_{in} = b_{in} + B_{in} p_t ; \bar{b}_{eq} = b_{eq} + B_{eq} p_t$, $p_t$ paramètre de contexte. Ces méthodes en-ligne doivent être différentiées d'une autre classe de méthodes dites explicites qui construisent hors ligne la fonction $\mathbf{k}_u = \mathbf{k}_u(p_t)$ et se résume ensuite à son évaluation en temps réel.

Les implémentations de la commande prédictive avec contraintes mixtes par des méthodes en-ligne constituent un sujet arrivé à maturité dans la littérature. En fait les applications jusqu'au début des années 2000 ont été basées exclusivement sur ce type d'approche (essentiellement des méthodes d'ensemble actif), la croissance de la puissance de calcul élargissant les domaines d'applications à des processus des plus en plus rapides. Des innovations en terme de procédure sont ensuite intervenues avec la parution d'algorithmes fiables du point intérieur [NN94]. L'objectif des points ci-dessous est de décrire les méthodes les plus classiques en mentionnant leurs avantages et inconvénients.

### 2.5.1 Ensemble actif

Il s'agit ici d'une des méthodes les plus connues pour résoudre les problèmes QP, qui doit son nom au fait que la procédure essaie itérativement de trouver la séparation entre l'ensemble de contraintes actives et inactives pour la solution optimale par la résolution d'une suite de problèmes QP avec contraintes égalité. Comme on l'a vu auparavant, les problèmes QP avec contraintes égalité se résument par projection à la résolution d'un problème sans contraintes, ce qui constitue un avantage certain en terme de temps de calcul. Seules les idées de base sont présentées ci-dessous, les détails peuvent être consultés dans [FLE81].

Si l'on connaît un point initial faisable $\mathbf{k}_u^0$ pour les contraintes :

$$\begin{cases} A_{in}\mathbf{k}_u^0 \le \bar{b}_{in} \\ A_{eq}\mathbf{k}_u^0 = \bar{b}_{eq} \end{cases} \tag{2.96}$$

on peut alors identifier les inégalités saturées et ensuite construire une matrice $A_{sat}$ et un vecteur $b_{sat}$ en regroupant toutes les contraintes égalité mais aussi les composantes de $A_{in}, \bar{b}_{in}$ qui correspondent aux inégalités saturées. Le résultat donne :

$$A_{sat}\mathbf{k}_u^0 = b_{sat} \tag{2.97}$$

Le problème d'optimisation quadratique :

$$\begin{array}{l} \arg\min\limits_{\mathbf{k}_u} 0{,}5\,\mathbf{k}_u^{\mathrm{T}} H \mathbf{k}_u + \bar{f}^{\mathrm{T}}\,\mathbf{k}_u \\ \text{sous les contraintes} : A_{sat}\mathbf{k}_u = b_{sat} \end{array} \tag{2.98}$$

peut être résolu et en supposant que la solution soit $\mathbf{k}_u^1$, un test d'optimalité globale doit être réalisé. Une telle opération revient à la vérification de la nonnégativité des multiplicateurs de Lagrange pour (2.98), $\lambda_i \ge 0$. Si ces conditions ne sont pas accomplies, une contrainte parmi celles associées à un coefficient de Lagrange négatif est éliminée de l'ensemble actif et la procédure est réitérée. Il existe plusieurs possibilités quant au choix de la contrainte à éliminer, un choix possible étant d'éliminer celle ayant le coefficient de Lagrange qui fournit la plus petite valeur.

**Remarque :** Si la solution $\mathbf{k}_u^1$ se trouve à l'extérieur du domaine faisable décrit par l'ensemble complet des contraintes (2.95), elle sera remplacée par la combinaison linéaire :

$$\mathbf{k}_u^{10} = \mu_1\mathbf{k}_u^0 + \mu_2\mathbf{k}_u^1, \quad 0 \le \mu_1 \le 1, \quad 0 \le \mu_2 \le 1, \quad \mu_1 + \mu_2 = 1 \tag{2.99}$$

qui se trouve sur la frontière. Le point $\mathbf{k}_u^{10}$ est faisable mais il lui correspond certaines contraintes saturées qui précédemment étaient inactives. Celles-ci sont ajoutées à l'ensemble actif et la procédure est réitérée.

**Remarque :** Trouver un point initial faisable est un problème difficile en lui-même. Une idée qui tient compte de la structure de l'optimisation est de choisir comme point initial $\mathbf{k}_u^0$ à l'instant $t+1$ la séquence :

$$\mathbf{k}_u^0(t+1) = \begin{bmatrix} u_{t+1|t}^T & \cdots & u_{t+N_y-1|t}^T & 0 \end{bmatrix}^T \tag{2.100}$$

en réutilisant une partie de la solution optimale au pas précédent :

$$\mathbf{k}_u^*(t) = \begin{bmatrix} u_{t|t}^T & \cdots & u_{t+N_y-1|t}^T \end{bmatrix}^T \tag{2.101}$$

Ceci peut malgré tout s'avérer être une combinaison infaisable si la nature de l'ensemble de contraintes ne permet pas de garantir la prolongation de la séquence de commande précédente. Une méthode de détermination d'un point faisable peut en de tels cas recourir à un problème de programmation linéaire :

$$\begin{aligned} & z^* = \min_{z,\mathbf{k}_u} z \\ & \text{sous les contraintes :} \begin{cases} A_{in}\mathbf{k}_u + \mathbf{1}z \le \overline{b}_{in} \\ A_{eq}\mathbf{k}_u + \mathbf{1}z = \overline{b}_{eq} \end{cases} \end{aligned} \tag{2.102}$$

ayant comme point de départ n'importe quel $\mathbf{k}_u^{init}$ qui satisfait $A_{eq}\mathbf{k}_u^{init} = \overline{b}_{eq}$ et $z^{init} = \max(A_{in}\mathbf{k}_u - \overline{b}_{in})$ ■

**Remarque :** Avec les méthodes d'ensemble actif, le nombre d'itérations pour trouver la solution optimale exacte est fini si le problème n'est pas susceptible de rencontrer de dégénérescence [FLE81]. Même si dans le pire cas la complexité peut être exponentielle en fonction de la taille des contraintes, pour des problèmes de taille faible les performances restent indéniables.

L'avantage de la méthode d'ensemble actif réside dans la simplicité de construction des solutions particulières à chaque itération. Dès lors, pour des ensembles de contraintes de complexité moyenne, la méthode d'ensemble actif reste une des solutions les plus performantes pour les algorithmes prédictifs en ligne. Les méthodes d'ensemble actif ne sont pas recommandées pour des problèmes de grande taille car le nombre d'itérations peut augmenter significativement à cause du nombre élevé de combinaisons possibles. Une autre remarque pratique est liée au fait que la majorité des solveurs existants utilise des matrices denses.

### 2.5.2 Problème linéaire complémentaire (Linear Complementary Problem - LCP)

Les performances en terme de charge de programmation des stratégies résolvant des problèmes de programmation linéaire à l'aide des méthodes par pivot (le succès de la méthode du Simplexe en est une preuve) sont bien connues. Leur application pour des problèmes d'optimisation quadratique (2.36), (2.47) est envisageable et une des techniques se base sur l'équivalence d'un problème QP avec un problème linéaire complémentaire (LCP) ([BSS93]).

**Définition :** Soit un vecteur $q \in \Re^d$ et une matrice $M \in \Re^{d\times d}$. Un problème linéaire complémentaire (LCP) consiste à trouver les vecteurs $w$ et $z$ tels que :

$$\begin{aligned} & w - Mz = q \\ & w_i \ge 0; z_i \ge 0 \quad \text{pour} \quad i = 1,\ldots,d \\ & w_i z_i = 0 \quad \text{pour} \quad i = 1,\ldots,d \end{aligned} \tag{2.103}$$

ou conclure qu'une telle solution n'existe pas.

Il faut remarquer que le cas $q \ge 0$ accepte la solution triviale $w = q$ et $z = 0$. Dans le cas contraire, plusieurs algorithmes par pivot existent pour les problèmes LCP parmi lesquels celui dû à Lemke [LEM68].

En revenant au problème d'optimisation lié à la commande prédictive avec des procédures en ligne (2.95), et afin de retrouver une formulation équivalente LCP, une première transformation doit être établie pour obtenir un formulation exclusivement avec contraintes de type inégalité. Dans cette direction, il est important de décrire les directions admissibles pour le problème d'optimisation, qui doivent être orthogonales aux lignes de $A_{eq}$. Soit $Z_{eq}$ une matrice qui satisfait :

$$A_{eq}Z_{eq} = 0 \;\text{ et }\; \text{rang}(Z_{eq}) = N\cdot m - \text{rang}(A_{eq}) \tag{2.104}$$

On peut donc écrire le vecteur de variables de décision sous la forme :

$$\mathbf{k}_u = A_{eq}^T(A_{eq}A_{eq}^T)^{-1}\overline{b}_{eq} + Z_{eq}\widetilde{\mathbf{k}}_u \tag{2.105}$$

et avec ce changement de variable, le problème d'optimisation (2.95) à l'instant $t$ peut être réécrit :

$$\begin{aligned} &\arg\min_{\tilde{\mathbf{k}}_u} = 0{,}5\,\tilde{\mathbf{k}}_u^T \tilde{H} \tilde{\mathbf{k}}_u + \tilde{f}^T \tilde{\mathbf{k}}_u \\ &\tilde{A}_{in}\tilde{\mathbf{k}}_u \le \tilde{b}_{in} \end{aligned} \tag{2.106}$$

Une restriction à l'emploi de techniques basées sur la stratégie LCP est la positivité des arguments du problème d'optimisation. Si tous les signaux réels liés aux systèmes sont bornés, on peut retrouver un vecteur qui regroupe les limites inférieures $\mathbf{k}_u^{\min} \le \mathbf{k}_u$ pour (2.47)[8]. Ceci implique l'existence d'un vecteur correspondant pour les variables de décision effective $\tilde{\mathbf{k}}_u^{\min} \le \tilde{\mathbf{k}}_u$. Ensuite, en opérant un nouveau changement de variable (en fait un simple décalage) : $\tilde{\mathbf{k}}_u = \tilde{\mathbf{k}}_u^{\min} + \hat{\mathbf{k}}_u$, on peut obtenir, en ajustant les matrices et les vecteurs correspondants $\tilde{H} \leftrightarrow \hat{H}; \tilde{f} \leftrightarrow \hat{f}; \tilde{A}_{in} \leftrightarrow \hat{A}_{in}; \tilde{b}_{in} \leftrightarrow \hat{b}_{in}$, le problème d'optimisation quadratique :

$$\begin{aligned} &\arg\min_{\hat{\mathbf{k}}_u} = 0{,}5\,\hat{\mathbf{k}}_u^T \hat{H} \hat{\mathbf{k}}_u + \hat{f}^T \hat{\mathbf{k}}_u \\ &\text{sous les contraintes :} \begin{cases} \hat{A}_{in}\hat{\mathbf{k}}_u \le \hat{b}_{in} \\ \hat{\mathbf{k}}_u \le 0 \end{cases} \end{aligned} \tag{2.107}$$

Les conditions KKT sont :

$$\begin{aligned} &\hat{A}_{in}\hat{\mathbf{k}}_u + y = \hat{b}_{in} \\ &-\hat{H}\hat{\mathbf{k}}_u - \hat{A}_{in}^T\lambda_1 + \lambda_2 = \hat{f} \\ &\hat{\mathbf{k}}_u^T\lambda_2 = 0, \lambda_1^T y = 0 \\ &\hat{\mathbf{k}}_u, y, \lambda_1, \lambda_2 \ge 0 \end{aligned} \tag{2.108}$$

avec les vecteurs des multiplicateurs de Lagrange et le vecteur des variables de relaxation. En notant :

$$M = \begin{bmatrix} 0 & -\hat{A}_{in} \\ \hat{A}_{in}^T & \hat{H} \end{bmatrix}, q = \begin{bmatrix} \hat{b}_{in} \\ \hat{f} \end{bmatrix}, w = \begin{bmatrix} y \\ \lambda_2 \end{bmatrix}, z = \begin{bmatrix} \lambda_1 \\ \hat{\mathbf{k}}_u \end{bmatrix} \tag{2.109}$$

les conditions KKT sont décrites comme de façon similaire aux relations obtenues en stratégie LCP :

$$w - Mz = q, w^T z = 0, w \ge 0, z \ge 0 \tag{2.110}$$

et les algorithmes spécialisés [LEM68] peuvent être employés pour obtenir la séquence optimale de commande à chaque pas d'échantillonnage.

Une telle application des solveurs LCP pour des problèmes de commande prédictive a été rapportée par Camacho [CB03] en soulignant le fait que les performances sont dépendantes de l'ensemble de contraintes considéré. Pour ce motif il est très important d'éliminer les contraintes qui n'interviennent pas dans la définition de la zone faisable.

---

[8] Il est indiqué de construire le vecteur des limitations inférieures $\mathbf{k}_u^{\min}$ pour tous les paramètres possibles dans (2.47). Ceci peut être fait par la résolution d'une séquence finie de problèmes de programmation linéaire :

$$\left.\begin{aligned} &\mathbf{k}_{u_i}^{\min} = \min_{\mathbf{k}_u, p} \begin{bmatrix} \underbrace{0 \cdots 0}_{i-1} & 1 & \underbrace{0 \cdots 0}_{Nm-i} \end{bmatrix} \\ &\text{sous les contraintes :} \begin{cases} A_{in}\mathbf{k}_u - B_{in}p_t \le b_{in} \\ A_{eq}\mathbf{k}_u - B_{eq}p_t = b_{eq} \end{cases} \end{aligned}\right\} \Rightarrow \mathbf{k}_u^{\min} = \begin{bmatrix} k_{u_1}^{\min} \\ \vdots \\ k_{u_i}^{\min} \\ \vdots \\ k_{u_{Nm}}^{\min} \end{bmatrix}$$

Si un tel problème présente un optimum non borné, la procédure de transformation en un problème LCP va échouer car les contraintes de positivité ne sont pas satisfaites.

### 2.5.3 Directions faisables

On trouve dans la littérature dédiée aux problèmes d'optimisation non-linéaire certains algorithmes qui peuvent être utilisés également lors d'applications en commande prédictive. Une de ces techniques est celle de la « méthode des directions faisables » qui, à partir d'un point faisable, trouve itérativement une suite d'autres points faisables améliorant le critère [9].

La stratégie consiste à partir d'un point faisable $\mathbf{k}_u^j$, puis trouver une direction faisable $d_j$ telle qu'il existe $\lambda > 0$ pour que $\mathbf{k}_u^j + \lambda d_j$ soit faisable et la fonction objectif pour $\mathbf{k}_u^j + \lambda d_j$ meilleure que celle correspondant à $\mathbf{k}_u^j$. Une fois la direction faisable construite, une optimisation unidimensionnelle est résolue pour déterminer la valeur optimale $\lambda^*$ de déplacement le long de $d_j$. Ce choix conduit à un point $\mathbf{k}_u^{j+1}$ et la procédure est réitérée. Pour choisir une direction qui se prête à l'application de cette stratégie, une idée peut être d'utiliser la direction selon la descente la plus profonde, qui coïncide avec celle du gradient négatif. Malgré tout, la présence de contraintes fait qu'un tel choix peut être infaisable. La méthode de Rosen [ROS60] contourne ce problème en utilisant un gradient projeté comme direction faisable.

**Définition :** Un matrice $P \in \Re^{r \times r}$ est dite matrice de projection si $P = P^T, P\,P = P$ ■

Pour la commande prédictive le but est de minimiser :

$$J(\mathbf{k}_u) = 0{,}5\,\mathbf{k}_u^T\, H \mathbf{k}_u + f^T \mathbf{k}_u$$
$$\text{sous les contraintes} : \begin{cases} A_{in}\mathbf{k}_u \le \bar{b}_{in} \\ A_{eq}\mathbf{k}_u = \bar{b}_{eq} \end{cases} \tag{2.111}$$

Pour un point faisable $\mathbf{k}_u^i$, un sous-ensemble de contraintes inégalité est vérifié en plus des contraintes égalité :

$$\bar{A}_{in}^i \mathbf{k}_u^i \le \bar{b}_{in}^i \tag{2.112}$$

de sorte que la forme compacte $E_i = \begin{bmatrix} \bar{A}_{in}^{i\,T} & A_{eq}^T \end{bmatrix}^T$ décrit les contraintes actives à l'instant $i$. Si $E_i$ est de rang plein, on peut construire la matrice de projection :

$$P = I - E_i^T (E_i E_i^T)^{-1} E_i \tag{2.113}$$

et ensuite décrire le vecteur :

$$d_i = -P\nabla J(\mathbf{k}_u^i) = -P(H\mathbf{k}_u^i + f) \tag{2.114}$$

Si $d_i \neq 0$, on dispose d'une direction faisable permettant d'améliorer la fonction de coût. La procédure peut être réitérée pour :

$$\mathbf{k}_u^{i+1} = \mathbf{k}_u^i + \lambda_{\max} d_i \tag{2.115}$$

avec $\lambda_{\max}$ choisi tel que $J(\mathbf{k}_u^i) \ge J(\mathbf{k}_u^{i+1})$ et $\mathbf{k}_u^{i+1}$ faisable.

La méthode des directions faisables travaille seulement sur le problème primal, et comme la faisabilité est maintenue pendant le processus d'optimisation, cette clase de procédures est dénommée « méthodes primales ». Il est généralement facile de démontrer leur convergence vers les solutions KKT. On trouve des exemples de ce type de stratégies différant seulement par la façon de construire la direction faisable comme la méthode de Beale [BEA55], la méthode de Rosen résumée ci-dessus [ROS60], la méthode du gradient réduit de Wolfe [WOL63].

### 2.5.4 Point intérieur

Ces algorithmes font partie de la classe dite « *primal-dual path following methods* », qui utilise une fonction barrière et des algorithmes de type Newton pour mettre en œuvre des procédures en temps polynomial. Leurs

[9] Le fait que ces méthodes induisent toujours des séquences intermédiaires à l'intérieur du domaine faisable peut représenter un avantage non négligeable lors des applications temps réel où les limitations de temps de calcul font que les solveurs sont obligés de proposer une solution qui ne respecte pas les conditions d'arrêt de l'algorithme. A titre de comparaison, les méthodes LCP et d'ensemble actif pratiquent l'évolution à l'extérieur du domaine faisable durant la recherche du point optimal et donc les séquences intermédiaires ne sont pas utilisables avant la fin de l'optimisation.

racines ne sont historiquement pas très éloignées [KAR84] et leur développement a été prodigieux. Un aperçu de l'état de l'art sur ce sujet peut être trouvé dans des ouvrages très complets comme par exemple ([NN94], [YE89]). Ces algorithmes peuvent être vus comme une généralisation des méthodes d'optimisation non linéaire classiques pour les problèmes d'optimisation avec contraintes convexes. Leurs performances font qu'ils sont particulièrement adaptés aux problèmes d'optimisation convexe de grandes dimensions.

Si l'on considère comme formulation de départ le problème lié à la commande prédictive :

$$\begin{array}{l} \arg\min\limits_{\mathbf{k}_u} = 0{,}5\,\mathbf{k}_u^T\, H\,\mathbf{k}_u + \bar{f}^T\mathbf{k}_u \\ \text{sous les contraintes : } A_{in}\mathbf{k}_u \le \bar{b}_{in} \end{array} \tag{2.116}$$

la première étape consiste à construire une fonction barrière (ou de pénalité intérieure) qui devient infinie sur la frontière du domaine faisable. Des choix typiques pour cette fonction peuvent être :

$$g(\mathbf{k}_u) = -\sum_i \log(b_i - a_i^T\mathbf{k}_u) \text{ ou } g(\mathbf{k}_u) = \frac{1}{\sum\limits_i (b_i - a_i^T\mathbf{k}_u)} \tag{2.117}$$

avec $a_i^T$ la $i$-ème ligne de $A_{in}$ et $b_i$ la $i$-eme composante du vecteur $\bar{b}_{in}$. En utilisant cette fonction, on peut remplacer l'optimisation (2.116) par :

$$\mathbf{k}_u^*(\gamma) = \arg\min_{\mathbf{k}_u} = \gamma(0{,}5\,\mathbf{k}_u^T\, H\,\mathbf{k}_u + \bar{f}^T\mathbf{k}_u) + g(\mathbf{k}_u) \tag{2.118}$$

avec $\gamma$ un scalaire jouant le rôle de pondération. La formulation (2.118) caractérise un problème d'optimisation non linéaire sans contraintes qui peut être résolu par des méthodes classiques de type Newton ([BSS93]). L'intégration de la fonction barrière dans le critère d'optimisation évite le débordement du domaine faisable durant la recherche de la solution optimale, en garantissant donc le fait que les solutions intermédiaires sont faisables. Si la solution optimale est $\mathbf{k}_u^*$ pour le problème initial, en augmentant la pondération $\gamma$, la solution de (2.118) va évoluer de sorte que $\mathbf{k}_u^*(\gamma) \to \mathbf{k}_u^*$ quand $\gamma \to \infty$, améliorant donc la qualité de la solution jusqu'à la limite $\left\|\mathbf{k}_u^*(\gamma) - \mathbf{k}_u^*\right\| < \varepsilon$ avec $\varepsilon$ suffisamment petit. Le chemin suivi par $\mathbf{k}_u^*(\gamma)$ s'appelle la « trajectoire centrale ». Les méthodes les plus efficaces travaillent simultanément sur le problème primal et dual.

Une autre stratégie pour les méthodes d'optimisation du point intérieur [BN01], [GSD04], [WRI97b] part du problème QP initial (2.95) et de la formulation des conditions d'optimalité KKT :

$$\begin{array}{l} H\,\mathbf{k}_u + \bar{f} + A_{eq}^T\nu + A_{in}^T\mu = 0 \\ A_{eq}\mathbf{k}_u = \bar{b}_{eq} \\ A_{in}\mathbf{k}_u + \xi = \bar{b}_{in} \\ \mu \ge 0; \xi \ge 0; \mu^T\xi = 0 \end{array} \tag{2.119}$$

avec $\nu, \mu$ les vecteurs des multiplicateurs de Lagrange et $\xi$ le vecteur des variables de relaxation qui permet la transformation des inégalités en égalités. Ensuite ces équations sont transcrites sous forme matricielle semblable à la forme prise en compte dans les techniques LCP :

$$\begin{array}{l} \begin{bmatrix} H & A_{eq}^T & A_{in}^T \\ -A_{eq} & 0 & 0 \\ -A_{in} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{k}_u \\ \nu \\ \mu \end{bmatrix} + \begin{bmatrix} \bar{f} \\ \bar{b}_{eq} \\ \bar{b}_{in} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \xi \end{bmatrix} \\ \nu \ge 0; \xi \ge 0; \mu^T\xi = 0 \end{array} \tag{2.120}$$

avec la matrice dans le membre de gauche positive définie. La méthode débouche au final sur la résolution d'un système d'équations non linéaires :

$$F(\mathbf{k}_u, \nu, \mu, \xi) = \begin{bmatrix} H\,\mathbf{k}_u + A_{eq}^T \nu + A_{in}^T \mu + \bar{f} \\ -A_{eq}\mathbf{k}_u + \bar{b}_{eq} \\ -A_{in}\mathbf{k}_u - \xi + \bar{b}_{in} \\ \Xi \mathrm{M} e = 0 \end{bmatrix} = 0 \qquad (2.121)$$

$$\mu \geq 0; \xi \geq 0$$

avec :

$$\Xi = diag\{\xi_1, \ldots, \xi_r\};$$
$$\mathrm{M} = diag\{\mu_1, \ldots, \mu_r\};$$
$$e = [1 \quad \ldots \quad 1]^T$$

Des méthodes de type Newton peuvent être employées en utilisant à chaque itération une approximation linéaire de $F$. Un exemple connu est illustré par la méthode du *point intérieur infaisable* IIP [WRI97b], pour lequel les solutions itératives peuvent être *infaisables* par rapport à l'ensemble de contraintes (2.95) mais intérieures au quadrant positif $\mu \geq 0; \xi \geq 0$. Une mesure de l'infaisabilité des solutions intermédiaires est représentée par l'écart de dualité :

$$\gamma = \mu^T \xi / r \qquad (2.122)$$

avec $r$ le nombre d'inégalités dans l'ensemble de contraintes. Cet écart est aussi utilisé comme une mesure de l'optimalité même si la faisabilité est atteinte seulement à la limite.

**Remarque :** Les méthodes du point intérieur deviennent de plus en plus populaires, au détriment des méthodes d'ensemble actif, car leur facteur de convergence est polynomial. En contrepartie, la charge de calcul par itération est plus importante, ce qui les rend inadaptées pour des problèmes de petite taille. Les performances sont fortement liées aux précautions prises pour éviter les problèmes de conditionnement numérique.

### 2.5.5 Aspects liés à la charge de calcul

La structure du problème QP a des implications fortes sur les performances des programmes d'optimisation en ligne. Il peut alors s'avérer intéressant, avant l'implémentation, de réorganisation des variables afin d'obtenir une forme « bloc-bande » pour les matrices intervenant dans l'optimisation, car les méthodes d'ensemble actif et du point intérieur peuvent profiter d'une telle formulation [WRI97a]. La structure du problème de commande prédictive permet en fait ce type de manipulation par l'introduction des équations de la dynamique dans l'ensemble de contraintes (ajout des égalités) de sorte que les prédictions soient intégrées comme variables de décision dans le problème d'optimisation. On a donc d'une part une forme compacte et dense, d'autre part une alternative « creuse » mais avec une structure bande, qui ensuite permet l'utilisation de programmes de factorisation moins coûteux (en utilisant aussi le fait que les matrices traitées ne varient pas sensiblement d'une itération à l'autre).

La différence entre ces deux alternatives réside dans le fait que la factorisation d'une matrice avec une demi-bande $d$ et une dimension totale $N_d(d+1)$ est $O(N_d(d+1)^3)$ pour une matrice creuse, comparée à $O(N_d^3(d+1)^3)$ pour une matrice dense de même dimension [WRI97a]. Pour une loi prédictive, avec un horizon de prédiction égal à l'horizon de commande $N$, l'utilisation d'une formulation « bloc-bande » est préférable si :

$$N^2(m + \nu/2) > (2n + 3m + \nu/2)^3 \qquad (2.123)$$

avec $n$ la dimension du vecteur d'état, $m$ la dimension du vecteur d'entrée et $\nu$ le nombre de contraintes inégalité par étape de prédiction [MAC02]. Si au contraire l'horizon de commande est beaucoup plus court que l'horizon de prédiction, la version dense est favorisée. D'autres aspects, comme la difficulté de construire une solution initiale faisable, peuvent jouer un rôle important.

La présentation des méthodes de résolution de problèmes d'optimisation quadratique pourrait considérer aussi d'autres techniques comme celle des plans coupants ou la méthode des ellipsoïdes, mais leur application aux problèmes de commande prédictive n'a pas donné lieu à beaucoup d'implantations et n'apporte pas d'avantages flagrants en comparaison des méthodes énumérées précédemment.

## 2.6 L'approche LMI : une alternative en ligne à la commande prédictive par QP

Le paragraphe précédent a permis de montrer que les méthodes du point intérieur constituaient des techniques d'optimisation très performantes. La classe de applications qui peuvent bénéficier de leurs performances se caractérise par les problèmes basés sur la programmation semi-définie. Le développement de ces méthodes du point intérieur, en liaison avec la formulation des problèmes fondamentaux d'Automatique en terme d'inégalités matricielles linéaires (LMI), a ouvert un domaine de recherche très productif. La commande prédictive n'échappe pas à la règle et, par le biais de LMI, de nouvelles constructions deviennent possibles pour l'implémentation de lois de commande optimales selon le principe de l'horizon glissant.

Le principe, résumé par Kothare *et al.* [KBM96] consiste à minimiser une borne supérieure de la fonction objectif à horizon infini avec des restrictions sur l'entrée et la sortie. La formulation ne conduit plus à la résolution d'un problème QP mais se réduit à la solution d'un problème d'optimisation convexe à base d'inégalités matricielles linéaires. Des résultats existent garantissant que la stratégie proposée assure la stabilité du système. En outre, cette formulation permet d'optimiser de façon significative les résultats de problèmes de commande robuste développés par le biais de la théorie des LMIs.

Pour un système :

$$x_{t+1} = Ax_t + Bu_t \tag{2.124}$$

la procédure est basée sur un critère de performance quadratique qui doit être minimisé :

$$J_t^\infty = \sum_{i=1}^{\infty} \left\{ x_{t+i|t}^T Q x_{t+i|t} + u_{t+i|t}^T R u_{t+i|t} \right\} \tag{2.125}$$

Une fonction quadratique :

$$V(x) = x^T P x \tag{2.126}$$

est supposée vérifier l'inégalité :

$$V(x_{t+i+1|t}) - V(x_{t+i|t}) \leq -\left[ x_{t+i+1|t}^T Q x_{t+i+1|t} + u_{t+i+1|t}^T R u_{t+i+1|t} \right] \tag{2.127}$$

pour tout $i \geq 0$ et $\left\{ x_{t+i|t}, u_{t+i|t} \right\}$ satisfaisant (2.124). Ensuite, en sommant (2.127) de $i = 0$ à $i = \infty$ on obtient :

$$-V(x_{t|t}) \leq -J_t^\infty \Rightarrow V(x_{t|t}) \geq -J_t^\infty \tag{2.128}$$

Il a été démontré [KBM96] que la limite supérieure $V(x_{t|t})$ de la fonction de coût peut être minimisée par l'application d'une loi de commande linéaire :

$$u_{t+i|t} = YS^{-1} x_{t+i|t}, i \geq 0 \tag{2.129}$$

avec $S > 0$ et $Y$ obtenu comme solution (si elle existe) du problème d'optimisation :

$$\begin{aligned} &\min_{\gamma, S, Y} \gamma \\ &\text{sous les contraintes :} \left\{ \begin{array}{l} \begin{bmatrix} 1 & x_{t|t}^T \\ x_{t|t} & S \end{bmatrix} \geq 0 \\ \begin{bmatrix} S & SA^T + Y^T B^T & SQ^{1/2} & Y^T R^{1/2} \\ AS + BY & S & 0 & 0 \\ Q^{1/2} S & 0 & \gamma I & 0 \\ Y^T R^{1/2} & 0 & 0 & \gamma I \end{bmatrix} \geq 0 \end{array} \right. \end{aligned} \tag{2.130}$$

La différence de conception par rapport aux stratégies prédictives énumérées précédemment réside dans le fait que le problème d'optimisation à résoudre à chaque pas d'échantillonnage possède comme argument la matrice de retour d'état $K_t$ de telle sorte que les actions de commande futures sont données par :

$$u_{t+i|t} = K_t x_{t+i|t}, i \geq 0 \tag{2.131}$$

ce qui est naturellement moins riche que la famille de commandes admises par les autres lois de commande. Cependant dans ce cas, les horizons de commande et de prédiction infinis permettent d'aboutir à de très bonnes performances grâce au principe de l'horizon glissant. Les contraintes, de type $x_t^T X x_t < 1, X > 0$, peuvent être formulées sous forme LMI de telle sorte que l'on recherche à chaque pas d'échantillonnage $t$, la valeur $K_t$ qui assure $x_{t+i|t}^T X x_{t+i|t} < 1, i > 0$. Avec la même philosophie, des contraintes sur les commandes, ou de performances ($H_2, H_\infty$) sont exprimables en termes LMI.

**Remarques :** On peut noter que cette approche se base sur l'utilisation de programmes d'optimisation spécifiques tout en essayant de ne pas s'éloigner des principes fondamentaux de la commande prédictive. Cette démarche est inverse des cas précédents pour lesquels les objectifs de commande définissaient un problème d'optimisation où l'on recherchait ensuite le meilleur algorithme de résolution. A partir de cette observation, la critique que l'on peut formuler est liée au fait que les programmes LMI minimisent une limite supérieure du critère et que cette limite risque d'être assez conservative ([KBM96] mentionne que le conservatisme est tout de même réduit). Du point de vue charge de calcul, même si les solveurs LMI sont relativement performants, les matrices manipulées sont de taille assez conséquente de sorte que ces méthodologies restent applicables avant tout pour des dynamiques relativement lentes. Les avantages de la méthode sont liés à l'utilisation des horizons infinis de prédiction, à la robustesse en stabilité et en performances. Un autre point fort de la méthode est que la faisabilité est acquise à l'instant $t$, elle sera préservée sur un horizon de prédiction infini.

## *2.7 Conclusions*

Ce chapitre a présenté les principes généraux de la commande prédictive, détaillant plus spécifiquement deux formulations, GPC et MPC, dans le cas à temps invariant, faisant intervenir des fonctions de coût quadratiques et des contraintes linéaires (mixtes – égalités et inégalités). Il est évident que la ressemblance entre la formulation (2.47) et (2.36) fait que les formulations prédictives conduisent au même type de problèmes d'optimisation quadratique, tant que la philosophie reste le même (voir la différence de formulation en comparaison avec les formulations de programmation semidéfinie). Les différences résident dans la composition du vecteur de paramètres de contexte. La paramétrisation du problème d'optimisation est issue du vecteur d'état dans le cas MPC (2.47) ou d'une séquence finie des entrée et sorties passées et des consignes futures dans le cas GPC (2.36), qui en fait représente un vecteur d'état étendu.

Le temps de calcul pour le problème d'optimisation en présence de contraintes mixtes est beaucoup plus élevé que dans le cas de contraintes égalité et les lois de commande ne peuvent pas se réduire à de simples régulateurs linéaires fixes (RST /gain par retour d'état). Pour une application temps réel, il est nécessaire que le temps de calcul soit plus petit que la période d'échantillonnage et donc le choix des programmes élaborant la séquence optimale de commande devient très sensible. Les principales méthodes existantes pour la résolution en ligne des problèmes d'optimisation prédictifs ont été décrites, montrant également que leurs performances (nombre d'itération et charge de calcul par itération) restent fortement liées à la dimension de la séquence de commande $N_u$ et aux horizons de prédiction et de contrainte de l'algorithme prédictif.

Les deux chapitres suivants reprendront cet aspect lié au temps de calcul, en accordant une attention particulière aux problèmes d'optimisation pour une loi prédictive, avec comme objectifs d'une part la compréhension de leur structure géométrique, d'autre part du point de vue applicatif la diminution de la charge informatique *en ligne* par des prétraitements opérés hors ligne.

# 3. Géométrie de la commande prédictive avec contraintes

Le chapitre précédent a permis la formulation des problèmes d'optimisation liés à la commande prédictive sous contraintes. On a pu formuler les contraintes fonctionnelles, de sûreté ou de performances imposées par un cahier des charges aux algorithmes prédictifs sous forme d'égalités et d'inégalités linéaires définissant un ensemble (polyédral) convexe. La dépendance de ces objets à un vecteur de paramètres a été mise en évidence, sans que toutefois les conséquences de cette particularité ne soient analysées. Ce chapitre a pour but l'identification des influences de la paramétrisation sur les contraintes. Etant donné le fait que l'ensemble des contraintes décrit la région faisable pour le problème d'optimisation, toute modification de sa topologie se répercute aussi sur l'expression de la séquence optimale et par conséquent sur la structure de la loi de commande.

Du point de vue pratique, les implications de la paramétrisation des contraintes sur les programmes d'optimisation en ligne ne sont pas négligeables même si jusqu'à présent les références dans la littérature sont presque inexistantes. Ce chapitre, considérant cet aspect, propose une analyse de la charge de calcul en essayant de pallier ce manque de références. Par ailleurs, les notions et résultats présentés ici peuvent être interprétés comme une façon de passer des méthodes d'optimisation en ligne exposées dans un sous-chapitre précédent aux méthodes explicites qui seront introduites au chapitre 4.

Les moyens employés pour aboutir à l'objectif proposé sont principalement des outils familiers au domaine de la géométrie informatique et plus spécifiquement des calculs polyédraux. Ce domaine, très intéressant en lui-même, apparaît dans de très nombreuses disciplines. La liaison avec l'automatique est très prononcée, en particulier pour la caractérisation des systèmes de commande avec contraintes, mais aussi pour la commande robuste lors de la description des incertitudes de modèle et des perturbations ou dans les études de stabilité basées sur la théorie des ensembles invariants. On retrouve dans divers thèmes de recherche des concepts comme ceux des diagrammes de Voronoy, de triangulation de Delaunay, des problèmes de projections ou d'énumération de facettes.

Lors des travaux liés à cette thèse, les techniques de représentations duales des domaines polyédraux (générateurs/contraintes) ont été utilisées pour les avantages qu'elles offrent dans la représentation des régions faisables pour la commande prédictive. Dans ce sens, une attention particulière a été portée aux concepts de polyèdres paramétrés et de sommets paramétrés, avec les programmes et algorithmes associés. Ces aspects de programmation polyédrale ont donné lieu à la création d'un noyau de calcul polyédral sous MATLAB, utilisé par la suite pour les enjeux de la commande prédictive. Ce chapitre propose alors une brève description des idées principales sur lesquelles s'appuient les programmes de calcul polyédraux, facilitant ensuite la présentation des résultats liés à la géométrie de la commande prédictive avec contraintes.

Ce chapitre introduit les éléments nécessaires à la compréhension de la géométrie des domaines faisables décrits par un nombre fini des contraintes. Sachant que les contraintes sont linéaires, il est nécessaire de considérer des domaines polyédraux. A partir de ces outils, l'objectif est de caractériser les redondances dans l'ensemble des contraintes, les méthodes diminuant l'impact de cette redondance sur les problèmes d'optimisation et finalement une représentation sans redondance est proposée. Le chapitre se termine par une discussion sur les éventuelles améliorations des performances des procédures d'optimisation en ligne utilisées par les lois prédictives. Le lecteur pourra également se reporter à l'Annexe 1, qui propose une démarche de classification des contraintes afin de mieux comprendre l'influence des paramètres de contexte sur la géométrie des solutions optimales et des conditions d'optimalité.

Les aspects développés dans ce chapitre se retrouvent dans les articles [OD05a], [OD05e] et [OD04c].

## *3.1 Domaines polyédraux*

Les polyèdres sont des représentations géométriques de systèmes d'équations et inéquations linéaires. Les travaux fondamentaux sur les représentations des polyèdres sont essentiellement dus à Minkovski [Sch86] qui a entrepris pour la première fois une étude systématique des ensembles convexes. Des contributions significatives ont été apportées par Farkas, Caratheodory et Weyl [Sch86]. Le point le plus important pour la suite du travail consiste à examiner les algorithmes effectifs permettant de représenter les objets polyédraux, ce qui constituera une approche originale pour la représentation des espaces faisables pour la commande prédictive. Le principal résultat du point de vue implantation numérique est la double représentation basée sur le théorème de décomposition de Motzkin [MRT53].

### 3.1.1 Cônes polyédraux

La version la plus simple d'un ensemble de contraintes est représentée par la relation :

$$A_{in}\mathbf{k}_u \le 0 \tag{3.1}$$

L'ensemble de points décrit par ce type de système d'inéquations est alors :

$$C = \left\{\mathbf{k}_u \middle| \mathbf{k}_u + \mathbf{k}_u^0 \in C, \forall \mathbf{k}_u^0 \in C\right\} = \left\{\mathbf{k}_u \middle| A_{in}\mathbf{k}_u \le 0\right\} \tag{3.2}$$

qui représente un *cône polyédral.* Il convient de s'intéresser également au plus petit cône s'écrivant comme la combinaison d'un nombre *fini* de vecteurs nommés *générateurs* :

$$C = \left\{\mathbf{k}_u \middle| \mathbf{k}_u + \mathbf{k}_u^0 \in C, \forall \mathbf{k}_u^0 \in C\right\} = \left\{\mathbf{k}_u \middle| \mathbf{k}_u = \rho_1 r_1 + \rho_2 r_2 + \ldots + \rho_{d_R} r_{d_R}, \rho_1, \ldots \rho_{d_R} \ge 0\right\} \tag{3.3}$$

Les directions faisables d'un cône $C$ sont les vecteurs $r$ vérifiant la propriété :

$$\forall\ \mathbf{k}_u^0 \in C\ ,\ (\mathbf{k}_u^0 + \rho\, r) \in C, \forall \rho \ge 0\,. \tag{3.4}$$

Une telle direction est souvent appelée *rayon* du cône C. Il faut considérer un rayon comme une direction dans laquelle *C* peut s'étendre jusqu'à l'infini, et non comme une suite finie de points.

**Définition 3.1 :** Un *rayon extrémal* d'un cône $C$ est le rayon qui ne peut pas s'exprimer par une combinaison linéaire de rayons de $C$.
L'ensemble de tous les rayons extrémaux d'un cône représente une base de générateurs minimale et ils sont uniques à une constante multiplicative près. ■

**Proposition (Farkas-Minkovski-Weyl) 3.2 :** Un cône est polyédral si et seulement s'il est généré de façon finie. ■

L'interprétation de ce résultat est que chaque système d'inégalités (3.2) possède un correspondant en termes de générateurs. Cette équivalence souligne également la différence entre la notion générale de cônes, qui peuvent être issus d'une base de générateurs infinie (Figure 3.1a), et les cônes polyédraux qui, bornés par un nombre fini d'inégalités, se caractérisent en définitive par un nombre fini de générateurs (Figure 3.1b). La preuve de ce résultat essentiel pour la représentation des cônes peut être trouvée dans [Sch86]. Même si la représentation des cônes est relativement claire, ceux-ci comportent des nuances intéressantes comme par exemple le fait qu'ils puissent inclure des directions infinies (Figure 3.1c).

**Définition 3.3 :** *L'espace de linéarité* d'un cône est défini par :

$$L = C \cap -C = \left\{\mathbf{k}_u \middle| A_{in}\mathbf{k}_u = 0\right\} \tag{3.5}$$

L'espace de linéarité représente le plus grand sous-espace linéaire contenu dans le cône initial *C.* Si la dimension de cet espace de linéarité est nulle, alors on dit que le cône $C$ est *pointu.* Suite à cette observation, on peut voir les cônes polyédraux comme une partition :

$$C = L + R \tag{3.6}$$

avec *L* l'espace de linéarité de *C* et *R* un cône pointu non vide (contenant au moins le point $\{0\}$)[10].

Pour retrouver cette même distinction (espace de linéarité *L* et cône pointu *R*) pour la représentation d'un cône polyédral *C* en termes d'un ensemble fini de rayons extrémaux, on peut rechercher l'existence éventuelle de paires de rayons $\{r, -r\} \in C$. Comme de telles paires sont équivalentes à l'évolution selon deux directions opposées, on en conclut que l'on détient alors des éléments qui font partie de l'espace de linéarité de *C.*

**Définition 3.4 :** *Une ligne* (ou un rayon bidirectionnel) faisant partie de l'espace de linéarité du cône $C$ est représentée par un vecteur $l$ vérifiant la propriété :

$$\forall \mathbf{k}_u^0 \in C\ ,\ (\mathbf{k}_u^0 + \lambda l) \in C, \forall \lambda\,. \tag{3.7}$$

[10] Le cône peut se translater dans l'espace en déplaçant ce point.

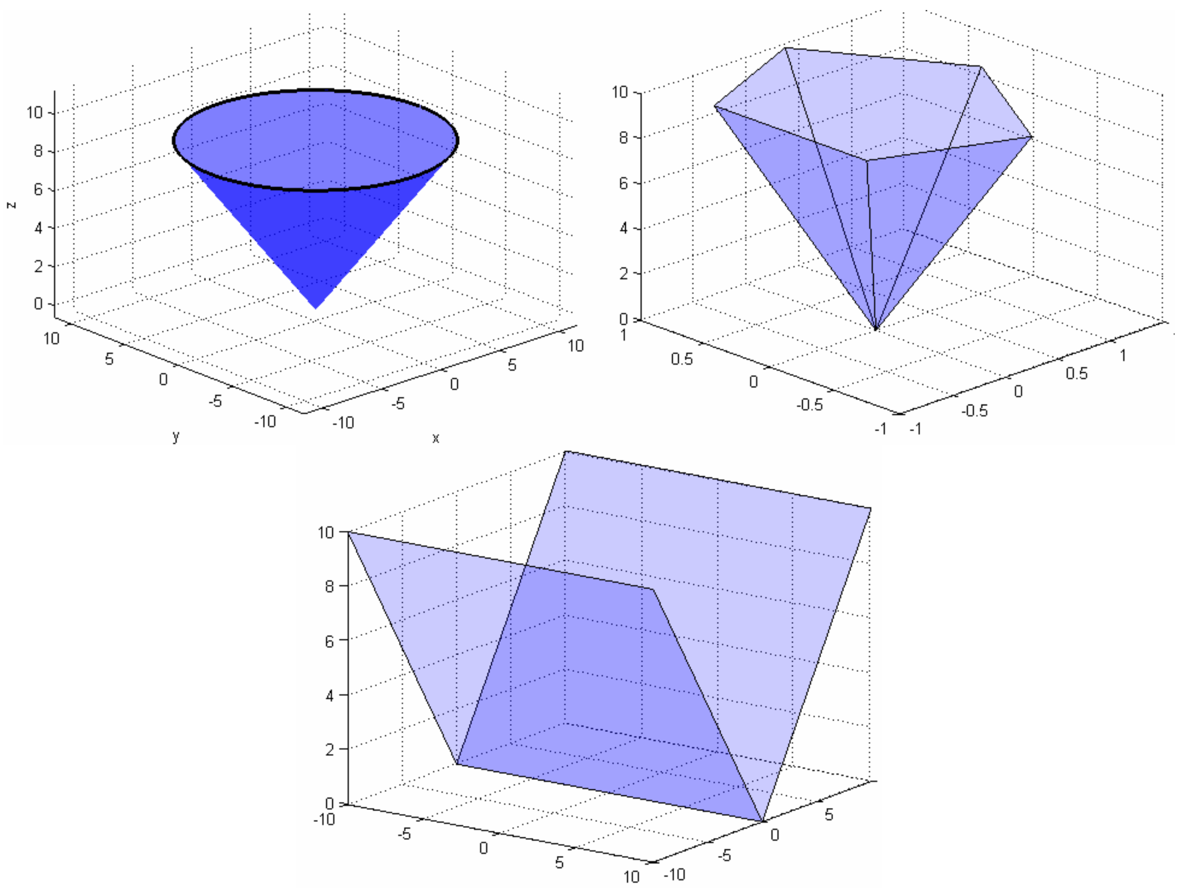



**Figure 3.1 : a) Cône issu d'une base infinie de générateurs b) Cône polyédral c) Cône polyédral incluant des lignes parmi ses générateurs.**

Tous les résultats présentés ci dessus peuvent être étendus au cas avec contraintes mixtes, car l'ajout de contraintes égalité n'affecte pas la description des générateurs. Ce résultat se retrouve aisément en formulant les égalités comme un ensemble d'inégalités.

> **Définition 3.5 :** Une inégalité $a\mathbf{k}_u \leq 0$ faisant partie de l'ensemble $A_{in}\mathbf{k}_u \leq 0$ est dite une égalité implicite (dans $A_{in}\mathbf{k}_u \leq 0$) si $a\mathbf{k}_u = 0$ pour tout $\mathbf{k}_u$ vérifiant $A_{in}\mathbf{k}_u \leq 0$. ■

Soit un ensemble de contraintes :

$$\begin{cases} A_{in}\mathbf{k}_u \leq 0 \\ A_{eq}\mathbf{k}_u = 0 \end{cases}$$

Dans l'hypothèse où les inégalités ne possèdent pas d'égalités implicites, la dimension d'un cône polyédral issu d'un tel ensemble d'inégalités et d'égalités correspond à la dimension de l'espace initial d'appartenance de $\mathbf{k}_u$ moins le rang de la matrice $A_{eq}$ des contraintes égalité.

### 3.1.2 Polytopes et polyèdres

Les cônes polyédraux étudiés jusqu'ici avaient en fait la particularité d'être engendrés par un ensemble de contraintes inégalité (ou mixtes) tel que le membre de droite des inégalités était le vecteur nul. Avant de considérer désormais le cas général de contraintes, introduisons tout d'abord un résultat important de la théorie des polyèdres.

> **Théorème (Caratheodory) 3.6 :** Soit un domaine $S \in \Re^r$. Si $\mathbf{k}_u$ fait partie de la couverture convexe[11] de $S$, alors il existe un ensemble fini de points $\{\mathbf{k}_u^1,\ldots,\mathbf{k}_u^d\} \in S$ tel que :

[11] La couverture convexe d'un domaine est le plus petit ensemble convexe qui le contient.

$$\begin{aligned} & \mathbf{k}_u \in S \Rightarrow \mathbf{k}_u = \lambda_1 \mathbf{k}_u^1 + \lambda_2 \mathbf{k}_u^2 + \ldots + \lambda_d \mathbf{k}_u^d, \\ & 0 \le \lambda_i \le 1, i = 1, \ldots d, \sum_{i=1}^{d} \lambda_i = 1 \end{aligned} \tag{3.8}$$

En fait, ceci constitue une généralisation, car le théorème de base de Cartheodory montre que n'importe quel point contenu dans $S \in \Re^p$ peut être représenté par une combinaison linéaire d'au maximum $p+1$ points de $S$. En généralisant, on obtient le résultat énoncé par le théorème (3.6). La preuve découle directement du théorème de Farkas (voir [SCH86] pour plus de détails).

**Définition 3.7 :** Un ensemble $S$ est un *polytope* convexe s'il est la couverture convexe d'un nombre fini de points. ■

Les points formant l'ensemble minimal permettant la description de la couverture convexe d'un domaine sont appelés les sommets de $S$.

**Définition 3.8 :** Un *sommet* pour un ensemble convexe $S$ correspond à tout point de $S$ qui ne permet pas une description par une combinaison linéaire des autres points distincts dans le même ensemble. ■

En réinterprétant la définition 3.7 par l'intermédiaire de la notion de sommet, on peut déduire que tout point d'un polytope est une combinaison convexe d'un nombre fini de sommets qui seront donc ses générateurs.

Toutes ces définitions sur les cônes polyédraux et les polytopes fournissent les éléments nécessaires pour une analyse exhaustive des objets géométriques décrits par des ensembles de contraintes linéaires.

**Définition 3.9 :** Un ensemble $P \in \Re^r$ représente un *polyèdre* (convexe) s'il est l'intersection d'un nombre fini de demi-espaces affines caractérisés par un ensemble d'égalités et d'inégalités linéaires :

$$P = \{\mathbf{k}_u | A_{in}\mathbf{k}_u \le b_{in}; A_{eq}\mathbf{k}_u = b_{eq}\} \subset \Re^r \tag{3.9}$$

Un polyèdre peut encore être vu comme un sous-espace de $\Re^r$ borné par un nombre fini d'hyperplans. ■

On fait intervenir ici la notion de demi-espaces affines pour différentier les ensembles définis par des inégalités de type $a_{in}\mathbf{k}_u \le b_{in}$, avec $a_{in}$ un vecteur non nul et $b_{in}$ un scalaire quelconque, de leur version $a_{in}\mathbf{k}_u \le 0$. Cette observation permet de faire la différence entre un cône polyédral et un polyèdre. Il apparaît clairement qu'un cône polyédral est un polyèdre. La réciproque n'est pas vraie et cet aspect essentiel est à la base de la décomposition des polyèdres résumée par le théorème suivant.

**Théorème (décomposition) 3.10 :** Un ensemble de points dans l'espace euclidien est un polyèdre si et seulement si :

$$P = S + C \text{ [12]} \tag{3.10}$$

avec $C$ un cône polyédral et $S$ un polytope. ■

Cette liaison entre les polyèdres, polytopes et cônes polyédraux est due aux travaux de Motzkin [MRT53]. En résumé, un polyèdre défini par un système de contraintes linéaires auquel on retire le cône polyédral lui correspondant donne un polytope décrit comme on a pu le voir par un ensemble fini de sommets. Si le cône polyédral dans (3.10) est réduit à un point $C = \{0\}$, alors le polyèdre est un polytope, ce que résume aussi le corollaire suivant.

**Corollaire 3.11 :** Un ensemble $P$ est un polytope si et seulement s'il est borné.

La décomposition (3.10) peut être raffinée en utilisant l'observation résumée par (3.6) qui partitionne le cône polyédral en un sous-espace de linéarité et un cône pointu. On peut donc réécrire (3.10) sous la forme :

$$P = S + L + R \tag{3.11}$$

[12] L'opération réalisant la somme de deux ensembles est à envisager ici en sens de Minkowski, c'est-à-dire $X + Y = \{x + y | x \in X, y \in Y\}$.

ou en utilisant la représentation explicite de générateurs pour chaque partie de la décomposition :

$$P = conv.hull\{s_1, \ldots, s_{d_S}\} + lin.space\{l_1, \ldots, l_{d_L}\} + cone\{r_1, \ldots, r_{d_R}\} \tag{3.12}$$

avec *conv.hullS* l'ensemble des combinaisons convexes de points de $S = \{s_1, \ldots, s_{d_S}\}$, *coneR* les combinaisons non négatives des rayons extrêmes et *lin.spaceL* la combinaison linéaire des rayons bidirectionnels.

Pour résumer la discussion sur la dualité des polyèdres, les formulations (3.9), (3.11) et (3.12) peuvent être réunies pour synthétiser l'équivalence des représentations implicites (contraintes) et explicites (générateurs) :

$$P = \left\{\mathbf{k}_u \in \Re^p \middle| A_{eq}\mathbf{k}_u = b_{eq};\, A_{in}\mathbf{k}_u \le b_{in}\right\} \Leftrightarrow \begin{cases} P = \left\{\mathbf{k}_u \in \Re^p \middle| \mathbf{k}_u = \sum_{i=1}^{d_S} \sigma_i s_i + \sum_{i=1}^{d_R} \rho_i r_i + \sum_{i=1}^{d_L} \lambda_i l_i \right\} \\ 0 \le \sigma_i \le 1,\ \sum_{i=1}^{v} \sigma_i = 1,\ \rho_i \ge 0\ ,\forall \lambda_i \end{cases} \tag{3.13}$$

En revenant au problème de commande prédictive qui a motivé tout ce développement géométrique, on peut faire apparaître la représentation duale du domaine faisable défini par un ensemble d'égalités et d'inégalités. Etant donné que les limitations physiques d'un système asservi sont décrites à chaque pas d'échantillonnage par un ensemble de contraintes linéaires mixtes, on dispose dès lors pour le domaine faisable d'une représentation géométrique basée sur les générateurs. Il est alors intéressant d'explorer ce domaine car les générateurs illustrent les performances extrêmes que le système commandé peut procurer.

A ce stade, le problème complet d'optimisation peut être réécrit selon cette représentation duale de générateurs. Il faut noter que les arguments sont désormais les coefficients $\sigma_i, \rho_i, \lambda_i$ qui représentent la façon de combiner les générateurs pour construire la séquence de commande optimale. Le problème d'optimisation devient donc :

$$\begin{aligned} &\min_{\mathbf{k}_u} J_t = 0{,}5\,\mathbf{k}_u^T H \mathbf{k}_u + x_t^T F \mathbf{k}_u \\ &\text{sous les contraintes :} \begin{cases} A_{in}\mathbf{k}_u \le B_{in} x_t + b_{in} \\ A_{eq}\mathbf{k}_u = B_{eq} x_t + b_{eq} \end{cases} \\ \Leftrightarrow\ &\min_{\boldsymbol{\sigma},\boldsymbol{\rho},\boldsymbol{\lambda}} J_t = 0{,}5 \begin{bmatrix} \boldsymbol{\sigma}^T & \boldsymbol{\rho}^T & \boldsymbol{\lambda}^T \end{bmatrix} \begin{bmatrix} S^T \\ R^T \\ L^T \end{bmatrix} H \begin{bmatrix} S & R & L \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma} \\ \boldsymbol{\rho} \\ \boldsymbol{\lambda} \end{bmatrix} + x_t^T F \begin{bmatrix} S & R & L \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma} \\ \boldsymbol{\rho} \\ \boldsymbol{\lambda} \end{bmatrix} \\ &\text{sous les contraintes :} \begin{cases} \boldsymbol{\sigma} \ge 0;\ \boldsymbol{\rho} \ge 0; \\ \mathbf{1}^T \boldsymbol{\sigma} = 1 \end{cases} \end{aligned} \tag{3.14}$$

introduisant la version matricielle des générateurs intervenant dans (3.13) :

$$S = \begin{bmatrix} s_1 \\ \vdots \\ s_{d_S} \end{bmatrix} \in \Re^{d_S \times r};\ R = \begin{bmatrix} r_1 \\ \vdots \\ r_{d_R} \end{bmatrix} \in \Re^{d_R \times r};\ L = \begin{bmatrix} l_1 \\ \vdots \\ l_{d_L} \end{bmatrix} \in \Re^{d_L \times r};\ \boldsymbol{\sigma} = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_{d_S} \end{bmatrix} \in \Re^{d_S};\ \boldsymbol{\rho} = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_{d_R} \end{bmatrix} \in \Re^{d_R};\ \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{d_L} \end{bmatrix} \in \Re^{d_L}$$

L'avantage principal de cette représentation vient du fait que les contraintes sont beaucoup plus simples car constituées exclusivement d'égalités et contraintes de positivité. Ainsi par exemple, toutes les pré-procédures pour la recherche d'un point faisable sont évitées et les méthodes par pivot construisent plus facilement les bases nécessaires lors du processus itératif de recherche de la solution optimale. Malgré tout, cette reformulation du problème d'optimisation peut s'avérer une option moins intéressante lorsque le nombre de générateurs est assez important. Comme les coefficients des générateurs dans (3.13) deviennent les variables d'optimisation de (3.14) une telle approche est inévitablement sensible à la complexité de cette représentation duale. Pour cette raison, une analyse doit être faite hors-ligne lors de la construction de la représentation duale. Théoriquement, la borne supérieure du nombre de générateurs $g(P)$ est assez pessimiste [McM70] :

$$g(P) \le C_{r-d}^{\,r-\left\lfloor \frac{d+1}{2} \right\rfloor} + C_{r-d}^{\,r-\left\lfloor \frac{d+2}{2} \right\rfloor} \text{ [13]} \tag{3.15}$$

[13] Cette limitation est en relation directe avec le nombre de générateurs du polytope le plus complexe, le polytope cyclique [Fuk04].

avec $p$ dimension de l'espace de recherche (dimension de $\mathbf{k}_u$) et $m$ nombre d'inégalités intervenant effectivement dans la constitution du polyèdre $P$. En pratique, la valeur de $p$ dépend du type de système à commander (le nombre d'entrées) et de l'horizon de prédiction. Le premier facteur est caractéristique du système à commander et est imposé, l'horizon de commande quant à lui est choisi, généralement faible (quelques pas d'échantillonnage) car au delà la commande est prolongée par une loi préétablie. Globalement, on aboutit à des valeurs du nombre de générateurs en général raisonnables qui de toute façon interviendront lors d'analyses effectuées hors-ligne.

### 3.1.3 Double description

Les paragraphes précédents ont montré que les polyèdres admettent une double description, mais pratiquement, on ne dispose que d'une seule formulation, ou bien celle des générateurs, ou bien la description implicite des contraintes. La question naturelle qui se pose alors est liée aux méthodes permettant de retrouver la représentation duale d'un polyèdre. Mais avant de continuer plus avant, il faut tout d'abord examiner si ce type de problème est bien posé. Existe-t-il une représentation duale unique ? La réponse est affirmative à la multiplication par une constante positive près, car les contraintes et les rayons demeurent insensibles à cette opération.

Il est intéressant d'observer que même si les rayons et les lignes (les rayons bidirectionnels) admettent une interprétation voisine, pratiquement le rapprochement est à opérer plutôt entre les sommets et les rayons, car les lignes représentent des directions pour lesquelles aucune contrainte n'est intervenue, alors que les rayons peuvent être considérés en fait comme des sommets placés à l'infini. En effet, cette deuxième interprétation simplifie les démarches en passant à une représentation homogène des contraintes. Ceci est fait en utilisant une transformation isomorphe qui assure une correspondance 1 :1 entre les faces de deux polyèdres :

$$\mathbf{k}_u \to \begin{pmatrix} \xi \mathbf{k}_u \\ \xi \end{pmatrix}; \xi > 0 \tag{3.16}$$

Cette transformation correspond à une relation d'équivalence de sorte qu'à chaque polyèdre $P$ de dimension $r$ corresponde un cône $C_P$ de dimension $p+1$ défini par :

$$C_P = \left\{ \begin{pmatrix} \xi \mathbf{k}_u \\ \xi \end{pmatrix} \in \Re^{p+1} \middle| \hat{A}_{in} \begin{pmatrix} \xi \mathbf{k}_u \\ \xi \end{pmatrix} \leq 0, \hat{A}_{eq} \begin{pmatrix} \xi \mathbf{k}_u \\ \xi \end{pmatrix} = 0 \right\} \tag{3.17}$$

avec : $\hat{A}_{eq} = \left[ A_{eq} \mid -b_{eq} \right]$; $\hat{A}_{in} = \left[ \begin{array}{c|c} A_{in} & -b_{in} \\ \hline 0 \cdots 0 & 1 \end{array} \right]$

Le polyèdre original $P$ peut être retrouvé par intersection de $C_P$ avec l'hyperplan défini par :

$$H = \left\{ \begin{bmatrix} \xi \mathbf{k}_u \\ \xi \end{bmatrix} \middle| \xi = 1 \right\}. \tag{3.18}$$

En utilisant la même transformation pour la représentation duale, on obtient une description homogène pour les lignes, rayons et sommets :

$$\hat{R} = \left[ \begin{array}{c|c} R & S \\ \hline 0 \cdots 0 & 1 \cdots 1 \end{array} \right]; \ \hat{L} = \left[ \begin{array}{c} L \\ \hline 0 \cdots 0 \end{array} \right] \tag{3.19}$$

et la représentation homogène duale sera :

$$C_P = \left\{ \begin{pmatrix} \xi \mathbf{k}_u \\ \xi \end{pmatrix} \in \Re^{r+1} \middle\| \begin{pmatrix} \xi \mathbf{k}_u \\ \xi \end{pmatrix} = \hat{R}\hat{\boldsymbol{\rho}} + \hat{L}\hat{\boldsymbol{\lambda}}; \quad \hat{\boldsymbol{\rho}} \geq 0 \right\} \tag{3.20}$$

Avec une représentation homogène, les procédures de passage d'une formulation vers sa duale fonctionnent de la même façon, quels que soient le point de départ et la destination, car les coefficients d'une direction (ligne) dans une représentation correspondent aux coefficients d'une contrainte égalité d'un domaine polyédral. La conséquence est qu'un seul algorithme s'avère suffisant, pouvant cheminer dans les deux sens.

Comme on a pu le constater, il existe des équivalences de structure entre les deux descriptions (égalités/inégalités et sommets/rayons/lignes), en conséquence pour faciliter la compréhension, on peut résumer :

- Les contraintes égalité par leur nature sont saturées par tous les générateurs (sommets/rayons/lignes).
- Toutes les lignes transforment les contraintes inégalité en contraintes égalité.
- Les rayons peuvent saturer une partie des contraintes inégalité.

Formellement, cette classification est transparente si l'on remplace (3.20) dans (3.17) :

$$\forall \hat{\boldsymbol{\rho}} \geq 0; \forall \hat{\boldsymbol{\lambda}} \quad \Rightarrow \quad \hat{A}_{in}(\hat{R}\hat{\boldsymbol{\rho}} + \hat{L}\hat{\boldsymbol{\lambda}}) \leq 0, \hat{A}_{eq}(\hat{R}\hat{\boldsymbol{\rho}} + \hat{L}\hat{\boldsymbol{\lambda}}) = 0 \Rightarrow \begin{cases} \hat{A}_{in}\hat{R} \leq 0, \hat{A}_{in}\hat{L} = 0 \\ \hat{A}_{eq}\hat{R} = 0, \hat{A}_{eq}\hat{L} = 0 \end{cases} \tag{3.21}$$

Cette formulation permet la mise en place d'une matrice dite « d'incidence » dont le nombre de lignes est égal au nombre de contraintes et le nombre de colonnes est égal au nombre de générateurs. Les éléments de cette matrice indiquent si les contraintes sont saturées ou non par les générateurs correspondants (Figure 3.2).

$$\begin{array}{c|cc} & L & R \\ \hline A_{eq} & \{0\} & \{0\} \\ A_{in} & \{0\} & \{0 \; ou \; 1\} \end{array}$$

**Figure 3.2 : Matrice d'incidence (saturation) contenant des éléments non nuls pour les inégalités et les rayons.**

En phase d'implantation, il est important de déterminer la représentation la plus adaptée. Ainsi, certaines opérations sont plus faciles à implémenter dans une représentation que dans l'autre. Par exemple, l'intersection est immédiate avec les contraintes, la construction de la couverture convexe est naturelle avec les générateurs, quant à l'inclusion, un test rapide a besoin des deux représentations. Dès lors, on a choisi de retenir une double représentation qui offre une fois réalisée des avantages lors des phases d'analyse. Ceci nécessite donc la construction de cette double représentation, opération complexe en elle même, connue sous le nom d' « énumération des sommets ». Une classification des algorithmes et les détails algorithmiques sont donnés en Annexe 2.

**Exemple 3.12**

On considère le polyèdre décrit par un ensemble de neuf contraintes (Figure 3.3) :

$$P = \left\{ \mathbf{k}_u \middle| \begin{bmatrix} -1 & -0.50849 \\ -2 & 0 \\ -0.91416 & 1.6 \\ 0.45704 & 1.6 \\ 1.3333 & -0.40724 \\ 1.3333 & 0.7237 \\ 0.59771 & -1.1429 \\ -0.3656 & -1.1429 \\ -2 & 0.51416 \\ -1 & 0 \end{bmatrix} \mathbf{k}_u \geq \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \right\} \underset{\substack{\text{représentation} \\ \text{homogene}}}{\Leftrightarrow} C = \left\{ \begin{bmatrix} \mathbf{k}_u \\ \xi \end{bmatrix} \middle| \begin{bmatrix} -1 & -0.50849 & 1 \\ -2 & 0 & 1 \\ -0.91416 & 1.6 & 1 \\ 0.45704 & 1.6 & 1 \\ 1.3333 & -0.40724 & 1 \\ 1.3333 & 0.7237 & 1 \\ 0.59771 & -1.1429 & 1 \\ -0.3656 & -1.1429 & 1 \\ -2 & 0.51416 & 1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{k}_u \\ \xi \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\} \tag{3.22}$$

Le traitement itératif des contraintes par l'intermédiaire d'un algorithme de type Chernikova pour retrouver la représentation duale peut être suivi Figure 3.4. On observe l'évolution de l'espace 2D sans contrainte au départ vers des domaines décrits par des sommets, des rayons bidirectionnels et rayons unidirectionnels pour arriver à partir de la troisième itération à un domaine sans rayons bidirectionnels et ensuite à la septième itération à des régions bornées décrites seulement par les sommets. Le résultat de l'appel sera donc un ensemble des générateurs défini par 8 sommets :

$$R = \begin{Bmatrix} -0.57452 & 0 & 0.39775 & 0 & -0.75 & -0.48614 & 0.5 & 0.5 \\ 0.57452 & -0.625 & -0.39775 & 0.875 & 0 & -0.48614 & 0.71503 & 0 \end{Bmatrix}$$

Pouvoir faire la différence parmi les directions qui peuvent évoluer à l'infini est un avantage très important pour la définition des degrés de liberté d'une loi de commande, car il ne faut pas oublier que les ensembles polyédraux sont en fait des domaines faisables à l'intérieur desquels les séquences optimales de commande sont à rechercher.

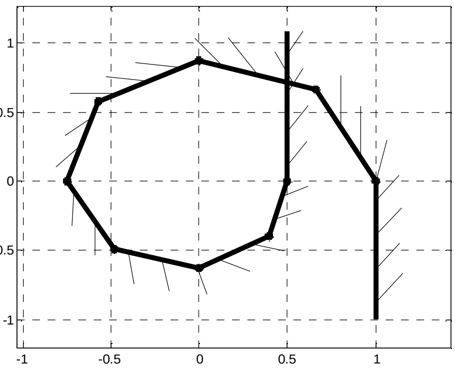



**Figure 3.3 : Hyperplans décrits par l'ensemble des contraintes (3.22).**

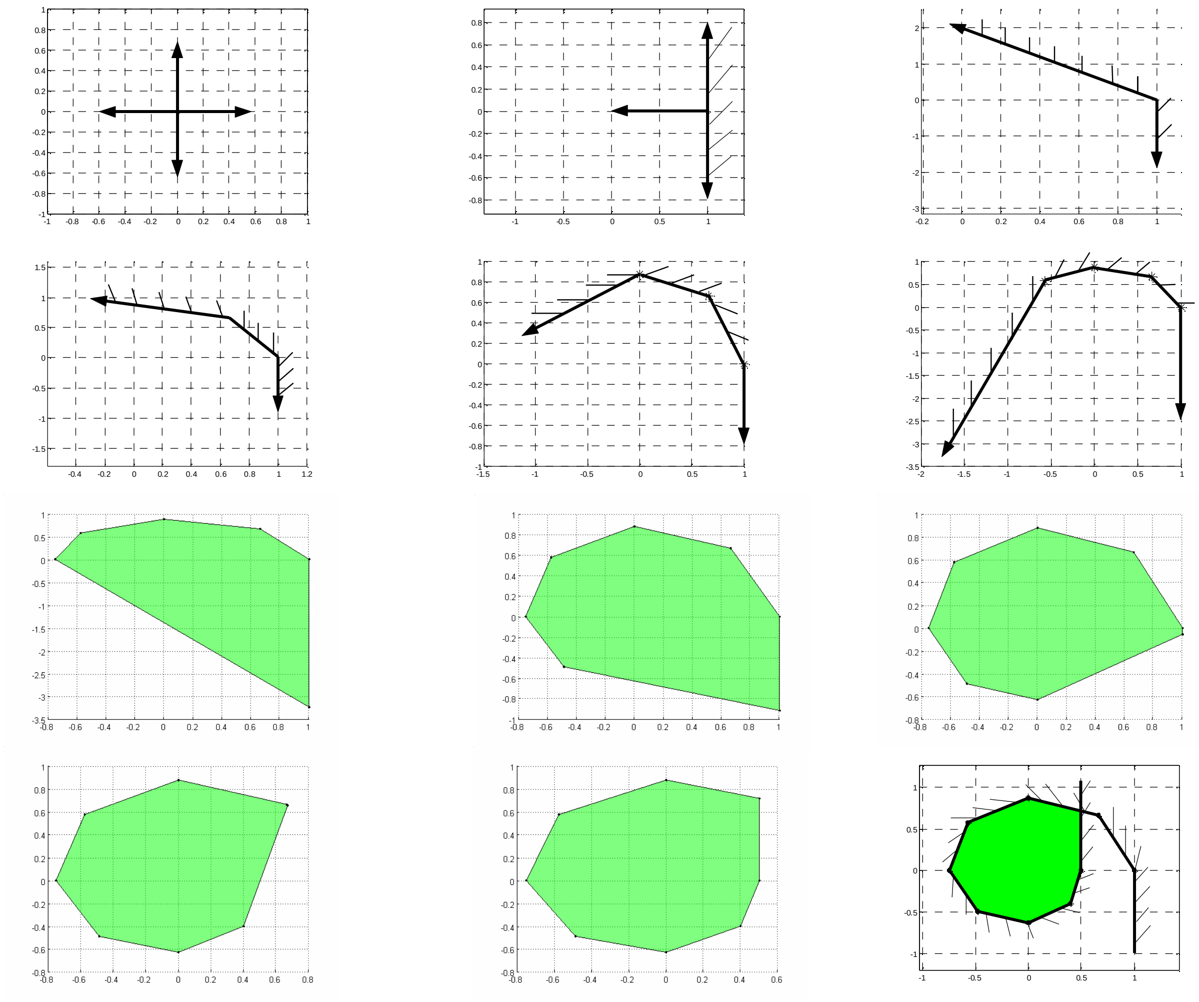


**Figure 3.4 : Evolution de l'espace décrit par le jeu des générateurs pendant les itérations de l'algorithme présenté.**

Mentionnons enfin que l'ordre dans lequel les contraintes sont considérées est très important pour les performances du processus de construction de la forme duale. Malheureusement, il est difficile dans le cas général de donner un ordre de priorité aux contraintes [FQ88].

### 3.1.4 Autres objets et opérations liés aux polyèdres

Un ingrédient important pour l'implémentation d'algorithmes de type Chernikova et en général pour les opérations sur les polyèdres sous forme duale est *la matrice d'incidence (saturation)* (Figure 3.2). Cette matrice décrit les rapports exacts entres les deux représentations. Evidemment, l'information contenue dans cette matrice est redondante car elle réside déjà dans la double représentation, mais comme son évaluation est très coûteuse, et l'effort de stockage négligeable, elle est associée à chaque objet polyédral calculé. Ce choix est déterminé par la nécessité de tester l'adjacence des générateurs et le nombre de contraintes qu'ils doivent saturer.

Même si les algorithmes de passage à la forme duale représentent le cœur des algorithmes de calcul polyédral, d'autres objets sont très utiles pendant les phases d'analyse, particulièrement pour faire la liaison avec la paramétrisation des polyèdres envisagée par la suite. Parmi ces objets figurent les faces.

**Définition 3.13 :** Une face de polyèdre $P \in \Re^p$ est définie par le sous-ensemble non vide :

$$F = \{\mathbf{k}_u \in P | c^T \mathbf{k}_u = d\} \tag{3.23}$$

dont la couverture linéaire a comme dimension $p-1$ et $c^T \mathbf{k}_u \leq d$ représente une inégalité vérifiée $\forall \mathbf{k}_u \in P$.

En fait, ce type d'inégalité fait partie du système initial des inégalités $A_{in} \mathbf{k}_u \leq b_{in}$.

**Remarque 3.14 :** Géométriquement une telle inégalité définit un hyperplan de support au polyèdre $P$ (un hyperplan de dimension $p-1$ ayant une intersection avec $P$ mais non avec son intérieur). Si le polyèdre initial $P$ est de dimension $p$, alors l'intersection d'un hyperplan de support avec $P$ est un polyèdre de dimension $p-1$. Cette condition est nécessaire pour différencier les cas dégénérés des hyperplans de dimension $p-1$ dont l'intersection avec $P$ représente un polyèdre de dimension inférieur à $p-1$ et qui le plus souvent provient des contraintes redondantes.

La définition des faces peut être étendue à des facettes de différentes dimensions. Ainsi, à partir d'un sous-ensemble de $m$ contraintes $C_{in} \mathbf{k}_u \leq d_{in}$ dans $A_{in} \mathbf{k}_u \leq b_{in}$, alors :

$$F^{p-m}(P) = \{\mathbf{k}_u \in P | C_{in} \mathbf{k}_u = d_{in}\} \tag{3.24}$$

définit une facette de dimension $p-m$. Les facettes considérées de façon classiques sont celles de dimension *0 – les sommets*, *1 – les arêtes* et $p-1$ *les faces*.

Récapitulons pour conclure ce paragraphe les principales opérations sur les polyèdres.

- **Intersection** : Pour cette opération, la représentation du domaine par le biais des contraintes est à privilégier. Il suffit de concaténer les systèmes de contraintes pour obtenir une première description du domaine résultant.
- **Couverture convexe** : Pour cette opération, la représentation du domaine par le biais des générateurs s'avère plus adaptée. Il suffit de concaténer les ensembles de générateurs pour obtenir une description primaire de la couverture convexe. La détermination de la forme duale résulte de l'utilisation des algorithmes de passage. La couverture convexe de domaines polyédraux est souvent dénommée union convexe.
- **Union** : Cette opération se distingue des précédentes par le fait que elle n'est pas fermée sur l'espace des polyèdres convexes. Pour permettre leur représentation, le noyau de calcul géométrique considère l'union comme une liste de domaines convexes. Remarquons que le plus petit domaine convexe contenant l'union des deux polyèdres convexes est leur couverture convexe.
- **Différence** : Comme précédemment, la différence de domaines convexes n'est pas un domaine convexe. Pour permettre tout de même l'utilisation de cette opération, le résultat peut être exprimé par le biais d'une union de polyèdres. Cependant, une telle union n'est pas unique car la décomposition de domaines non convexes en domaines convexes n'est pas unique.
- **Projection** : La projection de polyèdres sur des sous-espaces peut être vue comme l'image par une transformation linéaire :

  $$\Pr(P) = \{\mathbf{k}_u | \mathbf{k}_u^{'} = T\mathbf{k}_u + t\} \tag{3.25}$$

  Cette transformation n'est pas 1 :1 car non inversible. L'implémentation effective nécessite la projection des générateurs et le passage à la forme duale pour retrouver la forme avec contraintes.

## 3.2 *Polyèdres paramétrés*

Les paragraphes précédents ont permis d'introduire les notions nécessaires à la description des domaines polyédraux, avec pour conséquence importante l'implémentation de programmes flexibles de calcul polyédral. Dans cette direction, les développements ci-dessous s'intéressent à l'analyse des aspects géométriques des domaines faisables dans le cas de lois prédictives, en examinant en détail les régions décrites par des ensembles de contraintes intervenant dans le problème d'optimisation (3.1). Ces contraintes sont tout à fait similaires á celles intervenant dans la description par contraintes d'un polyèdre (3.13). Cependant, la présence de termes affectant la partie affine des contraintes fait que les domaines polyédraux subissent des modifications par l'impact du déplacement des hyperplans les définissant. Cette modification est classiquement dénommée paramétrisation, et par extension les domaines résultants sont représentés à l'aide du concept de *polyèdre paramétré* [LW97].

### 3.2.1 Sommets paramétrés

Un polyèdre paramétré est défini par un nombre fini d'inégalités et d'égalités linéaires pour lesquelles la partie affine dépend linéairement d'un vecteur de paramètres $x$ :

$$P(x)=\left\{\mathbf{k}_u\in\Re^p \,\middle|\, A_{eq}\mathbf{k}_u=B_{eq}x+b_{eq};\, A_{in}\mathbf{k}_u\le B_{in}x+b_{in}\right\} \tag{3.26}$$

Cette paramétrisation de la partie droite des contraintes influence nécessairement la correspondance avec la représentation duale par générateurs. Malgré tout, les trois catégories de générateurs (rayons, lignes et sommets) ne sont pas affectées de la même façon, à cause de leurs propriétés intrinsèques différentes éclairant les influences de la paramétrisation. Ainsi, il est intéressant d'analyser les différences entre les générateurs de $P(x)$ et ceux de $P(x=0)$. Soient $L=\{l_1,\ldots,l_{d_L}\}$ les rayons bidirectionnels de $P(x=0)$. On peut considérer chacun de ces vecteurs et examiner l'influence de la paramétrisation. En supposant l'existence d'un point $\mathbf{k}_u^0(x)\in P(x)$, on peut vérifier pour un certain jeu $l_i, i=\{1,\cdots,d_L\}$ :

$$\left.\begin{array}{l} \left.\begin{array}{l} A_{eq}\mathbf{k}_u^0(x)=B_{eq}x+b_{eq} \\ A_{in}\mathbf{k}_u^0(x)\le B_{in}x+b_{in} \end{array}\right\} \\ \forall\lambda, \left\{\begin{array}{l} A_{eq}(\mathbf{k}_u^0(0)+\lambda l_i)=b_{eq} \\ A_{in}(\mathbf{k}_u^0(0)+\lambda l_i)\le b_{in} \end{array}\right\} \Rightarrow \left\{\begin{array}{l} A_{eq}l_i=0 \\ A_{in}l_i=0 \end{array}\right\} \end{array}\right\} \Rightarrow \forall\lambda, \left\{\begin{array}{l} A_{eq}(\mathbf{k}_u^0(x)+\lambda l_i)=B_{eq}x+b_{eq} \\ A_{in}(\mathbf{k}_u^0(x)+\lambda l_i)\le B_{in}x+b_{in} \end{array}\right. \tag{3.27}$$

de sorte que les rayons bidirectionnels sont insensibles à la variation des paramètres et caractérisent donc $P(x)$.

Par un raisonnement similaire, il est possible de démontrer que les rayons unidirectionnels $R=\{r_1,\ldots,r_{d_R}\}$ ne changent pas avec l'évolution du vecteur de paramètres, amenant la conclusion suivante.

> **Proposition 3.15 :** Parmi l'ensemble des générateurs, seuls les sommets sont affectés par la paramétrisation des contraintes définissant le domaine polyédral, l'ensemble des rayons étant inchangé.

La double représentation des polyèdres paramétrés est alors définie de la façon suivante :

$$\begin{aligned} P(x)&=\left\{\mathbf{k}_u\in\Re^p \,\middle|\, A_{eq}\mathbf{k}_u=B_{eq}x+b_{eq};\, A_{in}\mathbf{k}_u\le B_{in}x+b_{in}\right\}= \\ &=\left\{\mathbf{k}_u(x)\in\Re^p \,\middle|\, \mathbf{k}_u(x)=\sum_{i=1}^{d_S}\sigma_i(x)s_i(x)+\sum_{i=1}^{d_R}\rho_i r_i+\sum_{i=1}^{d_L}\lambda_i l_i\right\}, 0\le\sigma_i\le 1,\ \sum_{i=1}^{v}\sigma_i=1,\ \rho_i\ge 0\ ,\forall\lambda_i \end{aligned} \tag{3.28}$$

avec $l_i$ les lignes, $r_i$ les rayons, $s_i(x)$ les sommets et $\sigma_i(x),\rho_i,\lambda_i$ les coefficients respectifs. Pour définir une structure géométrique globale, il est possible de réécrire la formulation du polyèdre paramétré (3.28) en un polyèdre non paramétré dans un espace de dimension augmentée :

$$D=\left\{\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\in\Re^{p+n} \,\middle|\, \left[A_{eq}\ \middle|\ -B_{eq}\right]\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}=b_{eq};\, \left[A_{in}\ \middle|\ -B_{in}\right]\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\le b_{in}\right\} \tag{3.29}$$

avec $\mathbf{k}_u\in\Re^p$ et $x\in\Re^n$.

Ce domaine polyédral n'est pas une simple construction artificielle, il représente, du point de vue de la commande prédictive, toutes les combinaisons faisables des paramètres de contexte et des séquences de commande futures. La présentation précédente devient sous forme homogène :

$$C_D = \left\{ \begin{bmatrix} \xi \mathbf{k}_u \\ \xi x \\ \xi \end{bmatrix} \in \Re^{p+n+1} \middle| \hat{A}_{eq} \begin{bmatrix} \xi \mathbf{k}_u \\ \xi x \\ \xi \end{bmatrix} = 0; \hat{A}_{in} \begin{bmatrix} \xi \mathbf{k}_u \\ \xi x \\ \xi \end{bmatrix} \leq 0 \right\} = \left\{ \begin{bmatrix} \xi \mathbf{k}_u \\ \xi x \\ \xi \end{bmatrix} \in \Re^{p+n+1} \middle| \begin{bmatrix} \xi \mathbf{k}_u \\ \xi x \\ \xi \end{bmatrix} = \hat{R}\boldsymbol{\rho} + \hat{L}\boldsymbol{\lambda}; \boldsymbol{\rho} \geq 0 \right\} \quad (3.30)$$

avec $\hat{A}_{eq} = \left[ A_{eq} \mid -B_{eq} \mid -b_{eq} \right]$; $\hat{A}_{in} = \left[ \begin{array}{c|c|c} A_{in} & -B_{in} & -b_{in} \\ \hline \multicolumn{2}{c|}{0 \cdots 0} & 1 \end{array} \right]$ et $\hat{R}, \hat{L}$ des matrices contenant par colonnes les générateurs et $\boldsymbol{\rho}, \boldsymbol{\lambda}$ les vecteurs des coefficients de la combinaison linéaire.

En analysant la description (3.28), les seuls éléments inconnus restent les sommets paramétrés, dont on sait qu'ils sont fonction du vecteur des paramètres. Comme (3.29) est équivalent à (3.28) on retrouve les sommets paramétrés parmi les facettes d'ordre $n$ du polyèdre dans l'espace étendu (données ($\Re^p$)+ paramètre($\Re^n$)). En effet le polyèdre paramétré est en fait issu de l'intersection de son correspondant augmenté avec un hyperplan de dimension $n$ :

$$P(x) = \mathrm{Proj}_{\mathbf{k}_u}\left( D \cap H(x) \right) \quad (3.31)$$

avec :

$$H(x_0) = \left\{ \begin{pmatrix} \mathbf{k}_u \\ x \end{pmatrix} \in \Re^{p+n} \middle| x = x_0 \right\} \quad (3.32)$$

et $\mathrm{Proj}_{\mathbf{k}_u}(.)$ l'opérateur de projection de l'espace $\Re^{p+n}$ sur l'espace original $\Re^p$ représenté par les premières $p$ coordonnées. En conséquence, les sommets paramétrés se retrouvent parmi les faces d'ordre $n$ de $D$ :

$$\mathbf{k}_{u\,i}(x) = \mathrm{Proj}_{\mathbf{k}_u}\left( F_i^n(D) \cap H(x) \right) \quad (3.33)$$

où la facette d'ordre $n$ est l'intersection de $D$ avec $p$ hyperplans de support.

Pour chaque polyèdre dans l'espace étendu $D$, les hyperplans de support représentent en fait les contraintes actives, il en résulte que chaque point $\left( \mathbf{k}_{u\,i}(x)^{\mathrm{T}} \quad x^{\mathrm{T}} \right)^{\mathrm{T}} \in F_i^n(D)$ se retrouve dans un sous-espace défini par :

$$\begin{bmatrix} A_{eq} \\ \overline{A}_{in_i} \end{bmatrix} \mathbf{k}_{u\,i}(x) = \begin{bmatrix} B_{eq} \\ \overline{B}_{in_i} \end{bmatrix} x + \begin{bmatrix} b_{eq} \\ \overline{b}_{in_i} \end{bmatrix} \quad (3.34)$$

avec $\overline{A}_{in_i}, \overline{B}_{in_i}, \overline{b}_{in_i}$ définissant le sous-ensemble des inégalités satisfaites par saturation.

Si la matrice $\left[ A_{eq\,i}^T \quad \overline{A}_{in\,i}^T \right]^T$ n'est pas inversible, la facette $F_i^n(D)$ décrit un ensemble de points pour lequel à un certain $x$ correspond plus d'un point $\mathbf{k}_u \in \Re^p$ faisable, et donc ne décrit pas un sommet de $P(x)$. L'existence de ce sommet est due en fait aux changements de forme de $P(x)$.

Pour la description des dépendances entre les sommets et ses paramètres, seules les combinaisons inversibles doivent être retenues. Elles permettent d'écrire :

$$\mathbf{k}_{u\,i}(x) = \begin{bmatrix} A_{eq} \\ \overline{A}_{in_i} \end{bmatrix}^{-1} \begin{bmatrix} B_{eq} \\ \overline{B}_{in_i} \end{bmatrix} x + \begin{bmatrix} A_{eq} \\ \overline{A}_{in_i} \end{bmatrix}^{-1} \begin{bmatrix} b_{eq} \\ \overline{b}_{in_i} \end{bmatrix} \quad (3.36)$$

Comme expliqué auparavant, du point de vue implémentation, il est préférable de considérer la représentation homogène des polyèdres. Compte tenu des développements précédents, cette représentation s'écrit :

$$\begin{bmatrix} \xi \mathbf{k}_{u\,i}(x) \\ \xi \end{bmatrix} = \left[ \begin{array}{c|c} \begin{bmatrix} A_{eq} \\ \overline{A}_{in_i} \end{bmatrix}^{-1} \begin{bmatrix} B_{eq} \\ \overline{B}_{in_i} \end{bmatrix} & \begin{bmatrix} A_{eq} \\ \overline{A}_{in_i} \end{bmatrix}^{-1} \begin{bmatrix} b_{eq} \\ \overline{b}_{in_i} \end{bmatrix} \\ \hline 0 \quad \cdots \quad 0 & 1 \end{array} \right] \begin{bmatrix} x \\ \xi \end{bmatrix} \quad (3.37)$$

L'implémentation de ces résultats nécessite alors l'énumération de facettes d'ordre *n*. Ces facettes étant à leur tour des polyèdres, ceux-ci peuvent être décrits par un nombre de $k(>n)$ générateurs pour chaque face $F_i^n(P)$. Si les projections suivantes:

$$\mathrm{Pr}_r\begin{pmatrix}\xi\mathbf{k}_{ui}(x)\\ \xi x\\ \xi\end{pmatrix}=\begin{pmatrix}\xi\mathbf{k}_{ui}(x)\\ \xi\end{pmatrix};\mathrm{Pr}_n\begin{pmatrix}\xi\mathbf{k}_{ui}(x)\\ \xi x\\ \xi\end{pmatrix}=\begin{pmatrix}\xi x\\ \xi\end{pmatrix} \tag{3.38}$$

sont définies, alors l'expression des sommets paramétrés (3.37) devient :

$$\begin{bmatrix}\xi\mathbf{k}_{ui}(x)\\ \xi\end{bmatrix}=\mathrm{Pr}_r(F_i)\mathrm{Pr}_n(F_i)^{-1}\begin{pmatrix}\xi x\\ \xi\end{pmatrix}\quad \text{avec}\quad F_\mathrm{i}=\begin{bmatrix}\cdots\begin{pmatrix}\xi\mathbf{k}_{ui}(x_j)\\ \xi x_j\\ \xi\end{pmatrix}\cdots\end{bmatrix}, j=1,\cdots,k \tag{3.39}$$

avec $\left[\xi\ \mathbf{k}_{ui}(x_j)\ \ \xi x_j\ \ \xi\right]^T, j=1,\cdots,k$ les sommets de la face $F_i^n(P)$. Le cas où l'inverse à droite $\mathrm{Pr}_n(F_i)^{-1}$ n'existe pas correspond aux situations déjà mentionnées pour lesquelles des facettes ne définissent pas un sommet paramétré et seront ignorées dans le processus de construction de la représentation duale d'un polyèdre paramétré.

### 3.2.2 Domaines de validité

Le paragraphe précédent a souligné la dépendance linéaire des sommets vis-à-vis du vecteur de paramètres. La première conséquence de cette dépendance est le fait que la position de chaque sommet peut évoluer. Il faut aussi interpréter en ce sens le fait que les rayons unidirectionnels et bidirectionnels ne changent pas avec la paramétrisation. Malgré tout, les sommets paramétrés ne sont pas tous valides en même temps. En se déplaçant, ils peuvent sortir de la région faisable et donc sortir de l'ensemble des générateurs. Cette particularité n'invalide pas la représentation des points à l'intérieur de $P(x)$ comme combinaison de générateurs selon (3.28), car alors les coefficients des sommets invalides, $\sigma_i(x)$, sont nuls. Cependant, pour avoir un image exacte de la forme du polyèdre, il s'avère utile de déterminer le domaine de validité de chaque sommet paramétré $s_i(x)$.

La région du sous-espace des paramètres pour laquelle un sommet donné $s_i(x)$ existe ne nécessite pas d'information supplémentaire par rapport à celle offerte par la facette d'ordre *n* qu'elle identifie dans le polyèdre de l'espace étendu. On retrouve le domaine de validité par projection :

$$V_{\sigma_i}=\mathrm{Pr}_n\left(F_i^n(D)\right) \tag{3.40}$$

Les domaines de validité des sommets paramétrés sont généralement indépendants et donc pour chaque paramètre $x$, il convient de parcourir les domaines de validité pour identifier les sommets donnant les points extrêmes de la région polyédrale.

> **Définition 3.16 :** Soit $VD\subset\Re^n$ une région dans l'espace des paramètres pour un polyèdre paramétré $P(x)$. On dit que $P(x)$ a une forme régulière sur $VD$ si pour $\forall x\in VD$ l'ensemble des sommets paramétrés valides est constant.

Une autre région intéressante de l'espace des paramètres est celle pour laquelle il n'existe aucun sommet valide. Les polyèdres caractérisés par ces paramètres sont vides car les contraintes sont incompatibles, il n'existe donc pas de combinaisons de type (3.34). Cet aspect est très utile dans l'analyse des domaines faisables pour la commande prédictive car il identifie les régions dans l'espace des paramètres de contexte pour lesquelles la loi prédictive n'est pas définie. Il résulte de ces remarques le résultat suivant.

> **Proposition 3.17 :** Soit une loi prédictive induisant la résolution du problème d'optimisation du type (2.51). Pour tout vecteur de paramètres de contexte qui se situe dans :
>
> $$\aleph=\Re^m\setminus\left\{\bigcup V_{\sigma_i}; i=1..n_S\right\} \tag{3.41}$$
>
> on dit que la loi prédictive est infaisable. $n_S$ représente le nombre de sommets paramètres et $V_{\sigma_i}$ les domaines de validité correspondants pour le polyèdre paramétré défini par les contraintes dans (2.51).

### 3.2.3 Exemple

On considère le polyèdre paramétré suivant décrit par huit inégalités linéaires.

$$P(x)=\left\{\mathbf{k}_u=\begin{bmatrix}u_1\\u_2\end{bmatrix}\left|\begin{bmatrix}0&1\\1&-1\\1&-1\\0&1\\0&-1\\-1&1\\-1&1\\0&-1\end{bmatrix}\begin{bmatrix}u_1\\u_2\end{bmatrix}\le\begin{bmatrix}0\\0\\1\\1\\0\\0\\-1\\-1\end{bmatrix}x+\begin{bmatrix}1\\1\\0\\0\\0\\0\\2\\2\end{bmatrix}\right.\right\}\tag{3.42}$$

On peut observer que seules quatre de ces contraintes sont concernées par la paramétrisation de la partie affine. L'évolution en fonction de $x$ de la forme du polyèdre $P(x)$ peut être suivie Figure 3.5. Les contraintes non-paramétrées décrivent la région de la Figure 3.5b, tandis que les contraintes complémentaires déterminent un domaine de même forme mais qui par dépendance linéaire en le paramètre, par intersection avec le domaine fixe, conduit à la forme de $P(x)$. Le fait que les contraintes varient en terme de positionnement de l'hyperplan de frontière influence aussi la position de sommets, qui se situent sur des trajectoires linéaires. Ainsi, pour $0\le x\le 1$, l'origine est le seul sommet fixe et les trois autres évoluent (Figure 3.5a) vers la forme décrite Figure 3.5b. Ceux-ci restent inchangés pour les valeurs des paramètres $1\le x\le 2$. Pour les valeurs de paramètres supérieures à deux, les deux dernières contraintes influencent la forme de $P(x)$ qui a dès lors trois sommets qui évoluent avec les changements de paramètres.

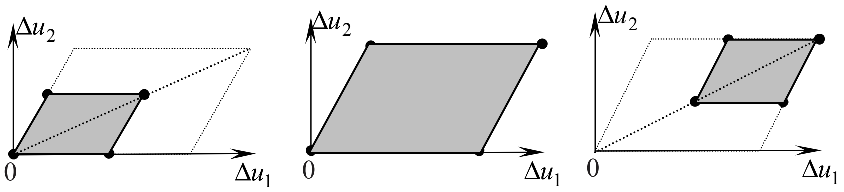



**Figure 3.5 : Formes possibles du polyèdre paramétré *P(x)* décrit par (3.42)**
**a)** $0\le x\le 1$ **b)** $1\le x\le 2$ **c)** $2\le x\le 3$

Pour retrouver la représentation duale de ce polyèdre paramétré, il est nécessaire tout d'abord d'identifier les éventuels rayons uni- et bidirectionnels. Comme ces générateurs ne sont pas influencés par la paramétrisation, fixer la valeur du paramètre peut alors simplifier la démarche. Pour notre exemple, il est évident que l'ensemble de tels générateurs est vide, par conséquent les sommets représentent les seules inconnues. Il faut alors construire la description dans un espace augmenté, puis la double description (3.29). Ensuite, une procédure d'énumération des arêtes doit être mise en œuvre, permettant de mettre en évidence 14 arêtes ou facettes d'ordre 1 (Figure 3.6). Seules 10 de ces facettes possèdent un inverse $\mathrm{Pr}_n(F_i)^{-1}$ et finalement correspondent à des sommets paramétrés avec des domaines de validité spécifiques (Table 3.1).

En analysant les domaines de validité des sommets paramétrés, on peut identifier les domaines de l'espace des paramètres qui correspondent à une forme régulière :

$$VD_1=\{0\le x\le 1\};VD_2=\{1\le x\le 2\};VD_3=\{2\le x\le 3\}\tag{3.43}$$

On peut également identifier les régions correspondant au polyèdre vide, région pour lesquelles aucun sommet paramétré n'existe :

$$N=(-\infty;0)\cup(3,\infty)\tag{3.44}$$

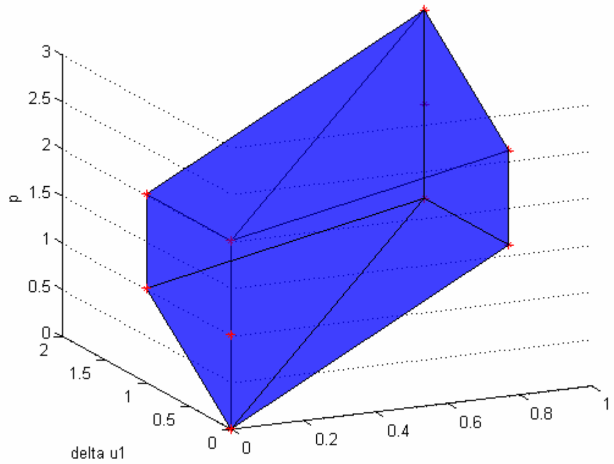



**Figure 3.6 : Polyèdre paramétré dans l'espace de dimension augmentée.**

| **Indice** | **Sommet** | **Domaine de validité** |
|---|---|---|
| **1** | $[2 \quad 1]^T$ | $1 \le x \le 3$ |
| **2** | $[1 \quad 1]^T$ | $1 \le x \le 2$ |
| **3** | $[x-1 \quad 1]^T$ | $2 \le x \le 3$ |
| **4** | $[1 \quad 0]^T$ | $1 \le x \le 2$ |
| **5** | $[x-1 \quad x-2]^T$ | $2 \le x \le 3$ |
| **6** | $[2x \quad x]^T$ | $0 \le x \le 1$ |
| **7** | $[x \quad 0]^T$ | $0 \le x \le 1$ |
| **8** | $[x \quad x]^T$ | $0 \le x \le 1$ |
| **9** | $[0 \quad 0]^T$ | $0 \le x \le 2$ |
| **10** | $[2x-4 \quad x-2]^T$ | $2 \le x \le 3$ |

**Table 3.1 : Sommets paramétrés et leurs domaines de validité.**

## *3.3 Détails d'implémentation*

Dans les paragraphes suivants, tous les résultats basés sur des développements géométriques appliqués à la commande prédictive mettent en œuvre une double représentation. Il peut apparaître certes parfois un peu lourd d'avoir à manipuler ce type de formalisme, la littérature se faisant souvent le reflet de cette critique, particulièrement à cause de la complexité des algorithmes et surtout de la taille de la sortie (la représentation duale issue de l'algorithme de Chernikova). Les développements à venir compenseront cette complexité en exploitant le gain d'information qu'offre cette double représentation.

S'il s'avère malgré tout que la complexité du problème de commande prédictive fasse que l'indicateur $O(r^{(d/2)})$, avec $r$ le nombre de contraintes et $d$ la dimension, dépasse la charge de calcul autorisée hors-ligne pour la construction de l'ensemble des générateurs, des méthodes alternatives reposant exclusivement sur la représentation des domaines polyédraux en termes de contraintes (de type Fourrier-Motzkin ou par algorithmes de projection) doivent être envisagées.

### 3.3.1 Spécificités du noyau de calcul polyédral développé sous Matlab™

Les travaux liés à ce mémoire ont permis le développement d'un noyau de calcul polyédral, basé comme il a été mentionné sur une double représentation des polyèdres, regroupé dans une bibliothèque appelée 'PPP' (Predictive control by Parameterized Polyehedra). Il représente à ce jour (à ma connaissance) le seul script Matlab™ qui se base sur l'algorithme de Chernikova pour le passage d'une représentation vers sa duale. Les avantages sont liés à la transparence du code et la portabilité, car sous sa forme classique, il ne nécessite pas d'outils particuliers, mis à part les programmes de base de calcul numérique.

Mentionnons également au titre des spécificités le fait que les égalités et inégalités sont considérées comme des entités séparées, avec comme avantages une représentation réduite en terme de place mémoire, car les égalités et les lignes ne doivent pas être dupliquées, la possibilité d'utiliser des solveurs performants pour les systèmes d'équations … Les objets polyédraux incluent des informations supplémentaires par rapport à la double représentation, comme la dimension, la dimension de l'espace de linéarité, le nombre de contraintes et leur répartition en égalités et inégalités. Un autre aspect original consiste en l'utilisation de la matrice d'incidence lors de la procédure d'énumération des facettes.

Spécifiquement, pour la phase d'énumération des facettes d'ordre *n* au sein d'un polyèdre de dimension $n+p$ (incluant la procédure de vérification de viabilité et d'élimination de redondance), la complexité est bornée dans le pire cas par $O((r\gamma+\gamma)C_r^{(p+n)-n})$, où $r$ est le nombre de contraintes, $\gamma$ le nombre des générateurs. Si *f* est le nombre effectif de facettes d'ordre *n*, alors on peut écrire :

$$O((r\gamma+\gamma)f) \leq O(\text{énumération}) \leq O((r\gamma+\gamma)C_r^p) \tag{3.45}$$

Enfin, il faut insister sur la possibilité de travailler avec des polyèdres et pas uniquement des polytopes, ainsi que l'implémentation de programmes spécifiques pour la manipulation de polyèdres paramétrés avec sommets paramétrés et domaines de validité correspondants.

### 3.3.2 Positionnement par rapport aux autres bibliothèques

Les résultats présentés ici ont été obtenus pour le cas d'espaces réels, mais leur validité reste inchangée si l'on se restreint aux espaces rationnels (dans ce cas, matrices, vecteurs et scalaires ne prennent que des valeurs rationnelles). Dans la pratique, les boites à outils logicielles existantes se distinguent précisément par la représentation qu'elles utilisent. Celles optant pour la représentation rationnelle ont l'avantage de discerner la saturation des contraintes sans erreur, en revanche le prix payé est une complexité élevée de représentation avec des coefficients pouvant excéder la représentation habituelle des entiers (phénomène d'*overflow*), même pour des petites tailles, et donc nécessiter des mécanismes spécifiques – voire l'utilisation de bibliothèques spécialisées multiprécision (GNU MP). En revanche, travailler avec une représentation réelle surmonte les problèmes de dépassement (overflow) mais nécessite de maîtriser les erreurs de représentation en virgule flottante. Discerner entre les contraintes saturées, vérifiées et non vérifiées constituent les problèmes les plus délicats à résoudre. Cette opération peut être affectée par les erreurs de représentation, qui rendent la vérification de la saturation indiscernable. Idéalement, il faudrait concevoir des algorithmes fournissant la double représentation des polyèdres sans augmenter le mauvais conditionnement d'une description initiale.

Un exemple de bibliothèque de calcul polyédral travaillant avec une représentation rationnelle est 'Polylib' développée à l'IRISA [WIL94]. Cet outil performant possède de nombreuses qualités, un de ses avantages étant l'utilisation explicite de programmes pour polyèdres paramétrés. Pour les applications développées dans ce mémoire, certains algorithmes de Polylib présentant un intérêt dans le cas des problèmes de commande prédictive ont ainsi été adaptés sous Matlab. Ceci implique une représentation par nombres réels en virgule flottante tout en essayant de limiter les erreurs dues à cette représentation dans un domaine acceptable.

D'autres algorithmes existent pour traiter les opérations de calcul polyédral, citons par exemple CDD [Fuk05], Convexhull, Porta, GBT [GBT05] …

## *3.4 Redondance dans les problèmes d'optimisation multiparamétrique*

Les notions d'analyse des domaines polyédraux présentées précédemment peuvent être employées dans le cadre de la représentation des domaines faisables des problèmes d'optimisation multiparamétrique. Partant de cet acquis, ce paragraphe examine les influences de la redondance qui peut caractériser un ensemble de contraintes linéaires. L'intérêt pour l'analyse de la redondance au sein d'un ensemble de contraintes linéaires n'est pas nouveau. On trouve dans la littérature plusieurs travaux à ce sujet, mentionnons tout particulièrement l'ouvrage de [KLTZ83].

> **Définition 3.18 :** Une contrainte dans un ensemble est dite redondante si elle peut être déduite des autres contraintes de l'ensemble. Un ensemble est dit non redondant s'il ne contient pas de contraintes redondantes.

**Remarque 3.19 :** Les contraintes redondantes alourdissent la description des domaines polyédraux. Il est donc préférable de les éliminer pour ne considérer qu'un ensemble de contraintes minimal. Cette procédure doit prendre en considération le fait que l'élimination d'une contrainte redondante peut transformer certaines autres contraintes redondantes en contraintes non redondantes. En conséquence, les contraintes redondantes ne peuvent pas être généralement toutes éliminées simultanément, il est préférable d'opter pour une démarche itérative.

Pour des contraintes affectées par la paramétrisation, comme celles intervenant dans les problèmes d'optimisation multiparamétriques, on distingue deux types de redondance :

- redondance globale, qui caractérise les contraintes pour toutes les valeurs possibles des paramètres. Ce type de redondance est éliminable sans aucune perte d'information pour la solution optimale.
- redondance locale, liée à une valeur particulière des paramètres, pour lesquels seul un sous-ensemble de contraintes est nécessaire dans la description du domaine faisable et donc de la solution optimale. La particularité est qu'avec la variation des paramètres les contraintes redondantes peuvent devenir non redondantes et vice-versa, de sorte que l'élimination des contraintes redondantes peut affecter localement la solution globale.

### 3.4.1 Redondance globale des contraintes

Pour un polyèdre paramétré :

$$P(x) = \left\{\mathbf{k}_u \;\middle|\; A_{in}\mathbf{k}_u \le B_{in}x + b_{in} \;,\; A_{eq}\mathbf{k}_u = B_{eq}x + b_{eq}\right\} \tag{3.46}$$

le problème de redondance globale des contraintes est équivalent à considérer la redondance des contraintes pour le polyèdre obtenu par changement de variable :

$$A_{in} \leftarrow \begin{bmatrix} A_{in} \\ -B_{in} \end{bmatrix}; A_{eq} \leftarrow \begin{bmatrix} A_{eq} \\ -B_{eq} \end{bmatrix} \tag{3.47}$$

On envisage donc par la suite le cas classique de polyèdre décrit par un ensemble de contraintes mixtes :

$$P = \left\{\mathbf{k}_u \;\middle|\; A_{in}\mathbf{k}_u \le b_{in} \;,\; A_{eq}\mathbf{k}_u = b_{eq}\right\} \tag{3.48}$$

avec $A_{in} \in \Re^{r\times p}; b_{in} \in \Re^{r}; A_{eq} \in \Re^{s\times p}; b_{eq} \in \Re^{s}$ et ne contenant pas d'égalités implicites parmi les inégalités.

L'existence de contraintes égalité redondantes est transparente dans le rang de la matrice $A_{eq}$ et peut être facilement écartée. Il est intéressant d'observer que, pendant la procédure d'élimination des égalités redondantes, les situations d'infaisabilité résultant d'incompatibilités ($\mathrm{rang}(A_{eq}) \ne \mathrm{rang}([A_{eq} \;\; b_{eq}])$) peuvent également être mises en évidence, selon l'algorithme suivant :

| | |
|---|---|
| 1) | Calcul $\rho_1 = rang(A_{eq})$ |
| 2) | Si $\rho_1 = s$, alors terminer avec le message « l'ensemble de contraintes est non redondant », sinon calcul de $\rho_2 = rang([A_{eq} \;\; b_{eq}])$ |
| 3) | Si $\rho_2 > \rho_1$, alors terminer avec le message d'infaisabilité « ensemble de contraintes incompatibles », sinon retrouver un ensemble d'égalités indépendantes (non redondantes) équivalent à l'ensemble initial par une factorisation (SVD) par exemple. |

L'élimination des inégalités redondantes est cependant plus laborieuse. Pour résumer, il faut vérifier que la i[ème] contrainte inégalité, $i \in \Im = \{1, \cdots, r\}$, est redondante si elle peut être extraite de la description (3.46) sans modifier le domaine $P(x)$, ce qui est équivalent à dire :

$$A_{in\,j}\mathbf{k}_u \le b_{in\,j}, \forall j \in \Im - \{i\}, A_{eq}\mathbf{k}_u = b_{eq} \Rightarrow A_{in\,i}\mathbf{k}_u \le b_{in\,i} \tag{3.49}$$

Une idée très simple mais coûteuse en temps de calcul (utilisée notamment en [GT91], [Ton03]) consiste à résoudre un problème d'optimisation pour chaque contrainte en partant de l'idée que les effets de son élimination se feront sentir sur la valeur d'un critère linéaire judicieusement choisi. L'algorithme est le suivant.

***Algorithme Elimination Red 1***

1) $\aleph = \{1, \cdots, r\}$ et $i = 1$
2) Résoudre :

$$m = \max_{z} A_{in\,i}$$
$$\text{sous les contraintes } \begin{cases} A_{in\,\Im-\{i\}} z \leq b_{in\,\Im-\{i\}} \\ A_{eq} z = b_{eq} \end{cases}$$

3) Si $m \leq b_{in\,i}$ alors $\aleph \leftarrow \aleph - \{i\}$
4) Si $i < r$ alors $i \leftarrow i+1$ et retour à l'étape 2, sinon fin.

Le résultat est l'ensemble $\aleph$ des indices correspondant aux inégalités non redondantes. Pour améliorer la robustesse numérique de l'étape 2), un problème perturbé peut être considéré :

$$m = \max_{z} A_{in\,i}$$
$$\text{sous les contraintes } \begin{cases} A_{in\,\Im-\{i\}} z \leq (1+\varepsilon) b_{in\,\Im-\{i\}} \\ A_{eq} z = b_{eq} \end{cases} \tag{3.50}$$

avec $\varepsilon$ une quantité non négative.

**Remarque 3.20 :** Les arrangements de contraintes peuvent influencer l'ensemble résultant $\aleph$ mais aussi le temps de calcul. Par exemple, si les contraintes redondantes sont parmi les premières traitées, les problèmes d'optimisation à résoudre diminuent en complexité.

Il existe aussi des techniques heuristiques d'élimination des contraintes de type inégalité redondantes. Si l'on possède un ensemble de contraintes $A_{in}\mathbf{k}_u \leq b_{in}$ avec un degré de redondance significatif, l'algorithme suivant peut s'avérer utile.

***Algorithme Elimination Red 2***

1) Sélectionner un sous-ensemble de contraintes $A_{in}^{'}\mathbf{k}_u \leq b_{in}^{'}$ parmi $A_{in}\mathbf{k}_u \leq b_{in}$
2) Soit $\aleph$ l'ensemble des indices des contraintes $A_{in}\mathbf{k}_u \leq b_{in}$ qui ne sont pas parmi $A_{in}^{'}\mathbf{k}_u \leq b_{in}^{'}$
3) Construire la forme duale de $P = \{\mathbf{k}_u | A_{in}^{'}\mathbf{k}_u \leq b_{in}^{'}\}$. Soit $S, R, L$ les ensembles des sommets, rayons uni- et bidirectionnels respectivement.
4) Pour chaque $i \in \aleph$ :

   Si $\forall s \in S, A_{in\,i} s \leq b_{in\,i}$, $\forall r \in R, A_{in\,i} r \leq 0$ et $\forall l \in L, A_{in\,i} l = 0$ alors $\aleph \leftarrow \aleph - \{i\}$

Le caractère heuristique de l'algorithme réside dans la façon de choisir le sous-ensemble de contraintes pour lequel la représentation duale est construite. Plus ce sous-ensemble est important, meilleure sera l'élimination de la redondance. Comme il n'existe aucun élément garantissant que le choix effectué permettra de détecter les redondances, le succès de la procédure est relatif. De façon pratique, pour les contraintes liées à la commande prédictive, le sous-ensemble des contraintes peut être choisi en sélectionnant les contraintes sur l'amplitude de commande ou sur les incréments de commande car leur représentation duale est immédiate.

### 3.4.2 Elimination de la redondance globale basée sur la description duale

Lors de la description de l'algorithme de passage à la description par générateurs d'un polyèdre (voir sous chapitre 3.1.3), on a supposé que l'on disposait d'un ensemble initial de contraintes. Aucune hypothèse n'a été faite quant à la présence de redondance dans cet ensemble initial. En effet, l'algorithme peut fonctionner en présence de redondances, même si une procédure d'élimination de redondance basée sur des optimisations successives comme celles présentées ci-dessus peut être mise en place, avant d'entamer la construction de la représentation duale, afin d'améliorer les performances.

> **Remarque 3.21 :** L'algorithme de Chernikova est réputé sensible à l'ordre des contraintes, ce qui peut influencer le temps de calcul de la représentation duale. En fait, cette sensibilité est liée tout particulièrement à l'existence des contraintes redondantes. Si celles-ci sont parmi les premières traitées lors du processus de passage, des modifications superflues sont introduites dans l'ensemble des générateurs avec des conséquences sur le temps de calcul.

Si l'existence de la redondance dans l'ensemble de contraintes est ignorée lors de l'application de l'algorithme de passage vers la forme duale, il est intéressant de savoir, une fois cette forme mise en œuvre, si une procédure d'élimination de redondance peut utiliser la nouvelle double représentation. D'où la proposition suivante.

> **Proposition 3.22 :** Une contrainte est non redondante dans un ensemble définissant un polyèdre si elle est saturée par au moins $p$ générateurs ($p$ dimension de l'espace qui contient le polyèdre) et si elle est unique (ne peut pas être retrouvée par une multiplication par une autre contrainte).

L'applicabilité des conditions exprimées dans cette proposition est fortement liée à la redondance des générateurs. Il est facile d'observer que, dans le cas d'existence de sommets redondants, la saturation d'une contrainte par $p$ générateurs peut être acquise sans certifier qu'elle représente un hyperplan de support pour le polyèdre (voir la remarque 3.14) et donc la détection de redondance devient plus difficile.

***<u>Algorithme Elimination_Red_3</u>***

Données d'entrée : matrice d'incidence pour un polyèdre avec un ensemble de générateurs non redondants
Données de sortie : indices des inégalités non redondantes

1) $\aleph = \{1, \cdots, r\}$ et $i = 1$
2) Compter le nombre d'éléments 0 de la ligne $i$ de la matrice d'incidence
3) Si ce nombre est plus petit que $p$ alors $\aleph \leftarrow \aleph - \{i\}$, $i = i + 1$ et si $i < r$ retour à 1)
4) S'il existe un $j \in \aleph, j < i$ tel que le « ou exclusif » entre la ligne $i$ et la ligne $j$ de la matrice d'incidence produit un vecteur nul alors $\aleph \leftarrow \aleph - \{i\}$,
5) $i = i + 1$ et si $i < r$ retour à 1)

La complexité de cet algorithme est réduite car l'information nécessaire est incluse uniquement dans la matrice d'incidence. Même si le processus de détection est itératif, le nombre d'opérations effectives est minimal.

***<u>Algorithme Elimination_Red_4</u>***

Données d'entrée : matrice d'incidence $INC$ pour un polyèdre avec un ensemble de générateurs redondants
Données de sortie : indices des inégalités non redondantes

1) $\aleph = \{1, \cdots, r\}$ et $i = 1$
2) Compter le nombre d'éléments 0 de la ligne $i$
3) Si ce nombre est plus petit que $p$ alors $\aleph \leftarrow \aleph - \{i\}$, $i = i + 1$ et si $i < r$ retour à 1)
4) S'il existe un $j \in \aleph, j < i$ tel que $(INC(i,:) + INC(j,:)) \oplus INC(j,:)$ ne contient aucun 0 alors $\aleph \leftarrow \aleph - \{i\}$.
   Si $(INC(i,:) + INC(j,:)) \oplus INC(i,:)$ ne contient aucun 0 alors $\aleph \leftarrow \aleph - \{j\}$.
5) $i = i + 1$ et si $i < r$ retour à 1)

L'opération décrite à l'étape 4) de l'algorithme a pour but de vérifier si la contrainte courante est saturée par un ensemble de générateurs non inclus parmi les générateurs saturés par une autre contrainte. L'inclusion est valable dans les deux sens de sorte qu'une contrainte déjà analysée puisse être signalée redondante lors de l'analyse d'une autre contrainte traitée ultérieurement, ce qui était impossible dans la version précédente.

### 3.4.3 Contraintes redondantes fonction des paramètres

Les algorithmes présentés aux deux paragraphes précédents ont traité le problème de la redondance globale de contraintes. Il a ainsi été montré que ce type de redondance conduit à la vérification de la redondance d'ensembles de contraintes non paramétrés. Or, même si l'on possède un ensemble de contraintes minimal selon ce critère, pour des valeurs particulières du vecteur de paramètres $x = x_0$, les contraintes utiles pour la définition de la région :

$$P(x_0) = \left\{ \mathbf{k}_u \middle| A_{in}\mathbf{k}_u \leq B_{in}x_0 + b_{in}, A_{eq}\mathbf{k}_u = B_{eq}x_0 + b_{eq} \right\} \tag{3.51}$$

induisent une redondance différente, dépendante de la valeur choisie $x = x_0$ et qui perdure sur un voisinage du point $x_0 \in \Re^n$, d'où la notion de redondance locale. Cet aspect est à relier au concept de forme régulière pour le polyèdre paramétré $P(x)$.

> **Définition 3.23 :** Le polyèdre paramétré $P(x)$ a une *forme régulière* sur $D \subset \Re^n$ si l'ensemble des contraintes non redondantes qui le définit reste constant pour tout $x \in D$.

Pour être plus précis, pour tout paramètre $x \in D$, l'ensemble des contraintes non redondantes contient toutes les égalités $A_{eq}, B_{eq}, b_{eq}$ et un sous-ensemble des inégalités $\widetilde{A}_{in}, \widetilde{B}_{in}, \widetilde{b}_{in}$ d'indices $\widetilde{\Im} \subseteq \Im = \{1,\ldots,r\}$. L'exemple suivant met en évidence les variations induites par l'entrée ou la sortie d'une contrainte d'un ensemble de contraintes redondantes.

**Exemple 3.24 :** On considère le polyèdre paramétré (Figure 3.7) :

$$P(x) = \left\{ \mathbf{k}_u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \middle| \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ -3 & -2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \\ -3 \\ -2 \\ -6 \end{bmatrix} \right\} \tag{3.52}$$

avec les indices des contraintes $\Im = \{1,2,3,4,5\}$.

On peut différentier cinq intervalles différents dans lesquels l'évolution du paramètre $x$ ne modifie pas la disposition des ensembles de contraintes redondantes $\Re$ respectivement non redondantes $\aleph\Re$ :

$$\begin{aligned} D_1 &= (-\infty,-1) \quad \rightarrow \quad \Re = \{1,2,3,4,5\}, \aleph\Re = \varnothing, \\ D_2 &= [-1,1] \quad \rightarrow \quad \Re = \{3,4\}, \aleph\Re = \{1,2,5\}, \\ D_3 &= [1,5] \quad \rightarrow \quad \Re = \{3\}, \aleph\Re = \{1,2,4,5\}, \\ D_4 &= [5,7] \quad \rightarrow \quad \Re = \varnothing, \aleph\Re = \{1,2,3,4,5\}, \\ D_3 &= [7,\infty] \quad \rightarrow \quad \Re = \{5\}, \aleph\Re = \{1,2,3,4\} \end{aligned} \tag{3.53}$$

Pour chacun de ces intervalles $D_1, \cdots D_5$, conformément à la définition 3.23, le polyèdre paramètre $P(x)$ a une *forme régulière.* On peut observer que pour $x \in D_1 = (-\infty,-1)$, l'ensemble des contraintes non redondantes est vide (toutes les contraintes sont incompatibles), ce qui conduirait à une infaisabilité lors d'une optimisation multiparamétrique sur $P(x)$.

La première idée venant à l'esprit consiste alors à partitionner l'espace des paramètres en zones correspondant à des formes régulières de $P(x)$. Malheureusement, la recherche de ce type de partition peut s'avérer difficile, en se basant exclusivement sur la description de $P(x)$, en terme d'intersection de demi-espaces. Une autre idée serait d'énumérer les combinaisons possibles de contraintes et d'identifier les valeurs des paramètres qui garantissent leur non redondance. La complexité d'une telle opération est cependant prohibitive. Si l'on reprend alors la description duale d'un polyèdre paramétré introduite lors des paragraphes précédents, il est intéressant de rechercher le correspondant dans ce cadre de la définition d'un domaine de forme régulière. Une telle définition est en liaison directe avec la définition 3.16 sur les domaines de validité des sommets paramétrés. En effet, un correspondant de la définition 3.23 en tenant compte de la représentation des générateurs est donné ci-dessous.

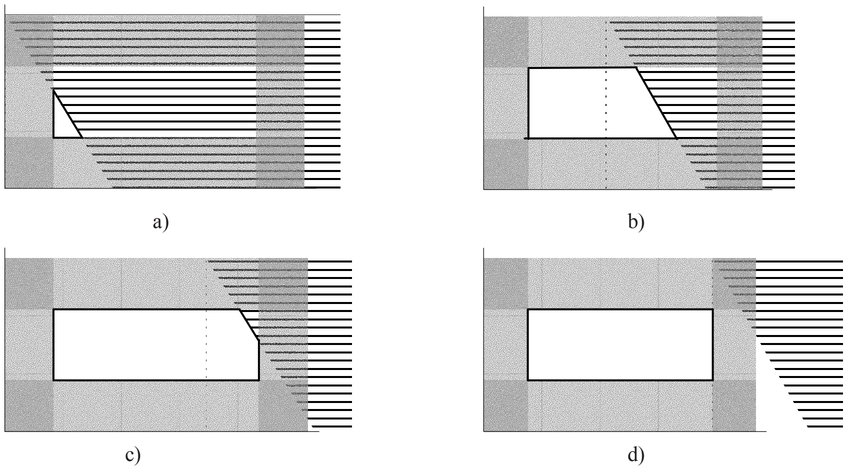



**Figure 3.7 : Formes régulières pour l'exemple 3.24.**
**De a) à d) domaines correspondant aux valeurs de $x$ dans $D_2,\ldots D_5$ .**

**Définition 3.25 :** Le polyèdre paramétré $P(x)$ a une *forme régulière* sur $D \subset \Re^n$ si le nombre de sommets reste constant et si chacun de ces sommets peut s'exprimer comme une fonction linéaire des paramètres $x \in D$ (l'ensemble des sommets paramétrés reste constant).

On retrouve donc dans cette définition les sommets paramétrés d'un domaine polyédral (3.51). On a pu voir que l'utilisation des algorithmes de passage à la représentation duale et l'énumération des facettes d'ordre *n* permet de retrouver la formulation de ces sommets dépendant des paramètres de contexte. Mais on s'intéresse ici à la caractérisation des domaines correspondant à une forme régulière, et donc tout particulièrement aux domaines de validité des sommets. Il est en effet clair que les sommets paramétrés peuvent se regrouper ou se scinder introduisant ainsi de nouveaux sommets. De plus, l'appartenance d'un sommet paramétré à une seule région de forme régulière n'est pas garantie car les autres sommets peuvent ne pas partager le même domaine de validité. Pour retrouver les partitions de l'espace des paramètres correspondant à ces zones, il faut calculer pour chaque domaine de validité, $V_{\sigma i}, i = 1,..,n_S$ ( $n_S$ nombre de sommets), l'intersection :

$$DC = \bigcap_{\substack{j=1,..,n_S \\ V_{\sigma j} \cap V_{\sigma i} \neq 0}} V_{\sigma j} \tag{3.54}$$

qui sera retenu comme un domaine correspondant à une forme régulière et la procédure sera réitérée pour tous les domaines non vides $V_{\sigma l} = V_{\sigma l} - DC, l = 1..n_S$ . Le résultat de cette partition est le découpage de la Figure 3.8.

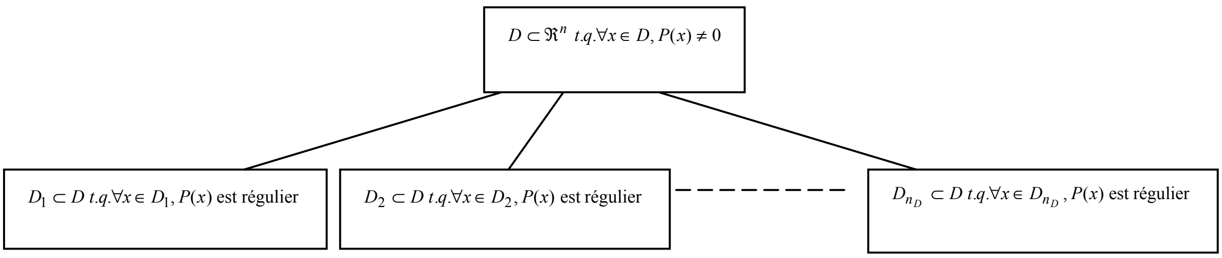



**Figure 3.8 : Principe de construction d'un découpage.**

**Remarque 3.26 :** La décomposition de l'espace des paramètres étant effectuée par soustractions successives conduit à un résultat sous forme d'une union de domaines convexes (la différence de deux domaines convexes n'est pas un domaine convexe mais peut être représentée par une union de convexes).

Pour résumer les idées présentées sur la décomposition de l'espace des paramètres d'un problème d'optimisation multiparamétrique en régions correspondant à des ensembles non redondants de contraintes, voici ci-dessous les principes d'un algorithme élaborant les régions traduisant les formes régulières d'un polyèdre paramétré associé. Le schéma de principe est illustré Figure 3.9.

***Algorithme ScanDom***

*Données d'entrée :* Ensemble de contraintes linéaires égalité et inégalité pour un problème d'optimisation multiparamétrique : $A_{in}\mathbf{k}_u \leq B_{in}x_0 + b_{in}$, $A_{eq}\mathbf{k}_u = B_{eq}x_0 + b_{eq}$

*Données de sortie :* Union des ensembles convexes caractérisant un découpage de l'espace des paramètres du problème d'optimisation. Les contraintes non redondantes correspondent aux régions de l'union.

1. calculer la forme duale pour $D = \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \middle| \begin{bmatrix} A_{in} & -B_{in} \end{bmatrix} \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \leq b_{in}, \begin{bmatrix} A_{eq} & -B_{eq} \end{bmatrix} \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} = b_{eq} \right\}$
2. Enumérer les facettes d'ordre $n$
3. Retenir celles ayant une projection non dégénérée sur l'espace des paramètres
4. Pour chaque facette retenue, construire le sommet paramétré correspondant $s_i(x)$ et son domaine de validité $V_{\sigma_i}$.
5. Construire une liste ($VD$) et initialiser les nœuds avec les domaines de validité $V_{\sigma_i}$
6. $i = 1$
7. Pour $i$ = 1:cardinal ($VD$)

i. Tant que $VD_i \neq 0$

$$D_k = \bigcap_{\substack{j=1..n \\ VD_j \cap VD_i \neq 0}} VD_j; \quad VD_l = VD_l - D_k, \quad l = i \cdots n$$

Stocker $D_k$

$k = k + 1$

fin

fin

8. Pour chaque $D_k$, stocker l'ensembles des contraintes et des générateurs non redondants.

$$P(x) = \left\{ \mathbf{k}_u \middle| A_{in}\mathbf{k}_u \leq B_{in}x + b_{in}, A_{eq}\mathbf{k}_u = B_{eq}x + b_{eq} \right\}$$

$$\Updownarrow$$

$$Si \quad x \in D_1 \qquad \cdots \qquad Si \quad x \in D_{n_D}$$

$$P(x) = \left\{ \mathbf{k}_u \middle| A_{in1}\mathbf{k}_u \leq B_{in1}x + b_{in}, A_{eq1}\mathbf{k}_u = B_{eq1}x + b_{eq1} \right\} \qquad \cdots \qquad P(x) = \left\{ \mathbf{k}_u \middle| A_{inn_D}\mathbf{k}_u \leq B_{inn_D}x + b_{inn_D}, A_{eq_{n_D}}\mathbf{k}_u = B_{eq_{n_D}}x + b_{eq_{n_D}} \right.$$

**Figure 3.9 : Principe de construction d'un découpage avec non redondance par morceaux. Les ensembles de contraintes de la partie inférieure sont des sous-ensembles de ceux de la formulation initiale.**

> **Remarque 3.27 :** L'argument de sortie de l'algorithme précédent est une union d'ensembles convexes dans l'espace des paramètres du problème d'optimisation initial. Il est possible que cette union ne couvre pas entièrement cet espace. Les zones non couvertes représentent en fait des combinaisons de paramètres infaisables.

### 3.4.4 Problèmes d'optimisation paramétrique sans redondance

Le polyèdre :

$$P(x) = \begin{cases} A_{in}\mathbf{k}_u \leq B_{in}x + b_{in} \\ A_{eq}\mathbf{k}_u = B_{eq}x + b_{eq} \end{cases} \tag{3.55}$$

peut être exprimé sous une forme équivalente par une représentation de contraintes non redondantes par morceaux :

$$P(x)=\begin{cases} Si\ \ x\in D_1 \ \ \text{alors} & \begin{cases} A_{in1}\mathbf{k}_u \le B_{in1}x+b_{in1} \\ A_{eq1}\mathbf{k}_u = B_{eq1}x+b_{eq1} \end{cases} \\ \vdots & \\ Si\ \ x\in D_{n_D} \ \ \text{alors} & \begin{cases} A_{in n_D}\mathbf{k}_u \le B_{in n_D}x+b_{in n_D} \\ A_{eq n_D}\mathbf{k}_u = B_{eq n_D}x+b_{eq n_D} \end{cases} \end{cases} \tag{3.56}$$

ou en utilisant une représentation duale des générateurs non redondants par morceaux :

$$P(x)=\begin{cases} Si\ \ x\in D_1 \ \ \text{alors} \ \ \mathbf{k}_u=\sum\limits_{i\in IPV_1}\sigma_i s_i+\sum\limits_{i=1}^{n_R}\rho_i r_i+\sum\limits_{i=1}^{n_L}\lambda_i l_i; \quad \sigma_i,\rho_i\ge 0; \quad \sum\limits_{i\in IPV_1}\sigma_i=1 \\ \vdots \\ Si\ \ x\in D_{n_D} \ \ \text{alors} \ \ \mathbf{k}_u=\sum\limits_{i\in IPV_{n_D}}\sigma_i s_i+\sum\limits_{i=1}^{n_R}\rho_i r_i+\sum\limits_{i=1}^{n_L}\lambda_i l_i; \quad \sigma_i,\rho_i\ge 0; \quad \sum\limits_{i\in IPV_{n_D}}\sigma_i=1 \end{cases} \tag{3.57}$$

où le nombre de régions de forme régulière de $P$ est $n_D$. Les ensembles $IPV_1,\ldots,\ IPV_{n_D}$ regroupent les indices des sommets valides pour chaque zone régulière. Ces sommets représentent des sous-ensembles de la totalité des sommets du polyèdre $P(x)$ initial : $\{s_1,\ldots,s_{n_S}\}$.

**Formulation par contraintes**

Si l'on s'intéresse aux programmes ayant pour objectif de déterminer des valeurs optimales parmi des valeurs faisables décrites par $P(x)$, l'existence de formes régulières implique que le jeu de contraintes actives pour le problème d'optimisation soit choisi parmi les indices de l'ensemble non redondant $\Im_i \subset \Im$ et non parmi tous les indices $\Im=\{1,\cdots,r\}$. Ainsi pour un problème quadratique multiparamétrique, une procédure basée sur des ensembles de contraintes non redondantes (3.56) sous une formulation implicite est la suivante.

***<u>Algorithme QP_RF1 (procédure à employer en ligne pour déterminer la commande optimale)</u>***

*Données d'entrée :* découpage de l'espace des paramètres faisables $D=D_1\bigcup D_2\bigcup\cdots\bigcup D_{n_D}$ et sous-ensembles des contraintes non redondantes correspondantes.

*Données de sortie :* solution optimale $\mathbf{k}_u^*$.

1) En utilisant le vecteur des paramètres de contexte ($x_t$), trouver l'ensemble des contraintes non redondantes en se positionnant dans une table préprogrammée :

*Si* $x_t\in D_1$ Sous-ensemble de contraintes non-redondantes : $\{(A_{in_1},B_{in_1},b_{in_1}),(A_{eq_1},B_{eq_1},b_{eq_1})\}$

… …

*Si* $x_t\in D_{n_D}$ Sous-ensemble de contraintes non-redondantes : $\{(A_{in_{n_D}},B_{in_{n_D}},b_{in_{n_D}}),(A_{eq_{n_D}},B_{eq_{n_D}},b_{eq_{n_D}})\}$

2) Résoudre le problème d'optimisation :

$$\mathbf{k}_u^*=\arg\min_{\mathbf{k}_u} 0.5\mathbf{k}_u^T H\mathbf{k}_u+x^T F\mathbf{k}_u$$

$$\begin{cases} A_{in_i}\mathbf{k}_u \ge B_{in_i}x+b_{in_i} \\ A_{eq_i}\mathbf{k}_u = B_{eq_i}x+b_{eq_i} \end{cases}$$

avec l'index $i$ correspondant au domaine trouvé à l'étape 1).

**Formulation par générateurs**

En utilisant la représentation (3.57), la même solution peut être construite issue d'un problème d'optimisation qui a recherché la meilleure combinaison des coefficients des générateurs. Dans ce cas, c'est le sous-ensemble des générateurs qui est non redondant.

***Algorithme QP_RF2 (procédure basée sur les sommets paramétrés)***

*Données d'entrée :* ensemble de rayons uni- et bidirectionnels ($\{1,\cdots,n_R\}$ et $\{1,\cdots,n_L\}$), découpage de l'espace des paramètres faisables $D = D_1 \cup D_2 \cup \cdots \cup D_{n_D}$ et sous-ensembles des sommets paramétrés valides correspondants, désignés par les indices $IPV_i \subset \{1,\cdots,n_S\}$.

*Données de sortie :* solution optimale $\mathbf{k}_u^*$.

1) En utilisant le vecteur des paramètres de contexte ($x_t$), trouver la matrice colonne contenant les sommets paramétrés valides (positionnement dans une table préprogrammée) :

$$\begin{array}{lll} Si & x_t \in D_1 & SomPar = \lfloor \cdots s_j \cdots \rfloor,\ j \in IPV_1 \\ \cdots & & \cdots \\ Si & x_t \in D_{n_D} & SomPar = \lfloor \cdots s_j \cdots \rfloor,\ j \in IPV_{n_D} \end{array}$$

2) $Gen = \lfloor SomPar \mid r_1 \ \cdots \ r_{n_R} \mid r_1 \ \cdots \ r_{n_R} \rfloor$

3) Résoudre le problème d'optimisation quadratique avec contraintes de positivité :

$$\{\{\sigma\};\{\rho\};\{\lambda\}\}^* = \underset{\{\sigma\};\{\rho\};\{\lambda\}}{\arg\min}\ 0{,}5\lfloor \sigma_1 \cdots \sigma_{n_{S_i}} \mid \rho_1 \cdots \rho_{n_R} \mid \lambda_1 \cdots \lambda_{n_L} \rfloor Gen^T H Gen \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_{n_{S_i}} \\ \hline \rho_1 \\ \vdots \\ \rho_{n_R} \\ \hline \lambda_1 \\ \vdots \\ \lambda_{n_L} \end{bmatrix} \mathbf{k}_u + x^T F^T \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_{n_{S_i}} \\ \hline \rho_1 \\ \vdots \\ \rho_{n_R} \\ \hline \lambda_1 \\ \vdots \\ \lambda_{n_L} \end{bmatrix}$$

sous les contraintes : $\{\sigma\};\{\rho\} \geq 0; \sum \sigma = 1$

4) $\mathbf{k}_u^* = \lfloor \sigma_1 \cdots \sigma_{n_{S_i}} \mid \rho_1 \cdots \rho_{n_R} \mid \lambda_1 \cdots \lambda_{n_L} \rfloor Gen^T$

Le choix entre la forme faisant intervenir les contraintes ou les générateurs dépend en fait de l'application. Il est évident que la formulation des contraintes dans ce dernier cas (générateur) est beaucoup plus simple, forçant en fait la recherche des combinaisons optimales des coefficients seulement dans le quadrant positif. Ceci se répercutera par exemple dans le cas des méthodes d'ensemble actif par des points sur la frontière plus faciles à identifier. De même, le remplacement des contraintes dans l'ensemble actif est plus facile. Le désavantage se fait sentir au niveau de la dimension de l'argument du problème d'optimisation. Ceci est en liaison directe avec le nombre de générateurs, chaque générateur ayant un coefficient associé. Comme on l'a déjà montré, dans le pire cas, la représentation duale (par générateurs) est assez complexe et la complexité du problème d'optimisation en souffre également.

**Autres considérations**

Notons que les algorithmes « QP_RF » nécessitent l'implémentation d'une procédure de positionnement dans une table préprogrammée de régions polyédrales ('look-up table'). Cette opération doit être réalisée avec précaution car dans le cas de découpages complexes, la charge informatique peut devenir importante. Parmi les méthodes possibles, citons :

- Une recherche séquentielle parmi les régions de la partition polyédrale, en vérifiant chaque système de contraintes. Cette méthode s'avère particulièrement inefficace car, dans le pire cas, elle nécessite l'évaluation de tous les ensembles de contraintes existants. Son seul avantage réside dans le fait qu'elle ne nécessite pas de constructions auxiliaires pour être implémentée.
- Un algorithme efficace de positionnement basé sur le placement des régions polyédrales dans les feuilles d'un arbre de recherche binaire. Chaque nœud de l'arbre est défini par une inégalité et le

mécanisme de positionnement les utilise pour évaluer la position du paramètre de contexte courant. La procédure commence dans le nœud racine et parcourt l'arbre jusqu'à la feuille qui représente le domaine valide à l'instant courant. Il faut mentionner que la construction de l'arbre binaire n'est pas immédiate, elle nécessite l'introduction de coupures supplémentaires dans les partitions déjà existantes et que la difficulté principale réside dans la construction d'un arbre bien équilibré. Tondel, Johansen et Bemporad [TJB01] ont proposé un algorithme pour construire un arbre binaire avec un temps de positionnement logarithmique dans le nombre de partitions initiales.

- La navigation entre les domaines correspondant aux formes régulières via un graphe de coïncidence pour les régions polyédrales dans l'espace des paramètres. Une telle construction peut être développée théoriquement si l'on considère que l'évolution des paramètres de contexte s'exprime par une fonction multivariable continue.

**Proposition 3.28 :** Si les paramètres de contexte d'un problème mpQP suivent une évolution décrite par une fonction continue, alors la transition entre les régions de forme régulière suit les transitions dans un graphe bidirectionnel défini de façon unique.

*Preuve :* La procédure de découpage fournit une décomposition :

$$D = \bigcup_{i=1}^{n_D} D_i \subset \Re^n \tag{3.58}$$

de l'espace des paramètres. Si les paramètres suivent une évolution continue, ils ne peuvent pas passer d'un domaine $x_{t_1} \in D_i$ à l'autre $x_{t_2} \in D_j$ sans transgresser la frontière entre les deux domaines. Il existe donc un point $\tau \in [t_1, t_2]$ tel que $x_\tau \in D_i \cap D_j$ (implication directe de la continuité). Chaque passage de la frontière d'un domaine peut être représenté comme transition dans un graphe bidirectionnel, ayant comme noeuds les domaines du découpage et arcs toutes les intersections non vides de domaines. ■

La conséquence de ce résultat est que chaque variation de paramètres correspond à une transition possible dans le graphe de coïncidence. Toutefois, dans les applications réelles de commande prédictive, l'hypothèse du théorème précédent n'est pas toujours vérifiée, car les évolutions de paramètres ne font pas toujours partie de la classe des fonctions continues. Ainsi par exemple, dans le cas GPC, l'existence des contraintes terminales implique que les paramètres de contexte incluent les valeurs futures de la référence. Clairement, le découpage dans l'espace de paramètres induit des découpages dans ce sous-espace des références futures. Or un signal de consigne rencontré fréquemment est celui composé de successions d'échelons, et ce type de signal n'évolue pas continûment au cours du temps. Des sauts entre des régions sans frontière commune deviennent alors possibles. Plus généralement, le fait que la commande prédictive utilise des modèles à temps discret implique que l'évolution des paramètres de contexte est décrite par une fonction pouvant être discontinue à chaque pas d'échantillonnage. Le résultat du théorème précédent n'est cependant pas inutile, car, même si des sauts sont possibles, les transitions vers les régions adjacentes au nœud valide au pas d'échantillonnage précédent sont les plus probables et doivent être vérifiées en priorité.

### 3.4.5 Exemple

Afin d'illustrer les méthodes précédentes, considérons la loi GPC avec contrainte mise en œuvre sur un système double intégrateur :

$$y_t = \frac{1}{1 - 2q^{-1} + q^{-2}} u_{t-1} \tag{3.59}$$

avec des contraintes sur l'incrément et l'amplitude de commande :

$$\begin{cases} -0{,}06 \le \Delta u_{t+k} \le 0{,}06; k = 0..N_u - 1 \\ -0{,}1 \le u_{t+k} \le 0{,}1; \quad k = 0..N_u - 1 \end{cases} \tag{3.60}$$

La loi de commande considérée a été obtenue avec les paramètres de réglage suivants $N_u = 2; N_1 = 1; N_2 = 3; \mathbf{\Lambda} = 0{,}4 \text{ trace}(\mathbf{G}^T\mathbf{G})\,\mathbf{I}$. Notons que, dans la formulation de base, le domaine faisable pour le problème d'optimisation ne fait intervenir qu'un seul paramètre de contexte, l'entrée passée. Ainsi le domaine paramétré est donné par :

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ -1 & -1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \Delta u_t \\ \Delta u_{t+1} \end{bmatrix}}_{\mathbf{k}_u} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} u_{t-1} + \begin{bmatrix} -0.06 \\ -0.06 \\ -0.06 \\ -0.06 \\ -0.1 \\ -0.1 \\ -0.1 \\ -0.1 \end{bmatrix} \tag{3.61}$$

En plus de ces contraintes explicites, on peut ajouter des contraintes implicites sur le paramètre :

$$-0{,}1 \leq u_{t-1} \leq 0{,}1 \tag{3.62}$$

Ainsi la représentation géométrique de ce polyèdre sous forme augmentée (arguments + paramètres) $[\Delta u_t \;\; \Delta u_{t+1} \;\; u_{t-1}]$ est donnée par le polytope de la Figure 3.10. Le domaine faisable bidimensionnel varie donc en fonction du paramètre, ici l'entrée passée. La représentation duale des générateurs est constituée de 14 sommets décrits dans le Tableau 3.1.

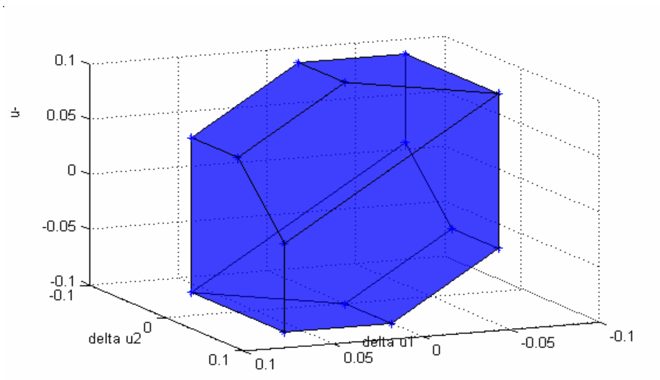



**Figure 3.10 : Polyèdre faisable dans l'espace étendu** $[\Delta u_t \;\; \Delta u_{t+1} \;\; u_{t-1}]$.

| | Δu(t) | Δu(t+1) | u(t-1) |
|---|---|---|---|
| **Sommets** | 0.06 | -0.06 | -0.1 |
| | -0.06 | 0.06 | 0.1 |
| | 0.06 | -0.06 | 0.04 |
| | 0.06 | 0 | 0.04 |
| | -0.06 | -0.06 | 0.02 |
| | 0.06 | 0.06 | -0.02 |
| | 0 | -0.06 | 0.1 |
| | -0.06 | 0.06 | -0.04 |
| | -0.06 | 0 | -0.04 |
| | 0 | 0 | 0.1 |
| | -0.06 | -0.06 | 0.1 |
| | 0 | 0 | -0.1 |
| | 0 | 0.06 | -0.1 |
| | 0.06 | 0.06 | -0.1 |

**Tableau 3.1 : Représentation par générateurs du domaine faisable.**

Les détails de l'objet polyédral obtenu via l'algorithme de Chernikova implémenté sous Matlab sont les suivants :

| | |
|---|---|
| ***P(x) :*** | *Dimension: 3* |
| | *NbConstraints: 10* |
| | *NbRays: 14* |
| | *NbEq: 0* |
| | *NbBid: 0* |
| | *Rays: [14x5 double]* |
| | *Constraints: [10x5 double]* |

Ce polyèdre possède 20 facettes de dimension 1 (arêtes), mais seulement 10 d'entre elles ont une projection non dégénérée sur l'espace des paramètres correspondant aux sommets paramétrés du domaines faisable pour le problème d'optimisation. En utilisant les 10 sommets paramétrés et leurs domaines de validité, le procédé de découpage fournit les régions de l'espace des paramètres ayant une forme régulière :

$$\begin{array}{ll} D_1 : \ -0.1 \le u_{t-1} \le -0.04 & D_2 : \ -0.04 \le u_{t-1} \le -0.02 \\ D_3 : \ -0.02 \le u_{t-1} \le 0.02 & D_4 : \ 0.02 \le u_{t-1} \le 0.04 \\ \multicolumn{2}{c}{D_5 : \ 0.04 \le u_{t-1} \le 0.1} \end{array} \tag{3.63}$$

Le résultat de ce découpage dans l'espace de paramètres est décrit Figure 3.11 et représente la base du procédé d'optimisation en ligne.

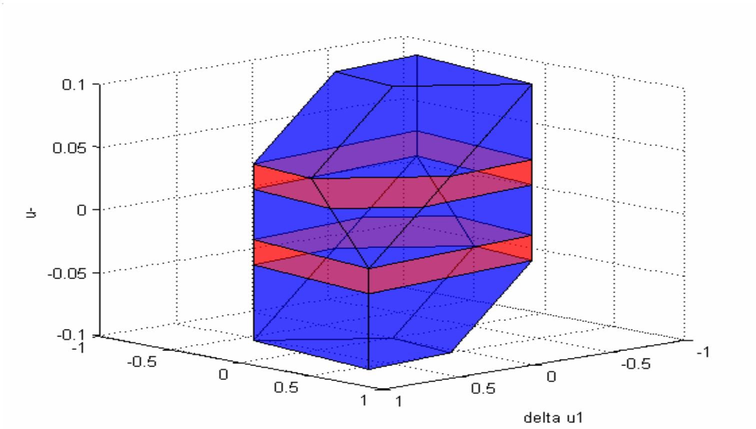



**Figure 3.11 : Régions de forme régulière.**

Il est évident qu'une évolution continue du signal de commande impliquera une transition à l'intérieur du graphe de coïncidence représenté Figure 3.12. C'est la raison pour laquelle les solveurs en ligne recherchent en priorité la solution parmi les nœuds adjacents au noeud courant. Les discontinuités de consigne induisent des discontinuités dans l'évolution des paramètre et en conséquence des sauts. Cependant, comme mentionné ci-dessus, dans la plupart des cas les transitions adjacentes sont privilégiées.

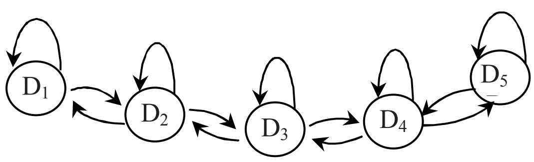



**Figure 3.12 : Graphe pour une évolution continue dans l'espace des paramètres du polyèdre.**

Le comportement en ligne de la boucle fermée avec une loi de commande GPC sous contraintes implémentée sous une forme sans redondance est illustré Figure 3.13 et traduit un comportement identique à celui observé lorsque le problème d'optimisation multiparamétrique initial est résolu à chaque pas d'échantillonnage. On obtient ainsi une première confirmation de la validité de la méthode. La réduction en terme de charge de calcul issue de la démarche n'est cependant pas évidente pour ce cas simple car le degré de redondance du problème initial n'est pas important.

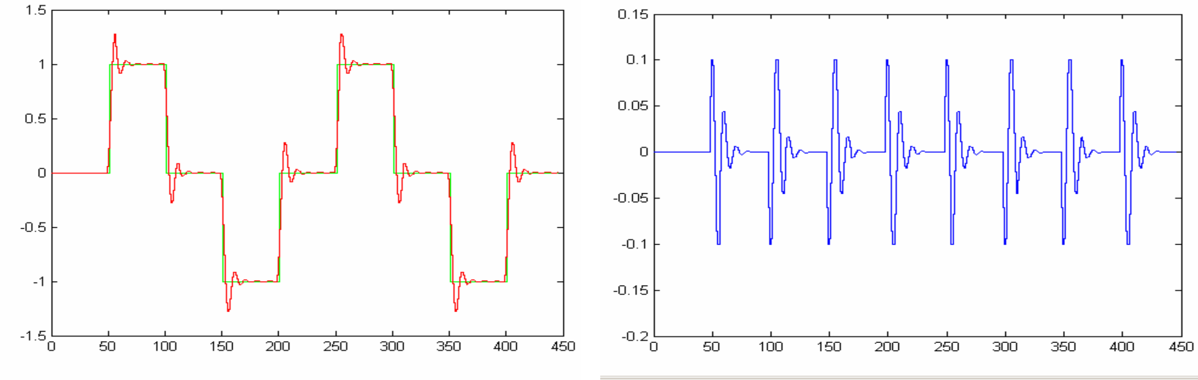



**Figure 3.13 : Comportement du système en boucle fermée.**

Néanmoins, l'approche classique résout un problème quadratique avec 10 contraintes tandis qu'en utilisant la technique proposée, le nombre de contraintes est réduit à 5 ou 6 au maximum. Le prix payé est le positionnement dans une des 5 régions (3.63), qui revient à deux ou trois comparaisons. La représentation duale par générateurs peut s'avérer intéressante car elle propose un programme quadratique avec 4 ou 5 variables sous des contraintes ordinaires de positivité.

## *3.5 Analyse de l'influence de la redondance*

Les paragraphes précédents ont permis d'élaborer des stratégies remplaçant le problème d'optimisation lié à la commande prédictive sous contraintes par des problèmes évitant la présence de redondance locale. Les algorithmes décrits construisent des découpages de l'espace des paramètres utilisés lors de la résolution en ligne. La question est dès lors de savoir si ce type de transformation peut engendrer des améliorations en terme de temps de calcul de la séquence de commande optimale à chaque pas d'échantillonnage. Pour résumer, il s'agit donc de déterminer les conséquences de la présence de redondance au sein des programmes d'optimisation en ligne. Les éventuels gains doivent être comparés aux pertes inhérentes au positionnement correspondant aux paramètres courants. Ainsi les avantages et inconvénients d'une éventuelle discrimination effectuée hors-ligne sur l'ensemble des contraintes affectées par la paramétrisation seront plus évidents.

### 3.5.1 Facteurs d'influence

Afin d'effectuer une comparaison quantitative réaliste entre les solveurs en ligne conventionnels et ces mêmes techniques additionnées d'un mécanisme permettant d'éviter la redondance, la dépendance de l'analyse à l'égard des paramètres suivants doit être prise en compte :

*Degré de redondance -* $dr_1 \cdots dr_{n_D}$

Il doit être mesuré pour chaque zone $D_1 \cdots D_{n_D}$ en se rapportant au problème original qui traite globalement tout l'ensemble $D = D_1 \cap D_{n_D}$. Formellement, il représente le rapport entre les nombres de contraintes redondantes vis-à-vis d'un domaine donné $D_i$ et la taille de l'ensemble global. Ainsi :

$$dr_i = \frac{\text{contraintes}_D - \text{contraintes}_{D_i}}{\text{contraintes}_D} * 100; \quad i = 1, \cdots n_D \tag{3.64}$$

Généralement, le degré de redondance $dr_i$ peut varier de façon conséquente d'une zone à l'autre et de ce fait, un indicateur moyen :

$$dr = \frac{1}{n_D} \sum_{i=1}^{n_D} dr_i \tag{3.65}$$

est préférable pour caractériser l'influence globale de la redondance au sein d'une description de contraintes. Si le degré de redondance n'est pas important pour toutes les régions de l'espace de paramètres, alors les sous-problèmes non redondants sont assez semblables au problème initial et par conséquent les gains seront faibles pour le solveur QP utilisé. Si par ailleurs le nombre de régions est important, le processus de positionnement est complexe et pour une redondance faible la décomposition en régions sans redondance n'est pas souhaitable. Au contraire, un degré important de redondance (en moyenne supérieure à 50%) pénalise les solveurs QP qui tentent de retrouver la solution optimale à partir de l'ensemble complet de contraintes.

*Nombre de régions* $n_D$

Le nombre de zones présentant une forme régulière résultant de la procédure de découpage est un indicateur essentiel pour l'évaluation du temps de calcul en ligne. Il offre une indication de la complexité de l'étape de positionnement qui accompagne l'optimisation en ligne dans le cas sans redondance. Idéalement, la charge de calcul lors du positionnement est logarithmique en fonction du nombre de régions. Pratiquement, la construction d'un arbre binaire de recherche peut nécessiter l'introduction de découpages supplémentaires. Dans ce cas, la profondeur de l'arbre résultant doit être prise en compte pour quantifier la charge de calcul lors du positionnement. Tous les gains éventuels doivent être pondérés par cet aspect.

*Nombre de contraintes* $r$

Il représente la totalité des contraintes non redondantes et peut être considéré comme un indicateur introduisant un aspect de relativité dans la mesure de la redondance. Il est évident que, pour un nombre infime de contraintes, la redondance, même si elle s'avère importante, n'influence pas de façon décisive la complexité du problème. En revanche, si la redondance est rapportée à un nombre important de contraintes, les gains peuvent être considérables, puisque le nombre d'itérations du programme QP est directement lié au processus d'identification de l'ensemble actif de contraintes. Un nombre élevé de contraintes augmente alors le nombre d'itérations. Il faut aussi remarquer que la taille des matrices manipulées par le solveur est liée au nombre de contraintes et donc pour un nombre élevé d'itérations la complexité des opérations primaires s'en trouve augmentée.

*Nombre d'itérations effectuées par le solveur QP (méthodes d'ensemble actif)*

Le nombre d'itérations effectuées par les solveurs n'est pas influencé par l'existence de contraintes redondantes car les décisions sont prises généralement en se basant sur les contraintes les plus contraignantes. Cependant il existe une différence au niveau de la charge de calcul par itération, principalement liée à la taille des matrices qui se trouve inutilement augmentée en présence de redondance. Comme les contraintes doivent être évaluées à chaque itération, un nombre élevé d'itérations souligne les avantages d'une représentation minimale. Il faut aussi noter que les solveurs puissants utilisent l'évaluation des multiplicateurs de Lagrange et donc l'existence de redondance ajoute des multiplicateurs inutiles (qui sont nuls par définition). Malgré tout, le nombre d'itérations n'est pas une variable maîtrisable, car dépendante du point initial, de la position de l'optimum global et de la structure du critère quadratique. Par exemple, lorsque l'optimum global se situe à l'intérieur de l'ensemble faisable, le solveur QP effectuera une seule itération et la redondance des contraintes ne constituera pas un inconvénient majeur.

*Nombre d'arguments du problème d'optimisation -* $p$

Il représente en fait la taille de $\mathbf{k}_u$. Généralement, des horizons de commande importants impliquent un espace de recherche pour la solution optimale plus conséquent. Une augmentation du nombre de variables de décision accroît de fait le nombre d'itérations à chaque pas d'échantillonnage. Ce paramètre n'influence pas directement l'analyse 'redondance vs. non redondance'. Indirectement, il détermine la taille des inégalités à évaluer et donc la complexité du problème de programmation quadratique.

*Dimension de l'espace des paramètres -* $n$

Il représente un indicateur lié avant tout à la classe de problèmes qu'il faut résoudre. Il est en relation directe avec la complexité du découpage, de la table préprogrammée et donc du mécanisme de positionnement. Par exemple, les zones dans l'espace des paramètres sont des intersections de demi-espaces et à chaque pas d'échantillonnage, le positionnement doit se faire en évaluant la position par rapport à de telles contraintes dont la taille est liée à la dimension de l'espace des paramètres.

**Remarque :** Le nombre d'itérations dans les méthodes de type 'ensemble actif' *n'est pas influencé par la présence de redondance* dans l'ensemble de contraintes. La redondance influence exclusivement le temps par itération.

### 3.5.2 Etude de cas

On propose ci-dessous un exemple numérique permettant d'illustrer toutes les influences mentionnées précédemment. Le problème de commande prédictive a été choisi tel que les commandes optimales se trouvent principalement sur la frontière du domaine faisable. Le domaine faisable a été construit de sorte que l'ensemble de contraintes puisse être facilement transformé sans modifier l'allure ou la structure de la commande.

***Description***

On considère le système :

$$\begin{bmatrix} x_{t+1}^1 \\ x_{t+1}^2 \end{bmatrix} = \begin{bmatrix} 0.9 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t^1 \\ x_t^2 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u_t \tag{3.66}$$

La séquence optimale appliquée selon le principe d'horizon glissant est basée sur le critère :

$$\min_{\{u_t,\ldots,u_{t+N_u-1}\}} \sum_{k=1}^{N_y} x_{t+k}^T Q x_{t+k} + \sum_{k=0}^{N_u-1} u_{t+k}^T R u_{t+k} \tag{3.67}$$

faisant intervenir les paramètres suivants :

$$N_u = 2; N_y = 3; Q = \begin{bmatrix} 10 & 0 \\ 0 & 100 \end{bmatrix}; R = 1 \tag{3.68}$$

En conséquence, le problème d'optimisation possède deux arguments :

$$\mathbf{k}_u = [u_t \ u_{t+1}]^T \tag{3.69}$$

***Contraintes***

La première contrainte dure à satisfaire se caractérise par :

$$x_2 \geq 0 \tag{3.70}$$

Deux autres types de contraintes sont également pris en compte :

1. *Contraintes non paramétrées :* Des polyèdres réguliers (pentagone, icosaèdre, hexagone …) inscrits dans le cercle de rayon unité comme sur la Figue 3.14 sont considérés comme domaines faisables pour les commandes allant de $t$ jusqu'à $t+N_u$. Formellement elles sont décrites par :

$$A_0 \begin{bmatrix} u_t \\ u_{t+1} \end{bmatrix} \geq \underbrace{\mathbf{1}_{NL}}_{b_0} \tag{3.71}$$

avec $\mathbf{1}_{NL}$ un vecteur de dimension $NL$ de coefficients unité, $A_0 \in \Re^{NL\times 2}$ et $NL$ le nombre de côtés du polytope. On peut noter que les coefficients de $A_0$ peuvent être interprétés comme les coefficients des points sur le cercle unité par dualité des représentations contraintes/générateurs.

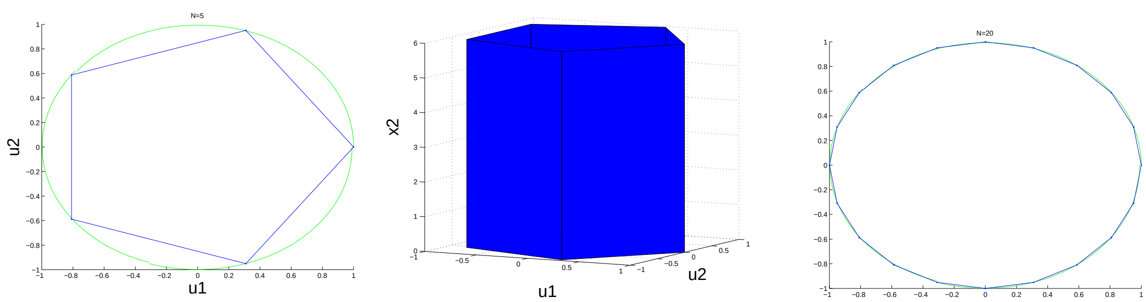



**Figure 3.14 : a) Domaine faisable pour $NL = 5$. b) Domaine faisable en fonction du paramètre $x_2$ c) Domaine faisable pour $NL = 20$.**

2. *Contraintes paramétrées :* Des contraintes similaires peuvent être considérées, avec la différence que le rayon du cercle est dépendant de l'état actuel du système. Si l'on considère une simple dépendance linéaire des contraintes vis-à-vis du paramètre $x_2$, on peut écrire :

$$\underbrace{A_0}_{=A_1}\begin{bmatrix} u_t \\ u_{t+1} \end{bmatrix} \geq \underbrace{\mathbf{0}_{NL}}_{b_1} + \underbrace{\begin{bmatrix} 0 & 0.6 \\ 0 & 0.6 \end{bmatrix}}_{B_1}\begin{bmatrix} x_t^1 \\ x_t^2 \end{bmatrix} \tag{3.72}$$

avec comme conséquence la description d'un domaine polyédral paramétré suivant une décroissance de son cercle circonscrit.

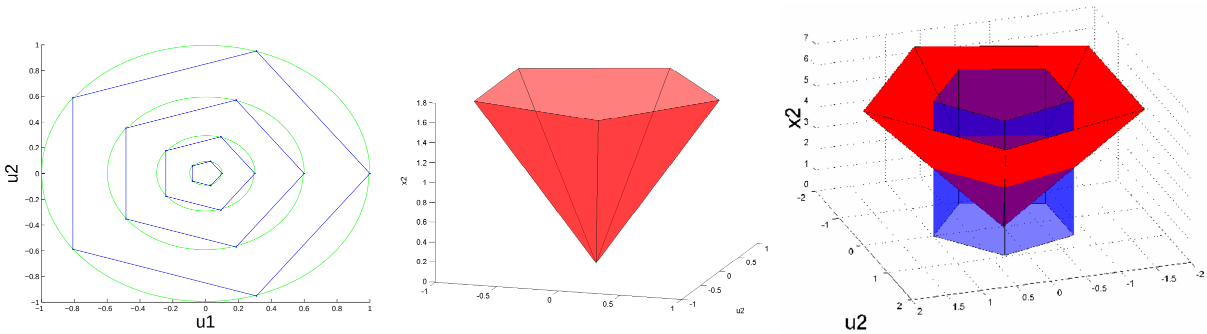



**Figure 3.15 : a) Domaines faisables paramétrés b) Dépendance du domaine faisable vis-à-vis de $x_2$ c) Contraintes paramétrées et non-paramétrées**

D'une façon similaire, d'autres ensembles de contraintes paramétriques peuvent être ajoutés en modifiant le jeu de matrices $\{A_1, B_1, b_1\}$ et en introduisant des dépendances linéaires différentes $\{A_2, B_2, b_2\}$, $\{A_{ML}, B_{ML}, b_{ML}\}$, comme Figure 3.16.

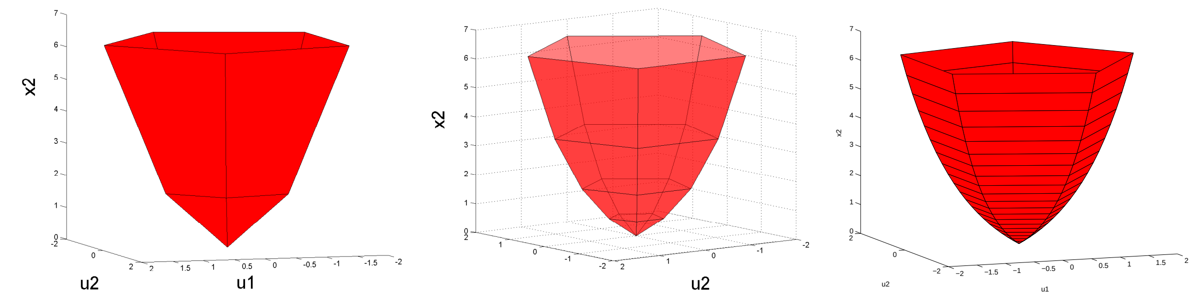



**Figure 3.16 : Domaines faisables paramétrés a) $ML = 2$ b) $ML = 4$ c) $ML = 20$.**

Le nombre de côtés des polygones réguliers ($NL$) et le nombre de dépendances linéaires dans le paramètre $x_2$ (ML) donnent un degré de liberté important lors de la définition de l'ensemble de contraintes. Ces nombres peuvent être modifiés facilement sans modifier pour autant la philosophie de la commande prédictive choisie et permettant donc de comparer différentes conceptions des programmes d'optimisations en ligne. Si ces ensembles de contraintes paramétrées sont choisis afin de représenter une approximation linéaire par morceaux pour l'évolution croissante exponentielle du rayon du cercle circonscrit, à la limite, quand $NL \to \infty$ et $ML \to \infty$, un ensemble non linéaire de contraintes peut être approché :

$$\begin{cases} \sum_{k=0}^{N_u} u_{t+k}^2 \leq 1 \\ \sum_{k=0}^{N_u} u_{t+k}^2 \leq \ln(x_t^2 + 1) \end{cases} \tag{3.73}$$

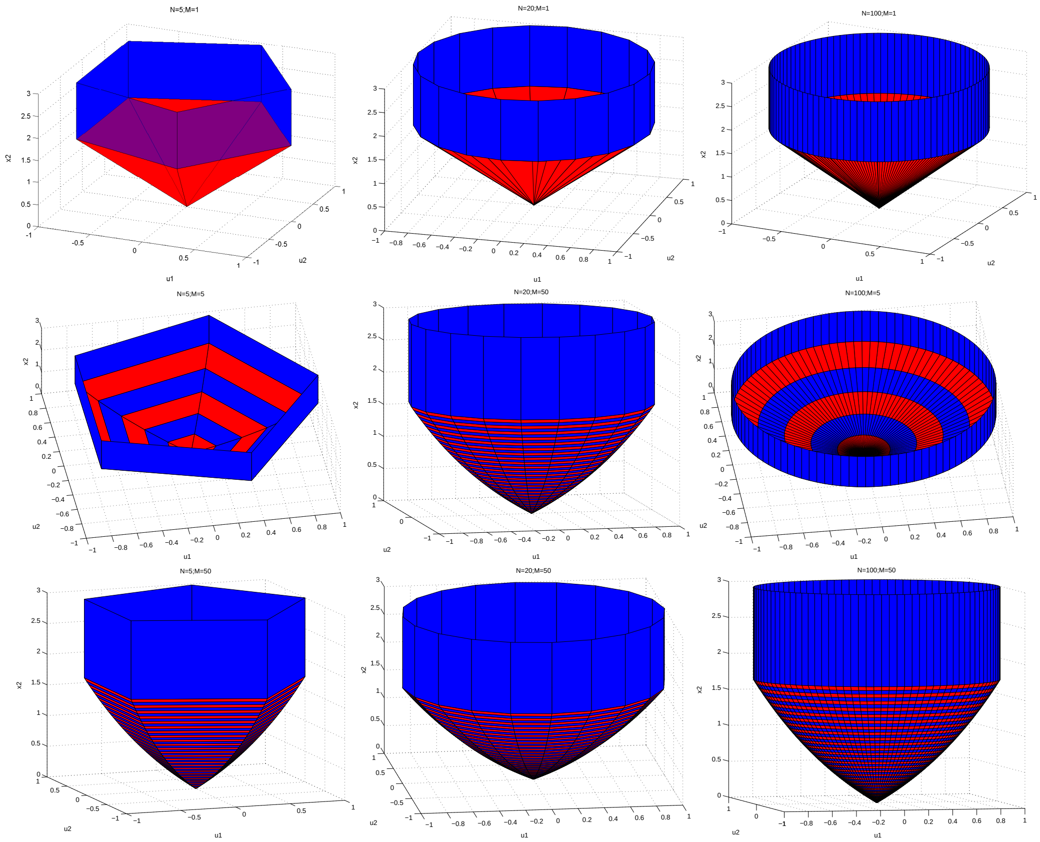


**Figure 3.17 : Ensembles de contraintes pour différentes complexités.**
**a) ML=1,NL=5 b) ML=1, NL=20 c) ML=1, NL=100 d) ML=5,NL=5 e) ML=5, NL=20**
**f) ML=5, NL=100 g) ML=50, NL=5 h) ML=50, NL=20 i) ML=50, NL=100.**

**Remarque :** On peut noter que si l'état initial satisfait (3.70) alors par la structure de l'ensemble de contraintes (3.71, 3.72), la condition initiale devient superflue.

La simulation de la trajectoire de l'état du système bouclé par une loi prédictive sous les contraintes décrites précédemment pour $NL = 10$ et $ML = 1$ est représentée Figure 3.18a, alors que les séquences optimales correspondantes sont présentées dans Figure 3.18b. Notons que l'optimum sous contraintes se trouve toujours sur la frontière du domaine faisable. Ce fait permettra d'évaluer les performances des solveurs en ligne contraints d'effectuer plusieurs itérations afin de trouver la solution.

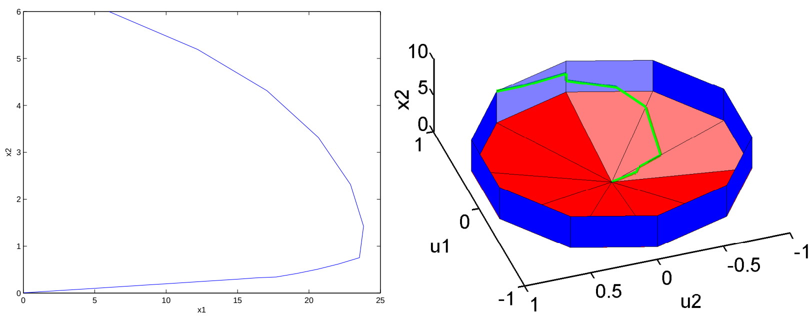
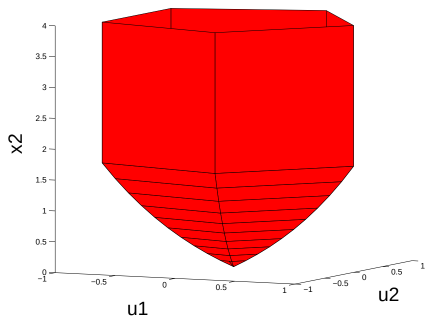



**Figure 3.18 : a) Evolution de l'état du système commandé**
**b) Distribution des commandes optimales en fonction du paramètre $x_2$.**

Pour le système décrit par les ensembles de contraintes paramétrées et non-paramétrées, un découpage de l'espace des paramètres (les états du système) correspondant aux domaines faisables de forme régulière peut être mis en évidence, Figure 3.19.



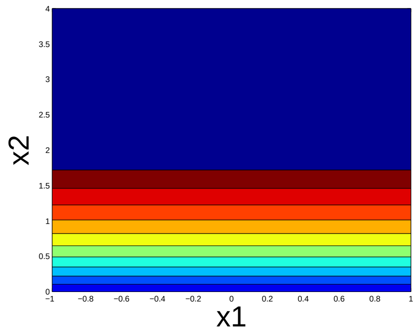




**Figure 3.19 : Découpage de l'espace d'état correspondant aux domaines faisables décrits par un même ensemble de contraintes.**

Les paramètres significatifs de la comparaison entre les programmes d'optimisation en ligne mentionnés précédemment (QP_RF1 et QP_RF2) seront déterminés pour l'exemple suivant :

- *nombre d'arguments du problème d'optimisation -* $p$. Il est donné par la dimension du vecteur $\mathbf{k}_u$, donc $p = 2$.
- *dimension de l'espace des paramètres -* $n = 2$
- *nombre de régions* $n_D$. Le nombre de régions est influencé exclusivement par l'ensemble des contraintes paramétrées. On a $n_D = ML$.
- *degré de redondance -* $dr_1 \cdots dr_{n_D}$. De façon similaire, ce paramètre est influencé par l'ensembles de contraintes paramétrées et dans notre cas on peut écrire $dr_i = 1/(ML+1)$.
- *nombre de contraintes* $r$. Il est donné par le jeu de contraintes paramétrées choisi et par le nombre de côtés de chaque ensemble. Ainsi on peut dire que $r = NL(ML+1)$.
- *nombre d'itérations effectuées par le solveur QP.* Le nombre d'itérations dépend de l'état du système, plus spécifiquement la valeur de $x_2$ modifie le domaine faisable, et les deux valeurs $x_1, x_2$ interviennent dans la fonction de coût, et vont donc modifier le nombre de pas nécessaires à l'élaboration de la solution optimale. Le fait que le nombre de côtés de l'ensemble faisable puisse varier induit aussi des variations du nombre d'itérations des programmes d'optimisation. Dans les simulations suivantes, le nombre de simulations change dans l'intervalle 2,..,21 avec des valeurs importantes pour les polyèdres réguliers de taille élevée. Quelques variations peuvent apparaître pour les programmes QP si le point initial évolue.

**Analyse de la charge informatique : ensemble complet de contraintes vs. ensembles non redondants par morceaux**

A partir de la description des contraintes (3.71-3.72), il apparaît qu'avec l'évolution du paramètre, seule une partie des contraintes est utile pour décrire la région faisable pour le problème d'optimisation. La mise en évidence des zones de l'espace des paramètres à contraintes non redondantes est immédiate à partir des programmes développés auparavant. Ainsi la charge informatique pour les solveurs en ligne peut être mesurée pour l'ensemble complet et pour des sous-ensembles non redondants localement. Par la suite, une analyse est présentée dans cette direction, sachant que les temps de calcul ont été obtenus en utilisant le mécanisme 'tic-toc' moyenné ensuite sur 100 simulations. Des indicateurs sont alors employés pour mesurer la différence de charge informatique :

$$Ef(t) = \frac{Tcall_r(t)}{Niter_r(t)} - \frac{Tcall_{nr}(t)}{Niter_{nr}(t)} \tag{3. 74}$$

avec $Niter_{r/nr}$ le nombre d'itérations dans le cas redondant/non-redondant pour le problème d'optimisation à résoudre à l'instant *t*. $Tcall_{r/nr}(t)$ est le temps effectif de calcul pour l'appel des solveurs. Ces indicateurs permettent une comparaison indépendante des conditions spécifiques du système à commander qui influencent le nombre d'itérations par appel. Pour obtenir un indicateur global de la charge de calcul par itération, la moyenne de $Ef(t)$ pour toute la simulation $(1 \ldots T_{sim})$ peut être calculée :

$$\bar{g} = \frac{\sum_{t=1}^{T_{sim}} Ef(t)}{T_{sim}} \tag{3.75}$$

Si l'on veut ignorer la fluctuation du nombre d'itérations pendant la simulation, on peut se baser uniquement sur les moyennes du temps de calcul par appel des programmes d'optimisation :

$$T_{nr} = \frac{\sum_{t=1}^{T_{sim}} Tcall_{nr}(t)}{T_{sim}} ; T_r = \frac{\sum_{t=1}^{T_{sim}} Tcall_r(t)}{T_{sim}} \tag{3.76}$$

- $NL = 5,\ ML = 1$

Ce cas correspond à un degré de redondance de $dr = 50\%$ pour chaque région. Le temps de calcul est présenté Figure 3.20a, ainsi que le nombre d'itérations pendant la simulation (Figure 3.20b) et le gain par itération $Ef(t)$ Figure 3.20c. On observe qu'il n'y a aucune différence entre les deux cas, le gain moyen par itération étant très faible $\bar{g} = 3.02 * 1e-7$. Le temps total par simulation vaut $T_{nr} = 100.22\%\ T_r$, rapport pouvant s'expliquer par le fait que le positionnement dans la région sans redondance s'ajoute au temps de calcul effectif qui reste semblable pour les deux approches. En rouge on retrouve le cas avec redondance et en bleu le cas sans redondance.

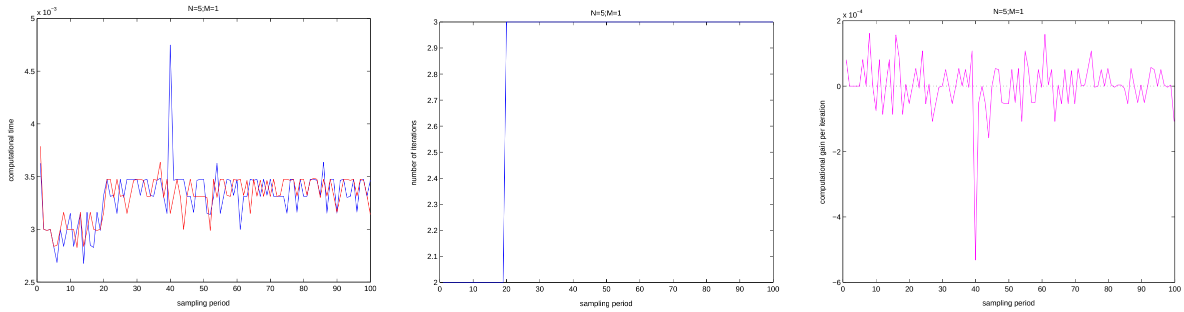



**Figure 3.20 : NL=5, ML=1 a) Temps d'appel
b) Nombre d'itérations QP (ensemble actif) c) gain par itération.**

- $NL = 20,\ ML = 1$

Le gain en temps moyen de calcul $T_{nr} = 97.11\%\ T_r$ est peu significatif pour ce problème de degré de redondance $dr = 50\%$. Les variations de $Ef(t)$ demeurent importantes même si l'utilisation d'ensembles non redondants améliore le temps par itération avec $\bar{g} = 4.05 * 1e-5$ en moyenne.

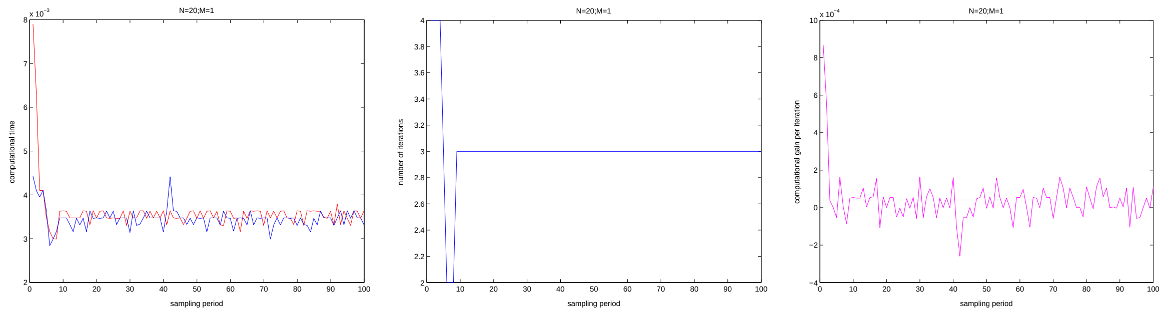



**Figure 3.21 : NL=20, ML=1 a) Temps d'appel
b) Nombre d'itérations QP (ensemble actif) c) gain par itération.**

- $NL = 100,\ ML = 1$

Même si le degré de redondance reste inchangé, on observe que l'augmentation du nombre de contraintes non redondantes influence le temps de calcul. Ceci est dû aussi à l'augmentation des itérations par appel. On observe des gains en temps de calcul à chaque pas d'échantillonnage qui se manifestent aussi sur les indicateurs globaux $T_{nr} = 90.8\%\ T_r$ et $\overline{g} = 7.87 * 1e\text{-}5$ .

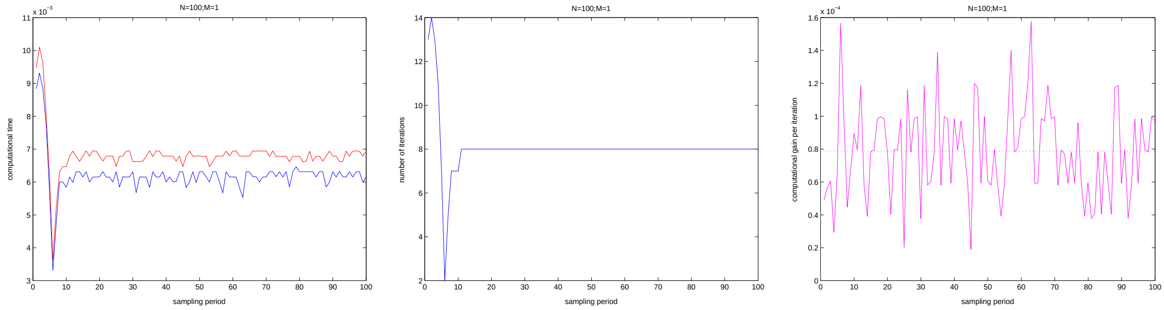



**Figure 3.22 : NL=100, ML=1 a) Temps d'appel**
**b) Nombre d'itérations QP (ensemble actif) c) gain par itération.**

- $NL = 5,\ ML = 5$

Ce cas correspond à un degré de redondance élevé mais aussi à un jeu de contraintes non redondants simple impliquant une optimisation avec peu d'itérations. Le résultat est un faible avantage pour le cas sans redondance $T_{nr} = 95\%\ T_r$ .

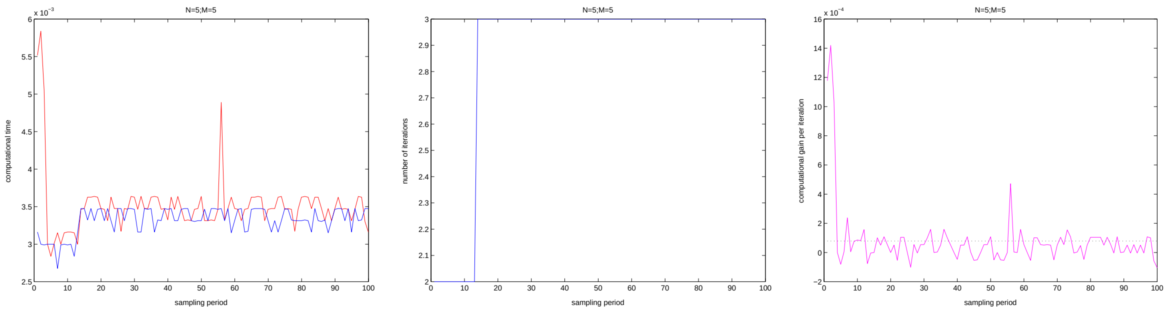



**Figure 3.23 : NL=5, ML=5 a) Temps d'appel**
**b) Nombre d'itérations QP (ensemble actif) c) gain par itération.**

- $NL = 20,\ ML = 5$

Le nombres de côtés du polyèdre régulier est plus important et donc la complexité de l'ensemble de contraintes non redondantes est accrue, d'où des gains en temps de calcul, même si le nombre d'itérations par optimisation reste dans l'intervalle 2 ...4.

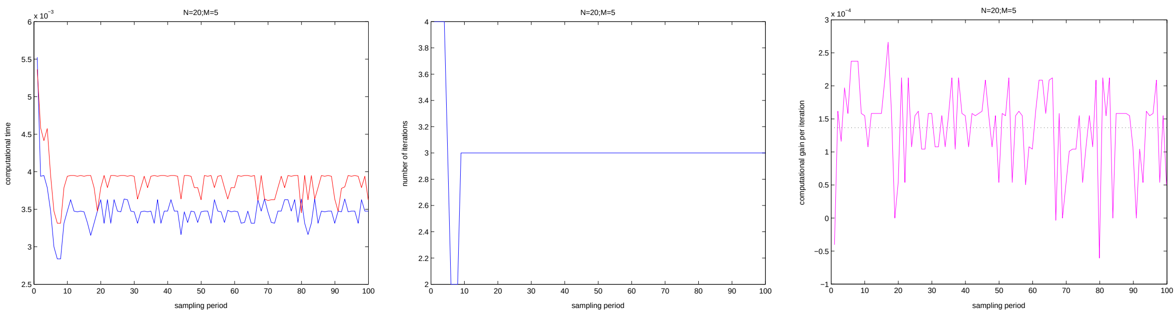



**Figure 3.24 : NL=20, ML=5 a) Temps d'appel**
**b) Nombre d'itérations QP (ensemble actif) c) gain par itération.**

- $NL = 100,\ ML = 5$

Le nombre d'itérations à chaque pas d'échantillonnage s'accroît, et l'influence de la redondance est claire avec des gains allant jusqu'à 30%, $T_{nr} = 70{,}42\%\ T_r$ .

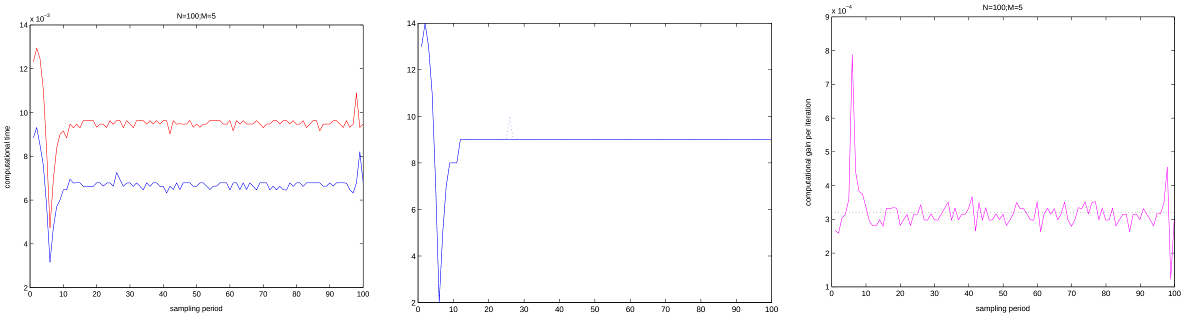



**Figure 3.25 : NL=100, ML=5 a) Temps d'appel**
**b) Nombre d'itérations QP (ensemble actif) c) gain par itération.**

- $NL = 5,\ ML = 50$

Cette catégorie de simulations pousse le degré de redondance à des valeurs très élevées, $dr = 98\%$. Dans ce cas, les gains avec l'approche sans redondance sont spectaculaires, mais encore une fois dépendants de la taille de l'ensemble de contraintes non redondantes par morceaux.

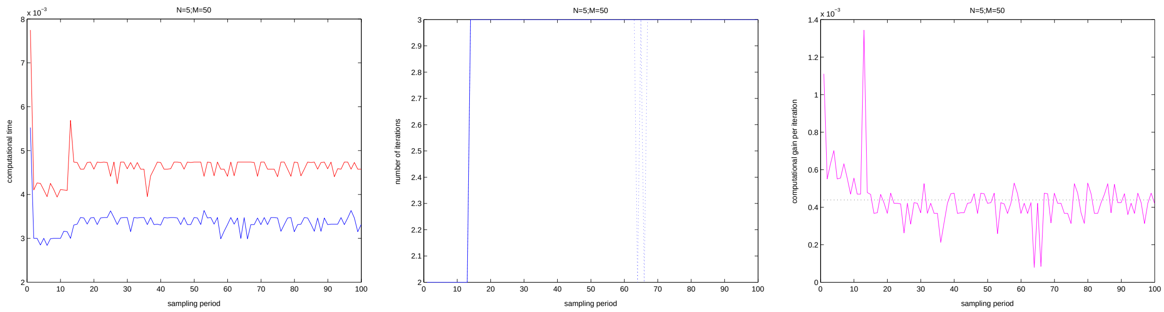



**Figure 3.26 : NL=5, ML=50 a) Temps d'appel**
**b) Nombre d'itérations QP (ensemble actif) c) gain par itération.**

- $NL = 20,\ ML = 50$

Le gain en temps de calcul s'élève à $T_{nr} = 46\%\ T_r$ correspondant à un gain par itération de $\overline{g} = 1{,}4*1e\text{-}3$.

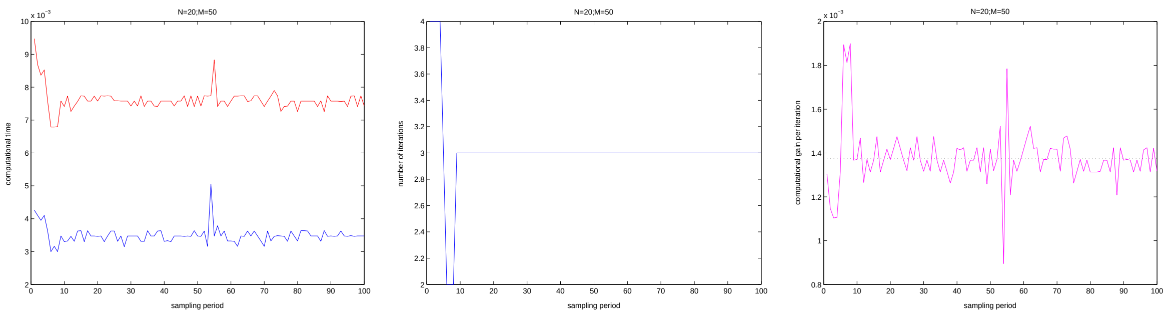



**Figure 3.27 : NL=20, ML=50 a) Temps d'appel**
**b) Nombre d'itérations QP (ensemble actif) c) gain par itération.**

- $NL = 100,\ ML = 50$

L'effort informatique diminue à $T_{nr} = 19\%\ T_r$ essentiellement dû à l'ensemble complet qui contient environ 5000 contraintes, même si les programmes en ligne doivent utiliser un mécanisme de positionnement dans les 51 régions possibles.

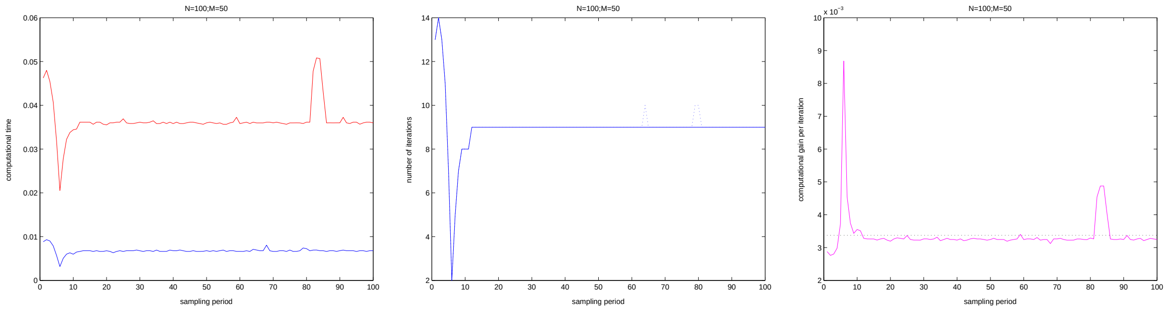



**Figure 3.27 : NL=100, ML=50 a) Temps d'appel**
**b) Nombre d'itérations QP (ensemble actif) c) gain par itération.**

La Table 3.2 fournit une description complète des paramètres et des indices de charge de calcul pour des choix divers des ensembles des contraintes (récapitulant ainsi les exemples précédents). On note le cas extrême correspondant à un degré de redondance faible et des ensembles de contraintes non redondantes simples pour lequel on ne retrouve aucune amélioration du temps de calcul. Les cas impliquant des degrés de redondance importants induisent des améliorations conséquentes, mais avec une forte dépendance vis-à-vis du nombre d'itérations par optimisation.

| Choix | | Paramètres | | | | | Indices | |
|---|---|---|---|---|---|---|---|---|
| ML | NL | $dr$ | $nd$ | $r$ | $\overline{Iter}$ | $n$ | $\bar{g}$ | $\frac{T_{nr}}{T_r}$ |
| 1 | 5 | 50% | 2 | 10 | 2.81 | 2 | $3.02 * 1e - 7$ | 100.22% |
| 1 | 20 | 50% | 2 | 40 | 3.01 | 2 | $4.05 * 1e - 5$ | 97.11% |
| 1 | 100 | 50% | 2 | 200 | 8.06 | 2 | $7.87 * 1e - 5$ | 90.08% |
| 5 | 5 | 83.33% | 6 | 30 | 2.87 | 2 | $7.81 * 1e - 5$ | 95.00% |
| 5 | 20 | 83.33% | 6 | 120 | 3.01 | 2 | $1.36 * 1e - 4$ | 89.50% |
| 5 | 100 | 83.33% | 6 | 600 | 8.98 | 2 | $3.19 * 1e - 4$ | 70.42% |
| 50 | 5 | 98.04% | 51 | 255 | 2.85 | 2 | $4.39 * 1e - 4$ | 70.42% |
| 50 | 20 | 98.04% | 51 | 1100 | 3.01 | 2 | $1.4 * 1e - 3$ | 46.00% |
| 50 | 100 | 98.04% | 51 | 5100 | 9.00 | 2 | $3.4 * 1e - 3$ | 19.00% |

**Table 3.2 : Evolution des paramètres et des indices durant l'analyse**

Pour les exemples précédents, la comparaison de la charge informatique peut être résumée par les deux graphiques de la Figure 3.28, donnant le gain par itération et le temps relatif par appel.

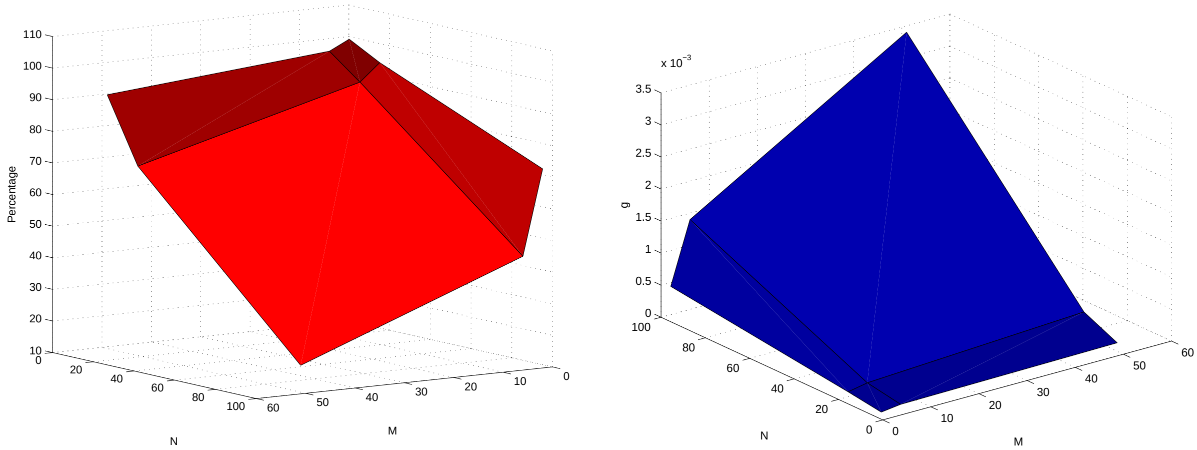



**Figure 3.28 : a) Gain moyen par itération $\bar{g}$ b) Temps de calcul relatif par optimisation $T_{nr}/T_r$**

**Autres cas**

Les résultats fournis ci-dessous ont été obtenus en modifiant la fonction de coût et les conditions initiales de la simulation du système bouclé par la loi MPC, ce qui permet de mettre en évidence d'autres phénomènes. La Figure 3.29 représente colonne de gauche la charge informatique, colonne de droite le nombre d'itérations pour l'optimisation liée à la loi MPC. Les première et deuxième lignes correspondent à des simulations telles que la fonction de coût possède un minimum global sans contraintes à l'intérieur du domaine faisable, tandis que les deux dernières lignes correspondent à des problèmes avec optimum sans contraintes en dehors de la région faisable. D'autre part, les première et troisième lignes représentent des appels de l'optimisation avec un point initial dans la région faisable et les deux autres à des points initiaux non faisables (donc le solveur doit commencer par retrouver un point faisable). Sur chaque graphique, les coordonnées en $x$ représentent le degré de redondance (de 1 à 6) et les coordonnées en $y$ représentent le nombre de contraintes non redondantes (de 5 à 40).

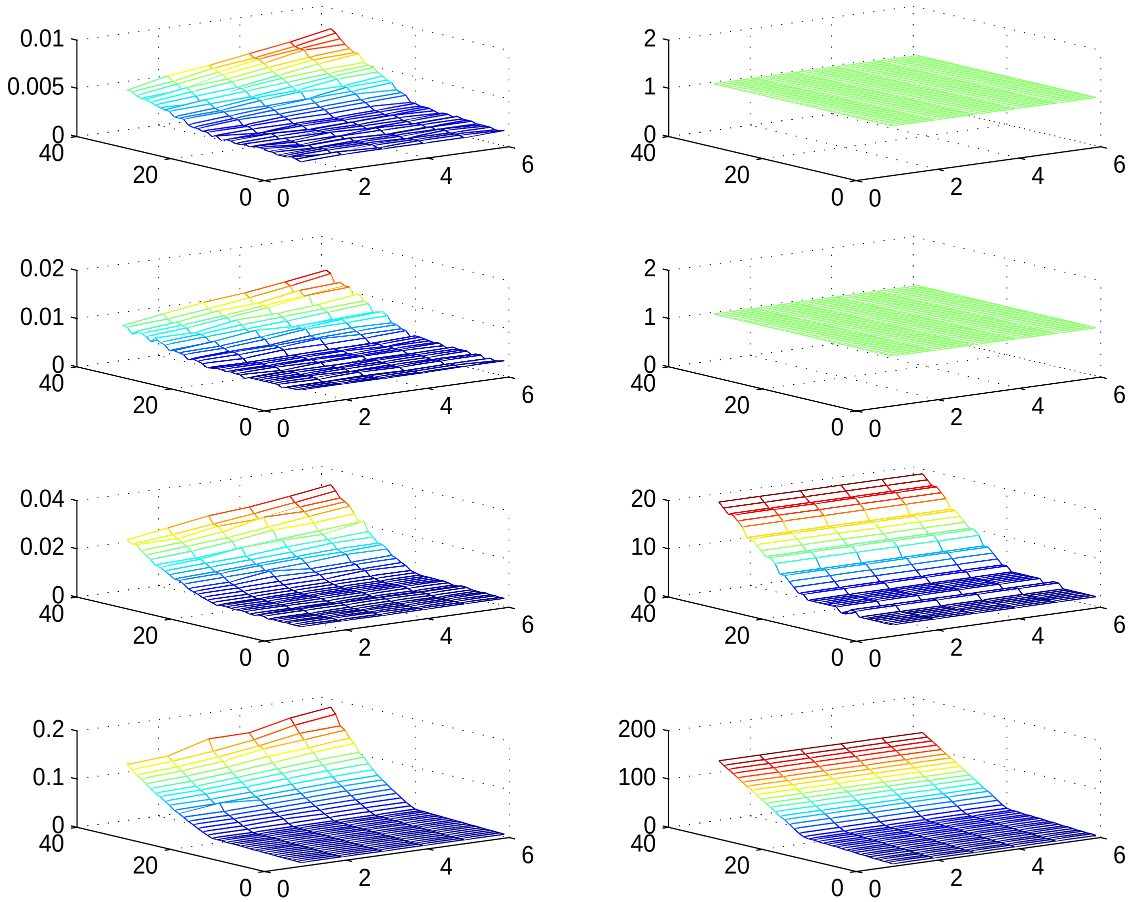



**Figure 3.29 : Autres simulations. A gauche, charge de calcul, à droite, nombres d'itérations. Première ligne, cas de l'optimum à l'intérieur avec point initial à l'intérieur. Deuxième ligne, point de départ à l'extérieur. Troisième ligne, simulation avec optimum sur la frontière et point initial faisable. Quatrième ligne, simulation avec optimum sur la frontière et point initial non faisable.**

**Remarque :** Pendant l'analyse de l'exemple précédent, on constate que le degré de redondance a évolué de façon inversement proportionnelle au nombre de zones de découpage. Généralement, les ensembles de contraintes ne présentent pas une telle propriété et le nombre de régions est indépendant des degrés de redondance pour chaque zone respectivement. Ainsi une étude doit être mise en place avant l'exécution effective en se basant sur le degré moyen de redondance pour le découpage global.

**Remarque :** Généralement les découpages de l'espace de paramètre fournissent des sous-ensembles de contraintes qui diffèrent seulement par une contrainte. A la vue de cette remarque, l'union de zones voisines peut offrir un bon compromis entre un faible degré de redondance et un nombre important de zones.

**Remarque :** Un aspect qui introduit un degré de relativité non négligeable lors de l'analyse 'redondant vs. non redondant' est le fait qu'il n'y a aucune connaissance a priori quant au nombre d'itérations pour les programmes QP. Ainsi, la décision de l'utilisation d'ensembles compacts ou d'ensembles non-redondants par morceaux doit être prise uniquement si les gains informatiques existent même quand l'optimum est à l'intérieur de la région faisable.

Pour résumer cette partie d'analyse, une amélioration du temps de calcul est possible par des techniques basées sur des ensembles non redondants par morceaux en relation avec des programmes QP en ligne. On distingue trois catégories de problèmes liés à la commande prédictive :

1) si le nombre de zones dans l'espace des paramètres correspondant à des régions régulières est très important, l'utilisation des contraintes non redondantes est dissuasive car le temps utilisé par le mécanisme de positionnement ne sera pas compensé par le gain lors des phases d'appel aux programmes d'optimisation.
2) dans le cas où le degré de redondance ou la complexité des ensembles non redondants localement est faible, la charge de calcul n'est pas significativement réduite.
3) si le nombre de régions correspondant à des zones régulières est raisonnable, impliquant un mécanisme de positionnement rapide et si le degré de redondance est significatif, alors la charge de calcul se trouve réduite. Cette amélioration dépend ensuite du nombre de variables de décision, de la méthode choisie et du nombre d'itérations effectuées.

## *3.6 Conclusions*

Ce chapitre a présenté une analyse géométrique du domaine faisable pour les programmes d'optimisation liés à la commande prédictive. Les contraintes considérées étant linéaires, et avec la paramétrisation de ces contraintes, les domaines faisables sont représentés par des polyèdres paramétrés. Ces objets ont été étudiés dans leur double représentation des générateurs, en mettant clairement en évidence le rôle des sommets paramétrés et de leurs domaines de validité.

Concernant la topologie des polyèdres paramétrés, il est important de souligner que les seuls générateurs affectés par la paramétrisation sont les sommets. De plus, avec la variation des paramètres, les sommets paramétrés peuvent se déplacer, fusionner ou se scinder, faisant apparaître d'autres sommets avec une dépendance linéaire en le vecteur des paramètres. En conséquence, des régions de l'espace des paramètres sont mises en évidence, pour lesquelles l'ensemble de générateurs reste inchangé, correspondant à des ensembles bien identifiés de contraintes non redondantes.

Ces observations ont permis d'élaborer une nouvelle structure d'implantation de la commande prédictive, délocalisant une partie de la charge de calcul en ligne vers une procédure de découpage hors-ligne correspondant à la recherche d'ensembles de contraintes localement non redondantes. Les avantages d'une telle approche apparaissent clairement pour des jeux de contraintes ayant un degré de redondance important.

Globalement, ces techniques montrent que la charge de calcul en ligne peut se trouver réduite par la prise en compte du comportement du système bouclé dans des régions spécifiques de l'espace des paramètres de contexte. Le chapitre suivant poursuivra ce processus de découpage de l'espace des paramètres, pour aboutir à des partitions permettant d'implémenter un algorithme de commande prédictive sans faire appel à des programmes d'optimisation en ligne, donc en utilisant une formulation explicite de la séquence optimale.

# 4. Solution explicite

L'analyse des algorithmes prédictifs montre que le domaine faisable se déduit de la spécification d'un ensemble de contraintes linéaires, constituant en fait des polyèdres paramétrés pour lesquels des formes régulières peuvent être identifiées à l'aide de la représentation duale. Or, comme on l'a vu, ce type de manipulation donnant accès à une multitude de problèmes (plus simples) d'optimisation par morceaux, se pose naturellement la question de savoir dans quelle mesure il est possible de raffiner le découpage de l'espace des paramètres de sorte que les optimisations associées soient encore plus simples. La simplicité maximale est bien sûr atteinte pour des problèmes d'optimisation avec contraintes égalité, pour lesquels la solution analytique complète est connue. De façon générale, le but ultime pour une simplicité la plus grande possible consiste à élaborer pour chaque partition une fonction simple à évaluer, de sorte que le recours aux solveurs en ligne pour les problèmes d'optimisations ne soit plus nécessaire. Cette alternative extrême est en fait très intéressante, ainsi les avantages liés à la mise en œuvre d'une solution explicite pour les domaines non paramétrés sont fournis par Seron, *et al.* [SGD02]. Cette approche peut dès lors être revisitée en tenant compte de la double représentation des domaine faisables (chapitre 3) et généralisée pour les domaines réguliers en se basant principalement sur les domaines de validité des sommets paramétrés. Ensuite par un regroupement de toutes les régions de l'espace des paramètres, un découpage décrivant ce que l'on appelle la solution explicite du problème d'optimisation multiparamétrique de départ est mis en évidence, et par conséquent une formulation explicite des lois de commande prédictive.

La construction de solutions explicites pour les problèmes d'optimisation multiparamétrique constitue un sujet de recherche très actif, avec des résultats couvrant une multitude d'approches et de systèmes, pouvant se regrouper en deux familles importantes. Une première se base sur les multiplicateurs de Lagrange et leur domaine de validité [BMDP02], [BOR03], [TJB01], avec un avantage non négligeable lié au fait que l'exploration se fait directement dans l'espace des paramètres, la description explicite est alors construite efficacement. Des améliorations peuvent être apportées si une analyse d'atteignabilité est prise en compte lors de la construction. La deuxième catégorie de méthodes inclut l'approche géométrique développée par Seron *et al.* [SGD], qui travaille dans l'espace des arguments du problème d'optimisation pour inférer des découpages dans l'espace des paramètres. L'avantage de cette technique est la classification des contraintes, le désavantage étant lié à l'énumération des combinaisons possibles, pouvant s'avérer gourmande en temps de calcul. L'objectif des développements à venir, qui feront la spécificité et l'originalité de la méthodologie proposée, est de montrer que ces méthodes peuvent être combinées avec la double représentation des domaines faisables pour obtenir une vision plus détaillée de la topologie de la solution optimale et exprimer explicitement sa dépendance en fonction des paramètres. Plus spécifiquement, les descriptions explicites de la structure d'une séquence optimale de commande seront recherchées comme une projection d'optimums sans contraintes sur des polyèdres paramétrés. Pour être complet sur ce sujet, mentionnons des travaux récents et intéressants pour la construction des solutions explicites par des techniques de programmation dynamique. L'avantage de cette idée est qu'elle peut être utilisée combinée avec les méthodes déjà mentionnées précédemment, voir [MD04], [PABC04]. Enfin, les formulations explicites des problèmes multiparamétriques ne se restreignent pas au cas discret [14], on trouve en effet des solutions explicites multiparamétriques dans le cadre d'optimisations avec variables continues.

Ce chapitre présente donc la construction géométrique de la solution explicite basée sur les polyèdres paramétrés, pour la commande prédictive avec modèles d'état, mais également une description de la forme explicite des lois GPC. Même si cette deuxième formulation peut être vue pour les systèmes SISO comme un cas particulier de la première, leur analyse n'est pas anodine car dès lors que ces lois GPC sous contraintes se basent aussi sur une formulation RST, il devient possible d'appliquer les méthodes d'analyse spécifiques développées dans la cas d'une loi de commande prédictive polynomiale sans contrainte. Ce chapitre s'oriente également vers la construction de lois explicites à partir d'un pré-découpage de l'espace d'état en régions de contraintes non redondantes.

Ce chapitre détaille les résultats originaux contenus dans [OD04d], [OD05d], [OD05a].

## *4.1 Introduction*

Avant de poursuivre les développements théoriques, il est nécessaire à ce stade d'introduire un certain nombre de définitions permettant de clarifier l'objectif de la démarche proposée. Ainsi, tentons de définir le concept de formulation explicite d'une loi de commande, qui, dans le cas prédictif, est directement lié à la formulation explicite du problème d'optimisation multiparamétrique associé.

[14] La commande prédictive proposée a été décrite en temps discret et les problèmes d'optimisation multiparamétrique associée héritaient de cette caractéristique.

**Définition :** La solution numérique des problèmes d'optimisation est dite soit *explicite* soit *implicite*. Quand les variables dépendantes sont définies par des ensembles couplés d'équations, par des matrices de forme variable ou quand des techniques itératives sont nécessaires pour obtenir la solution, la méthode numérique est implicite. En revanche, quand un calcul direct des variables dépendantes peut être fait mettant en évidence des quantités connues, le calcul est alors explicite.

Dans le cas d'un problème de programmation quadratique multiparamétrique :

$$\begin{aligned}&\min_{\mathbf{k}_u} J_t = 0{,}5\,\mathbf{k}_u^T H\,\mathbf{k}_u + x_t^T F\,\mathbf{k}_u\\ &\text{sous les contraintes} : \begin{cases} A_{in}\mathbf{k}_u \le B_{in}x_t + b_{in}\\ A_{eq}\mathbf{k}_u = B_{eq}x_t + b_{eq}\end{cases}\end{aligned} \tag{4.1}$$

on peut écrire la solution sous forme compacte :

$$\begin{aligned}\mathbf{k}_u^*(x_t) = \Big[&(H^{-1}\overline{A}^T(\overline{A}H^{-1}\overline{A}^T)\overline{A}H^{-1} - H^{-1})F + H^{-1}\overline{A}^T(\overline{A}H^{-1}\overline{A}^T)^{-1}\overline{B}\Big]x_t +\\ &+ H^{-1}\overline{A}^T(\overline{A}H^{-1}\overline{A}^T)^{-1}\overline{b}\end{aligned} \tag{4.2}$$

avec $\overline{A}$ matrice (de rang plein) formée par $A_{eq}$ et un sous-ensemble de $A_{in}$ correspondant aux contraintes inégalité saturées par la valeur optimale, et $\overline{B},\overline{b}$ définis de façon similaire pour la partie droite des contraintes. Cette description de la valeur optimale a l'avantage de ne faire intervenir que des quantités connues ($x$ – le vecteur de paramètres de contexte – est supposé connu à chaque pas d'échantillonnage) mais elle correspond toujours à une formulation implicite car les matrices ont des structures variables en fonction des contraintes saturées et ne sont donc pas utilisables pour un calcul direct de la solution. En revanche, le fait qu'il n'existe pas une forme prédéfinie des matrices $\overline{A}$, $\overline{B},\overline{b}$ permet deux déductions importantes :

- la représentation explicite du problème n'est pas une fonction globalement linéaire en le vecteur des paramètres de contexte,
- cet aspect signale la présence de termes affines au sein de la solution explicite.

La forme explicite d'une loi de commande prédictive sous contraintes implique l'application selon le principe de l'horizon fuyant de la solution explicite du problème quadratique multiparamétrique (4.1) associé. Concrètement elle est représentée par une fonction non-linéaire dans le vecteur des paramètres de contexte $x$ :

$$\begin{aligned}&u_t^{GPC} = f(x_t)\\ &f(.) : \Re^n \to \Re^m\end{aligned} \tag{4.3}$$

avec pour $p = m \cdot N_u$ et $N_u$ l'horizon de commande :

$$\begin{aligned}&f(.) = \Big[\underbrace{\mathbf{1}\ 0\ \cdots\ 0}_{N_u \text{ termes}}\Big]\mathbf{k}_u(.);\ \ \mathbf{1} = \Big[\underbrace{1\ 1\ \cdots\ 1}_{m}\Big]\\ &\mathbf{k}_u(.) : \Re^n \to \Re^p\end{aligned} \tag{4.4}$$

Le schéma de fonctionnement est donné Figure 4.1, sur laquelle on observe la migration de l'effort de calcul en ligne vers une pré-procédure hors ligne. La loi effective à appliquer en ligne nécessite uniquement une évaluation de la fonction.

## *4.2 Construction de la solution explicite. Approche par polyèdres paramétrés*

En formulant les conditions géométriques d'optimalité (Annexe 1), on observe que le problème (4.1) est caractérisé par deux éléments constitutifs : une fonction de coût et un domaine faisable, les deux affectés par la paramétrisation. A chaque pas d'échantillonnage, il résulte un domaine faisable polyédral qui est par conséquent convexe. Il existe alors une propriété utile pour la caractérisation des points extérieurs aux domaines faisables convexes, résumée par le théorème suivant.

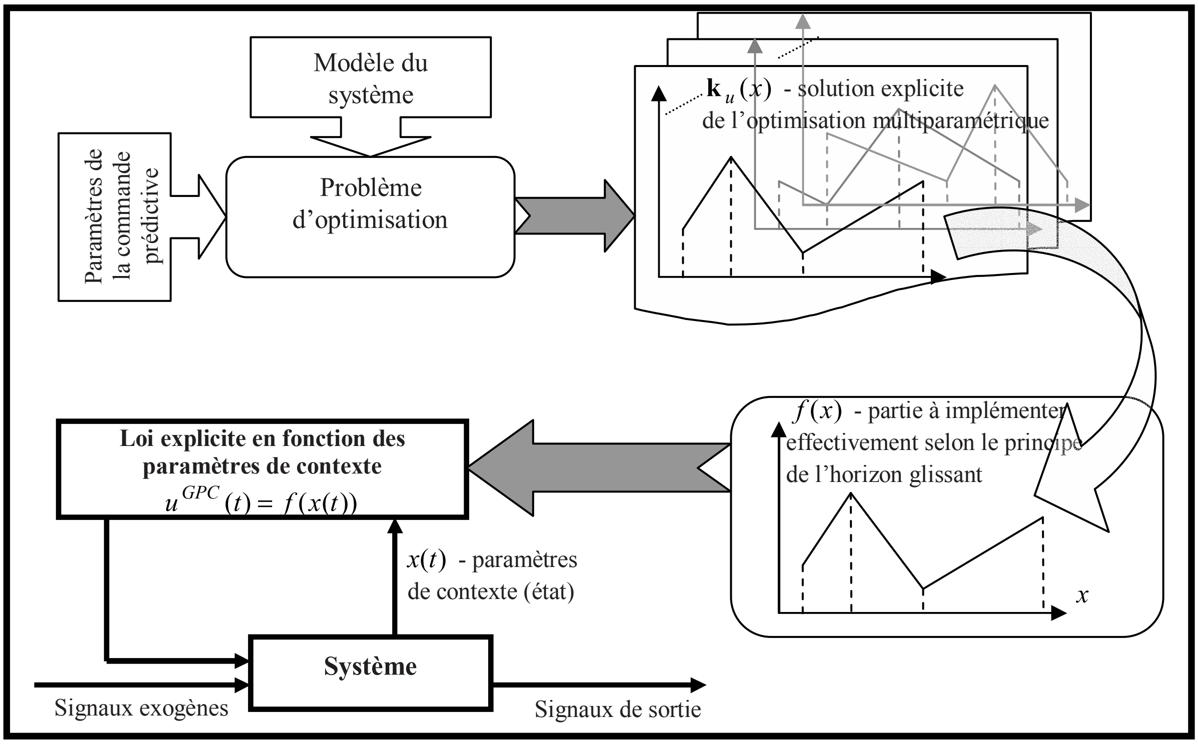



**Figure 4.1 : Schéma de fonctionnement d'une application basée sur une forme explicite de la loi prédictive.**

**Théorème 4.1**[15] **:** Soit un point extérieur à un domaine polyédral $P$ – $\mathbf{k}_u^{ext} \notin P$. Alors pour une matrice positive définie $H$, il existe un point unique $\mathbf{k}_u^* \in P$ qui satisfait :

$$(\mathbf{k}_u^{ext} - \mathbf{k}_u^*)^T H(\mathbf{k}_u - \mathbf{k}_u^*) \leq 0, \quad \forall \mathbf{k}_u \in P \tag{4.5}$$

■

Ce résultat s'avère utile pour garantir l'unicité de la solution optimale obtenue en se rapportant à la position de l'optimum sans contraintes du problème (4.1).

### 4.2.1 Solution explicite d'un problème quadratique avec domaine faisable non paramétré

Dans le cadre du problème d'optimisation multiparamétrique (4.1), considérons tout d'abord le cas où le domaine faisable n'est pas paramétré :

$$\begin{aligned} &\min_{\mathbf{k}_u} J_t = 0{,}5\,\mathbf{k}_u^T H\,\mathbf{k}_u + x_t^T F\,\mathbf{k}_u \\ &\text{sous les contraintes} : \begin{cases} A_{in}\mathbf{k}_u \leq b_{in} \\ A_{eq}\mathbf{k}_u = b_{eq} \end{cases} \end{aligned} \tag{4.6}$$

La représentation du critère par une fonction quadratique implique que l'on puisse déterminer le point optimal en ignorant les contraintes :

$$\mathbf{k}_u^{sc}(x) = H^{-1} F^T x \tag{4.7}$$

Bien sûr, ce point évolue de façon linéaire en fonction du vecteur de paramètres et le lieu géométrique des points :

$$\begin{bmatrix} \mathbf{k}_u^{sc} \\ x \end{bmatrix} \in \Re^{p+n}$$

dans l'espace étendu (arguments + paramètres) est représenté par un polyèdre :

[15] Ce théorème est une généralisation du résultat donné dans [BSS93], pp. 43. La preuve de ce théorème s'obtient de façon similaire.

$$P = \left\{ \begin{bmatrix} \mathbf{k}_u^{sc} \\ x \end{bmatrix} \in \Re^{p+n} \middle| \begin{bmatrix} I & -H^{-1}F^T \end{bmatrix} \begin{bmatrix} \mathbf{k}_u^{sc} \\ x \end{bmatrix} = 0 \right\} \tag{4.8}$$

Comme conséquence du théorème 4.1, étant donné le domaine polyédral $P$ donné par l'ensemble de contraintes, si le point $\mathbf{k}_u^{sc}(x)$ n'appartient pas à $P$, il existe un point unique $\mathbf{k}_u^* \in P$ qui satisfait (4.5) et donc représente la solution optimale pour (4.1). Géométriquement, ce point doit être cherché sur la frontière de $P$ à l'intersection avec les courbes isocoût du critère, matérialisées par des hyperellipses (Figure 4.2a). La matrice $H$ étant positive définie, le changement de variable suivant peut être introduit :

$$\widetilde{\mathbf{k}}_u = H^{1/2}\mathbf{k}_u \tag{4.9}$$

Une telle transformation implique la modification de la forme du domaine faisable. De même, les hyperellipses se transforment en hypersphères (Figure 4.2b) centrées cette fois sur le point :

$$\widetilde{\mathbf{k}}_u^{sc}(x) = H^{-1/2}F^T x \tag{4.10}$$

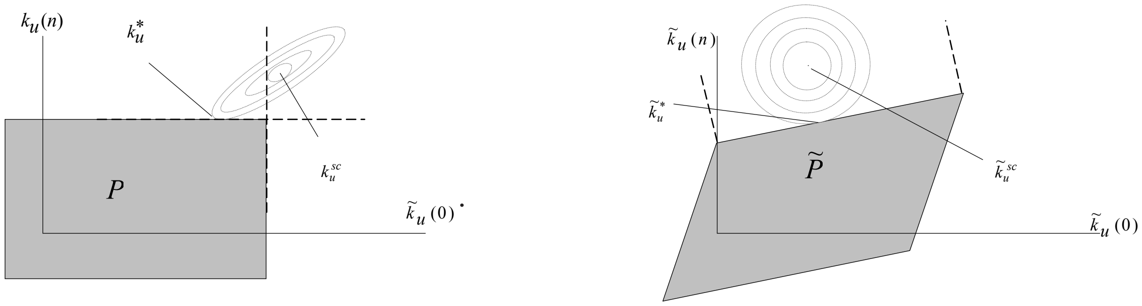



**Figure 4.2 : Géométrie du problème d'optimisation (4.1)**
**a) Avant changement de variable (4.1) b) Après changement de variable (4.8)**

Dès lors, le problème de recherche du point optimal se transforme en un problème d'évaluation d'une distance euclidienne minimale satisfaisant les conditions d'optimalité en vertu du corollaire 3.2.

**Corollaire 4.2 :** Pour un point extérieur $\mathbf{k}_u^{ext} \notin P$, il existe un point unique $\mathbf{k}_u^* \in P$ caractérisé par une distance minimale par rapport à $\mathbf{k}_u^{ext}$. Le point $\mathbf{k}_u^*$ satisfait :

$$(\mathbf{k}_u^{ext} - \mathbf{k}_u^*)^T(\mathbf{k}_u - \mathbf{k}_u^*) \le 0, \forall \mathbf{k}_u \in P \tag{4.11}$$

■

O peut alors construire de façon géométrique la solution analytique globale de :

$$\begin{aligned} &\min_{\widetilde{\mathbf{k}}_u} J_t = 0{,}5\,\widetilde{\mathbf{k}}_u^T\,\widetilde{\mathbf{k}}_u + x_t^T\widetilde{F}\widetilde{\mathbf{k}}_u \\ &\text{sous les contraintes :} \begin{cases} \widetilde{A}_{in}\widetilde{\mathbf{k}}_u \le b_{in} \\ \widetilde{A}_{eq}\widetilde{\mathbf{k}}_u = b_{eq} \end{cases} \end{aligned} \tag{4.12}$$

avec $\widetilde{\mathbf{k}}_u$ défini comme dans (4.7), $\widetilde{F} = FH^{-1/2}$ et $\widetilde{A}_{in/eq} = A_{in/eq}H^{-1/2}$. Cela passe par un partitionnement de l'espace $\Re^p$ en régions délimitées par des normales aux faces du polyèdre $\widetilde{P}$. Une première région peut être identifiée comme étant le domaine faisable, $R_0 = \widetilde{P}$ **,** car on sait que l'optimum sans contraintes y coïncide avec celui sous contraintes :

$$\widetilde{\mathbf{k}}_u^*(x) = \widetilde{\mathbf{k}}_u^{sc}(x) = H^{-1/2}F^T x \quad \text{si} \quad \widetilde{\mathbf{k}}_u^{sc}(x) \in R_0 = \widetilde{P} \tag{4.13}$$

Ensuite la délimitation de la région $R_1$ est décrite en fonction de la face $f_1$ : $a_{in\,i}\widetilde{\mathbf{k}}_u = b_{in\,i}$ et les lignes normales à cette face : $\pi_1 : a_{\pi_1}\widetilde{\mathbf{k}}_u = b_{\pi_1}$ et $\pi_2 : a_{\pi_2}\widetilde{\mathbf{k}}_u = b_{\pi_2}$, soit :

$$R_1 : \begin{cases} a_{in\,i}\widetilde{\mathbf{k}}_u = b_{in\,i} \\ a_{\pi_1}\widetilde{\mathbf{k}}_u = b_{\pi_1} \\ a_{\pi_2}\widetilde{\mathbf{k}}_u = b_{\pi_2} \end{cases} \tag{4.14}$$

et de façon similaire pour toutes les régions $R_2$, $R_3$, $R_4$ (Figure 4.3). La solution associée sera la projection orthogonale sur la face considérée. La procédure est poursuivie de façon similaire jusqu'à arriver aux régions pour lesquelles les solutions optimales sous contraintes se trouvent aux sommets du domaine faisable. Par exemple, Figure 4.3b, les points contenus dans la région $R_5$ (délimitée par les normales aux contraintes saturées $\pi_1$ et $\pi_3$) se projettent par la projection euclidienne sur $\widetilde{P}$ en le sommet $s_1$. On obtient finalement un partitionnement en $n_R$ régions :

$$R_0 \cup R_1 \cup \cdots \cup R_{n_R} = \Re^p \tag{4.15}$$

avec $n_R$ donné par le nombre de combinaisons possibles des contraintes actives.

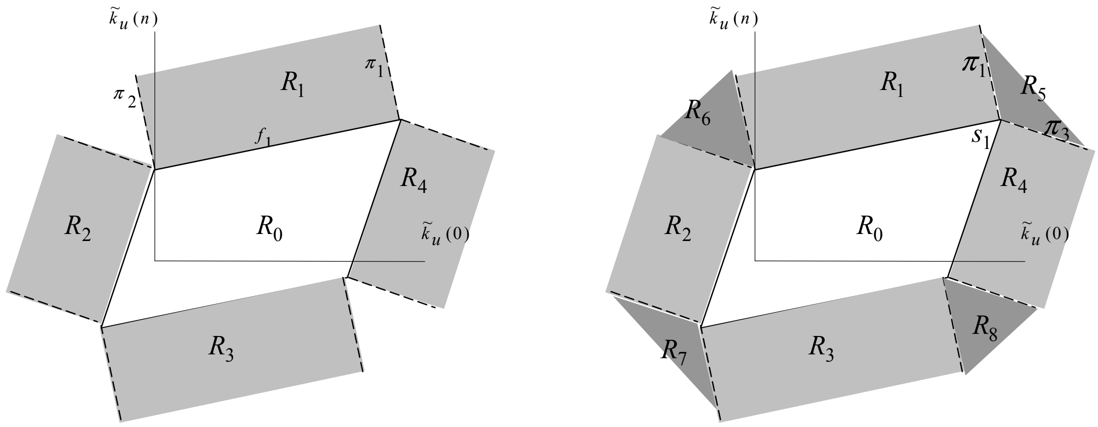



**Figure 4.3 : Partitionnement de l'espace $\Re^p$ selon l'approche [SDG]**

Alors la solution explicite du problème d'optimisation (4.12) est donc :

$$\begin{cases} \widetilde{\mathbf{k}}_u^*(x) = \widetilde{U}_1 \widetilde{\mathbf{k}}_u^{sc}(x) + \widetilde{v}_1 & si \quad \widetilde{\mathbf{k}}_u^{sc}(x) \in \widetilde{R}_1 \\ \cdots \\ \widetilde{\mathbf{k}}_u^*(x) = \widetilde{U}_{n_R} \widetilde{\mathbf{k}}_u^{sc}(x) + \widetilde{v}_{n_R} & si \quad \widetilde{\mathbf{k}}_u^{sc}(x) \in \widetilde{R}_{n_R} \end{cases} \underset{\text{inverse à (4.9)}}{\overset{\text{par transformation}}{\Leftrightarrow}} \begin{cases} \mathbf{k}_u^*(x) = U_1 x + v_1 & si \quad x \in R_1 \\ \cdots \\ \mathbf{k}_u^*(x) = U_{n_R} x + v_{n_R} & si \quad x \in R_{n_R} \end{cases} \tag{4.16}$$

Introduisons maintenant une méthode conservant la même philosophie de recherche de projections euclidiennes, mais utilisant la représentation duale des polyèdres (définie lors du chapitre précédent) pour construire la formulation explicite. Le domaine faisable pour le problème considéré (4.6) :

$$P = \left\{ \mathbf{k}_u \;\middle|\; A_{in} \mathbf{k}_u \le b_{in} \; ; A_{eq} \mathbf{k}_u = b_{eq} \right\} \tag{4.17}$$

est indépendant de l'évolution des paramètres. Cette première observation nous aide dans le développement de la description duale de l'ensemble des combinaisons des paramètres et des séquences de commande faisable :

$$D = \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \middle| \mathbf{k}_u \in P \right\} \tag{4.18}$$

En effet, si l'on détient la représentation duale de (4.17) :

$$P = \left\{ \mathbf{k}_u \middle| \mathbf{k}_u = \sum_{i=1}^{d_S} \sigma_i s_i + \sum_{i=1}^{d_R} \rho_i r_i + \sum_{i=1}^{d_L} \lambda_i l_i \right\}, \qquad 0 \le \sigma_i \le 1, \; \sum_{i=1}^{v} \sigma_i = 1 \, , \rho_i \ge 0 \; , \forall \lambda_i \tag{4.19}$$

alors la représentation duale de (4.18) est immédiate :

$$D=\left\{\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\middle|\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}=\sum_{i=1}^{d_S}\sigma_i\begin{bmatrix}s_i\\ 0\end{bmatrix}+\sum_{i=1}^{d_R}\rho_i\begin{bmatrix}r_i\\ 0\end{bmatrix}+\sum_{i=1}^{d_L}\lambda_i\begin{bmatrix}l_i\\ 0\end{bmatrix}+\sum_{i=d_{L+1}}^{d_{L+n}}\lambda_i\begin{bmatrix}0\\ l_i\end{bmatrix}\right\},$$
$$0\le\sigma_i\le 1,\ \sum_{i=1}^{v}\sigma_i=1,\ \rho_i\ge 0\ ,\forall\lambda_i \tag{4.20}$$

avec $n$ dimension de l'espace des paramètres et $l_i, i=d_{L+1},...,d_{L+n}$ un ensemble de générateurs de ce sous-espace. En résumé, on a simplement :

$$D=\left\{\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\middle|\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}=\sum_{i=1}^{d_S}\sigma_i\begin{bmatrix}s_i\\ 0\end{bmatrix}+\sum_{i=1}^{d_R}\rho_i\begin{bmatrix}r_i\\ 0\end{bmatrix}+\sum_{i=1}^{d_{L+n}}\lambda_i\begin{bmatrix}l_i\\ 0\end{bmatrix}\right\},$$
$$0\le\sigma_i\le 1,\ \sum_{i=1}^{v}\sigma_i=1,\ \rho_i\ge 0\ ,\forall\lambda_i \tag{4.21}$$

**Remarque :** Le fait que l'ensemble de générateurs soit complété par des rayons bidirectionnels pour le sous-espace de paramètres confirme le fait que le problème d'optimisation est faisable.

Ensuite comme mentionné auparavant la solution de (4.6) se déduit en construisant l'hypersurface :

$$S\widetilde{D}=\left\{\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\in D\ \middle|\ t.q.\ \forall\begin{bmatrix}\kappa\\ x\end{bmatrix}\in D,\left(\begin{bmatrix}H^{-1}F^Tx\\ x\end{bmatrix}-\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\right)^T H\left(\begin{bmatrix}\kappa\\ x\end{bmatrix}-\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\right)\le 0\right\} \tag{4.22}$$

On retrouve ainsi la présence de la paramétrisation dans l'expression de la solution, due à son influence sur l'optimum sans contraintes. Du point de vue programmation, la construction (4.22) est difficile à réaliser géométriquement car le lieu géométrique des points optimaux dans le cas sans contraintes se trouve sur des courbes isocout ellipsoïdales (Figure 4.4).

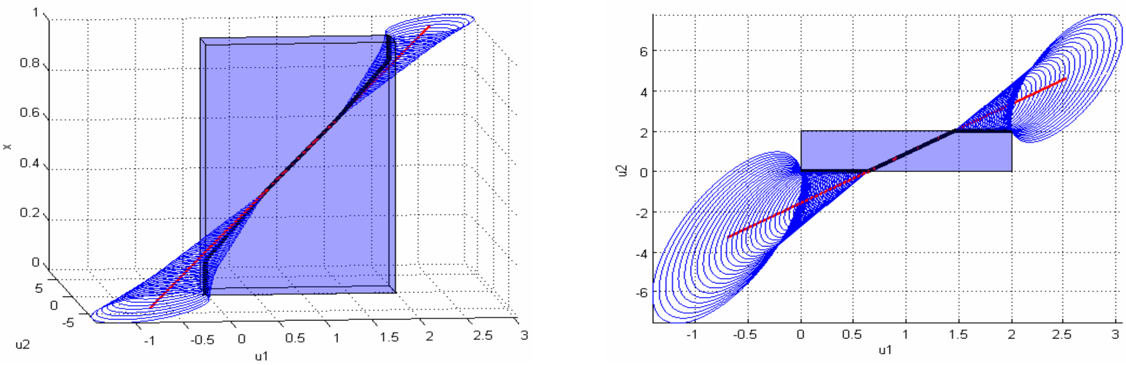



**Figure 4.4 : Interprétation des zones de projection constante du point de vue paramétrisation. Isocout de type ellipsoide.**

Pour contourner cette difficulté, il est possible d'appliquer la stratégie de transformation précédente, soit, en se basant sur une transformation similaire à (4.9) le domaine :

$$\widetilde{D}=\left\{\begin{bmatrix}\widetilde{\mathbf{k}}_u\\ x\end{bmatrix}\in D\ \middle|\ \begin{bmatrix}\widetilde{\mathbf{k}}_u\\ x\end{bmatrix}=\begin{bmatrix}H^{1/2}&0\\ 0&I\end{bmatrix}\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\right\} \tag{4.23}$$

se ramenant ainsi à une logique de construction basée sur des projections euclidiennes pour :

$$S\widetilde{D}=\left\{\begin{bmatrix}\widetilde{\mathbf{k}}_u\\ x\end{bmatrix}\in \widetilde{D}\ \middle|\ t.q.\ \forall\begin{bmatrix}\kappa\\ x\end{bmatrix}\in \widetilde{D},\left(\begin{bmatrix}H^{-1/2}F^Tx\\ x\end{bmatrix}-\begin{bmatrix}\widetilde{\mathbf{k}}_u\\ x\end{bmatrix}\right)^T\left(\begin{bmatrix}\kappa\\ x\end{bmatrix}-\begin{bmatrix}\widetilde{\mathbf{k}}_u\\ x\end{bmatrix}\right)\le 0\right\} \tag{4.24}$$

L'espace autour du domaine faisable doit être maintenant partitionné en zones induisant une projection (selon des courbes isocout sphériques) du lieu géométrique de l'optimum sans contraintes (Figure 4.5) définissant des ensembles convexes.

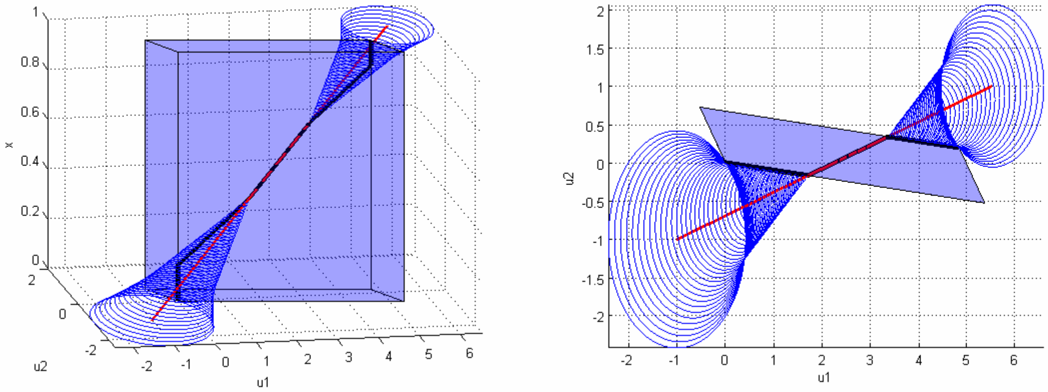



**Figure 4.5 : Interprétation des zones à projection constante du point de vue paramétrisation. Isocout de type sphérique**

Sachant que les points de $S\widetilde{D}$ (4.24) peuvent suivre les différentes facettes du domaine $\widetilde{D}$, il résulte que $S\widetilde{D}$ se caractérise dans le cas général par une union d'ensembles convexes, ce que l'on peut constater Figure 4.5.

**Elimination de la paramétrisation au sein de la fonction de coût**

Avant de décrire les étapes de construction de l'ensemble $S\widetilde{D}$, on va montrer que la disposition du polyèdre $\widetilde{D}$ et du lieu géométrique de l'optimum $R$ peut être ramenée à la caractérisation du rapport entre une droite passant par l'origine et le polyèdre issu d'une transformation isomorphe. En appliquant la transformation :

$$\begin{bmatrix}\hat{\mathbf{k}}_u \\ x\end{bmatrix}=\begin{bmatrix}\widetilde{\mathbf{k}}_u \\ x\end{bmatrix}-\begin{bmatrix}H^{-1/2}F^T x \\ x\end{bmatrix}=\begin{bmatrix}I & -H^{-1/2}F^T \\ 0 & I\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{k}}_u \\ x\end{bmatrix}\Rightarrow \hat{D}=\left\{\begin{bmatrix}\hat{\mathbf{k}}_u \\ x\end{bmatrix}\middle|\begin{bmatrix}\hat{\mathbf{k}}_u \\ x\end{bmatrix}=\begin{bmatrix}I & -H^{-1/2}F^T \\ 0 & I\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{k}}_u \\ x\end{bmatrix},\begin{bmatrix}\widetilde{\mathbf{k}}_u \\ x\end{bmatrix}\in\widetilde{D}\right\} \quad (4.25)$$

l'optimum sans contraintes est en effet ramené à l'origine et l'ensemble décrivant l'optimum sous contraintes transformé en la forme suivante :

$$S\hat{D}=\left\{\begin{bmatrix}\hat{\mathbf{k}}_u \\ x\end{bmatrix}\in\hat{D}\ \middle|\ t.q.\ \forall\begin{bmatrix}\kappa \\ x\end{bmatrix}\in\hat{D},\left(\begin{bmatrix}0 \\ x\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{k}}_u \\ x\end{bmatrix}\right)^T\left(\begin{bmatrix}\kappa \\ x\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{k}}_u \\ x\end{bmatrix}\right)\leq 0\right\} \quad (4.26)$$

La transformation étant réversible, on sera en mesure de reconstituer la solution du problème original (Figure 4.6).

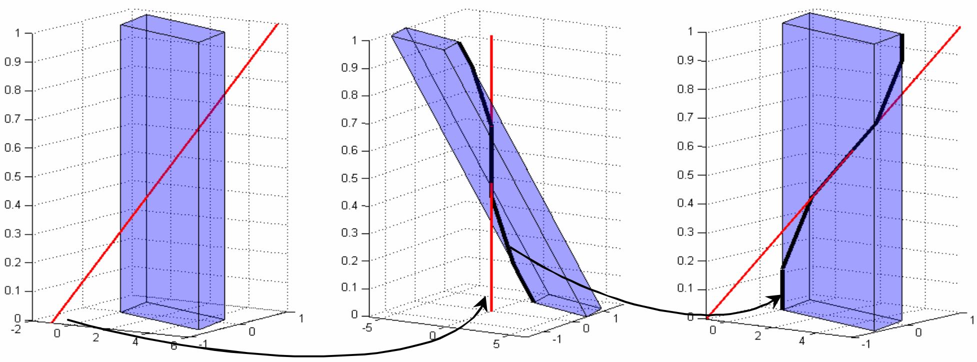


**Figure 4.6 : Elimination de la paramétrisation de l'optimum sans contraintes**

Il est intéressant d'observer qu'en regroupant les transformations (4.23) et (4.25) on obtient :

$$\hat{\mathbf{k}}_u = H^{1/2}\mathbf{k}_u - H^{-1/2}F^T x \quad (4.27)$$

Ceci signifie qu'on opère en fait une élimination de la paramétrisation de la fonction de coût, et la migration de celle-ci vers le domaine faisable par l'intermédiaire des contraintes. En effet la fonction de coût devient :

$$\min_{\hat{\mathbf{k}}_u} J_t = 0{,}5\,\hat{\mathbf{k}}_u^T\ \hat{\mathbf{k}}_u$$
$$\text{sous les contraintes}: \begin{cases} \hat{A}_{in}\hat{\mathbf{k}}_u \le b_{in} + \hat{B}_{in}x \\ \hat{A}_{eq}\hat{\mathbf{k}}_u = b_{eq} + \hat{B}_{eq}x \end{cases} \tag{4.28}$$

où $\hat{A}_{in} = A_{in}H^{-1/2}; \hat{B}_{in} = A_{in}H^{-1}F^T; \hat{A}_{eq} = A_{eq}H^{-1/2}; \hat{B}_{eq} = A_{eq}H^{-1}F^T$ .

**Partition**

Pour construire effectivement $S\hat{D}$ , il faut partitionner l'espace étendu $\Re^{p+n}$ (arguments $\hat{\mathbf{k}}_u \in \Re^p$ + paramètres $x \in \Re^n$) de telle sorte que la projection du lieu géométrique de l'optimum sans contraintes – $\hat{R}$ (le correspondant de $R$ par la transformation (4.27)) sur le domaine $\hat{D}$ – ait comme résultat un ensemble convexe. Notons que la projection se fait par rapport au sous-espace des paramètres $\Re^n$ . Lors de ce partitionnement, les sommets paramétrés de $\hat{D}$ jouent un rôle déterminant. En effet, on a vu que $\hat{D}$ est issu du polyèdre initial $D$ par une suite de transformations résumées par (4.27). Sous la forme initiale (4.21), les sommets n'étaient pas paramétrés, cet aspect a été induit par la transformation (4.27) :

$$\begin{bmatrix} \hat{s}_i \\ x \end{bmatrix} = \begin{bmatrix} H^{1/2}s_i \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -H^{-1/2}F^T \\ 0 & I \end{bmatrix}\begin{bmatrix} 0 \\ x \end{bmatrix}; i = 1,\dots,n_S \tag{4.29}$$

qui transforme donc les sommets du polyèdre initial $P$ en des sommets paramétrés :

$$\hat{s}_i(x) = H^{1/2}s_i - H^{-1/2}F^T x \tag{4.30}$$

pour :

$$\hat{P}(x) = \left\{\hat{\mathbf{k}}_u \,\middle|\, \hat{A}_{in}\hat{\mathbf{k}}_u \le b_{in} + \hat{B}_{in}x; \hat{A}_{eq}\hat{\mathbf{k}}_u = b_{eq} + \hat{B}_{eq}x\right\} \tag{4.31}$$

Leur domaine de validité est tout l'espace des paramètres et leur nombre reste inchangé par rapport à $D$ .

La procédure de partitionnement considère toutes les combinaisons possibles des contraintes mais ne retient que celles pour lesquelles il existe des sommets paramétrés les satisfaisant. Cette décision n'impose que l'inspection de la matrice de saturation pour le polyèdre paramétré $\hat{P}(x)$ . En résumé, le partitionnement doit passer par une étape d'identification des contraintes actives.

***Procédure Split** ( $\hat{D}$ , sommets paramétrés de $\hat{P}(x)$ )*

Pour $i = 0 \cdots n$
1) Générer toutes les combinaisons de *i* contraintes parmi celles qui décrivent $\hat{D}$
2) Retenir les combinaisons qui sont satisfaites par au moins $n - i + 1$ sommets paramétrés
3) Stocker les combinaisons des contraintes et les sommets paramétrés qui décrivent une région valide

fin

Cette étape d'identification fournit les sommets paramétrés caractérisant les régions et les contraintes correspondantes. Il reste alors à construire effectivement les polyèdres décrits par ces sommets pour disposer du partitionnement. La méthode la plus directe consiste à travailler avec les générateurs, composés des sommets paramétrés et des directions orthogonales aux contraintes saturées. Le résultat est une union de $\overline{n}_D$ ensembles convexes $DP_i$ qui couvre tout l'espace étendu $\Re^{p+n}$ (Figure 4.7) :

$$\Re^{p+n} = \bigcup_{i=1}^{\overline{n}_D} DP_i \tag{4.32}$$

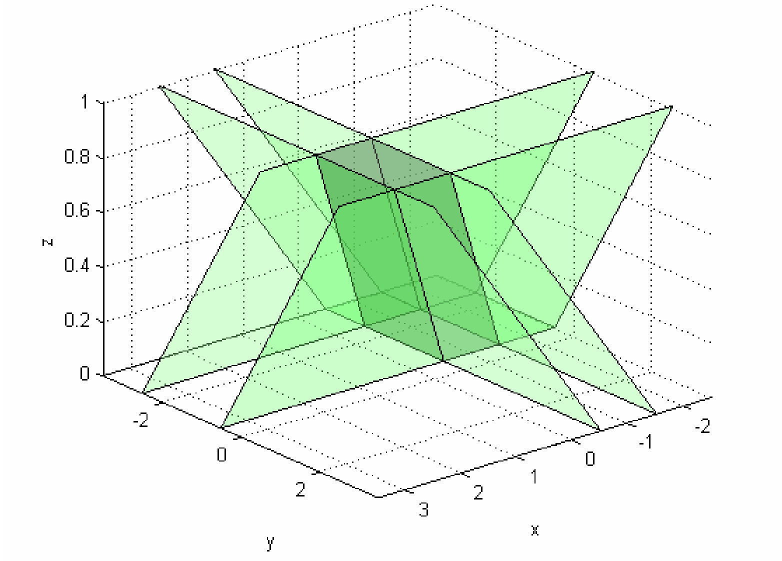



**Figure 4.7 : Exemple de partitionnement**

Parmi toutes ces zones qui assurent une projection convexe sur $\hat{D}$, on s'intéresse lors de la construction de $S\hat{D}$ aux zones $DP_i$ ayant une intersection non vide avec $\hat{R}$ (Figure 4.8a). Une fois ces zones identifiées, l'intersection nous offre aussi l'information nécessaire sur le sous-ensemble des paramètres pour lequel la projection appartient à cette zone (Figure 4.8b).

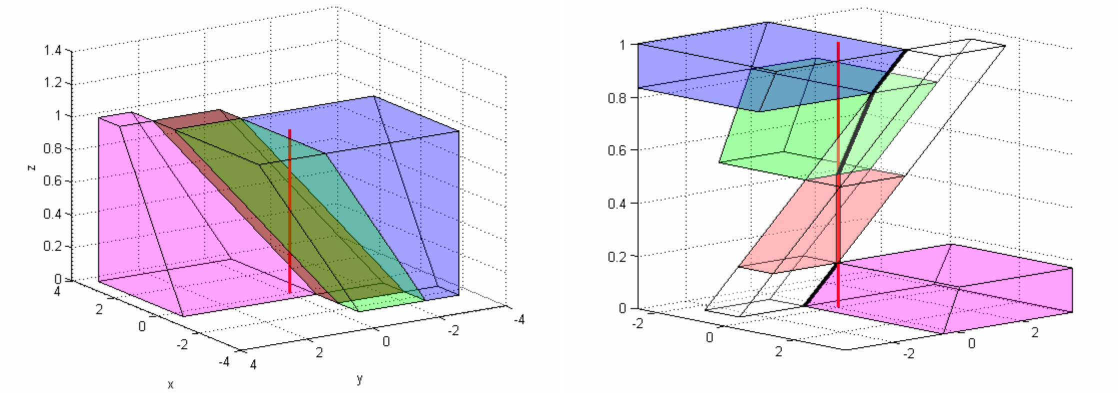



**Figure 4.8 : a) Partitions d'intersection non vide avec $\hat{R}$**
**b) leur limitation au domaine de validité de l'intersection**

Soit $Z_1,...,Z_{n_{\hat{D}}}$ les $n_{\hat{D}}$ zones construites à partir de $DP_i$ ayant une intersection non vide avec $\hat{R}$ et limitées au domaine de validité de cette intersection dans l'espace des paramètres. La solution recherchée pour (4.6) est déduite par application d'une transformation inverse de (4.27) (image du trait noir Figure 4.7b) pour :

$$S\hat{D} = \left(\hat{D} \cap \hat{R}\right) \bigcup_{i=1}^{n_{\hat{D}}} \left(\hat{D} \cap Z_i\right) \tag{4.33}$$

Le résumé des étapes précédentes aboutissant à la construction de cette union fournissant la solution optimale procure un squelette d'algorithme applicable aux problèmes d'optimisation multiparamétriques (4.6).

***<u>Procédure ExplicitPPP</u>***

***Partie A***

1) Construction de la représentation duale du polyèdre $P = \left\{\mathbf{k}_u \mid A_{in}\mathbf{k}_u \le b_{in} \; ; A_{eq}\mathbf{k}_u = b_{eq}\right\}$.
2) Elaboration du domaine étendu $D$ (4.18).
3) Construction du lieu géométrique de l'optimum sans contraintes $R$ (4.8).

4) Application de la transformation (4.27). Il en résulte $\hat{D}, \hat{R}$ et l'ensemble des sommets paramétrés de $\hat{P}(x)$.

***Partie B***

5) Application de la procédure *Split* pour la construction du partitionnement $\Re^{p+n} = \hat{D} \bigcup_{i=1}^{\overline{n}_D} DP_i$

6) Intersection du lieu géométrique de l'optimum sans contraintes $\hat{R}$ avec chaque domaine $DP_i, i = 1,..,\overline{n}_D$ et stockage de toutes les zones $Z_j, j = 1,..,n_{\hat{D}}$ définies par :

$$Z_j = \left\{ \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \middle| \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \notin \bigcup_{k=1}^{j-1} Z_k ; \exists i \ t.q. \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \in DP_i ; x \in \Pr_x \left( DP_i \cap \hat{R} \right) \right\} \text{ avec } \Pr_x \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} = x$$

7) $S\hat{D} = \left(\hat{D} \cap \hat{R}\right) \cup \left( \Pr^x_{\hat{D} \cap Z_1} \hat{R} \right) \cup \ldots \cup \left( \Pr^x_{\hat{D} \cap Z_{n_{\hat{D}}}} \hat{R} \right)$

***Partie C***

8) Application de la transformation inverse de (4.27) pour retrouver $SD$.

L'opération $\Pr^x_{D \cap Z} R$ représente la projection selon $x$ de l'ensemble $R \in \Re^{r+n}$, $\dim(R) = n$ sur le domaine polyédral $D \cap Z \in \Re^{r+n}$, avec $\dim(D \cap Z) \geq n$ :

$$\Pr^x_{D \cap Z} R = \left\{ \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \in D \cap Z \middle| t.q. \forall \begin{bmatrix} \kappa \\ x \end{bmatrix} \in \hat{D}, \left( \begin{bmatrix} 0 \\ x \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \right)^T \left( \begin{bmatrix} \kappa \\ x \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \right) = 0 \right\}$$

**Remarque :** L'expression de la solution optimale d'un problème d'optimisation en terme d'une union d'ensembles convexes $S\hat{D}$ peut paraître loin des définitions habituelles des solutions explicites. Malgré tout, elle est tout à fait équivalente en termes géométriques. Pour éclaircir cet aspect, il convient de remarquer que $S\hat{D}$ est constitué d'ensembles convexes. En effet leur dimension est toujours égale à *n,* la dimension de l'espace des paramètres. De plus, comme leur construction est basée sur les sommets paramétrés, une conséquence immédiate est qu'il existe une correspondance 1 :1 entres les paramètres et les points de $S\hat{D}$. Dès lors, l'évaluation de la solution optimale pour une certaine valeur du vecteur de paramètres revient à :

$$\mathbf{k}_u^*(x_0) = \Pr_{\mathbf{k}_u} \begin{bmatrix} \mathbf{k}_u^* \\ x_0 \end{bmatrix} = \Pr_{\mathbf{k}_u} \left( SD \cap \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \middle| x = x_0 \right\} \right) \tag{4.34}$$

avec $\Pr_{\mathbf{k}_u} \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} = \mathbf{k}_u$. Comme la projection est une fonction linéaire, on peut globalement extraire cette dépendance et réécrire la solution analytique de (4.6) en terme de fonctions linéaires par morceaux.

**Exemple 4.3**

Considérons le problème de commande d'un système double intégrateur :

$$x_{t+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 1 \\ 0,5 \end{bmatrix} u_t \tag{4.35}$$

soumis à des contraintes sur les commandes non influencées par la paramétrisation :

$$-1 \leq u_t \leq 1 \tag{4.36}$$

Le problème d'optimisation associé à la synthèse de la loi prédictive, avec horizon de prédiction égal à l'horizon de commande $N = N_u = 2$, les pondérations $Q = I_2; R = 1$ et $P$ la solution de l'équation de Riccati (2.40), prend la forme :

$$\begin{aligned}
&\min_{\mathbf{k}_u} \mathbf{k}_u^T \begin{bmatrix} 14{,}656 & 7.3274 \\ 7{,}3274 & 7.4035 \end{bmatrix} \mathbf{k}_u + x_t^T \begin{bmatrix} 7{,}6564 & 3{,}848 \\ 17{,}655 & 10{,}807 \end{bmatrix} \mathbf{k}_u \\
&\text{sous} \quad \text{les contraintes :} \left\{ \begin{bmatrix} -1 \\ -1 \end{bmatrix} \le \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{k}_u \le \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right.
\end{aligned} \tag{4.37}$$

En suivant la procédure décrite précédemment, un découpage est donné Figure 4.9a. La commande appliquée effectivement selon le principe de l'horizon glissant est représentée par la première composante de la solution optimale. On peut ainsi représenter la loi de commande par une fonction linéaire par morceaux (Figures 4.9b et c) et conclure finalement à une partition qui identifie cinq régions avec des lois linéaires distinctes. Ces régions ne sont pas convexes mais exprimables sous forme d'une union finie d'ensembles convexes (comparer les Figures 4.9a et d).

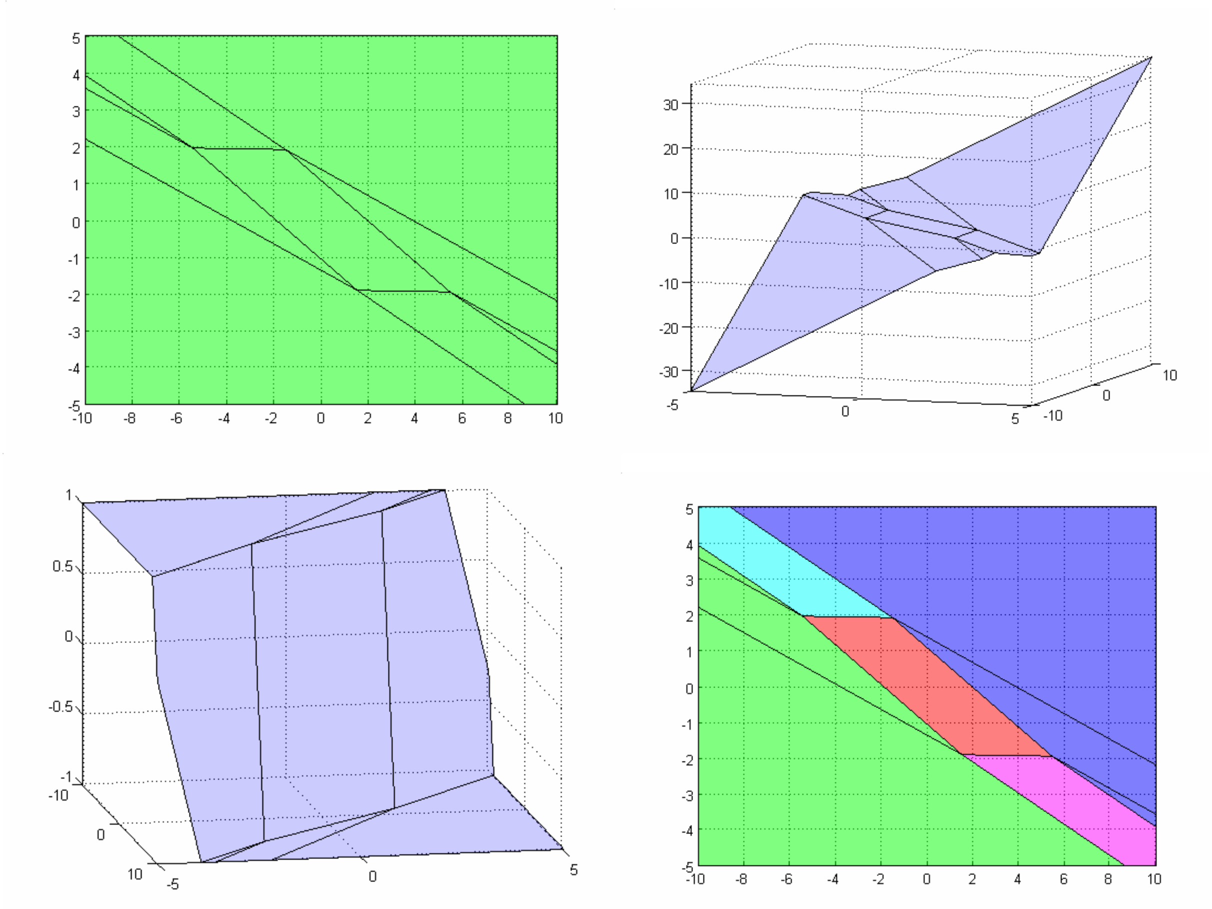
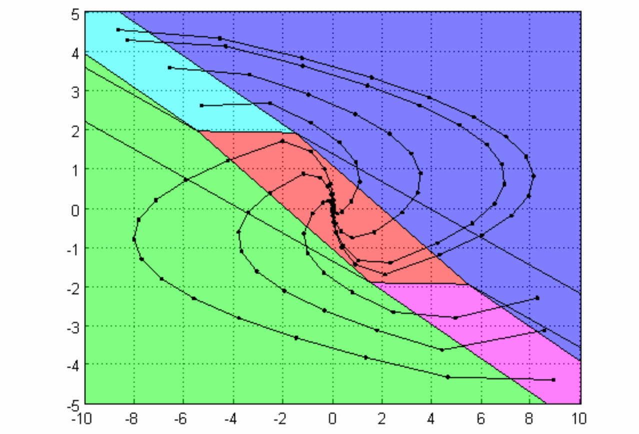
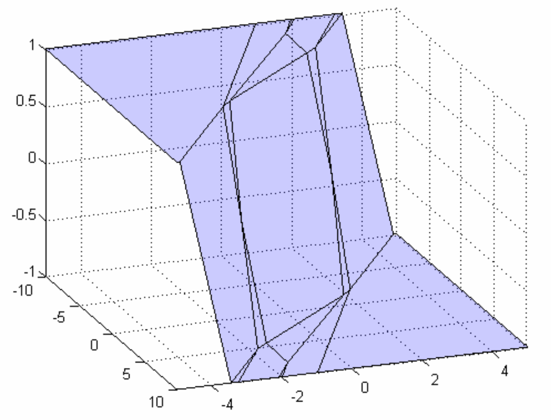



**Figure 4.9 : a) Partitionnement pour le problème du double intégrateur**
**b) Solution du problème modifié (4.27)**
**c) Solution réelle d) Découpage en différentiant les zones ayant une loi différente**

Pendant le processus de développement de cette solution explicite, la procédure a permis d'identifier les quatre sommets paramétrés issus de la transformation des coordonnées (4.29) et ensuite les 9 régions ayant une projection convexe. Puisque le lieu géométrique de l'optimum sans contraintes a une intersection non vide avec toutes ces régions, on retrouve les 9 régions dans la solution finale.

La validité du résultat peut être vérifiée en comparant la simulation du système asservi d'une part par une loi de commande synthétisée par des techniques d'optimisation en ligne, par la solution explicite d'autre part. Le résultat est représenté Figure 4.10a. On retrouve ensuite, Figure 4.10b et 4.10c le même double intégrateur mais avec un horizon de prédiction $N = N_u = 3$. L'augmentation du nombre de variables d'optimisation a modifié le nombre de zones de la solution explicite, puisque les régions avec projection convexe sont passées de 27 à 29.





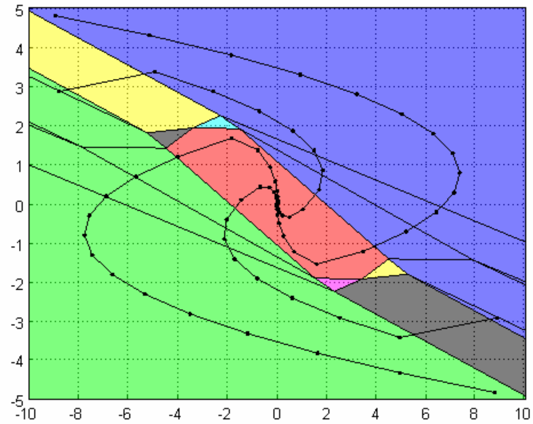


**Figure 4.10 : a) Simulations pour la loi MPC avec $N = 2$**
**b) Forme explicite avec $N = 3$**
**c) Simulations pour MPC avec $N = 3$**

### 4.2.2 Approche par polyèdres paramétrés pour la solution explicite – cas général

On a vu lors du développement de la solution explicite pour un problème d'optimisation avec contraintes linéaires non paramétrées et une fonction de coût influencée par un vecteur des paramètres qu'une procédure géométrique de découpage de l'espace des paramètres est envisageable sans modification du domaine faisable. Néanmoins, on a montré par la même stratégie que l'on pouvait par changement de variables arriver aux mêmes résultats, même si dans ce cas il devient nécessaire de faire intervenir le concept de polyèdres paramétrés. Cette approche va nous permettre maintenant de progresser vers une approche géométrique de la construction des solutions explicites pour des problèmes d'optimisations multiparamétriques dans le cas général. En effet, on considère désormais :

$$\min_{\mathbf{k}_u} J_t = 0{,}5\,\mathbf{k}_u^T\ H\,\mathbf{k}_u + x^T\ F\,\mathbf{k}_u$$
$$\text{sous les contraintes}: \begin{cases} A_{in}\mathbf{k}_u \le b_{in} + B_{in}x_t \\ A_{eq}\mathbf{k}_u = b_{eq} + B_{eq}x_t \end{cases} \tag{4.38}$$

On procède alors à un changement de variable de sorte que la paramétrisation de la fonction de coût soit entièrement transférée vers la paramétrisation du domaine faisable :

$$\hat{\mathbf{k}}_u = H^{1/2}\mathbf{k}_u + H^{-1/2}F^T x \tag{4.39}$$

d'où le problème :

$$\min_{\hat{\mathbf{k}}_u} J_t = 0{,}5\,\hat{\mathbf{k}}_u^T\ \hat{\mathbf{k}}_u$$
$$\text{sous les contraintes}: \begin{cases} \hat{A}_{in}\hat{\mathbf{k}}_u \le b_{in} + \hat{B}_{in}x \\ \hat{A}_{eq}\hat{\mathbf{k}}_u = b_{eq} + \hat{B}_{eq}x \end{cases} \tag{4.40}$$

où $\hat{A}_{in} = A_{in}H^{-1/2}; \hat{B}_{in} = B_{in} + A_{in}H^{-1}F^T; \hat{A}_{eq} = A_{eq}H^{-1/2}; \hat{B}_{eq} = B_{eq} + A_{eq}H^{-1}F^T$. Toutes ces opérations sont possibles sous l'hypothèse que $H = H^T > 0$, ce qui est le cas pour la commande prédictive.

En analysant le problème (4.40), on constate que la fonction de coût possède un optimum sans contraintes $\hat{\mathbf{k}}_u^*$ unique à l'origine. L'optimisation revient donc à retrouver la projection de ce point sur le domaine :

$$P = \left\{ \hat{\mathbf{k}}_u \,\middle|\, \hat{A}_{in}\hat{\mathbf{k}}_u \leq b_{in} + \hat{B}_{in}\, x\, ; \hat{A}_{eq}\hat{\mathbf{k}}_u = b_{eq} + \hat{B}_{eq}\, x \right\} \tag{4.41}$$

On constate que ce domaine représente un polyèdre paramétré qui peut être représenté également par un ensemble de générateurs :

$$\hat{P}(x) = \left\{ \hat{\mathbf{k}}_u(x) \middle| \hat{\mathbf{k}}_u(x) = \sum_{i=1}^{d_S} \sigma_i(x) s_i(x) + \sum_{i=1}^{d_R} \rho_i r_i + \sum_{i=1}^{d_L} \lambda_i l_i ; 0 \leq \sigma_i \leq 1; \sum_{i=1}^{d_S} \sigma_i = 1; \rho_i \geq 0; \forall \lambda_i \right\} \tag{4.42}$$

Chaque sommet paramétré $s_i(x), i = 1,.., d_S$ a un domaine de validité correspondant $VD_i \subset \Re^n$.

Par rapport au paragraphe précédent, la formulation diffère ici par le fait qu'on ne possède aucune information a priori sur les ensembles de contraintes linéaires impliquées. Auparavant, on détenait un ensemble globalement non redondant qui garantissait le fait que $\hat{P}(x) \neq 0, \forall x \in \Re^n$. Il est donc nécessaire d'analyser les modifications à apporter à la procédure de construction des formulations explicites.

**Cas des domaines réguliers**

Sachant que le paramètre intervient désormais dans la définition des contraintes, il est nécessaire de relaxer l'affirmation $\hat{P}(x) \neq 0, \forall x \in \Re^n$. Si l'on conserve l'hypothèse que l'ensemble des contraintes est globalement non redondant, on détient en fait un domaine régulier pour lequel l'ensemble des générateurs reste constant. En vertu des résultats présentés au chapitre 3, on peut déduire qu'étant régulier, le polyèdre paramétré $\hat{P}(x)$ sera non vide pour toutes les valeurs des paramètres pour lesquelles il existe des sommets $s_i(x), i = 1,.., d_S$ dans l'ensemble de générateurs. En conséquence, comme l'ensemble de générateurs est constant et que le polyèdre $\hat{P}(x)$ n'existe que là où les sommets sont bien définis, on conclut :

$$\hat{P}(x) \neq 0, \forall x \in VD \subset \Re^n \tag{4.43}$$

où $VD = VD_1 = ... = VD_{d_S}$ représente le domaine de validité des sommets paramétrés qui reste le même pour tout l'ensemble.

Les types de polyèdres paramétrés bénéficiant de cette relaxation se rangent en deux catégories :

- Les polyèdres incluant exclusivement des contraintes sur les paramètres :

$$\begin{cases} b_{in} + \hat{B}_{in}\, x \geq 0 \\ b_{eq} + \hat{B}_{eq}\, x = 0 \end{cases} \tag{4.44}$$

- Les polyèdres contenant des sommets fusionnant pour une certaine valeur de paramètres et qui représentent ainsi la frontière de leurs domaines de validité. Formellement on peut décrire une telle situation pour un polyèdre défini par (4.40) par :

$$\exists x_0 \in VD\ t.q.\ s_1(x_0) = s_2(x_0) = ... = s_{d_S}(x_0) \tag{4.45}$$

**Exemple 4.4**
Pour illustrer ce cas de polyèdres paramétrés avec domaines réguliers, considérons l'exemple du domaine faisable obtenu après application de la transformation (4.39) :

$$\hat{P}(x) : \begin{bmatrix} 0 & 2 \\ -2 & 0 \\ 0 & -2 \\ 2 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{k}_u \geq \begin{bmatrix} 6 \\ -4 \\ -2 \\ 6 \\ -1 \end{bmatrix} x + \begin{bmatrix} -5 \\ 1 \\ 1 \\ -3 \\ 0 \end{bmatrix} \tag{4.46}$$

On y retrouve une contrainte sur le paramètre (représentée par la dernière ligne de l'ensemble (4.46)) et on observe que pour $x=1$ les sommets paramétrés sont superposés car le polyèdre se réduit à un point :

$$\hat{P}(x=1)=\left\{\begin{bmatrix}1{,}5\\0{,}5\end{bmatrix}\right\}. \tag{4.47}$$

En passant par la double représentation, on obtient la description graphique de ce polyèdre paramétré dans l'espace étendu de vecteur $\begin{bmatrix}\mathbf{k}_u^T & x\end{bmatrix}^T$, Figure 4.11a.

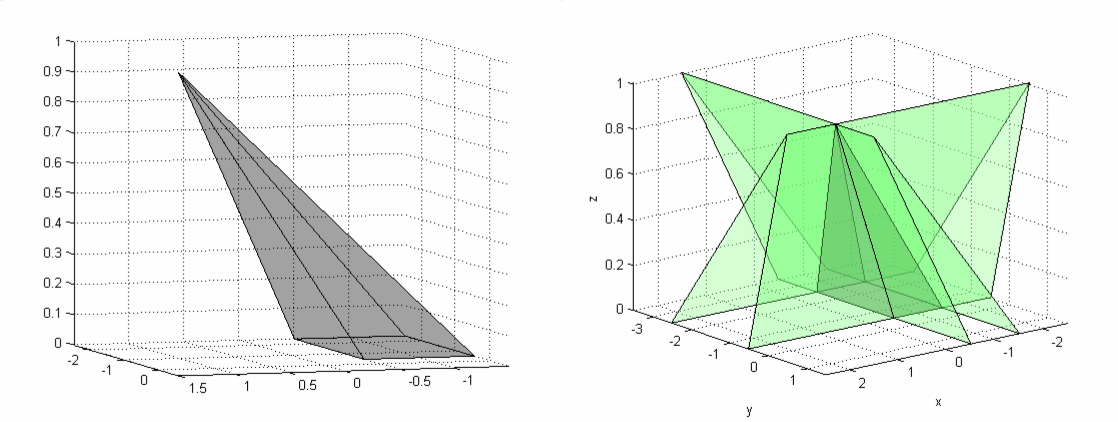



**Figure 4.11 : a) Polyèdre paramétré dans l'espace étendu $\begin{bmatrix}\mathbf{k}_u^T & x\end{bmatrix}^T$**
**b) Partition en régions de projection sur la même frontière.**

Il faut remarquer que, par rapport au cas des polyèdres paramétrés de type (4.19-4.20) pour lesquels la construction de l'ensemble des générateurs se faisait par ajout de rayons bidirectionnels complémentaires, dans le cas des domaines réguliers la construction de la double représentation doit passer par un appel à une procédure de type Chernikova classique. Le résultat sera :

$$\hat{D}=\left\{\begin{bmatrix}\mathbf{k}_u\\x\end{bmatrix}\middle|\begin{bmatrix}\mathbf{k}_u\\x\end{bmatrix}=\sum_{i=1}^{d_S}\sigma_i\hat{s}_i+\sum_{i=1}^{d_R}\rho_i r_i+\sum_{i=1}^{d_L}\lambda_i l_i\right\}, 0\le\sigma_i\le 1,\ \sum_{i=1}^{v}\sigma_i=1,\ \rho_i\ge 0\ ,\forall\lambda_i \tag{4.48}$$

A partir de $\hat{D}$ on peut alors retrouver par projection les sommets paramétrés de $\hat{P}(x)$ et leurs domaines de validité – $VD$ (voir chapitre 3).

Une fois acquise cette description de domaine régulier dans l'espace étendu – $\hat{D}$, on peut utiliser la procédure '*ExplicitPPP*' pour retrouver la solution explicite en terme d'ensemble :

$$S\hat{D}=\left\{\begin{bmatrix}\hat{\mathbf{k}}_u\\x\end{bmatrix}\in\hat{D}\middle| t.q.\forall\begin{bmatrix}\kappa\\x\end{bmatrix}\in\hat{D},\left(\begin{bmatrix}0\\x\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{k}}_u\\x\end{bmatrix}\right)^T\left(\begin{bmatrix}\kappa\\x\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{k}}_u\\x\end{bmatrix}\right)\le 0\right\} \tag{4.49}$$

En analysant l'ensemble des étapes de cette procédure, on constate qu'il n'existe pas dans la *partie B* de la procédure de référence explicite au fait que les combinaisons faisables de paramètres doivent couvrir l'espace complet des paramètres. En effet, tant que le domaine faisable reste régulier, la seule conséquence de la relaxation (4.43) se retrouve au niveau de (4.32) (et donc de l'étape 5) qui devient :

$$VD=\bigcup_{i=1}^{\overline{n}_D}DP_i \tag{4.50}$$

En conclusion pour le cas des problèmes d'optimisation avec domaines faisables décrits par polyèdres paramétrés de forme régulière, il est nécessaire de reformuler la *Partie A* de la procédure '*ExplicitPPP*' pour construire la forme duale et ensuite appliquer la *Partie B* pour le polyèdre $\hat{D}$ résultant.

**Exemple 4.4 (suite)**
En appliquant la procédure '*ExplicitPPP*' à l'exemple (4.46), on obtient tout d'abord la partition de l'espace étendu en régions ayant des projections convexes sur les faces de $\hat{D}$ (Figure 4.11b). Ensuite, après intersection avec la verticale passant par l'origine, seules quatre des régions (4.32) sont retenues (Figure 4.12a) et finalement

leurs intersections avec $\hat{D}$ déterminent l'union des sous-ensembles convexes $S\hat{D}$ qui décrivent la formulation explicite de la solution (Figure 4.12b).

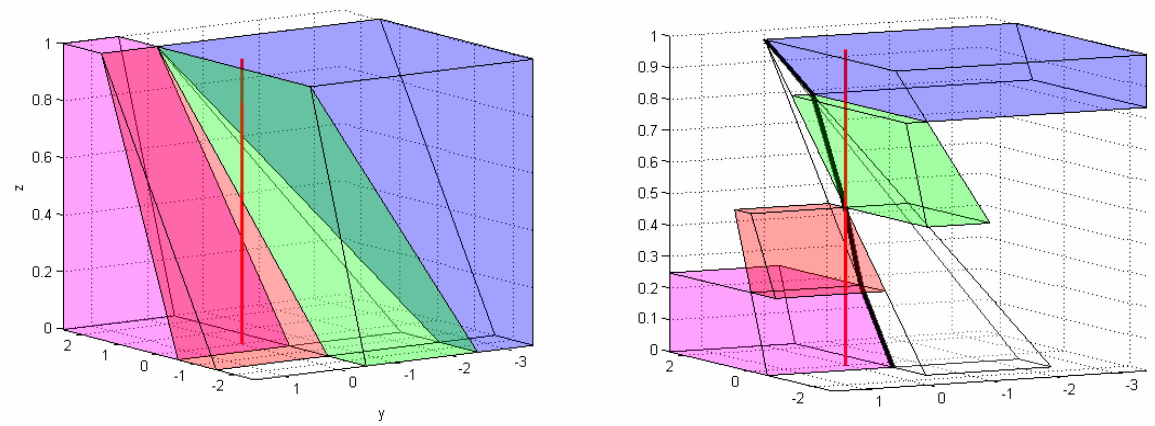



**Figure 4.12 : a) Partitions ayant une intersection avec le lieu géométrique de l'optimum sans contraintes b) Solution optimale issue de l' intersection de l'espace faisable avec les zones adjacentes ayant une intersection avec l'optimum sans contraintes.**

**Cas général**

On dispose à ce stade d'une procédure d'élaboration de la solution explicite d'un problème d'optimisation avec un domaine faisable de forme régulière (et donc décrit par un ensemble non redondant de contraintes pour toute valeur du vecteur de paramètres). Mais dans le cas général, le problème :

$$\min_{\mathbf{k}_u} J_t = 0{,}5\,\mathbf{k}_u^T \; H\,\mathbf{k}_u + x_t^T \; F\,\mathbf{k}_u$$
$$\text{sous les contraintes :}\begin{cases} A_{in}\mathbf{k}_u \le b_{in} + B_{in}x_t \\ A_{eq}\mathbf{k}_u = b_{eq} + B_{eq}x_t \end{cases} \tag{4.51}$$

ne garantit pas globalement une telle propriété. Néanmoins, on a pu voir lors de l'analyse géométrique des domaines faisable que des sous-ensembles de contraintes localement non redondantes peuvent être identifiés, par un pré-découpage de l'espace des paramètres.

La procédure de construction de solutions explicites dans le cas général va utiliser cette approche pour identifier les problèmes qui peuvent être traités par une procédure de type '*ExplicitPPP'*. En fait, cette procédure ne peut être appliquée directement dans le cas général (4.51) car elle fait appel à une énumération des régions adjacentes du domaine faisable, basée sur les sommets paramétrés. La procédure ci-dessous est donc une version modifiée adaptée au cas général.

***<u>Procédure ExplicitPPPGen</u>***

1) Construction de la double représentation :

$$P(x) = \{\mathbf{k}_u | A_{in}\mathbf{k}_u \le b_{in} + B_{in}x ; A_{eq}\mathbf{k}_u = b_{eq} + B_{eq}x\} =$$
$$= \left\{ \mathbf{k}_u(x) \middle| \mathbf{k}_u(x) = \sum_{i=1}^{d_S} \sigma_i(x) s_i(x) + \sum_{i=1}^{d_R} \rho_i r_i + \sum_{i=1}^{d_L} \lambda_i l_i \right\}, \tag{4.52}$$
$$0 \le \sigma_i \le 1,\ \sum_{i=1}^{v} \sigma_i(x) = 1\,,\ \rho_i \ge 0\ ,\forall \lambda_i$$

Pour chaque sommet paramétré $s_i(x)$, identifier les domaines de validité $VD_i, i = 1,..,n_S$ dans l'espace des paramètres de contexte. L'union de ces domaines représente l'ensemble maximal des paramètres faisables :

$$VD = \bigcup_{i=1}^{n_S} VD_i \tag{4.53}$$

2) En se basant sur les domaines de validité $VD_i, i = 1,..,n_S$, identifier les ensembles de paramètres $DR_j$ pour lesquels $P(x)$ a une forme régulière. Si le nombre de ces domaines est $n_{DR}$, alors il faut que :

$$VD = \bigcup_{j=1}^{n_{DR}} DR_j \tag{4.54}$$

3) Pour chaque domaine $DR_j$, $j = 1,..,n_{DR}$, retrouver l'ensemble des contraintes non redondantes :

$$DR_j \rightarrow \begin{cases} A_{in_j}\mathbf{k}_u \le b_{in_j} + B_{in_j} x_t \\ A_{eq_j}\mathbf{k}_u = b_{eq_j} + B_{eq_j} x_t \end{cases} \tag{4.55}$$

4) Pour $j = 1,..,n_{DR}$ :

i) Construire la double représentation du polyèdre décrit par l'ensemble des contraintes :

$$P_j(x) : \begin{cases} A_{in_j}\mathbf{k}_u \le b_{in_j} + B_{in_j} x_t \\ A_{eq_j}\mathbf{k}_u = b_{eq_j} + B_{eq_j} x_t \\ x_t \in DR_j \end{cases} \tag{4.56}$$

ii) Augmenter $P_j(x)$ à un polyèdre non paramétré dans une dimension étendue :

$$D_j = \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \middle| \mathbf{k}_u(x) \in P_j(x) \right\} \tag{4.57}$$

iii) Construire le lieu géométrique de l'optimum sans contraintes :

$$R = \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \in \Re^{p+n} \middle| \begin{bmatrix} I & H^{-1}F^T \end{bmatrix} \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} = 0 \right\} \tag{4.58}$$

iv) En effectuant une changement de coordonnées on obtient :

$$\begin{aligned} & P(x) \xrightarrow{\hat{\mathbf{k}}_u = H^{1/2}\mathbf{k}_u - H^{-1/2}F^T x} \hat{P}(x) \\ & D_j \xrightarrow{\hat{\mathbf{k}}_u = H^{1/2}\mathbf{k}_u - H^{-1/2}F^T x} \hat{D}_j \\ & R \xrightarrow{\hat{\mathbf{k}}_u = H^{1/2}\mathbf{k}_u - H^{-1/2}F^T x} \hat{R} = \left\{ \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \in \Re^{p+n} \middle| \hat{\mathbf{k}}_u = 0 \right\} \end{aligned} \tag{4.59}$$

v) Appliquer la procédure '*Split*' pour la construction du partitionnement :

$$DR_j = \hat{D}_j \bigcup_{i=1}^{\overline{n}_{Dj}} DP_{ij} \tag{4.60}$$

correspondant à des régions adjacentes de $\hat{D}$ caractérisées par des projections sur la même frontière de $\hat{D}$.

vi) Intersection de $\hat{R}$ avec chaque domaine $DP_{ij}, i = 1,..,\overline{n}_{Dj}$ et stockage de toutes les zones $Z_{kj}, k = 1,..,n_{\hat{D}_j}$ ($n_{\hat{D}j} \le \overline{n}_{Dj}$) définies par :

$$Z_{kj} = \left\{ \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \middle| \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \notin \bigcup_{l=1}^{k-1} Z_{lj} ; \exists i \ t.q. \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \in DP_{ij} ; x \in \mathrm{Pr}_x\left(DP_{ij} \cap \hat{R}\right) \right\} \tag{4.61}$$

vii) $$S\hat{D}_j = \left(\hat{D}_j \cap \hat{R}\right) \cup \left(\Pr^x_{\hat{D}_j \cap Z_{1j}} \hat{R}\right) \cup \ldots \cup \left(\Pr^x_{\hat{D}_j \cap Z_{n_{\hat{D}_j} j}} \hat{R}\right) \tag{4.62}$$

5) $$S\hat{D} = \bigcup_{j=1}^{n_{DR}} S\hat{D}_j \tag{4.63}$$

6) $$S\hat{D} \xrightarrow{\mathbf{k}_u = H^{-1/2}\hat{\mathbf{k}}_u + H^{-1}F^T x} SD \tag{4.64}$$

7) $SD$ est une union d'ensembles convexes $SD = \bigcup SD_i$ de dimension $n$ dans $\Re^{p+n}$ pouvant être stockés comme des fonctions affines par morceaux :

$$\mathbf{k}_u(x) = K_{Lin_i} x + K_{tl_i} \quad pour \quad x \in \Pr_x SD_i \tag{4.65}$$

**Remarque :** La solution explicite (4.65) peut contenir des régions différentes dans l'espace de paramètres – $\Pr_x SD_i, \Pr_x SD_j$ pour lesquelles la solution explicite possède une formulation similaire :

$$K_{Lin_i} x + K_{tl_i} = K_{Lin_j} x + K_{tl_j}$$

Dans ce cas, si l'union $\Pr_x SD_i \cup \Pr_x SD_j$ représente un ensemble convexe, alors cette union doit être calculée (des procédures effectives étant présentées dans [BFT02]) pour permettre une expression plus compacte de (4.65).

**Remarque :** Toute la procédure proposée ci-dessus consiste en une succession d'opérations géométriques qui procure au fur et à mesure de son développement une structuration hiérarchisée des partitions de l'espace des paramètres. En effet, on peut observer que l'étape 4 est déjà basée sur un découpage de l'espace faisable en régions avec des formes régulières qui sont ensuite partitionnées à leur tour selon les lois de projection caractéristiques, d'où un deuxième niveau hiérarchique (Figure 4.13). Du point de vue implémentation, le partitionnement final est suffisant en lui-même pour décrire la solution explicite. Néanmoins, détenir un niveau intermédiaire de découpage offre d'une part un complément d'information sur les causes du changement de la description locale (4.65), et d'autre part peut aider lors du processus de recherche de la région active pour une valeur particulière du vecteur de paramètres de contexte $x$.

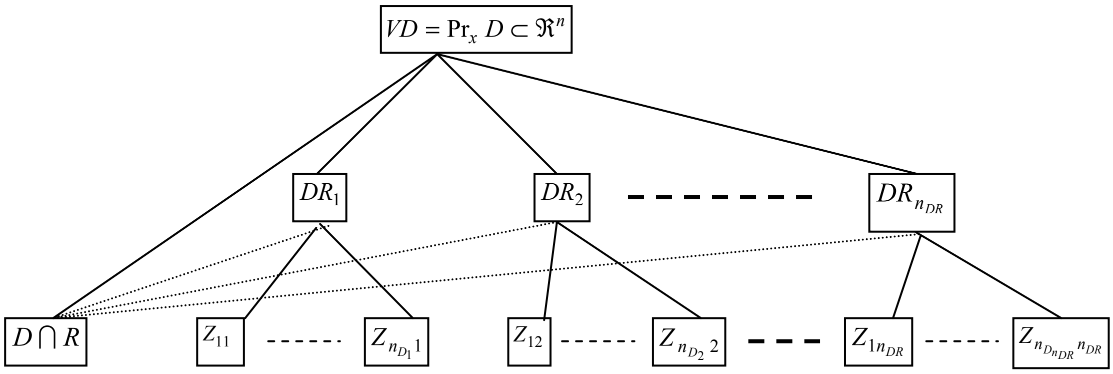



**Figure 4.13 : Découpages dans l'espace des paramètres pour un problème d'optimisation multiparamétrique**

**Exemple didactique 4.5**
Considérons pour expliquer les différentes étapes de la procédure précédente un exemple de problème d'optimisation simple qui permet une visualisation de toutes les étapes dans l'espace étendu $\Re^3$ :

$$\min_{\mathbf{k}_u} \mathbf{k}_u^T \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \mathbf{k}_u + x^T \begin{bmatrix} -1 & 1 \end{bmatrix} \mathbf{k}_u$$
$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -2 & 0 \\ 0 & 1 \\ 0 & -1 \\ -1 & 0 \\ 1 & 0 \\ 0 & -2 \end{bmatrix} \mathbf{k}_u \leq \begin{bmatrix} 0 \\ 0 \\ -2 \\ 1 \\ 0 \\ 0 \\ 1 \\ -2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 3 \\ 0 \\ 0 \\ 3 \\ 1 \end{bmatrix} \tag{4.66}$$

Les étapes parcourues pour la construction sont les suivantes :

1) La représentation duale :

$$\mathbf{k}_u(x) = \left\{ \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{x\in VD_1}; \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{x\in VD_2}; \underbrace{\begin{bmatrix} x+1.5 \\ 0 \end{bmatrix}}_{x\in VD_3}; \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{x\in VD_4}; \underbrace{\begin{bmatrix} 1 \\ x-0.5 \end{bmatrix}}_{x\in VD_5}; \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{x\in VD_1}; \underbrace{\begin{bmatrix} 0 \\ x+0.5 \end{bmatrix}}_{x\in VD_3}; \underbrace{\begin{bmatrix} x-0.5 \\ 1 \end{bmatrix}}_{x\in VD_5}; \underbrace{\begin{bmatrix} x-0.5 \\ x-0.5 \end{bmatrix}}_{x\in VD_5}; \underbrace{\begin{bmatrix} x+1.5 \\ x+1.5 \end{bmatrix}}_{x\in VD_3} \right\}; \tag{4.67}$$
$$VD_1 = [-0.5;0.5]; VD_2 = [-1.5;0.5]; VD_3 = [-1.5;-0.5]; VD_4 = [-0.5;1.5]; VD_5 = [0.5;1.5]$$

qui correspond aux faces d'ordre 1 du polyèdre non paramétré de la Figure 4.14a. L'union des domaines de validité est :

$$VD = \bigcup_{i=1}^{5} VD_5 \tag{4.68}$$

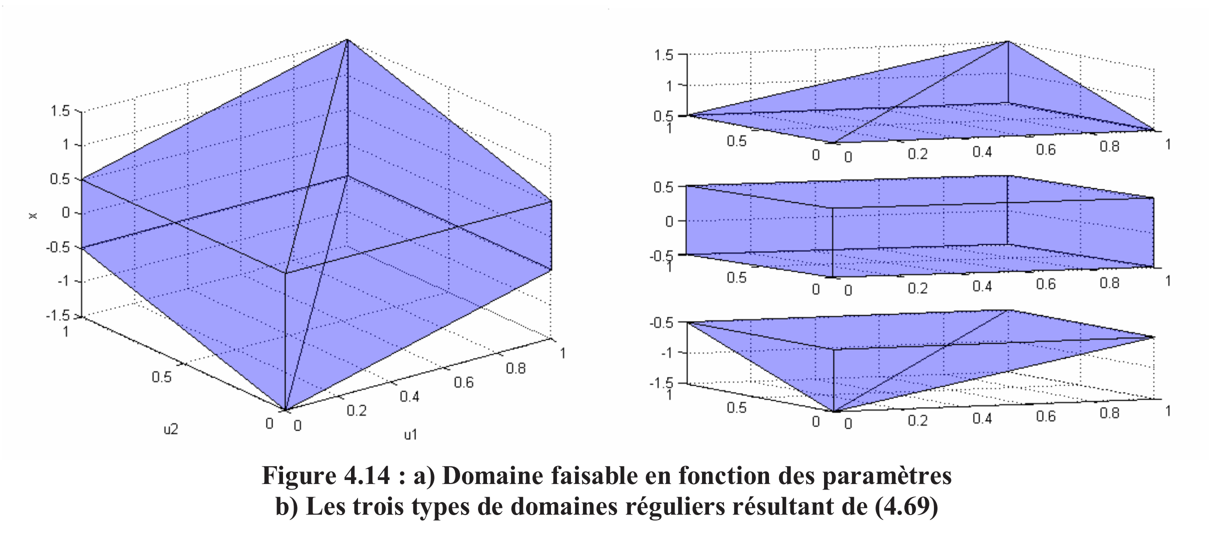



**Figure 4.14 : a) Domaine faisable en fonction des paramètres**
**b) Les trois types de domaines réguliers résultant de (4.69)**

2-3) Par intersection entre les domaines de validité précédents, on obtient les zones de forme régulière :

$$\begin{aligned} &VD = DR_1 \cup DR_2 \cup DR_3; \\ &DR_1 = [-1.5;-0.5]; DR_2 = [-0.5;0.5]; DR_3 = [0.5;1.5]; \end{aligned} \tag{4.69}$$

Pour chaque domaine (Figure 4.14b), on connaît les sommets paramétrés valides et les contraintes qu'ils saturent.

4) Pour $j = 1,..,3$

i)-iv) L'application du changement de coordonnées et la construction des polyèdres dans l'espace étendu et du lieu géométrique de l'optimum sans contraintes est donnée Figure 4.15.

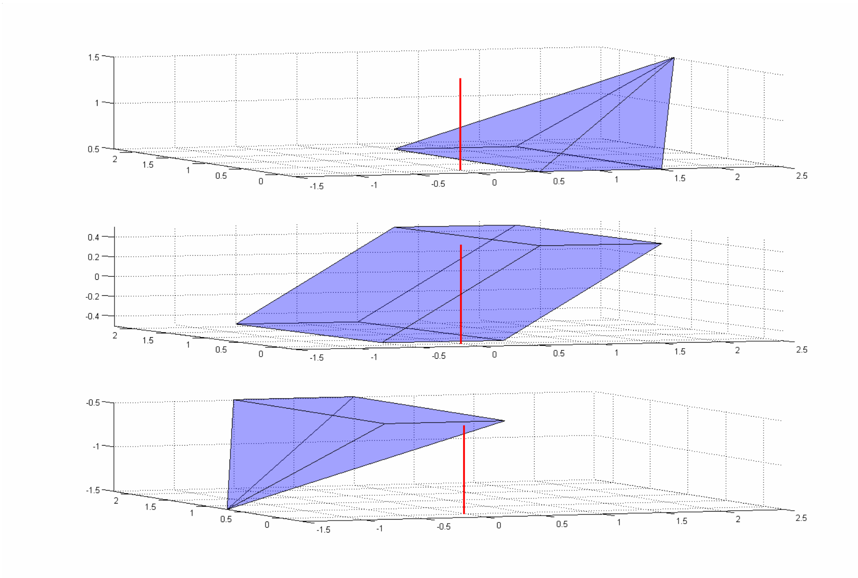


**Figure 4.15 : Transformation des domaines réguliers et leur rapport avec le lieu géométrique de l'optimum sans contraintes (rouge).**

v) Le partitionnement en régions adjacentes à $DR_j$ se fait de la façon suivante :

$$DR_1 = \hat{D}_1 \bigcup_{i=1}^{8} DP_{i1}; DR_2 = \hat{D}_2 \bigcup_{i=1}^{8} DP_{i2}; DR_3 = \hat{D}_3 \bigcup_{i=1}^{8} DP_{i3} \tag{4.70}$$

On peut observer (Figure 4.16a) que chaque région régulière contient 4 sommets paramétrés et on retrouve dans la version étendue les 9 régions décrites Figure 4.3. L'avantage de la double représentation est qu'elle nous permet de déterminer leurs limitations dans l'espace des paramètres et leur rapport à l'optimum sans contraintes.

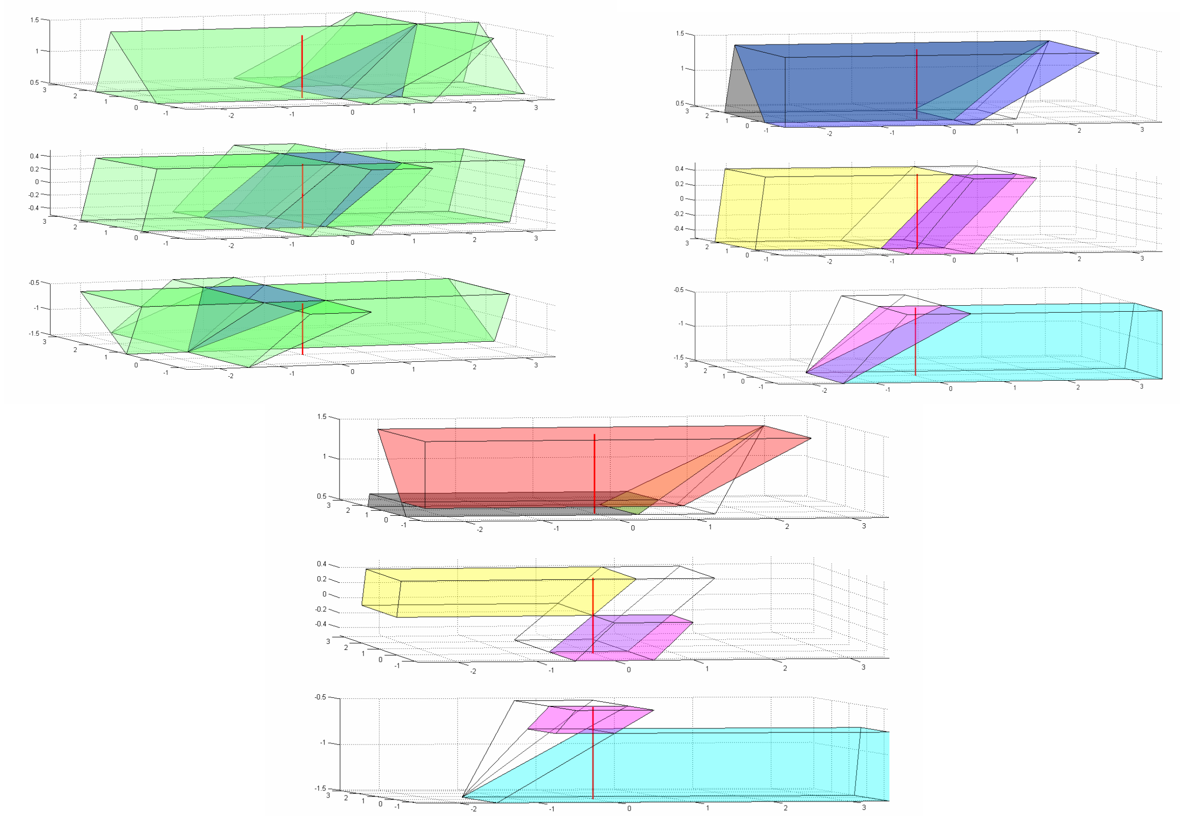


**Figure 4.16 : a) Partition en régions avec projection sur la même frontière**
**b) Sélection des régions d'intersection non vide avec le lieu géométrique de l'optimum**
**c) Régions $Z_{ij}$ qui serviront pour la description de la solution optimale**

vi) Ce rapport est le résultat d'intersections successives permettant de ne retenir que deux régions par zone régulière (Figures 4.16b et 4.16c) :

$$DR_1 \rightarrow \{Z_{11};Z_{21}\}; DR_1 \rightarrow \{Z_{12};Z_{22}\}; DR_1 \rightarrow \{Z_{13};Z_{23}\} \tag{4.71}$$

vii) La solution optimale pour chaque zone de forme régulière (Figure 4.17a) est donnée par l'intersection avec le domaine faisable :

$$S\hat{D}_1 = \left(\hat{D}_1 \cap Z_{11}\right) \cup \left(\hat{D}_1 \cap Z_{21}\right); S\hat{D}_2 = \left(\hat{D}_2 \cap Z_{12}\right) \cup \left(\hat{D}_2 \cap Z_{22}\right); S\hat{D}_3 = \left(\hat{D}_3 \cap Z_{13}\right) \cup \left(\hat{D}_3 \cap Z_{23}\right) \tag{4.72}$$

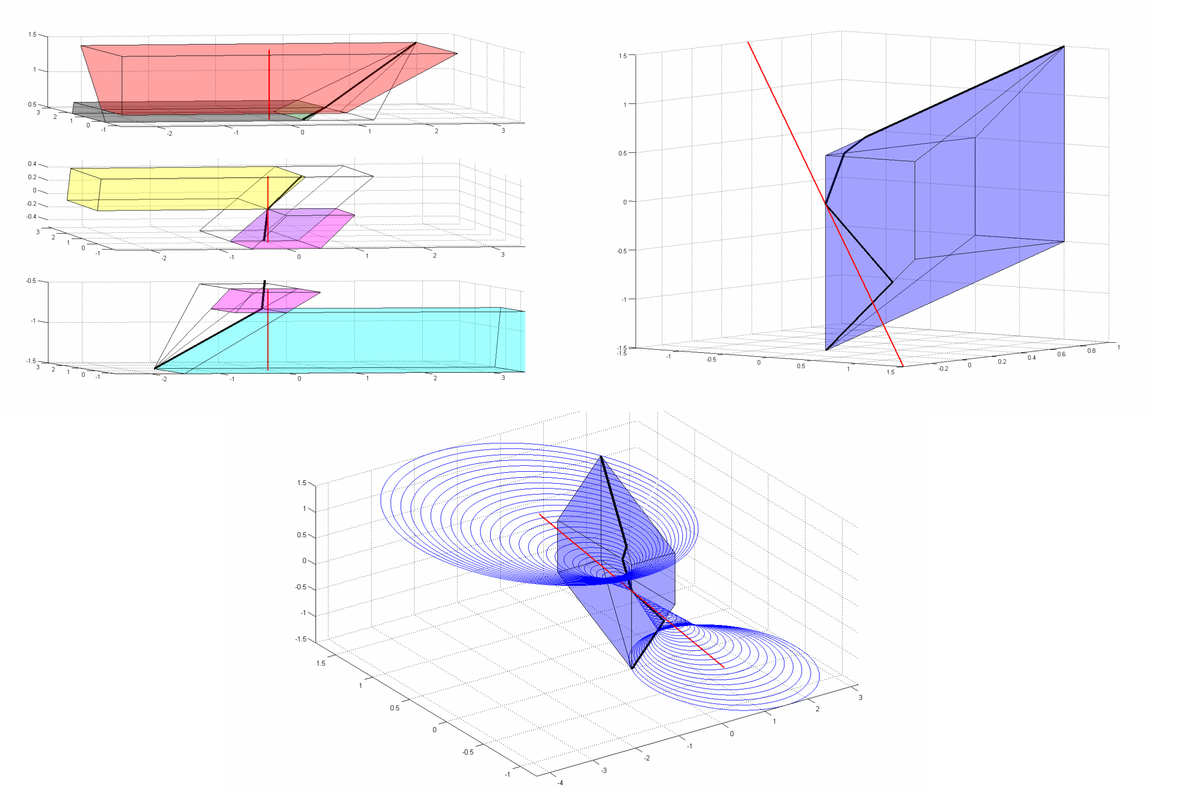


**Figure 4.17 : a) Construction de l'union des ensembles convexes qui décrivent la solution optimale**
**b) Solution explicite après la transformation inverse**
**c) Illustration des courbes isocout touchant le domaine faisable aux points contenus dans $S\hat{D}$**

5-6) L'union des solutions partielles procure la solution finale (Figure 4.17b) :

$$S\hat{D} = \bigcup_{j=1}^{3} S\hat{D}_j \xrightarrow{\mathbf{k}_u = H^{-1/2}\hat{\mathbf{k}}_u + H^{-1}F^T x} SD \tag{4.73}$$

7) La solution (4.73) peut être réécrite :

$$\mathbf{k}_u(x) = \begin{cases} \begin{bmatrix} 0 \\ -0{,}5 \end{bmatrix} x \quad \text{pour} \quad x \in [-0{,}75;0] \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} x \quad \text{pour} \quad x \in [0;0.5] \\ \begin{bmatrix} 2 \\ -0{,}5 \end{bmatrix} x + \begin{bmatrix} 1{,}5 \\ 0 \end{bmatrix} \quad \text{pour} \quad x \in [-1{,}5;-0{,}75] \\ \begin{bmatrix} 2 \\ 0 \end{bmatrix} x + \begin{bmatrix} -0{,}5 \\ 0 \end{bmatrix} \quad \text{pour} \quad x \in [0{,}5;0{,}66] \\ \begin{bmatrix} 2 \\ 1{,}5 \end{bmatrix} x + \begin{bmatrix} -0{,}5 \\ -1 \end{bmatrix} \quad \text{pour} \quad x \in [0{,}66;1{,}5] \end{cases} \tag{4.74}$$

**Remarque :** Pour cet exemple, on peut vérifier la validité de la solution optimale $SD$ selon la définition :

$$SD = \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \in D \middle| t.q. \forall \begin{bmatrix} \kappa \\ x \end{bmatrix} \in D, \left( \begin{bmatrix} H^{-1} F^T x \\ x \end{bmatrix} - \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \right)^T H \left( \begin{bmatrix} \kappa \\ x \end{bmatrix} - \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \right) \leq 0 \right\} \tag{4.75}$$

qui peut être illustrée par les courbes isocout comme indiqué Figure 4.17c.

**Remarque :** Les découpages intermédiaires effectués durant la procédure de construction de la solution explicite mettent en évidence une structure hiérarchisée pour le positionnement lors de l'évaluation de (4.74), Figure 4.18.

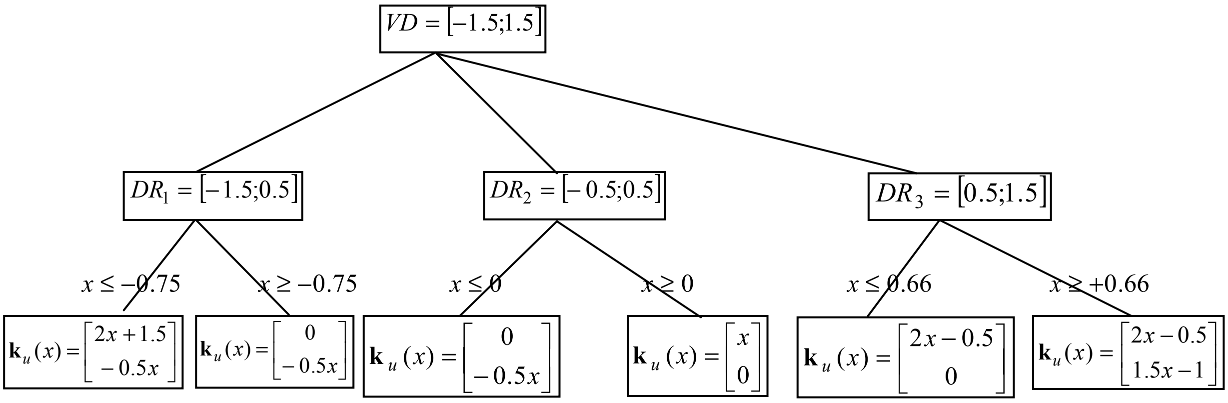



**Figure 4.18 : Hiérarchie des partitions basée sur les zones de topologie régulière. Les feuilles contiennent la solution explicite du problème.**

**Exemple 4.6 (suite de l'exemple 4.3)**
Les méthodes basées sur la double représentation des polyèdres paramétrés peuvent être utilisées pour la construction des lois prédictives explicites. En considérant de nouveau le système double intégrateur :

$$x_{t+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 1 \\ 0{,}5 \end{bmatrix} u_t \tag{4.76}$$

cette fois sous des contraintes mixtes sur les commandes et sur les états :

$$-1 \leq u_t \leq 1 \qquad \begin{bmatrix} -5 \\ -5 \end{bmatrix} \leq x_t \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix} \tag{4.77}$$

mais aussi avec des contraintes terminales :

$$x_{t+N} \in X_N \tag{4.78}$$

avec $X_N$ un ensemble polyédral construit à partir de la loi optimale sans contraintes $K_{LQR}$. Le problème d'optimisation associé à la synthèse de la loi prédictive ( $N = N_u = 2$ ) a la forme donnée par la relation (4.79).

On observe dans l'ensemble des contraintes (4.79) la présence de la paramétrisation qui influence la structure du domaine faisable et par la suite également la solution explicite. Cette solution sera décrite par une partition de l'espace des paramètres de contexte (les états $x_t \in \Re^2$ ).

- On distingue tout d'abord un partitionnement en régions caractérisées par un ensemble de contraintes localement non redondantes (Figure 4.19a). Il est constitué de 27 polyèdres, leur union représentant toutes les combinaisons faisables des paramètres. Une conséquence immédiate est que, pour tous les états se trouvant en dehors de la zone verte de la Figure 4.19a, la loi prédictive ne peut pas proposer de commande viable.
- Ensuite la procédure de positionnement du domaine faisable par rapport à l'optimum sans contraintes conduit à une seconde discrimination à l'intérieur des partitions déjà existantes, résultant en un total de 55 régions (Figure 4.19b) pour lesquelles on met en évidence des dépendances linéaires affines qui caractérisent explicitement la solution optimale de (4.79) comme une fonction des paramètres de contexte.

$$\min_{\mathbf{k}_u=[u_t \;\; u_{t+1}]^T} \mathbf{k}_u^T \begin{bmatrix} 14.656 & 7.3274 \\ 7.3274 & 7.4035 \end{bmatrix} \mathbf{k}_u + x_t^T \begin{bmatrix} 7.6564 & 3.848 \\ 17.655 & 10.807 \end{bmatrix} \mathbf{k}_u$$

$$\text{sous les contraintes :} \left\{ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0.57735 & 0 \\ -0.57735 & 0 \\ 0 & 1 \\ 0 & -1 \\ -0.48171 & -0.38153 \\ 0.48171 & 0.38153 \\ -0.41159 & -0.40756 \\ 0.41159 & 0.40756 \end{bmatrix} \mathbf{k}_u \leq \begin{bmatrix} 5 \\ 5 \\ 1 \\ 1 \\ 2.8868 \\ 2.8868 \\ 1 \\ 1 \\ 0.38549 \\ 0.38549 \\ 1.5093 \\ 1.5093 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ -0.57735 & -0.57735 \\ 0.57735 & 0.57735 \\ 0 & 0 \\ 0 & 0 \\ 0.20036 & 0.76305 \\ -0.20036 & -0.76305 \\ 0.0080541 & 0.81512 \\ -0.0080541 & -0.81512 \end{bmatrix} x_t \right. \tag{4.79}$$

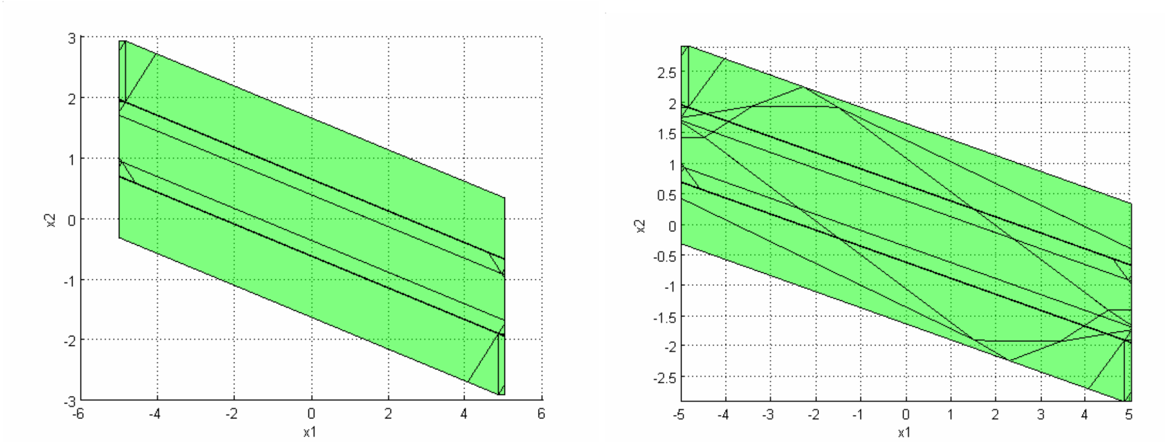


**Figure 4.19 : a) Découpage de l'espace des paramètres (états du système) en régions correspondant à des domaines faisables réguliers**
**b) Découpage de l'espace des paramètres correspondant à la solution explicite.**

Si l'on s'intéresse uniquement à la première partie de la solution optimale, qui sera appliquée effectivement selon le principe de l'horizon fuyant, on obtient une loi de commande linéaire par morceaux (Figure 4.20a). On peut distinguer une symétrie de cette fonction, provenant de la structure symétrique des contraintes. Finalement, la Figure 4.20b propose des simulations de l'évolution de la trajectoire de l'état du système bouclé avec la loi de commande explicite, tout en observant les régions traversées par les différentes lois linéaires.

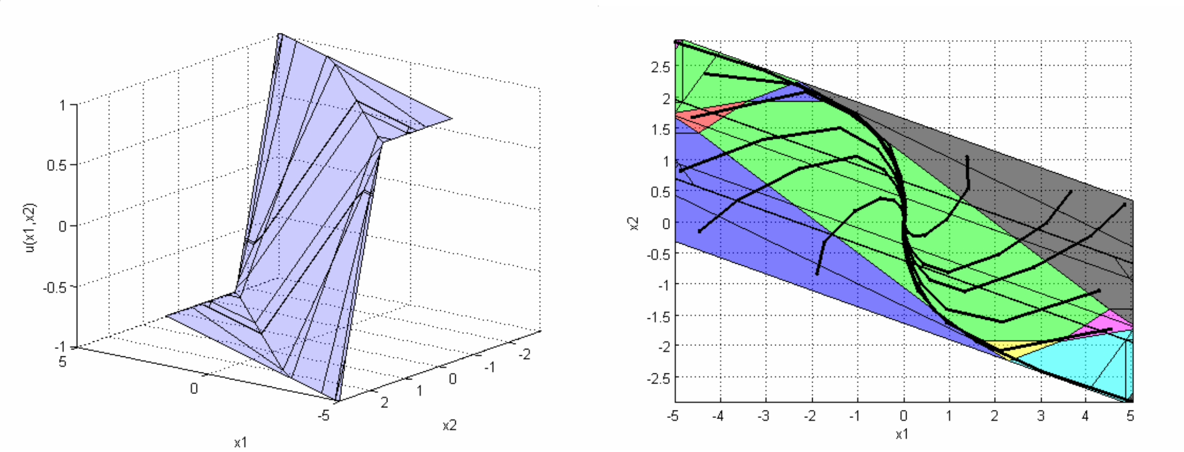


**Figure 4.20 : a) Fonction non linéaire qui incarne d'une façon explicite la loi de commande prédictive**
**b) Evolutions du système dans l'espace d'état pour différents points initiaux**

### *4.3 Améliorations algorithmiques*

La méthodologie présentée jusqu'ici doit se comprendre comme un outil de base pour le développement des solutions explicites. Si les étapes énumérées sont implémentées de façon numériquement fiable, tout problème d'optimisation peut virtuellement se traduire par l'expression de la solution explicite. Néanmoins, pour pouvoir utiliser ces idées de façon pratique lors de la synthèse d'un régulateur prédictif, et ceci dans le cas général, il faut que les performances en terme de programmation soient optimisées, même si le processus se déroule hors ligne et que l'on peut supposer que les ressources en place mémoire et temps de calcul ne sont pas limitées.
Les paragraphes suivants ont pour but, dans cet état d'esprit, d'identifier certains aspects susceptibles d'être modifiés pour améliorer la charge de calcul globale.

#### 4.3.1 Intersection de l'optimum global avec les régions adjacentes au domaine faisable

Dans la procédure '*ExplicitPPPGen',* l'étape analysant l'intersection de l'optimum sans contraintes $\hat{R}$ avec les domaines $DP_{ij}, i = 1,..,\overline{n}_{Dj}$ pour la sélection des zones $Z_{kj}, k = 1,.., n_{\hat{D}_j}$ ( $n_{\hat{D}j} \leq \overline{n}_{Dj}$ ) :

$$Z_{kj} = \left\{ \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \middle| \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \in DP_{ij}; x \in \mathrm{Pr}_x \left( DP_{ij} \cap \hat{R} \right) \right\} \tag{4.80}$$

implique le calcul de toutes les intersections $DP_{ij} \cap \hat{R}$ :

$k = 1$
Pour $i = 1,..,\overline{n}_{Dj}$
Si $DP_{ij} \cap \hat{R} \neq 0$
alors $Z_{kj} = \left\{ \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \middle| \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \in DP_{ij}; x \in \mathrm{Pr}_x \left( DP_{ij} \cap \hat{R} \right) \right\}$
$k = k + 1$
Fin

Ce calcul peut s'avérer coûteux et problématique au niveau du critère d'arrêt. En effet, les intersections successives ne doivent pas être effectuées si toutes les zones $Z_{kj}, k = 1,.., n_{\hat{D}_j}$ ont déjà été obtenues. Plusieurs stratégies de conception des critères d'arrêt existent, l'idée principale étant toujours de vérifier si :

$$\mathrm{Pr}_x \left( DR_j \right) \equiv \mathrm{Pr}_x \left( \bigcup_i Z_{ij} \right) \tag{4.81}$$

Cette condition peut être vérifiée en modifiant lors de chaque intersection l'ensemble $\hat{R}$ de sorte qu'il représente également un indicateur d'arrêt. Pour cela il est remplacé par :

$$\widetilde{R}_j = \hat{R} \cap \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \middle| x \in \mathrm{Pr}_x \left( DR_j \right) \right\} \tag{4.82}$$

et la construction des régions $Z_{kj}, k = 1,.., n_{\hat{D}_j}$ se fait selon la procédure :

$k = 1$
Pour $i = 1,..,\overline{n}_{Dj}$
Si $DP_{ij} \cap \widetilde{R}_j \neq 0$
alors $Z_{kj} = \left\{ \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \middle| \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \in DP_{ij}; x \in \mathrm{Pr}_x \left( DP_{ij} \cap \widetilde{R}_j \right) \right\}$
$\widetilde{R}_j = \widetilde{R}_j - \widetilde{R}_j \cap Z_{kj}$
Si $\widetilde{R}_j \neq 0$
$k = k + 1$

Sinon Fin
Fin

Cette procédure assure l'arrêt une fois la condition (4.81) remplie, mais a le désavantage de transformer l'ensemble $\widetilde{R}_j$ d'un convexe à une union de convexes par l'emploi de l'opération $\widetilde{R}_j = \widetilde{R}_j - \widetilde{R}_j \bigcap Z_{kj}$. Pour éviter les effets négatifs de cette opération, une procédure optimale peut être la suivante.

$k = 1$
Pour $i = 1,..,\overline{n}_{Dj}$
Si $DP_{ij} \bigcap \hat{R} \neq 0$
alors $Z_{kj} = \left\{ \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \middle| \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \in DP_{ij} ; x \in \mathrm{Pr}_x \left( DP_{ij} \bigcap \hat{R} \right) \right\}$
$\widetilde{R}_j = \widetilde{R}_j - \widetilde{R}_j \bigcap Z_{kj}$
Si $\widetilde{R}_j \neq 0$
$k = k + 1$
Sinon Fin
Fin

Dans ce cas les intersections se font en utilisant le lieu géométrique non altéré mais l'ensemble $\widetilde{R}_j$ est utilisé pour éviter le déroulement des opérations une fois que toutes les régions nécessaires ont été construites. Ainsi, en appliquant cette modification pour l'exemple du double régulateur avec contraintes mixtes (exemple 4.6), le nombre d'intersections testées est réduit de 275 à 167.

### 4.3.2 Enumération et traitement des régions adjacentes

La modification présentée ci-dessus s'avère utile pour l'économie du processus de construction, mais on peut toujours imaginer des exemples limites pour lesquels les régions utiles, $Z_{kj}, k = 1,..,n_{\hat{D}_j}$, seraient données par les dernières régions adjacentes à être explorées. Dans ce cas, il est préférable de limiter la création des régions $DP_{ij}$ inutiles.

**Proposition 4.7 :** Soit le problème d'optimisation (4.51), la matrice $H > 0$, $\mathbf{k}_u^*(x)$ le point d'optimum et $\mathbf{k}_u^{sc}(x)$ le point d'optimum en l'absence de contraintes. Si $\mathbf{k}_u^*(x) \neq \mathbf{k}_u^{sc}(x)$, le sous-ensemble des contraintes actives pour le point d'optimum $\mathbf{k}_u^*(x)$ contient au moins une contrainte pour laquelle :

$$A_{in_i}\, \mathbf{k}_u^{sc}(x) < B_{in_i}\, x + b_{in_i} \tag{4.83}$$

**Preuve :** Supposons qu'il existe un point de minimum $\mathbf{k}_u^*(x) \neq \mathbf{k}_u^{sc}(x)$ ne saturant aucune contrainte de (4.83). Mais les équations du type :

$$A_{in_i} \mathbf{k}_u(x) = B_{in_i} x + b_{in_i} \tag{4.84}$$

définissent des hyperplans de séparation entre le point $\mathbf{k}_u^{sc}(x)$ et l'ensemble convexe :

$$P(x) = \left\{ \mathbf{k}_u \middle| A_{in} \mathbf{k}_u \leq b_{in} + B_{in} x ; A_{eq} \mathbf{k}_u = b_{eq} + B_{eq} x \right\} \tag{4.85}$$

Comme $H > 0$ et puisque $\mathbf{k}_u^*(x)$ n'est pas contenu dans un tel hyperplan de séparation, il existe des points $\overline{\mathbf{k}}_u$ :

$$(\mathbf{k}_u^{sc}(x) - \overline{\mathbf{k}}_u)^T H (\mathbf{k}_u^*(x) - \overline{\mathbf{k}}_u) \leq 0 \tag{4.86}$$

et donc le point $\mathbf{k}_u^*(x)$ n'est pas l'optimum car il existe un meilleur candidat. Le résultat est donc prouvé par l'absurde. La proposition peut être également démontrée en utilisant les idées qui assurent la convergence en temps fini des méthodes d'ensemble actif. ■

Comme conséquence de ce résultat, on peut conclure que toutes les zones adjacentes basées exclusivement sur des contraintes :

$$A_{in_i}\,\mathbf{k}_u^{sc}(x) \geq B_{in_i}\,x + b_{in_i} \qquad \forall x \tag{4.87}$$

peuvent être éliminées de l'énumération.

Rechercher les contraintes entrant dans cette catégorie peut être vu comme un problème d'optimisation en lui-même. Si l'on met en œuvre une démarche géométrique, les idées sont les suivantes :

1) Construire $\hat{D}$ comme dans (4.25) et le lieu géométrique de l'optimum sans contraintes $R$ (4.58)

2) Pour chaque inégalité $A_{in_i}\,\mathbf{k}_u(x) \geq B_{in_i}\,x + b_{in_i}$ dans l'ensemble $A_{in}\mathbf{k}_u \geq B_{in}x + b_{in}$

a. Construire $R_i = R \bigcap \left\{ \begin{bmatrix} A_{in_i} & -B_{in_i} \end{bmatrix} \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} > b_{in_i} \right\}$

b. Construire $VP_i = \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \in P \,\middle|\, \begin{bmatrix} A_{in_i} & -B_{in_i} \end{bmatrix} \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} = b_{in_i} \text{ et } x \in \mathrm{Pr}_x\, R_i \right\}$

c. Si $VP_i = 0$, la contrainte $A_{in_i}\,\mathbf{k}_u(x) \geq B_{in_i}\,x + b_{in_i}$ peut être ignorée lors de l'énumération des régions adjacentes au domaine faisable au cours de la construction des solutions explicites pour (4.51). ■

Lors de l'opération d'énumération de zones adjacentes (procédure '*Split*') le découpage autour du domaine faisable peut être réduit en utilisant l'information sur les contraintes qui n'influencent pas la solution optimale. Le fait que le nombre de régions adjacentes soit diminué a des implications sur la charge de calcul du processus complet de création de la solution explicite car certaines zones à inspecter sont évitées.

Un exemple dans cette direction peut être décrit par (4.66). On a vu qu'il est possible de construire la solution explicite en se basant sur les zones régulières et la partition en zones adjacentes. Le résultat est représenté par $3\times 9$ régions qui doivent ensuite être traitées. Si l'on applique la technique décrite précédemment, on obtient alors le sous-ensemble des contraintes qui interviennent effectivement dans la constitution des régions adjacentes. Ainsi 5 contraintes doivent être considérées et peuvent être identifiées Figure 4.21a. On observe les régions $VP_i$ non nulles qui font que la contrainte respective est retenue. Il apparaît clairement que pour un $x$ donné on puisse identifier plusieurs $VP_i$, voir extrémités (pour $x \in [-1{,}5\,;\,-1]$ et $x \in [1\,;\,1{,}5]$). Finalement le fait que seules 5 parmi les 8 contraintes présentées dans la définition du problème soient retenues induit que le jeu des régions à inspecter pour construire la formulation explicite de la solution contient $6+6+8$ domaines.

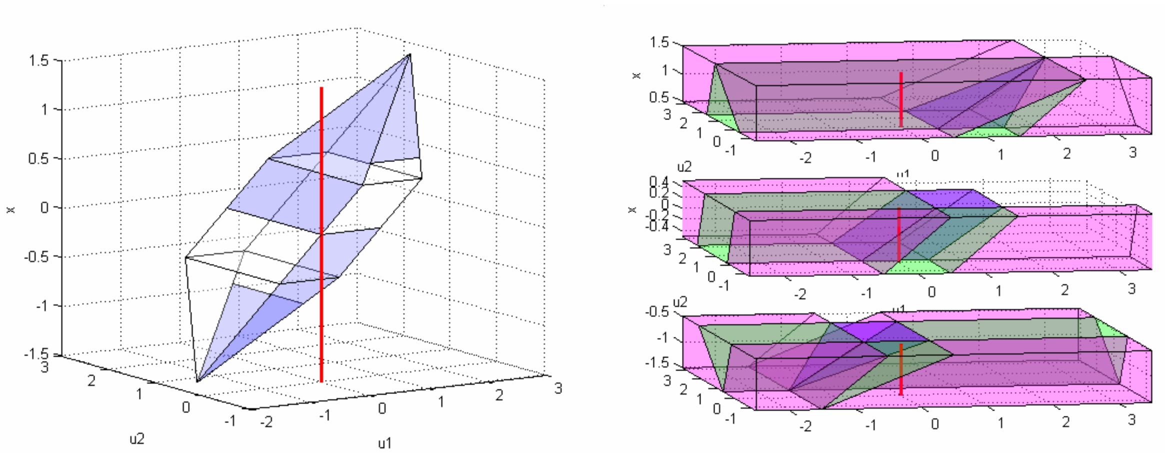


**Figure 4.21 : a) Recherche des contraintes présentes parmi les contraintes actives (contenant des régions en bleu – $VP_i$)**
**b) Partition réduite grâce à l'information apportée par les contraintes mises en évidence**

**Remarque :** Idéalement la charge de calcul liée à la procédure de construction des solutions explicites basée sur les polyèdres paramétrés peut être encore réduite si toutes les contraintes qui n'influencent pas la solution optimale sont éliminées de l'ensemble traité. Une telle procédure nécessite des passages successifs de la représentation de générateurs vers celle de contraintes et s'avère numériquement sensible.

### 4.3.3 Non utilisation des zones de forme régulière

On observe dans la procédure '*ExplicitPPPGen'* que le nombre de régions adjacentes au domaine faisable est en relation directe avec le nombre de zones pour lesquelles le domaine faisable présente une forme régulière. En fait, toutes les partitions correspondant aux formes régulières se retrouvent dans la partition finale pour l'expression de la solution explicite. En conséquence, une amélioration possible peut être envisagée en n'utilisant pas les formes régulières, sachant que les domaines réguliers ont été utilisés pour la généralisation des techniques géométriques appliquées aux ensembles de contraintes non paramétrées. Dans le contexte envisagé ici, si l'on élimine le recours à une prépartition en zones de forme régulière, on obtient une procédure plus efficace du point de vue des opérations géométriques.

***<u>Procédure ExplicitPPPGenAmelioré</u>***

1) Construction de la double représentation :

$$P(x)=\{\mathbf{k}_u | A_{in}\mathbf{k}_u \le b_{in}+B_{in}x; A_{eq}\mathbf{k}_u = b_{eq}+B_{eq}x\}=$$
$$=\left\{\mathbf{k}_u(x) \middle| \mathbf{k}_u(x)=\sum_{i=1}^{d_S}\sigma_i(x)s_i(x)+\sum_{i=1}^{d_R}\rho_i r_i+\sum_{i=1}^{d_L}\lambda_i l_i\right\}, \tag{4.88}$$
$$0\le\sigma_i\le 1,\ \sum_{i=1}^{v}\sigma_i(x)=1,\ \rho_i\ge 0,\ \forall\lambda_i$$

Pour chaque sommet paramétré $s_i(x)$, identifier les domaines de validité $VD_i, i=1,..,n_S$ dans l'espace des paramètres de contexte. L'union de ces domaines :

$$VD=\bigcup_{i=1}^{n_S} VD_i \subset \Re^n \tag{4.89}$$

représente l'ensemble maximal des paramètres faisables.

2) Augmenter $P(x)$ à un polyèdre non paramétré dans un espace de dimension étendu :

$$D=\left\{\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix} \middle| \mathbf{k}_u(x)\in P(x)\right\} \tag{4.90}$$

3) Construction le lieu géométrique de l'optimum sans contraintes :

$$R=\left\{\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\in\Re^{p+n} \middle| \begin{bmatrix}I & H^{-1}F^T\end{bmatrix}\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}=0\right\} \tag{4.91}$$

4) En effectuant un changement de coordonnées, on obtient :

$$D \overset{\hat{\mathbf{k}}_u=H^{1/2}\mathbf{k}_u-H^{-1/2}F^T x}{\rightarrow} \hat{D};\ R \overset{\hat{\mathbf{k}}_u=H^{1/2}\mathbf{k}_u-H^{-1/2}F^T x}{\rightarrow} \hat{R}=\left\{\begin{bmatrix}\hat{\mathbf{k}}_u\\ x\end{bmatrix}\in\Re^{p+n} \middle| \hat{\mathbf{k}}_u=0\right\} \tag{4.92}$$

5) Pour $i=0\cdots n$

i) Générer toutes les combinaisons des $i$ contraintes parmi celles qui décrivent $\hat{D}$

ii) Retenir les combinaisons qui sont satisfaites par au moins $n-i+1$ sommets paramétrés. Il en résulte un partitionnement de la région autour de $\hat{D}$ en :

$$\Re^p \oplus VD=\hat{D}\bigcup_{i=1}^{\bar{n}_D} DP_i \tag{4.93}$$

qui correspond à des régions adjacentes à $\hat{D}$ caractérisées par des projections sur la même frontière de $\hat{D}$.

6) Intersection de $\hat{R}$ avec chaque domaine $DP_i, i = 1,..,\overline{n}_D$ et sélection de toutes les zones $Z_k, k = 1,..,n_{\hat{D}}$ ($n_{\hat{D}} \le \overline{n}_D$) définies par :

$$Z_k = \left\{ \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \middle| \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \notin \bigcup_{l=1}^{k-1} Z_l ; \exists i \; t.q. \begin{bmatrix} \hat{\mathbf{k}}_u \\ x \end{bmatrix} \in DP_i ; x \in \mathrm{Pr}_x \left( DP_i \cap \hat{R} \right) \right\} \tag{4.94}$$

7) $$S\hat{D} = \left(\hat{D} \cap \hat{R}\right) \cup \left(\mathrm{Pr}^x_{\hat{D} \cap Z_1} \hat{R}\right) \cup \ldots \cup \left(\mathrm{Pr}^x_{\hat{D} \cap Z_{n_{\hat{D}}}} \hat{R}\right) \tag{4.95}$$

8) $$S\hat{D} \xrightarrow{\mathbf{k}_u = H^{-1/2}\hat{\mathbf{k}}_u + H^{-1}F^T x} SD \tag{4.96}$$

9) $SD$ peuvent être stockées sous la forme de fonctions affines par morceaux :

$$\mathbf{k}_u(x) = K_{Lin_i} x + K_{tl_i} \quad pour \quad x \in \mathrm{Pr}_x \, SD_i \tag{4.97}$$

*Les avantages* de cette procédure résident dans l'élimination de partitions inutiles pour les zones ayant la même expression linéaire de la solution explicite. Du point de vue charge de calcul, des zones ayant le même type de projection sont traitées une seule fois même si elles se divisent en plusieurs zones régulières.

*Les inconvénients* viennent du fait que la hiérarchisation des découpages dans l'espace des paramètres est perdue, on obtient directement la partition finale ; de plus, lors de la création effective des régions adjacentes, les combinaisons possibles des contraintes sont beaucoup plus nombreuses car on les recherche dans l'ensemble complet et non dans les sous-ensembles non redondants par morceaux. Ainsi, le nombre de combinaisons pour la structure initiale, comparé avec cette solution modifiée, s'exprime par :

$$\sum_{k=1}^{n_{DR}} \sum_{i=0}^{n} C^i_{r_k} \quad vs. \quad \sum_{i=0}^{n} C^i_r \tag{4.98}$$

avec $n$ dimension de l'espace des paramètres, $r$ nombre de contraintes dans l'ensemble initial, $n_{DR}$ nombre de régions de forme régulière, $r_k$ nombre de contraintes non redondantes pour chacune de ces régions. Cependant les sommets paramétrés peuvent être utilisés pour discerner les combinaisons inutiles.

Reprenons pour mesurer l'impact de la modification des procédures de développement de la solution explicite les exemples utilisés tout au long de ce paragraphe. Pour l'exemple 4.5, le partitionnement de l'espace étendu (Figure 4.22a) contient maintenant 21 vs. $3\times 9$ polyèdres dans la version initiale, ou 15 vs. 6+6+8 si l'on considère la procédure de traitement (4.83). Après intersection avec le lieu géométrique de l'optimum sans contraintes, 5 régions sont retenues, par rapport à 6 dans la version initiale (Figure 4.22b). Finalement la solution explicite est donnée Figure 4.22c.

Les améliorations sont plus claires pour l'exemple 4.6 d'une loi prédictive explicite pour le système double intégrateur. Dans la version initiale avec énumération et traitement des régions adjacentes, on dénombrait 27 régions de forme régulière et ensuite un total de 275 régions adjacentes au domaine faisable. La solution explicite était décrite par 55 domaines dans l'espace des paramètres. Si l'on adopte la technique d'énumération des régions en ignorant les formes régulières, on aboutit à 33 régions adjacentes et 15 partitions pour l'expression de la solution explicite (Figure 4.23a). La fonction linéaire par morceaux (Figure 4.23b) est similaire à celle de la Figure 4.19a, de même pour les trajectoires du système bouclé (Figure 4.23c). Le temps nécessaire pour construire la solution est réduit de 2,8s à 0,8s. Si l'on ajoute le fait que les 27 zones de forme régulière nécessitent à leur tour un temps de construction d'environ 3,7s l'amélioration du temps de calcul est évidente. Toutefois on perd un aperçu de la topologie du domaine faisable ainsi que le contrôle de la redondance locale des contraintes.

Le choix entre l'utilisation ou non des zones régulières est en fait sensible et dépend fondamentalement du degré de redondance pour chaque région. L'analyse précédente a montré une augmentation possible du nombre de régions lorsque l'on traite la redondance (prise en compte des formes régulières). Pour des classes de systèmes particulières (formes des contraintes éventuellement séparables) cependant, cette augmentation peut déboucher sur un gain lors des étapes de positionnement. Cet aspect sera abordé de nouveau au paragraphe 4.5.

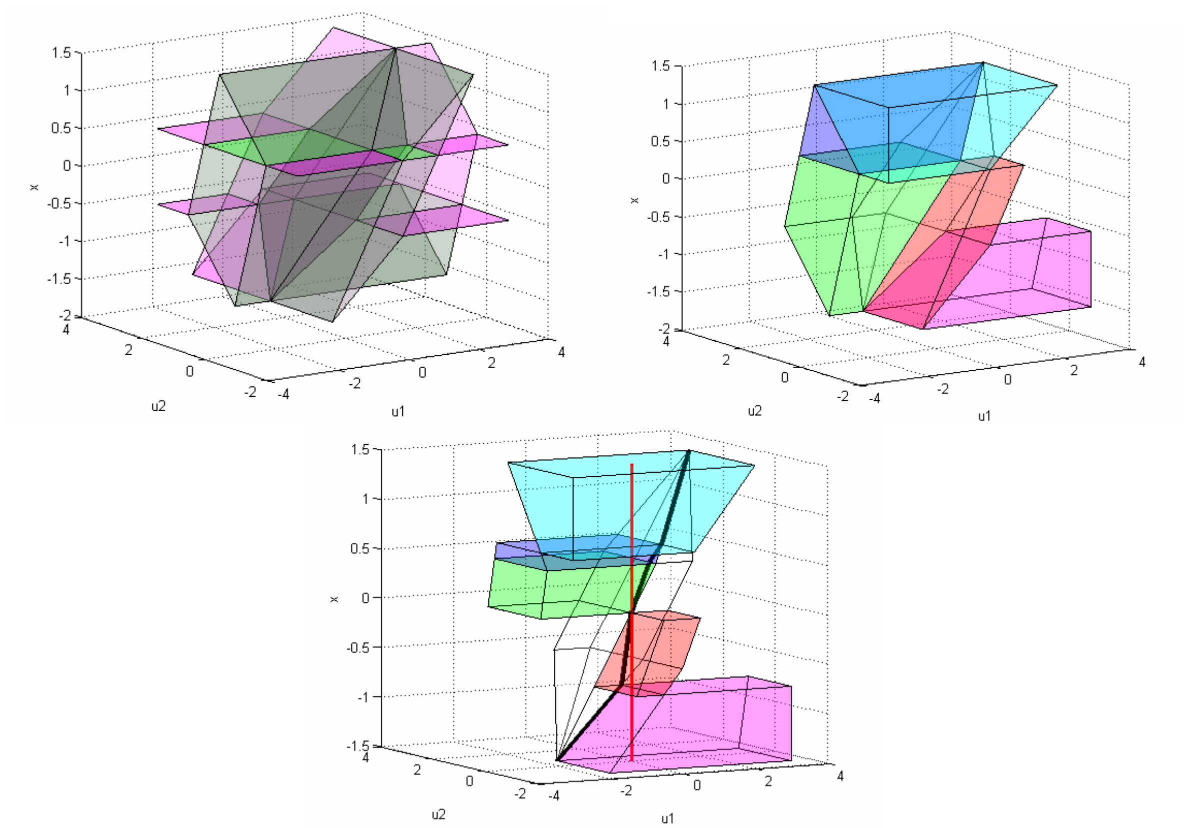



**Figure 4.22 : a) Partitionnement de l'espace étendu en régions ayant une projection convexe sur le domaine faisable**
**b) Sélection des zones ayant une intersection avec le lieu géométrique de l'optimum sans contraintes**
**c) Solution explicite sous forme d'une union de 5 ensembles convexes**

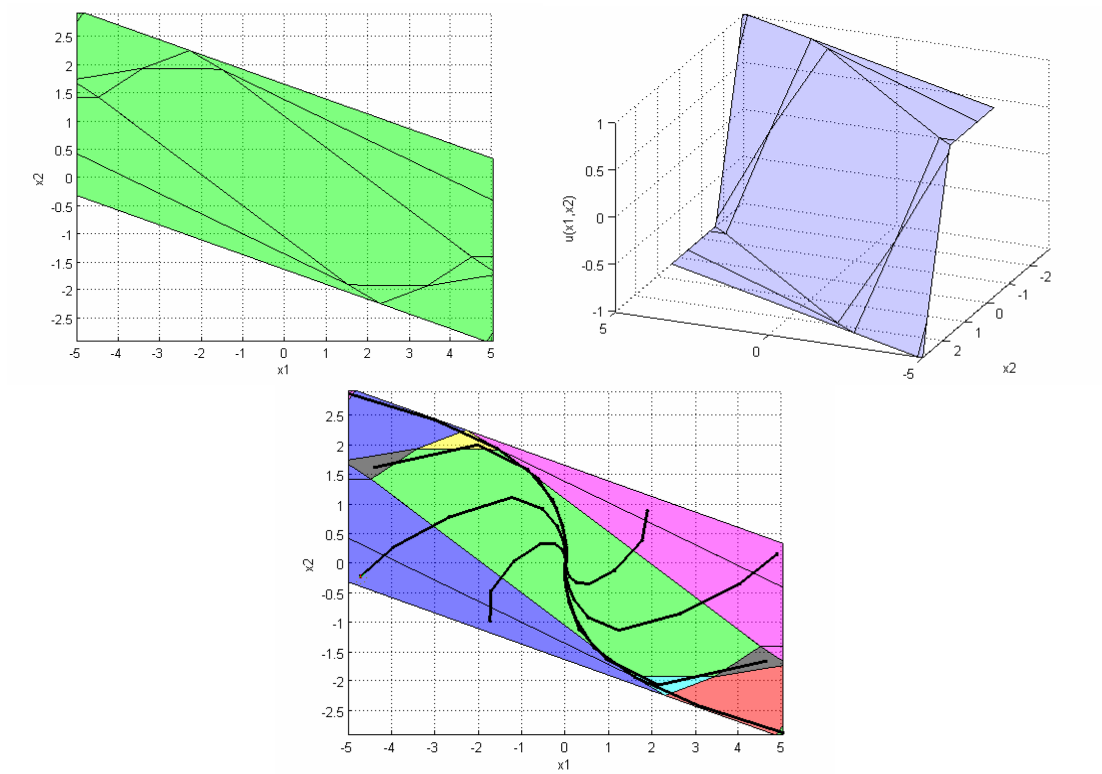



**Figure 4.23 : a) Partitionnement de l'espace des paramètres correspondant à la solution explicite du problème d'optimisation associé au double intégrateur**
**b) Loi de commande prédictive – formulation explicite**
**c) Différentes trajectoires pour le système bouclé avec la loi prédictive**

### 4.3.4 Méthodes alternatives de constructions des solutions explicites

La construction des solutions explicites pour les problèmes d'optimisation multiparamétriques est depuis peu l'objet d'attentions particulières [BMDP02], [SGD02] en ce qui concerne la recherche de formulations pouvant améliorer les performances des lois de commande prédictives sous contraintes. L'approche qui a été présentée dans le cadre de ce travail est une approche exclusivement géométrique basée sur le concept de polyèdre paramétré. L'objet de ce paragraphe est de comparer la stratégie développée aux autres techniques recensées dans la littérature.

On trouve effectivement dans la littérature certaines méthodes alternatives utilisant également la géométrie informatique, mais aussi des propriétés algébriques du problème d'optimisation, ou encore le principe de programmation dynamique liée à l'optimalité de la loi de commande résultante. Pratiquement ces méthodes diffèrent dans la façon d'explorer l'espace des paramètres, ainsi que dans les outils employés. Les développements ci-dessous ont pour but de décrire très succinctement les techniques marquantes, en mentionnant les avantages et inconvénients par rapport à la méthode basée sur les polyèdres paramètres. Il faut être conscient malgré tout que la séparation n'est pas toujours franche, car par exemple il est possible d'utiliser des méthodes géométriques pour éliminer la redondance locale des contraintes et poursuivre ensuite avec des méthodes algébriques pour explorer chaque partition.

Dans un but de comparaison entre l'approche élaborée dans ce chapitre et les autres stratégies existantes, considérons comme indicateur le nombre de partitions dans l'espace des paramètres, qui dépend du nombre de paramètres – $n$, du nombre de contraintes – $r$, et indirectement du nombre de degrés de liberté du problème d'optimisation, équivalent à la dimension de $\mathbf{k}_u$ diminué du nombre de contraintes égalité. Si $NZ$ représente le nombre de partitions, une borne supérieure en est donnée par :

$$NZ \leq \sum_{i=0}^{n} C_r^i$$

Alors, la méthode développée basée sur l'utilisation des polyèdres paramétrés va produire :

$$NZ_1 = \sum_{k=1}^{n_{DR}} \sum_{i=0}^{n} C_{r_k}^i \leq NZ$$

en utilisant les formes régulières et si elles sont ignorées :

$$NZ_2 = NZ \leq \sum_{i=0}^{n} C_r^i$$

**Méthodes basées sur les conditions KKT – [BMDP02]**

En partant de la formulation (4.38) et en appliquant une transformation de type (4.25), la paramétrisation de la fonction de coût est transférée exclusivement vers la paramétrisation des contraintes. Ainsi pour le problème :

$$\begin{aligned} &\min_{\mathbf{k}_u} 0{,}5\mathbf{k}_u^T H \mathbf{k}_u \\ &\text{sujet à : } A_{\text{in}}\mathbf{k}_u \leq b_{in} + B_{in}x \end{aligned} \tag{4.99}$$

la méthode part de la caractérisation de la solution en un point spécifique $\mathbf{k}_u^*(x_0)$ et avec les conditions KKT :

$$\begin{aligned} &H\mathbf{k}_u + A_{in}^T \boldsymbol{\lambda} = 0; \boldsymbol{\lambda} \in \Re^r \\ &\boldsymbol{\lambda} \geq 0 \\ &\boldsymbol{\lambda}^T (A_{in}\mathbf{k}_u - b_{in} - B_{in}x) = 0 \\ &A_{in}\mathbf{k}_u \leq b_{in} + B_{in}x \end{aligned} \tag{4.100}$$

obtient la caractérisation locale de la solution optimale au voisinage de $x_0$ :

$$CR_0 : \begin{cases} A_{in}H^{-1}\overline{A}_{in}^T(\overline{A}_{in}H^{-1}\overline{A}_{in}^T)^{-1}(\overline{b}_{in} + \overline{B}_{in}x) \leq b_{in} + B_{in}x \\ (\overline{A}_{in}H^{-1}\overline{A}_{in}^T)^{-1}(\overline{b}_{in} + \overline{B}_{in}x) \leq 0 \end{cases} \tag{4.101}$$

avec $\overline{A}_{in}, \overline{b}_{in}, \overline{B}_{in}$ identifiant les inégalités saturées pour $\{x_0; \mathbf{k}_u^*(x_0)\}$. Une fois cette région critique non vide déterminée, le reste de l'espace des paramètres est exploré en quête de nouvelles régions faisables. L'exploration proposée dans [BMDP02] utilise une méthode introduite par [DP00] qui inverse récursivement les inégalités décrivant $CR_0$ pour définir de nouveaux domaines. Ensuite, pour chacun de ces nouveaux domaines, il est nécessaire de trouver un point initial, résoudre le problème QP associé et retrouver les contraintes actives pour pouvoir utiliser (4.101).

Comme caractéristiques liées à l'implémentation, la méthode travaille avec une représentation de l'espace faisable exclusivement en termes de contraintes, les solutions sont données en considérant un ensemble borné de l'espace des paramètres ('*bounding box*'). Une attention toute particulière doit être portée sur les problèmes de dégénérescence (le fait que la matrice $\overline{A}_{in}$ puisse avoir des lignes linéairement dépendantes), mais aussi sur la présence de redondance dans l'ensemble de contraintes (4.101). Un des inconvénients de cette méthode est que, pendant l'exploration, l'inversion des inégalités de (4.101) peut provoquer des partitions superflues dans l'espace des paramètres, comparables aux découpages inutiles de la procédure *ExplicitPPP* qui utilise les zones régulières du domaine faisable. La solution consiste à recourir à l'union des régions ayant la même solution, dans la mesure où cette union reste un ensemble convexe (des procédures sont données dans [BFT02]). La complexité des partitions est évaluée par la borne supérieure [BOR03], [BMDP02] :

$$NZ \leq \sum_{i=0}^{2^r-1} i!\, r^i \tag{4.102}$$

La méthode décrite par [BMDP02] peut être améliorée en ce qui concerne l'exploration de l'espace des paramètres par les techniques présentées dans [TJB03a]. La principale contribution est liée au fait que les contraintes actives pour une région peuvent être déterminées en exploitant le résultat suivant.

**Proposition 4.8 :** Soit deux ensembles $\Im_1, \Im_2$ formés des indices des contraintes saturées de :

$$A_{\mathrm{in}}\mathbf{k}_u \leq b_{in} + B_{in}x$$

pour (4.99) et $CR_1, CR_2$ les régions critiques correspondantes dans l'espace des paramètres. Si l'on suppose que les ensembles des contraintes définis par $\Im_1, \Im_2$ ne sont pas dégénérés et que les régions $CR_1, CR_2$ sont voisines, alors $\Im_1 \subset \Im_2$ et $card.\,\Im_1 = card.\,\Im_2 -1$ ou $\Im_1 \supset \Im_2$ et $card.\,\Im_1 = card.\,\Im_2 +1$. ■

L'avantage est que la résolution des problèmes d'optimisation quadratique n'est plus nécessaire pour obtenir le jeu des contraintes actives.

Dans [Bao02] la stratégie de [BMDP02] est employée avec une partition de l'espace des paramètres modifiée, de sorte que les domaines proches d'une région déjà explorée sont identifiés selon les directions faisables perpendiculaires aux faces de cette région critique. L'ensemble des contraintes actives est déterminé en résolvant un problème QP. L'avantage de cette stratégie est d'éviter l'introduction de partitions superflues.

**Méthodes basées sur la structure géométrique**

Mentionnons dans cette catégorie les travaux de [DeD00], [SGD02], basés sur les idées résumées au début du paragraphe 4.2.1. Il s'agit d'exploiter la structure géométrique du problème d'optimisation et en travaillant dans l'espace des arguments, de déterminer une partition induisant une partition de l'espace de paramètres. Pour les problèmes mpQP et des systèmes des contraintes non affectés par la paramétrisation, l'aperçu géométrique offre un apport d'information, mais la généralisation aux contraintes affectées par un vecteur des paramètres transforme la méthode en une technique fortement liée à l'énumération des combinaison possibles des contraintes actives.
Du point de vue implémentation, les méthodes [SGD02] utilisent la représentation implicite des domaines faisables et identifient les régions adjacentes en se basant sur les perpendiculaires aux faces (ce qui s'avère pratique pour les saturations de commande par exemple).
La construction des solutions explicites à partir de la double représentation des polyèdres peut être incluse dans cette catégorie de méthodes, mais alors les sommets paramétrés peuvent être utilisés pour éviter l'énumération de toutes les combinaisons possibles des contraintes actives.

**Méthodes basées sur la programmation dynamique**

Les méthodes utilisant la programmation dynamique pour élaborer la solution explicite construisent en fait itérativement des solutions explicites pour des problèmes de taille plus petite. Ces approches peuvent améliorer significativement la charge de calcul pour la commande prédictive, mais cet avantage ne se retrouve pas dans le cas général d'un problème d'optimisation multiparamétrique.
La méthode consiste à utiliser des évaluations partielles de la fonction de coût, se basant sur le principe d'optimalité [Bel57]. Ce principe stipule que toute portion d'une trajectoire optimale est elle-même optimale, et par conséquent les formulations explicites partielles obtenues pour des horizons inférieurs à celui choisi pour construire la forme analytique globale peuvent être employées. Même si le caractère linéaire par morceaux de cette fonction est facile à discerner, la construction effective pour des horizons de prédiction importants est prohibitive. Néanmoins, même si ces méthodes sont difficiles à assimiler, les stratégies basées sur la programmation dynamique peuvent être associés aux stratégies géométriques [GSD04] ou à celles utilisant les conditions KKT [PABC04] permettant d'améliorer le temps de calcul.

**Conclusion**

La construction de solutions explicites peut être également associée à la construction d'ensembles contrôlables, d'ensembles invariants ou à l'analyse de l'atteignabilité, pour fournir un instrument complet de synthèse des lois de commande prédictives. L'outil le plus complexe et probablement le plus performant est la bibliothèque MPT, développé à ETH Zurich [KGB04]. Il intègre des modules performants d'optimisation (YALMIP) et du point de vue géométrique utilise des procédures ESP (Equality Set Projection) développées à l'Université de Cambridge [JKM04]. A part cet outil puissant, des recherches sur des sujets similaires existent à l'Université de Newcastle en Australie et à l'Imperial College de Londres autour de la bibliothèque de calcul GBT [GBT05]. Les travaux liés à ce mémoire ont donné naissance à un noyau de calcul polyédral utilisable pour le développement de solutions explicites – PPP (Predictive control by Parameterized Polyhedra).

Finalement, toutes les méthodes utilisent d'une façon ou d'une autre des arguments géométriques et s'avèrent sensibles numériquement lors de certaines opérations. Par exemple, la projection, très importante lors de la construction des solutions explicites, s'implémente de façon immédiate si l'on possède une double représentation (la qualité repose alors sur la précision de la double représentation). Mais si l'on utilise une représentation par contraintes, il faut dans ce cas recourir aux méthodes de type Fourier-Motzkin (avec une robustesse numérique faible) ou des méthodes alternatives [JKM04]. En conclusion, la robustesse numérique des solutions explicites reste influencée par la qualité des procédures de calcul géométrique utilisées.

## *4.4 Analyse des solutions explicites*

Les paragraphes précédents ont détaillé l'obtention de la solution explicite du problème (4.38). A partir de ces algorithmes, des résultats ont été établis quant à la structure de la fonction $\mathbf{k}_u^*(x)$ (4.65). Ce paragraphe propose d'aller un peu plus loin en analysant les propriétés de la loi prédictive sous sa forme explicite, à partir notamment des caractéristiques de continuité et de convexité de la fonction de coût (4.38). De plus, ce paragraphe se penche aussi sur les aspects pratiques liés à l'implémentation, en particulier sur le mécanisme de positionnement et les besoins en ressource mémoire.

### 4.4.1 Propriétés de la loi de commande

Les résultats sur la convexité et la continuité des solutions explicites sont résumés par le théorème suivant.

> **Théorème 4.9 :** Soit un problème d'optimisation quadratique multiparamétrique :
>
> $$\min_{\mathbf{k}_u} J_t = \mathbf{k}_u^T H \mathbf{k}_u + 2x^T F \mathbf{k}_u + x^T G x \tag{4.103}$$
>
> $$\text{sous les contraintes} : \begin{cases} A_{in}\mathbf{k}_u \leq b_{in} + B_{in}x \\ A_{eq}\mathbf{k}_u = b_{eq} + B_{eq}x \end{cases} \tag{4.104}$$
>
> avec $H > 0; \mathbf{k}_u \in \Re^p; x \in \Re^n$ et pour lequel l'ensemble des paramètres faisables est $VD \subset \Re^n$. La fonction explicite qui représente l'argument optimal de (4.103-4.104) : $\mathbf{k}_u^*(x) : VD \to \Re^p$ est continue et linéaire affine par sous-domaines polyédraux dans $VD$.

La solution optimale :

$$J^*(x) = \mathbf{k}_u^{*T} H \mathbf{k}_u^* + 2x^T F \mathbf{k}_u^* + x^T G x \tag{4.105}$$

est continue, quadratique par sous-domaines polyédraux dans $VD$ et convexe.

**Preuve :**

*Ensemble faisable convexe :* Avant de montrer les principales propriétés énoncées, analysons la structure de l'ensemble des paramètres faisables $VD \subset \Re^n$. Pour $x_1, x_2 \in VD$ il existe $\mathbf{k}_u^*(x_1), \mathbf{k}_u^*(x_2)$ tels que (4.104) soient satisfaites. $\forall \alpha \in [0,1]$ on peut définir $x_\alpha = \alpha x_1 + (1-\alpha)x_2$ induisant un candidat faisable pour (4.104) en considérant $\mathbf{k}_u(x_\alpha) = \alpha \mathbf{k}_u^*(x_1) + (1-\alpha)\mathbf{k}_u^*(x_2)$, ce qui montre que $x_\alpha \in VD$ et donc $VD$ est convexe. Le même résultat peut être obtenu avec des arguments géométriques car $VD$ représente une projection du domaine polyédral décrit par les contraintes de (4.104) dans l'espace étendu $\Re^{p+n}$ sur le sous-espace des paramètres $\Re^n$.

*Optimum linéairement affine sur des polyèdres :* Soit la solution optimale décrite géométriquement par une union d'ensembles convexes $SD = SD_0 \cup \ldots \cup SD_{n_D} \equiv (D \cap R) \cup \left(\mathrm{Pr}^x_{D \cap Z_1} R\right) \cup \ldots \cup \left(\mathrm{Pr}^x_{D \cap Z_{n_D}} R\right) \subset \Re^{p+n}$. Il faut observer que chaque composante polyédrale convexe, générée soit par $\left(\hat{D}_j \cap \hat{R}\right)$ soit par $\left(\mathrm{Pr}^x_{\hat{D} \cap Z_i} \hat{R}\right)$ est de dimension *n* (la dimension de $\hat{R}$) même si issue de l'espace $\Re^{p+n}$. La conséquence de cette propriété est que pour chaque ensemble $SD_i$ l'ensemble des contraintes contient *p* égalités linéairement indépendantes :

$$\overline{A}\mathbf{k}_u^* + \overline{B}x = \overline{b} \tag{4.106}$$

En exprimant algébriquement cette propriété, on peut écrire que :

$$\mathbf{k}_u^*(x) = \begin{cases} \ldots \\ K_{Lin_i} x + K_{tl_i} \quad \textit{pour} \quad x \in CR_i \\ \ldots \end{cases} \tag{4.107}$$

où en utilisant la pseudo inverse $K_{Lin_i} = -\overline{A}^{\perp}\overline{B};\, K_{tl_i} = \overline{A}^{\perp}\overline{b}$ et les régions critiques sont données par :

$$CR_i = \mathrm{Pr}_x\, SD_i\,. \tag{4.108}$$

On a ainsi démontré le caractère linéaire affine par sous-domaines polyédraux de la fonction explicite $\mathbf{k}_u^*(x)$. Cette propriété peut être également mise en évidence en ayant recours aux conditions d'optimalité décrites au paragraphe 2.4 (relation (2.74)). Quant aux régions critiques polyédrales, leurs structures peuvent être justifiées aussi à partir des conditions KKT (4.100).

*Continuité de la solution explicite :* La solution explicite de (4.103-4.104) $\mathbf{k}_u^*(x)$ est définie par sous-domaines polyédraux (régions critiques - $CR_i$). Ces domaines polyédraux sont fermés et contiennent des frontières. Ainsi pour deux régions voisines, il existe une frontière commune. Comme $H > 0$ pour (4.103), l'optimum pour chaque point sur la frontière est unique et en conséquence la solution explicite $\mathbf{k}_u^*(x)$ doit être continue tout au long de la frontière. En généralisant à toutes les régions dans la description (4.107), la continuité de la solution explicite globale est assurée.

*Structure et continuité de la fonction de coût optimale :* Pour analyser la structure de la fonction de coût, il suffit de reporter la fonction (4.107) donnant la solution explicite dans la fonction de coût (4.103). Il en résulte :

$$J^*(x) = \begin{cases} \ldots \\ x^T(0.5K_{Lin_i}^T H K_{Lin_i} + F K_{Lin_i} + G)x + (K_{tl_i}^T H K_{Lin_i} + K_{tl_i}^T F^T)x + 0.5K_{tl_i}^T H K_{tl_i} \\ \qquad \text{pour } x \in CR_i \\ \ldots \end{cases} \tag{4.109}$$

qui est une fonction quadratique par morceaux. La continuité est une conséquence triviale du fait que la composition de fonctions continues préserve cette propriété.

*Convexité de la fonction de coût optimale :* Pour démontrer la convexité, on considère de nouveau $x_1, x_2 \in VD$ et pour $\forall \alpha \in [0,1]$ on définit $x_\alpha = \alpha x_1 + (1-\alpha) x_2$. Par optimalité de $J^*(x_\alpha)$ il vient :

$$J^*(x_\alpha) \le \mathbf{k}_u^T(x_\alpha) H \mathbf{k}_u(x_\alpha) + 2x_\alpha^T F \mathbf{k}_u(x_\alpha) + x_\alpha^T G x_\alpha$$

et ensuite après quelques manipulations :

$$\begin{aligned} & J^*(x_\alpha) - (\alpha J^*(x_1) + (1-\alpha) J^*(x_2)) \le \\ & \le -\alpha(1-\alpha)\Big\{ [\mathbf{k}_u(x_1) - \mathbf{k}_u(x_2)]^T H [\mathbf{k}_u(x_1) - \mathbf{k}_u(x_2)] + \\ & \quad + 2[x_1 - x_2]^T F [\mathbf{k}_u(x_1) - \mathbf{k}_u(x_2)] + [x_1 - x_2]^T G [x_1 - x_2] \Big\} \le 0 \end{aligned} \tag{4.110}$$

Le fait que (4.110) soit vérifié pour $\forall x_1, x_2 \in VD, \forall \alpha \in [0,1]$ prouve la convexité de la fonction de coût optimale. ■

Le théorème précédent montre donc que la fonction de coût, qui est continue, convexe et quadratique par morceaux, peut servir comme fonction de Lyapunov pour démontrer la stabilité asymptotique sous certaines conditions (voir chapitre 5). En ce qui concerne la loi de commande prédictive basée sur une solution explicite d'un problème d'optimisation multiparamétrique, on a vu que seule la première composante est appliquée :

$$u_t^{MPC} : \Re^n \to \Re^m ; u^{MPC}(x_t) = [\mathbf{1} \ 0 \ \cdots \ 0] \mathbf{k}_u(x_t) \tag{4.111}$$

de sorte qu'elle hérite des propriétés de continuité et de dépendance linéaire en le vecteur des paramètres.

**Remarque :** Il est intéressant d'observer que les propriétés de continuité de la fonction de coût optimale peuvent être étendues sous certaines conditions. En effet, la notion de différentiabilité de la fonction de coût $J^*(x)$ peut être garantie si le problème (4.103-4.104) est non-dégénéré [Bor] (i.e. les contraintes actives sont linéairement indépendantes pour toute valeur de paramètre). Pour illustrer ce fait, observons la fonction de coût optimale de l'exemple 4.4 pour lequel on a obtenu la solution explicite (Figure 4.24a). Comme les ensembles des contraintes actives ne laissent pas présager de dégénérescence, la fonction est de classe $C^1$. Un exemple de problème pour lequel ces hypothèses ne sont pas remplies est proposé dans [GAL94] :

$$\begin{aligned} & \min_{\mathbf{k}_u} J_t = \mathbf{k}_u^T \begin{bmatrix} 3 & 3 & -1 \\ 3 & 11 & 23 \\ -1 & 23 & 75 \end{bmatrix} \mathbf{k}_u \\ & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix} \mathbf{k}_u = \begin{bmatrix} 0 \\ 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ & \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{k}_u \le \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{4.112}$$

Dans ce cas, la dérivée présente un point de discontinuité en $x = 3$ (Figure 4.24b).

### 4.4.2 Cas de contraintes symétriques

Les contraintes imposées aux séquences de commandes prédites sont souvent symétriques. Comme la fonction de coût jouit de la même propriété de symétrie, le problème d'optimisation quadratique :

$$\begin{aligned} & \min_{\mathbf{k}_u} J_t = \mathbf{k}_u^T H \mathbf{k}_u + x^T F \mathbf{k}_u \\ & \text{sous les contraintes} : \begin{cases} -b_{in} + B_{in} x \le A_{in} \mathbf{k}_u \le b_{in} + B_{in} x \\ A_{eq} \mathbf{k}_u = B_{eq} x \end{cases} \end{aligned} \tag{4.113}$$

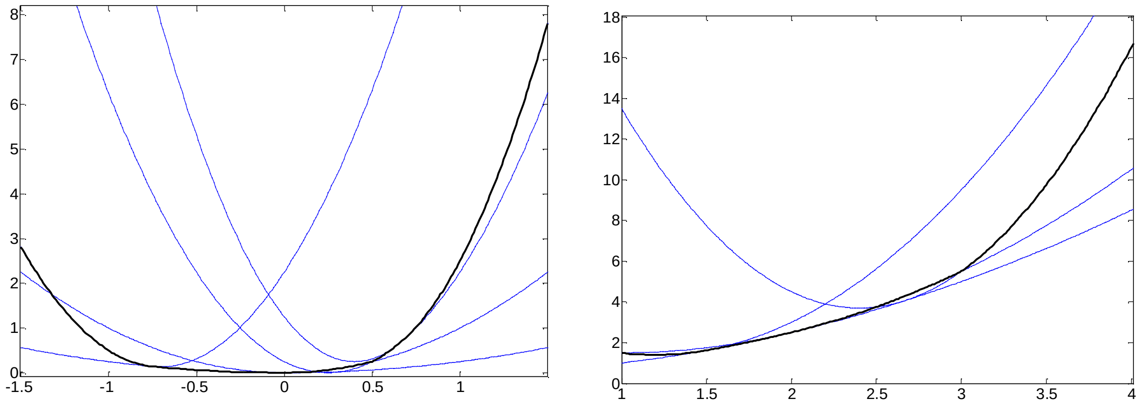



**Figure 4.24 : a) Fonction de coût optimale (en noir) pour l'exemple 4.4**
**b) Fonction de coût optimale correspondant à (4.112)**

peut exploiter sa structure pour réduire la complexité du processus de construction de la solution explicite. En effet le domaine faisable :

$$P(x)=\left\{\mathbf{k}_u \in \Re^p \,\middle|\, -b_{in}+B_{in}x \le A_{in}\mathbf{k}_u \le b_{in}+B_{in}x;\ A_{eq}\mathbf{k}_u = B_{eq}x\right\} \tag{4.115}$$

a la forme d'un polyèdre non paramétré dans l'espace étendu :

$$D=\left\{\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\in \Re^{p+n} \,\middle|\, -b_{in} \le \begin{bmatrix}A_{in} & -B_{in}\end{bmatrix}\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\mathbf{k}_u \le b_{in};\begin{bmatrix}A_{eq} & -B_{eq}\end{bmatrix}\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}=0\right\} \tag{4.116}$$

qui présente une symétrie par rapport à l'origine car :

$$\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\in D \Rightarrow \begin{bmatrix}-\mathbf{k}_u\\ -x\end{bmatrix}\in D \tag{4.117}$$

Ainsi les régions ayant une projection sur la même frontière vont respecter la symétrie et comme le lieu géométrique de l'optimum sans contraintes a la forme :

$$R=\left\{\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\in \Re^{p+n} \,\middle|\, \mathbf{k}_u = -H^{-1}F^T x\right\} \Rightarrow \begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\in R \Rightarrow \begin{bmatrix}-\mathbf{k}_u\\ -x\end{bmatrix}\in R \tag{4.118}$$

les partitions correspondant à la solution explicite auront une symétrie dans l'espace des paramètres. Pratiquement, lors de la construction de la solution explicite de (4.113), il suffit d'analyser le positionnement par rapport à $D$ de :

$$R_+=\left\{\begin{bmatrix}\mathbf{k}_u\\ x\end{bmatrix}\in \Re^{p+n} \,\middle|\, x \ge 0\right\}$$

La forme finale linéaire affine par morceaux de (4.113) est résumée par le théorème suivant, version généralisée de celle présentée dans [Ton03].

**Théorème 4.10 :** Si toutes les contraintes d'un problème d'optimisation quadratique multiparamétrique de type (4.113) sont symétriques par rapport à l'origine dans l'espace étendu $\Re^{p+n}$, la solution optimale $\mathbf{k}_u^*(x)$ est symétrique par rapport à l'origine dans $\Re^n$.

**Preuve :** Considérons un point faisable $x_0 \in \Re^n$ et la solution optimale correspondante $\mathbf{k}_u^*(x_0)$. Soit :

$$\bar{A}\mathbf{k}_u^*(x_0)=\bar{b}+\bar{B}x_0 \tag{4.119}$$

le sous-ensemble des contraintes actives pour la paire $\{\mathbf{k}_u^*(x_0), x_0\}$, tel que :

$$\mathbf{k}_u^*(x_0)=\left[(H^{-1}\bar{A}^T(\bar{A}H^{-1}\bar{A}^T)^{-1}\bar{A}H^{-1}-H^{-1})F^T+H^{-1}\bar{A}^T(\bar{A}H^{-1}\bar{A}^T)^{-1}\bar{B}\right]x_0+ \\ +H^{-1}\bar{A}^T(\bar{A}H^{-1}\bar{A}^T)^{-1}\bar{b} \tag{4.120}$$

Pour $x = -x_0$ l'ensemble des contraintes actives est donné par $\left\{-\overline{A};\overline{b};-\overline{B}\right\}$ et donc :

$$\mathbf{k}_u^*(-x_0) = \begin{bmatrix} (H^{-1}(-\overline{A}^T)((-\overline{A})H^{-1}(-\overline{A})^T)^{-1}(-\overline{A})H^{-1} - H^{-1})F^T + \\ \qquad + H^{-1}(-\overline{A})^T((-\overline{A})H^{-1}(-\overline{A})^T)^{-1}(-\overline{B}) \end{bmatrix}(-x_0) \\ + H^{-1}(-\overline{A})^T((-\overline{A})H^{-1}(-\overline{A})^T)^{-1}\overline{b} = -\mathbf{k}_u^*(x_0) \tag{4.121}$$

### 4.4.3 Implantation de la formulation explicite de la commande prédictive. Complexité en ligne.

Les développements exposés jusqu'à ce point ont permis l'expression explicite des lois prédictives, comme fonctions linéaires par morceaux. Pratiquement, cette expression globale de la loi prédictive est précalculée hors ligne et stockée dans un tableau ('*look-up table*') contenant comme index les régions polyédrales dans l'espace des paramètres et comme information correspondante les fonctions linéaires associées.

A l'aide de cette *table* l'implémentation en ligne de la commande prédictive explicite doit suivre les étapes suivantes :

1) Mesurer (ou estimer) le vecteur des paramètres courant $x$
2) Si $x \notin VD$, arrêter la procédure, le problème est infaisable. Faire appel à des méthodes de récupération de la faisabilité.
3) Rechercher dans la *table* la région critique contenant x.
4) Implémenter la loi linéaire correspondant à cette région active.

Le rôle de l'index dans la table est de permettre une association rapide des valeurs recherchées, à savoir ici les commandes. Le vrai problème soulevé lors de la navigation parmi ces régions est lié au fait que l'on ne dispose pas d'une opération d'ordre de sorte qu'à chaque pas d'échantillonnage il est nécessaire d'identifier 'le domaine courant' par une procédure itérative. Cette identification se fait donc en ligne et peut engendrer une charge informatique importante si le nombre des partitions est élevé. Parmi les méthodes possibles, citons :

- La recherche séquentielle de l'appartenance du vecteur des paramètres courant dans toutes les régions critiques. Cette démarche implique la vérification des contraintes définissant chaque domaine polyédral, d'où un nombre important d'opérations élémentaires et dans le pire cas une inspection de toutes les régions. Cette technique, naïve, peut être améliorée vis-à-vis du nombre d'opérations à effectuer mais aussi de la place mémoire requise.
- La recherche séquentielle parmi toutes les régions critiques implique l'évaluation à plusieurs reprises des contraintes qui définissent les frontières, car on les retrouve dans les représentations implicites des régions voisines même si leur signe est opposé. Une méthode alternative consiste à regrouper toutes les contraintes et s'assurer que leur évaluation n'ait lieu qu'une seule fois pour le point courant. Ensuite décider à quelle région appartient le point courant se fait toujours séquentiellement mais la double vérification des contraintes est évitée. Cette méthode présente aussi l'avantage de ne pas stocker plusieurs fois les contraintes pour des entrées différentes du tableau. Elle reste améliorable car dans le pire cas l'évaluation de toutes les contraintes n'est pas évitée.
- Une méthode efficace d'organisation des contraintes utilisant aussi les propriétés de convexité des régions critiques consiste à placer les inégalités dans les nœuds d'un arbre de recherche binaire. La technique de positionnement exploite cet arbre en le traversant à partir de la racine et en recherchant la région critique active dans une feuille. Du point de vue mémoire, cette méthode est équivalente à la précédente. En revanche, la charge de calcul est considérablement réduite car seules les contraintes se trouvant sur les nœuds reliant la racine à la feuille active sont évaluées.
- Un point sensible de la méthode précédente est le fait que dans le pire cas le nombre d'inégalités à évaluer est égal à la profondeur de l'arbre de recherche. Dès lors, l'efficacité peut être accrue si l'arbre de recherche est bien équilibré. Dans ce sens, [TJB03b] propose une méthode de construction d'un arbre de profondeur minimale en cherchant à réduire le nombre de régions restant à tester dans le passage d'un niveau à l'autre de l'arbre de recherche. Les performances rapportées dans [TJB03b]traduisent une complexité logarithmique par rapport au nombre de régions critiques.
- D'autres méthodes de positionnement pour l'évaluation des solutions explicites utilisent une fonction basée sur la fonction de coût pour discriminer la région critique active [BOR03].

- Le processus de positionnement peut utiliser lors de la création de l'arbre binaire de recherche le prédécoupage en régions correspondant aux formes régulières du domaine faisable (Figure 4.13). Une telle approche bénéficie de la double hiérarchisation des partitionnements pour construire des arbres de recherche binaires équilibrés.

Pour résumer, la complexité en ligne est totalement liée à la complexité du partitionnement de l'ensemble des paramètres faisables. La comparaison entre l'implémentation de la commande prédictive sous contraintes à l'aide des programmes classiques QP en ligne et celle fondée sur la représentation explicite sera donc fortement influencée par les performances du mécanisme de positionnement.

**Remarque :** Si l'on a la liberté pour les programmes QP de choisir des options comme la tolérance ou le nombre d'itération à effectuer, avec comme résultat une solution sous-optimale, dans le cas des formulations explicites la sous-optimalité provient d'un regroupement des régions critiques et du remplacement de la solution optimale initiale sur le domaine résultant par une interpolation conservant les propriétés de linéarité et de continuité. Il en résulte une partition de moindre complexité et donc un positionnement plus rapide [RG05] [GJ02].

### 4.4.4 Charge de calcul et capacité mémoire requise

Les besoins en terme de capacité de stockage de la solution explicite se mesurent à partir de la dimension de la table contenant cette solution. Cette table regroupant les informations liées aux régions critiques et à la loi de commande correspondante, un premier indicateur est la complexité du partitionnement dans l'espace des paramètres, liée au nombre d'entrées dans le tableau. Ainsi il convient de retenir le nombre de régions critiques $N_r$ comme paramètre à utiliser pour l'évaluation des besoins en place mémoire. Deuxièmement, les partitions sont des objets polyédraux qui peuvent être obtenus via une double représentation si l'on utilise des approches géométriques. Or, pour le positionnement, la représentation des générateurs n'apporte aucune information, elle peut donc être évacuée. Même si l'on retient uniquement les contraintes, mesurer les besoins de stockage n'est pas une tâche facile car les régions peuvent être définies par un nombre variable d'hyperplans. Une borne supérieure pour chaque région critique peut être obtenue en considérant la formulation (4.101) – qui contient un nombre de contraintes de :

$$N_c^i = r + n_c^i \tag{4.122}$$

avec $r$ nombre total de contraintes du problème d'optimisation et $n_c^i$ nombre de contraintes actives pour la région critique considérée. Au total, pour toutes les régions critiques, il vient :

$$\sum_{i=1}^{N_r} \left( r + n_c^i \right) \tag{4.123}$$

Comme expliqué auparavant, lors de l'utilisation des arbres de recherche binaires, le nombre de contraintes stockées peut être réduit de moitié en exploitant leur complémentarité. La borne supérieure (4.123) est donc souvent très pessimiste car beaucoup de ces contraintes sont redondantes. Il est également nécessaire, en plus de ces inégalités décrivant les régions critiques, de stocker les lois linéaires pour l'évaluation de la loi prédictive explicite. Au total, il convient de s'assurer que la mémoire disponible lors de l'implémentation atteint en termes de nombres réels :

$$N_{TOT} = N_r (m \cdot n + m) + (n+1) \sum_{i=1}^{N_r} \left( r + n_c^i \right) \tag{4.124}$$

Cette valeur reste une borne supérieure qui ne prend pas en compte le fait que les régulateurs dans des régions voisines puissent avoir des descriptions similaires ou les effets de l'existence de symétries. [JG02] a détaillé récemment l'implémentation de lois de commande prédictive sur des structures FPGA (*field programmable gate array*) et ASIC (*application specific integrated circuit*) avec environ 20000 portes logiques. Ce résultat ouvre la voie à la commande de processus avec des périodes d'échantillonnage faibles, de l'ordre de quelques microsecondes. Néanmoins, la capacité mémoire requise devient très vite importante pour des ensembles de contraintes complexes et des horizons de prédiction importants, à cause de la complexité de la solution explicite.

### 4.4.5 Analyse de l'influence de la redondance locale des contraintes sur la solution explicite

Les paragraphes précédents ont proposé une méthode géométrique d'élaboration de solutions explicites, dans un cadre général avec un éventuel découpage de l'espace des paramètres en zones correspondant aux ensembles de contraintes localement non redondantes.
En fait, ce découpage peut être utilisé de façon indépendante en relation avec n'importe quelle autre stratégie. En effet, avec la technique présentée au chapitre 3, un découpage du domaine original :

$$VD = \bigcup_{k=1}^{n_D} D_k$$

est réalisé en utilisant les sommets paramétrés et leurs domaines de validité. Il en résulte $n_D$ problèmes d'optimisation multiparamétrique avec contraintes non redondantes par morceaux (Figure 4.25).

| | |
|---|---|
| $x \in D_1$ | $\arg\min_{k_u} = k_u^T H \mathbf{k}_u + 2x^T F \mathbf{k}_u$ <br> $\begin{cases} A_{in_1}\mathbf{k}_u \le B_{in_1}x + b_{in_1} \\ A_{eq}\mathbf{k}_u = B_{eq}x + b_{eq} \end{cases}$ |
| … | … |
| $x \in D_{n_d}$ | $\arg\min_{\mathbf{k}_u} = \mathbf{k}_u^T H \mathbf{k}_u + 2x^T F \mathbf{k}_u$ <br> $\begin{cases} A_{in_{n_d}}\mathbf{k}_u \le B_{in_{n_d}}x + b_{in_{n_d}} \\ A_{eq}\mathbf{k}_u = B_{eq}x + b_{eq} \end{cases}$ |

$\rightarrow$

| | | |
|---|---|---|
| $x \in D_1$ | $x \in D_{11}$ | $\mathbf{k}_u = K_{11}x + c_{11}$ |
| | … | … |
| | $x \in D_{1nD_1}$ | $\mathbf{k}_u = K_{1nD_1}x + c_{1nD_1}$ |
| … | … | |
| $x \in D_{n_d}$ | $x \in D_{n_d 1}$ | $\mathbf{k}_u = K_{n_d 1}x + c_{n_d 1}$ |
| | … | … |
| | $x \in D_{n_d n_{D_{n_d}}}$ | $\mathbf{k}_u = K_{n_d n_{D_{n_d}}}x + c_{n_d n_{D_{n_d}}}$ |

**Figure 4.25 : Partition de l'espace des paramètres en régions sans redondance puis construction de la solution explicite pour chaque sous-problème**

Si l'on désire élaborer une solution explicite pour le problème original, alors chaque sous-problème doit être exploré. L'union des descriptions partielles représentera la fonction explicite globale exigée (Figure 4.25b). Au cours de ce développement, deux découpages sont donc construits, tout d'abord pour les zones sans redondance, puis un deuxième en tant que résultat de chaque sous-problème quadratique multiparamétrique. Le résultat global est un arbre à deux niveaux comme modélisé Figure 4.13. Lors de l'implémentation en ligne, la construction d'un arbre de recherche est une priorité pour l'évaluation efficace de la loi de commande. Disposer d'une construction hiérarchisée peut s'avérer bénéfique en développant un arbre de recherche binaire. La raison principale vient du fait que les feuilles du niveau inférieur de la Figure 4.13 sont uniformément distribuées et mènent à des arbres de recherche équilibrés.
Les avantages et inconvénients liés à la manipulation d'une formulation compacte redondante ou à la définition des contraintes sans redondance locale doivent être examinés à l'aide des indicateurs suivants :

- nombre de partitions de la solution explicite globale
- complexité de l'arbre de recherche
- volume des calculs hors ligne lors de la construction de la solution explicite et de l'arbre de recherche
- problèmes numériques

***Avantages***

Le gain en terme de volume de calculs hors-ligne pour chaque sous-problème comparé au problème original est évident. On peut remarquer par ailleurs que ce prétraitement se réfère exclusivement à l'ensemble des contraintes et ainsi, si la conception de la commande prédictive devait être modifiée au niveau de la fonction de coût (pondérations,…), les régions sans redondance restent les mêmes et sont donc réutilisables. Cet aspect constitue un gain non négligeable dans le cas de la redéfinition de la fonction de coût. Du point de vue de l'implémentation en ligne, les gains doivent être mesurés par le nombre de noeuds et la profondeur de l'arbre binaire de recherche associé à la solution explicite globale. Les découpages supplémentaires peuvent contribuer positivement par la construction d'un arbre de recherche binaire équilibré. Finalement, les erreurs numériques sont réduites dans la configuration sans redondance car la taille des sous-problèmes est réduite.

**Exemple 4.11**
Ces avantages sont illustrés ci-dessous par un problème mpQP simple :

$$\begin{array}{l} \min \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} 14.6557 & 7.3274 \\ 7.3274 & 7.4035 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 7.6564 & 3.8480 \\ 17.6549 & 10.8069 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ \begin{cases} -1 \le u_1 \le 1; -1 \le u_2 \le 1; -5 \le x \le 5 \\ -1 + 0.5(x_1 + x_2) \le u_1 \le 1 + 0.5(x_1 + x_2) \end{cases} \end{array} \tag{4.125}$$

La solution explicite est présentée Figure 4.26a. Le découpage en régions sans redondance se compose de 2 domaines de l'espace des paramètres (Figure 4.26b), que l'on retrouve dans la solution explicite (Figure 4.26c). Le mécanisme de positionnement bénéficie de ce découpage puisque les régions critiques présentent une symétrie par rapport à cette droite. Ceci signifie que l'arbre binaire de recherche peut être construit avec pour racine l'évaluation de l'appartenance à l'un des deux demi-espaces.

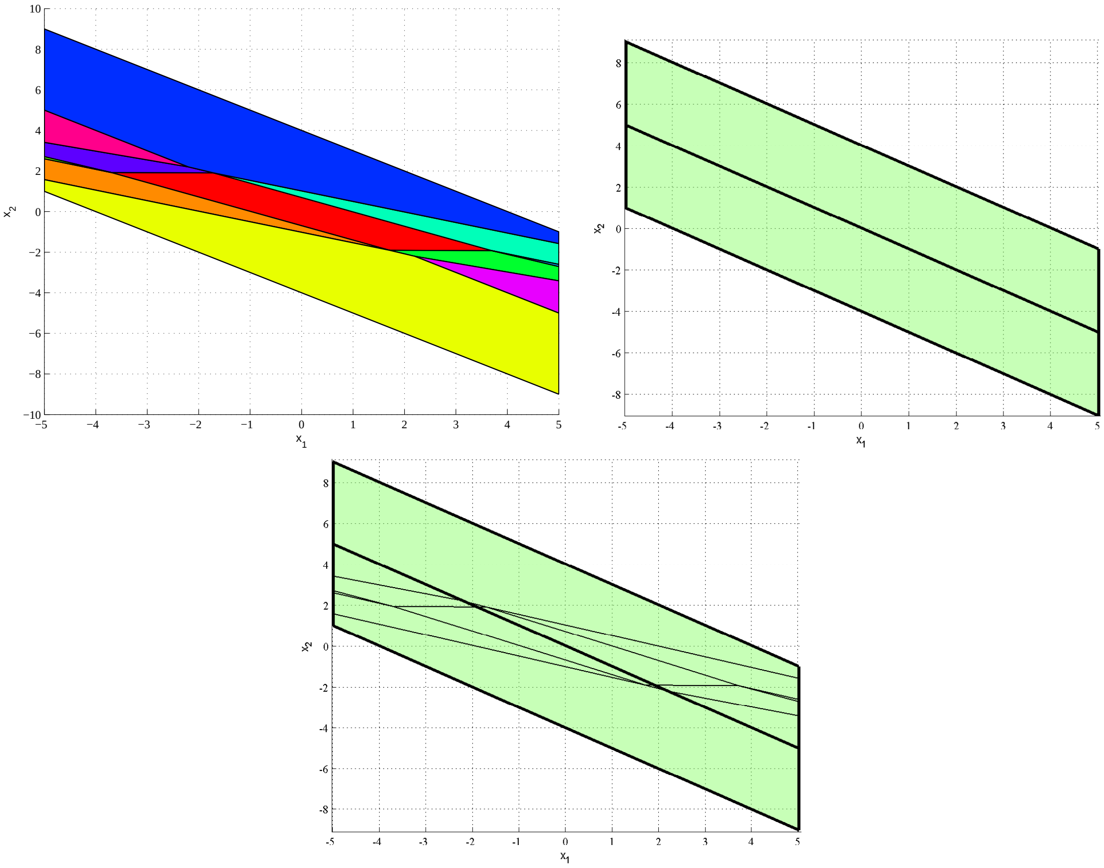



**Figure 4.26 : a) Partition pour la solution explicite en ignorant la redondance locale de contraintes**
**b) Découpage en zones sans redondance**
**c) Partition correspondant à la solution explicite construite pour chaque sous-problème sans redondance**

***Inconvénients***

Les découpages de l'espace des paramètres liés à la formulation sans redondances peuvent introduire dans certains cas des partitionnements superflus, l'ensemble faisable changeant de topologie sans affecter pour autant l'optimum sans contraintes. Du point de vue numérique, le découpage en régions sans redondance peut induire un nombre important de sous-problèmes et le volume de calcul global pour construire la solution explicite peut être rédhibitoire.

Finalement, afin de choisir entre une approche générale élaborant la solution explicite en travaillant directement sur le problème sous sa forme compacte ou une approche sans redondance résolvant une séquence de sous-problèmes sans redondance locale, une analyse des degrés de redondance doit être effectuée, similaire à celle développée dans le cas des procédures d'optimisation en ligne. Pour illustrer cette analyse, le benchmark décrit au chapitre 3 (paragraphe 3.5.2) est de nouveau considéré.

**Benchmark**

Le nombre de côtés réguliers des polygones (*N*) et le nombre de dépendances linéaires considérées au sein des contraintes paramétrées (*M*) donnent la liberté de modifier le degré de redondance et la taille des contraintes non redondantes sans bouleverser la philosophie de la commande prédictive du problème. Avec l'évolution du vecteur d'état, une partie des contraintes (*N*) est employée pour fournir la région faisable tandis que toutes les autres (*M N*) sont redondantes. Partitionner l'espace des paramètres en zones de contraintes non redondantes est immédiat (Figure 4.27) puisque les ensembles de contraintes sont séparables.

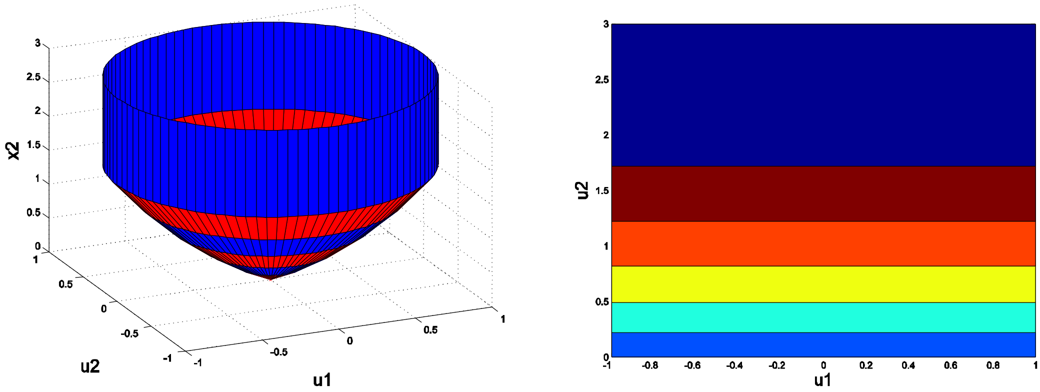



**Figure 4.27 : a) Domaine faisable en fonction de *x***
**b) Régions dans l'espace des paramètres correspondant aux ensembles de contraintes sans redondance (cas M=5, N=50)**

La charge de calcul et les solutions explicites résultantes doivent être comparées pour analyser les effets de la redondance. Les paramètres influençant cette comparaison sont principalement :

- **Degré de redondance**, variable avec le nombre d'ensembles paramétrés *M* et le type de polyèdre régulier *N*.
- **Nombre de régions sans redondance**, dépend exclusivement de *M*.
- **Taille de l'ensemble original**, liée à la somme des contraintes, $N(M+1)\text{x}N_u$.
- **Nombre de variables de décision** $N_u=2$, fixe.
- **Dimension de l'espace des paramètres** $n=2$, fixe.
- **Nombre de partitions de la solution explicite pour chaque $D_i$**, varie avec *N*.

Pour illustrer le fait que le découpage en régions sans contraintes redondantes peut être utilisé avec n'importe quel solveur de problèmes d'optimisation multiparamétrique, les résultats des simulations suivantes sont obtenus avec les procédures de la bibliothèque MPT [KGB04].

La Figure 4.28 présente la commande en fonction des paramètres. Les zones polyédrales correspondant aux solutions explicites sont au nombre de 109 pour la formulation initiale contre 114 dans le cas sans redondance. En fait, les découpages (Figure 4. 27b) se retrouvent dans la solution explicite sans redondance Figure 4.28b.

La solution explicite pour le problème original est minimale en terme de régions polyédrales, tandis que la formulation sans redondance doit faire intervenir une étape d'identification des régions ayant la même forme linéaire affine afin de retrouver la même topologie. Il apparaît clairement dans la représentation de la fonction linéaire par morceaux 6 régions qui peuvent être remplacées par leur couverture convexe (Figure 4.29).

L'arbre binaire de recherche construit pour la solution minimale conduit à une structure avec 163 noeuds. La même méthode [TJB03b] appliquée pour les régions sans contraintes redondantes produira un ensemble d'arbres de recherche ayant {20, 18, 18, 18, 18, 18} nœuds. Un arbre global de recherche peut être construit en positionnant ces sous-arbres sur les feuilles d'un arbre de recherche (Figure 4.13). Finalement la dimension totale de l'arbre de recherche pour la solution explicite dans le cas sans redondance est de 115 nœuds (110 augmentés de 5 liés au choix du sous-arbre considéré). Ce résultat montre que le prétraitement de l'ensemble des contraintes peut avoir comme conséquence une amélioration de l'évaluation en ligne de la solution explicite.

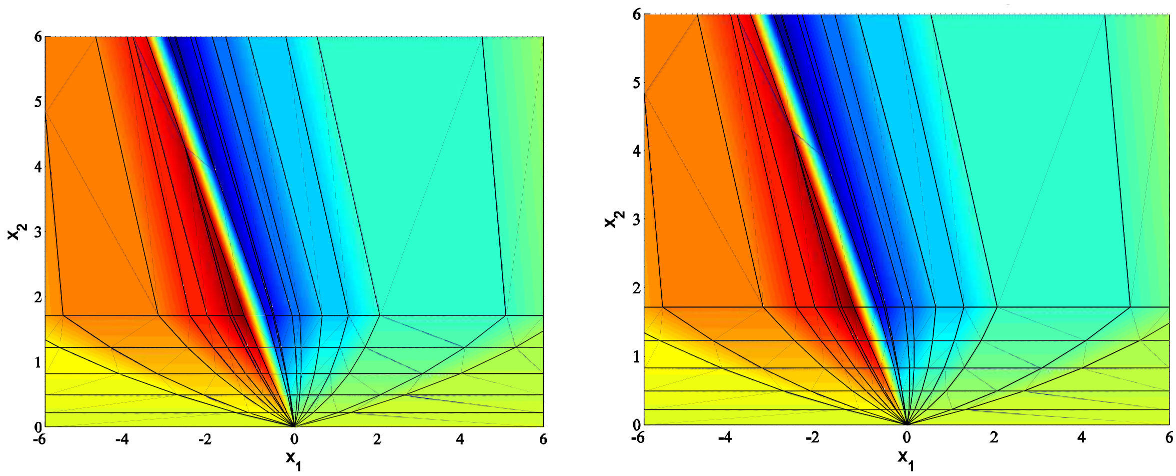



**Figure 4.28 : a) 109 régions pour la solution explicite du problème original b) 114 régions pour la solution explicite obtenue pour les sous-problèmes sans redondance (cas M=5, N=50)**

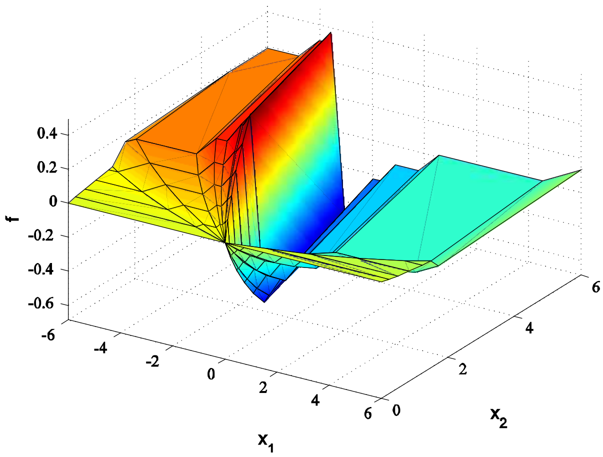



**Figure 4.29 : Formulation explicite de la loi de commande prédictive**

La simulation avec différentes valeurs pour $M$ et $N$ (ce qui modifie la dimension de l'ensemble de contraintes) fournit un avis global synthétisé Figure 4.30. Les avantages de la solution sans redondance ressortent clairement puisque la complexité de l'arbre de recherche est toujours en faveur de cette solution. La charge de calcul pour la construction de la formulation explicite à partir de l'ensemble compact de contraintes se concentre dans l'exploration de l'espace des paramètres et la création des régions critiques (42,6% du temps de calcul total avec le solveur MPT pour $M = 3,\ N = 40$) et augmente avec la dimension de l'ensemble des contraintes. Ce pourcentage devient 98,3% pour $M = 5,\ N = 80$. Or, comme ces régions critiques sont décrites par les contraintes, le temps de calcul est fortement influencé par la présence de la redondance.

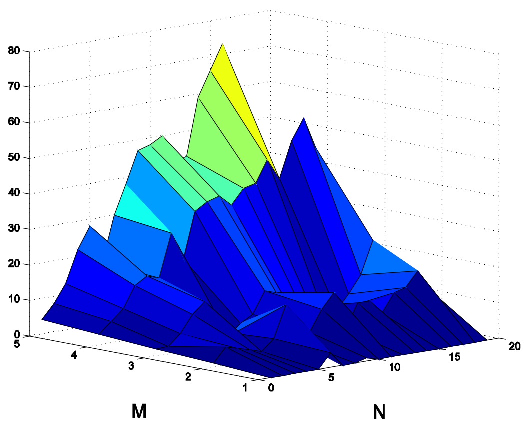



**Figure 4.30 : Différence entre les nombres de noeuds des arbres de recherche pour l'évaluation des solutions explicites pour le problème initial et pour la séquence des problèmes sans redondance**

En effet pendant la construction de la solution explicite, une partie importante (plus de 70% du temps d'appel du solveur mpQP) est consacrée à l'élimination des contraintes redondantes durant le processus itératif d'identification des contraintes redondantes. Ceci constitue donc déjà une motivation forte pour la mise en œuvre d'un pré-processeur de redondance.

Afin de détailler l'analyse numérique, le Tableau 4.1 présente les indicateurs de performance, d'une part pour le problème original et d'autre part en utilisant un préprocesseur de redondance. Les indices de performance sont représentés par le nombre de régions polyédrales de la solution explicite ($NZ_{expl}$), le nombre de noeuds de l'arbre de recherche ($NZ_{STree}$), le nombre de noeuds pour chaque arbre de recherche correspondant à une sous-région ($NZ_{reg}$), le temps moyen hors-ligne pour construire la solution explicite ($T_{expl}$) et le temps moyen pour construire l'arbre de recherche ($T_{STree}$).

| Paramètres | | Indices de performance pour le cas sans redondance | | | | | | Indices de performance pour le problème original | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $M$ | $N$ | $T_{RF}$ | $NZ_{expl}$ | $NZ_{reg}$ | $NZ_{Stree}$ | $T_{expl}$ | $TS_{Tree}$ | $NZ_{expl}$ | $NZ_{Stree}$ | $T_{expl}$ | $TS_{Tree}$ |
| 1 | 5 | 0.093 | 10 | {5,5} | 9 | 0.172 | 0.359 | 9 | 9 | 0.64 | 0.547 |
| 1 | 50 | 2.45 | 38 | {19,19} | 39 | 0.985 | 2.859 | 37 | 39 | 1.087 | 4.581 |
| 1 | 95 | 13.61 | 73 | {36,37} | 79 | 2.735 | 10.484 | 72 | 79 | 4.86 | 25.921 |
| 3 | 5 | 0.23 | 20 | {5,5,5,5} | 19 | 0.422 | 0.610 | 17 | 21 | 0.344 | 1.515 |
| 3 | 50 | 10.94 | 76 | {19,19,19,19} | 77 | 1.438 | 4.172 | 73 | 90 | 2.687 | 28.953 |
| 3 | 95 | 80.32 | 139 | {36,35,35,33} | 146 | 3.624 | 14.235 | 136 | 168 | 114.047 | 153.796 |
| 5 | 5 | 0.50 | 30 | {5, 5, 5, 5, 5, 5} | 29 | 0.876 | 0.983 | 25 | 33 | 0.671 | 3.454 |
| 5 | 50 | 25.81 | 114 | {19,19,19,19,19,19} | 115 | 3.485 | 7.719 | 109 | 163 | 5 | 94.172 |
| 5 | 80 | 248.80 | 167 | {28,31,27,27,27,27} | 171 | 5.984 | 16.954 | 162 | 244 | 256.544 | 366 |

**Tableau 4.1 : Evolution des paramètres et indices selon les simulations**

Du point de vue implémentation en ligne, l'arbre de recherche est l'indicateur principal et en comparant les colonnes donnant $NZ_{STree}$ pour les deux configurations, on observe des gains sensibles, d'autant plus importants que le nombre de zones sans redondance augmente ($M = 5$) car le prédécoupage assure un meilleur développement d'un arbre de recherche équilibré (voir la colonne $NZ_{reg}$). Un cas intéressant est celui avec $M = 1$, pour lequel les arbres de recherche sont équivalents. $NZ_{STree}$ est l'indicateur le plus approprié pour le volume de calcul en ligne.
Si l'on compare les deux colonnes $NZ_{expl}$, on remarque des valeurs assez semblables, avec une différence due aux découpages artificiels introduits par la formulation sans redondance. Ce fait contraignant, déjà mentionné au paragraphe 4.3.3, devient cependant un avantage après la construction de l'arbre de recherche binaire, pour le positionnement en ligne dans le tableau de recherche de la solution explicite.

Les temps de calcul nécessaires pour développer la formulation explicite ($T_{expl}$) et l'arbre correspondant ($TS_{Tree}$), ne doivent pas être négligés. Une comparaison considérant uniquement le temps nécessaire aux appels des procédures mpQP montre que la charge de calcul est toujours réduite. Ainsi, pour $M = 5, N = 50$, le temps moyen par appel dans le cas initial est de 5 s contre 3,485 s dans le cas sans redondances. La différence est encore plus évidente en mesurant le temps nécessaire pour la construction de l'arbre de recherche, 94,172 s contre 7,719 s. Certes, ces procédures se déroulent hors-ligne et une procédure d'élaboration de la solution explicite plus lente n'est pas a priori trop pénalisante, mais, pour des problèmes complexes, le volume de calcul en ligne devient rédhibitoire pour le cas initial (Figure 4.31 comparée à la Figure 4.32) [16]. Pour résumer, le cas sans redondances induit un temps de calcul beaucoup plus réduit (pour $M = 1$ le gain est de 50% et il devient de 83% pour $M = 3$ et 97% pour $M = 5$). Même en ajoutant $T_{RF}$, le temps nécessaire pour effectuer le prédécoupage, le temps total est réduit par rapport au problème original. Un autre aspect important déjà mentionné est qu'un découpage en sous-problèmes sans redondances peut être réutilisé pour différentes fonctions de coût tandis que rien ne peut être réutilisé pour le problème original [17].

[16] Les points absents des figures sont dus aux échecs numériques pour le problème original.
[17] Les procédures mpQP de la toolbox MPT [KGB04] ont été utilisées sous Matlab 6.5 avec un processeur à 1600 Mhz.

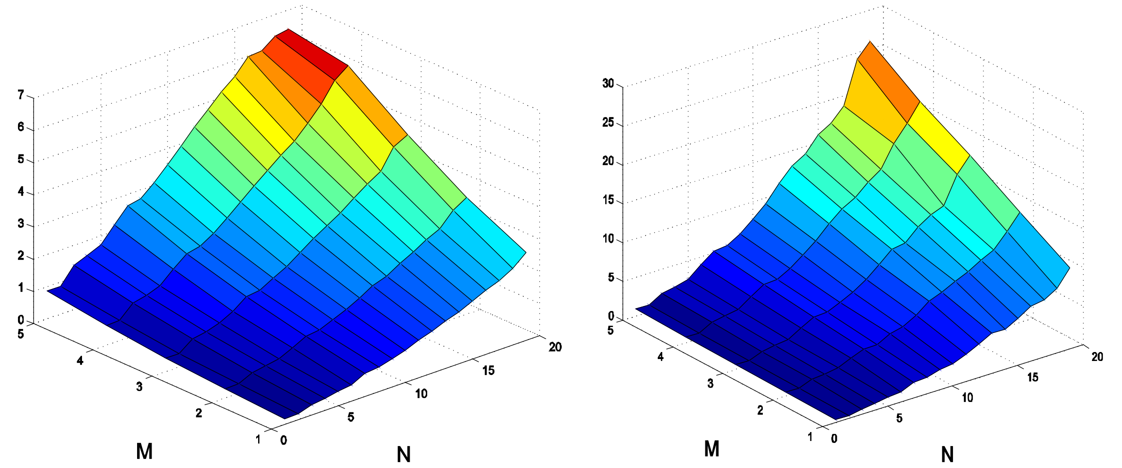



**Figure 4.31 : Temps de calcul pour construire la solution explicite à l'aide de découpage en zones sans redondance. a) Temps nécessaire pour construire la solution (solveur mpQP) b) Temps nécessaire pour construire l'arbre de recherche associé**

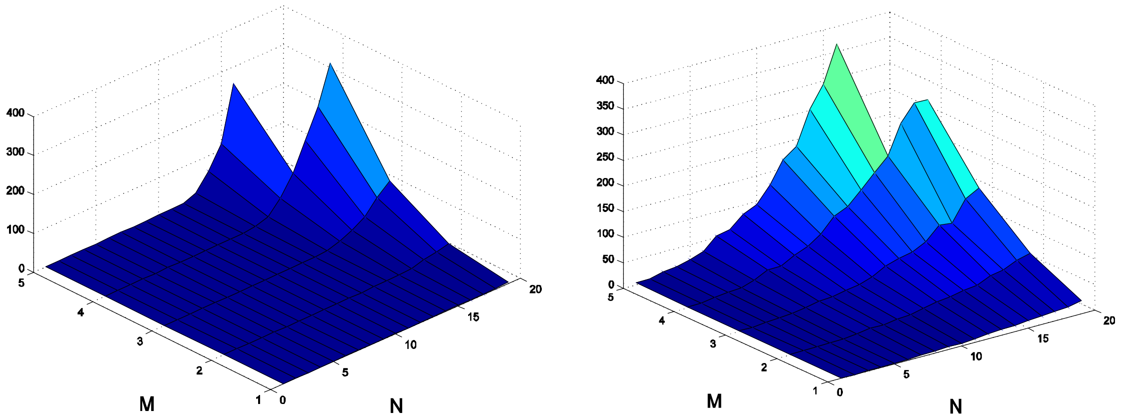



**Figure 4.32 : Temps de calcul pour construire la solution explicite du problème initial a) Temps nécessaire pour construire la solution (solveur mpQP) b) Temps nécessaire pour construire l'arbre de recherche associé**

Il reste à ce stade à aborder les problèmes numériques (rapidement mentionnés au sujet de la Figure 4.32). A ce propos, il est intéressant de noter que les mêmes procédures agissent plus longtemps sans erreur pour des problèmes plus complexes si le prétraitement de la redondance est mis en œuvre. Ainsi dans le Tableau 4.2 partie gauche, on retrouve des combinaisons de paramètres pour des problèmes pour lesquels les procédures de construction de la solution explicite ont échoué.

| Échecs numériques – formulation originale | | | | | | Appels réussis – Cas sans redondance | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $M$ | 1 | 2 | 3 | 4 | 5 | $M$ | 1 | 2 | 3 | 4 | 5 |
| $N$ | 425 | 225 | 125 | 100 | 85 | $N$ | 850 | 500 | 250 | 200 | 200 |

**Tableau 4.2 : Échecs dus aux erreurs numériques**

Pour illustrer le fait que les prédécoupages en régions sans redondance peut repousser la limite des dimensions des problèmes pouvant être traités sans problème numérique, le Tableau 4.2 partie droite donne les combinaisons des paramètres de problèmes ayant pu être traités avec succès. Il est probablement possible de trouver des valeurs de paramètres encore plus grandes mais le temps de calcul devient prohibitif.

**Remarque** : Pendant l'analyse précédente, le degré de redondance a varié de façon inversement proportionnelle avec le nombre de zones de découpage. Cette configuration ne reflète cependant pas le cas général, et le nombre de régions est a priori indépendant des degrés de redondance de chaque zone respective. Ainsi, une étude cas par cas doit être effectuée avant l'implémentation, en considérant au moins le degré moyen de redondance.

**Remarque** : Généralement les découpages de l'espace des paramètres fournissent des ensembles de contraintes variant très peu d'une partition à l'autre et parfois l'union de quelques zones voisines peut être un bon compromis entre un faible degré de redondance et un nombre important de zones.

En conclusion, même si elles sont dépendantes du degré de redondance, du nombre de zones ou du nombre de contraintes dans l'ensemble original, certaines améliorations peuvent exister lors de la construction des solutions explicites si des prédécoupages en régions sans redondance sont effectués. Les améliorations principales sont liées à la construction et à la complexité de l'arbre de recherche (avec des influences sur l'implémentation en ligne) mais également à la charge de calcul induite par les procédures impliquées.

## *4.5 Commande Prédictive Généralisée formulée de façon explicite*

Tous les développements précédents se sont focalisés sur la solution des problèmes d'optimisation quadratique multiparamétrique et leurs applications pour la formulation explicite de lois prédictives dans le cadre des modèles par variables d'état, car, dans ce cas, la fonction optimale obtenue procure une vision directe des lois à appliquer effectivement sous forme de retour d'état. Il est évident que les mêmes méthodes de construction des lois explicites peuvent être utilisées dans le cadre de la commande prédictive généralisée (GPC) avec contraintes, avec une approche par fonction de transfert.

On a vu dans les chapitres précédents que la commande GPC est caractérisée par l'utilisation de modèles de type CARIMA et que la séquence des commandes optimales appliquées selon le principe de l'horizon glissant est le résultat d'un problème d'optimisation multiparamétrique :

$$\begin{aligned}
&\min_{\mathbf{k_u}} \mathbf{k_u^T} \underbrace{(\mathbf{G^T G} + \lambda \mathbf{I}_{\mathbf{N}_u \times \mathbf{N}_u})}_{H} \mathbf{k_u} + 2\mathbf{p}_t^T \underbrace{\left[\mathbf{ih} \;\; \mathbf{if} \;\; -\mathbf{I}_{(N_2-N_1+1)\times(N_2-N_1+1)}\right]^T \mathbf{G}}_{F} \mathbf{k_u} \\
&A_{in}\mathbf{k}_u \le b_{in} + B_{in}\mathbf{p}_t \\
&A_{eq}\mathbf{k}_u = b_{eq} + B_{eq}\mathbf{p}_t
\end{aligned} \tag{4.126}$$

avec $\mathbf{p}_t = \left[\Delta u_{t-1} \;\cdots\; \Delta u_{t-n_b-1} \;\; y_t \;\cdots\; y_{t-n_a-1} \;\; w_{t+N_1} \;\cdots\; w_{t+N_2}\right]^T$ le vecteur des *paramètres du contexte* pour le problème d'optimisation (4.126) et $\mathbf{G}, \mathbf{ih}, \mathbf{if}$ l'expression matricielle des solutions des équations diophantiennes intervenant dans l'élaboration des prédicteurs.

### 4.5.1 Solution explicite par morceaux

Dans le chapitre 2, l'analyse de la commande de type GPC sans contraintes a montré que sa structure permet une implémentation à base de régulateurs polynomiaux RST à deux degrés de liberté :

$$S(q^{-1})\Delta u_t = -R(q^{-1})y_t + T(q)w_t \tag{4.127}$$

En présence de contraintes, il est intéressant d'examiner l'évolution de la structure de la loi GPC lorsque la solution explicite du problème d'optimisation multiparamétrique associé est élaborée. Les détails de synthèse de la solution GPC explicite sont résumés par la proposition suivante.

**Proposition 4.12 :** La loi de commande GPC avec contraintes peut être structurée sous forme polynomiale à l'aide de régulateurs 'RST affines' définis pour des régions polyédrales dans l'espace des paramètres de contexte :

$$\begin{aligned}
&S_i(q^{-1})\Delta u_t = -R_i(q^{-1})y_t + T_i(q)w_t \\
&\qquad \text{pour} \quad \left[\Delta u_{t-1} \cdots \Delta u_{t-n_b-1} \,\middle|\, y_t \cdots y_{t-n_a-1} \,\middle|\, w_{t+N_1} \cdots w_{t+N_2}\right]^T \in P_i
\end{aligned} \tag{4.128}$$

**Preuve :** Pour démontrer cette mise en forme, il suffit de reprendre la solution explicite de (4.126), définie comme une fonction continue linéaire sur des régions polyédrales dans l'espace des paramètres de contexte :

$$\mathbf{k}_u^*(\mathbf{p}) = \begin{cases} \cdots \\ K_{Lin_i}\mathbf{p} + K_{tl_i} \quad \text{pour} \quad \mathbf{p} \in CR_i \\ \cdots \end{cases} \tag{1.129}$$

On peut donc écrire la loi de commande GPC sous la forme :

$$\Delta u_t^{GPC} = [\mathbf{1}\ \ \mathbf{0}\ \ \mathbf{0}]\ \mathbf{k}_u^*(\mathbf{p}) \tag{1.130}$$

ou de façon équivalente avec la loi linéaire par morceaux :

$$\Delta u_t^{GPC} = \begin{cases} \cdots \\ [\mathbf{1}\quad \mathbf{0}\quad \mathbf{0}]K_{Lin_i}\mathbf{p} + [\mathbf{1}\quad \mathbf{0}\quad \mathbf{0}]K_{tl_i} \quad \textit{pour} \quad \mathbf{p} \in CR_i \\ \cdots \end{cases} \tag{1.131}$$

avec :

$$\begin{aligned} &[\mathbf{1}\quad \mathbf{0}\quad \mathbf{0}]K_{Lin_i} = \left[-\widetilde{S}_i \quad -\widetilde{R}_i \quad \widetilde{T}_i\right]; \\ &[\mathbf{1}\quad \mathbf{0}\quad \mathbf{0}]K_{tl_i} = tl_i; \end{aligned} \tag{1.132}$$

D'où :

$$\begin{aligned} \Delta u_t^{GPC} &= \begin{cases} \cdots \\ \left[-\widetilde{S}_i \ -\widetilde{R}_i \ \widetilde{T}_i\right]\mathbf{p} + tl_i \quad \text{pour} \quad \mathbf{p} \in CR_i \\ \cdots \end{cases} \\ &= \begin{cases} \cdots \\ -\widetilde{S}_i\left[\Delta u_{t-1} \cdots \Delta u_{t-n_b-1}\right]^T - \widetilde{R}_i\left[y_t \cdots y_{t-n_a-1}\right]^T + \widetilde{T}_i\left[w_{t+N_1} \cdots w_{t+N_2}\right]^T + tl_i \\ \qquad \text{pour} \quad \mathbf{p} \in CR_i \\ \cdots \end{cases} \\ &= \begin{cases} \cdots \\ -\widetilde{S}_i\left[q^{-1} \cdots q^{-n_b-1}\right]^T \Delta u_t - \widetilde{R}_i\left[q^0 \cdots q^{-n_a-1}\right]^T y_t + \widetilde{T}_i\left[q^{N_1} \cdots q^{N_2}\right]^T w_t + tl_i \\ \qquad \text{pour} \quad \mathbf{p} \in CR_i \\ \cdots \end{cases} \end{aligned} \tag{1.133}$$

ce qui est équivalent à une loi polynomiale affine par sous-domaines polyédraux :

$$S_i(q^{-1})\Delta u_t = -R_i(q^{-1})y_t + T_i(q)w_t + tl_i \quad \text{pour} \quad \mathbf{p} \in CR_i \tag{1.134}$$

avec : $S_i(q^{-1}) = \widetilde{S}_i\left[q^{-1} \cdots q^{-n_b-1}\right]^T; R_i(q^{-1}) = \widetilde{R}_i\left[q^0 \cdots q^{-n_a-1}\right]^T; T_i(q) = \widetilde{T}_i\left[q^{N_1} \cdots q^{N_2}\right]^T$ ■

L'implémentation d'une telle loi nécessite un mécanisme de positionnement pour identifier le régulateur RST actif à chaque pas d'échantillonnage (Figure 4.33).

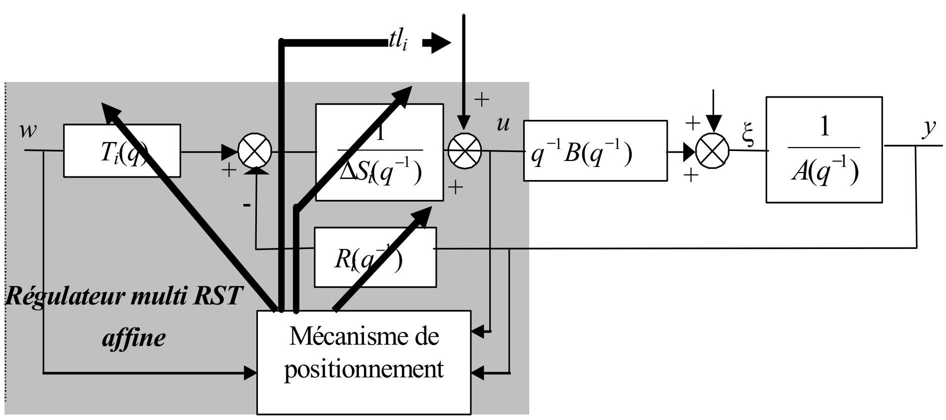



**Figure 4.33 : Régulateur polynomial équivalent avec la loi GPC sans contraintes**

### 4.5.2 Exemple

Afin d'illustrer les méthodes précédentes, considérons la loi de commande GPC sous contraintes pour le système double intégrateur :

$$y_t = \frac{1}{1-2q^{-1}+q^{-2}} u_{t-1} \tag{4.135}$$

avec des contraintes sur les incréments et l'amplitude de la commande :

$$\begin{cases} -0.06 \le \Delta u_{t+k} \le 0.06;\ k = 0..N_u - 1 \\ -0.1 \le u_{t+k} \le 0.1;\ k = 0..N_u - 1 \end{cases} \tag{4.136}$$

La loi de commande se base sur les paramètres de réglage $N_1 = 1; N_u = 2; N_2 = 3; \mathbf{\Lambda} = 0.4 \cdot \mathrm{trace}(\mathbf{G}^{\mathrm{T}}\mathbf{G})\,\mathbf{I}$. Avec ce choix et la forme du modèle, le domaine faisable ne possède qu'un seul paramètre, la valeur passée de l'entrée. La représentation géométrique de ce domaine dans la dimension étendue est donnée par le polyèdre de la Figure 4.34a. Le vecteur complet des paramètres contient également les variables influençant la fonction de coût. Ainsi le domaine paramétré pour le problème d'optimisation est :

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ -1 & -1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \end{bmatrix}}_{k_u} \ge \left[ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \middle\| \; 0_{8\times 6} \right] \underbrace{\begin{bmatrix} u(t-1) \\ y(t) \\ y(t-1) \\ y(t-2) \\ w(t+1) \\ w(t+2) \\ w(t+3) \end{bmatrix}}_{p} + \begin{bmatrix} -0.06 \\ -0.06 \\ -0.06 \\ -0.06 \\ -0.1 \\ -0.1 \\ -0.1 \\ -0.1 \end{bmatrix} \tag{4.137}$$

En plus de ces contraintes explicites, il faut considérer des contraintes implicites sur des paramètres :

$$-0.1 \le u(t-1) \le 0.1 \tag{4.138}$$

Le domaine de recherche est bidimensionnel paramétré par 7 variables dépendant des entrées et des sorties passées et des consignes futures. Le fait que l'optimum contraint dépende de ces 7 paramètres est évident après la substitution :

$$\bar{\mathbf{k}}_u = (G^T G + \Lambda)^{1/2} \mathbf{k}_u + (G^T G + \Lambda)^{1/2} G^T \left[ 0_{(N_2-N_1+1)\times 1} \ \textbf{ih if} \ -I_{N_2-N_1+1} \right] p \tag{4.139}$$

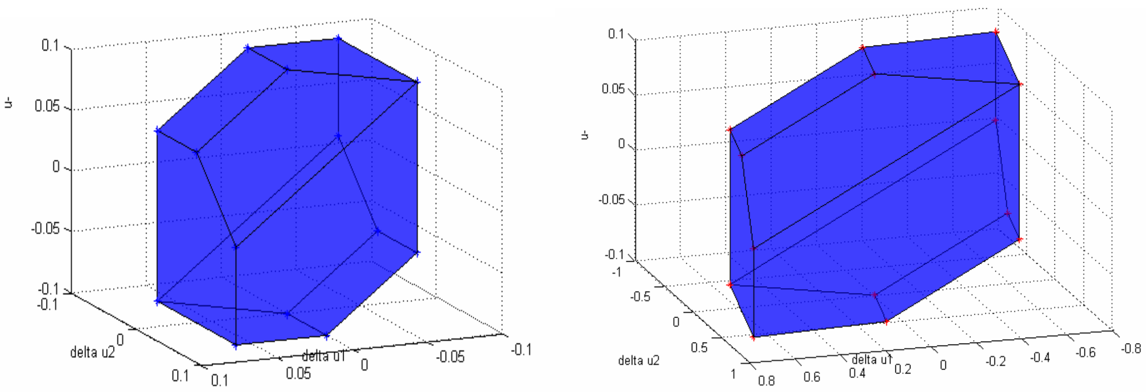



**Figure 4.34 : a) Polyèdre faisable avant substitution**
**b) Projection du polytope sur le sous-espace $[\Delta u_t\ \Delta u_{t+1}\ u_{t-1}]$**

La représentation géométrique du domaine dans la dimension homogène étendue après le changement de variable (4.39) est donnée par un polytope (combinaison des certains sommets - leur projection sur les coordonnées [ $\Delta u_t\ \Delta u_{t+1}\ u_{t-1}$ ] est représentée Figure 4.34b) et un espace de linéarité de dimension 6. Les rayons bidirectionnels sont décrits par les 6 premières lignes du Tableau 4.3 et le polytope par les sommets détectés dans une représentation homogène en tant que rayons dans le tableau 4.3.

| | $\Delta u_t$ | $\Delta u_{t+1}$ | $u_{t-1}$ | $y_{t-1}$ | $y_{t-2}$ | $y_{t-3}$ | $w_{t-1}$ | $w_{t-2}$ | $w_{t-3}$ | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| LIGNES | -0.56 | -0.29 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | -0.28 | -0.08 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | -0.10 | 0.01 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| | 7.68 | 3.36 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | -11.08 | -4.98 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | 4.35 | 1.98 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| RAYONS | 0.53 | 0.53 | -0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | -0.53 | 0.41 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0.53 | -0.41 | 0.04 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0.6 | 0.06 | 0.04 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | -0.67 | -0.55 | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0.67 | 0.55 | -0.02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | -0.06 | -0.48 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | -0.53 | 0.41 | -0.04 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | -0.6 | -0.06 | -0.04 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | -0.67 | -0.55 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | -0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0.06 | 0.48 | -0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0.67 | 0.55 | -0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Tableau 4.3 : Générateurs du domaine faisable dans l'espace étendu**

Le polyèdre précédent possède 20 faces de dimension 7 mais seules 12 ont une projection non dégénérée sur l'espace des paramètres (Tableau 4.4). Ils correspondent aux sommets paramétrés du domaine de faisabilité.

| Sommets | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{k}_u(1)$ | 0.06 | 0.06 | 0.06 | -0.06 | 0.04-x | -0.1-x | -0.06 | -0.06 | 0.1-x | -0.04-x | 0.1-x | -0.1-x |
| $\mathbf{k}_u(2)$ | 0.06 | -0.06 | 0.04-x | 0.06 | 0.06 | 0.06 | -0.06 | -0.04-x | -0.06 | -0.06 | 0 | 0 |

**Tableau 4.4 : Sommets paramétrés**

En utilisant ces sommets et leurs domaines de validité, la procédure *ScanDom* fournit les zones dans l'espace des paramètres ayant des formes régulières, les sommets paramétrés associés et les contraintes non redondantes correspondantes. Les sommets peuvent être identifiés par leur index dans le Tableau 4.5, les contraintes redondantes possèdent des zéros dans le vecteur des indicateurs.

Pour chaque domaine $D_k;\ k=1..5$, des sous-domaines sont créés basés sur des formes régulières similaires avec celles de la Figue 4.35. On considère ensuite l'intersection de ces régions adjacentes au domaine faisable avec l'optimum sans contraintes du problème quadratique. Toutes les intersections sont non dégénérées et le résultat en termes de régions de projection constante est donné par :

$$\begin{array}{ll} D_1 = \bigcup D_{1i}\,; i=1..11 & D_2 = \bigcup D_{2i}\,; i=1..11 \\ D_3 = \bigcup D_{3i}\,; i=1..13 & D_4 = \bigcup D_{4i}\,; i=1..11 \\ \multicolumn{2}{c}{D_5 = \bigcup D_{5i}\,; i=1..11} \end{array} \tag{4.140}$$

| *Domaine* | *Validité* | *Sommets* | *Contraintes non redondantes* |
|---|---|---|---|
| $D_1$ | $-0.1 \leqslant u_{t-1} \leqslant -0.04$ | {1,2,6,10,12} | [1 1 0 1 0 0 1 1 0 0] |
| $D_2$ | $-0.04 \leqslant u_{t-1} \leqslant -0.02$ | {1,2,4,8,10} | [1 1 1 1 0 0 0 1 0 0] |
| $D_3$ | $-0.02 \leqslant u_{t-1} \leqslant 0.02$ | {2,3,4,5,7} | [1 1 1 1 0 1 0 0 0 0] |
| $D_4$ | $0.02 \leqslant u_{t-1} \leqslant 0.04$ | {2,3,4,5,8,10} | [1 1 1 1 0 1 0 1 0 0] |
| $D_5$ | $0.04 \leqslant u_{t-1} \leqslant 0.1$ | {4,5,7,9,11} | [1 1 1 1 0 0 0 1 0 0] |

**Tableau 4.5 : Sous-domaines de l'espace des paramètres, sommets paramétrés et contraintes non redondantes correspondantes**

soit 57 sous-domaines. Le nombre maximal d'évaluations pour se positionner à l'intérieur de ce tableau des lois RST, en utilisant seulement l'arbre de la Figure 4.13 et ensuite une recherche séquentielle pour chaque niveau est de $5+11=16$ tests d'inclusion dans des domaines polyédraux. Si l'on utilise des arbres binaires de recherche, le positionnement se réduit à l'évaluation de 8 inégalités ce qui représente un gain en temps de calcul important par rapport aux procédures basées sur l'optimisation en ligne, soit par des techniques de type ensemble actif, soit par des méthodes de type point intérieur.

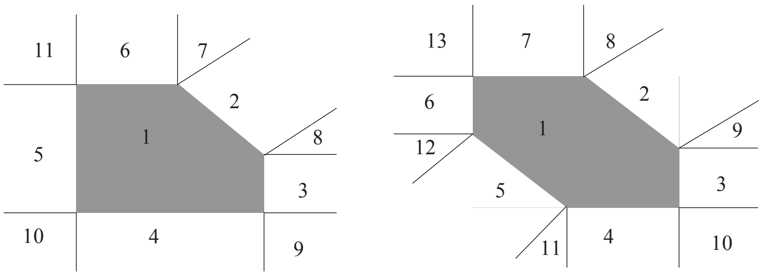



**Figure 4.35 : Types possibles de formes régulières et de sous-domaines de partitionnement Gauche – $D_1, D_2, D_4$ et $D_5$ . Droite – $D_3$.**

Le résultat de ce découpage dans l'espace des paramètres est représenté Figure 4.36 pour la projection du polytope sur les coordonnées $[\Delta u_t \; \Delta u_{t+1} \; u_{t-1}]$ et constitue la base de la procédure de partitionnement pour tout l'espace d'optimisation.

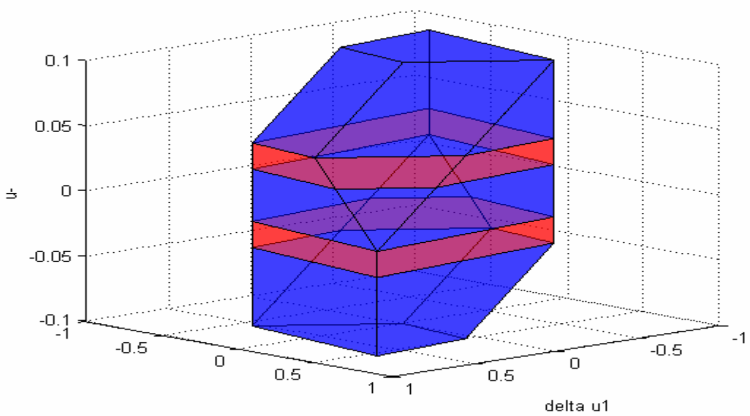



**Figure 4.36 : Projections des domaines de section régulière**

Le loi de commande explicite consiste en un tableau de 57 lois polynomiales affines RST. Un exemple de la forme de ces lois est :

$$
\begin{aligned}
&S_{23}(q^{-1})\Delta u_t = -R_{23}(q^{-1})\, y_t + T_{23}(q) w_t + tl_{23} \\
&S_{23}(q^{-1}) = 1 - q^{-1} ; \\
&R_{23}\,(q^{-1}) = \; -1.18 + 1.71\, q^{-1} - \; 0.67 q^{-2} \\
&T_{23}\;(q) = 0.014\, q + 0.043 q^{2} \; + 0.088 q^{3} ; \\
&tl_{23} \; = \; -0.018
\end{aligned}
\tag{4.141}
$$

Le comportement en ligne décrit Figure 4.37 est obtenu à partir de la loi explicite sous forme polynomiale RST sur des sous-domaines polyédraux, calculée hors-ligne, avec le mécanisme de positionnement en ligne décrit précédemment. Il confirme les résultats obtenus par les stratégies d'optimisation en ligne complète.

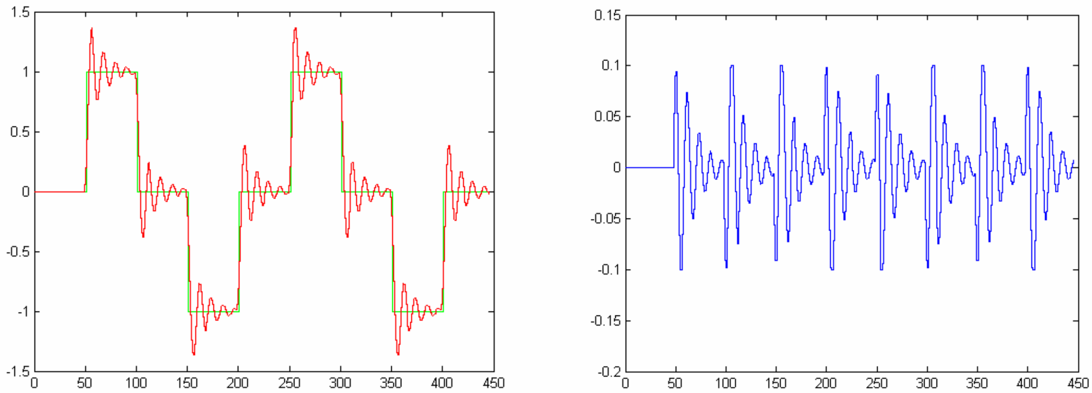



**Figure 4.37 : Comportement en ligne de la loi GPC explicite**

## *4.6 Conclusions*

Ce chapitre a présenté des méthodes de construction de la forme explicite d'une loi de commande prédictive reposant sur la solution explicite de problèmes d'optimisation quadratique multiparamétrique. Comparativement aux méthodes récentes basées sur l'énumération des combinaisons des contraintes actives ou sur l'exploration de l'espace des paramètres à l'aide des conditions KKT, les travaux ont montré que les solutions explicites peuvent être développées exclusivement avec des opérations géométriques. Une telle approche offre un aperçu de la topologie de l'ensemble faisable représenté par un polyèdre paramétré et des techniques peuvent être mises en place pour éviter l'inspection des régions qui n'affectent pas l'optimum global.

De plus, la stratégie proposée a été étendue à la résolution de sous-problèmes multiparamétriques avec contraintes non redondantes localement. Si les degrés de redondance sont importants, une telle approche s'avère bénéfique pour l'amélioration de la charge de calcul et pour la construction des arbres binaires de recherche pour l'évolution de la solution explicite.

Finalement, les formulations explicites de la loi de commande prédictive sous contraintes calculées hors-ligne prennent la forme générale de fonctions continues linéaires par morceaux, et des mécanismes de positionnement dans une table contenant ces lois linéaires doivent être mis en place en ligne. Ainsi les optimisations complètes en ligne sont évitées, et les commandes à chaque pas d'échantillonnage sont calculées via des lois polynomiales RST affines par morceaux dans le cas de la commande GPC ou des lois linéaires en le vecteur d'état pour les modèles par représentation d'état.

# 5. Stabilité et faisabilité

La problématique de la commande prédictive sous contraintes a été traitée jusqu'ici en considérant que le problème d'optimisation avait une solution à chaque pas d'échantillonnage. Ce chapitre se préoccupe désormais des situations d'infaisabilité et insiste également sur les implications de cet aspect sur la stabilité du système bouclé par le correcteur prédictif. On examine également à la fin de ce chapitre les méthodes possibles permettant de récupérer une situation d'infaisabilité.
Si l'on revient sur le contenu des chapitres précédents, on se rend compte que le problème de la commande MPC a été considéré comme donné a priori, et s'est attardé principalement sur les aspects structurels et informatiques, sans considérer le choix des horizons de prédiction, des pondérations ou des contraintes terminales. Ces aspects sont pourtant intimement liés aux performances de la loi de commande (principalement la stabilité) et ont constitué pendant les années 90 des sujets de recherche privilégiés du milieu académique.

Le but de ce chapitre est de faire ressortir des règles de choix des paramètres de la loi prédictive garantissant la faisabilité et ultérieurement la stabilité de la boucle asservie. Même si le domaine a été déjà couvert longuement dans la littérature, les développements suivants proposent une approche basée essentiellement sur les ensembles faisables en tant que domaines polyédraux, pouvant enrichir significativement la théorie existante. Ainsi des conditions nécessaires et suffisantes de faisabilité des lois GPC sont élaborées avec leurs implications sur la stabilité. Une autre contribution est la formulation d'un superviseur de consigne sous la forme d'un problème d'optimisation multiparamétrique élaboré au final sous forme explicite.

L'ensemble de ces contributions se retrouve dans les articles [OD03], [OD04a], [OD04b], [OD05b].

## *5.1 Motivations*

Les chapitres précédents ont formulé le problème de commande prédictive en considérant exclusivement sa structure, son évolution dans le temps et l'application de la séquence optimale dans les meilleures conditions en conformité avec le principe de l'horizon fuyant. Mais la loi prédictive nécessite un choix approprié de paramètres de réglage permettant de satisfaire des demandes de performance et de stabilité. Cet aspect est très sensible et a fait l'objet de nombreux développements dans la littérature, une compilation des résultats peut être consultée dans [MRRS02], [May01], [GSD04]. Du point de vue stabilité, la présence de contraintes lors de la synthèse de la loi de commande prédictive modifie les résultats classiques existants. Considérons comme point de départ de cette analyse une proposition classique pour les systèmes avec contraintes sur l'entrée.

> **Proposition 5.1 :** Un système linéaire continu $\dot{x} = Ax + Bu$ peut être globalement stabilisé par une loi de commande lisse et bornée si et seulement s'il ne possède aucune valeur propre dans le plan complexe à partie réelle positive, et si tous les modes incontrôlables sont à partie réelle négative (voir [Son90] pour la démonstration et surtout [SY91] pour une discussion intéressante sur les limites de la 'saturation aveugle').

Ce résultat illustre parfaitement le fait qu'il est parfois difficile de stabiliser globalement des systèmes avec contraintes car une partie des avantages de la commandabilité est annulée par l'apparition des contraintes. On peut se contenter alors de trouver des lois de commande offrant des garanties de stabilité locale. Ceci est donc une première limitation que la commande prédictive va devoir subir à son tour.

S'intéresser à la faisabilité n'est pas un hasard car l'infaisabilité est souvent précurseur d'instabilité. Considérons pour cela la commande GPC d'un système stable en boucle ouverte :

$$A\left(q^{-1}\right)y(t) = B\left(q^{-1}\right)u(t-d_c) \tag{5.1}$$

pour lequel la loi de commande doit satisfaire les contraintes :

$$\begin{cases} A_\gamma(q^{-1})\,\gamma(t) = B_\gamma(q^{-1})\,u(t-d_c) \\ \gamma_{\min} \le \gamma(t+j|t) \le \gamma_{\max}, \quad N_{c1} \le j \le N_{c2} \end{cases} \tag{5.2}$$

où $\gamma$ est le signal contraint et $N_{c1}, N_{c2}$ déterminent les horizons des contraintes. Si la variable contrainte $\gamma(t)$ correspond à un système à déphasage minimal $\left(A_\gamma, B_\gamma\right)$, toute séquence de commande pour laquelle la faisabilité est assurée va garantir aussi que les quantités $u$ , $\Delta u$ et $y$ restent dans un domaine borné, impliquant la stabilité BIBO [SC94a]. Ceci est donc un exemple où la faisabilité implique la stabilité.

Des difficultés apparaissent quand les contraintes agissent sur une variable liée à un système à déphasage non minimal (5.2). En effet, pendant le transitoire, la présence des contraintes demande un effort 'inverse' de commande pour rester dans la zone admissible et à long terme crée un effet 'boule de neige' qui conduit le système bouclé à l'instabilité. Si l'horizon de prédiction est assez long, le phénomène d'infaisabilité va se manifester de sorte que l'on peut dire dans ce cas que l'infaisabilité précède l'instabilité.

**Exemple 5.2**
Pour illustrer ce phénomène, considérons l'exemple d'un système à déphasage non minimal :

$$(1-q^{-1}+0.25q^{-2})\gamma(t)=(-0.25-0.25q^{-1}+0.75q^{-2})u(t) \tag{5.3}$$

avec la contrainte:

$$-0.3 \le y \le 1 \tag{5.4}$$

Pour un horizon de contraintes d'un seul pas de prédiction et $N_{c1}=N_{c2}=1$, la loi GPC est faisable mais le système en boucle fermée est instable comme on peut le constater Figure 5.1.

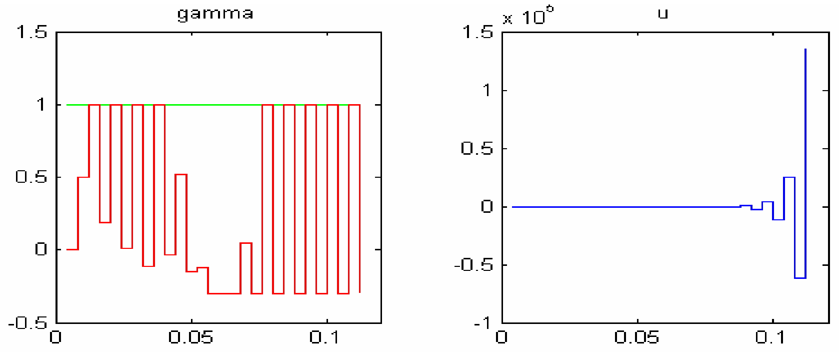



**Figure 5.1 : Comportement instable d'une loi de commande faisable**
**a) Sortie contrainte b) Commande**

Si l'horizon de contraintes est augmenté, $N_{c2}=6$, l'infaisabilité est signalée à la $6^{eme}$ itération, confirmant le fait que l'infaisabilité précède l'instabilité quand l'horizon de prédiction est assez long. Une analyse hors-ligne de ce phénomène d'infaisabilité peut alors s'avérer utile car des règles peuvent être élaborées pour choisir l'horizon de contraintes (ou le système de contraintes) apportant à la fois faisabilité et stabilité. Pour l'exemple choisi, la Figure 5.2 donne les simulations pour $N_{c1}=1$ et $N_{c2}=12$, choix impliquant une loi GPC faisable pour la consigne considérée. On peut aussi observer que la faisabilité de la loi GPC implique la stabilité de la boucle fermée.

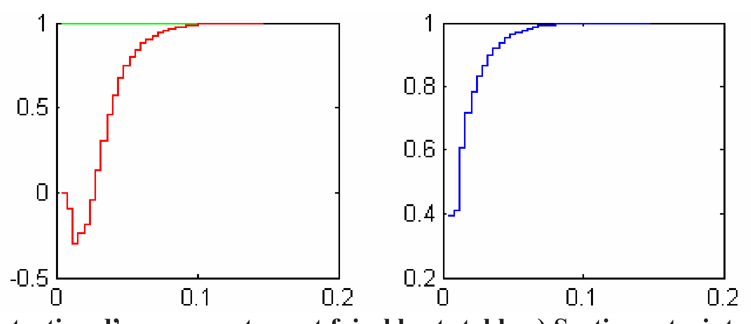



**Figure 5.2 : Illustration d'un comportement faisable et stable a) Sortie contrainte b) Commande**

Ce système fait partie d'une classe de systèmes pour lesquels des lois de commande prédictive mal réglées induisent des commandes augmentant sans fin en valeur absolue. Tel en est le cas pour tous les systèmes à déphasage non minimal, pour lesquels la sortie reste bornée mais la commande peut exploser. Dans le cas sous contraintes, le problème devient plus difficile car le phénomène d'infaisabilité agit souvent comme un témoin de l'instabilité de la stratégie de commande. Il vient comme conséquence immédiate que si l'on assure la faisabilité de la loi de commande pour des contraintes sur tous les signaux du système, la stabilité de type BIBO est garantie. Ce résultat est malheureusement le plus faible vis-à-vis de la stabilité que l'on puisse obtenir.

La commande prédictive est comme on a pu le voir précédemment, basée sur un problème d'optimisation. L'optimalité d'une séquence de commande est la seule propriété sur laquelle on peut se baser pour déduire des garanties de stabilité. Ce résultat n'est pas facile à assurer et les méthodes usuelles agissent sur trois composants principaux :

- le plus souvent un système de contraintes renforcé en ajoutant des contraintes dites 'terminales',
- la description d'une loi 'terminale' s'appliquant à la fin du domaine qui sous-entend ainsi un horizon de prédiction pseudo infini,
- une fonction de coût terminale qui fait que le critère prédictif est considéré comme une fonction de Lyapunov.

Les deux premiers points peuvent être en fait traduits par des contraintes artificielles à prendre en considération à chaque pas d'échantillonnage. La stabilité est assurée théoriquement, avec une complexification du jeu de contraintes, mais en contrepartie le danger que la région de l'espace des variables de décision devienne vide est bien réel. La loi de commande étant basée sur un problème d'optimisation, tout ajout de contraintes au jeu initial implique en effet une diminution du domaine faisable et à la limite, si ce domaine devient nul, conduit aux situations d'infaisabilité. Si l'on veut faire la correspondance avec la formulation explicite des lois de commande MPC, tout ajout de contraintes a des implications sur les domaines pour lesquels la loi de commande est définie dans l'espace des paramètres.

Il s'avère qu'il est donc possible de synthétiser une loi de commande prédictive, de construire sa formulation explicite (selon les méthodes présentées lors des chapitres précédents), tout en étant dans l'impossibilité de l'utiliser car les conditions initiales ou les paramètres pendant l'évolution dynamique du système atteignent des régions pour lesquelles la loi MPC n'est pas définie. L'analyse des performances d'une loi prédictive conduit alors à l'idée que la faisabilité est une condition qui doit être prise en compte lors de la phase de réglage des paramètres de la loi de commande. Ensuite, en se basant sur ces conditions nécessaires, des solutions assurant la stabilité de la boucle fermée peuvent être élaborées. Toutefois, même si la stabilité nécessite la faisabilité, du point de vue pragmatique il faut voir l'infaisabilité comme un phénomène néfaste invalidant la loi MPC/GPC. Quand la procédure d'optimisation recherche une séquence optimale inexistante, le système physique ne peut pas se contenter d'un message 'infeasible problem', et des mesures doivent être prises pour sauvegarder son fonctionnement.

On peut alors se demander comment ce phénomène d'infaisabilité d'une loi de commande prédictive se manifeste en pratique, plus précisément comment survient l'incompatibilité entre contraintes. La réponse est assez complexe, les causes principales engendrant ce type de comportement sont les suivantes.

- la description du problème de commande sous contraintes est intrinsèquement infaisable. Les objectifs ne peuvent pas être atteints, le plus souvent la référence ne tenant pas compte des contraintes sur les variables manœuvrées. On dit que le système de contraintes est incompatible.
- Même si les valeurs de référence sont atteignables en régime nominal, une variation soudaine peut conduire à l'infaisabilité. Si le changement ne peut pas se faire avec des commandes à énergie limitée, les contraintes deviennent temporairement incompatibles.
- Une situation similaire d'infaisabilité peut apparaître si les limitations des variables manœuvrées sont modifiées alors que le système est en action.
- La structure du système à commander (instable, phase non minimale) fait que l'évolution des paramètres conduit hors de la zone faisable.
- Une perturbation extérieure qui n'a pas été prise en considération quand la loi de commande a été synthétisée peut conduire à l'infaisabilité. Le plus souvent, ce phénomène apparaît quand le système est en action à la proximité des frontières faisables et qu'une perturbation le conduit dans une zone interdite.
- Une différence flagrante entre le modèle et le système réel induit l'infaisabilité.
- Un mauvais conditionnement du problème d'optimisation de la loi MPC/GPC la rend infaisable.

Les causes d'infaisabilité sont donc multiples, on se concentre dans ce chapitre sur l'élimination de l'infaisabilité dans le cas déterministe, ignorant pour le moment les infaisabilités provenant des incertitudes de modèle, des perturbations et autres erreurs de prédiction, ces aspects seront envisagés au chapitre suivant se préoccupant de la problématique de robustesse.

### 5.1.1 Infaisabilité issue d'une incompatibilité des contraintes opérationnelles

Dans le cas déterministe, l'infaisabilité peut concerner essentiellement les problèmes de commande prédictive intrinsèquement mal définis. Un exemple très éloquent est celui d'un système SISO de gain unité pour lequel on doit satisfaire des contraintes incompatibles dans ce contexte : $u \leq 1$ et $y \geq 2$ . Ce problème devient beaucoup plus préoccupant en présence de perturbations car même si les contraintes sont compatibles, une erreur de mesure sur la sortie par exemple peut invalider la loi de commande MPC/GPC.

Cette infaisabilité est facile à contourner avant l'application effective car elle peut être détectée par une simple évaluation du domaine faisable au moment initial. Une méthode classique consiste à détecter les limites pour tous les signaux devant satisfaire des contraintes et dans le cas où ces limites sont incompatibles signaler l'infaisabilité.

> **Observation 5.3 :** L'infaisabilité issue d'une incompatibilité des contraintes opérationnelles n'est pas directement liée aux signaux de référence car celles-ci sont en général indépendantes de la trajectoire à suivre. Malgré tout, dès lors que le critère prédictif contient des termes liés à la consigne à suivre, l'évolution du système est indirectement mêlée à cette infaisabilité. ■

Si l'on utilise une approche géométrique, le domaine faisable peut être évalué comme un polyèdre paramétré constitué par le jeu de contraintes opérationnelles :

$$\left\{x \middle| A_{eq}x = B_{eq}; A_{in}x \leq b_{in}\right\} \text{ ou } \left\{[x \quad u]^T \middle| [C \quad D][x \quad u]^T \leq \gamma\right\} \tag{5.5}$$

Si dans la représentation duale il n'existe aucun sommet, alors on peut décider que la loi GPC est infaisable dès la première itération. Sinon, on détient un premier domaine faisable qui sera ensuite traité par les contraintes terminales.

**Exemple 5.4**
Considérons le système suivant :

$$\begin{aligned}
&H(q^{-1}) = \frac{4q^{-1} + 6.743q^{-2}}{1 - 0.82q^{-1} - 0.16q^{-2} - 0.017q^{-3}} \\
&-1 \leq \Delta u_i \leq 0.1 \quad 1 \leq i \leq N_u \\
&-1 \leq u_i \leq 1 \quad 1 \leq i \leq N_u \\
&-5 \leq y_i \leq 5 \quad N_1 \leq i \leq N_2
\end{aligned} \tag{5.6}$$

Pour des conditions initiales nulles, le vecteur des variables contraintes étant $\theta = [\Delta u_1 \ldots \Delta u_{N_u} \; y_{N_1} \ldots y_{N_2}]$ , on peut construire la représentation duale du domaine faisable. Pour $N_u = 3$ , $N_1 = 1$ , $N_2 = 6$ , cette représentation est donnée par la *couverture convexe* des six points de la Figure 5.3.

$\theta_1$=[32 32 -139 128 576 628 -186 -893 -1600]/320
$\theta_2$=[32 32 -39 128 576 1028 1214 1407 1600]/320
$\theta_3$=[-22 49 49 -88 -112 376 1109 1815 2450]/490
$\theta_4$=[40 -31 40 160 436 646 1127 1528 2000]/400
$\theta_5$=[-122 49 49 -488 -1512 -1924 -2091 -2185 -2450]/490
$\theta_6$=[40 -131 40 160 36 -754 -1173 -1672 -2000]/400

**Figure 5.3 : Couverture convexe de l'exemple 5.4**

Le même système soumis à des contraintes plus rudes $2 \leq y_i \leq 5 \quad N_1 \leq i \leq N_2$ hérite d'un domaine faisable vide correspondant à une situation d'infaisabilité, indépendamment de la consigne à suivre.

> **Remarque 5.5 :** L'inspection de la faisabilité du domaine initial est loin de fournir un 'certificat de faisabilité'. Un autre cas d'infaisabilité, cette fois plus caché et potentiellement plus dangereux se manifeste pour les systèmes faisables pendant les premiers pas d'échantillonnage mais qui deviennent infaisables une fois que les paramètres du problème d'optimisation multiparamétrique arrivent dans des zones où la loi de commande n'est pas définie (et est donc infaisable). Comme on a pu le voir pour l'exemple 5.2, l'infaisabilité peut apparaître plus tardivement dans l'évolution du système bouclé due à

l'évolution des paramètres du problème d'optimisation dans des zones qui compromettent la compatibilité des contraintes. ■

Le résultat le plus avancé que l'on puisse obtenir est un domaine de l'espace des paramètres pour lequel on est assuré qu'une évolution future ne peut pas amener l'infaisabilité, ce que l'on appelle *l'invariance* et plus précisément *l'invariance positive*..

**Remarque 5.6 :** Ce problème suffisamment compliqué se complexifie encore un peu plus si l'on considère que toute l'analyse doit inclure aussi l'évolution des signaux de consigne. Très souvent en effet, des changements radicaux de consigne provoquent des situations d'infaisabilité par les sauts qu'ils engendrent dans l'espace des paramètres.

### 5.1.2 Infaisabilité se manifestant lors d'ajout de contraintes terminales

On a vu que l'infaisabilité peut apparaître comme résultat d'une incompatibilité entre les contraintes opérationnelles. Mais même si leur compatibilité est assurée, on ne peut pas préjuger des propriétés de stabilité de la boucle fermée. C'est pour cette raison que l'on ajoute souvent au jeu de contraintes opérationnelles des contraintes dites 'terminales' dont le but est d'assurer la stabilité de la boucle fermée.

Dans ce cas, ce type de contraintes prend en compte explicitement les consignes ou dans le cas de représentation explicite travaille sur le système étendu. De ce fait, les contraintes terminales sont très dépendantes de la consigne à suivre, d'où la distinction à apporter entre une *consigne faisable* et une *consigne infaisable*. Il est donc en général intéressant de déterminer les limites des consignes admissibles, mais la dynamique du modèle à suivre peut avoir aussi des implications sur la faisabilité. Pour résumer, lors d'infaisabilité liée aux contraintes terminales, le signal de commande est en mesure de satisfaire les contraintes et de proposer une séquence optimale. Cependant, les trajectoires prédites vont s'éloigner de la trajectoire de référence une fois les contraintes sur la commande atteintes et en conséquence à un certain moment la loi devient infaisable car l'algorithme est dans l'impossibilité de proposer une commande satisfaisant les contraintes (d'où une erreur de poursuite).

| Horizon de contraintes | Infaisabilité contraintes opérationnelles | Infaisabilité contraintes terminales |
|---|---|---|
| $N_c = 1$ | Non | Oui |
| $N_c = 6$ | Oui | Oui |
| $N_c = 12$ | Non | Non |

**Tableau 5.1 : Infaisabilité pour l'exemple 5.2 (avec contraintes supplémentaires sur la commande dans le cas de l'infaisabilité 5.1.2)**

## *5.2 Résultats liés à la faisabilité. Lien avec la représentation explicite des lois prédictives*

Tous les résultats mentionnés dans les paragraphes suivants font intervenir les notations et notions reportées en Annexe 3 (en particulier les concepts de domaine faisable, ensemble invariant, ensemble admissible).

L'application de la commande optimale sous contraintes basée sur une philosophie en boucle fermée a longtemps été écartée car la résolution d'un problème de type Hamilton–Jacobi en ligne s'avère très contraignante. D'autres solutions ont alors émergé, par exemple en 1987, dans une très courte note, Szneier et Damborg [SD87] observent qu'une séquence optimale peut être appliquée en ligne grâce à l'équivalence entre un problème à horizon infini et un problème à horizon fini plus facilement implantable. Mais comme le meilleur est l'ennemi du bien, la diminution exagérée de l'horizon de contrainte (permettant de gagner en temps de calcul) peut cependant dégrader les performances et engendrer l'instabilité. La solution consistant alors à ajouter des contraintes terminales pour garantir la stabilité peut conduire de son côté à l'infaisabilité des lois de commandes à cause de régions d'applicabilité restreintes.

### 5.2.1 Classification des situations d'infaisabilité

Avant de poursuivre cette analyse, il convient de préciser certaines notations. Le problème (MPC) considéré par la suite ($P^{MPC}$) est défini $\forall x^0 \in X_0$ par :

$$\begin{aligned}
\Phi(x_0) = & \min_{k_u=\{u_0,u_1,u_2,\ldots\}} S(x_N) + \sum_{k=1}^{N} x_k^T Q x_k + \sum_{k=0}^{N_u-1} u_k^T R u_k \\
\text{sous les contraintes } & x_{k+1} = Ax_k + Bu_k, \forall k \geq 0 \\
& \mathrm{Cx}_k + Du_k \leq \gamma, \forall 0 \leq k \leq N_c, \gamma > 0 \\
& u_k = Kx_k, N_u \leq k \leq N \\
& x_N \in X_N
\end{aligned} \tag{5.7}$$

**Définition 5.7 :** Une loi de commande prédictive (MPC) est dite faisable par abus de notation si le problème d'optimisation associé (de type mpPO) est faisable. ■

**Définition 5.8 :** L'ensemble $X_f$, dit ensemble faisable pour un problème MPC, est représenté par tout les états initiaux $x^0$ pour lesquels une séquence de commande $k_u^O = \{u_0, u_1, \ldots, u_{N_u-1}\}$ existe pour le problème ($P^{MPC}$) :

$$X_f(N, N_u, N_c, K, P, Q, R, X_N) = \left\{ x \in \Re^n \middle| \exists k_u^O(x) \text{ qui satisfait } (P^{MPC}) \right\}$$ ■

**Remarque 5.9 :** Géométriquement, si l'on considère comme défini en Annexe 3 par (A3.2) que ($P^{MPC}$) est équivalent à un problème (mpPO), alors les contraintes pour chaque ($P^{MPC}$) représentent en fait un simple polyèdre paramétré :

$$\mathrm{D}(x^0) = \left\{ k_u \middle| A_{in} k_u \leq b_{in} + B_{in} x_0 \right\}$$

ou dans un version non paramétrée :

$$\widetilde{\mathrm{D}} = \left\{ \begin{bmatrix} k_u \\ x^0 \end{bmatrix} \middle| \begin{bmatrix} A_{in} & B_{in} \end{bmatrix} \begin{bmatrix} k_u \\ x^0 \end{bmatrix} \leq b_{in} \right\} \tag{5.8}$$

et donc l'ensemble $X_f$ représente en fait une projection de $D$ sur l'espace d'état :

$$X_f = \mathrm{Pr}_x D$$

**Notation 5.10 :** La solution de ($P^{MPC}$) est donc notée $k_u^O(x)$ et la trajectoire correspondante $\chi^O(x)$. Le *kème* élément de cette séquence sera noté : $u_k = k_u^O(x,k)$ et $x_k = \chi^O(x,k)$.

**Notation 5.11 :** La loi MPC consiste en l'application itérative de la solution ($P^{MPC}$) telle que :

$$\begin{aligned}
& k_u^{MPC}(x^0) = \left\{ k_u^O(x_0,0), k_u^O(x_1,0), k_u^O(x_2,0), \ldots \right\} \\
& \chi^{MPC}(x^0) = \{x_0, x_1, x_2, \ldots\}, \\
& x_0 = x^0, x_k = \chi^O(x_{k-1},1), \forall k > 0
\end{aligned} \tag{5.9}$$

En se basant sur les définitions antérieures et en se rapportant aux relations entre l'ensemble des points initiaux $X_0$, l'espace faisable $X_f$ et leurs propriétés, on peut caractériser le phénomène d'infaisabilité d'une loi MPC :

**CLASSE I :**

$X_0 - X_f \neq 0$. Pour tout $x^0 \notin X_f$ il n'existe aucune séquence de commande $k_u^O(x^0)$ et la loi prédictive est infaisable.

**Exemple 5.12 :** Considérons le système :

$$x_{k+1} = 2x_k + u_k$$

soumis aux contraintes : $\begin{cases} -1 \le x_k \le 1 \\ -1 \le u_k \le 1 \end{cases}$

($P^{MPC}$) avec $N = N_u = N_c = 3, K = 0, P = Q = R = 1, X_N = [-0,5;0,5]$ donne $X_f = [-1;-0,969] \cup [0,969;1]$ et donc si l'on choisit $X_0 = [-1;1]$, l'infaisabilité se manifeste pour tout $x^0 \in X_0 - X_f = [-1;-0,969] \cup [0,969;1]$.

La mise en évidence de ce type d'infaisabilité se fait par le calcul de $X_f$. Si ce calcul s'avère peu évident (horizon de contraintes important, $N_u \ne N$) et si $X$ est borné, il suffit alors de vérifier la faisabilité du problème ($P^{MPC}$) pour les valeurs extrêmes de $X$.

**CLASSE II :**

$X_0 = X_f$. La loi MPC est infaisable pour l'ensemble $X_0$ si et seulement si $X_f$ n'est pas positif invariant par rapport à la loi MPC (donc s'il existe au moins un point $x^0 \in X_0$ tel que $\chi^O(x^0,1) \notin X_f$).

**Exemple 5.13 :** Considérons le modèle du double intégrateur :

$$\mathrm{x}_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u_k, k \ge 0,$$

$$u_k \in U = \{u \in \Re \mid -1 \le u \le 1\}, x_k \in X_0 = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| -1 \le x_1 \le 1; -1 \le x_2 \le 1 \right\}$$

et une loi MPC obtenue avec $N = N_u = N_c = 2, R = 10, P = Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X_N = \Re^2$.

Pour $X_0 = X_f = conv\left\{ \begin{matrix} 1 \\ 0.5 \end{matrix}; \begin{matrix} -1 \\ 1 \end{matrix}; \begin{matrix} 0.5 \\ 1 \end{matrix}; \begin{matrix} 1 \\ -1 \end{matrix}; \begin{matrix} -0.5 \\ -1 \end{matrix}; \begin{matrix} -1 \\ -0.5 \end{matrix} \right\}$ le système bouclé par la loi MPC est infaisable car, par rapport à cette loi de commande, l'ensemble $X_f$ n'est pas positif invariant. Ce fait est illustré Figure 5.4 où l'évolution d'un point $x^0 \in X_f$ est représentée. La trajectoire démontre qu'il existe $\chi^O(x^0,1) \notin X_f$ de sorte que l'invariance positive n'est pas vérifiée.

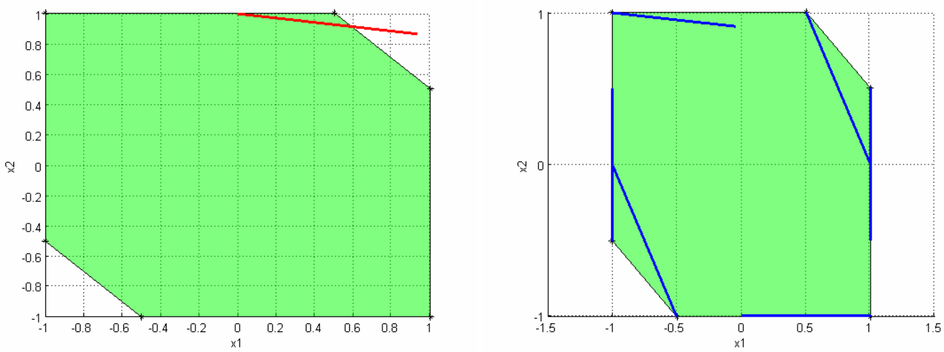



**Figure 5.4 : a) $X_f$ et l'évolution d'un point illustrant l'infaisabilité**
**b) $X_f$ et l'évolution de ses point extrêmes**

La mise en évidence de ce type d'infaisabilité se fait par le calcul de l'ensemble $X_f$ et la vérification de son caractère positif invariance. La loi $k_u^O(x)$ n'étant que linéaire par morceaux, vérifier l'évolution des sommets de $X_f$ ne suffit pas. Par ailleurs, la vérification de l'évolution de tous les pavées de $X_f$ ayant des lois linéaires s'avére une tâche difficile. Pour ces configurations, l'apport de la formulation explicite de la loi MPC est tout à fait bénéfique.

**CLASSE III :**

$X_0 \subset X_f$. La loi MPC est infaisable si et seulement s'il n'existe aucun ensemble positif invariant par rapport à la loi MPC - $IF$, qui satisfait $IF \subset X_f$ et $X_0 \subset IF$. Dans ce cas, il existe au moins un point $x^0 \in X_0$, t.q. $\exists k > 0$ pour lequel $\chi^{MPC}(x^0, k) \notin X_f$. [18]

**Exemple 5.14 :** Reprenons le même système et la même loi MPC que pour l'exemple 5.13 dans les deux cas :

**a)** $X_0 = conv\left\{\begin{matrix}0.5\\0.8\end{matrix};\begin{matrix}0.5\\-0.8\end{matrix};\begin{matrix}-0.5\\0.8\end{matrix};\begin{matrix}-0.5\\-0.8\end{matrix}\right\}$

Dans ce cas, la loi est infaisable car il n'existe aucun ensemble positif invariant inclus dans $X_f$ qui contient $X_0$. La Figure 5.5a illustre les deux domaines ainsi qu'une trajectoire infaisable à partir de $x^0 = \begin{bmatrix}-0.3\\-0.7\end{bmatrix} \in X_0$.

**b)** $X_0 = conv\left\{\begin{matrix}0.5\\0.5\end{matrix};\begin{matrix}0.5\\-0.5\end{matrix};\begin{matrix}-0.5\\0.5\end{matrix};\begin{matrix}-0.5\\-0.5\end{matrix}\right\}$

Dans ce cas, un ensemble positif invariant peut être construit pour prouver la faisabilité. La Figure 5.5b décrit les trois ensembles : $X$, $X_f$ et $IF = conv\left\{\begin{matrix}1\\0.5\end{matrix};\begin{matrix}1\\-0.5\end{matrix};\begin{matrix}-1\\0.5\end{matrix};\begin{matrix}-1\\-0.5\end{matrix}\right\}$

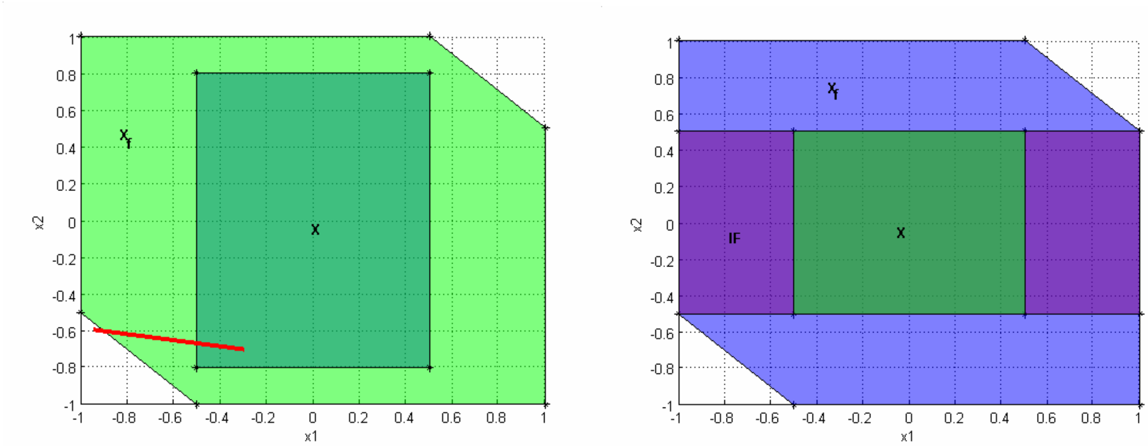



**Figure 5.5 a) $X_f - X_0 \neq 0$ mais la loi est infaisable**
**b) $X_f - X_0 \neq 0$ et $X_0 \subset IF \subset X_f$ donc la loi MPC est faisable pour $X_0$**

La mise en évidence de ce type d'infaisabilité est difficile dans la mesure où il faut démontrer l'inexistence d'un ensemble invariant, ce qui est une tâche laborieuse. Si un tel ensemble peut être construit, il constitue un certificat de faisabilité.

**Remarque 5.15 :** Si l'ensemble invariant $IF$ existe, alors en utilisant le fait qu'un ensemble invariant peut être 'contracté' (réduction proportionnelle), on peut trouver $IF_{\min}$ qui respecte les mêmes propriétés. Dès lors, si $IF_{\min}$ est compact, tout comme $X_f$, en définissant une 'métrique' adéquate pour évaluer la

[18] Pour démontrer ce fait, on procède en raisonnant par l'absurde : il suffit de considérer que la loi MPC est faisable pour tout $x^0 \in X_0$ et de considérer l'ensemble $IF$ de tous les point contenus dans les trajectoires $\chi^{MPC}(x^0)$ qui partent de $x^0 \in X_0$. Evidemment $X_0 \subset IF \subset X_f$ et $IF$ est positif invariant, ce qui contredit l'hypothèse.

distance entre ces deux ensembles, cette distance $d(IF_{\min}, X_f)$ est une mesure de la robustesse de la faisabilité pour la loi MPC par rapport à l'ensemble $X_0$.

Le dépistage des situations d'infaisabilité décrites varie donc selon les configurations. La suite des développements a pour but de montrer comment il est possible de modifier les paramètres MPC de sorte que les critères de faisabilité soient plus faciles à vérifier.

### 5.2.2 Outils pour l'analyse de faisabilité. Application pour les lois GPC sous contraintes

On ne considère dans cette partie que les problèmes d'infaisabilité de classe II, très répandus en commande GPC, pour lesquels on cherche à élaborer des procédures permettant l'analyse de la faisabilité d'une configuration donnée. Ce problème est traité dans la littérature plutôt du point de vue synthèse, voir par exemple [Ker00]. L'approche adoptée consiste à ignorer l'obtention de la loi de commande et à vérifier si elle garantit la faisabilité. Le formalisme de la représentation GPC a été privilégié car certains concepts se traduisent alors plus clairement, malgré tout, les résultats restent valables de façon générale (cas multivariable par exemple).

**Conditions nécessaires de faisabilité**

Ce paragraphe a pour objectif l'élaboration d'un point de vue purement théorique de conditions nécessaires de faisabilité, afin d'aider au positionnement des degrés de liberté de la loi GPC et de maîtriser les contraintes terminales qui sont souvent la cause de l'infaisabilité Cette démarche ne tient pas compte de la structure du domaine faisable mais plutôt de la nécessité de retrouver un horizon de prédiction pseudo infini, idée à la base de tous les résultats garantissant la faisabilité.

Soit le problème d'optimisation à résoudre à chaque *t* pour la loi GPC ( $N_c$ nombre de contraintes terminales) :

$$P(t, N, N_u, N_c, w(N+N_c)) \tag{5.10}$$

L'argument pour la solution optimale à chaque pas est noté :

$$k_u(t, N, N_u, N_c, w(N+N_c)) \tag{5.11}$$

l'élément *i* de cette séquence s'exprime par :

$$k_u(t, N, N_u, N_c, w(N+N_c))[i] \tag{5.12}$$

La proposition suivante introduit une condition nécessaire pour la faisabilité d'une loi GPC.

**Proposition 5.16 :** Si une loi GPC est faisable à chaque instant $t \geq 0$, alors l'existence de toutes les séquences suivantes est assurée :

$$k_u(0, N+k, N_u+k, N_c, w(N+N_c+k)), \forall k \geq 0 \tag{5.13}$$

ou de façon équivalente tous les problèmes suivants sont faisables :

$$P(0, N+k, N_u+k, N_c, w(N+N_c+k)), \forall k \geq 0 \tag{5.14}$$

**Preuve:** La faisabilité des lois GPC est équivalente à la faisabilité de $P(t, N, N_u, N_c, w(N+N_c))$ pour tout $t$ et à l'existence d'une solution optimale $k_u(t, N, N_u, N_c, w(N+N_c))$.

Or pour $t=0$, $P(0, N, N_u, N_c, w(N+N_c))$ est faisable et donc $k_u(0, N, N_u, N_c, w(N+N_c))$ existe et (5.13) est vérifié pour $k=0$. Supposons alors l'existence des solutions $k_u(0, N+i, N_u+i, N_c, w(N+N_c+i))$, $i = 0 \cdots k-1$. A partir des optimisations jusqu'à $k-1$, la séquence suivante peut être construite :

$$\aleph = \{k_u(0, N, N_u, N_c, w(N+N_c))[1], \ \ldots\ldots, k_u(k-1, N, N_u, N_c, w(N+N_c))[1]\} \tag{5.15}$$

qui représente la séquence des $k$ premières commandes GPC. Or, à l'instant $k$, par hypothèse, la loi GPC sous contrainte définit un problème d'optimisation faisable $P(k, N, N_u, N_c, w(N+N_c))$ de solution $k_u(k, N, N_u, N_c, w(N+N_c))$. En additionnant cette séquence à la séquence déjà existante $\aleph$, on obtient une suite de $N_u+k$ commandes :

$$\overline{\aleph} = \{\aleph, k_u(k, N, N_u, N_c, w(N+N_c))\} \tag{5.16}$$

Chaque élément de ce vecteur satisfait les contraintes opérationnelles de $P(0,N+k,N_u+k,N_c,w(N+k+N_c))$ car son jeu de contraintes est inclus dans l'union de toutes les contraintes sur lesquelles les éléments de $\overline{\aleph}$ sont construits. Alors, comme la partie finale de $\overline{\aleph}$ est $k_u(k,N,N_u,N_c,w(N+N_c))$ satisfaisant les contraintes terminales $P(k,N,N_u,N_c,w(N+N_c))$, identiques à celles de $P(0,N+k,N_u+k,N_c,w(N+k+N_c))$, on peut conclure que $\overline{\aleph}$ est donc une solution faisable de ce problème, d'où l'existence de $k_u(0,N+k,N_u+k,N_c,w(N+N_c+k))$. ■

**Exemple 5.17 :** Pour illustrer le résultat précédent, considérons le système du deuxième ordre :

$$(1-q^{-1}+0.25q^{-2})y(t)=u(t) \tag{5.17}$$

Une loi GPC sans contraintes avec les horizons $N_1=1$, $N_2=4$, $N_u=2$ permet d'asservir sans difficulté ce système pour une consigne en créneaux d'amplitude 0,3 puis 1. Les performances de cette commande sont maintenues même si des contraintes sur l'incrément de commande sont ajoutées : $-0.1\le \Delta u\le 0.1$ (Figures 5.6a et b). Des difficultés apparaissent quand les contraintes terminales sont ajoutées ( $N_c=1$, Figures 5.6c et d). Les deux types de contraintes sont incompatibles même si pour chaque type de contraintes pris séparément, la commande reste faisable (les Figures 5.6e et f représentent le comportement de la loi GPC avec contraintes terminales seules). Pour le cas mixte, le second front ascendant de la consigne nécessite une commande importante afin de satisfaire les contraintes terminales et la loi de commande devient infaisable. L'infaisabilité apparaît seulement à la 12$^{\text{ème}}$ itération, elle pouvait être détectée en observant que le problème $P(0,N+12,N_u+12,N_c,w(N+13))$ est infaisable et donc que la loi GPC est infaisable.

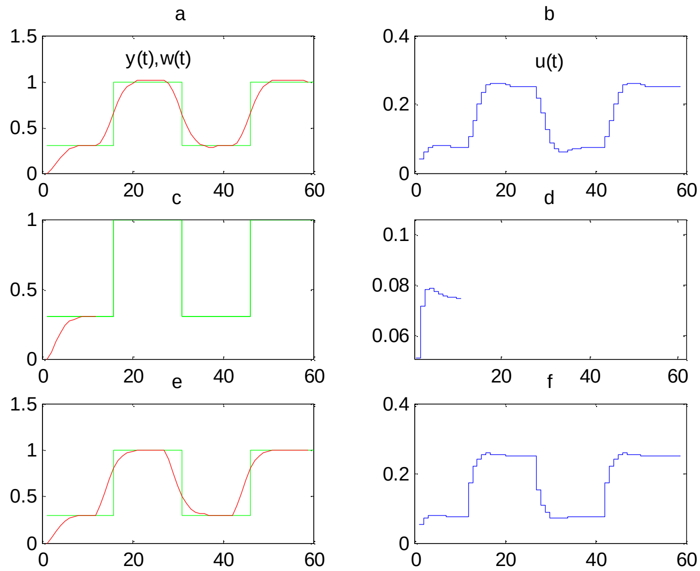



**Figure 5.6 : a, b) Sortie, consigne et entrée pour la loi GPC**
**c, d) GPC avec contraintes mixtes ; e, f) GPC avec contraintes terminales seulement**

En général, si pour un jeu de paramètres GPC il existe un $k$ tel que $P(0,N+k,N_u+k,N_c,w(N+k+N_c))$ est infaisable, alors il existe un $t$ tel que $P(t,N,N_u,N_c,w(N+N_c))$ est aussi infaisable. En pratique, toutes les combinaisons $(N,N_u,N_c)$ induisant cette caractéristique doivent être évitées. Avec ces principes, une procédure avec un nombre d'itérations garanti permettant de vérifier les conditions nécessaires n'est pas réalisable. Ces conditions peuvent être vérifiées pour des cas limites (valeurs critiques de *k*), et surtout pour des consignes de structures particulières (voir [OD03] pour des consignes de type échelon).

Partant de ce constat, une approche alternative est proposée ci-dessous afin de définir des conditions nécessaires de faisabilité, implantables de façon réaliste, se basant sur la représentation duale du domaine polyédral issu des contraintes.

***Domaine polyédral défini par les contraintes***

Les contraintes associées à un système asservi ont été longuement détaillées lors du chapitre 2, puis regroupées sous la forme compacte donnée par la relation (2.35). Si l'on considère par la suite que l'origine fait partie du domaine, et si l'on intègre les contraintes égalité au sein des contraintes inégalité, la relation donnant les contraintes peut s'écrire sous la forme :

$$-b_{in_{\min}} + B_{in}\mathbf{p} \leq A_{in}\mathbf{k}_u \leq b_{in_{\max}} + B_{in}\mathbf{p} \tag{5.18}$$

où les vecteurs $b_{in_{\min}}, b_{in_{\max}}$ ne contiennent que des termes positifs et **p** représente le vecteur des paramètres de contexte comme défini au paragraphe 2.2.8. Cette relation devient encore :

$$-b_{in_{\min}} \leq \begin{bmatrix} A_{in} \\ -B_{in} \end{bmatrix} [\mathbf{k}_u \quad \mathbf{p}] \leq b_{in_{\max}} \tag{5.19}$$

de sorte que le domaine défini par les contraintes associées au système asservi s'écrive sous forme compacte après restructuration du vecteur des paramètres de contexte :

$$\underbrace{\begin{bmatrix} \mathbf{F} \\ -\mathbf{F} \end{bmatrix}}_{\tilde{\mathbf{F}}} \boldsymbol{\theta}(t) \leq \underbrace{\begin{bmatrix} \boldsymbol{\Gamma}_{\max} \\ \boldsymbol{\Gamma}_{\min} \end{bmatrix}}_{\boldsymbol{\Gamma}}; \ \boldsymbol{\Gamma} > \mathbf{0}; \boldsymbol{\theta}(t) = [\mathbf{y}_{past}(t) \quad \mathbf{u}_{past}(t) \quad \mathbf{k}_u(t) \quad \mathbf{w}(t)]^T \tag{5.20}$$

De point de vue géométrique ce jeu d'inégalités représente en fait la version non paramétrée de l'espace faisable. Il définit donc la version étendue de l'ensemble faisable appelé auparavant $X_f$. Du point de vue GPC, cette description du jeu de contraintes est dépendante de l'horizon de prédiction $N$, de l'horizon de commande $N_u$ et des contraintes terminales (définissant $X_N$). Ce domaine peut être représenté par sa forme duale :

$$\begin{aligned} P &= \{\boldsymbol{\theta} | \tilde{\mathbf{F}}\boldsymbol{\theta} \leq \boldsymbol{\Gamma}\} = conv.hull\{\theta_1, \ldots, \theta_v\} + cone\{y_1, \ldots, y_r\} + lin.spaceP = \\ &= \left\{\boldsymbol{\theta} \middle| \ \boldsymbol{\theta} = \sum_{i=1}^{v} \lambda_i \theta_i + \sum_{i=1}^{r} \gamma_i y_i + \sum_{i=1}^{l} \mu_i z_i ; 0 \leq \lambda_i \leq 1, \ \sum_{i=1}^{v} \lambda_i = 1, \gamma_i \geq 0 \ , \forall \mu_i \right\} \end{aligned} \tag{5.21}$$

Classiquement, l'ensemble associé aux lois GPC est en fait un polytope. La méthode développée fait appel plus précisément à sa projection sur le sous-espace des paramètres de contexte **p**, noté $\boldsymbol{\theta}^*$ pour la suite :

$$\boldsymbol{\theta}^*(t) = [\mathbf{y}_{past}(t) \quad \mathbf{u}_{past}(t) \quad \mathbf{w}(t)]^T = \Pr_{[\mathbf{y}_{past} \quad \mathbf{u}_{past} \quad \mathbf{w}]} \boldsymbol{\theta}(t) \tag{5.22}$$

En exploitant la dynamique du modèle CARIMA et les relations définies au chapitre 2, on peut écrire :

$$\boldsymbol{\theta}^*(t+1) = \boldsymbol{\Phi}^* \ \boldsymbol{\theta}(t) \tag{5.23}$$

où la matrice d'évolution $\boldsymbol{\Phi}^*$ a une structure complètement détaillée dans [OD04b], non reprise ici pour alléger la démarche.

***Conditions nécessaires de faisabilité par analyse des points extrêmes***

Les conditions nécessaires de faisabilité de la loi prédictive peuvent se déduire en se basant sur l'évolution des points extrêmes du domaine faisable.

**Proposition 5.18 :** Les séquences optimales de commande correspondant à chaque combinaison extrême des paramètres de contexte $\boldsymbol{\theta}^*$ doivent conduire le système à l'intérieur du domaine faisable pour la loi GPC.

**Preuve:** Comme on l'a vu auparavant les contraintes auxquelles obéit la loi GPC définissent un polytope :

$$D(\boldsymbol{\theta}^*(t)) = \left\{\mathbf{k}_u(\boldsymbol{\theta}^*(t)) \middle| \mathbf{k}_u(\boldsymbol{\theta}^*(t)) = \sum_{i=1}^{v} \lambda_i k_{ui}(\boldsymbol{\theta}^*(t)); \lambda_i \geq 0, \ \sum_{i=1}^{v} \lambda_i = 1 \right\} \tag{5.24}$$

Ce polyèdre paramétré peut être étendu à un polyèdre non paramétré dans un espace de dimension augmentée. En termes de générateurs, celui-ci est décrit par :

$$P = \left\{\boldsymbol{\theta} \middle| \ \boldsymbol{\theta} = \sum_{i=1}^{v} \lambda_i \boldsymbol{\theta}_i ; \lambda_i \geq 0, \ \sum_{i=1}^{v} \lambda_i = 1 \right\} \tag{5.25}$$

L'existence des sommets ne garantit pas le fait que le système asservi par la loi GPC va les atteindre. Plusieurs sommets peuvent correspondre aux mêmes paramètres de contexte. En conséquence, une manipulation utile est la projection de ce domaine sur le sous-espace des paramètres de contexte (Figure 5.7).

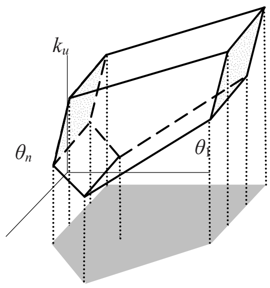



**Figure 5.7 Domaine faisable et sa projection sur le sous-espace des paramètres de contexte**

Cette opération se fait explicitement en multipliant chaque sommet $\theta_i$ par la matrice $\left[e_j^T\right]$ où $j$ sont les indices des paramètres de contexte dans le vecteur $\boldsymbol{\theta}$ et $e_j$ sont les vecteurs orthonormés de la base standard euclidienne. L'ensemble résultant noté $P^*$ est la couverture convexe de ces points [19] :

$$P^* = conv\left\{\boldsymbol{\theta}^* \middle| \; \boldsymbol{\theta}^* = \left[e_j\right]\theta_i\right\} \tag{5.26}$$

Les points extrêmes de ce domaine peuvent être utilisés pour définir l'ensemble des paramètres de contexte faisables. Pour chaque point de l'ensemble $P^*$, une séquence de commandes existe. On peut alors représenter toutes ces combinaisons de paramètres de contexte avec leurs séquences de commande optimale sous la forme d'une hypersurface (Figure 5.8) dans l'espace étendu $P$. On a vu dans les chapitres précédents (théorème 4.9) que cette hypersurface est linéaire par morceaux.

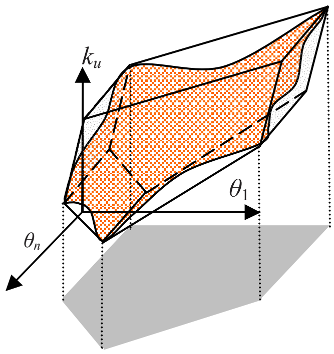



**Figure 5.8 : Solution GPC optimale pour chaque combinaison possible des paramètres de contexte**

L'évolution des paramètres de contexte va définir l'ensemble convexe :

$$P_+^* = \left\{\boldsymbol{\theta}^* \middle| \boldsymbol{\theta}^*(t) = \boldsymbol{\Phi}^* \begin{bmatrix} \boldsymbol{\theta}^*(t-1) \\ k_u(\boldsymbol{\theta}^*(t-1)) \end{bmatrix}; \boldsymbol{\theta}^*(t-1) \in P^* \right\} \tag{5.27}$$

Si l'on revient à la définition de l'infaisabilité de classe II, celle-ci se résume par la relation :

$$P^* \supset P_+^* \tag{5.28}$$

Sans description explicite de la structure de loi GPC, il faut se contenter d'évaluer l'évolution des sommets de $P^*$.

**Les conditions nécessaires sont donc résumées par** :

$$\boldsymbol{\theta}^*(t+1) \in P_+^* \tag{5.29}$$

[19] De point de vue implémentation, retenir les coordonnées du sous-espace de projection risque de produire un jeu de points redondants. Une procédure d'élimination de la redondance doit être réalisée pour ces sommets.

La Figure 5.9 présente une illustration de ce résultat.

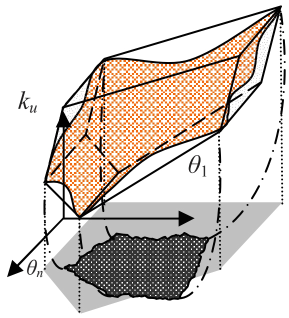



**Figure 5.9: Evolution des points extrêmes de paramètres de contexte**

> **Remarque 5.19 :** La loi GPC est une loi non linéaire (même si elle est en fait linéaire par morceaux), on ne peut de ce fait pas conclure à partir des conditions énoncées sur la faisabilité de la loi pour tout l'ensemble des paramètres de contexte. Dès lors, les propriétés énoncées constituent des conditions nécessaires mais pas suffisantes.

En suivant les étapes de la démonstration, un algorithme simple peut être mis en place, basé sur les procédures de l'analyse polyédrale et la programmation quadratique, pour vérifier la validité des conditions nécessaires.

**Algorithme 5.20**

1. Calculer les sommets de $P$ par la représentation duale du jeu des contraintes dans un espace de dimension étendue.
2. Projeter le polytope sur le sous-espace $\boldsymbol{\theta}^*$ et déterminer $P^*$.
3. Supprimer les points redondants.
4. Calculer la séquence optimale pour les sommets de $P^*$. (cette étape est indispensable car à un sommet de $P^*$ ne correspond pas un seul sommet de $P$ dans le cas général).
5. Pour chaque point ($\boldsymbol{\theta}_i^*(t), i = 1,..,\nu$), calculer les points correspondants ($\boldsymbol{\theta}_i^*(t+1)$) dans $P_+^*$ à l'aide de la matrice d'évolution.
6. Vérifier si chaque point $\boldsymbol{\theta}_i^*(t+1) \in P^*$. Si n'est pas le cas, on a trouvé un point qui contredit les conditions nécessaires et donc la loi GPC est infaisable pour le jeux de contraintes donné.

**Exemple 5.21 :** Considérons de nouveau l'exemple 5.2 d'un système du second ordre à déphasage non minimal [OD03] dont la réponse indicielle en boucle ouverte est donnée Figure 5.10.

$$(1 - q^{-1} + 0.25q^{-2})y(t) = (-0.25 - 0.25q^{-1} + 0.75q^{-2})u(t) \tag{5.30}$$

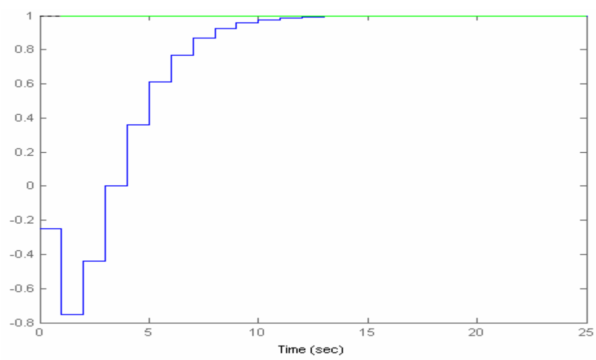



**Figure 5.10 : Réponse en boucle ouverte à une consigne en échelon**

Pour une loi GPC sous contraintes obtenue avec $N_1 = 1$, $N_2 = 4$, $N_u = 2$, le système bouclé a une trajectoire conduisant la commande à une situation d'infaisabilité (Figure 5.11) pour une consigne en échelon avec des contraintes sur l'amplitude de la sortie $-1 \le y \le 1$.

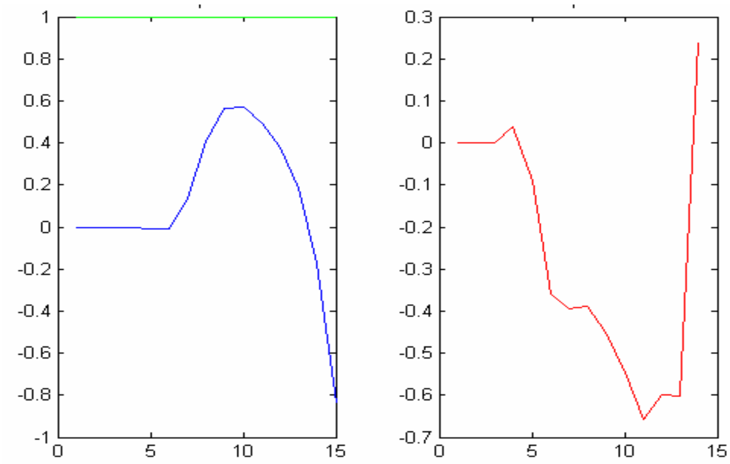



**Figure 5.11 : Comportement du système asservi par la loiGPC**
**a) Sortie en fonction du temps (s) b) Commande en fonction du temps (s)**

En appliquant l'algorithme 5.20, le domaine décrit par les contraintes est :

$$D = \left\{ \mathbf{k}_u \middle| \mathbf{1}\underline{y} \le \mathbf{G}\,\mathbf{k}_u + \mathbf{l} \le \mathbf{1}\overline{y} \right\} \tag{5.31}$$

où avec les notations du chapitre 2, **l** est la réponse libre et G la matrice formée des coefficients de la réponse indicielle du modèle :

$$\mathbf{G} = \begin{bmatrix} -0.25 & 0 \\ -0.75 & -0.25 \\ -0.40 & -0.80 \\ 0 & -0.40 \end{bmatrix} \tag{5.32}$$

Le polytope dans l'espace étendu a la forme :

$$P = \left\{ -\mathbf{\Gamma}_{\min} \le \mathbf{F}\ \mathbf{\theta}(t) \le \mathbf{\Gamma}_{\max} \quad \mathbf{\Gamma}_{\min}, \mathbf{\Gamma}_{\max} \ge \mathbf{0} \right\} \tag{5.33}$$

avec : $\mathbf{\Gamma}_{\min}{}^{\mathrm{T}} = \mathbf{1}^{\mathrm{T}}\,\underline{y} = \mathbf{1}^{\mathrm{T}}; \mathbf{\Gamma}_{\max}{}^{\mathrm{T}} = \mathbf{1}^{\mathrm{T}}\,\overline{y} = \mathbf{1}^{\mathrm{T}}$

$$\mathbf{F} = \begin{bmatrix} 2 & -1.25 & 0.25 & -0.25 & 0.75 & -0.25 & 0 \\ 2.75 & -2.25 & 0.5 & 0.25 & 1.5 & -0.75 & -0.25 \\ 3.2 & -3 & 0.7 & 0.8 & 2 & -0.4 & -0.8 \\ 3.6 & -3.4 & -0.8 & -1.2 & -2.4 & 0 & 0.4 \end{bmatrix}; \mathbf{\theta}(t) = \begin{bmatrix} \mathbf{y}_{past}(t) \\ \Delta\mathbf{u}_{past}(t) \\ \mathbf{k}_u(t) \end{bmatrix}$$

Trois contraintes implicites sont ajoutées sur les sorties passées et le résultat est un domaine $P$ possédant 128 sommets. Leurs projections sur le sous-espace des paramètres de contexte $\mathbf{\theta}^*(t) = \left[\mathbf{y}_{past}(t)\ \ \Delta\mathbf{u}_{past}(t)\right]^T$ définissent un ensemble $P^*$ qui peut être réduit par élimination des sommets n'intervenant pas dans la définition de la couverture convexe. Il en résulte 64 sommets. Les séquences de commande optimale de la loi GPC se calculent facilement pour chacun de ces points extrêmes. Par transformation linéaire :

$$\mathbf{\theta}^*(t+1) = \begin{bmatrix} \mathbf{y}_{past}(t+1) \\ \Delta\mathbf{u}_{past}(t+1) \end{bmatrix} = \begin{bmatrix} 2 & -1.25 & 0.25 & -0.25 & 0.75 & -0.25 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_{past}(t) \\ \Delta\mathbf{u}_{past}(t) \\ \mathbf{k}_u(t) \end{bmatrix} \tag{5.34}$$

on obtient l'évolution de tous ces points. La vérification de leur appartenance à $P^*$ conclut l'algorithme. Pour le cas étudié, 32 sommets se positionnent en dehors de l'espace faisable $P^*$, la loi GPC n'est donc pas faisable pour toutes les combinaisons dans le domaine défini par les contraintes.

**Conditions suffisantes de faisabilité**

Reprenons dans le cas GPC la description du domaine faisable pour un jeu de contraintes sur les entrées, les sorties et des contraintes terminales, comme introduit lors de l'examen des conditions nécessaires :

$$-\mathbf{\Gamma}_{\min} \leq \mathbf{F}\ \boldsymbol{\theta}(t) \leq \mathbf{\Gamma}_{\max}; \quad \mathbf{\Gamma}_{\min}, \mathbf{\Gamma}_{\max} > \mathbf{0}; \qquad \boldsymbol{\theta}(t) = \begin{bmatrix}\mathbf{y}_{past}(t) & \mathbf{u}_{past}(t) & \mathbf{k}_u(t) & \mathbf{w}(t)\end{bmatrix}^T$$

ou encore :

$$\underbrace{\begin{bmatrix}\mathbf{F} \\ -\mathbf{F}\end{bmatrix}}_{\tilde{\mathbf{F}}} \boldsymbol{\theta}(t) \leq \underbrace{\begin{bmatrix}\mathbf{\Gamma}_{\max} \\ \mathbf{\Gamma}_{\min}\end{bmatrix}}_{\mathbf{\Gamma}}; \quad \mathbf{\Gamma} > \mathbf{0}$$

On s'intéresse comme indiqué au préalable aux problèmes de classe II, pour lesquels $X_0 = X_f$. L'objectif est avant tout ici d'élaborer des conditions suffisantes pour qu'un jeu $(N, N_u, X_N)$ définisse une structure GPC faisable (et donc un ensemble $X_f$ invariant). Comme on l'a vu lors de la classification, la faisabilité dans le cas choisi revient à analyser l'invariance positive de l'ensemble. Remarquons que si de telles conditions existent, il devient alors possible de scinder les paramètres de réglage GPC, une partie garantissant la faisabilité, une autre partie (pondérations du critère) permettant d'harmoniser les performances de la loi de commande et de garantir la stabilité. Considérons la relation traduisant l'évolution du vecteur $\boldsymbol{\theta}$ :

$$\boldsymbol{\theta}(t+1) = \underbrace{\begin{bmatrix} \mathbf{\Phi}_{11} & \mathbf{\Phi}_{12} & \mathbf{\Phi}_{13} & 0 \\ 0 & \mathbf{\Phi}_{22} & \mathbf{\Phi}_{23} & 0 \\ \mathbf{\Phi}_{31}(\boldsymbol{\theta}(t)) & \mathbf{\Phi}_{32}(\boldsymbol{\theta}(t)) & \mathbf{\Phi}_{33}(\boldsymbol{\theta}(t)) & \mathbf{\Phi}_{34}(\boldsymbol{\theta}(t)) \\ 0 & 0 & 0 & \mathbf{\Phi}_{44} \end{bmatrix}}_{\mathbf{\Phi}} \boldsymbol{\theta}(t) + \begin{bmatrix} 0 \\ 0 \\ E(\boldsymbol{\theta}(t)) \\ 0 \end{bmatrix} \mathbf{\Gamma} \tag{5.35}$$

Le bloc $\mathbf{\Phi}_{44}$ correspond à la dynamique de la consigne considérée ici comme un système autonome. Rappelons que la présence de la consigne dans la formulation de l'ensemble contraint provient des contraintes de performance et des contraintes terminales s'exprimant toujours en fonction de la trajectoire à suivre. Dans certains cas particuliers, des valeurs non nulles de $\mathbf{\Phi}_{41}, \mathbf{\Phi}_{42}, \mathbf{\Phi}_{43}$ peuvent être considérées. L'évolution linéaire traduite par (5.35) est en fait déterminée en partie ($\mathbf{\Phi}_{11}, \mathbf{\Phi}_{12}, \mathbf{\Phi}_{13}, \mathbf{\Phi}_{22}, \mathbf{\Phi}_{23}$) par la structure du modèle CARIMA du système à commander. Le détail de ces sous-matrices n'est pas mentionné ici pour alléger la description, il peut être trouvé dans [OD05b]. La partie qui nous intéresse est donnée par la loi optimale GPC :

$$\mathbf{k}_u(t+1) = \begin{bmatrix}\mathbf{\Phi}_{31}(\boldsymbol{\theta}(t)) & \mathbf{\Phi}_{32}(\boldsymbol{\theta}(t)) & \mathbf{\Phi}_{33}(\boldsymbol{\theta}(t)) & \mathbf{\Phi}_{34}(\boldsymbol{\theta}(t))\end{bmatrix} \boldsymbol{\theta}(t) + E(\boldsymbol{\theta}(t))\mathbf{\Gamma} \tag{5.36}$$

Le fait que la loi soit linéaire affine par morceaux explique la présence du terme libre et la dépendance des coefficients vis-à-vis du vecteur $\boldsymbol{\theta}(t)$.

Avant d'élaborer les résultats sur les conditions de faisabilité, mentionnons au préalable un résultat classique sur l'invariance des ensemble polyédraux.

> **Théorème 5.22 [Bla99], [Hen89] :** Les propositions suivantes sont équivalentes :
>
> (i) Le domaine polyédral $D \subset \mathfrak{R}^n$ contenant l'origine est positif invariant pour la dynamique :
>
> $\boldsymbol{\theta}(t+1) = \mathbf{\Phi}\,\boldsymbol{\theta}(t)$
>
> (ii) Il existe un scalaire $\lambda \leq 1$ et une matrice non négative $H \in \mathfrak{R}^{r \times r}$ tels que :
>
> $H\ \bar{1} \leq \lambda \bar{1}$
>
> $H\,\mathbf{F} = \mathbf{F}\,\mathbf{\Phi}$

(iii) Il existe un scalaire $\lambda \leq 1$ et une matrice non négative $P \in \Re^{s \times s}$ tels :

$\bar{1}P \leq \lambda \bar{1}$

$\mathbf{\Lambda} P = \mathbf{\Phi} \, \mathbf{\Lambda}$

où $\mathbf{\Lambda}$ est une matrice formée des sommets du domaine polyédral décrit par les contraintes (5.20). La preuve se base sur le lemme de Farkas, voir [Hen89].

Le **principal résultat permettant la formulation de conditions suffisantes de faisabilité** est le suivant.

**Proposition 5.23 [OD04a], [OD05b]** : Soit un système décrit par un modèle CARIMA $\Sigma(A, B)$ et un jeu de contraintes :

$$-\mathbf{\Gamma}_{\min} \leq \mathbf{F} \; \mathbf{\theta}(t) \leq \mathbf{\Gamma}_{\max}, \;\; \mathbf{\Gamma}_{\min}, \mathbf{\Gamma}_{\max} > \mathbf{0} \; ; \;\; \mathbf{\theta}(t) = \begin{bmatrix} \mathbf{y}_{past}(t) & \mathbf{u}_{past}(t) & \mathbf{k}_u(t) & \mathbf{w}(t) \end{bmatrix}^T \qquad (5.37)$$

avec : $\mathbf{F} \in \Re^{p \times n}, \;\; p \geq n, \;\; \mathrm{rang}(\mathbf{F}) = n$ .

Si les hypothèses suivantes sont vérifiées :

**i.** Le problème d'optimisation lié à une loi GPC sous contrainte est faisable à l'instant $t = 0$ .

**ii.** Il existe au moins une loi linéaire décrite par les matrices $\mathbf{\Phi}_{31}, \mathbf{\Phi}_{32}, \mathbf{\Phi}_{33}$ et $\mathbf{\Phi}_{34}$, telle qu'une matrice carrée $\mathbf{\Theta} \in \Re^{N_u+N_u+N+N_c \times N_u+N_u+N+N_c}$ puisse être trouvée avec les propriétés :

$\mathbf{F}\,\mathbf{\Phi} - \mathbf{\Theta}\,\mathbf{F} = 0$ ($\mathbf{F}$ issue de (5.37), $\mathbf{\Phi}$ issue de (5.35))

et :

$$\begin{bmatrix} \mathbf{\Theta}^+ & -\mathbf{\Theta}^- \\ -\mathbf{\Theta}^- & \mathbf{\Theta}^+ \end{bmatrix} \begin{bmatrix} \mathbf{\Gamma}_{\max} \\ \mathbf{\Gamma}_{\min} \end{bmatrix} \leq \begin{bmatrix} \mathbf{\Gamma}_{\max} \\ \mathbf{\Gamma}_{\min} \end{bmatrix} \qquad (5.38)$$

où $\mathbf{\Theta}^+, -\mathbf{\Theta}^-$ sont des matrices formées d'éléments non négatifs et $\mathbf{\Theta} = \mathbf{\Theta}^+ - \mathbf{\Theta}^-$ ;

alors la loi GPC sous contraintes est faisable à tout instant $\mathrm{t} \geq 0$ . ■

**Preuve :** On va procéder par récurrence. Supposons que les propriétés **i.** et **ii.** soient satisfaites à l'instant $t = 0$ et donc en particulier qu'il existe une matrice invariante $\mathbf{\Theta}$ telle que (5.38) soit vérifiée. La faisabilité à $t = 0$ est assurée via les hypothèses. Supposons maintenant que la loi GPC soit faisable à l'instant *t*. Alors il existe $\mathbf{\theta}(t)$ telle que $\widetilde{\mathbf{F}}\,\mathbf{\theta}(t) \leq \mathbf{\Gamma}$. Dès lors que **ii.** est vérifiée (en particulier (5.38) est toujours vérifiée), il peut être déduit que l'ensemble décrit par $\widetilde{\mathbf{F}}\,\mathbf{\theta}(t) \leq \mathbf{\Gamma}$ est positif invariant par rapport à la dynamique :

$$\mathbf{\theta}(t+1) = \mathbf{\Phi} \;\; \mathbf{\theta}(t)$$

En interprétant la structure de $\mathbf{\Phi}$, l'existence d'au moins une séquence de commande à l'instant $t+1$ est assurée :

$$\mathbf{k}_u(t+1) = \begin{bmatrix} \mathbf{\Phi}_{31} & \mathbf{\Phi}_{32} & \mathbf{\Phi}_{33} & \mathbf{\Phi}_{34} \end{bmatrix} \mathbf{\theta}(t)$$

Comme l'ensemble faisable pour le problème d'optimisation à l'instant $t+1$ n'est pas vide, il est clair que la loi GPC résultant de cette formulation sera faisable. Il reste à démontrer qu'à cet instant futur $t+1$, la propriété **ii.** est vérifiée. La conclusion est alors immédiate car la séquence optimale trouvée par GPC se retrouve à l'intérieur de l'ensemble $\widetilde{\mathbf{F}}\,\mathbf{\theta}(t) \leq \mathbf{\Gamma}$ et donc l'hypothèse **ii.** est vérifiée comme à l'instant précédent car les matrices $\widetilde{\mathbf{F}}, \mathbf{\Phi}$ et le vecteur $\mathbf{\Gamma}$ sont invariants dans le temps. Ceci conclut la preuve. ■

Le résultat obtenu procure des conditions pour le cas général, mais la plupart des applications font intervenir des contraintes symétriques. Cette particularité permet d'assouplir les conditions précédentes.

**Corollaire 5.24 [OD04a], [OD05b]** : Soit un système décrit par un modèle CARIMA $\Sigma(A, B)$ et un jeu de contraintes :

$$-\mathbf{\Gamma}^* \leq \mathbf{F} \;\; \mathbf{\theta}(t) \leq \mathbf{\Gamma}^*, \;\; \mathbf{\Gamma}^* > \mathbf{0} \qquad (5.39)$$

où : $\mathbf{F} \in \Re^{p \times n}$, $p \geq n$, $\text{rang}(\mathbf{F}) = n$

Si les hypothèses suivantes sont vérifiées :

**i.** Le problème d'optimisation lié à une loi GPC sous contrainte est faisable à l'instant $t = 0$.

**ii.** Il existe au moins une loi linéaire décrite par les matrices $\mathbf{\Phi}_{31}, \mathbf{\Phi}_{32}, \mathbf{\Phi}_{33}$ et $\mathbf{\Phi}_{34}$, telle qu'une matrice carrée $\mathbf{\Theta} \in \Re^{N_u+N_u+N+N_c \times N_u+N_u+N+N_c}$ puisse être trouvée avec les propriétés :

$\mathbf{F\,\Phi} - \mathbf{\Theta\,F} = 0$ ($\mathbf{F}$ issue de (5.39), $\mathbf{\Phi}$ issue de (5.35))

et :

$$(|\mathbf{\Theta}| - \mathbf{I})\,\mathbf{\Gamma}^* \leq \mathbf{0} \quad \text{[20]} \tag{5.40}$$

alors la loi GPC sous contraintes est faisable à tout instant $t \geq 0$. ■

**Preuve:** L'approche est similaire à la précédente en remarquant que la condition d'invariance positive bénéficie de la symétrie des contraintes par rapport à l'origine, et utilise les résultats connus portant sur les ensembles polyédraux à faces parallèles [Bi88].

L'idée de fond de la Proposition 5.23 est que la loi GPC, qui est optimale (mais non linéaire) par rapport à un certain critère, sera faisable à chaque itération s'il existe une autre loi de commande linéaire faisable à chaque pas (Figure 5.12). Même si cette loi n'est pas optimale, elle a la propriété que l'ensemble polyédral défini par les contraintes est invariant par rapport à sa dynamique. La faisabilité GPC est assurée par le fait que quelle que soit la décision de l'instant courant, le prochain pas peut appliquer la commande qu'aurait proposée la loi invariante. Dès lors, la faisabilité est préservée, et si la loi GPC propose une autre action de commande, ceci ne peut aller que vers une amélioration des performances par rapport à l'index choisi, sans altérer la faisabilité.

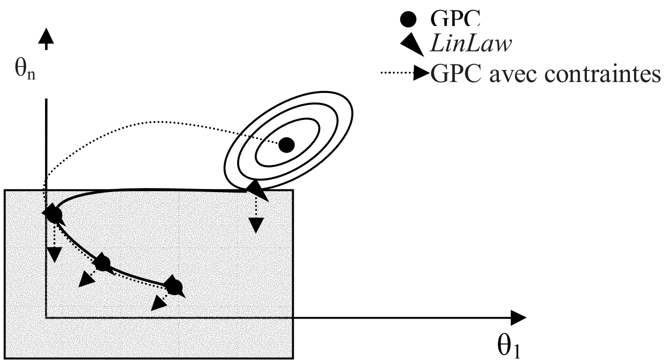



**Figure 5.12 : Evolution relative au domaine des contraintes pour 3 lois : GPC sans contraintes, GPC avec contraintes et loi faisable *LinLaw***

Le résultat peut paraître conservatif mais il faut observer que l'on cherche à trouver une loi :

$$\mathbf{k}_u(t+1) = [\mathbf{\Phi}_{31} \;\; \mathbf{\Phi}_{32} \;\; \mathbf{\Phi}_{33} \;\; \mathbf{\Phi}_{34}]\,[\mathbf{y}_{past}(t) \;\; \mathbf{u}_{past}(t) \;\; \mathbf{k}_u(t) \;\; \mathbf{w}]^{\mathrm{T}}$$

qui fait partie d'une famille beaucoup plus riche que la famille des lois de commande linéaires. Ces lois incluent en effet une partie liée à la solution optimale précédente $\mathbf{k}_u(t)$. Il faut donc qu'il existe une loi capable de prolonger cette solution faisable avec une seule commande $u(t+N_u+1)$.

**Remarque 5.25** : La satisfaction de l'hypothèse **ii.**, qui suppose en fait l'existence des matrices $\mathbf{\Phi}_{31}, \mathbf{\Phi}_{32}, \mathbf{\Phi}_{33}$ et $\mathbf{\Phi}_{34}$, revient finalement à trouver une loi de commande par retour d'état pour un système linéaire sous contraintes d'état uniquement. Réécrivons pour cela (5.35) sous la forme :

[20] La notation $|\mathbf{\Theta}|$ correspond au module élément par élément.

$$\boldsymbol{\theta}(t+1)=\underbrace{\begin{bmatrix}\boldsymbol{\Phi}_{11} & \boldsymbol{\Phi}_{12} & \boldsymbol{\Phi}_{13} & 0\\ 0 & \boldsymbol{\Phi}_{22} & \boldsymbol{\Phi}_{23} & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & \boldsymbol{\Phi}_{44}\end{bmatrix}}_{\boldsymbol{\Phi}_0}\boldsymbol{\theta}(t)+\underbrace{\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ \boldsymbol{\Phi}_{31} & \boldsymbol{\Phi}_{32} & \boldsymbol{\Phi}_{33} & \boldsymbol{\Phi}_{34}\\ 0 & 0 & 0 & 0\end{bmatrix}}_{\boldsymbol{\Phi}_v}\boldsymbol{\theta}(t) \tag{5.41}$$

qui est une caractérisation plus particulière d'un système linéaire sous contraintes :

$$\begin{cases}\boldsymbol{\theta}(t+1)=\left[\boldsymbol{\Phi}_0+\underbrace{B_v K_v}_{\boldsymbol{\Phi}_v}\right]\boldsymbol{\theta}(t)\\ -\boldsymbol{\Gamma}_{\min}\le \mathbf{F}\ \boldsymbol{\theta}(t)\le \boldsymbol{\Gamma}_{\max}\end{cases} \tag{5.42a,b}$$

De là, en se basant sur les résultats de la Proposition 5.23, une forme d'invariance positive par rapport aux commandes peut être déduite, comme dans le théorème suivant.

**Théorème 5.26 [Bla99] :** L'ensemble $D\subset\Re^n$ est invariant par rapport aux commandes pour le système :

$$x_{k+1}=Ax_k+Bu_k$$

si et seulement si il existe $\lambda\le 1$, une matrice $P$ non négative et une autre matrice $H$ tels que :

$$A\boldsymbol{\Lambda}+BH=\boldsymbol{\Lambda}P$$

$$\bar{1}^T P\le\lambda\bar{1}^T$$

L'ensemble polyédral représenté par les contraintes (5.24b) correspond à la région des états $\boldsymbol{\theta}(t)$ où la loi de commande $\mathbf{k}_u=\boldsymbol{\Phi}_v\ \boldsymbol{\theta}(t)$ ne viole pas les contraintes. Cette région contient donc le sous-espace des conditions initiales pour lequel le problème de commande est bien posé. En conséquence, la condition **i.** de la Proposition 5.23 peut être ignorée dès lors que l'infaisabilité de classe I est évitée. Donc en fait la condition **ii**. constitue le cœur de la restriction.

Pour mieux interpréter l'idée dominante, notons qu'un choix possible pour $\boldsymbol{\Phi}_{31},\boldsymbol{\Phi}_{32},\boldsymbol{\Phi}_{33}$ et $\boldsymbol{\Phi}_{34}$ est :

$$\mathbf{k}_u(t+1)=\left[\begin{array}{cc|cccccc|c} & & 0 & 1 & 0 & \cdots & 0 & 0 & \\ & & 0 & 0 & 1 & \ddots & 0 & 0 & \\ & & 0 & \ddots & 0 & \ddots & \ddots & \vdots & \\ 0 & 0 & \vdots & \ddots & \ddots & \ddots & \ddots & 0 & 0\\ & & 0 & \cdots & \cdots & \cdots & 0 & 1 & \\ & & 0 & \cdots & \cdots & \cdots & \cdots & 0 & \\ \hline \phi_{31} & \phi_{32} & & & \phi_{33} & & & & \phi_{34}\end{array}\right]\left[\mathbf{y}_{past}(t)\ \ \mathbf{u}_{past}(t)\ \ \mathbf{k}_u(t)\ \ \mathbf{w}\right]^{\mathrm{T}}$$

ou de façon équivalente :

$$\begin{aligned}\mathbf{k}_u(t+1)&=\left[u_{t+1|t+1}\ \ u_{t+2|t+1}\ \ldots\ u_{t+N_u|t+1}\ \ u_{t+N_u+1|t+1}\right]=\\ &=\left[u_{t+1|t}\ \ u_{t+2|t}\ \ldots\ u_{t+N_u|t}\ \ \phi_{31}\mathbf{y}_{past}(t)+\phi_{32}\mathbf{u}_{past}(t)+\phi_{33}\mathbf{k}_u(t)+\phi_{34}\mathbf{w}\right]\end{aligned} \tag{5.43}$$

Avec une telle structure spécifique, il est clair qu'en fait l'existence d'une seule commande :

$$u_{t+N_u+1|t+1}=\phi_{31}\mathbf{y}_{past}(t)+\phi_{32}\mathbf{u}_{past}(t)+\phi_{33}\mathbf{k}_u(t)+\phi_{34}\mathbf{w}$$

va remplir les hypothèses nécessaires pour satisfaire les conditions suffisantes de faisabilité.

Cette structure n'est qu'un cas particulier, car, en général, les candidats validant l'hypothèse font partie d'une famille beaucoup plus riche. Alors, une fois les hypothèses de faisabilité validées, le jeu de paramètres assurant cette faisabilité peut ensuite être choisi pour améliorer les performances et assurer la stabilité.

Les développements précédents ont permis de juger de l'utilité des conditions suffisantes, mais rien n'a été dit sur la façon de vérifier les conditions mentionnées dans la Proposition 5.23. Il s'avère que ceci n'est pas une tâche aisée, nécessitant une mise en forme du problème. La partie **i.** ne requiert pas une analyse complexe. Il suffit de vérifier que l'état initial du système est faisable, ce qui revient à vérifier la faisabilité de classe I.
En revanche, la condition **ii.**, comme indiqué précédemment, doit prouver l'existence des matrices $\mathbf{\Phi}_{31}, \mathbf{\Phi}_{32}, \mathbf{\Phi}_{33}, \mathbf{\Phi}_{34}$ et $\mathbf{\Theta}$ . Même si $\mathbf{\Theta}$ peut paraître facile à déterminer à partir de la relation $\mathbf{F\Phi} - \mathbf{\Theta F} = 0$, il n'en est rien dans le cas d'une matrice $\mathbf{F}$ de forme générale. Pour cette raison, une des hypothèse mentionne que $\mathbf{F} \in \Re^{p \times n}$, $p \geq n$ et $\text{rang}(\mathbf{F}) = n$, de sorte que la pseudo-inverse existe. Ceci n'affecte pas le résultat pour des domaines faisables représentés par des polytopes, car, dans ce cas, les conditions mentionnées sont automatiquement vérifiées. Dans le cas de domaines faisables ouverts, les résultats de la proposition restent valables mais l'évaluation des conditions **ii.** devient plus difficile car l'existence de la pseudo-inverse de $\mathbf{F}$ est nécessaire.

La mise en forme suivante se base sur les conditions de l'hypothèse **ii.** du Corollaire 5.24 (contraintes symétriques), la généralisation aux matrices non symétriques se faisant de la même façon. Afin d'éliminer les inconnues et en utilisant le fait que la matrice $\mathbf{F}$ est de rang plein par colonnes, on peut remplacer :

$$\mathbf{\Theta} = \mathbf{F\Phi}(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}} \tag{5.44}$$

En séparant la partie inconnue de $\mathbf{\Phi}$ à l'aide de (5.41) :

$$\mathbf{\Theta} = \underbrace{\mathbf{F}\,\mathbf{\Phi}_0 \underbrace{(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\,\mathbf{F}^{\mathrm{T}}}_{\mathbf{D}}}_{\mathbf{\Theta}^*} + \mathbf{F}\,\mathbf{\Phi}_v \underbrace{(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\,\mathbf{F}^{\mathrm{T}}}_{\mathbf{D}} \tag{5.45}$$

Mais la matrice $\mathbf{\Phi}_v$ a une structure particulière :

$$\mathbf{\Phi}_v = \begin{bmatrix} 0 \\ \mathbf{\Phi}_{31} \;\; \mathbf{\Phi}_{32} \;\; \mathbf{\Phi}_{33} \;\; \mathbf{\Phi}_{34} \\ 0 \end{bmatrix} \tag{5.46}$$

et une partition de $\mathbf{F}$ reprenant les dimensions par bloc de $\mathbf{\Phi}_v$ peut être considérée :

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_l & \mathbf{F}^* & \mathbf{F}_r \end{bmatrix} \tag{5.47}$$

Avec ces notations (5.40) peut être réécrite :

$$\left| \mathbf{F}^* \underbrace{\begin{bmatrix} \mathbf{\Phi}_{31} & \mathbf{\Phi}_{32} & \mathbf{\Phi}_{33} & \mathbf{\Phi}_{34} \end{bmatrix}}_{\mathbf{\Phi}^*} \mathbf{D} + \mathbf{\Theta}^* \right| \mathbf{\Gamma}^* \leq \mathbf{\Gamma}^* \tag{5.48}$$

Vérifier que l'ensemble défini par ce jeu de contraintes n'est pas vide devient donc une condition impérative dans la démarche afin de déterminer les coefficients de la matrice $\mathbf{\Phi}^*$ pour valider l'hypothèse **ii.** La présence du module implique, pour simplifier l'évaluation, sachant que le vecteur $\mathbf{\Gamma}^*$ ne possède que des composantes positives, l'utilisation de procédures décrites ci-dessous.

> **Proposition 5.27 [VHB88]** : L'inégalité matricielle (5.48) avec $\mathbf{\Theta}^* \in \Re^{p \times p}$, $\mathbf{F}^* \in \Re^{p \times N_u}$, $\mathbf{\Phi}^* \in \Re^{N_u \times n}$, $\mathbf{D} \in \Re^{n \times p}$ et $\Gamma_i > 0$ est équivalente au jeu d'équations :
>
> $$\sum_{j=1}^{n} \Big( \sum_{k=1}^{N_u} (F_{ik} \Phi^*_{kj} p_{js} \sum_{q=1}^{m} (d_{jq} \Gamma_q)) \Big) \leq \Gamma_i - \sum_{r=1}^{n} \Theta^*_{ir} p_{rs} \Gamma_r \qquad i = 1..p, \qquad s = 1..2^p \tag{5.49}$$
>
> où :
>
> $$p_s = \begin{bmatrix} p_{1s} & p_{2s} & \cdots & p_{ms} \end{bmatrix}^T, \quad p_s \in \{-1,+1\}^p \tag{5.50}$$
>
> représente les combinaisons possibles de vecteurs de dimension $p$ de valeurs –1 ou +1. ■

Cette équivalence conduit à une formulation assez complexe, mais utile (car implantable) permettant de vérifier (5.40) et en conséquence d'examiner les conditions suffisantes de faisabilité pour un jeu de paramètres de la loi GPC.

> **Remarque 5.28** : les matrices qui sont traitées dans les opérations énumérées sont liées au nombre de contraintes et de variables et généralement impliquent des dimensions larges avec une certaine sensitivité numérique et donc des précautions doivent être prises pour améliorer leur conditionnement.

**Exemple 5.29 :** Considérons pour illustrer la technique d'analyse proposée le système du second ordre :

$$(1+q^{-1}+0.25q^{-2})y(t)=2.25\,u(t-1) \tag{5.51}$$

Une loi GPC synthétisée avec les horizons de prédiction $N_1=1, N_2=1, N_u=2$ et $N_c=1$ (i.e. une seule contrainte terminale implémentée par $|y(t+N_2+N_c)-w(t+N_2)|\le 0{,}001$, peut rencontrer des problèmes dans le cas sous contraintes car en boucle ouverte le système présente un dépassement important allant jusqu'à 125% pour une simple consigne en échelon. Si l'on ajoute une contrainte de type 'pas de dépassement' :

$$-1\le y\le 1 \tag{5.52}$$

une analyse de la faisabilité est fortement conseillée. Notons que l'horizon de prédiction a été volontairement diminué pour permettre une lisibilité simple des résultats, mais, dans le cas général, un horizon de prédiction plus long peut être choisi et d'autres types de contraintes peuvent être ajoutés.

La condition **i.** du Corollaire 5.24 peut être vérifiée par simple examen du domaine des conditions initiales faisables. Dans le cas de ce système, il s'agit de la couverture convexe de 32 sommets représentant un hypercube à 5 dimensions. L'origine est contenue à l'intérieur de ce domaine et donc pour notre exemple les conditions initiales nulles sont accessibles.
La principale difficulté est de trouver les matrices $\boldsymbol{\Phi}_{31}, \boldsymbol{\Phi}_{32}, \boldsymbol{\Phi}_{33}, \boldsymbol{\Phi}_{34}$ et $\boldsymbol{\Theta}$ pour satisfaire **ii.** Pour le système modélisé par (5.51), les contraintes peuvent se mettre sous la forme matricielle suivante :

$$\mathbf{F}=\underbrace{\left[\begin{array}{ccc|c|cc|c} 0 & 0.75 & 0.25 & - & 2.25 & 0 & 0 \\ \hline 0.75 & 0.25 & 0 & - & 0 & 2.25 & -1 \\ 0 & 0 & 0 & - & 0 & 0 & 1 \\ 1 & 0 & 0 & - & 0 & 0 & 0 \\ 0 & 1 & 0 & - & 0 & 0 & 0 \\ 0 & 0 & 1 & - & 0 & 0 & 0 \end{array}\right]}_{[\;\mathbf{F}_l\;|\;\mathbf{F}^*\;|\mathbf{F}_r]};\boldsymbol{\Gamma}^*=\left[\begin{array}{c} 1 \\ 0.001 \\ 1 \\ \hline 1 \\ 1 \\ \hline 1 \end{array}\right]^{\mathrm{T}} \tag{5.53}$$

traduisant des contraintes sur le vecteur :

$$\boldsymbol{\theta}(t)=\begin{bmatrix} y(t) & y(t-1) & y(t-2) & - & \Delta u(t) & \Delta u(t+1) & w(t+1)\end{bmatrix}^{\mathrm{T}} \tag{5.54}$$

On remarque que, puisque la structure du système est simple (le polynôme $B$ est de degré 0), la partie correspondant aux entrées passées dans le vecteur $\boldsymbol{\theta}$ est vide. De plus, à cause des horizons de prédiction faibles, des contraintes additionnelles sur les sorties passées doivent être ajoutées pour augmenter le rang de la matrice **F**.

La dynamique du vecteur $\boldsymbol{\theta}$ est décrite par :

$$\boldsymbol{\theta}(t+1)=\left[\begin{array}{ccc|c|cc|c} 0 & 0.75 & 0.25 & & 2.25 & 0 & 0 \\ 1 & 0 & 0 & - & 0 & 0 & 0 \\ 0 & 1 & 0 & & 0 & 0 & 0 \\ \hline & - & & - & & - & - \\ x_1 & x_2 & x_3 & - & x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 & & x_{10} & x_{11} & x_{12} \\ \hline & 0 & & - & & 0 & 1 \end{array}\right]\boldsymbol{\theta}(t) \tag{5.55}$$

En appliquant (5.45-5.48) on arrive à :

$$\left| \mathbf{F}^* \underbrace{\begin{bmatrix} x_1 & x_2 & x_3 & - & x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 & - & x_{10} & x_{11} & x_{12} \end{bmatrix}}_{[\Phi_{31} | \Phi_{32} | \Phi_{33} | \Phi_{34}]} \mathbf{D} + \mathbf{\Theta}^* \right| \mathbf{\Gamma}^* \le \mathbf{\Gamma}^*$$

avec : (5.56)

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0.44 & 0 & 0 & 0 & -0.33 & -0.11 \\ 0 & 0.44 & 0.44 & -0.33 & -0.11 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{et} \quad \mathbf{\Theta}^* = \begin{bmatrix} 0 & 0 & 0 & 0.75 & 0.25 & 0 \\ 0.75 & 0 & -1 & 0.25 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Il faut dès lors résoudre le système d'inéquations :

$$\begin{cases} |x_4| + 0.001|x_5| + |x_5 + 2.25x_6| + |0.75 + 2.25x_1 - 0.75x_5| + \\ \qquad + |0.25 + 2.25x_2 - 0.75x_4 - 0.25x_5| + |2.25x_3 - 0.25x_4| \le 1 \\ \\ |0.75 + x_{10}| + 0.001|x_{11}| + |-1 + x_{11} + 2.25x_{12}| + |0.25 + 2.25x_7 - 0.75x_{11}| + \\ \qquad + |2.25x_8 - 0.75x_{10} - 0.25x_{11}| + |2.25x_9 - 0.25x_{10}| \le 0.001 \end{cases} \tag{5.57}$$

En transformant ces inégalités en un système d'inégalités linéaires, une solution peut être trouvée en appliquant une procédure classique d'optimisation linéaire. L'existence d'une solution est importante, plus que les valeurs effectives. A titre d'information, la solution trouvée ici est la suivante :

$$\mathbf{k}_u(t) = [\mathbf{\Phi}_{31} | \mathbf{\Phi}_{32} | \mathbf{\Phi}_{33} | \mathbf{\Phi}_{34}]\boldsymbol{\theta}(t)$$

avec :

$$\mathbf{\Phi}_{31} = \begin{bmatrix} -0.0487 & -0.1428 & -0.0545 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{\Phi}_{33} = \begin{bmatrix} -0.4908 & 0.1872 \\ 0 & 0 \end{bmatrix} \quad \mathbf{\Phi}_{34} = \begin{bmatrix} 0.3612 \\ 0 \end{bmatrix} \tag{5.58}$$

Cette existence représente une preuve de la faisabilité de la loi GPC avec ce choix d'horizons de prédiction et de contraintes terminales. Le comportement en ligne du système bouclé par le correcteur GPC, représenté Figure 5.13, est la confirmation de la faisabilité en ligne. Bien sûr, il existe une famille entière de lois GPC faisables, correspondant en fait aux différents critères de coût qui peuvent être choisis par des règles classiques [BD96].

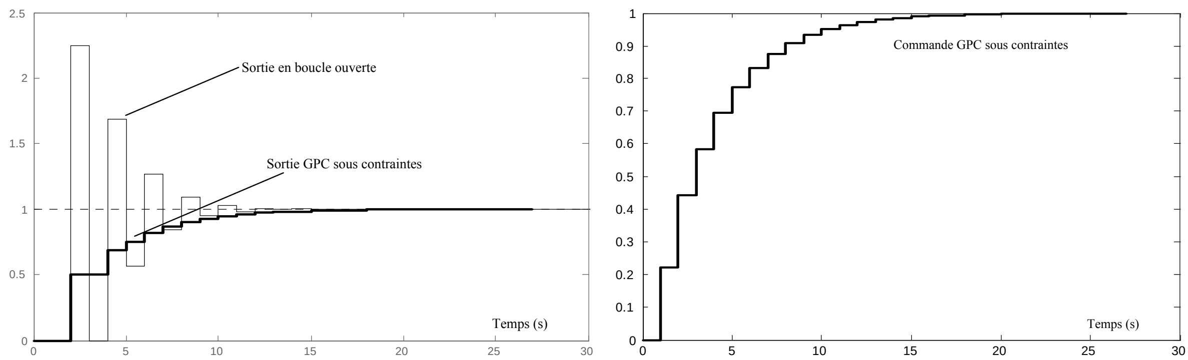



**Figure 5.13 : a) Trajectoire en boucle ouverte et trajectoire en boucle fermée b) Commande**

Des conditions suffisantes peuvent être trouvées pour d'autres choix d'horizons (par exemple $N_1 = 1, N_2 = 1, N_u = 3$ et $N_c = 2$ ) mais dans ce cas la complexité du système d'inéquations équivalent à (5.49) augmente de façon spectaculaire, il est alors nécessaire de vérifier qu'un domaine défini par 896 inégalités avec 21 variables est non vide.

**Exemple 5.30 :** Considérons le système :

$$\begin{aligned}&(1-0.2q^{-1}+1q^{-2})y(t)=u(t-1)\\&-1\leq y(t)\leq 1\end{aligned} \tag{5.59}$$

et une loi GPC avec $N_u=3, N_c=2$, et une consigne en créneaux décrite par le processus de génération suivant :

$$\mathbf{w}_c(t+1)=\underbrace{\begin{bmatrix}0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\1&0&0&0&0&0\end{bmatrix}}_{\mathbf{\Phi}_{44}}\mathbf{w}_c(t);\quad \mathbf{w}_c(0)=\begin{bmatrix}0\\0\\0\\1\\1\\1\end{bmatrix}^{\mathrm{T}} \tag{5.60}$$

Des conditions suffisantes peuvent être trouvées pour la faisabilité de la loi GPC par la même procédure. Les simulations (Figure 5.14) confirment le résultat.

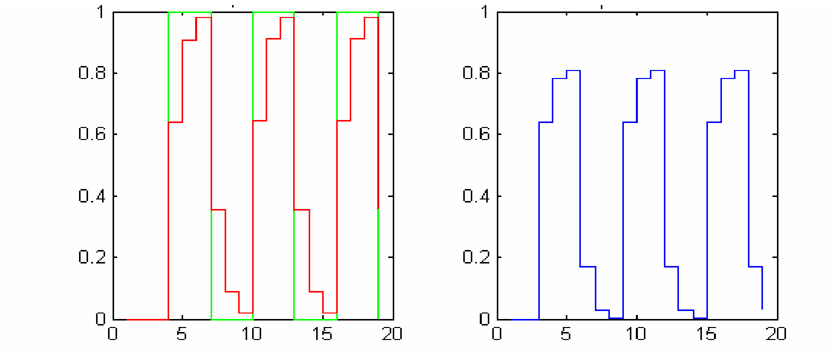



**Figure 5.14 : a) Sortie et consigne    b) Signal de commande de la loi GPC**

**Remarque fondamentale 5.31 :** Les résultats présentés ci-dessus s'appliquent en fait dans le cas général d'un système modélisé par représentation d'état sans modification. Ils ont été présentés ici dans le cas GPC avec fonction de transfert pour illustrer que ce dernier formalisme peut bénéficier également de tous les outils d'analyse liés à la faisabilité de la commande prédictive.

**Conditions nécessaires et suffisantes pour la faisabilité de classe II**

Les paragraphes précédents ont permis l'élaboration de procédures bornées permettant d'évaluer des conditions nécessaires et des conditions suffisantes de faisabilité. Leur degré de complexité est différent :

- les conditions nécessaires sont implémentées par une opération de projection, la résolution d'un nombre fini d'optimisations quadratiques et l'évaluation finale de la validité d'un jeu d'inclusions.
- Les conditions suffisantes ont une description plus compacte et sont équivalentes à la faisabilité d'un système d'inégalités. Ce jeu d'inégalités est cependant d'une complexité exponentielle en fonction du nombre de variables impliquées.

Ces deux procédures peuvent éviter si nécessaire l'évaluation relativement complexe de la forme explicite d'une loi de commande MPC. Néanmoins l'existence de la forme explicite permet l'expression de conditions nécessaires et suffisantes de faisabilité.

Pour une loi GPC, appliquant la solution optimale d'un problème d'optimisation multiparamétrique selon le principe de l'horizon glissant, la description explicite est rappelée Tableau 5.2.

Les domaines $D_i$ sont des domaines polyédraux et donc pour chacun on peut recenser les sommets :

$$\Delta_i=\{v_{i1},v_{i2},\ldots,v_{i\delta_i}\} \tag{5.61}$$

$$\begin{array}{ll} \theta_k \in D_1 & u_k^{GPC}(\theta_k) = \Psi_1 \theta_k + \psi_1 \\ \theta_k \in D_2 & u_k^{GPC}(\theta_k) = \Psi_2 \theta_k + \psi_2 \\ \dots & \dots \\ \theta_k \in D_n & u_k^{GPC}(\theta_k) = \Psi_n \theta_k + \psi_n \end{array}$$

**Tableau 5.2 : Forme générale de la description explicite d'une loi prédictive sous contraintes**

où $\delta_i$ est le cardinal de l'ensemble des sommets pour chaque $D_i$. L'union de tous ces points sera :

$$\Delta = \bigcup_{i=1}^{n} \Delta_i$$

En définissant l'ensemble :

$$\Delta^+ = \left\{ v^+ \middle| \exists v \in \Delta \ \text{ t.q. } \ v^+ = Av + Bu_k^{GPC}(v) \right\} \tag{5.62}$$

on dispose de tous les éléments permettant d'énoncer **le théorème sur les conditions nécessaires et suffisantes de faisabilité**.

**Théorème 5.32 :** Une loi de commande GPC est faisable par rapport à l'ensemble $X_f = \bigcup_{i=1}^{n} D_i$ si et seulement si $\Delta^+ \subset X_f$.

**Preuve :**

"$\Rightarrow$" : La loi prédictive étant faisable pour $X_f$ implique que pour tous les point $\theta \in X_f$ on a $\theta^+ \in \Omega_f$ par la propriété d'invariance positive de $X_f$. Comme $\Delta \subset X_f$ la conséquence immédiate est que $\Delta^+ \subset X_f$.

"$\Leftarrow$" : $\forall \theta \in X_f \Rightarrow \forall \theta \in \bigcup_{i=1}^{n} D_i \Rightarrow \exists i \quad t.q. \quad \theta \in D_i$ et donc :

$\theta = \alpha_1 v_{i1} + \alpha_2 v_{i2} + \ldots + \alpha_{\delta_i} v_{i\delta_i}; 0 \le \alpha_j \le 1; j = 1,.., \delta_i; \sum_{j=1}^{\delta_i} \alpha_j = 1$ car le domaine $D_i$ est convexe.

$$\begin{aligned} \theta \in D_i \Rightarrow \theta^+ &= (A + B\Psi_i)\theta + \psi_i = \alpha_1 (A + B\Psi_i) v_{i1} + \alpha_2 (A + B\Psi_i) v_{i2} + \ldots \\ &\qquad + \alpha_{\delta_i} (A + B\Psi_i) v_{i\delta_i} + (\alpha_1 + \alpha_2 + \ldots + \alpha_{\delta_i}) \psi_i = \\ &= \alpha_1 v_{i1}^+ + \alpha_2 v_{i2}^+ + \ldots + \alpha_{\delta_i} v_{i\delta_i}^+ \end{aligned}$$

Mais $v_{i1}^+, v_{i2}^+, \ldots, v_{i\delta_i}^+ \in X_f$ et donc $\theta^+ \in X_f$ qui prouve l'invariance positive de $X_f$. ■

**Remarque 5.33 :** Le théorème est constructif et offre une procédure finie de vérification de la faisabilité pour toute loi GPC/MPC. Le fait que cette procédure implique l'existence de la formulation explicite de la loi de commande fait que sa complexité est très dépendante de la structure effective des domaines $D_i$ et de leur ensemble de générateurs.

**Exemple 5.34 :** Considérons le système suivant :

$$\begin{aligned} &\theta_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \theta_k + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u_k^{MPC}, k \ge 0, \\ &u_k \in U = \{ u \in \Re \,|\, -1 \le u \le 1 \}, \theta_k \in X = \left\{ \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \middle| -1 \le \theta_1 \le 1; -1 \le \theta_2 \le 1 \right\} \end{aligned} \tag{5.63}$$

**Cas I** : Loi MPC avec $N = N_u = N_c = 2, R = 10, P = Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X_N = \Re^2$. La forme explicite est donnée Tableau 5.3.

| | | |
|---|---|---|
| $\theta_k \in D_1$ | $\Delta_1 = \left\{ \begin{bmatrix} -0.068182 \\ -1 \end{bmatrix}; \begin{bmatrix} 1 \\ -1 \end{bmatrix}; \begin{bmatrix} 0.068182 \\ 1 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.02381 \end{bmatrix}; \begin{bmatrix} -1 \\ 1 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.02381 \end{bmatrix} \right\}$ | $u_k^{GPC} = [-0.044444 \quad -0.13333]\theta_k$ |
| $\theta_k \in D_2$ | $\Delta_2 = \left\{ \begin{bmatrix} -0.068182 \\ -1 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.5 \end{bmatrix}; \begin{bmatrix} -0.5 \\ -1 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.02381 \end{bmatrix} \right\}$ | $u_k^{GPC} = [-2 \quad -2]\theta_k - 2$ |
| $\theta_k \in D_n$ | $\Delta_3 = \left\{ \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}; \begin{bmatrix} 0.068182 \\ 1 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.02381 \end{bmatrix}; \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \right\}$ | $u_k^{GPC} = [-2 \quad -2]\theta_k + 2$ |

**Tableau 5.3 : Description explicite de la loi prédictive sous contraintes, Exemple 5.34 Cas I**

On calcule ensuite :

$$X_f = conv\left\{ \begin{matrix} 1 \\ 0.5 \end{matrix}; \begin{matrix} -1 \\ 1 \end{matrix}; \begin{matrix} 0.5 \\ 1 \end{matrix}; \begin{matrix} 1 \\ -1 \end{matrix}; \begin{matrix} -0.5 \\ -1 \end{matrix}; \begin{matrix} -1 \\ -0.5 \end{matrix} \right\}$$

$$\Delta = \left\{ \begin{bmatrix} -0.068182 \\ -1 \end{bmatrix}; \begin{bmatrix} 1 \\ -1 \end{bmatrix}; \begin{bmatrix} 0.068182 \\ 1 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.02381 \end{bmatrix}; \begin{bmatrix} -1 \\ 1 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.02381 \end{bmatrix}; \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.5 \end{bmatrix}; \begin{bmatrix} -0.5 \\ -1 \end{bmatrix} \right\}$$

$$\Delta^+ = \left\{ \begin{bmatrix} -1 \\ -0.86364 \end{bmatrix}; \begin{bmatrix} 0.044444 \\ -0.91111 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.86364 \end{bmatrix}; \begin{bmatrix} 1 \\ -0.02381 \end{bmatrix}; \begin{bmatrix} -0.044444 \\ 0.91111 \end{bmatrix}; \begin{bmatrix} -1 \\ 0.02381 \end{bmatrix}; \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}; \begin{bmatrix} -1 \\ 0.5 \end{bmatrix}; \begin{bmatrix} -1 \\ 0 \end{bmatrix} \right\}$$

Le système bouclé par la loi MPC est donc infaisable (Figure 5.15). Sur cette figure apparaissent les partitions de la loi explicite et l'évolution de leurs sommets par application de la loi de commande. Certains points de l'espace des paramètres évoluent alors en dehors du domaine. Cette caractéristique apparaît car aucune contrainte terminale n'a été prise en compte et les pondérations mal choisies.

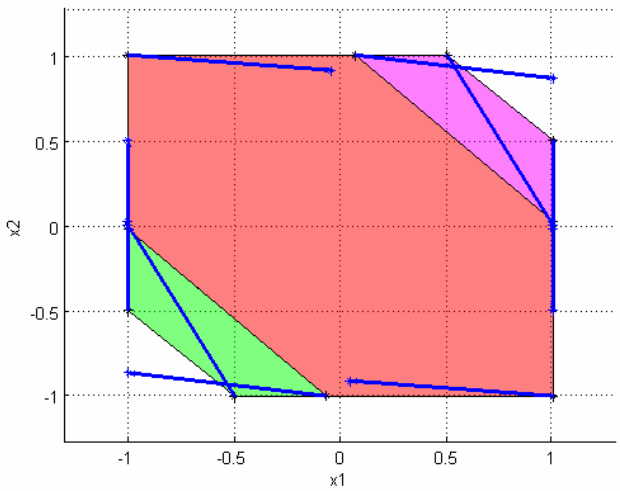



**Figure 5.15 : Vérification des conditions nécessaires et suffisantes de faisabilité, Exemple 5.34 Cas I**

**Cas II** : Loi MPC avec $N = N_u = N_c = 2, R = 1, P = Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X_N = \Re^2$. La forme explicite est donnée Tableau 5.4.

| | | |
|---|---|---|
| $\theta_k \in D_1$ | $\Delta_1 = \left\{ \begin{bmatrix} -0.375 \\ -1 \end{bmatrix}; \begin{bmatrix} 1 \\ -1 \end{bmatrix}; \begin{bmatrix} 0.375 \\ 1 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.16667 \end{bmatrix}; \begin{bmatrix} -1 \\ 1 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.16667 \end{bmatrix} \right\}$ | $u_k^{GPC} = [-0.22222 \quad -0.66667]\theta_k$ |
| $\theta_k \in D_2$ | $\Delta_2 = \left\{ \begin{bmatrix} -0.375 \\ -1 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.5 \end{bmatrix}; \begin{bmatrix} -0.5 \\ -1 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.16667 \end{bmatrix} \right\}$ | $u_k^{GPC} = [-2 \quad -2]\theta_k - 2$ |
| $\theta_k \in D_n$ | $\Delta_3 = \left\{ \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}; \begin{bmatrix} 0.375 \\ 1 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.16667 \end{bmatrix}; \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \right\}$ | $u_k^{GPC} = [-2 \quad -2]\theta_k + 2$ |

**Tableau 5.4 : Description explicite de la loi prédictive sous contraintes, Exemple 5.34 Cas II**

On calcule ensuite :

$$X_f = conv\left\{ \begin{matrix} 1 \\ 0.5 \end{matrix}; \begin{matrix} -1 \\ 1 \end{matrix}; \begin{matrix} 0.5 \\ 1 \end{matrix}; \begin{matrix} 1 \\ -1 \end{matrix}; \begin{matrix} -0.5 \\ -1 \end{matrix}; \begin{matrix} -1 \\ -0.5 \end{matrix} \right\}$$

$$\Delta = \left\{ \begin{bmatrix} -0.375 \\ -1 \end{bmatrix}; \begin{bmatrix} 1 \\ -1 \end{bmatrix}; \begin{bmatrix} 0.375 \\ 1 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.16667 \end{bmatrix}; \begin{bmatrix} -1 \\ 1 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.16667 \end{bmatrix}; \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}; \begin{bmatrix} -1 \\ -0.5 \end{bmatrix}; \begin{bmatrix} -0.5 \\ -1 \end{bmatrix} \right\}$$

$$\Delta^+ = \left\{ \begin{bmatrix} -1 \\ -0.25 \end{bmatrix}; \begin{bmatrix} 0.22222 \\ -0.55556 \end{bmatrix}; \begin{bmatrix} 1 \\ 0.25 \end{bmatrix}; \begin{bmatrix} 1 \\ -0.16667 \end{bmatrix}; \begin{bmatrix} -0.22222 \\ 0.55556 \end{bmatrix}; \begin{bmatrix} -1 \\ 0.16667 \end{bmatrix}; \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}; \begin{bmatrix} -1 \\ 0.5 \end{bmatrix}; \begin{bmatrix} -1 \\ 0 \end{bmatrix} \right\}$$

Le système bouclé par la loi MPC est cette fois faisable (Figure 5.16). Sur cette figure apparaissent les partitions de la loi explicite et l'évolution de leurs sommets par application de la loi de commande. Tous les sommets de l'espace des paramètres évoluent à l'intérieur du domaine.

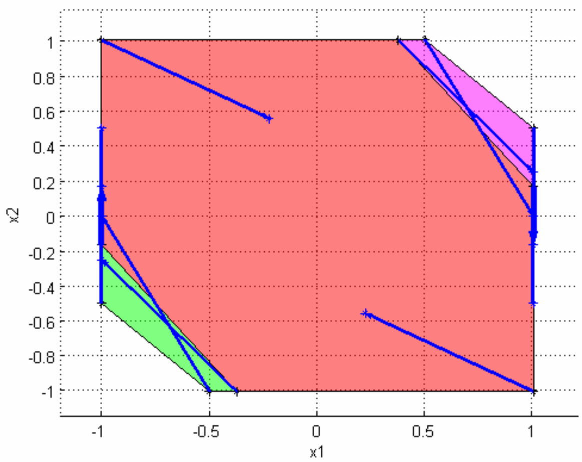



**Figure 5.16 : Vérification des conditions nécessaires et suffisantes de faisabilité, exemple 5.34 Cas II**

### 5.2.3 Solution générale pour la synthèse d'une loi MPC faisable

Les définitions du paragraphe précédent ont montré que garantir la faisabilité dans le cas III implique de construire un ensemble invariant par rapport à la loi MPC qui peut être formulé explicitement. Si les deux ensembles $X_f$ et $X_0$ sont faciles à représenter, déterminer un ensemble invariant est en général un problème complexe. Une idée des difficultés rencontrées pendant la construction itérative d'un tel ensemble est donnée par le résultat élémentaire suivant.

**Proposition 5.35 :** Soit le système :

$$x_{k+1} = Ax_k + Bu_k, k \geq 0, x_0 = x^0,$$
$$Cx_k + Du_k \leq \gamma$$

$X_0$ l'ensemble convexe des conditions initiales $x^0 \in X_0$ pour une loi de commande linéaire par morceaux :

$$u_k^{MPC}(x_k) = \begin{cases} \Psi_1 x_k + \psi_1 & pour \quad x_k \in D_1 \\ \vdots \\ \Psi_n x_k + \psi_n & pour \quad x_k \in D_n \end{cases} \tag{5.64}$$

et $X_f$ l'ensemble convexe qui satisfait $X_0 \subset X_f = \bigcup_{i=1}^{n} D_i$. L'ensemble :

$$X_1 = \{x | \exists x_0 \in X_0 \ t.q. \ x = Ax_0 + Bu_k^{MPC}(x_k)\}$$

n'est pas un ensemble convexe dans le cas général ■

Ce résultat de base montre qu'en essayant de développer la suite récursive des ensembles atteignables à partir de l'ensemble initial $X_0$, on ne peut que les décrire par une union d'ensembles convexes. Une autre possibilité est d'appliquer récursivement une procédure constructive similaire à celle du théorème (5.32) mais encore une fois des problèmes informatiques apparaissent à cause de l'explosion combinatoire du nombre de points extrémaux.

Un autre impedimentum à la construction itérative de l'ensemble invariant par rapport à une loi de commande linéaire par morceaux est que $X_1 \cap X_0 \neq 0$. Dès lors que le domaine d'intérêt pour l'infaisabilité est constitué exactement par la différence de ces deux ensembles $X_1 - X_0$, et que la différence n'est pas une opération fermée sur le corps des ensembles convexes, voici donc un nouvel argument en défaveur d'une démarche itérative de construction de l'espace invariant.

La complexité d'une opération finie d'analyse d'une loi de commande MPC donnée, s'inscrivant dans la classe III du point de vue faisabilité, indique qu'utiliser des arguments basés exclusivement sur l'analyse des ensembles et l'évolution avec la solution optimale d'un problème mpQP est inadapté et que l'infaisabilité mérite d'être traitée en revenant aux principes gouvernant les lois basées sur le principe de l'horizon fuyant. Il faut donc traiter l'aspect de synthèse d'une loi prédictive à partir de la loi optimale à horizon infini. La suite des développements adopte cette démarche pour fournir deux algorithmes concrets de synthèse de lois faisables. Des hypothèses seront volontairement ajoutées afin de pouvoir déduire par la suite des propriétés de stabilité, suffisantes seulement pour la faisabilité.

**Première stratégie de conception basée sur l'horizon de prédiction**

Pour un système :

$$x_{k+1} = Ax_k + Bu_k, k \geq 0, x_0 = x^0,$$
$$Cx_k + Du_k \leq \gamma$$
$$x^0 \in X_0$$

une loi MPC faisable peut être obtenue en appliquant l'algorithme suivant.

**Algorithme 5.36 :**

1) Calculer $IC_{\max}$ (cf. Annexe 3, notation A 3.11)

2) Vérifier si $X_0 \subset IC_{\max}$[21]. Si cette inclusion n'est pas vérifiée, l'algorithme s'arrête car même si la faisabilité peut être garantie, la stabilité asymptotique ne peut pas être assurée.

[21] Cette relation d'inclusion assure qu'il existe une séquence de commande qui amène l'état vers l'origine asymptotiquement.

3) Choisir $Q$ et $R$ telles que les pondérations désirées de l'effort de commande et minimisation de l'erreur soient bien distribuées. Calculer le gain optimal pour le problème de commande optimale à horizon infini, $K_{LQ}$, et la solution d'équation de Riccati associée, $P$.

4) Calculer $O_\infty$ (cf. Annexe 3, notation A3.26) pour le système initial bouclé par $u_k = K_{LQ} x_k$

5) En utilisant $O_\infty$ (espace maximal admissible), trouver $\widetilde{N} < \infty$ tel que $(\mathbf{P^N}) \equiv (\mathbf{P})$ (problème de commande à horizon fini et infini respectivement, voir Annexe 3) pour tout $X_0$.

6) Déterminer la formulation explicite de la loi MPC avec $N = \widetilde{N}$ :

$$\begin{aligned}
&u^{MPC}(x) = k_u^*(x,1) \\
&k_u^*(x_0) = \arg\min \sum_{i=0}^{N-1} \left\{ x_i^T Q x_i + u_i^T R u_i \right\} + x_N^T P x_N \\
&x_{k+1} = Ax_k + Bu_k, k \geq 0, \\
&Cx_k + Du_k \leq \gamma
\end{aligned} \tag{5.65}$$

**Proposition 5.37 :** La loi obtenue à l'étape 6) de l'algorithme précédent est garantie faisable.

**Preuve :** Il faut trouver $IF$ positif invariant tel que $X_f \supset IF \supset X_0$. En choisissant $IF = X_f$, le résultat est démontré.

**Théorème 5.38 :** Toute loi MPC obtenue après une formulation équivalente à (5.65) avec $N > \widetilde{N}$ est faisable.

**Preuve :** Pour tout $N > \widetilde{N}$ le problème d'optimisation dans (5.65) est équivalent au problème à horizon infini, les lois MPC sont donc identiques.

**Deuxième stratégie de conception basée sur l'ensemble final invariant**

Pour la procédure décrite précédemment, l'existence d'une contrainte à la fin de l'intervalle de prédiction $x_N \in O_\infty$ est masquée par le choix de l'horizon de prédiction important qui fait qu'une telle formulation est inutile (on se positionne alors dans $O_\infty$). En revanche, ce type de contrainte peut être à la base d'une autre construction de loi MPC avec garantie de faisabilité.

**Algorithme 5.39 :**

1) – 4) similaires à celles de l'algorithme 5.36.

5) Choisir un *N* et déterminer la formulation explicite de la loi MPC :

$$\begin{aligned}
&u^{MPC}(x) = k_u^*(x,1) \\
&k_u^*(x_0) = \arg\min \sum_{i=0}^{N-1} \left\{ x_i^T Q x_i + u_i^T R u_i \right\} + x_N^T P x_N \\
&x_{k+1} = Ax_k + Bu_k, k \geq 0, \\
&Cx_k + Du_k \leq \gamma \\
&x_N \in O_\infty
\end{aligned} \tag{5.66}$$

6) Calculer $X_f$ pour la loi résultante.

7) Si $X_f \supset X_0$ arrêter l'algorithme, sinon augmenter *N* et revenir à 5).

L'algorithme proposé s'arrête après un nombre fini d'itérations car pour $N = \widetilde{N}$, comme dans l'algorithme 5.39, l'inclusion testée à l'étape 7) est vérifiée. Il est intéressant d'observer que $N = \widetilde{N}$ reste une limite maximale car le *N* trouvé peut être diminué de façon sensible avec l'existence de la contrainte terminale. En effet, celle-ci change la philosophie de la commande, en s'approchant d'un régulateur en *mode dual* (terminologie liée plutôt au cas non linéaire - [MM93]), un premier mode qui force le système vers l'ensemble terminal et un deuxième équivalent à la loi à horizon infini. Une proposition similaire à la Proposition 5.37 peut enfin être énoncé pour l'algorithme 5.39 car $X_f$ constitue aussi un ensemble invariant.

### 5.2.4 Suivi de trajectoire. Faisabilité du problème de suivi. Limites des trajectoires faisables

Le problème de faisabilité de la commande prédictive a été analysé jusqu'à présent en considérant le retour de l'état vers l'origine (contexte de régulation, sauf le cas GPC défini directement avec un formalisme de suivi de trajectoire). En fait le problème abordé précédemment est un cas particulier d'une classe de problèmes plus larges pour laquelle le système doit suivre une trajectoire donnée. On peut mettre en évidence, selon la nature des consignes à suivre, deux classes spécifiques de problématique :

a) *Asservissement*

Il s'agit de suivre une fonction polynomiale de degré donné. Soit le système défini par :

$$\begin{aligned} & x_{k+1} = Ax_k + Bu_k, k \geq 0, x_0 = x^0 \in X_0, \\ & Cx_k + Du_k \leq \gamma \\ & y_k = Hx_k \end{aligned} \tag{5.67}$$

Pour l'asservissement, le signal de référence $r$ est généré en fait par un système linéaire :

$$\begin{aligned} & z_{k+1} = A_Z z_k; \\ & r_k = H_Z z_k; \\ & z_0 \in Z_0 \end{aligned} \tag{5.68}$$

soit en compactant les deux dynamiques le système étendu suivant :

$$\begin{aligned} & \widetilde{x}_{k+1} = \begin{bmatrix} z_{k+1} \\ x_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_M & 0 \\ 0 & A \end{bmatrix}}_{\widetilde{A}} \widetilde{x}_k + \underbrace{\begin{bmatrix} 0 \\ B \end{bmatrix}}_{\widetilde{B}} u_k, \\ & \underbrace{[0 \;\; C]}_{\widetilde{C}} \widetilde{x}_k + \widetilde{D}u_k \leq \widetilde{\gamma}; \widetilde{D} = D; \widetilde{\gamma} = \gamma \end{aligned} \tag{5.69}$$

et : $$\widetilde{x}_0 \in \widetilde{X}_0 = \left\{ \begin{bmatrix} z_0 \\ x0 \end{bmatrix} \middle| z_0 \in Z_0; x_0 \in X_0 \right\}$$

Pour ce modèle, en supposant que le vecteur $\widetilde{x}_{k+1} = [z_{k+1} \;\; x_{k+1}]^T$ est disponible (sinon un observateur doit être construit), on peut décrire le problème d'asservissement en commande prédictive par :

$$\begin{aligned} & u^{MPC}(\widetilde{x}_k) = k_u^*(\widetilde{x}_k, 1) \\ & k_u^*(x_0) = \arg\min_{k_u} \sum_{i=0}^{N-1} \left\{ \widetilde{x}_i^T \widetilde{Q} \widetilde{x}_i + u_i^T R u_i \right\} + \widetilde{x}_N^T P \widetilde{x}_N \\ & \widetilde{x}_{k+1} = \widetilde{A}\widetilde{x}_k + \widetilde{B}u_k, k \geq 0, \\ & \widetilde{C}\widetilde{x}_k + \widetilde{D}u_k \leq \widetilde{\gamma} \\ & \widetilde{x}_N \in \widetilde{X}_N \end{aligned} \tag{5.70}$$

avec $\widetilde{Q} = [H_Z \quad -H] Q \begin{bmatrix} H_Z \\ -H \end{bmatrix}$. $Q$ et $R$ sont les pondérations sur l'erreur et la commande dans le critère de coût.

*b) Suivi d'un modèle de référence*

Le système doit suivre la dynamique d'un autre système considéré comme modèle de référence, lui-même excité par un signal polynomial (c'est le cas de certaines applications sensibles à la continuité du signal à suivre, par exemple en aéronautique)

Pour le même système (5.67), suivre un modèle de référence peut se traduire par :

$$\begin{aligned} &\zeta_{k+1} = A_M \zeta_k + B_M \tilde{y}_k ; \\ &r_k = H_M \zeta_k ; \\ &\zeta_0 \in Z_0 \end{aligned} \tag{5.71}$$

et le signal de commande est donné par :

$$\begin{aligned} &\xi_{k+1} = A_R \xi_k ; \\ &\tilde{y}_k = H_R \xi_k ; \\ &\xi_0 \in \Xi_0 \end{aligned} \tag{5.72}$$

Après quelques substitutions, on peut montrer que (5.71)-(5.72) sont équivalentes à :

$$\begin{aligned} &z_{k+1} = A_Z z_k ; \\ &r_k = H_Z z_k ; \\ &z_0 \in Z_0 \end{aligned} \tag{5.73}$$

avec : $z_k = \begin{bmatrix} \zeta \\ \xi \end{bmatrix}; A_Z = \begin{bmatrix} A_M & B_M H_R \\ 0 & A_R \end{bmatrix}; H_Z = \begin{bmatrix} H_M & 0 \end{bmatrix}$ et $Z_0 = \left\{ \begin{bmatrix} \zeta_0 \\ \xi_0 \end{bmatrix} \middle| \zeta_0 \in \mathrm{Z}_0 ; \xi_0 \in \Xi_0 \right\}$

et l'on se ramène alors au cas de l'asservissement (5.69-5.70).

*c) Suivi*

Ce cas considère le suivi d'une fonction temporelle ayant une allure donnée (ce cas inclut par exemple les trajectoires connues uniquement sur un certain horizon fini sans information particulière sur son évolution asymptotique). Pour le même système (5.67), le suivi de trajectoire revient à amener sa sortie $y_k$ vers la sortie du système :

$$\begin{aligned} &z_{k+1} = A_Z z_k + B_Z w_k \\ &r_k = H_Z z_k ; \\ &z_0 \in Z_0 \end{aligned} \tag{5.74}$$

où $w_k$ est connu sur un horizon $N_w$ tel que l'on dispose à l'instant $k$ de toutes les valeurs $w_{k+j|k}$, $j = 0, \ldots, N_w - 1$. La faisabilité d'une loi de commande prédictive dans ce cas doit être analysée plus spécifiquement car elle diffère légèrement du cas classique de régulation vers l'origine.

Parmi les trois problèmes identifiés ci-dessus, définissons ceux pouvant faire appel aux outils déjà décrits :

- Pour les problèmes d'asservissement et de suivi de modèle de référence, l'application d'une loi prédictive semble relativement identique au cas classique. La situation peut en fait être différente car, dans la formulation (5.69-5.70), la paire $(\tilde{A}, \tilde{B})$ n'est pas contrôlable. Il faut donc vérifier si la stabilisabilité est assurée [22]. Si tel est le cas, il est alors possible de procéder avec les méthodes classiques, en élaborant la solution de l'équation de Riccati et en construisant l'espace invariant pour le système (5.69) et effectuant la synthèse d'un régulateur MPC. La solution si elle existe assure la faisabilité par rapport à un ensemble faisable $\tilde{X}_f$.

  Un erreur fréquente consiste à synthétise la loi MPC sans tenir compte du domaine de la famille de la consigne à suivre, représenté par l'espace de conditions initiales $z_0 \in Z_0$. Si l'on vérifie la faisabilité uniquement par rapport aux conditions initiales de la partie à commander $x_0 \in X_0$, on commet une erreur entraînant des conséquences néfastes car il existe toujours un point initial pour le modèle de référence qui ne peut pas être suivi. En revanche, si lors de la synthèse on assure que la loi MPC est faisable par rapport à $\tilde{x}_0 \in \tilde{X}_f \supset \tilde{X}_0$, le problème dans le cas nominal ne pose pas de difficultés.

[22] Des problèmes numériques apparaissent souvent pour des consignes persistantes ayant comme modèle un système avec des racines sur le cercle unité, pour lesquels les solveurs d'équations de Riccati éprouvent quelques difficultés.

**Exemple 5.40 :** Considérons le cas du double intégrateur :

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u_k, u_k \in U = \{u \in \Re | -1 \leq u \leq 1\}, x_k \in \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| -10 \leq x_1 \leq 10; -10 \leq x_2 \leq 10 \right\}$$

$$z_{k+1} = 0.9 z_k ; z_0 \in Z_0 = \{z \in \Re | -10 \leq z \leq 10\}, x_0 \in X_0 = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| -1 \leq x_1 \leq 1; -0,8 \leq x_2 \leq 0,8 \right\}$$

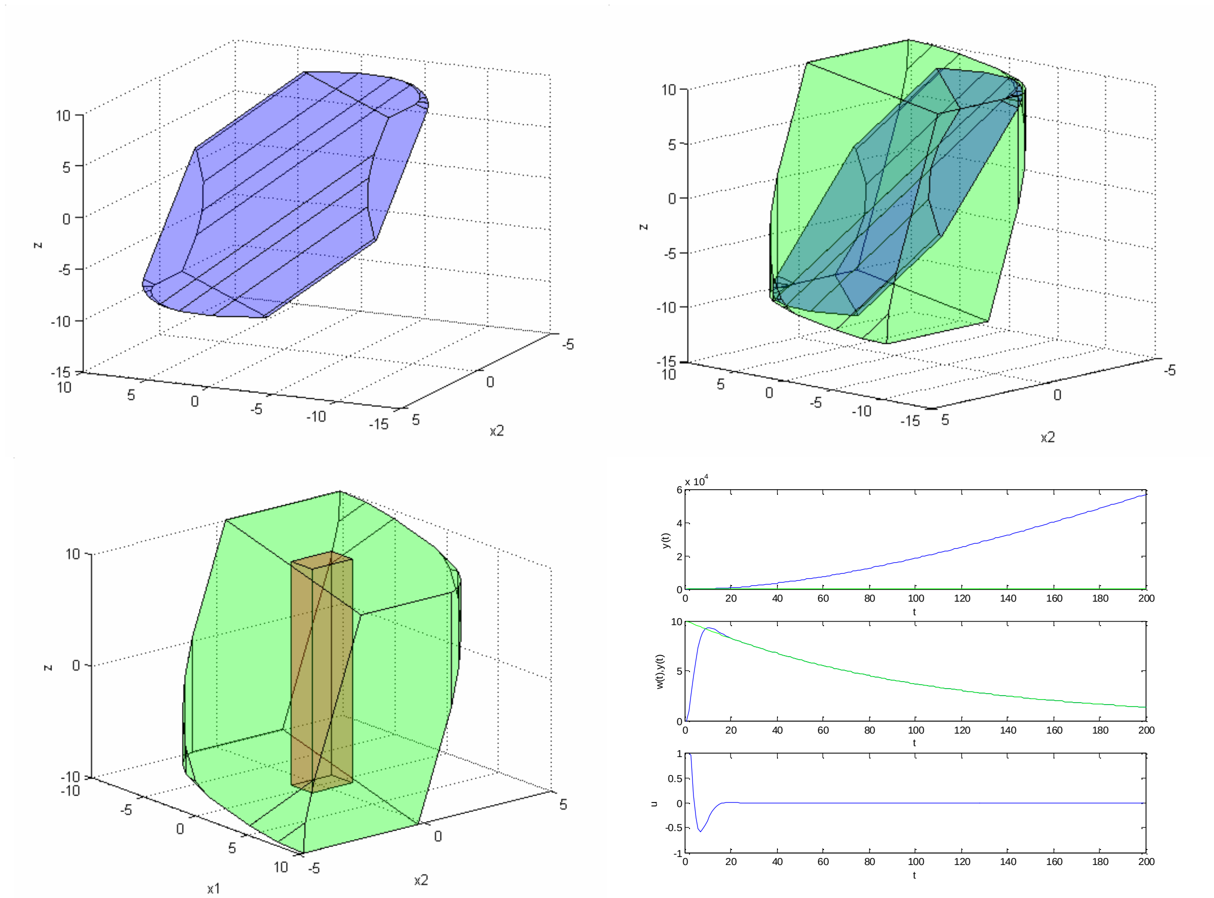



**Figure 5.17 : a) Ensemble maximal invariant $O_\infty$**
**b) Ensemble faisable $\widetilde{X}_f$ pour la loi MPC avec $N = 4$ et ensemble terminal $O_\infty$**
**c) Inclusion $\widetilde{X}_f \supset \widetilde{X}_0$**
**d) Simulation du système sans bouclage, avec bouclage et signal de commande**

La loi sans contrainte avec $N = 4$ est donnée par : $u_k = [0.23335 \;\; 0.67756] x_k + 0.22664 z_k$ et l'on peut alors construire l'espace maximal invariant (Figure 5.17a) constituant l'ensemble terminal pour la loi MPC. L'espace faisable de cette loi est reporté Figure 5.17b. Pour un horizon $N = 4$, le système asservi par la loi mentionnée sera faisable pour les ensembles des condition initiales proposés $z_0 \in Z_0$, $x_0 \in X_0$ (Figure 5.17c). La simulation démontre les qualités de suivi du système bouclé, en comparaison avec le système initial instable (Figure 5.17d).

Comme mentionné auparavant, la situation d'infaisabilité est liée à la non adaptation de la loi MPC aux conditions initiales. Si l'ensemble des conditions initiales pour le modèle de référence $z_0 \in Z_0$ n'est pas inclus dans l'ensemble étendu pour lequel la loi MPC est bien définie, on risque de se retrouver dans une situation d'infaisabilité. Ainsi, pour le modèle précédent, la loi MPC avec l'horizon $N = 4$ sera faisable pour $x_0 \in X_0 = \left\{ [x_1 \quad x_2]^T \middle| -4 \leq x_1 \leq 4; -1 \leq x_2 \leq 1 \right\}$ et $z_0 \in Z_0 = \{z \in \Re | -5 \leq z \leq 5\}$, voir Figure 5.18a, mais devient infaisable pour $z_0 \in Z_0 = \{z \in \Re | -10 \leq z \leq 10\}$ Figure 5.18b. En augmentant l'horizon de prédiction de la loi MPC jusqu'à $N = 5$, l'espace invariant augmente et la faisabilité est assurée car $\widetilde{X}_f \supset \widetilde{X}_0$, Figure 5.18c.

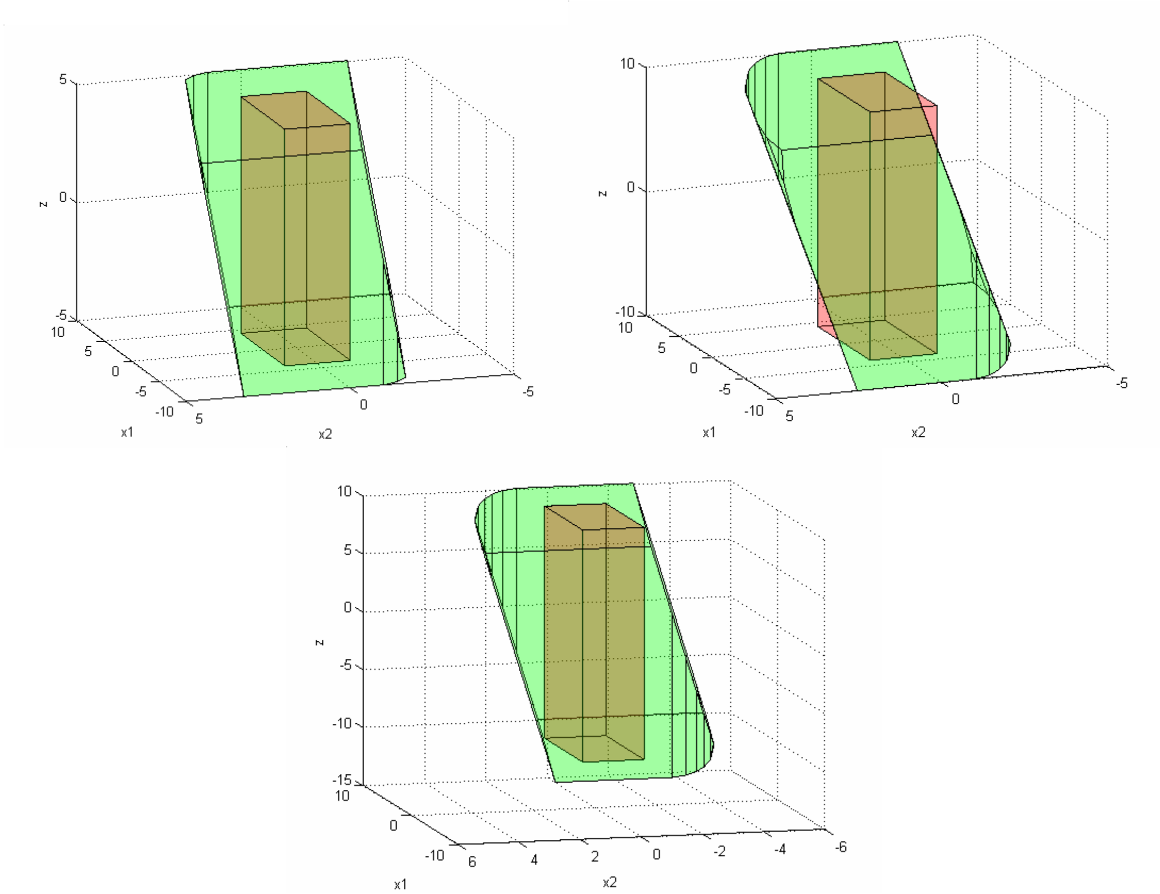



**Figure 5.18 : a) Ensemble faisable $\widetilde{X}_f$ et ensemble des conditions initiales $\widetilde{X}_0$, $N = 4$**
**b) $\widetilde{X}_f$ et $\widetilde{X}_0$ avec $z_0 \in \widetilde{Z}_0 = \{z \in \Re \mid -10 \le u \le 10\}$, pour $N = 4$**
**c) Inclusion $\widetilde{X}_f \supset \widetilde{X}_0$ pour $N = 5$**

- Le cas de suivi de trajectoire est beaucoup plus délicat et dans le cas où aucune contrainte n'est explicitée sur le signal $r_k$, il est généralement facile de montrer qu'il n'existe pas de loi de commande garantissant la faisabilité indépendamment de $r_k$. Il est malgré tout possible de décrire les valeurs extrêmes de la consigne qui peuvent être traitées. L'élément essentiel dans ce cas est la loi de commande MPC synthétisée.

  Une première difficulté survenant pendant cette opération consiste à trouver un ensemble terminal invariant. Si l'on compare au cas où le modèle à suivre est autonome, le domaine invariant terminal est ici défini par rapport à une loi constante qui doit prendre en considération le fait que la sortie du modèle de référence peut avoir une valeur stationnaire non nulle (influencée par le signal exogène $w_k$). En conséquence, la solution de l'équation de Riccati construite lors de la version à modèle de référence autonome n'est malheureusement pas le meilleur choix car elle induit une erreur stationnaire non nulle. Un choix classique dans l'esprit de la régulation vers des consignes non nulle [KS72] consiste à modifier le gain LQR du système à commander par un gain terminal qui prend en compte une valeur stationnaire non nulle de sorte que la loi terminale ait la forme :

$$u_{N+k} = K_1 = K_{LQR}\widetilde{x}_{N+k} + ([H \quad 0] * (I - \widetilde{A} + \widetilde{B}K_{LQR})^{-1}\widetilde{B})^{-1} * H_z z_{N+k} \tag{5.75}$$

  avec l'hypothèse que les inversions impliquées soient bien définies. Les structures du système étendu et de la formulation MPC peuvent bien sûr être modifiées pour tenir compte de la valeur du signal exogène comme paramètre du problème d'optimisation multiparamétrique :

$$\left.\begin{array}{ll} x_{k+1} = Ax_k + Bu_k, k \ge 0, x_0 = x^0 \in X_0, & z_{k+1} = A_Z z_k + B_z w_k; \\ Cx_k + Du_k \le \gamma & \text{et } r_k = H_Z z_k; \\ y_k = Hx_k & z_0 \in Z_0 \end{array}\right\} \Leftrightarrow \left\{\begin{array}{l} \widetilde{z}_{k+1} = \widetilde{A}_Z \widetilde{z}_k + \widetilde{B}_z \widetilde{w}_k; \\ r_k = \widetilde{H}_Z \widetilde{z}_k; \\ \widetilde{z}_0 \in \widetilde{Z}_0 \end{array}\right.$$

avec :

$$\widetilde{z}_k = \begin{bmatrix} z_k \\ w_k \\ w_{k+1} \\ \vdots \\ w_{k+N_w-1} \end{bmatrix}; \widetilde{w}_k = w_{k+N_w}; \widetilde{A}_Z = \left[\begin{array}{c|cccc} A_M & B_M & 0 & \cdots & 0 \\ \hline & 0 & 1 & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 0 \\ & \vdots & \vdots & \ddots & 1 \\ & 0 & 0 & \cdots & 0 \end{array}\right]; \widetilde{B}_Z = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}; \widetilde{C}_Z = [C_M \quad 0 \quad \cdots \quad 0];$$

$$u^{MPC}(\widetilde{x}_k) = k_u^*(\widetilde{x}_k, 1) = k_u^*(x_k, \widetilde{z}_k, 1) = k_u^*(x_k, z_k, w_k, \cdots, w_{k+N_w-1}, 1)$$

$$k_u^*(x_0) = \arg\min_{k_u} \sum_{i=0}^{N-1} \left\{ \widetilde{x}_i^T \widetilde{Q} \widetilde{x}_i + u_i^T R u_i \right\} + \widetilde{x}_N^T P \widetilde{x}_N$$

$$\widetilde{x}_{k+1} = \widetilde{A}\widetilde{x}_k + \widetilde{B}u_k, k \geq 0,$$

$$\widetilde{C}\widetilde{x}_k + \widetilde{D}u_k \leq \widetilde{\gamma}$$

$$\widetilde{x}_N \in \widetilde{X}_N$$

Les performances de la commande prédictive, liées à la capacité de prédiction, seront cependant considérablement détériorées si l'horizon de prédiction est plus grand que l'horizon sur lequel on dispose d'information sur le signal exogène.

**Exemple 5.41 :** Pour l'exemple précédent, les figures suivantes (trajectoire et commande) proposent de visualiser la trajectoire obtenue avec la loi MPC pour plusieurs situations illustrant les points précédents.

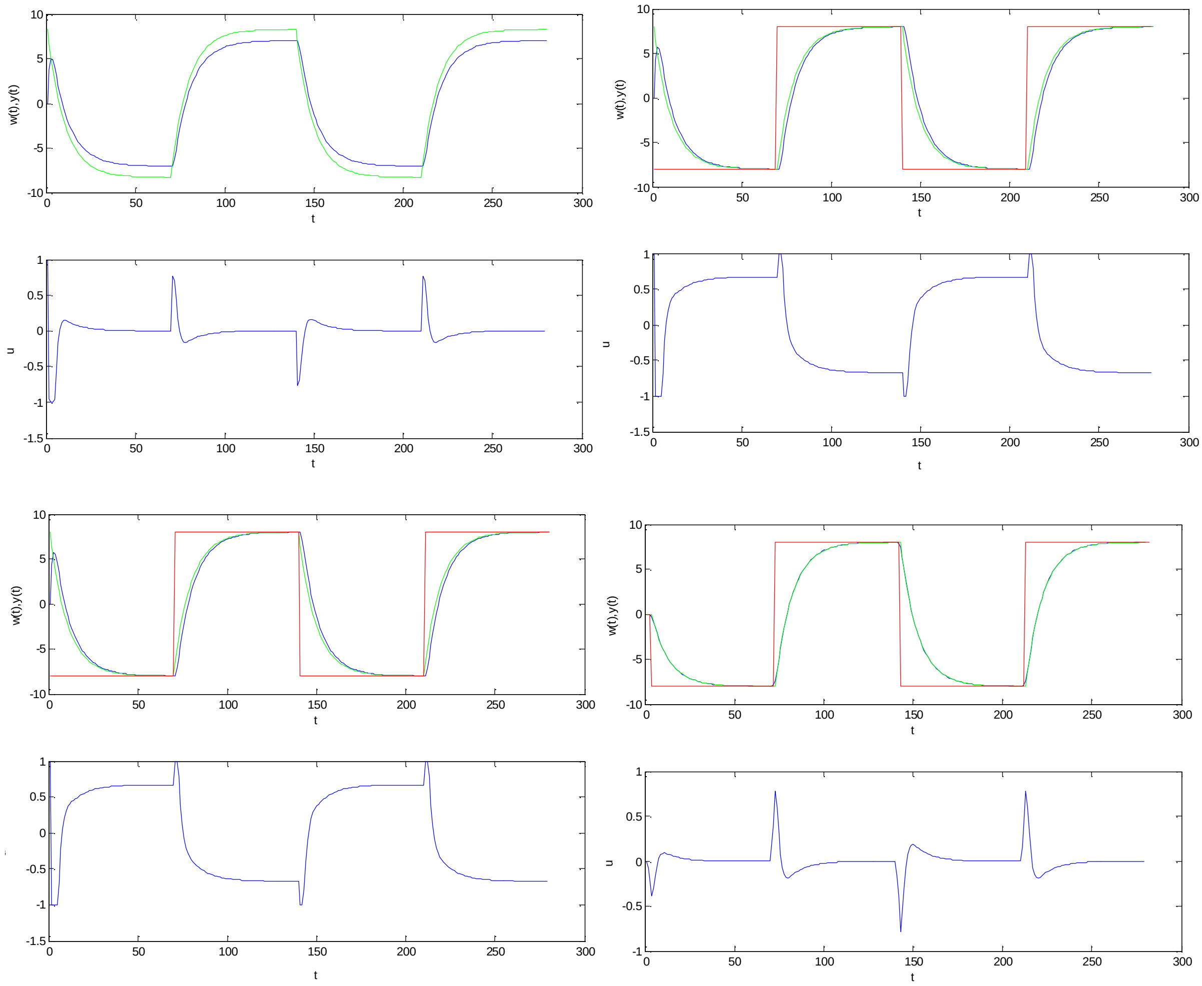



**Figure 5.19 : a) Simulation de la loi MPC avec $N = 4$, pour le système autonome et gain LQR b) MPC ($N = 4$) et gain modifié c) MPC ($N = 4, N_w = 1$) d) MPC ($N = N_w = 4$)**

Une loi synthétisée en se basant sur le gain LQR montre qu'elle ne permet pas le suivi d'un signal exogène qui n'aurait pas l'origine comme point stationnaire (Figure 5.19a). On peut alors modifier le gain terminal comme dans (5.75) mais sans prendre en compte l'existence du signal exogène, la loi prédictive est loin d'avoir de bonnes performances (Figure 5.19b). Si l'on dispose d'informations sur le signal exogène (hypothèse a priori réaliste en commande prédictive) mais pour un horizon $N_w < N$, les performances sont améliorées, des erreurs conséquentes subsistent car les prédictions pour $N_w < t \leq N$ sont erronées (Figure 5.19c). Les meilleurs résultats sont obtenus pour $N_w = N$, Figure 5.19d.

Terminons cette analyse sur la synthèse de la loi MPC en suivi de trajectoire en supposant pour la suite qu'une telle loi existe et que l'on dispose de l'ensemble des conditions initiales faisables :

$$\widetilde{X}_f = \left\{\widetilde{x} \middle| \exists u^{MPC}(\widetilde{x}|)\right\}$$

qui, étant un ensemble convexe polyédral, est décrit par un jeu d'inégalités :

$$\widetilde{X}_f = \left\{\widetilde{x} \middle| \Lambda\widetilde{x} \leq \lambda\right\} = \left\{[z \quad x]^T \middle| \Lambda_z z + \Lambda_x x \leq \lambda\right\}$$

Le signal exogène $w_k$ étant donné, on peut décrire le phénomène d'infaisabilité pour un certain $z_{k+1} = A_z z_k + B_z \widetilde{w}_k$ comme étant défini par la relation $x_{k+1} \notin \left\{x \middle| \Lambda_x x \leq \lambda - \Lambda_z z_{k+1}\right\}$, mettant ainsi l'accent sur l'ensemble faisable du système commandé.

La loi MPC est généralement synthétisée de sorte qu'en l'absence de signal exogène $w_{k+N_w} = 0$, la faisabilité soit préservée (condition naturelle pour garantir les qualités de régulation de la boucle) et donc que l'on assure que $x_{k+1} \in \left\{x \middle| \Lambda_x x \leq \lambda - \Lambda_z \widetilde{z}_{k+1}\right\}$, où $\widetilde{z}_{k+1} = \widetilde{A}_z \widetilde{z}_k$. On peut distinguer au niveau du suivi de trajectoire trois possibilités vis-à-vis de la faisabilité :

1) Si $B_z w_{k+N_w} = 0$ la loi MPC est faisable (car $\widetilde{z}_{k+1} = z_{k+1}$),
2) $B_z w_{k+N_w} \neq 0$ et $D = \left\{x \middle| \Lambda_x x \leq \lambda - \Lambda_z \widetilde{z}_{k+1}\right\} \cap \left\{x \middle| \Lambda_x x \leq \lambda - \Lambda_z \widetilde{z}_{k+1} - \Lambda_z B_z \widetilde{w}_k\right\} \neq 0$ (La Figure 5.20b représente l'ensemble $\widetilde{X}_f$ et la Figure 5.20a illustre un cas d'intersection non vide) :
   - si $x_{k+1} \notin D$, on se retrouve en situation d'infaisabilité (due à la inadéquation entre l'état du système et la valeur du signal exogène),
   - si $x_{k+1} \in D$, la loi est faisable.
3) $B_z \widetilde{w}_k \neq 0$ et $D = \left\{x \middle| \Lambda_x x \leq \lambda - \Lambda_z \widetilde{z}_{k+1}\right\} \cap \left\{x \middle| \Lambda_x x \leq \lambda - \Lambda_z \widetilde{z}_{k+1} - \Lambda_z B_z \widetilde{w}_k\right\} = 0$ (la Figure 5.20c illustre cette situation), il y a alors infaisabilité car le signal exogène induit un saut de l'état du modèle à suivre au delà des limitations du système à commander.

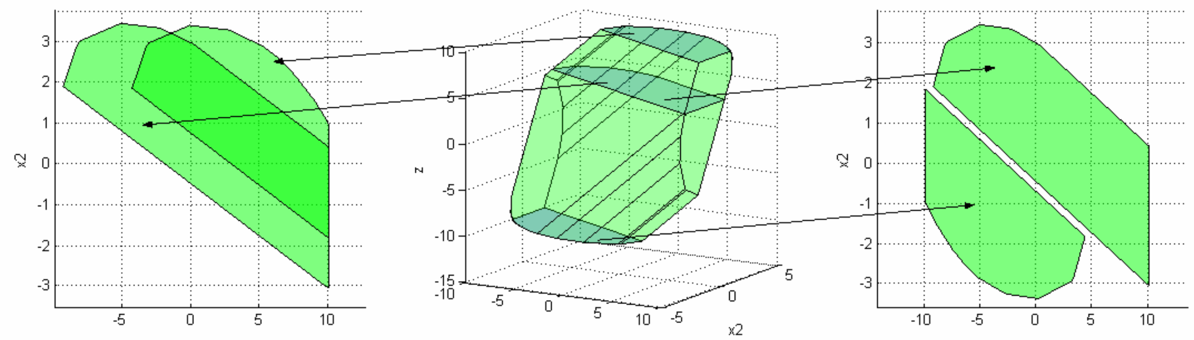



**Figure 5.20 : a) Ensembles faisables pour le système à commander (Exemple 5.40, $\widetilde{z}_{k+1} = 5, z_{k+1} = 10$) b) $\widetilde{X}_f$ (Exemple 5.40) c) Ensembles faisables (Exemple 5.40, $\widetilde{z}_{k+1} = 5, z_{k+1} = -10$)**

Il est aussi possible d'analyser le problème d'infaisabilité en examinant la faisabilité du signal exogène et donc en considérant l'ensemble $\widetilde{X}_f$ comme un polyèdre paramétré dont le vecteur de paramètres coïncide avec le vecteur d'état du système à commander et pour lequel l'infaisabilité se traduit par $z_{k+1} \notin \left\{z \middle| \Lambda_z z \leq \lambda - \Lambda_x x_{k+1}\right\}$. Pour les systèmes monovariables, les valeurs limites de consigne à suivre ou de signal exogène peuvent donner une image de la liberté dont dispose le régulateur (Tableau 5.5).

**Conditions de faisabilité**

$$r_k^{\min}(x_k) \le r_k = H_z z_k \le r_k^{\max}(x_k) \qquad\qquad w_k^{\min}(x_k, z_k) \le w_k \le w_k^{\max}(x_k, z_k)$$

$$\begin{array}{ll} r_k^{\min}(x_k) = \min\limits_{z} H_Z z & w_k^{\min} = \min\limits_{w} w \\ \text{sous les contraintes } \Lambda_z z \le \lambda - \Lambda_x x_k & \text{sous les contraintes } \Lambda_z B_z w \le \lambda - \Lambda_x x_{k+1} - \Lambda_z A_z z_k \\ r_k^{\max}(x_k) = \max\limits_{z} H_Z z & w_k^{\max} = \max\limits_{w} w \\ \text{sous les contraintes } \Lambda_z z \le \lambda - \Lambda_x x_k & \text{sous les contraintes } \Lambda_z B_z w \le \lambda - \Lambda_x x_{k+1} - \Lambda_z A_z z_k \end{array} \tag{5.76}$$

**Tableau 5.5 : Valeurs limites de consigne/signal exogène à suivre**

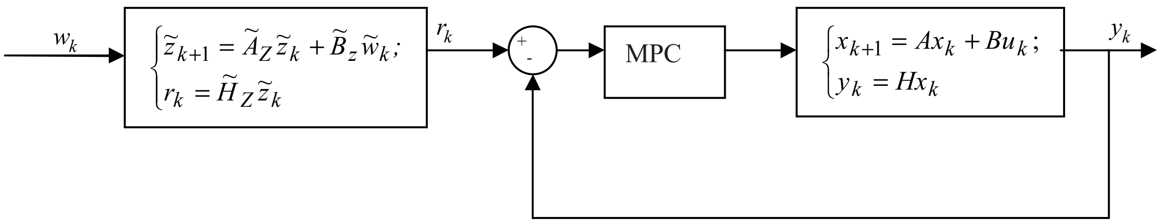
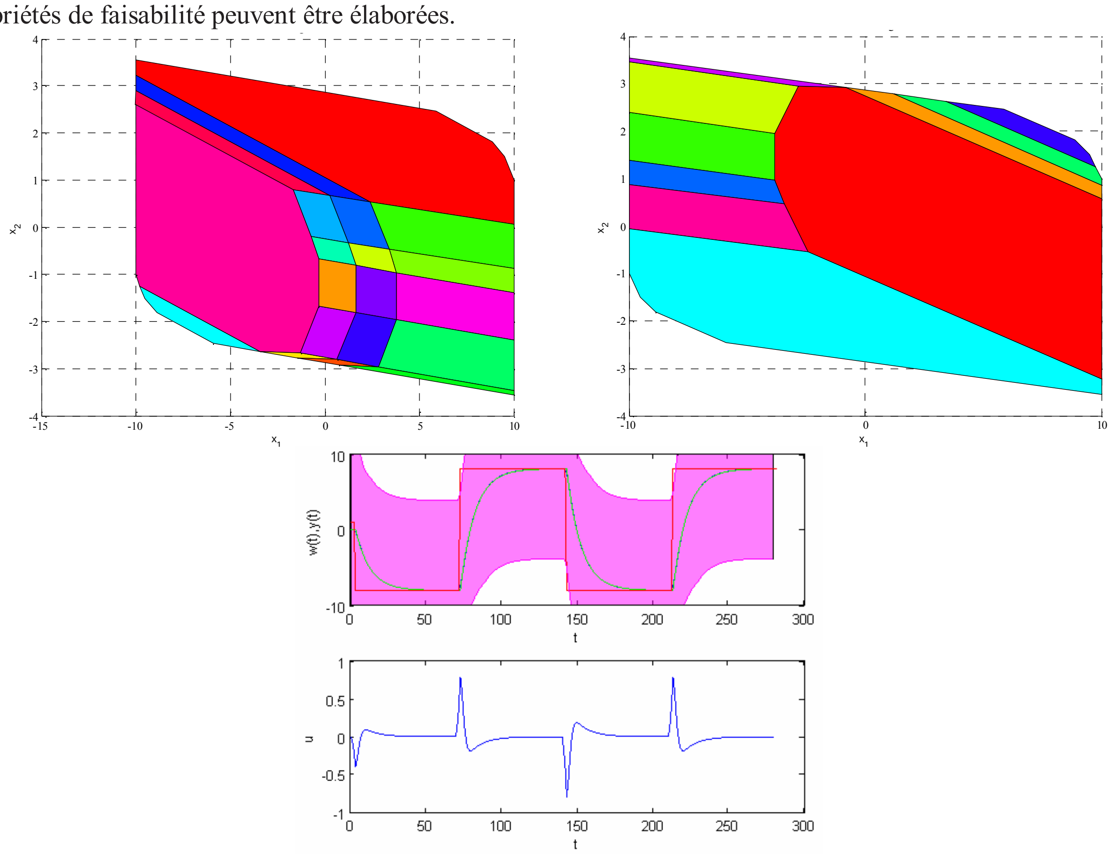



**Figure 5.21 : Schéma de commande prédictive en suivi de trajectoire**

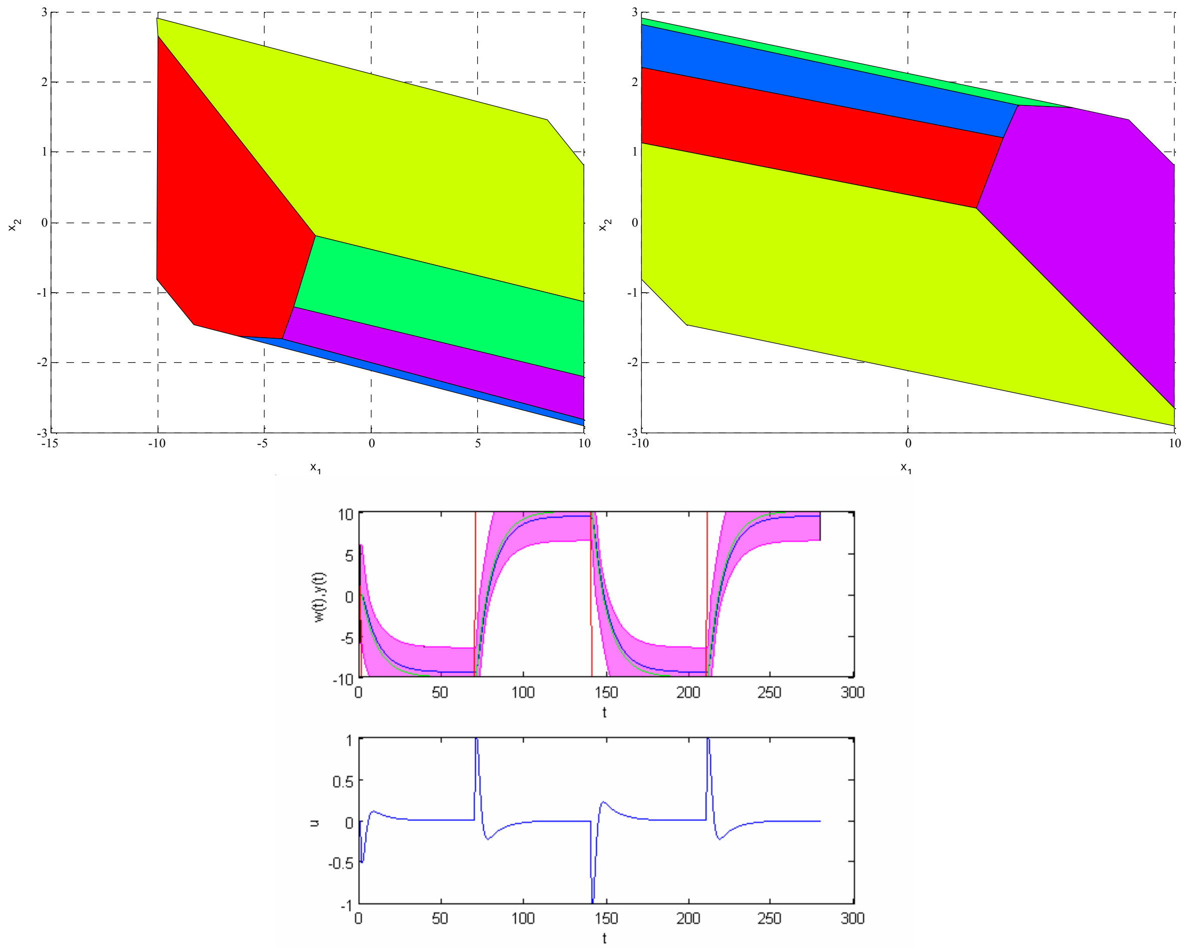
$r_k^{\min}, r_k^{\max}, w_k^{\min}, w_k^{\max}$ sont donc les solutions de problèmes d'optimisation linéaires multiparamétriques. Une procédure pouvant être généralement employée pour la synthèse MPC dans le cas de suivi prend en compte à la place du modèle de consigne (5.74) uniquement la partie autonome en ignorant le signal exogène $w_k$.
Ensuite l'évolution d'un point $\begin{bmatrix} z_k^T & x_k^T \end{bmatrix}^T \in \tilde{X}_f$ vers un point $\begin{bmatrix} (A_Z z_k)^T & x_{k+1}^T \end{bmatrix}^T \in \tilde{X}_f$ est assurée.

**Exemple 5.42 :** Pour le système double intégrateur présenté lors des exemples précédents, la description explicite des valeurs limites de la référence et du signal exogène admissibles permettant de ne pas perdre les propriétés de faisabilité peuvent être élaborées.

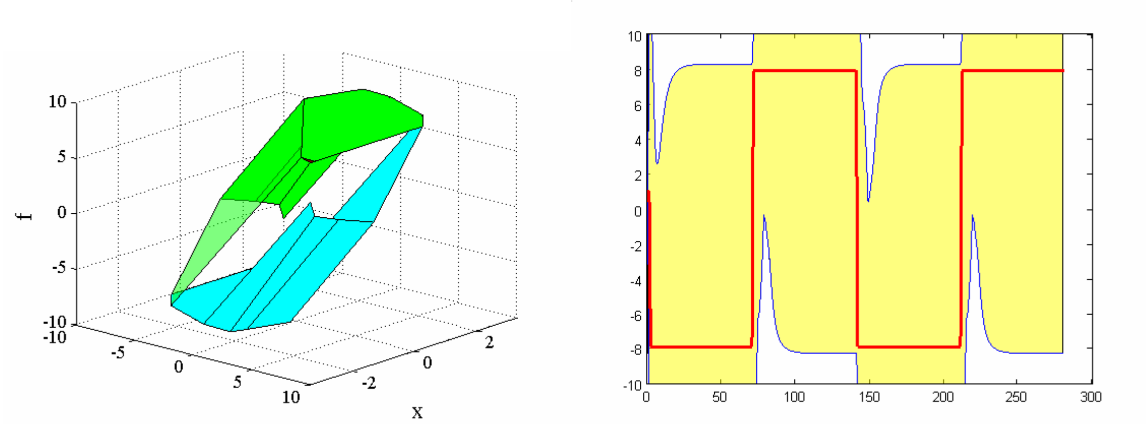



**a) Découpage – formulation explicite de la limite inférieure (MPC $N = 3$) pour $r_k$**
**b) Limite supérieure c) Gabarit faisable**

Ainsi, la Figure 5.22a-b présente les découpages liés aux descriptions explicites du problème d'optimisation paramétrique pour un horizon de prédiction $N = 3$ (suivi d'une référence $r_k$). La Figure 5.22c illustre l'évolution de la zone faisable pour une consigne en créneau. La Figure 5.22d-e-f représente une description similaire pour un horizon de prédiction d'un seul pas. Comme l'ensemble terminal reste le même, le gabarit faisable est réduit induisant un nombre réduit de zones de la description explicite. Finalement, La Figure 5.22g représente le gabarit du signal exogène calculé comme solution d'un problème d'optimisation de type (5.76).




**d) Découpage de l'espace faisable – formulation explicite de la limite inférieure (MPC $N = 1$) pour $r_k$**
**e) Limite supérieure f) Gabarit faisable**




**g) Fonctions explicites (MPC $N = 1$) correspondant aux découpages en d)-e)**
**h) Gabarit faisable pour le signal exogène $w_k$**

**Figure 5.22 : Limitations de la faisabilité**

L'infaisabilité se réduit donc à vérifier que $\begin{bmatrix}(A_Z z_k + B_Z w_k)^T & x_{k+1}^T\end{bmatrix}^H \in \widetilde{X}_f$, interprétable comme un problème de robustesse de la loi MPC initiale. Si l'on se donné la liberté de limiter les valeurs du signal exogène, un mécanisme de prévention de la faisabilité peut être mis en place comme on va le voir au paragraphe suivant.

## *5.3 Récupérer une situation d'infaisabilité*

Le paragraphe précédent a établi qu'un algorithme MPC classique ne peut offrir de garantie de faisabilité pour un signal exogène quelconque $w_k$. Dans ce cas, la solution est soit de revoir le choix des paramètres de réglage (voir paragraphe 5.5), soit d'imaginer un mécanisme de supervision de la trajectoire permettant de forcer la faisabilité d'un système bouclé par le correcteur MPC en proposant à chaque pas d'échantillonnage le suivi de la trajectoire faisable la plus proche, au lieu de la trajectoire optimale infaisable [BM96], [CMP04]. Ce paragraphe analyse ce mécanisme, dans le cas général d'un système pouvant être multivariable.

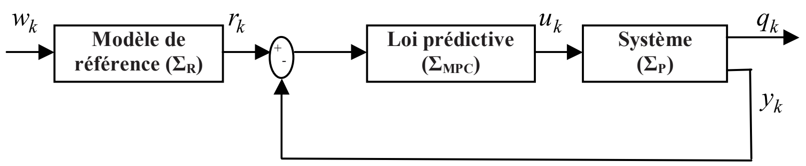
### 5.3.1 Superviseur de trajectoire

Formellement, à chaque pas, la sortie du modèle de référence $r_k$ est remplacée par une fonction de l'état courant et de la référence elle-même :

$$g_k = g(x_k, r_k)$$

telle que $g_k$ constitue la meilleure approximation de $r_k$ au pas *k*, compatible également avec les contraintes de fonctionnement. Deux conditions sont imposées à cette fonction :

1) $g_k \to \hat{r}$ quand $r_k \to r$, avec $\hat{r}$ la meilleure approximation faisable de $r$,
2) si la référence atteint un régime établi et la loi MPC a été synthétisée pour une erreur stationnaire nulle, le superviseur assure un temps d'établissement fini vers la meilleure approximation $\hat{r}$.




**MPC classique**

**Figure 5.24 : Schéma MPC classique pour le suivi de trajectoire mais sans garantie de faisabilité**

Dans le cas SISO, on peut se baser sur la relation $r_k^{\min}(x_k) \le r_k \le r_k^{\max}(x_k)$ et lorsque le signal $r_k$ ne respecte pas ces limitations, on le remplace alors par la limite qu'il dépasse. Dans le cas général MIMO, une indication utile peut être le vecteur faisable le plus proche de la consigne suggéré par rapport à un critère quadratique (on considère le cas sans perturbations) tout en respectant le domaine faisable $X_f$ pour la loi MPC choisie. En supposant que la consigne (le signal exogène) est connue par avance sur un horizon $N_w$, le superviseur sera donc implémenté comme un problème d'optimisation quadratique :

$$\widetilde{r}_k = H_Z \widetilde{z}_k^*$$

$$\left\{\widetilde{z}_k, \widetilde{w}_k, \ldots, \widetilde{w}_{k+N_w}\right\}^*(z_k, x_k, w_k, \ldots, w_{k+N_w}) = \underset{\left\{\widetilde{z}_k, \widetilde{w}_k, \ldots, \widetilde{w}_{k+N_w-1}\right\}}{\arg\min} \sum_{j=0}^{N_w} \begin{bmatrix} (H_Z\widetilde{z}_{k+j} - H_z z_{k+j})^T S_1 (H_Z\widetilde{z}_{k+j} - H_Z z_{k+j}) + \\ + (\widetilde{w}_{k+j} - w_{k+j})^T S_2 (\widetilde{w}_{k+j} - w_{k+j}) \end{bmatrix}$$

$$\text{tel que} \quad \begin{cases} \widetilde{z}_{k+j+1} = A_Z \widetilde{z}_{k+j} + B_Z \widetilde{w}_{k+j} \\ z_{k+j+1} = A_Z z_{k+j} + B_Z w_{k+j} \\ x_{k+j+1} = A x_{k+j} + B u_{k+j}^{MPC}(x_{k+j}, \widetilde{z}_j) \end{cases} ; \quad \begin{bmatrix} \widetilde{z}_{k+j} \\ x_{k+j} \end{bmatrix} \in X_f ; j = 0, .., N_w$$

avec $S_1$, $S_2$ choisies pour définir 'la meilleure approximation'.

Cette formulation qui peut paraître assez compliquée peut se simplifier substantiellement en considérant :

$$A_z \leftarrow \left[\begin{array}{c|cc} A_z & B_Z & 0 \\ \hline 0 & 0_{N_w\times 1} & I \\ & 0_{1\times 1} & 0_{1\times N_w} \end{array}\right]; H_z \leftarrow \begin{bmatrix} H_z \\ 0 \\ \vdots \\ 0 \end{bmatrix}; z_k \leftarrow \begin{bmatrix} z_k \\ w_k \\ \vdots \\ w_{k+N_w} \end{bmatrix}; \tilde{z}_k \leftarrow \begin{bmatrix} \tilde{z}_k \\ \tilde{w}_k \\ \vdots \\ \tilde{w}_{k+N_w} \end{bmatrix}; w_k \leftarrow w_{k+N_w+1};$$

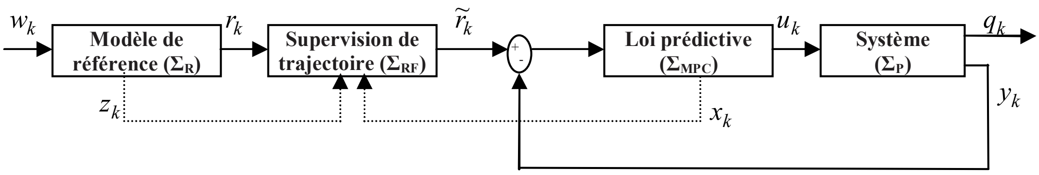
D'où la réécriture du problème d'optimisation :

$$\tilde{r}_k = H_Z \tilde{z}_k^{\,*}$$

$$\tilde{z}_k^{\,*}(z_k, x_k) = \arg\min_{\tilde{z}_k} \sum_{j=0}^{N_w} (H_Z\tilde{z}_{k+j} - H_Z z_{k+j})^T S (H_Z\tilde{z}_{k+j} - H_Z z_{k+j}) \tag{5.77}$$

$$\text{tel que} \quad \begin{cases} z_{k+j+1} = A_Z z_{k+j} \\ \tilde{z}_{k+j+1} = A_Z \tilde{z}_{k+j} \\ x_{k+j+1} = Ax_{k+j} + Bu_{k+j}^{MPC}(x_{k+j}, \tilde{z}_j) \end{cases} ; \begin{bmatrix} \tilde{z}_{k+j} \\ x_{k+j} \end{bmatrix} \in X_f ; j = 0,.., N_w$$

Le schéma de régulation modifié est représenté Figure 5.25.




**MPC avec garantie de faisabilité**

**Figure 5.25 : Schéma MPC pour le suivi de trajectoire avec garantie de faisabilité par supervision de la référence**

En ce qui concerne le problème d'optimisation (5.77), notons que l'argument optimal $z^*$ dépend de la paramétrisation en $\{x_k, z_k\}$, mais en revanche les contraintes dépendent exclusivement du vecteur $x_k$ de sorte que $z_k$ ne subit pas de restriction, en ayant comme domaine d'existence $z_k \in \Re^p$. Cet aspect est très important car les contraintes sur $z_k$ représentent en fait des limitations sur le signal exogène, et donc encore une fois des risques d'infaisabilité. En appliquant le résultat de (5.77) en conjonction avec la loi MPC $u_{k+j}^{MPC}(x_{k+j}, \tilde{z}_j)$, la faisabilité du système asservie est assurée à chaque pas.

**Exemple 5.43 :** On revient au système :

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u_k, u_k \in U = \{u \in \Re \mid -1 \le u \le 1\}, x_k \in \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| -10 \le x_1 \le 10; -10 \le x_2 \le 10 \right\}$$

$$z_{k+1} = 0.8 z_k + 0.2 w_k$$

Pour une loi MPC avec horizon de prédiction $N = 2$, plusieurs consignes en créneaux sont testées, présentant des amplitudes de plus en plus grandes pour déterminer la présence d'infaisabilité et faire intervenir le superviseur de trajectoire pour éviter ce phénomène. Ainsi, Figure 5.26a, on constate que la loi MPC reste faisable pour cette trajectoire d'amplitude 5. La Figure 5.26b montre une simulation pour une amplitude de 8 nécessitant 9 activations du superviseur. Une amplitude de 15 qui n'est pas compatible avec les contraintes dures du système va forcer l'activation du superviseur à chaque pas d'échantillonnage mais on peut observer Figure 5.26c que la trajectoire supervisée tend vers une valeur établie de 10, maximum du domaine autorisé.

### 5.3.2 Implémentation

Les développements suivants envisagent principalement le cas $N_w = 1$ pour démontrer certaines propriétés. Le cas $N_w > 1$ sera ensuite abordé en soulignant ses spécificités. La principale propriété utilisée considère que les deux blocs décisionnels, ($\Sigma_{RF}$) et ($\Sigma_{MPC}$), basés sur des problèmes d'optimisation multiparamétriques, peuvent être exprimés sous forme explicite d'après les chapitres précédents par deux fonctions affines par morceaux :

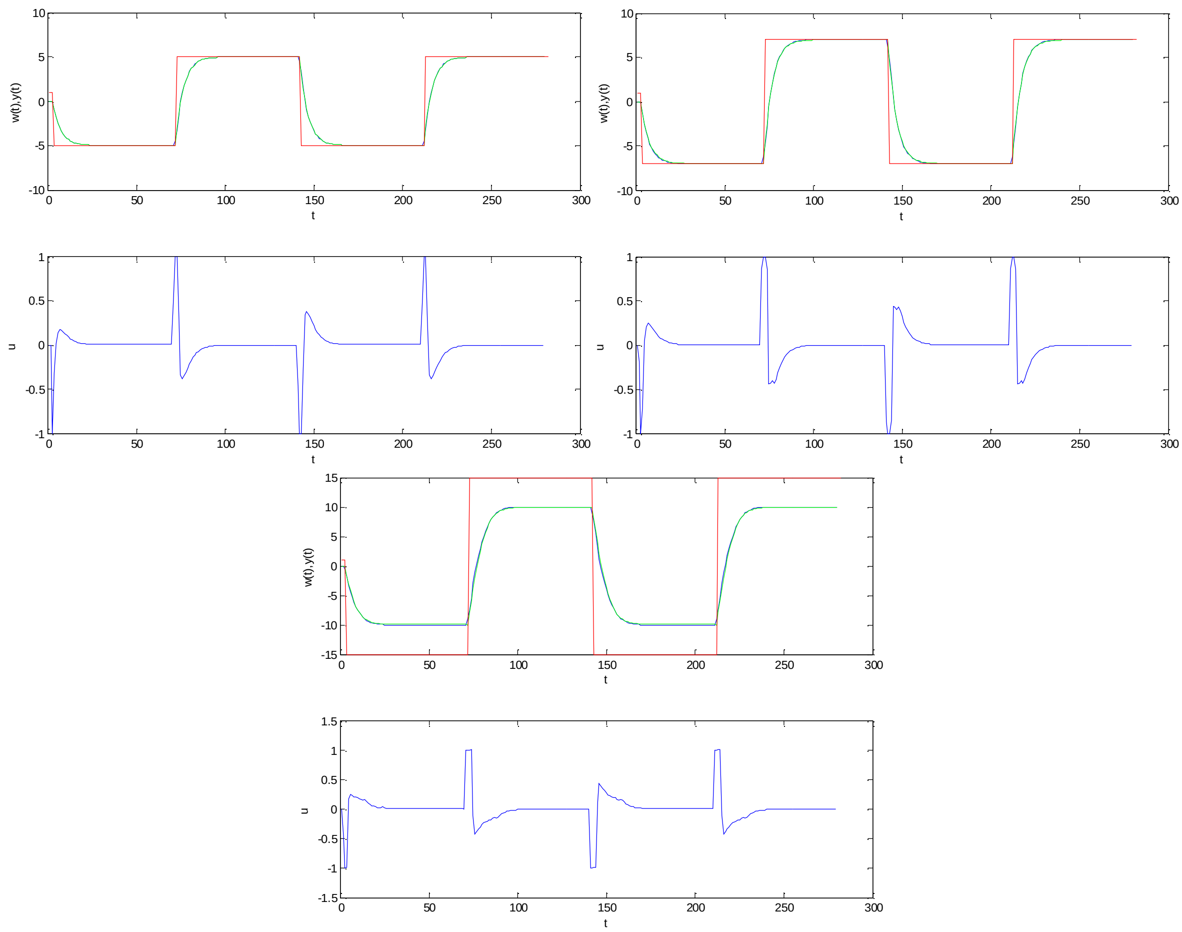



**Figure 5.26 : Simulations de trajectoire avec superviseur de référence**
**a) Trajectoire d'amplitude 5 faisable sans action du superviseur**
**b) Trajectoire d'amplitude 8 faisable avec action du superviseur**
**c) Trajectoire d'amplitude 15 faisable avec superviseur dans les limites du domaine**

$$\tilde{z}_k = L_x^i x_k + L_z^i z_k + \lambda^i \text{, pour } i \text{ tel que } \begin{bmatrix} x_k & z_k \end{bmatrix}^T \in R_i \tag{5.78}$$

$$u_k = K_x^i x_k + K_z^i \tilde{z}_k + \kappa^i \text{, pour } i \text{ tel que } \begin{bmatrix} x_k & \tilde{z}_k \end{bmatrix}^T \in D_i \tag{5.79}$$

**Remarque 5.44** : La réunion $R = \bigcup_{i=1}^{\rho} R_i$ représente le domaine pour lequel les garanties de faisabilité de la loi MPC sont validées. La complexité de découpage sera intimement liée à l'espace que l'utilisateur veut se examiner concernant le signal exogène. Pour diminuer la complexité de la formulation explicite du superviseur de trajectoire, des contraintes $\tilde{z} \in Z$ peuvent être ajoutées dans la formulation (5.77). La possibilité de décrire explicitement les deux blocs de commande indique que le mécanisme qui garantit la faisabilité de la commande est en fait implémenté par une succession de mécanismes de positionnement (Figure 5.27).

Etant donné que l'implémentation temps réel d'un tel mécanisme implique en fait la construction d'un arbre binaire de recherche, il peut s'avérer intéressant d'essayer de concaténer les deux fonctions linéaires par morceaux en une seule loi de commande incluant également le mécanisme de supervision, qui serait donc faisable 'partout'.

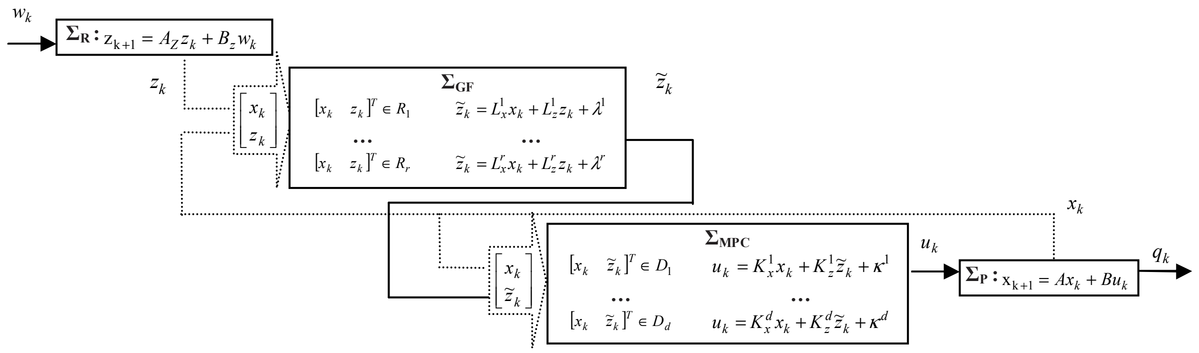



**Figure 5.27 :Schéma MPC avec garantie de faisabilité par supervision de la référence formulation explicite**

**Proposition 5.45 :** Soient deux fonctions linéaires affines par morceaux :

$$f : R \to D;\ R = \bigcup_{i=1}^{r} R_i;\ et\ f(x) = A_{f_i} x + b_{f_i}\ \ \forall x \in R_i\ ,\ (R\ ,R_i\ \text{ensembles convexes})$$

$$g : D \to F;\ D = \bigcup_{i=1}^{d} D_i;\ et\ g(x) = A_{g_i} x + b_{g_i}\ \ \forall x \in D_i\ ,\ (D\ ,D_i\ \text{ensembles convexes})$$

Il existe une fonction :

$$h : R \to F;\ R = \bigcup_{i=1}^{n_h} DR_i;\ et\ h(x) = A_{h_i} x + b_{h_i} = g(f(x))\ \forall x \in DR_i\ ,\ (DR_i\ \text{ensembles convexes}). \quad \blacksquare$$

**Preuve :** Pour démontrer l'existence du découpage $D = \bigcup_{i=1}^{n_h} DR_i$ il suffit de construire pour chaque $R_i, i = 1,..,r$ tous les ensembles $DR_{ij} = \{x | x \in R_i\ et\ f(x) \in D_j\}, j = 1,..,d$. Seuls les ensembles non nuls sont retenus et associés aux fonctions $h(x) = A_{g_j} A_{f_i} x + A_{g_j} b_{f_i} + b_{g_j}$. On obtient $R_i = \bigcup_j DR_{ij}$ car par hypothèse $f(R_i) \subset D$ et donc l'existence d'un $n_h$ fini est assuré tel que $h : D \to F;\ D = \bigcup_{i=1}^{n_h} H_i;\ et\ h(x) = A_{h_i} x + b_{h_i} = f(g(x))\ \forall x \in H_i$.

**Remarque 5.46 :** Il est important de noter que $n_h \le n_d * n_e$ et comme la complexité d'algorithmes d'évaluation de fonctions $f(.)$, $g(.)$, $h(.)$ dans ce cas est logarithmique dans le nombre de partitions, la fonction résultante $h(.)$ possède une complexité d'évaluation inférieure à l'évaluation séquentielle de $f(.)$ et $g(.)$.

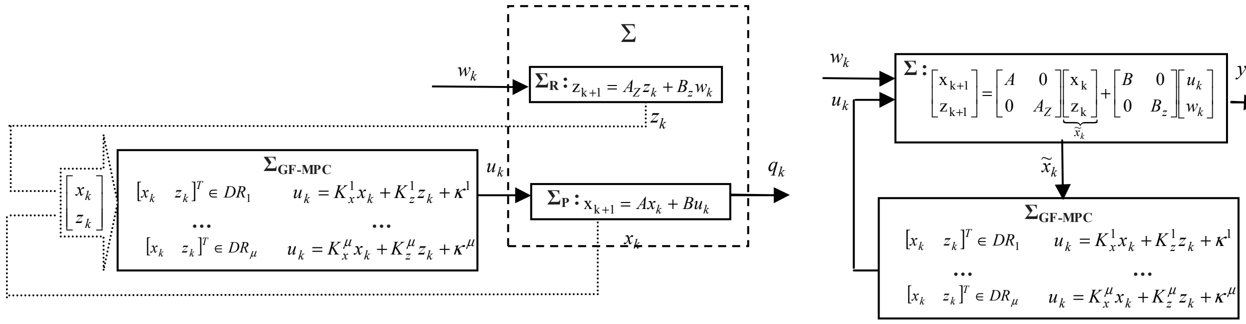



**Figure 5.28 :Schéma MPC avec garantie de faisabilité par supervision de la référence a) une seule formulation explicite b) formulation compacte**

Avec ce résultat on peut décrire la procédure de synthèse d’une loi MPC avec garantie de faisabilité (La partie droite de la Figure 5.28 est une représentation compacte de la partie gauche).

**Algorithme 5.47**

1) Construire la loi MPC en ignorant le signal exogène. Développer sa solution explicite :

$$u = K_x^i x + K_z^i z + k^i; \quad pour \quad [x \quad z]^T \in D_i, i = 1,..,d$$

2) Déterminer le domaine de faisabilité de cette loi $\widetilde{X}_f = \bigcup_{i=1}^{d} D_i$ .

3) Construire la solution explicite du problème d’optimisation lié au mécanisme de supervision de trajectoire.

$$\widetilde{z} = L_x^i x + L_z^i z + \lambda^i; \quad pour \quad [x \quad z]^T \in R_i, i = 1,..,r$$

4) Concaténer les deux fonctions linéaires par morceaux en une seule représentation :

Pour $i = 1,..,r$

  Pour $j = 1,..,d$

    Programmer $DR_{ij} = \{x | x \in R_i \ et \ f(x) \in D_j\}, j = 1,..,d$

    Si $DR_{ij}$ est de dimension pleine, retenir ce domaine et associer la loi :

$$u = K_x^i x + K_z^i L_x^j x + K_z^i L_z^j z + k^i + K_z^i \lambda^j$$

  Fin

Fin

**Remarque 5.48 :** Concaténer les deux descriptions explicites en une seule loi de commande affine par morceaux est plus complexe du point de vue mémoire car au lieu de deux tableaux simples on doit traiter un seul tableau beaucoup plus grand. Ce désavantage est compensé par les améliorations du temps de positionnement en ligne. Si pour le cas non compacté on devrait dans le pire cas parcourir un arbre de profondeur égale à la somme des profondeurs de chaque tableau, dans la version unifiée la complexité de l’arbre de recherche est réduite au minimum car les zones inutiles du deuxième tableau sont éliminées de l’énumération.

Le cas où $N_w$ est plus grand que l’horizon de prédiction peut être inclus dans le cas précédent car on peut toujours intégrer toute la connaissance du signal exogène dans la représentation d’état du modèle à suivre. Ensuite la loi prédictive peut être synthétisée et toute la procédure de supervision de trajectoire est applicable sans modification.

Une idée intéressante consiste à diminuer l’horizon de prédiction et à élargir le domaine faisable par une procédure qui garantit la faisabilité. Même si les performances peuvent être améliorées pour quelques familles de consignes (faibles horizons de prédiction), la complexité de la formulation explicite n’est pas réduite et en général une augmentation de l’horizon de prédiction offre une augmentation de l’ensemble faisable et une maîtrise des performances de commande. La procédure permettant d’éviter l’infaisabilité est favorable uniquement dans le cas où l’information sur le signal exogène est limitée, diminuant ainsi la prédiction.

**Remarque 5.49 :** Du point de vue mémoire requise, l’utilisation de la formulation compacte peut parfois devenir intéressante car, pour la loi MPC, on ne conserve que la formulation explicite de l’action de commande immédiate et non toute la séquence optimale. En revanche, lors d’une implémentation séparée, le superviseur de trajectoire doit garder toute la séquence optimale et donc la formulation explicite sera plus consommatrice de ressource mémoire.

## *5.4 Relation entre faisabilité et stabilité*

Si, jusqu’à ce stade, la faisabilité a été la caractéristique principale à acquérir pour une loi MPC, il faut également envisager la mise en œuvre de conditions qui, si elles sont remplies, permettront de dire que faisabilité implique la stabilité.

Malgré tout, les résultats de stabilité liés exclusivement à la faisabilité sont peu répandus. Par exemple, si tous les signaux du système sont explicitement ou implicitement limités, alors la faisabilité implique une stabilité de type BIBO. On demande en fait souvent des résultats plus forts comme par exemple la stabilité asymptotique ou exponentielle. Un outil pratique pour l'analyse de stabilité d'un système en boucle fermée est lié à l'existence d'une fonction de Lyapunov, qui dans le cas prédictif se construit autour de la fonction de coût. Pour cela, considérons la loi MPC, résultat d'une implémentation selon le principe de l'horizon glissant de la solution optimale du problème :

$$\begin{aligned} &\min_{\mathbf{k}_u(t)} J(t) \\ &\mathbf{k}_u(t) \in U(x_t) \end{aligned} \tag{5.80}$$

avec :

$$\mathbf{k}_u(t) = \left\{u_{t|t}, u_{t+1|t}, \ldots, u_{t+N-1|t}\right\}$$

et $U(x_t)$ un ensemble convexe pour chaque $x_t$ est défini tel que l'origine soit contenue à l'intérieur de l'ensemble $U(\mathbf{0})$. Pour ce problème, on peut retenir les combinaisons optimales (argument et coût) pour deux instants consécutifs par :

$$\mathbf{k}_u^*(t) = \left\{u_{t|t}^*, u_{t+1|t}^*, \ldots, u_{t+N-1|t}^*\right\} \to J^*(t) \tag{5.81}$$

$$\mathbf{k}_u^*(t+1) = \left\{u_{t+1|t+1}^*, u_{t+2|t+1}^*, \ldots, u_{t+N|t+1}^*\right\} \to J^*(t+1) \tag{5.82}$$

L'argumentation de la stabilité peut être basée sur l'existence d'une solution faisable à l'instant $t+1$ :

$$\mathbf{k}_u^{fais}(t) = \left\{u_{t+1|t}^*, u_{t+2|t}^*, \ldots, u_{t+N-1|t}^*, u_{t+N}^{fais}\right\} \to J^{fais}(t+1) \tag{5.83a}$$

qui satisfait la relation :

$$J^*(t+1) \le J^{fais}(t+1) \le J^*(t) \tag{5.83b}$$

La première inégalité ne soulève aucun problème, car elle est vérifiée par définition ($J^*(t+1)$ est l'optimum quant à $J^{fais}(t+1)$, il est seulement faisable). Pour résumer, (5.83b) implique que la série $J^*(t)$ est non croissante et comme par ailleurs elle est bornée inférieurement par 0 (comme somme de termes quadratiques), il résulte que $J^*(t)$ converge, $J^*(t) \xrightarrow{t\to\infty} l$. Il faut alors s'assurer que $l=0$.

En fait les fonctions de coût ont été définies par :

$$J(t) = x_{t+N|t}^T P x_{t+N|t} + \sum_{k=1}^{N-1} \left\{ x_{t+k|t}^T Q x_{t+k|t} + u_{t+k|t}^T R u_{t+k|t} \right\} \tag{5.84}$$

où $P, Q$ sont des matrices semi définies positives et $R$ est positive définie. On peut distinguer plusieurs situations caractéristiques pour les concepts qui assurent la stabilité.

*1) Loi prédictive choisie telle que* $x_{t+N|t} = 0$

Dans ce cas l'existence d'une solution candidate (faisable) :

$$\mathbf{k}_u^{fais}(t) = \left\{u_{t+1|t}^*, u_{t+2|t}^*, \ldots, u_{t+N-1|t}^*, 0\right\}$$

est assurée et la fonction de coût satisfait :

$$J^*(t+1) \le J^{fais}(t+1) = J^*(t) - x_t^T Q x_t - u_{t|t}^T R u_{t|t} \le J^*(t)$$

et donc les hypothèses (5.83a-5.83b) sont validées. Il reste à démontrer que $l=0$. Or :

$$x_t^T Qx_t + u_{t|t}^T Ru_{t|t} \le J^*(t) - J^*(t+1) \Rightarrow \lim_{t\to\infty}(x_t^T Qx_t + u_{t|t}^T Ru_{t|t}) \le \lim_{t\to\infty} J^*(t) - \lim_{t\to\infty} J^*(t+1) = 0$$
$$\Rightarrow \lim_{t\to\infty} x_t = 0;\ \lim_{t\to\infty} u_t = 0$$

et ainsi la stabilité de la boucle fermée est démontrée.

*2) Systèmes stables en boucle ouverte*

L'existence de la même solution candidate (faisable) :

$$\mathbf{k}_u^{fais}(t) = \left\{u_{t+1|t}^*, u_{t+2|t}^*, \ldots, u_{t+N-1|t}^*, 0\right\}$$

est assurée si le domaine de contraintes prend en compte un ensemble terminal inclus dans l'ensemble maximal admissible du système autonome $\mathrm{O}_\infty(A)$. En choisissant le coût terminal donné par un terme quadratique pondéré par la solution de l'équation de Lyapunov :

$$\mathrm{P} = \mathrm{A}^{\mathrm{T}} PA + Q$$

on assure :

$$J^*(t+1) \le J^{fais}(t+1) = J^*(t) - x_t^T Qx_t - u_{t|t}^T Ru_{t|t} \le J^*(t)$$

car :

$$x_N^T Px_N = x_{N+1}^T Px_{N+1} + x_N^T Qx_N$$

et donc le résultat est similaire au point précédent.

*3) Systèmes stabilisables en boucle ouverte*

Une solution candidate peut être :

$$\mathbf{k}_u^{fais}(t) = \left\{u_{t+1|t}^*, u_{t+2|t}^*, \ldots, u_{t+N-1|t}^*, K_{LQ} x_N\right\}$$

avec :

$$K_{LQ} = -(R + B^T PB)^{-1} B^T PA$$
$$P = Q + K_{LQ}{}^T RK_{LQ} + (A + BK_{LQ})^T P(A + BK_{LQ})$$

Sa faisabilité sera assurée si le domaine de contraintes prend en compte un ensemble terminal inclus dans l'ensemble maximal admissible du système bouclé par cette loi linéaire $\mathrm{O}_\infty(A + BK_{LQ})$. En choisissant le coût terminal donné par un terme quadratique pondéré par la solution de l'équation de Riccati précédente, on assure :

$$J^*(t+1) \le J^{fais}(t+1) = J^*(t) - x_t^T Qx_t - u_{t|t}^T Ru_{t|t} \le J^*(t)$$

car :

$$x_N^T Px_N = x_{N+1}^T Px_{N+1} + x_N^T Qx_N + u_N^T Ru_N$$

et donc on peut retrouver la stabilité de la boucle fermée.

L'analyse de ces trois classes de systèmes pour lesquelles les résultats de stabilité peuvent être établis par la construction d'une solution faisable à partir de la solution courante nous conduit vers la solution générale :

- Garantir la faisabilité d'une structure basée sur le principe de l'horizon glissant,
- Garantir l'existence parmi les solutions faisables d'un candidat défini par prolongation de la solution optimale de l'instant précédent,
- Ajuster la fonction de coût MPC de telle façon qu'elle satisfasse :

$$J^*(t+1) \le J^{fais}(t+1) \le J^*(t)$$

Lors de l'étude de la faisabilité, les principaux paramètres de réglage ayant une influence claire sur cette propriété étaient principalement les horizons et le jeu de contraintes terminales. On s'aperçoit maintenant que, lorsque l'on parle de stabilité, le choix de la fonction de coût joue un rôle déterminant au moins pour assurer la monotonie, ce qui constitue une condition nécessaire pour pouvoir l'assimiler à une fonction de Lyapunov.

> **Remarque 5.50 :** Les développements ont été menés en supposant que le problème d'optimisation MPC a une solution exacte. Les garanties de faisabilité peuvent-elles être perdues si l'on considère des solutions sous-optimales ? La réponse dépend en fait de la stratégie mise en œuvre pour garantir la faisabilité. Si l'on a utilisé des ensembles terminaux (stratégie de conception 2 du paragraphe 5.2.3), alors la faisabilité sera préservée car une troncature de la solution courante reste faisable à l'instant suivant. Si l'on s'est basé sur l'équivalence entre le problème à horizon infini et celui à horizon fini (stratégie de conception 1 du paragraphe 5.2.3), alors la sous-optimalité fait perdre cette caractéristique. La première solution est donc préférable.

Pour conclure sur ce point et pour garantir la stabilité de la commande prédictive sous contraintes pour les systèmes linéaires invariants dans le temps, une des directions les plus utilisées se base sur le concept 'd'ensemble invariant' [GT91]. L'ensemble invariant dans un contexte prédictif est l'ensemble objectif pour la trajectoire du système. Il doit être atteint pour un horizon de prédiction donné. Cette approche est connue sous le nom de paradigme du 'mode dual' [MRRS00], [Ros03] parce que l'effort de commande est scindé en deux modes, le premier assurant la satisfaction des contraintes et l'évolution vers l'espace invariant, tandis que le deuxième mode est représenté par un régulateur linéaire n'activant pas les contraintes (il s'agit en fait d'un problème de commande sans contraintes) une fois arrivé dans l'espace invariant.
La preuve de stabilité est donnée par le fait que le critère d'invariance positive est équivalent à l'existence d'une fonction de Lyapunov [Ros03], [VHB88].

Globalement, la liaison entre faisabilité, ensembles invariants et stabilité de la commande prédictive sous contraintes, est résumée par [MRRS00].

> **Théorème 5.51 [MRRS00]** : Pour une loi de commande prédictive $K : X \to \Re^m$ satisfaisant les conditions :
>
> A1 $X_f \subset X$ , $X_f$ fermée, $0 \in X_f$ . (ensemble terminal)
>
> A2 $\exists K_f(x) \in U, \forall x \in X_f$ (loi associée à l'ensemble terminal satisfaisant les contraintes sur la commande)
>
> A3 $\forall x_t ; x_t \xrightarrow{K_f(x_t)} x_{t+1}\ t.q.\ x_{t+1} \in X_f$ ( $X_f$ est positif invariant par rapport à la loi $K_f(.)$ )
>
> A4 $\exists F : \Re^n \to \Re \quad t.q. \quad F(x_{t+1}) - F(x_t) + l(x_t, k_f(x_t)) \le 0$ où $l(x_t, k_f(x_t))$ représente l'apport de l'état $x_t$ à la fonction de coût (Il existe une fonction de Lyapunov définie sur l'ensemble $X_f$ )
>
> la stabilité de la loi de commande est assurée à condition que la loi $K : X \to \Re^m$ amène l'état dans l'ensemble terminal $X_f$ à la fin de l'horizon de prédiction (donc que la loi soit faisable).

Ce résultat fournit les conditions suffisantes de stabilité.

## 5.5 *Influence des paramètres MPC sur la faisabilité*

Les différents paramètres d'une loi MPC influencent la faisabilité, et leur interaction fait qu'il est difficile de discerner l'impact de chacun. Quelques règles générales peuvent être formulées, basées sur la comparaison entre le problème ($P^{MPC}$) et ( $P^{MPC}_\infty$ ) [23] :

$$\begin{aligned}
&\Phi^\infty(x_0) = \min_{k_u=\{u_0,u_1,u_2,\ldots\}} \sum_{k=1}^{\infty} x_k^T Q x_k + \sum_{k=0}^{\infty} u_k^T R u_k \\
&\text{s.t. } x_{k+1} = Ax_k + Bu_k, \forall k \ge 0 \qquad\qquad (P^{MPC}_\infty) \\
&\qquad Cx_k + Du_k \le \gamma, \forall k > 0, \gamma > 0
\end{aligned} \tag{5.85}$$

[23] Dans [BGW90], Bitmead et ses coauteurs recommandent la synthèse à horizon infini pour élaborer des lois avec garantie de stabilité.

ce dernier n'étant soumis qu'à un seul type de comportement infaisable, celui de classe I (donc facile à repérer). En d'autres termes, la loi MPC à horizon infini ( $\mathrm{P}_{\infty}^{\mathrm{MPC}}$ ) est toujours faisable si elle est faisable au moment initial. Malheureusement la détermination de $X_f^{MPC\infty}$ n'est pas immédiate, de sorte que cette loi ne peut donc être utilisée que comme outil théorique.

- $N$, horizon de prédiction. Ce paramètre joue un rôle déterminant au niveau de l'anatomie de la fonction de coût car les prédictions représentent la base de l'estimation des erreurs, qui à leur tour ont un rôle déterminant (avec l'effort de commande) pour la satisfaction des spécifications. En revanche, son influence sur la faisabilité est masquée par le fait qu'il n'interagit pas dans la caractérisation de $X_f$. Malgré tout, l'horizon de prédiction intervient dans la caractérisation de l'infaisabilité par la structure de la loi $\mathbf{k}_u^{MPC}(.)$ et donc par l'éventuelle invariance positive de $X_f$, ou encore dans le cas d'infaisabilité de classe III dans la construction d'un ensemble positif invariant $IF$.

> **Remarque 5.51 :** Du point de vue pratique, avoir une prédiction plus longue est un desideratum général. Il faut cependant vérifier $N \le N_c$ pour éviter d'admettre des prédictions qui n'obéissent pas aux contraintes. On considère donc par la suite avec cette remarque $N = N_c$.

- $N_c$, horizon de contrainte. C'est le paramètre dont l'influence est la plus claire. Il modifie en effet le domaine $X_f$, et bien sûr la structure de la loi $\mathbf{k}_u^{MPC}(.)$. Théoriquement $N_c = \infty$ assure la faisabilité de la loi MPC car la troncature $\{u_1; u_2, ...\}$ de la séquence optimale $\mathbf{k}_u(x^0) = \{u_0, u_1, ...\}$ pour $\mathrm{P}^{\mathrm{MPC}}(x_0)$ reste une séquence faisable pour $\mathrm{P}^{\mathrm{MPC}}(x_1)$. Avoir un horizon de contraintes infini pour un problème MPC n'est pas implantable, il faut donc se contenter d'un horizon fini. La méthode à employer pour choisir un horizon de contraintes fini tout en préservant la faisabilité est la suivante :

1. En ignorant les contraintes, calculer la loi de commande optimale :

$$\begin{aligned} u_k &= K_{LQ} x_k \\ K_{LQ} &= -(R + B^T PB)^{-1} B^T PA \\ P &= A^T PA + Q - A^T P^T B (R + B^T PB)^{-1} B^T PA \end{aligned} \tag{5.86}$$

2. Construire l'espace maximal admissible pour le système bouclé par la loi optimale $\mathrm{O}_\infty(A + BK_{LQ})$,
3. Trouver $N_{opt} = \min\{N \in \Im \,|\, \forall x^0 \in X, \chi_N^{MPC}(N) \in O_\infty(A + BK_{LQ})\}$,
4. Fixer $N_c = N_{opt}$.

- $X_N$, ensemble terminal. Il a été introduit dans le concept du 'mode dual' [MM93] qui, comme il a déjà été dit, décrit une loi MPC en découpant les objectifs de commande en deux parties, une première qui tente d'amener en temps fini le système sur $X_N$ et une deuxième qui (en se basant sur une loi existante stabilisante localement dans $X_N$) prend le contrôle une fois l'ensemble terminal atteint. Il est donc évident que l'existence de cet ensemble terminal est directement liée à celle d'une loi de commande prédéfinie (le plus classiquement il s'agit de la loi linéaire obtenue pour le problème sans contraintes). L'avantage important de l'intégration d'un tel ensemble terminal dans le critère MPC provient du fait que $X_f$ hérite de l'invariance positive de $X_N$ induisant des méthodes efficaces pour assurer la faisabilité (classe II ou III). Une seule hypothèse doit être satisfaite néanmoins, celle de l'invariance positive de $X_N$.

> **Remarque 5.52 :** Le cas d'un horizon de contraintes choisi tel que la loi $u_k = K_{LQ} x_k$ satisfasse les contraintes pour $k \ge N$ est tout à fait équivalent au choix d'un ensemble terminal $X_N = O_\infty$, même si cela ne ressort pas explicitement. En général en choisissant $X_N$ ensemble positif invariant par rapport à la loi $\mathbf{k}_u^{MPC}(.)$, on obtient en fait un problème d'optimisation avec horizon de contrainte pseudo infini.

> **Remarque 5.53 :** On peut se demander comment grouper les deux parties du régulateur à 'mode dual' en une formulation unique, ou comment être sûr au moins qu'une même formulation assure le passage d'un mode à l'autre. La question est intéressante car l'existence d'un ensemble terminal n'assure pas que la

trajectoire du système bouclé va l'atteindre. Cette distinction est subtile et fait en réalité la différence entre les lois MPC faisables et les lois MPC stables. Le paragraphe précédent a montré que pour espérer apporter la stabilité, l'existence d'un ensemble terminal doit s'accompagner de l'existence d'un coût terminal dans la fonction de coût.

- $S(x_N)$, coût terminal. Ce paramètre trouve ses origines dans les tentatives de réglage de la loi MPC pour assurer la stabilité de la boucle fermée dans le cas sans contraintes. Sa présence assure le fait que la fonction de coût est équivalente à celle à horizon de prédiction pseudo infini. Malheureusement, dans le cas avec contraintes, la présence d'un coût terminal ne garantit pas en lui-même ni la faisabilité ni la stabilité. En revanche, il constitue un ingrédient important lorsque l'évolution du système rentre à la fin de l'horizon de prédiction dans un certain ensemble terminal $X_N$ pour lequel il existe une loi $u_k = Kx_k$ satisfaisant :

  $$(C+DK)x_k \in X_N \forall k \geq N$$

  Ensuite on a :

  $$\sum_{k=1}^{\infty} x_k^T Q x_k + \sum_{k=0}^{\infty} u_k^T R u_k = x_N^T P x_N + \sum_{k=1}^{N-1} x_k^T Q x_k + \sum_{k=0}^{N-1} u_k^T R u_k$$

  avec :

  $$P = Q + K^T RK + (A+BK)^T P(A+BK)$$

  et donc on peut proposer une forme de coût terminal :

  $$S(x_N) = x_N^T P x_N$$

  Pour les systèmes stables, un choix possible est de laisser le système en évolution libre à la fin de l'horizon de prédiction, $K = 0$ [RM93] et le coût sera basé sur la solution de l'équation de Lyapunov :

  $$P = A^T PA + Q$$

  Dans le cas général, pour un choix $K = K_{LQ}$, le coût terminal se base sur la solution de l'équation (5.86).

- $N_u$, horizon de commande. Lors de la synthèse de la loi MPC, il est traditionnel de choisir un horizon de commande inférieur à celui de prédiction $N_u \leq N$. La motivation est claire – la causalité du modèle à commander. Un choix $N_u << N$ peut améliorer le temps de calcul lors d'une optimisation en ligne ou peut diminuer la complexité de la formulation explicite. En revanche, du point de vue faisabilité, un tel choix a des conséquences néfastes. Ainsi, chaque commande $u_k = Kx_k$ enlève un degré de liberté à la commande globale et l'ensemble $X_f$ est réduit en conséquence, augmentant donc le risque d'infaisabilité.

  **Remarque 5.54 :** Si $N_c = N$, et $S(x_N) = x_N^T P x_N, P = Q + K^T RK + (A+BK)^{\mathrm{T}} P(A+BK)$, alors le problème $P^{MPC}(N, N_u, Q, R, X_N)$ est équivalent à un autre problème $P^{MPC}(N_u, N_u, Q, R, \widetilde{X}_N)$ où $\widetilde{X}_N$ est construit à partir de $X_N$ par une procédure similaire à (5.65) et (5.66).

- $K$, gain caractérisant la loi de commande à la fin de l'horizon de commande. Ce paramètre intervient exclusivement dans le cas où $N_c > N_u$. Comme indiqué auparavant, la présence de contraintes de ce type sur les commandes restreint la liberté de la loi de commande et donc la faisabilité. Parmi les gains possibles, il est utile de choisir celui qui élargit le plus le domaine $X_f$. Si le modèle du système à commander est instable en boucle ouverte, la loi définie par le gain $K$ doit le stabiliser en boucle fermée.

- $R,Q$, pondérations. Elles caractérisent la fonction de coût et leur structure détermine la répartition entre l'effort de commande et l'erreur à minimiser. Du point de vue faisabilité, leur influence se fait sentir par la forme de la loi MPC. En modifiant les pondérations $R,Q$, l'ensemble $X_f$ demeure inchangé mais la loi $\mathbf{k}_u^{MPC}(.)$ est transformée et en conséquences les propriétés d'invariance positive des ensembles qui en dépendent

## *5.6 Conclusions*

Ce chapitre a présenté des résultats liés à la faisabilité des lois de commande prédictive sous contraintes par la faisabilité des problèmes d'optimisation associés. L'un des concepts essentiels pour atteindre cet objectif est celui d'invariance positive. Ainsi, si les ensembles terminaux spécifiés pour un horizon de prédiction donné valident la propriété d'invariance positive par rapport à une loi prédéfinie, la faisabilité à l'instant initial implique la faisabilité globale de la loi de commande prédictive.

Ces idées ont été détaillées et structurées sous forme de résultats originaux proposant des conditions nécessaires et des conditions suffisantes pour la faisabilité d'une loi prédictive basée sur le principe de l'horizon glissant, et élaborant une séquence de commande résultant d'un problème d'optimisation multiparamétrique. Les causes d'infaisabilité ont également été décrites et ont permis leur classification en fonction des rapports entre trois ensembles : ensemble terminal, ensemble des états admissibles, ensemble des états initiaux).

Finalement, pour le phénomène d'infaisabilité se manifestant lors de la poursuite de consignes et se caractérisant par une incompatibilité avec les contraintes imposées lors de la synthèse de la loi de commande, un mécanisme de récupération de la faisabilité a été présenté, qui, étant à son tour basé sur un problème d'optimisation multiparamétrique, permet une expression explicite par une fonction linéaire par morceaux. La concaténation de la loi prédictive sous contraintes avec ce mécanisme de récupération de faisabilité propose un schéma compact de commande avec garantie de faisabilité. Son implantation se réduit à une fonction linéaire par morceaux dans l'espace des paramètres de contexte.

Tous les développements présentés à ce stade du mémoire (mise en forme du problème d'optimisation, analyse géométrique, solutions explicites ou analyse de faisabilité) ont traité le cas de modèles n'étant pas affectés par les perturbations ou des incertitudes. Les éléments nécessaires pour suivre, à partir d'un modèle nominal, les principes de la commande prédictive et élaborer une loi de commande non linéaire (linéaire par morceaux) prenant en compte explicitement les contraintes ont ainsi été détaillés.

Malheureusement, lors de la mise en œuvre ce type de lois de commande, l'influence des perturbations n'est pas négligeable, d'une part, et la commande des systèmes doit répondre aux questions de robustesse d'autre part. Pour satisfaire ces besoins et dans le prolongement des développements précédents, le chapitre suivant propose d'étudier les transformations requises par la prise en compte d'incertitudes lors de la synthèse de lois prédictives, les modifications induites au sein des problèmes d'optimisation associés et les méthodes pouvant conduire à des lois explicites robustes. Une fois cet aspect traité, tous les éléments nécessaires à l'implémentation des lois prédictives pour des application les plus diverses auront été mis en œuvre.

# 6. Commande prédictive robuste. Cas sous contraintes.

La conception de la loi de commande prédictive pour le système nominal est habituellement fondée sur deux hypothèses :

a) il n'y a aucune incertitude sur le modèle,
b) la perturbation a un comportement bien défini (l'hypothèse la plus classique consiste à dire que les perturbations conservent un niveau constant au-delà de l'horizon de prédiction).

La loi de commande résultante présente en conséquence de bonnes performances pour le modèle nominal mais des performances dégradées en présence d'incertitudes et/ou de perturbations sur le système. De plus, en présence de contraintes, l'existence de ces incertitudes et/ou perturbations peut se traduire par une violation des contraintes.

Prendre en compte les incertitudes de modélisation et les perturbations pour assurer le bon fonctionnement de l'algorithme de commande sur les systèmes réels, impose de mettre en œuvre une robustification des lois de commande prédictive. Des techniques provenant de la théorie de la commande robuste, par exemple la paramétrisation de Youla [Kuc79], sont utilisables pour les lois prédictives sans contraintes ainsi que pour des cas particuliers de contraintes, comme par exemple des contraintes de type égalité, conduisant à manipuler des lois se traduisant par une forme explicite globalement linéaire (par exemple des lois RST pour le cas GPC). Dans le cas général de contraintes mixtes, ces méthodes sont inapplicables et des techniques spécifiques doivent être mises en place pour construire des lois prédictives robustes. Ces méthodes doivent principalement cibler la nature temporelle du problème d'optimisation sur laquelle s'appuie la commande prédictive et essayer de minimiser les effets des perturbations et des erreurs de modèle.

Pour mener à bien cette étude, ce chapitre débute par un rappel des idées de base de la synthèse de commande prédictive robuste. Après une analyse du cas spécifique des contraintes de type égalité, les développements s'orientent ensuite vers les formulations permettant l'implémentation de la stratégie sous la forme de problèmes d'optimisation 'min-max' multiparamétriques. Poursuivant la démarche avec les outils des chapitres précédents, les formulations introduites sont analysées du point de vue géométrique avec comme résultat les solutions explicites obtenues en utilisant le concept de polyèdre paramétré qui caractérise l'espace faisable. Finalement les solutions explicites se structurent par des lois de commande linéaires par morceaux.

Ce chapitre contient des contributions originales sur l'analyse géométrique des domaines faisables que l'on peut retrouver dans [OD05f], [OD05g], [OD05i].

## *6.1 Introduction*

Dans la littérature traitant de la commande à horizon glissant, on trouve, au delà des techniques de réglage, de l'analyse de faisabilité, stabilité et performances, l'observation qu'un comportement aberrant peut souvent être supprimé en améliorant le modèle de prédiction – l'élément primordial de la philosophie prédictive. Malheureusement, le retour à la modélisation et à l'identification est très coûteux, parfois inefficace, et dès lors toute la stratégie de commande doit être reconsidérée.

Or les aspects de la commande prédictive présentés lors des chapitres précédents se sont basés sur un modèle ayant une structure et des paramètres fixes, appelée classiquement *modèle nominal*. Cette démarche est en fait le résultat d'une étape de modélisation au cours de laquelle des simplifications ont été effectuées ou, très souvent, seules les caractéristiques essentielles du comportement ont été retenues en se basant sur l'information incomplète du processus à piloter. Comme ce modèle ne peut offrir une description précise du comportement réel, mais plutôt une approximation de celui-ci, les algorithmes de commande doivent tenir compte des erreurs de prédiction affectant éventuellement l'évolution du système bouclé. Une loi de commande insensible aux incertitudes de modélisation s'appelle une *loi de commande robuste*.

Une loi de commande est dite 'robuste' si elle préserve la stabilité et les performances du système réel malgré les erreurs de modèle et les incertitudes. Le concept même de boucle fermée peut être vu comme une étape dans la quête de séquences de commande robustes. Néanmoins, en Automatique avancée, une loi robuste caractérise les techniques qui prennent en compte explicitement la divergence entre le modèle nominal et le processus réel. Du point de vue de la commande prédictive, cette étape représente en fait le passage vers la prise en compte d'incertitudes de modèle lors de la synthèse des régulateurs.

La présence de contraintes dans la description du problème, ajoutée aux incertitudes de modèle et aux perturbations, complique la synthèse des lois de commande prédictives. Les problèmes d'optimisation qui, dans le cas nominal ont trouvé dans les critères quadratiques une forme d'expression naturelle, ne sont plus appropriés car ils impliquent un moyennage des erreurs de prédiction induites par la présence des incertitudes [ZM93]. En conclusion, les incertitudes doivent apparaître explicitement dans la fonction objectif. L'algorithme de commande doit assurer la minimisation de cette fonction objectif pour la combinaison dans l'ensemble des modèles incertains admissibles la plus défavorable.

Plusieurs modifications des algorithmes MPC classiques ont été proposées pour assurer la stabilité en présence d'incertitudes de modélisation. Ces changements peuvent être groupés selon la stratégie utilisée en quatre catégories :

- **Modifier l'action du régulateur MPC classique en gelant les commandes**. Cette méthode est motivée par le fait qu'on peut toujours stabiliser un système stable en boucle ouverte en rendant le régulateur moins agressif. A la limite, avec un correcteur complètement passif, le système retourne à son comportement en boucle ouverte stable. Ces méthodes se heurtent à plusieurs limitations fondamentales : d'une part le régulateur doit fournir un comportement cohérent en boucle fermée pour évaluer correctement la nécessité du gel, d'autre part les performances peuvent être affectées inutilement si le modèle est précis.
- **Introduire des contraintes sur la fonction de coût.** Dans ce cas, le problème de stabilité robuste comporte l'utilisation de contraintes sur la fonction de coût en plus des contraintes dures sur les commandes et les états du système. Ces contraintes sur la fonction de coût sont choisies de sorte qu'elles empêchent la séquence de coûts optimaux d'augmenter au delà des coûts obtenus pour le modèle nominal. L'optimisation à résoudre fait partie de la classe des problèmes de programmation semi-définie. Il peut être montré que le système bouclé est exponentiellement stable et des perturbations qui décroissent asymptotiquement peuvent être rejetées.
- **Ajouter des contraintes terminales sur l'état.** Cette technique implique l'ajout d'un ensemble de contraintes qui force toutes les évolutions possibles de l'état du système à converger sur un horizon fini, ou à atteindre une région terminale pour laquelle il existe une loi stabilisant le système de façon robuste. Si l'horizon est court, ce type de contraintes peut provoquer des problèmes de faisabilité, ou ces contraintes peuvent être jugées parfois conservatrices et induire des pertes inacceptables de performance.
- **Calculer une séquence de commande pour le pire cas**. C'est l'approche la plus étudiée pour garantir la stabilité robuste. Lee et Yu [LY97] récapitulent le développement des algorithmes *de type 'min-max'*, précisant que les formulations qui suivent une philosophie en boucle ouverte s'avèrent peu performantes. Ils proposent une approche en boucle fermée offrant de meilleures performances et analysent les enjeux informatiques en proposant des approximations améliorant la charge de calcul du problème. L'inconvénient principal de l'approche 'min-max' est que les problèmes d'optimisation s'avèrent très complexes et leur résolution très coûteuse.

Ce chapitre aborde la synthèse des lois prédictives robustes en se basant sur la minimisation d'une fonction de coût pour la combinaison de perturbation la plus défavorable, ce que l'on appelle une approche 'min-max', autorisant de plus la présence des contraintes terminales pour renforcer la stabilité robuste. La démarche s'intéresse tout particulièrement à la géométrie de ces problèmes de type' min-max', qui se réduisent en fait à des problèmes de programmation linéaire multiparamétrique. De ce fait, il est possible de revenir, comme pour les chapitres précédents, aux concepts des polyèdres paramétrés et à la géométrie des domaines faisables, à la différence que dans ce cas, comme on le verra par la suite la fonction de coût étant linéaire, on force l'optimum sous contraintes à se retrouver sur une combinaison des sommets paramétrés.

Du point de vue conception, la commande prédictive suit lors de l'application la philosophie de l'horizon glissant. Ainsi les techniques conventionnelles font l'hypothèse que, pour l'optimisation à résoudre à chaque pas d'échantillonnage, la séquence de commandes est donnée en boucle ouverte. Une telle hypothèse ignore le fait que, pour les instants futurs, les commandes sont en fait construites à partir des nouvelles mesures et donc après la mise à jour des paramètres de contexte, et tiennent compte des perturbations. Pour le cas nominal, faire une telle différence est insignifiant, car le comportement futur est totalement déterministe, mais pour les modèles incertains, cette différence peut améliorer la qualité de la séquence de commande. Pratiquement, prendre en compte une information supplémentaire pour chaque état futur prédit revient à résoudre le problème selon les principes de la programmation dynamique. Cette approche est appelée dans la littérature '*commande prédictive robuste en boucle fermé*e' [LY97] pour la différencier de celle où la séquence des commandes optimales ignore l'apport d'information apporté par le bouclage.

Tous les développements à venir sont réalisés avec des incertitudes bornées par des limitations de type déterministe. Cette précision importante délimite ces développements des études qui considèrent les incertitudes comme des variables stochastiques et réalisent une optimisation de l'espérance du coût sur l'horizon de prédiction [GSD04].

## *6.2 Les contraintes égalité*

Avant de traiter le problème de la commande prédictive robuste dans le cadre général des contraintes linéaires mixtes (égalités et inégalités), considérons le cas particulier de contraintes de type égalité. Des exemples classiques de ce type des problèmes sont constitués par les algorithmes prédictifs avec contraintes à la fin de l'horizon de prédiction.

Envisageons pour formaliser cette démarche le cadre GPC, comme décrit lors du chapitre 2. La robustification de ces types de lois prédictives a été traitée in extenso pour le cas sans contraintes dans [RD02]. L'objectif est ici d'étendre ces résultats au cas spécifique des lois GPC soumises à des contraintes de type égalité, introduites comme contraintes de stabilité à la fin de l'intervalle de prédiction :

$$y_{t+N_2+i} = w_{t+N_2}\ ;\ i = 1..N_c \tag{6.1}$$

ou dans une version aboutissant à une erreur stationnaire nulle pour des consignes d'ordre plus élevé que les échelons :

$$y_{t+N_2+i} = w_{t+N_2+i}\ ;\ i = 1..N_c \tag{6.2}$$

Une telle loi de commande se base sur une optimisation multiparamétrique semblable à (2.16) sous les contraintes (2.34) :

$$\min_{\mathbf{k_u}} \mathbf{k_u^T}(\mathbf{G^T G} + \lambda \mathbf{I}_{\mathbf{N}_u \times \mathbf{N}_u})\mathbf{k_u} + 2\mathbf{p}_t^T \left[\mathbf{ih}\ \ \mathbf{if}\ \ -\mathbf{J}_{(N_2-N_1+1)\times(N_2-N_1+1)}\right]^T \mathbf{G}\mathbf{k_u}$$

$$\left[\mathbf{if}_\gamma\ \ \mathbf{ih}_\gamma\ \ \mathbf{G}_\gamma\ \ -\mathbf{J}\right]\begin{bmatrix}\boldsymbol{\gamma}_{past}\\ \Delta\mathbf{u}_{past}\\ \mathbf{k}_u\\ \mathbf{w}\end{bmatrix} = \begin{bmatrix}0\\0\\ \vdots\\ 0\end{bmatrix},\quad \text{avec } \mathbf{J} = \left[0 \in \Re^{N_c \times N_2}\ \middle|\ \mathbf{I}_{N_c}\right], \mathbf{w} = \begin{bmatrix} w_{t+1}\\ \vdots\\ w_{t+N_2+N_c}\end{bmatrix} \tag{6.3}$$

Les lois de commande explicites obtenues dans ce cas (par le biais de la solution de type (2.74)) rentrent dans la catégorie des régulateurs polynomiaux RST. Si le choix de l'horizon de prédiction, de l'horizon de commande, des contraintes et des pondérations a été réalisé de façon judicieuse [CS91], on dispose d'un système bouclé stable, représenté Figure 2.3.

### 6.2.1 Robustification préservant le transfert stable entrée-sortie

Le fait que la loi de commande prédictive résultante du problème d'optimisation (6.3) permette une expression analytique sous forme RST permet de se demander si les méthodes de robustification applicables pour le cas sans contraintes ne pourraient pas être utilisées dans ce cas sous contraintes.

> **Proposition 6.1 :** Une transformation de la loi RST peut être opérée de façon à minimiser l'effet de bruit dans la boucle de régulation ou pour augmenter la taille de la plus grande incertitude additive admise sans perte de stabilité. Cette transformation affecte la dynamique en régulation mais n'intervient pas dans le transfert entrée-sortie et en conséquence au nominal les contraintes de type égalité restent vérifiées.

> **Remarque 6.2 :** Cette transformation ne peut être interprétée que comme une robustification au sens large car, par rapport aux contraintes, il n'existe aucune garantie que la séquence optimale (conçue via la prédiction) reste faisable en présence de perturbations. Ce problème est connu dans la littérature sous le nom de 'satisfaction robuste des contraintes' ([Ker00]).

Néanmoins, les contraintes à la fin de l'horizon de prédiction sont introduites comme critères de renforcement de la stabilité et ne représentent pas de limitations dures issues des lois physiques. En conséquence, il est possible d'exploiter la structure linéaire des lois de commande résultantes pour appliquer des méthodes classiques de robustification basées sur la paramétrisation de Youla. En effet, la structure RST garantie permet la description de toute la famille de régulateurs stabilisants, conférant au système bouclé le même transfert entrée/sortie que celui déduit du régulateur initial :

$$\begin{aligned} \widetilde{S} &= S - q^{-1}BQ \\ \widetilde{R} &= R + \Delta AQ \\ \widetilde{T} &= T \end{aligned} \tag{6.4}$$

où $Q$ peut être n'importe quelle fonction de transfert stable. Une variante de cette paramétrisation est donnée par :

$$\begin{aligned} \Delta\widetilde{S} &= \Delta S - q^{-1}BQ \\ \widetilde{R} &= R + AQ \\ \widetilde{T} &= T \end{aligned} \tag{6.5}$$

ne répercutant pas l'action intégrale du correcteur par rapport à l'expression (6.3).
On peut représenter graphiquement Figure 6.1 la paramétrisation (6.4) où l'on considère d'une part le système réel décrit par $\{A_0(q^{-1}), B_0(q^{-1})\}$, d'autre part le modèle utilisé lors de la prédiction $\{A(q^{-1}), B(q^{-1})\}$ et le régulateur explicite donné par les trois polynômes $\{R(q^{-1}), S(q^{-1}), T(q^{-1})\}$.

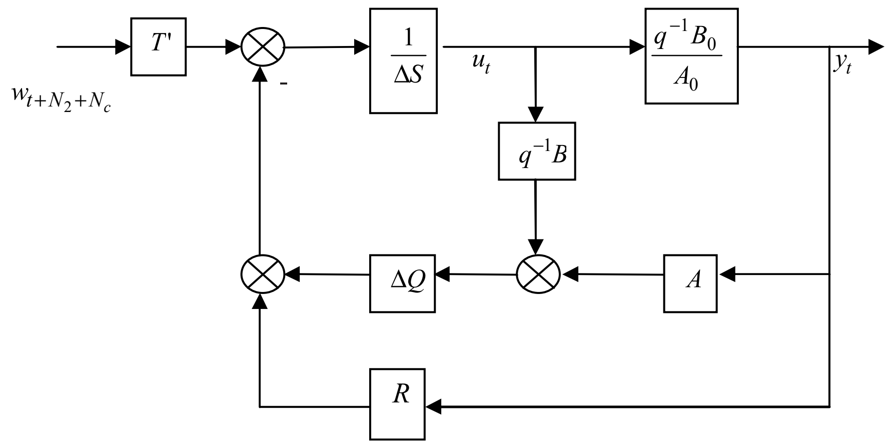



**Figure 6.1 : Régulateur polynomial équivalent avec paramétrisation de tous les régulateurs stabilisants conduisant au même transfert entrée/sortie**

**Remarque 6.3 :** Il existe une différence structurelle entre la formulation classique des régulateurs à deux degrés de liberté RST, où l'on manipule des expressions polynomiales en l'opérateur retard, et l'expression (6.4). En effet, les grandeurs mentionnées dans (6.4) sont des fonctions de transfert car $Q$ est un transfert stable. Pour retrouver des expressions polynomiales, il suffit de décomposer $Q$ selon son numérateur et dénominateur :

$$Q = \frac{Q_{num}}{Q_{den}} \Rightarrow \begin{cases} \widetilde{S} = \dfrac{SQ_{den} - q^{-1}BQ_{num}}{Q_{den}} = \dfrac{\overline{S}}{Q_{den}} \\ \widetilde{R} = \dfrac{RQ_{den} + \Delta AQ_{num}}{Q_{den}} = \dfrac{\overline{R}}{Q_{den}} \\ \widetilde{T} = T \end{cases} \tag{6.6}$$

ce qui est équivalent (Figure 6.2) à l'expression polynomiale :

$$\begin{aligned} \overline{S} &= SQ_{den} - q^{-1}BQ_{num} \\ \overline{R} &= RQ_{den} + \Delta AQ_{num} \\ \overline{T} &= TQ_{den} \end{aligned} \tag{6.7}$$

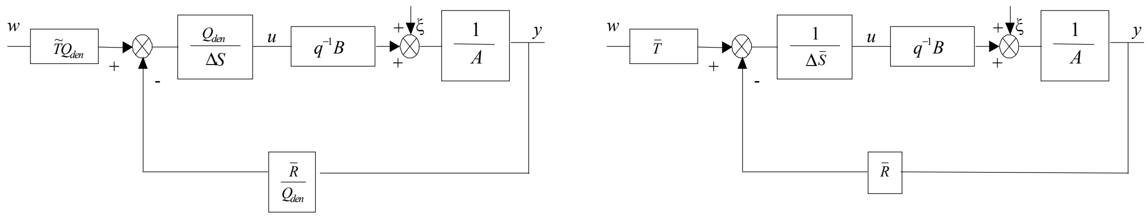



**Figure 6.2 : Equivalence entre un correcteur RST issu de la paramétrisation et un correcteur RST polynomial classique**

Les degrés des polynômes $\{\bar{R}(q^{-1}), \bar{S}(q^{-1}), \bar{T}(q^{-1})\}$ sont affectés par la paramétrisation :

- Degré de $\bar{R}(q^{-1})$ : $n_a + \max(n_{Q_{den}}, 1 + n_{Q_{num}})$
- Degré de $\bar{S}(q^{-1})$ : $n_b + \max(n_{Q_{den}}, 1 + n_{Q_{num}})$
- Degré de $\bar{T}(q^{-1})$ : $N_2 + N_C - N_1 + n_{Q_{den}}$

En fait la paramétrisation (6.4) est une version [KCR00] simplifiée ($Q_2 = 0$) de la paramétrisation de Youla d'un correcteur à deux degrés de liberté :

$$\begin{aligned} \tilde{S} &= S - q^{-1}BQ_1 \\ \tilde{R} &= R + \Delta AQ_1 \\ \tilde{T} &= T - A_0 Q_2 \end{aligned} \tag{6.8}$$

avec $Q_1$ et $Q_2$ des transferts stables et le polynôme $A_0$ provenant de la factorisation selon la dynamique de commande et la dynamique de l'observateur $A_0 A_C = \Delta AS + q^{-1}BR$ de l'équation caractéristique de la boucle fermée avec le correcteur stabilisant initial RST, synthétisé à partir de (6.3).

Comme l'illustre la Figure 6.3, $Q_2(q^{-1})$ modifie le comportement entrée/sortie tandis que $Q_1(q^{-1})$ modifie le comportement lié au rejet des perturbations. Par la suite $Q_2(q^{-1})$ sera considéré nul, en supposant que le comportement entrée-sortie exigé est figé par le régulateur initial, qui tient compte notamment de la présence des contraintes de type égalité sur la prédiction de la sortie à la fin de l'horizon de prédiction.

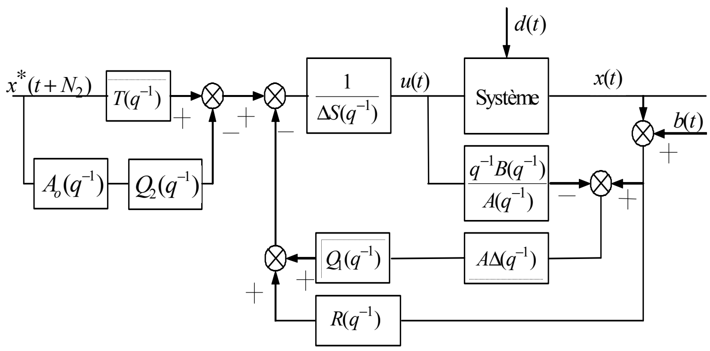



**Figure 6.3 : Forme générale de la paramétrisation de Youla pour les loi de type RST**

En se basant sur cette paramétrisation, des spécifications de robustesse en stabilité face à des incertitudes non structurées peuvent être obtenues. De plus, des spécifications de performance nominale par l'intermédiaire de gabarits temporels peuvent être prises en compte.
La robustification vis-à-vis d'une incertitude est maximisée par la minimisation de la norme $H_\infty$ suivante :

$$\min_{Q_1 \in \Re_\infty} \left\| P(q^{-1}) W(q^{-1}) \right\|_\infty \tag{6.9}$$

où $W$ représente un transfert de pondération pour la bande de fréquence où les incertitudes sont les plus importantes et le transfert $P$ est donné par :

$$P\left(q^{-1}\right)=-\frac{q^{-1}BR}{A_0A_c}-\frac{q^{-1}B\Delta A}{A_0A_c}Q_1 \tag{6.10}$$

pour des incertitudes multiplicatives directes. La minimisation convexe qui en résulte pour trouver le paramètre $Q_1(q^{-1})$ en respectant un gabarit temporel pour le rejet de perturbations et en considérant aussi l'effet de bruit de mesure sur la commande est donnée par :

$$\min_{\substack{Q_1\in\Re H_\infty \\ \Phi_{env1}(Q_1)<0,\Phi_{env2}(Q_1)<0}} \left\|\left(-\frac{q^{-1}BR}{A_oA_c}-\frac{q^{-1}B\Delta A}{A_oA_c}Q_1\right)W(q^{-1})\right\|_\infty \tag{6.11}$$

où $\Phi_{env1}(Q_1)$ définit le gabarit à respecter par la perturbation et $\Phi_{env2}(Q_1)$ le gabarit à respecter par l'effet du bruit de mesure sur la commande. L'ensemble des paramètres $Q_1$ qui respectent une spécification temporelle exprimée par l'intermédiaire d'un gabarit à l'intérieur duquel doit rester la sortie est défini par :

$$\Phi_{env}(Q)=\max\left(\max_{t\geq 0}\left(s_{ij}(t)-s_{\max}(t),s_{\min}(t)-s_{ij}(t)\right)\right) \tag{6.12}$$

avec $s_{ij}(t)$ la réponse d'une fonction de transfert donnée (perturbation/sortie, bruit de mesure/commande) à une entrée déterminée (échelon, …).

Lors de l'implémentation présentée dans [Rod03] ce problème d'optimisation a été approché par un problème de programmation linéaire permettant une synthèse facile des régulateurs robustes préservant le transfert entrée-sortie.

### 6.2.2 Exemple

Considérons l'application de la technique GPC pour la commande en vitesse d'un moteur asynchrone, similaire à celle présenté dans [BD96], pour laquelle le modèle, résultat d'une identification, est défini par la fonction de transfert suivante, à la période d'échantillonnage de 5ms :

$$F\left(q^{-1}\right)=\frac{q^{-1}B(q^{-1})}{A(q^{-1})}=\frac{1.344q^{-1}+3.024q^{-2}}{1-0.98q^{-1}-0.02q^{-2}} \tag{6.13}$$

En choisissant comme paramètres :

$$N_1=1; N_2=4; N_u=6; N_c=3; \lambda=200$$

avec $N_c$ le nombre de contraintes égalité à la fin de l'horizon de prédiction, on obtient la forme polynomiale du régulateur :

$$\begin{aligned} R(q^{-1})&=0{,}2378-0{,}17693q^{-1}-0{,}0036337q^{-2}\\ S(q^{-1})&=1+0{,}54942q^{-1}\\ T(q)&=0{,}0025281q+0{,}0081022q^2+0{,}0066093q^3+0{,}039992q^4 \end{aligned} \tag{6.14}$$

La simulation du fonctionnement de cette loi en présence de bruit de mesure est représentée Figure 6.4, où l'on peut observer Figure 6.4a les conséquences négatives sur l'allure du signal de commande. Avec la paramétrisation de Youla de tous les régulateurs stabilisants qui préservent le comportement entrée-sortie du système, et en utilisant le formalisme de robustification décrit dans [Rod03], pour des gabarits de rejet de perturbation choisis comme indiqué Figure 6.5 et une pondération $W=(1-0{,}7\,q^{-1})/0{,}3$, on obtient un paramètre de Youla d'ordre 2 :

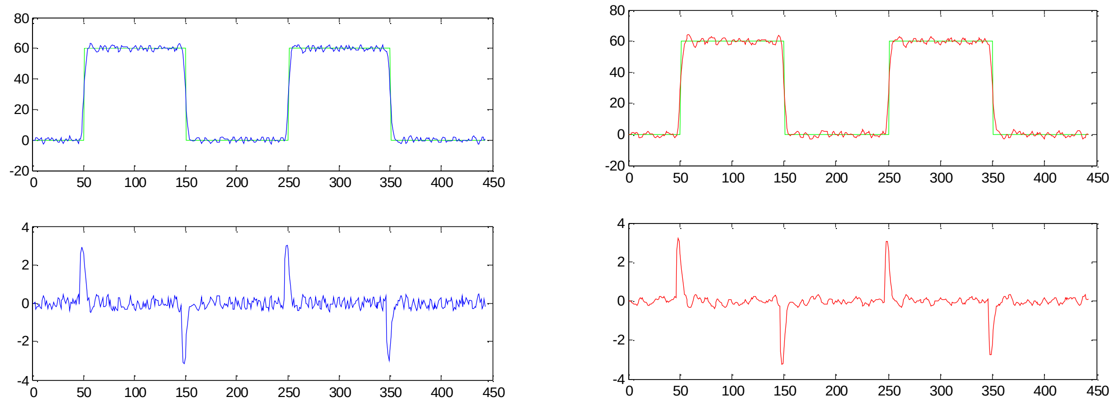


**Figure 6.4 : a) Simulation du comportement du système bouclé avec la loi GPC initiale en présence de bruit b) Même simulation après robustification de la loi RST**

$$Q(q^{-1}) = \frac{-0{,}1567 \ + \ 0{,}0822 \ q^{-1}}{1 \ - \ 1{,}355 \ q^{-1} \ + \ 0{,}5165 \ q^{-2}} \tag{6.15}$$

avec les pôles :

$$p_1 = 0{,}67731 \ + \ 0{,}24039\, i; \quad p_2 = \ 0{,}67731 \ - 0{,}24039\, i$$

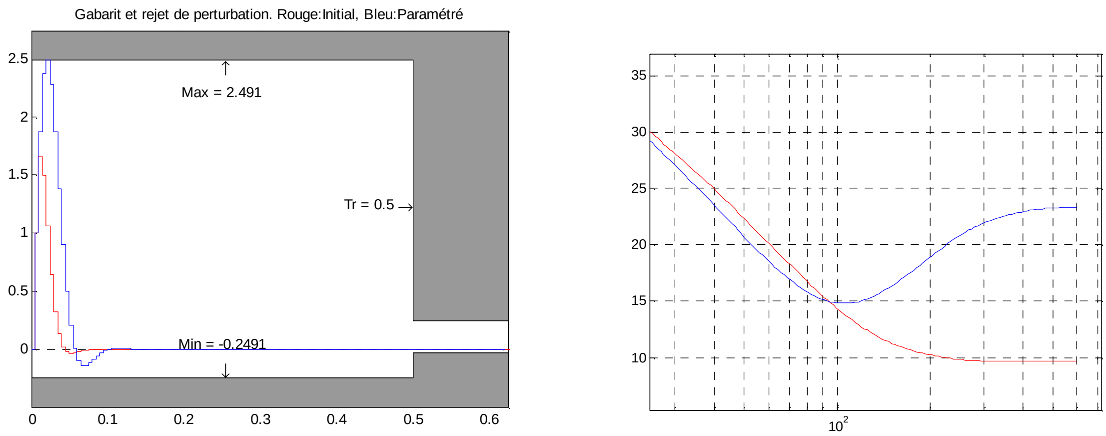



**Figure 6.5 : a) Gabarit et rejet de perturbation pour le régulateur initial (rouge) et pour le régulateur robustifié (bleu) b) Marge de robustesse avec le correcteur initial et robustifié**

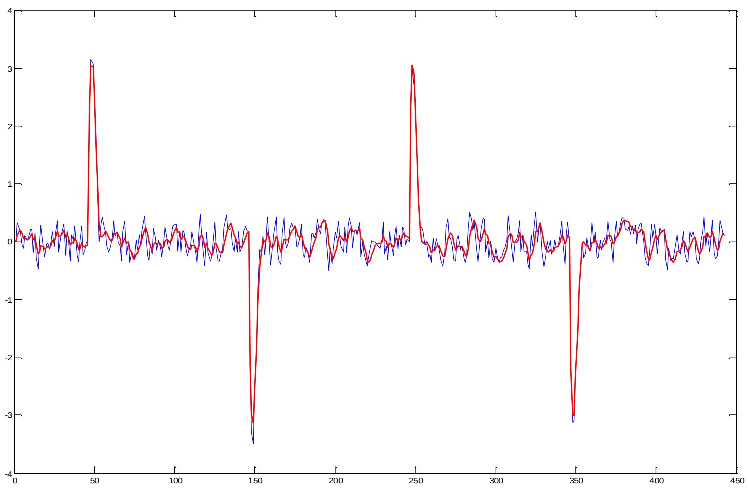


**Figure 6.6 : Détails des signaux de commande pour les simulations de la Figure 6.4**

On observe que l'influence du bruit de mesure est nettement diminuée pour le cas robuste (rouge) en comparaison avec l'influence dans le cas de la loi GPC initiale (courbe beaucoup plus filtrée). Le maintien du même comportement entrée-sortie pour la loi robustifiée est garanti par la construction (6.7).

### 6.2.3 Limites de la méthode

Les développements précédents ont examiné une méthode de robustification se basant sur les techniques classiques de paramétrisation des régulateurs stabilisants pour un modèle nominal. Cette méthode peut être utilisée pour réduire l'influence des perturbations et incertitudes sur les régulateurs linéaires (RST) synthétisés dans le cadre de lois prédictives avec contraintes égalité à la fin de l'horizon de prédiction, mais qui ne sont pas liées à des contraintes dures (provenant de limitations physiques).

Il faut souligner que cette transformation des régulateurs linéaires ne garantit pas une satisfaction robuste des contraintes au delà du modèle nominal. Elle reste donc liée à la conception des lois de commande sans contraintes, leur philosophie étant difficilement généralisable.

On peut alors conclure que la conception des lois prédictives pour les systèmes incertains doit considérer simultanément l'action préjudiciable des erreurs de modélisation et les contraintes de fonctionnement pour pouvoir synthétiser des lois préservant l'hypothèse de faisabilité. Les contraintes terminales représentées cette fois par des inégalités peuvent être généralement choisies selon les principes de l'invariance positive [Bla99] en tenant compte de la présence des incertitudes.

## *6.3 Commande prédictive robuste comme problème d'optimisation linéaire multiparamétrique*

En opposition avec le cas nominal, où des fonctions de coût quadratiques sont employées pour la synthèse des lois prédictives, dans le cas des modèles affectés par des perturbations, la modification du critère de coût pour assurer la convergence même dans le pire cas des erreurs de prédiction conduit à une formulation de lois de commande prédictives robustes se basant sur une optimisation 'min-max'. Par la suite, considérant l'équivalence entre un tel problème et un problème de programmation linéaire multiparamétrique, l'implémentation à l'aide des procédures d'optimisation en ligne se réalise de façon tout à fait naturelle.

### 6.3.1 Commande prédictive généralisée

Comme on l'a déjà introduit lors du chapitre 2, la commande GPC se base sur un modèle CARIMA :

$$A(q^{-1})y_t = B(q^{-1})u_{t-1} + C(q^{-1})\frac{\xi_t}{\Delta(q^{-1})} \tag{6.16}$$

Pour le cas nominal où aucune information supplémentaire n'est disponible sur la nature des perturbations, on a pu considérer $C(q^{-1})=1$. L'étude exhaustive de [Rod03] sur la robustification des lois GPC dans le cas sans contraintes montre que ce polynôme $C(q^{-1})$ peut jouer un rôle important dans la caractérisation du correcteur résultant. Ainsi, si une estimation du bruit agissant sur le système est disponible, le polynôme $C(q^{-1})$ peut servir pour la minimisation de la variance. [CMT87] montre que l'on peut également se servir de ce polynôme comme paramètre de réglage, permettant la robustification au détriment de l'optimalité de la prédiction. Ces techniques sont difficilement transposables au cas des commandes sous contraintes, de sorte qu'il s'avère nécessaire de se tourner vers des méthodes assurant la prise en compte explicite des contraintes et d'information déterministe sur les perturbations lors de la synthèse.

**Stratégie d'optimisation se basant sur les prédictions en boucle ouverte**

A partir du modèle (6.16), le prédicteur à $j$-pas se définit par :

$$\hat{y}(t+j) = \underbrace{F_j(q^{-1})y_t + H_j(q^{-1})\Delta u_{t-1}}_{\text{réponse libre}} + \underbrace{G_j(q^{-1})\Delta u_{t+j-1} + \widetilde{J}_j(q^{-1})v_{t+1}}_{\text{réponse forcée}} \tag{6.17}$$

où les polynômes $F_j(q^{-1}), G_j(q^{-1}), H_j(q^{-1}), J_j(q^{-1})$ sont les solutions des équations diophantiennes :

$$\begin{aligned} &\Delta(q^{-1})A(q^{-1})J_j(q^{-1}) + q^{-j}F_j(q^{-1})C(q^{-1}) = C(q^{-1}) \\ &G_j(q^{-1})C(q^{-1}) + q^{-j}H_j(q^{-1})C(q^{-1}) = B(q^{-1})J_j(q^{-1}) \end{aligned} \tag{6.18}$$

de sorte que la perturbation $v_{t+j}$ affecte la prédiction d'une quantité :

$$\widetilde{J}_j\left(q^{-1}\right)=J_j\left(q^{-1}\right)\Delta\left(q^{-1}\right) \tag{6.19}$$

**Remarque 6.4 :** Les coefficients des polynômes $G_j\left(q^{-1}\right)$ représentent en fait les coefficients de la réponse indicielle pour le système décrit par les polynômes $\left(A\left(q^{-1}\right), B\left(q^{-1}\right)\right)$. De manière similaire, les polynômes $J_j\left(q^{-1}\right)$ sont formés des coefficients de la réponse impulsionnelle du système décrit par les polynômes $\left(A\left(q^{-1}\right),1\right)$.

On considère pour la suite que les perturbations sont incluses dans un domaine polyédral explicitement défini par un ensemble de contraintes linéaires :

$$V=\left\{v\in\Re\middle|Mv\leq l;\, l\geq 0\right\} \tag{6.20}$$

Le domaine $V$ est supposé centré sur l'origine, ou tout au moins contenant l'origine, de sorte que le système nominal soit inclus parmi les réalisations possibles des perturbations.

Incluant cette description déterministe des perturbations, la synthèse de la loi de commande prédictive se base alors sur un critère envisageant le cas de perturbations le plus défavorable. Cependant, les fonctions de coût basées sur des termes quadratiques n'offrent pas en présence de perturbations une description fidèle de ce que l'on appelle *'le pire cas'*, et en conséquence la fonction de coût doit être construite à l'aide des modules des erreurs de prédiction et de l'effort de commande sur un horizon glissant :

$$\widetilde{U}=\max_{k_v(t)=\{v_t,\ldots,v_{t+N2}\}} U=\sum_{j=N_1}^{N_2}\left|w_{t+j}-\hat{y}_{t+j}\right|+\sum_{j=1}^{N_u}\lambda_j\left|\Delta u_{t+j-1}\right| \tag{6.21}$$

avec $\hat{y}(t+j)$ la prédiction de la sortie, $N_1, N_2$ les horizons de prédiction sur la sortie, $N_u$ l'horizon de commande, $\lambda_j$ le facteur de pondération de la commande et $w$ la consigne. A noter que la fonction de coût $J$ est l'équivalent de la somme des termes de norme $1/\infty$ pour d'autres formalismes MPC robuste [KCR00], [KM02], [SM98] (cet aspect sera abordé de nouveau lors du formalisme d'état).

La formulation GPC robuste doit considérer la plus mauvaise combinaison des perturbations et résoudre ainsi à chaque période d'échantillonnage un problème d'optimisation 'min-max' :

$$\begin{aligned}\widetilde{U}&=\min_{\mathbf{k}_u(t)=\left\{\Delta u_t,\ldots,\Delta u_{t+N_u}\right\}}\ \max_{\mathbf{k}_v(t)=\left\{v_t,\ldots,v_{t+N_2}-1\right\}}\widetilde{U}=\\&=\min_{\mathbf{k}_u(t)}\max_{\mathbf{k}_v(t)}\sum_{j=N_1}^{N_2}\left|G_j\left(q^{-1}\right)\Delta u_{t+j-1}+\widetilde{J}_j\left(q^{-1}\right)v_{t+j}+F_j\left(q^{-1}\right)y_t+H_j\left(q^{-1}\right)\Delta u_{t-1}-w_{t+j}\right|\\&\qquad+\sum_{j=1}^{N_u}\lambda_j\left|\Delta u_{t+j-1}\right|\end{aligned} \tag{6.22}$$

avec $\mathbf{k}_u=\left\{\Delta u_t,\ldots,\Delta u_{t+N_u-1}\right\}$ et $\mathbf{k}_v=\left\{v_t,\ldots,v_{t+N_2-1}\right\}$ les séquences de commande et de perturbations susceptibles d'intervenir dans le futur. Seul le premier élément de $\mathbf{k}_u^*$ est effectivement appliqué puisque, à la période d'échantillonnage suivante, une stratégie d'horizon glissant est appliquée.

L'optimisation (6.22) doit être résolue par rapport aux divers types de contraintes liées à des considérations opérationnelles ou de performance. Ces contraintes peuvent être représentées de façon similaire au cas nominal par un ensemble d'inégalités linéaires :

$$A_{in}\mathbf{k}_u+C_{in}\mathbf{k}_v\leq b_{in}+B_{in}p \tag{6.23}$$

où $\mathbf{p}_v=\left[\Delta u_{t-1}\ldots\Delta u_{t-n_b}\, y_t\ldots y_{t-n_a}\, w_{t+N_1}\ldots w_{t+N_2}\right]^T$ représente le vecteur des paramètres de contexte qui intervient également dans la fonction de coût (6.22).

**Remarque 6.5** : Les contraintes terminales dans les formulations classiques GPC sont généralement représentées par des contraintes de type égalité à la fin de l'horizon de prédiction. Dans le cas du GPC robuste, il peut être très difficile de satisfaire de telles contraintes car chaque combinaison extrémale des perturbations va annuler un degré de liberté de la commande impliquant de graves problèmes de faisabilité.

Pour remédier à cet inconvénient, une classe spécifique de contraintes terminales est considérée :

$$\hat{y}_{t+N_2} \in Y_f$$

avec $Y_f$ un domaine polyédral (positif invariant par rapport à une loi pré-établie [Bla99]), car une fois vérifiées à l'instant initial, elles renforcent la faisabilité robuste pour tous les instants futurs.

Pour résumer, le problème d'optimisation à résoudre à chaque pas d'échantillonnage est donné par :

$$\min_{\mathbf{k}_u(t)} \max_{\mathbf{k}_v(t)} \sum_{j=N_1}^{N_2} \left| G_j\left(q^{-1}\right)\Delta u_{t+j-1} + \widetilde{J}_j\left(q^{-1}\right)v_{t+j} + S_j p \right| + \sum_{j=1}^{N_u} \lambda_j \left|\Delta u_{t+j-1}\right|$$

$$\begin{cases} A_{in}\mathbf{k}_u + C_{in}\mathbf{k}_v \le b_{in} + B_{in}p \\ \hat{y}_{t+N_2} \in Y_f \\ Mv_{t+k} \le l,\ k = 0,\ldots,N_2 - 1 \end{cases} \tag{6.24}$$

On est donc bien en présence d'un problème d'optimisation de type '*min – max*'.

**Remarque 6.6 :** La fonction de coût pour la loi de commande GPC robuste (6.21) a été construite via une prédiction en boucle ouverte sur tout l'horizon de prédiction $1,\ldots,N_2$. Ce choix peut s'avérer conservatif car la séquence de perturbations dans le pire cas peut dégrader irréversiblement l'évolution du système pour une séquence $\mathbf{k}_u$ fixée.

**Stratégie d'optimisation se basant sur les prédictions en boucle fermée**

L'optimisation sous contraintes (6.24) fournit une séquence de commande robuste conservatrice puisque que l'on considère toutes les réalisations de perturbation en ignorant que des mesures sont disponibles au cours du temps. La loi de commande peut être améliorée si, à chaque étape de la prédiction, l'optimisation ne prend en compte que les seules combinaisons des perturbations à venir. Une telle approche, appelée 'prédiction en boucle fermée' a comme résultat une formulation de la loi prédictive en termes d'une optimisation 'min-max' imbriquée :

$$\min_{\Delta u_t}\left\{\lambda_0\left|\Delta u_t\right| + \max_{v_t}\left\{\left|\hat{y}_{t+k} - w_{t+k}\right| + \min_{\Delta u_{t+1}}\left\{\ldots + \min_{\Delta u_{t+N_u-1}}\left\{\lambda_{N_u-1}\left|\Delta u_{t+N_u-1}\right| + \max_{v_{t+N_u-1},\ldots,v_{t+N_2-1}} \sum_{k=N_u}^{N_2}\left|\hat{y}_{t+k} - w_{t+k}\right|\right\}\right\}\ldots\right\}\right\}$$

$$\begin{cases} A_{in}\mathbf{k}_u + C_{in}\mathbf{k}_v \le b_{in} + B_{in}p \\ \hat{y}_{t+N_2} \in Y_f \\ Mv_{t+k} \le l,\ k = 0,\ldots,N_2 - 1 \end{cases} \tag{6.25}$$

Cette formulation 'en boucle fermée' évite les problèmes de faisabilité, comme souligné dans [Ker04]. Les bénéfices comparés à ceux de la formulation en boucle ouverte s'expliquent par l'apport d'information lors de la résolution du problème de minimisation. Pour le problème (6.24), l'information avec laquelle opère la minimisation consiste en le vecteur de variables de contexte $\mathbf{p}$ et la séquence de commande doit garantir la satisfaction des contraintes pour toutes les réalisations possibles de perturbations. Pour la formulation (6.25), on observe que la $j$-ème minimisation s'opère en possession des paramètres $\{\mathbf{p}, \Delta u_t,\ldots,\Delta u_{t+j-1}, v_t,\ldots,v_{t+j-1}\}$, donc en supposant que les mesures jusqu'à l'instant $j$ sont disponibles. Ainsi les commandes sont traitées comme des degrés de liberté lors de la synthèse au lieu de composantes d'une séquence qui peut être assez conservative :

$$\widetilde{J}(\mathbf{p}, \Delta u_t, \ldots, \Delta u_{t+j-1}, v_t, \ldots, v_{t+j-1}) = \min_{\Delta u_{t+j}} \left\{ \lambda_j \left|\Delta u_t\right| + \max_{v_{t+j}} \left\{ \left|\hat{y}_{t+j+1} - w_{t+j+1}\right| + \ldots \right.\right.$$

$$\left. + \min_{\Delta u_{t+N_u-1}} \left\{ \lambda_{N_u-1} \left|\Delta u_{t+N_u-1}\right| + \max_{v_{t+N_u-1}, \ldots, v_{t+N_2-1}} \sum_{k=N_u}^{N_2} \left|\hat{y}_{t+k} - w_{t+k}\right| \right\} \right. \tag{6.26}$$

$$\begin{cases} A_{in}\mathbf{k}_u + C_{in}\mathbf{k}_v \le b_{in} + B_{in}\mathbf{p} \\ \hat{y}_{t+N_2} \in Y_f \\ Mv_{t+k} \le l,\ k = 0, \ldots, N_2 - 1 \end{cases}$$

**Remarque 6.7 :** Les formulations (6.24) et (6.25) ont été décrites dans une version générale offrant la liberté de choisir l'horizon de prédiction différent de l'horizon de commande. Pour garantir la faisabilité en présence de perturbation, on doit disposer de plus de degrés de liberté et donc généralement l'horizon de commande est égal à l'horizon de prédiction.

### 6.3.2 Cas général avec représentation d'état

Une mise en forme semblable peut être développée pour les lois prédictives basées sur la représentation d'état. Le but est de retrouver le problème d'optimisation qui garantisse la satisfaction des contraintes pour la combinaison des perturbations la plus défavorable.

**Stratégie d'optimisation en boucle ouverte**

Considérons le problème de commande prédictive formulé pour un système linéaire discret invariant dans le temps affecté par une perturbation d'entrée :

$$x_{t+1} = Ax_t + Bu_t + Ev_t \tag{6.27}$$

et sujet à un ensemble de contraintes linéaires :

$$Cx_t + Du_t \le d \tag{6.28}$$

Les vecteurs $x_t \in \Re^n$ et $u_t \in \Re^m$ représentent les états et les entrées, tandis que $v_t \in \Re^p$ constitue le vecteur inconnu des perturbations se trouvant à l'intérieur d'un polytope contenant l'origine, défini par un ensemble de contraintes linéaires :

$$V = \left\{ v \middle| Mv \le l; l \ge 0 \right\} \tag{6.29}$$

Dans la suite, la paire ($A$, $B$) est supposée stabilisable, de même on suppose que la mesure complète de l'état courant est disponible à chaque instant $t$. La formulation du problème de commande prédictive implique la construction du critère de coût qui sera optimisé à chaque pas d'échantillonnage. A la différence du cas nominal pour lequel des fonctions de coût quadratiques sont utilisées, dans le cas des modèles affectés par des perturbations, une optimisation 'min-max' est préférable, comme précédemment décrit, soit :

$$\begin{aligned} &\min_{u_t, \ldots, u_{t+N_u-1}} \left\{ \max_{v_t, \ldots, v_{t+N-1}} \left\{ S_{P_\lambda}(x_{t+N|t}) + \sum_{k=1}^{N-1} \left\| Qx_{t+k|t} \right\|_\infty + \sum_{k=0}^{N_u-1} \left\| Ru_{t+k} \right\|_\infty \right\} \right\} \\ &\text{sujet à} \quad Cx_{t+k|t} + Du_{t+k} \le d, k = 1, \ldots, N \\ &\qquad\qquad Mv_{t+k} \le l, k = 0, \ldots N-1 \\ &\qquad\qquad x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k} + Ev_{t+k}, k \ge 0 \\ &\qquad\qquad x_{t+N|t} \in P_\lambda \end{aligned} \tag{6.30}$$

avec $Q$, $R$ des matrices de pondération, $\left\|*\right\|_\infty \equiv \max_{i=1,\cdots,\dim}\left(*^i\right)$, où $*^i$ est la composante $i$ du vecteur $* \in \Re^{\dim}$.

Les prédictions de l'état $x_{t+k|t}$ sont obtenues à partir du vecteur d'état courant $x_t$ en appliquant la séquence de commande $\mathbf{k}_u = \left\{ u_t, \ldots, u_{t+N_u-1} \right\}$ au modèle (6.27) sur un certain horizon de commande $N_u$. Dans le cas général, l'horizon de commande ($N_u$) et l'horizon de prédiction ($N_2$) peuvent être différents et le vecteur de commande a une formulation fixe pour $N_u \le k \le N_2$, comme dans le chapitre 2 pour le cas nominal, mais ces contraintes seront omises ici par souci de simplicité. La séquence de perturbations $\mathbf{k}_v = \{v_t, \ldots, v_{t+N-1}\}$ va quant à elle influencer la prédiction sur tout l'horizon de prédiction. La stabilité du problème MPC dépend des horizons choisis et du coût terminal. Afin de garantir la stabilité, un horizon infini de prédiction devrait être employé. Un tel choix transforme cependant (6.30) en un problème difficile à résoudre. La solution consiste alors à choisir un horizon de prédiction fini et à considérer qu'à la fin de l'horizon de prédiction la trajectoire du système est amenée à l'intérieur d'un ensemble positif invariant, $P_\lambda$, pouvant être calculé hors-ligne [Ker00]. Une fonction $S_{P_\lambda}(x)$ peut être associée à cette région terminale, apparaissant dans (6.30) en tant que coût terminal pénalisant l'évolution du système commandé de $N$ à $\infty$.

En appliquant la stratégie d'horizon glissant, l'optimisation (6.30) est résolue à chaque pas d'échantillonnage $t$ en utilisant le vecteur des états mesurés $x_t$ (jouant le rôle du paramètre pour l'optimisation). Si $\mathbf{k}_u^* = \{u_t^*, \ldots, u_{t+N_u-1}^*\}$ est la solution de (6.30), l'entrée appliquée au système (6.27) est la première valeur de la séquence et le procédé est réitéré selon le principe de l'horizon glissant.

Une attention spéciale doit être accordée au choix de l'horizon de commande. En effet, ce paramètre est sensible car il reflète le nombre de degrés de liberté disponibles pour assurer la satisfaction des contraintes pour toutes les combinaisons possibles des perturbations. D'autre part, avec moins d'arguments, le volume de calcul lié au problème d'optimisation est diminué. Dans le cas de lois prédictives robustes, l'horizon de commande est généralement égal à l'horizon de prédiction $N_u = N_2$, car l'effet cumulatif des perturbations impose en contrepartie des degrés de liberté suffisants pour pouvoir contrecarrer l'effet d'infaisabilité. Néanmoins, la formulation (6.30), dénommée 'problème basé sur une philosophie en boucle ouverte', peut être très conservative, surtout si les combinaisons des perturbations intervenant sont loin du pire cas imaginé lors de la synthèse.

**Remarque 6.8 :** Pratiquement, le problème d'optimisation (6.30) doit être résolu dans l'ordre suivant :

$$\begin{aligned}
&\mathbf{k}_u^* = \min_{\mathbf{k}_u=\{u_t,\ldots,u_{t+N_u-1}\}} \widetilde{J}(u_t, \ldots, u_{t+N_u-1}, x_t) \\
&\text{sous les contraintes : } Cx_{t+k} + Du_{t+k} \le d, k = 1, \ldots, N \\
&\qquad x_{t+k+1} = Ax_{t+k} + B_{t+k} + Ev_{t+k}; Mv_{t+k} \le l; k \ge 0 \\
&\qquad x_{t+N} \in P_\lambda
\end{aligned} \tag{6.31}$$

avec :

$$\begin{aligned}
&\widetilde{J}(u_t, \ldots, u_{t+N_u-1}, x_t) = \max_{\mathbf{k}_v=\{v_t,\ldots,v_{t+N_u-1}\}} \widetilde{J}(u_t, \ldots, u_{t+N_u-1}, v_t, \ldots, v_{t+N_u-1}, x_t) \\
&Mv_{t+k} \le l; k \ge 0
\end{aligned} \tag{6.32}$$

Ne faire intervenir les contraintes que lors du problème de minimisation est motivé par le fait que la perturbation $v_{t+k}$ se manifeste indépendamment de la valeur de l'état du système. Les contraintes liées aux états sont donc évacuées du problème de maximisation qui définit uniquement le critère à utiliser lors de la construction de la séquence de commande optimale. Indirectement, la séquence de commande garantit la préservation de l'état dans la région faisable pour toute valeur possible des perturbations.

**Stratégie d'optimisation en boucle fermée**

Le besoin de considérer une certaine dynamique d'évolution impliquant les perturbations et les entrées lors de l'élaboration de la commande peut être résolu en n'autorisant qu'un seul mouvement à la fois à chacun des protagonistes. Ceci signifie que, au lieu d'un problème d'optimisation 'min-max', le problème MPC doit faire intervenir une formulation 'min-max imbriquée', en remplaçant (6.30) par :

$$\begin{aligned}
&\min_{u_t}\left\{\max_{v_t}\left\{\min_{u_{t+1}}\dots\min_{u_{t+N_u-1}}\left\{\max_{v_{t+N_u},\dots,v_{t+N-1}}\left\{S_{P_\lambda}(x_{t+N|t})+\sum_{k=1}^{N-1}\left\|Qx_{t+k|t}\right\|_\infty+\sum_{k=0}^{N_u-1}\left\|Ru_{t+k}\right\|_\infty\right\}\right\}\dots\right\}\right\}\\
&\text{sujet à}\quad Cx_{t+k|t}+Du_{t+k}\le d, k=1,\dots,N\\
&\qquad Mv_{t+k}\le l, k=0,\dots N-1\\
&\qquad x_{t+k+1|t}=Ax_{t+k|t}+Bu_{t+k}+Ev_{t+k}, k\ge 0\\
&\qquad x_{t+N|t}\in P_\lambda
\end{aligned} \tag{6.33}$$

ou d'une manière équivalente :

$$\begin{aligned}
&\min_{u_t}\left\{\left\|Ru_t\right\|_\infty+\max_{v_t}\left\{\left\|Qx_{t+1|t}\right\|_\infty+\min_{u_{t+1}}\left\{\dots+\min_{u_{t+N_u-1}}\left\{\left\|Ru_{t+N_u-1}\right\|_\infty+\max_{v_{t+N_u-1},\dots,v_{t+N}}\left\{S_{P_\lambda}(x_{t+N|t})+\sum_{k=N_u}^{N-1}\left\|Qx_{t+k|t}\right\|_\infty\right\}\right\}\dots\right\}\right\}\right\}\\
&\text{sujet à}\quad Cx_{t+k|t}+Du_{t+k}\le d, k=1,\dots,N\\
&\qquad Mv_{t+k}\le l, k=0,\dots N-1\\
&\qquad x_{t+k+1|t}=Ax_{t+k|t}+Bu_{t+k}+Ev_{t+k}, k\ge 0\\
&\qquad x_{t+N|t}\in P_\lambda
\end{aligned} \tag{6.34}$$

Cette formulation est appelée dans la littérature 'problème basé sur une philosophie en boucle fermée', soulignant ainsi le fait que la séquence optimale est conçue en se basant sur le principe de rétroaction qui permet de minimiser les effets néfastes de la perturbation à l'étape où ils sont produits. Ce principe est en relation directe avec la programmation dynamique, utilisée lors de la résolution des problèmes de type (6.34).

> **Remarque 6.9 :** Seules des perturbations additives ont été prises en compte lors des développements précédents, mais la formulation peut être étendue aux incertitudes paramétriques [BBM01] pour les deux types de prédiction, en boucle ouverte et en boucle fermée.

### 6.3.3 Des problèmes de type *min-max* vers la programmation linéaire multiparamétrique

Avant de considérer le problème de commande prédictive robuste, il convient de récapituler certains détails sur la structure d'un problème d'optimisation de type' min – max' :

$$\begin{aligned}
&J^*(x_t)=\min_{\mathbf{k_u}}\max_{\mathbf{k_v}} J(x_t,\mathbf{k_u},\mathbf{k_v})\\
&\text{suj. à } A_{in}\mathbf{k_u}+C_{in}\mathbf{k_v}\le b_{in}+B_{in}x_t
\end{aligned} \tag{6.35}$$

avec $\mathbf{k_u}=\left\{u_t,\dots,u_{t+N_u-1}\right\}$, $\mathbf{k_v}=\left\{v_t,\dots,v_{t+N-1}\right\}$ et une fonction de coût $J(x_t,\mathbf{k_u},\mathbf{k_v})$ convexe basée sur une somme de termes de norme infinie :

$$J(x_t,\mathbf{k}_u(t),\mathbf{k}_v(t))=\sum_{i=1}^{n_T}\left\|T_i^x x_t+T_i^u\mathbf{k}_u(t)+T_i^v\mathbf{k}_v(t)+T_i^0\right\|_\infty \tag{6.36}$$

avec $T_i^x, T_i^u, T_i^v$ les matrices liées aux pondérations entre l'effort de commande et l'erreur de prédiction et $T_i^0$ un terme linéaire apparaissant surtout lors d'applications en suivi de consigne.

Les matrices $A_{in}, B_{in}, b_{in}, C_{in}$ traduisent sous une forme compacte l'ensemble des contraintes linéaires présentes lors de la construction des lois prédictives. La fonction de coût et l'ensemble des contraintes dépendent du vecteur d'état courant $x_t$ qui joue le rôle d'un paramètre. On retrouve donc un problème d'optimisation multiparamétrique encore plus gourmand en temps de calcul que la version quadratique, principalement à cause de la structure particulière de la fonction de coût. Ce type de critère préserve tout de même les caractéristiques de convexité et, fait intéressant du point de vue implémentation, il est équivalent à un problème de programmation linéaire.

**Redéfinition de l'optimisation interne**

L'influence des perturbations sous la forme (6.35) peut être examinée en reconsidérant toutes les combinaisons extrémales formant le domaine polyédral $V$ [38]. Ces combinaisons peuvent être décrites par l'intermédiaire des sommets de $V$. Si les combinaisons de sommets sont énumérées pour chaque étape de prédiction, on peut caractériser ainsi la séquence $\mathbf{k}_v$ :

$$v_t \in V \subset \Re^p \Rightarrow \mathbf{k}_v \in V^N \subset \Re^{N\,p} \tag{6.37}$$

> **Remarque 6.10 :** Pour l'optimisation interne, les arguments sont les éléments de $\mathbf{k}_v$. Les variables $x_t, \mathbf{k}_u$ sont considérées comme des paramètres :
>
> $$\begin{aligned} &\max_{\mathbf{k}_v} J(x_t, \mathbf{k}_u(t), \mathbf{k}_v(t)) \\ &\mathbf{k}_v \in V^N \end{aligned} \tag{6.38}$$

L'ensemble des contraintes est constitué uniquement d'inégalités définissant le domaine polyédral de perturbations comme dans (6.29), et de contraintes imposées par la dynamique du système (6.27). Ceci apparaît clairement à partir de la définition de la loi de commande prédictive, qui autorise n'importe quelle combinaison des perturbations satisfaisant (6.29), indifféremment des autres contraintes sur les états ou grandeurs de commande. Si l'une de ces combinaisons de perturbations n'est pas permise par l'ensemble de contraintes de (6.35), cela signifie en fait que la loi MPC est infaisable.

Tenant compte de la convexité de la fonction objectif et de la remarque précédente, on peut conclure que l'optimum pour l'optimisation interne dans (6.35) se trouve sur la frontière du domaine faisable, plus précisément sur un des sommets de $V^N$ lorsqu'il est défini en tant que polytope. En conséquence (6.35) devient :

$$\begin{aligned} &\min_{\mathbf{k}_u} \max_{\mathbf{v}_{\mathbf{k}_v}} J(x_t, \mathbf{k}_u, \mathbf{k}_{v_l}) \\ &\text{suj. à } \; A_{in}\mathbf{k}_\mathbf{u} + C_{in}\mathbf{k}_{\mathbf{v}_l} \le b_{in} + B_{in}x_t \\ &\qquad\quad l \in L, \mathbf{k}_{v_l} \in V^N \end{aligned} \tag{6.39}$$

avec $L = \{1, 2, \ldots, N_v\}$ et $N_v$ le nombre de sommets dans $V^N$, $\mathbf{k}_{v_l}$ les sommets effectifs.

Ceci signifie que l'optimisation interne dans (6.35) n'agit que sur l'ensemble des sommets de $V^N$. On peut donc reformuler cette optimisation par :

$$\begin{aligned} &\min_{\mathbf{k}_u, \mu} \mu \\ &\text{suj. à } \; A_{in}\mathbf{k}_u + C_{in}\mathbf{k}_{v_l} \le b_{in} + B_{in}x_t \\ &\qquad\quad J(x_t, \mathbf{k}_u, \mathbf{k}_{v_l}) \le \mu \\ &\qquad\quad l \in L, \mathbf{k}_{v_l} \in V^N \end{aligned} \tag{6.40}$$

**Redéfinition du problème d'optimisation externe**

Le changement effectué sur le problème de maximisation se trouvant à l'intérieur du critère du problème d'optimisation ne fait que translater le problème de fonction de coût vers le système de contraintes, qui devient non linéaire. Ceci constitue l'obstacle principal à l'implémentation de la formulation (6.35) et par suite à l'analyse du domaine faisable ou à la construction éventuelle de la solution explicite. Mais le problème n'est pas insurmontable puisque la structure de la fonction de coût, donnée en tant qu'ensemble de termes de norme infinie peut être exploitée pour transformer les contraintes en un ensemble linéaire.

Afin d'éviter la difficulté inhérente à la manipulation (6.38), une formulation équivalente de programmation linéaire (LP) [KM02] peut être réalisée basée sur l'idée que chaque terme de norme infinie peut être borné. Le

[38] L'idée a été introduite initialement dans [CM86].

problème d'optimisation est alors équivalent à la minimisation de la somme de ces limites. Ceci est résumé par le résultat suivant où la fonction de coût est considérée en tant que somme de termes de norme infinie linéaires en le vecteur des inconnus $\mathbf{k}_u$ et des paramètres $x$ (afin de les identifier dans le problème (6.40), on peut observer que pour une séquence fixe $\mathbf{k}_{v_l} = ct$, la fonction de coût $J(x_t, \mathbf{k}_u, \mathbf{k}_{v_l})$ est une somme de termes $\left\| T_i^x x + T_i^u \mathbf{k}_u + T_i^v \mathbf{k}_{v_l} + T_i^0 \right\|_\infty = \left\| T_i^x x + T_i^u \mathbf{k}_u + \widetilde{T}_i^0 \right\|_\infty$ avec $\widetilde{T}_i^0$ défini en fonction des valeurs extrémales considérées pour $\mathbf{k}_v$).

**Proposition 6.11 :** Les formulations (A) et (B) sont équivalentes (voir démonstration dans [Ker04]) :

$$(A)\begin{cases} J(x) = \min\limits_{\mu, \mathbf{k}_u} \mu \\ \sum\limits_{i=1}^{n_T} \left\| T_i^x x + T_i^u \mathbf{k}_u(t) + \widetilde{T}_i^0 \right\|_\infty \le \mu \end{cases} \tag{6.41}$$

$$(B)\begin{cases} J(x) = \min\limits_{\mu, \mathbf{k}_u} \mu \\ -\mathbf{1}\sigma_i \le T_i^x x + T_i^u \mathbf{k}_u(t) + \widetilde{T}_i^0 \le \mathbf{1}\sigma_i ; 1 \le i \le n_T \\ \sum\limits_{i=1}^{n_T} \sigma_i \le \mu \end{cases} \tag{6.42}$$

où $\sigma_i, \rho \in \Re$ et $\mathbf{1}$ est un vecteur unité de longueur appropriée.

**Formulation comme problème de programmation linéaire**

Avec les deux transformations précédentes, l'optimisation (6.35) peut être réécrite de la manière suivante :

$$\begin{aligned} &\mathbf{k_u}^*(x_t) = \min_{\mu, \mathbf{k_u}, \{\sigma_i^j\}} \mu \\ &\text{sujet à } \begin{cases} -T_i^x x_t - T_i^v \mathbf{k}_{v_l} - T_i^0 \le T_i^u \mathbf{k_u} + \mathbf{1}\sigma_i^j \le -T_i^x x_t - T_i^v \mathbf{k}_{v_l} - T_i^0, 1 \le i \le n, 1 \le l \le N_v \\ \begin{bmatrix} \sum\limits_{i=1}^{n} \sigma_i^1 \\ \vdots \\ \sum\limits_{i=1}^{n} \sigma_i^{N_v} \end{bmatrix} - \mathbf{1}\mu \le 0 \\ A_{in} \mathbf{k_u} \le b_{in} + B_{in} x_t - C_{in} \mathbf{k}_{\mathbf{v}_l}, 1 \le l \le N_v \end{cases} \end{aligned} \tag{6.43}$$

**Exemple 6.12 :** Pour illustrer les deux transformations précédentes, on considère le problème d'optimisation indépendant des paramètres (Figure 6.7) :

$$\begin{aligned} &J = \min_{x_1} \max_{x_2} \left\| \begin{matrix} 2x_1 + x_2 - 3 \\ x_1 - x_2 + 1 \end{matrix} \right\|_\infty + \left\| \begin{matrix} x_1 - 2x_2 + 1 \\ 2x_1 + 3x_2 - 7 \end{matrix} \right\|_\infty \\ &\text{sujet à } \begin{cases} x_2 \in [-1, 1] \\ x_1 \in [0, 6] \end{cases} \end{aligned} \tag{6.44}$$

soit selon les transformations précédentes, équivalent à :

$$\begin{aligned} &J = \min_{x_1, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \mu} \mu \\ &-\begin{bmatrix} \sigma_1 \\ \sigma_1 \end{bmatrix} \le \begin{bmatrix} 2x_1 - 2 \\ x_1 \end{bmatrix} \le \begin{bmatrix} \sigma_1 \\ \sigma_1 \end{bmatrix}; \quad -\begin{bmatrix} \sigma_2 \\ \sigma_2 \end{bmatrix} \le \begin{bmatrix} 2x_1 - 4 \\ x_1 + 2 \end{bmatrix} \le \begin{bmatrix} \sigma_2 \\ \sigma_2 \end{bmatrix}; \\ &-\begin{bmatrix} \sigma_3 \\ \sigma_3 \end{bmatrix} \le \begin{bmatrix} x_1 - 1 \\ 2x_1 - 4 \end{bmatrix} \le \begin{bmatrix} \sigma_3 \\ \sigma_3 \end{bmatrix}; \quad -\begin{bmatrix} \sigma_4 \\ \sigma_4 \end{bmatrix} \le \begin{bmatrix} x_1 + 3 \\ 2x_1 - 10 \end{bmatrix} \le \begin{bmatrix} \sigma_4 \\ \sigma_4 \end{bmatrix}; \\ &\sigma_1 + \sigma_3 \le \mu; \\ &\sigma_2 + \sigma_4 \le \mu; \\ &x_1 \in [0, 6] \end{aligned} \tag{6.45}$$

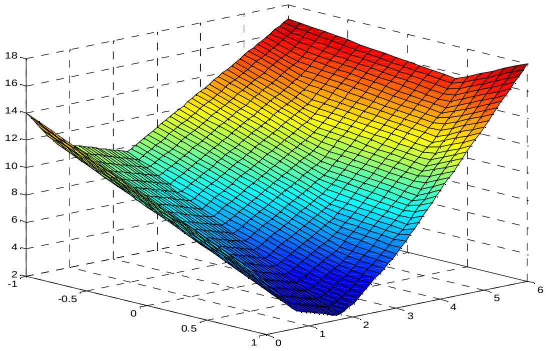



**Figure 6.7 : Fonction de coût de l'exemple 6.12**

ce qui peut être résolu par n'importe quel solveur LP, avec la solution :

$$[x_1\ \sigma_1\ \sigma_2\ \sigma_3\ \sigma_4\ \rho]^* = [2.33\ 4.33\ 5.33\ 1.33\ 2.66\ 9.66] \tag{6.46}$$

Donnant la solution $J = 9{,}66$.

## *6.4 Utilisation des polyèdres paramétrés pour l'obtention de lois explicites*

Du point de vue charge de calcul, la complexité des problèmes d'optimisation dans le cas robuste restreint le domaine d'application à des systèmes très lents, l'élargissement du domaine d'application constitue donc un véritable enjeu pour la commande prédictive. Dans cette direction, des solutions explicites peuvent être trouvées de façon similaire au cas nominal. Les solutions seront toujours construites en explorant l'espace des paramètres.

Le paragraphe précédent a détaillé la formulation de lois de commande prédictives robustes en terme de programmation linéaire. Le but des développements à venir est de proposer une approche originale basée sur les outils géométriques pour l'analyse des domaines faisables afin de déterminer la forme explicite de cette loi de commande non linéaire. En effet, l'expression en fonction des paramètres de contexte peut s'avérer une alternative intéressante à la résolution en ligne de problèmes d'optimisation de type 'min-max'. Pour réaliser cette approche, une procédure de construction des solutions explicites pour des problèmes d'optimisation linéaires multiparamétriques (mpLP) doit être élaborée. La littérature relative à la solution de problèmes mpLP mentionne les travaux de Gal et Nedoma [GN72] et les développements ultérieurs de Bemporad et ses coauteurs, [BBM01]. Ces derniers implémentent des algorithmes essentiellement liés aux conditions algébriques d'optimalité.

La procédure proposée se focalise dans ce qui suit essentiellement sur l'ensemble de contraintes et la correspondance géométrique du domaine faisable associé [OD04d]. Ce domaine faisable s'exprime sous la forme d'un polyèdre paramétré, les sommets paramétrés étant un des aspects fondamentaux de cette structure. Comme la formulation du problème d'optimisation n'est pas modifiée, l'introduction de variables auxiliaires binaires est évitée. Ainsi l'algorithme résultant diffère des solutions basées sur des méthodes de type '*branch and bound*' ou faisant appel à des solveurs de type '*mixed integer linear'*, il se concentre principalement sur l'énumération des faces d'un polyèdre de dimension augmentée.

**Polyèdre paramétré**

Reprenons les définitions du chapitre 2 pour caractériser le domaine faisable de (6.43) par les contraintes :

$$P = \left\{ \mathbf{x} \in \Re^n \middle| \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} ; \mathbf{A}_{in}\mathbf{x} \le \mathbf{b}_{in} \right\} \tag{6.47}$$

ou par sa représentation duale de Minkowski de ses générateurs :

$$\begin{gathered} P = \left\{ \mathbf{x} \in \Re^n \middle| \mathbf{x} = \sum_{i=1}^{v} \lambda_i \mathbf{x}_i + \sum_{i=1}^{r} \gamma_i \mathbf{y}_i + \sum_{i=1}^{l} \mu_i \mathbf{z}_i \right\} \\ 0 \le \lambda_i \le 1,\ \sum_{i=1}^{v} \lambda_i = 1 , \gamma_i \ge 0\ , \forall \mu_i \end{gathered} \tag{6.48}$$

Pour les problèmes de type (6.43), le domaine faisable est décrit en fait par des polyèdres paramétrés puisque les inégalités et les égalités possèdent une partie affine dépendant linéairement d'un vecteur de paramètres. Cet aspect apparaît clairement en réécrivant simplement le problème (6.43) sous la forme :

$$\begin{aligned} &\chi^*(\mathbf{p}) = \min_{\chi} f^T \chi \\ &A_{in}\chi \leq b_{in} + B_{in}\mathbf{p} \end{aligned} \tag{6.49}$$

où le vecteur d'arguments du problème d'optimisation $\chi = \left[\mu, \left\{\sigma_i^j\right\}, \mathbf{k_u}\right]$ contient également la commande optimale $\mathbf{k}_u^*(\mathbf{p})$ nécessaire pour l'application selon la philosophie prédictive. Le paramètre de contexte $\mathbf{p}$ regroupe toutes les variables connues au moment de la résolution du problème (fonction de la philosophie choisie pour la prédiction – *boucle ouverte/ boucle fermée*).

Pour le problème (6.49), le domaine faisable est décrit par une double représentation, en différenciant éventuellement les contraintes égalité des inégalités sous la forme :

$$\begin{aligned} P(\mathbf{p}) &= \left\{\chi \in \Re^n \middle| A_{eq}\chi = B_{eq}\mathbf{p} + b_{eq};\, A_{in}\chi \leq B_{in}\mathbf{p} + b_{in}\right\} \\ &= \left\{\chi(\mathbf{p}) \middle| \chi(\mathbf{p}) = \sum_{i=1}^{n_s} \sigma_i(\mathbf{p}) s_i(\mathbf{p}) + \sum_{i=1}^{n_r} \rho_i r_i + \sum_{i=1}^{n_l} \lambda_i l_i \right\} \\ &0 \leq \sigma_i(\mathbf{p}) \leq 1,\ \sum_{i=1}^{n_s} \sigma_i(\mathbf{p}) = 1,\ \rho_i \geq 0\ ,\forall \lambda_i \end{aligned} \tag{6.50}$$

### 6.4.1 Solution du problème de programmation linéaire non paramétré

Intéressons-nous tout d'abord au problème non paramétré :

$$\begin{aligned} &\chi^* = \arg\min_{\chi} f^T \chi \\ &A_{in}\chi \leq b_{in} \end{aligned} \tag{6.51}$$

Le but est d'examiner la solution d'un point de vue géométrique en se basant sur l'hypothèse qu'une double représentation du domaine faisable existe. Dans ce cas, la solution se caractérise à partir de la représentation des générateurs, de la même manière que dans [Che64]. Le résultat est résumé par la proposition suivante.

**Proposition 6.13 :** Pour un problème de programmation linéaire (6.51), trois cas peuvent survenir :

a) Si le polyèdre associé $P = \{\chi | A_{in}\chi \leq b_{in}\}$ est vide (il n'existe aucune sommet associé), le problème est infaisable

b) S'il existe un rayon bidirectionnel $l_i$ tel que $f^T l_i \neq 0$ ou s'il existe un rayon $r_i$ tel que $f^T r_i < 0$, alors le minimum est non borné

c) Si tous les rayons bidirectionnels $l_i$ sont tels que $f^T l_i = 0$ et tous les rayons $r_i$ sont tels que $f^T r_i \geq 0$, alors le minimum du critère pour le problème d'optimisation est défini par :

$$F^* = \min\left\{f^T \sigma_i | \sigma_i \text{ - sommet de domaine faisable } P\right\}$$

et la caractérisation globale de la solution de (6.51) est :

$$\chi \in X = conv.hull\left\{\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{\tilde{n}_s}\right\} + cone\left\{\tilde{r}_1, \ldots, \tilde{r}_{\tilde{n}_r}\right\} + lin.space P$$

où $\tilde{\sigma}_i$ sont les sommets atteignant le minimum et $\tilde{r}_i$ sont tels que $f^T r_i = 0$. ■

**Remarque 6.14 :** La solution du problème (6.51) n'est pas unique. En effet, on peut trouver sur la direction identifiée par le vecteur $f$ un sommet déterminant un minimum, mais son unicité n'est pas assurée. De plus, si un deuxième sommet induit la même valeur pour la fonction de coût, le nombre des minimums est en fait infini car toutes les combinaisons convexes entre les deux sommets sont solutions.

**Remarque 6.15 :** Il est intéressant d'observer que la solution du problème de programmation linéaire est fortement liée aux générateurs du domaine faisable. Par leur simple rapport à la fonction de coût, on obtient une solution complète, y compris les situations d'infaisabilité.

### 6.4.2 Solution du problème de programmation linéaire multiparamétrique

Par extension de ce résultat au cas multiparamétrique (avec $\chi \in \Re^n$), un résultat similaire peut être alors établi pour la caractérisation de la solution multiparamétrique.

**Proposition 6.16 :** La solution d'un problème d'optimisation linéaire multiparamétrique est caractérisée par :

a) S'il existe un rayon bidirectionnel $l_i$ tel que $f^T l_i \neq 0$ ou s'il existe un rayon $r_i$ tel que $f^T r_i < 0$, alors le minimum est illimité

b) Pour les sous-domaines de l'espace des paramètres $D_{if} \in \Re^n$ de polyèdre associé $P(\mathbf{p})=\{\chi | A_{in}\chi \leq B_{in}\mathbf{p}+b_{in}\}$ vide pour $\mathbf{p} \in D_{if}$, le problème est infaisable. Ce phénomène peut être reformulé en terme de sommets paramétrés : 'pour les sous-domaines $D_{if} \in \Re^n$ où aucun sommet paramétré n'est disponible, le problème est infaisable'

c) Si tous les rayons bidirectionnels $l_i$ sont tels que $f^T l_i = 0$ et tous les rayons $r_i$ sont tels que $f^T r_i \geq 0$, alors il existe un découpage de l'espace des paramètres $D_1 \cup \ldots \cup D_{n_D} = \Re^n - D_{if}$ tel que pour chaque région $D_i$, le minimum du critère du problème d'optimisation (6.49) soit défini par :

$$F_{D_i}^* = \min\left\{ f^T \sigma_i(\mathbf{p}) | \mathbf{p} \in D_i;\ \sigma_i(\mathbf{p}) \text{ - sommet paramétré de domaine faisable } P(\mathbf{p}) \right\}$$

La solution complète pour ce sous-domaine est :

$$\chi(\mathbf{p}) \in \mathrm{X}_{D_i}(\mathbf{p}) = conv.hull\left\{ \tilde{\sigma}_1^{D_i}(\mathbf{p}), \ldots, \tilde{\sigma}_{\tilde{n}_{s_i}}^{D_i}(\mathbf{p}) \right\} + cone\left\{ \tilde{r}_1, \ldots, \tilde{r}_{\tilde{n}_r} \right\} + lin.space P$$

où $\tilde{\sigma}_1^{D_i}(\mathbf{p})$ sont les sommets paramétrés correspondant au minimum $F_{D_i}^*$ et $\tilde{r}_i$ sont tels que $f^T r_i = 0$. ■

**Remarque 6.17 :** Les deux premières catégories (a – b) de solutions sont en fait identiques au cas non paramétré car elles sont basées sur des relations faisant intervenir les rayons uni/bidirectionnels, lesquels demeurent inchangés suite à la paramétrisation.

Remarquons cependant que l'objectif est ici de trouver la solution explicite du problème de programmation linéaire dérivé du problème d'optimisation MPC robuste possédant quelques particularités :

- l'espace de linéarité est vide puisque la fonction de coût initiale de la formulation 'min-max' est convexe, positive et bornée.
- il n'existe aucun rayon unidirectionnel tel que $f^T r_i < 0$ parce que ceci impliquerait que la fonction de coût n'est pas convexe.
- pour chaque $\mathbf{p}$ une seule valeur dans $\mathrm{X}_{D_i}(\mathbf{p})$ est utilisée en ligne par l'algorithme de commande prédictive.

**Proposition 6.18 :** La solution d'un problème d'optimisation linéaire multiparamétrique dans le cadre de la commande prédictive robuste satisfait :

a) Le problème est infaisable pour les sous-domaines $D_{if} \in \Re^n$ où aucun sommet paramétré n'est disponible

b) un découpage de l'espace des paramètres $D_1 \cup \ldots \cup D_{n_D} = \Re^n - D_{if}$ tel que pour chaque région $D_i$ la solution optimale $\chi^*(\mathbf{p})$ sera contenue dans l'ensemble :

$$\chi^*(\mathbf{p}) \in \mathrm{X}_{D_i}(\mathbf{p}) = conv.hull\left\{ \tilde{\sigma}_1^{D_i}(\mathbf{p}), \ldots, \tilde{\sigma}_{\tilde{n}_s}^{D_i}(\mathbf{p}) \right\}$$

où les sommets paramétrés $\tilde{\sigma}_j^{D_i}$ satisfont :

$$f^T \tilde{\sigma}_1^{D_i}(\mathbf{p}) = \ldots = f^T \tilde{\sigma}_{\tilde{n}_{s_i}}^{D_i}(\mathbf{p}) = \min\left\{ f^T \sigma(\mathbf{p}) | \mathbf{p} \in D_i;\ \sigma(\mathbf{p}) \text{ - sommet paramétré de domaine faisable } P(\mathbf{p}) \right\}$$

**Remarque 6.19 :** Lorsque que les paramètres dans (6.50) varient à l'intérieur de l'espace des paramètres, les sommets du domaine d'optimisation peuvent se dédoubler (split), se décaler ou fusionner. L'optimum global explicite doit alors suivre cette évolution de façon à engendrer une fonction continue du paramètre, ce qui implique l'existence de méthodes permettant de déterminer une valeur unique de la solution explicite pour les valeurs de paramètres pour lesquels la solution du problème d'optimisation (6.49) n'est pas unique (des techniques en ce sens peuvent être trouvées dans [GAL94]).

D'un point de vue pratique, l'implémentation de ce résultat est directe et suit les étapes suivantes :

1) Trouver l'expression du domaine faisable paramétré dans l'espace augmenté {données+paramètres}

$$A_{in}\mathbf{x} \le B_{in}\mathbf{p} + b_{in} \quad \Leftrightarrow \quad \begin{bmatrix} \mathrm{A}_{\mathrm{in}} & -B_{in} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} \le b_{in}$$

2) Trouver les faces d'ordre $n$ où $n$ est la dimension de l'espace des paramètres.

3) Ne retenir que les faces correspondant aux sommets paramétrés (en ignorant celles ayant une projection non inversible sur l'espace des paramètres).

4) Calculer le domaine de validité $VD_i$ pour chaque sommet paramétré $\sigma_i\,; i = 1, \ldots, n_s$.

5) Comparer chaque paire de sommets. Dans le cas d'une intersection non vide de leurs domaines de validité, dédoubler (split) les domaines en utilisant la fonction de coût linéaire. L'expression finale est une union des régions correspondant aux sommets paramétrés contenant l'optimum.

Une attention particulière doit être accordée à l'étape 5 avec la comparaison itérative des sommets et de leurs domaines de validité. Une routine possible peut être basée sur le procédé suivant.

***<u>Procédure CutDomains</u>*** *<u>(VD: ensemble de tous les domaines valides)</u>*

$n$=cardinal ($VD$)
i=1; j=2
while i<n+1
while j<n+1
  if $VD_j \cap VD_i \neq \emptyset$
      if $f^T x_i \le f^T x_j$ then $VD_j = VD_j - VD_i$
      if $f^T x_j \le f^T x_i$ then $VD_i = VD_i - VD_j$
      j=j+1;
    endif
end
i=i+1
end

**Remarque 6.20 :** Le procédé est initialisé avec l'ensemble des domaines de validité obtenus après l'énumération des faces d'ordre $n$ du polyèdre dans l'espace étendu (étape 2). ■

**Remarque 6.21 :** La différence de deux domaines convexes n'est pas une opération fermée et la sortie de la procédure est ainsi une union des sous-domaines convexes dans l'espace des paramètres. ■

Du point de vue de la commande prédictive robuste, la différence :

$$\aleph = \Re^n \backslash \left\{ \bigcup D_k\,; k = 1..n_D \right\} \tag{6.52}$$

décrit les régions de paramètres infaisables.

Une fois la partition en sous-domaines de l'espace des paramètres $D_k \subset \Re^n$ créée, celle-ci peut être utilisée en ligne pour évaluer la valeur optimale des commandes en évitant ainsi les procédures d'optimisation classiques.

***Algorithme 6.22 (solveur en ligne)***

| |
|---|
| 1) Trouver l'ensemble d'appartenance $D_k; k = 1..n_D$ pour le paramètre courant ***p***. Retour infaisable si aucun $D_k$ n'est trouvé. |
| 2) Calculer la commande optimale à partir de la solution explicite $k_{u_{MPC}} \leftarrow \boldsymbol{\chi}_k^*(\mathbf{p})$ (calculer les sommets paramétrés qui constituent la solution). La commande optimale est donnée par le premier élément de la séquence des commandes optimales $\boldsymbol{\chi}_k^*(\mathbf{p})$. |
| 3) Revenir à l'étape 1 avec un nouveau vecteur ***p***. ■ |

Au final, la structure globale de la fonction explicite est décrite par la proposition suivante.

> **Proposition 6.23 :** La formulation explicite de la loi de commande prédictive robuste est représentée par une fonction linéaire affine par régions polyédrales dans l'espace des paramètres de contexte.

**Preuve :** La solution des problèmes de programmation linéaire multiparamétrique est décrite par l'intermédiaire d'une union des régions polyédrales dans l'espace des paramètres de contexte. Pour chacun de ces domaines, l'optimum se trouve sur un sommet paramétré ou est donné par une combinaison de sommets paramétrés. Les sommets paramétrés à leur tour étant en fait des fonctions linéaires affines en le vecteur des paramètres (voir chapitre 2), il en résulte que la solution explicite est aussi une fonction linéaire affine par morceaux. Finalement en réunissant toutes les régions de l'espace des paramètres, on obtient la solution globale en terme d'une fonction linéaire affine par morceaux. ■

> **Remarque 6.24 :** Cette solution n'est pas unique puisque la solution du problème d'optimisation n'est pas unique. Il sera choisi dans la famille des solutions une forme linéaire affine par morceaux qui garantisse la continuité de la loi de commande prédictive.

### 6.4.3 Exemples de formulations explicites pour un problème mpLP

Considérons le problème décrit dans[GAL94] :

$$F^*(\mathbf{p}) = \min_{x_1,x_2} \quad -2x_1 - x_2$$
$$\begin{cases} x_1 + 3x_2 \le 9 - 2p_1 + p_2 \\ 2x_1 + x_2 \le 8 + p_1 - 2p_2 \\ x_1 \le 4 + p_1 + p_2 \\ -x_1 \le 0 \\ -x_2 \le 0 \end{cases} \tag{6.53}$$

pour :

$$-10 \le p_1 \le 10; -10 \le p_2 \le 10$$

La première étape vers la construction de la solution explicite est la construction du polyèdre non paramétré dans l'espace étendu de dimension 4. Le résultat sera un objet polyédral possédant 5 générateurs et 9 contraintes :

```
P =     Dimension: 4
    NbConstraints: 9
          NbRays: 5
            NbEq: 0
           NbBid: 0
            Rays: [5x6 double]
     Constraints: [9x6 double]
```

Il en résulte les sommets paramétrés récapitulés Tableau 6.1. La Figure 6.8 reproduit les régions dans l'espace des paramètres correspondant à des domaines faisables de forme régulière.

| | | |
|---|---|---|
| $s_1(\mathbf{p})$ | $\begin{bmatrix} x_1(\mathbf{p}) \\ x_2(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} 1 & -1.4 \\ -1 & 0.8 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ | $\begin{cases} 2.4p_2 + 1 \geq 0 \\ 0.33p_1 - 0.466p_2 + 1 \geq 0 \\ -0.5p_1 + 0.4p_2 + 1 \geq 0 \end{cases}$ |
| $s_2(\mathbf{p})$ | $\begin{bmatrix} x_1(\mathbf{p}) \\ x_2(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} 4 \\ 1.66 \end{bmatrix}$ | $\begin{cases} -2.4p_2 - 1 \geq 0 \\ 0.25p_1 + 0.25p_2 + 1 \geq 0 \\ -0.6p_1 + 1 \geq 0 \end{cases}$ |
| $s_3(\mathbf{p})$ | $\begin{bmatrix} x_1(\mathbf{p}) \\ x_2(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -0.66 & 0.33 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \end{bmatrix}$ | $\begin{cases} 0.33p_1 - 0.466p_2 + 1 \geq 0 \\ 0.25p_1 + 0.25p_2 + 1 \geq 0 \\ -0.22p_1 + 0.11p_2 + 1 \geq 0 \end{cases}$ |
| $s_4(\mathbf{p})$ | $\begin{bmatrix} x_1(\mathbf{p}) \\ x_2(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} 9 \\ 0 \end{bmatrix}$ | $\begin{cases} 0.5p_1 - 0.4p_2 - 1 \geq 0 \\ 0.6p_1 - 1 \geq 0 \\ -0.22p_1 + 0.11p_2 + 1 \geq 0 \end{cases}$ |
| $s_5(\mathbf{p})$ | $\begin{bmatrix} x_1(\mathbf{p}) \\ x_2(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & -4 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} 4 \\ 0 \end{bmatrix}$ | $\begin{cases} 2.4p_2 + 1 \geq 0 \\ 0.25p_1 - 0.25p_2 + 1 \geq 0 \\ -0.6p_1 - 2.4p_2 \geq 0 \end{cases}$ |
| $s_6(\mathbf{p})$ | $\begin{bmatrix} x_1(\mathbf{p}) \\ x_2(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & -2 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 8 \end{bmatrix}$ | $\begin{cases} -0.33p_2 + 0.46 - 1 \geq 0 \\ 0.25p_1 + 0.25p_2 + 1 \geq 0 \\ 0.125p_1 - 0.25p_2 + 1 \geq 0 \end{cases}$ |
| $s_7(\mathbf{p})$ | $\begin{bmatrix} x_1(\mathbf{p}) \\ x_2(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} 0.5 & -1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} 4 \\ 0 \end{bmatrix}$ | $\begin{cases} -0.5p_1 + 0.4p_2 + 1 \geq 0 \\ 0.1p_1 + 0.4p_2 \geq 0 \\ 0.125p_1 - 0.25p_2 + 1 \geq 0 \end{cases}$ |
| $s_8(\mathbf{p})$ | $\begin{bmatrix} x_1(\mathbf{p}) \\ x_2(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} 4 \\ 0 \end{bmatrix}$ | $\begin{cases} -p_1 - 4p_2 \geq 0 \\ -0.6p_1 + 1 \geq 0 \\ -0.25p_1 + 0.25p_2 + 1 \geq 0 \end{cases}$ |
| $s_9(\mathbf{p})$ | $\begin{bmatrix} x_1(\mathbf{p}) \\ x_2(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ | $\begin{cases} 0.125p_1 - 0.25p_2 + 1 \geq 0 \\ -0..22p_1 + 0.11p_2 + 1 \geq 0 \\ 0.25p_1 + 0.25p_2 + 1 \geq 0 \end{cases}$ |

**Tableau 6.1 : Sommets paramétrés**

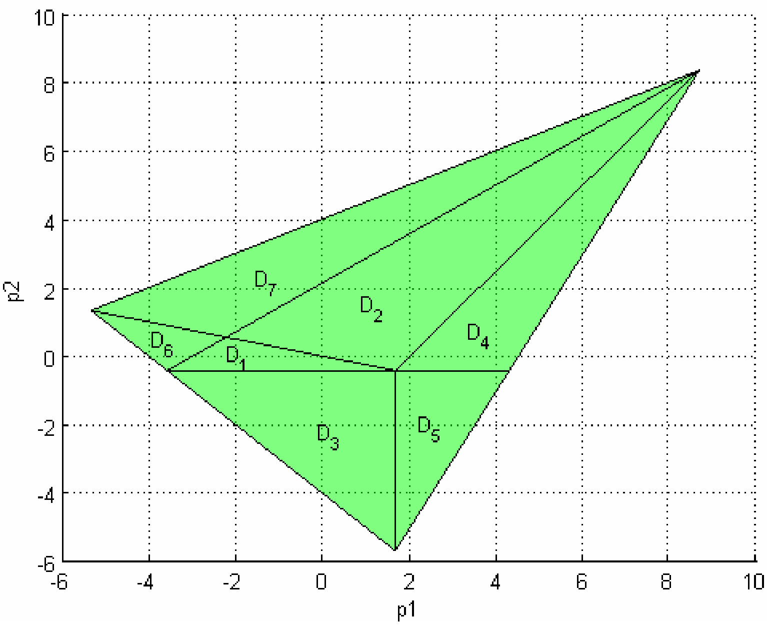



**Figure 6.8 : Découpage de l'espace des paramètres en régions de forme régulière**

Pour chacune de ces régions, il suffit alors de chercher les sommets qui procurent la valeur optimale de la fonction de coût pour pouvoir enfin décrire l'ensemble des solutions possibles du problème d'optimisation. Ainsi on peut observer que pour les régions $D_3$, $D_4$ et $D_5$, il existe un seul sommet paramétré qui décrit l'optimum ($s_2(\mathbf{p})$, $s_4(\mathbf{p})$ et $s_4(\mathbf{p})$ respectivement). En revanche, pour les autres régions $D_1$, $D_2$, $D_6$ et $D_7$, il existe une paire de sommets paramétrés ($\{s_1(\mathbf{p}), s_5(\mathbf{p})\}$, $\{s_1(\mathbf{p}), s_7(\mathbf{p})\}$, $\{s_5(\mathbf{p}), s_6(\mathbf{p})\}$ et $\{s_6(\mathbf{p}), s_7(\mathbf{p})\}$ respectivement) pour laquelle la valeur optimale de la fonction de coût est atteinte. Ceci peut être résumé par les données récapitulées Tableau 6.2.

| Région critique | Argument optimal | Valeur optimale |
|---|---|---|
| $D_1$ | *Conv.hull* $\{s_1(\mathbf{p}), s_5(\mathbf{p})\}$ | $F^*(\mathbf{p}) = -p_1 + 2p_2 - 8$ |
| $D_2$ | *Conv.hull* $\{s_1(\mathbf{p}), s_7(\mathbf{p})\}$ | $F^*(\mathbf{p}) = -p_1 + 2p_2 - 8$ |
| $D_6$ | *Conv.hull* $\{s_5(\mathbf{p}), s_6(\mathbf{p})\}$ | $F^*(\mathbf{p}) = -p_1 + 2p_2 - 8$ |
| $D_7$ | *Conv.hull* $\{s_6(\mathbf{p}), s_7(\mathbf{p})\}$ | $F^*(\mathbf{p}) = -p_1 + 2p_2 - 8$ |
| $D_3$ | $s_2(\mathbf{p})$ | $F^*(\mathbf{p}) = -p_1 - 2p_2 - 9.66$ |
| $D_4 \cup D_5$ | $s_4(\mathbf{p})$ | $F^*(\mathbf{p}) = 4p_1 - 2p_2 - 18$ |

**Tableau 6.2 : Valeur optimale selon les régions de forme régulière**

La région faisable dans l'espace des paramètres peut être partitionnée en régions ayant la même valeur optimale de la fonction de coût linéaire (Figure 6.9).

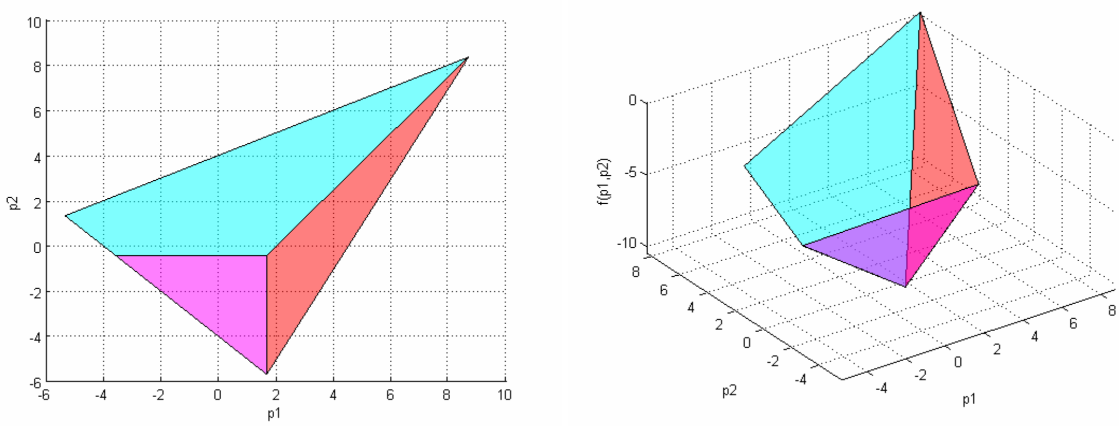



**Figure 6.9 : a) Partition dans l'espace des paramètres correspondant à des expressions linéaires de la valeur optimale de la fonction de coût b) Fonction effective**

Cette expression explicite de la solution optimale peut être obtenue également par des méthodes basées sur l'énumération des combinaisons de contraintes actives pour la valeur optimale. On peut trouver la correspondance avec la méthode présentée ici en remarquant que pour $D_3$, la première et la troisième contrainte sont saturées, pour $D_4 \cup D_5$ la première et la cinquième contrainte sont saturées et pour $D_1$, $D_2$, $D_6$ et $D_7$ la deuxième contrainte est active.

On constate aussi dans le cas des formulations mpLP que l'on retrouve les problèmes de dégénérescence (par exemple à la frontière entre $D_2$ et $D_4$, ou $D_3$ et $D_5$) qui sont encore une fois traités de façon élégante à l'aide des régions correspondant à une forme régulière du domaine faisable.

Il reste encore à clarifier pour ce type de problème la façon d'isoler au sein de la famille des solutions possibles un candidat ayant la propriété de continuité. En effet, on peut observer au passage de zones, $D_1$ à $D_6$ et $D_2$ à $D_7$ des sauts dans l'argument optimal dus à une infinité de solutions possibles.
Certaines combinaisons possibles sont décrites Figure 6.10 (sans être exhaustives). Du point de vue positionnement en ligne, il est préférable d'avoir un minimum de partitions.

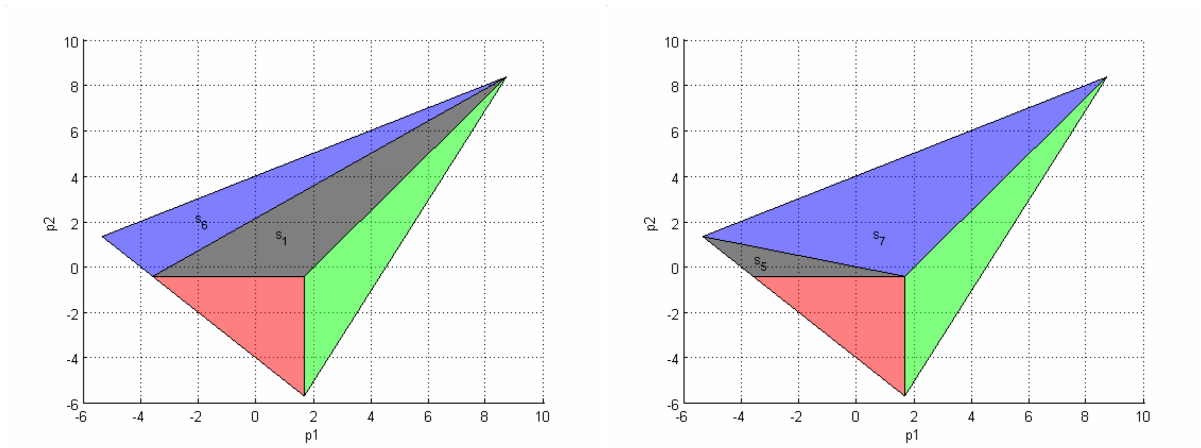



**Figure 6.10 : Exemples de partitions caractérisant une solution explicite continue et linéaire par morceaux**

## *6.5 Exemples de commande prédictive robuste*

**Exemple 6.24** : Considérons l'exemple décrit dans [SM98] :

$$x_{t+1} = x_t + u_t + v_t$$

Afin d'illustrer les idées théoriques présentées précédemment, une prédiction sur deux pas est mise en œuvre. Ainsi le problème d'optimisation suivant doit être résolu à chaque pas d'échantillonnage :

$$J(x_t) = \min_{u_t, u_{t+1}} \sum_{k=0}^{1} \left|x_{t+k|t}\right| + 10\left|u_{t+k}\right|$$

$$\begin{cases} -1.2 \le x_{t+k|t} \le 2, k = 0,1,2 \\ -1 \le x_{t+2|t} \le 1, \\ -1 \le v_{t+k} \le 1, k = 0,1 \end{cases} \tag{6.54}$$

En ignorant dans un premier temps les perturbations, la solution explicite du problème se déduit par l'approche géométrique présentée au paragraphe précédent en examinant les 22 sommets paramétrés. Après l'étape de discrimination des domaines de validité, la loi explicite RMPC est donnée Tableau 6.3.

| | Loi de commande affine | Domaine de validité |
|---|---|---|
| $D_1$ | $u_t = -x_t - 1$ | $-1.2 \le x_t \le -1$ |
| $D_2$ | 0 | $-1 \le x_t \le 0$ |
| $D_3$ | 0 | $0 \le x_t \le 1$ |
| $D_4$ | $u_t = -x_t + 1$ | $1 \le x_t \le 2$ |

**Tableau 6.3 : Domaines de validité**

On peut observer l'existence de deux domaines ayant la même loi de commande ; cette particularité provient du fait que la fonction de coût change de pente, le maximum se trouve ainsi sur différents sommets paramétrés dans l'espace augmenté, mais la contribution à la première valeur de commande (appliquée selon le principe de l'horizon glissant) reste la même. Dans ce cas-ci, leur union étant un ensemble convexe, ils peuvent être réunis en une seule région :

$$D_{2-3} = \{-1 \le x_t \le 1\}$$

Dans le cas général, cette opération peut être réalisée par des outils de reconnaissance de l'union convexe des polyèdres (voir [BFT02] pour des détails sur cette opération géométrique).

La mise en œuvre de cette loi de commande pour une condition initiale $x_0 = -1{,}2$ maintient la trajectoire du système à l'intérieur des contraintes dans le cas sans perturbation (Figure 6.11a). Si le même régulateur est implanté en présence d'une perturbation $v_k = -1/k,\ k \geq 1$ agissant sur le système, la trajectoire va violer les contraintes (Figure 6.11b).

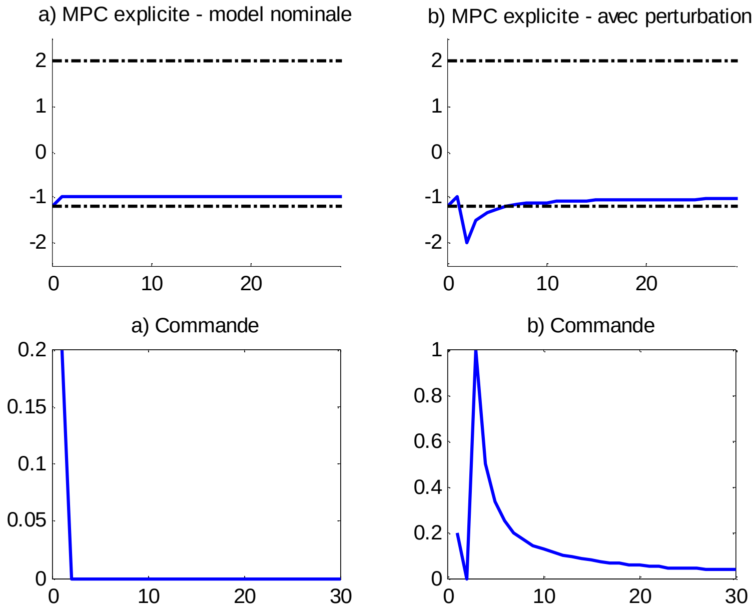



**Figure 6.11 : a) Comportement (sortie/commande) de la loi explicite (Tableau 6.3) appliquée au nominal ; b) Comportement (sortie/commande) de la loi explicite (Tableau 6.3) appliquée en présence de perturbations**

On considère la formulation explicite robuste MPC réalisée dans la version 'min-max' :

$$V(x_t) = \min_{u_t, u_{t+1}} \max_{v_t, v_{t+1}} \sum_{k=0}^{1} \left| x_{t+k|t} \right| + 10 \left| u_{t+k} \right|$$
$$\begin{cases} -1.2 \leq x_{t+k|t} \leq 2, k = 0,1,2 \\ -1 \leq x_{t+2|t} \leq 1, \\ -1 \leq v_{t+k} \leq 1, k = 0,1 \end{cases} \tag{6.55}$$

Il est alors nécessaire de résoudre le problème d'optimisation linéaire équivalent. Une premier constat est que, sous cette forme, il n'y a aucune solution pour $x_0 = -1{,}2$ car l'optimisation est infaisable. On s'aperçoit en effet qu'aucune loi de commande :

$$u_{t|t} = a_1 x_t + b_1$$
$$u_{t+1|t} = a_2 x_t + b_2 u_{t|t} + c_2$$

ne peut garantir la satisfaction robuste des contraintes. Quand on écrit explicitement les contraintes dans (6.55) pour les combinaisons extrémales des perturbations, ce blocage apparaît clairement :

$$\begin{aligned} \begin{bmatrix} v_t \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} &\Rightarrow -1 \leq x_t + u_{t|t} + u_{t+1|t} + 2 \leq 1 \Rightarrow -3 \leq x_t + u_{t|t} + u_{t+1|t} \leq -1 \\ \begin{bmatrix} v_t \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} &\Rightarrow -1 \leq x_t + u_{t|t} + u_{t+1|t} - 2 \leq 1 \Rightarrow 1 \leq x_t + u_{t|t} + u_{t+1|t} \leq 3 \end{aligned} \tag{6.56}$$

ce qui signifie qu'il n'y a aucune combinaison de commande maintenant la faisabilité pour toutes les réalisations possibles des perturbations. Cette impossibilité demeure en fait aussi longtemps qu'un type de prédiction 'en boucle ouverte' est considéré, pour lequel il n'est pas possible de remédier aux dommages cumulés des perturbations. Dans ce cas, une formulation 'en boucle fermée' fournit les degrés de liberté nécessaires. On doit alors résoudre le problème imbriqué :

$$V(x_t)=\min_{u_t}\max_{v_t}\min_{u_{t+1}}\max_{v_{t+1}}\sum_{k=0}^{1}\left|x_{t+k|t}\right|+10\left|u_{t+k}\right|$$
$$\begin{cases}-1.2\leq x_{t+k|t}\leq 2, k=0,1,2\\ -1\leq x_{t+2|t}\leq 1,\\ -1\leq v_{t+k}\leq 1, k=0,1\end{cases}\tag{6.57}$$

Avec l'approche géométrique présentée dans la section précédente, la solution explicite peut être obtenue en résolvant la minimisation interne tout d'abord :

$$V(x_t,u_t,v_t)=\min_{u_{t+1}}\max_{v_{t+1}}\left|x_t\right|+\left|x_t+u_t+v_t\right|+10\left|u_t\right|+10\left|u_{t+1}\right|$$
$$\begin{cases}-1.2\leq x_{t+k|t}\leq 2, k=0,1,2\\ -1\leq x_{t+2|t}\leq 1,\\ -1\leq v_{t+k}\leq 1, k=0,1\end{cases}\tag{6.58}$$

La solution est directe puisqu'il y a exactement 2 sommets paramétrés sur lesquels le minimum se trouve et la loi de commande associée est :

$$u_{t+1|t}=-(x_t+u_{t|t}+v_t)=-x_{t+1}\quad \text{pour}\quad -1.2\leq x_t\leq 2\tag{6.59}$$

Il faut insister sur le fait que la loi de commande utilise ici l'information additionnelle disponible pour la prédiction, ce qui n'était pas le cas du problème (6.55). Avec ce résultat, le problème externe d'optimisation s'exprime par :

$$V(x_t)=\min_{u_t}\max_{v_t}\quad \left|x_t\right|+\left|11x_{t+1|t}\right|+\left|10u_t\right|$$
$$\begin{cases}-1.2\leq x_{t+k|t}\leq 2, k=0,1\\ -1\leq v_t\leq 1\end{cases}\tag{6.60}$$

la solution explicite est une fois de plus directe puisqu'il n'y a que deux sommets paramétrés non dégénérés décrivant le lieu géométrique du minimum. En appliquant cette loi RMPC :

$$u_t=-x_t\qquad \text{pour}\quad -1.2\leq x_t\leq 2$$

le système affecté par des perturbations est régulé à l'origine (Figure 6.12), avec satisfaction des contraintes.

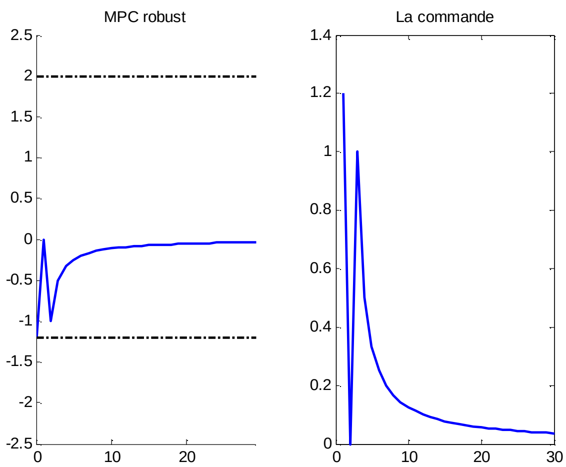



**Figure 6.12 : Trajectoire du système pour la loi explicite robuste**

Les solutions des problèmes d'optimisation (6.54) – cas nominal, (6.58), (6.60) ont été obtenues en utilisant des procédures basées sur la double représentation des polyèdres paramétrées en 2, 0,39 et 0,91 secondes respectivement. Cependant, pour des systèmes complexes, le temps de calcul peut exploser car le nombre de sommets paramétrés évolue de façon exponentielle en fonction du nombre de contraintes supplémentaires ajoutées pendant les étapes de transformation du problèmes 'min-max' en problème linéaire équivalent.

**Exemple 6.25** : On considère un problème similaire mais cette fois avec une approche GPC. Le système :

$$y_t = y_{t-1} + u_t + v_t \tag{6.61}$$

doit être régulé à l'origine en minimisant une fonction de coût avec les horizons $N_u = 2, N_2 = 2$ :

$$J = \sum_{k=0}^{1} \left\{ |y_{t+k}| + (k+2)|u_{t+k}| \right\} \tag{6.62}$$

Les contraints opérationnelles et terminales suivantes ainsi que les limitations des perturbations suivantes seront prises en compte :

$$\begin{cases} -1.2 \le \hat{y}_{t+k} \le 1.2, & k = 0,1,2 \\ -1 \le \hat{y}_{t+2} \le 1 \\ -0.4 \le v_{t+k} \le 0.4, & k = 0,1 \end{cases} \tag{6.63}$$

Si ces perturbations sont ignorées, la solution explicite peut être trouvée via l'approche géométrique en élaborant la double description d'un polyèdre paramétré de dimension 7 (deux arguments $\{u_t ; u_{t+1}\}$, deux paramètres $\{y_t ; y_{t-1}\}$ et 3 variables auxiliaires pour chaque terme du critère $|y_{t+1}|$, $|u_t|$ et $|u_{t+1}|$ comme dans (6.42)). Les sorties connues $\left[y_t \; y_{t-1}\right]^T$ nécessaires pour la construction des prédictions doivent figurer dans l'espace des paramètres. On trouve 30 sommets paramétrés provenant des faces d'ordre 2 ayant des projections non dégénérées sur l'espace des paramètres. Des règles de projection se déduisent alors des comparaisons de leurs domaines de validité, le résultat étant un tableau (look-up table) de correcteurs polynomiaux affines (Tableau 6.4).

| Loi de commande affine | Domaine de validité |
|---|---|
| $\Delta u_t = (-1.5 + q^{-1}) y_t + 0.5$ | $D_1 = \left\{ \begin{bmatrix} 3 & -2 \\ 0 & -1 \\ -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} \ge \begin{bmatrix} 1 \\ -1.2 \\ -1.2 \\ -1.2 \end{bmatrix} \right\} \cup \left\{ \begin{bmatrix} 0 & -1 \\ -3 & 2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} \ge \begin{bmatrix} -1.2 \\ 1 \\ -1.2 \\ -1.2 \end{bmatrix} \right\}$ |
| $\Delta u_t = 0$ | $D_2 = \left\{ \begin{bmatrix} -2 & 1 \\ 0 & -1 \\ 3 & -2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} \ge \begin{bmatrix} 0 \\ -1.2 \\ -1 \\ -1.2 \end{bmatrix} \right\} \cup \left\{ \begin{bmatrix} 0 & 1 \\ -3 & 2 \\ 0 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} \ge \begin{bmatrix} -1.2 \\ -1 \\ -1.2 \\ 0 \end{bmatrix} \right\}$ |

**Tableau 6.4. Tableau des solutions explicites, exemple 6.25**

On peut observer que $D_2$ est un ensemble convexe constitué par l'union de deux autres sous domaines convexes (Figure 6.13c domaine en rouge). Ce découpage superflu est dû au fait que la fonction de coût change de pente et que l'optimum se trouve sur un sommet paramétré différent dans l'espace augmenté, même si la projection est similaire. En général de telles unions peuvent être effectuées avec des outils de reconnaissance d'unions convexes de polyèdres [BFT02].

La loi de commande explicite trouvée a été simulée avec $\left[y_t \; y_{t-1}\right]^T = [-1 -1]^T$ comme condition initiale. Si le système est affecté par une perturbation $v_{t+k} = (-1)^k / k, \; k \ge 1$, la trajectoire violera les contraintes à cause de ses effets nuisibles non pris en considération lors de la synthèse. La loi de commande explicite n'est pas définie pour la deuxième et la troisième itération mais ensuite, dès lors que la valeur de la perturbation diminue, elle redevient faisable. A noter qu'en raison de la pondération plus importante sur l'effort de commande dans le critère, le système évolue vers une valeur en régime stationnaire différente de 0 quand les perturbations diminuent. La violation des contraintes est évidente Figure 6.13 décrivant le découpage de la loi explicite de commande et les trajectoires de sortie et de commande. On observe que, pour les conditions initiales choisies, la trajectoire dépasse la zone des combinaisons faisables.

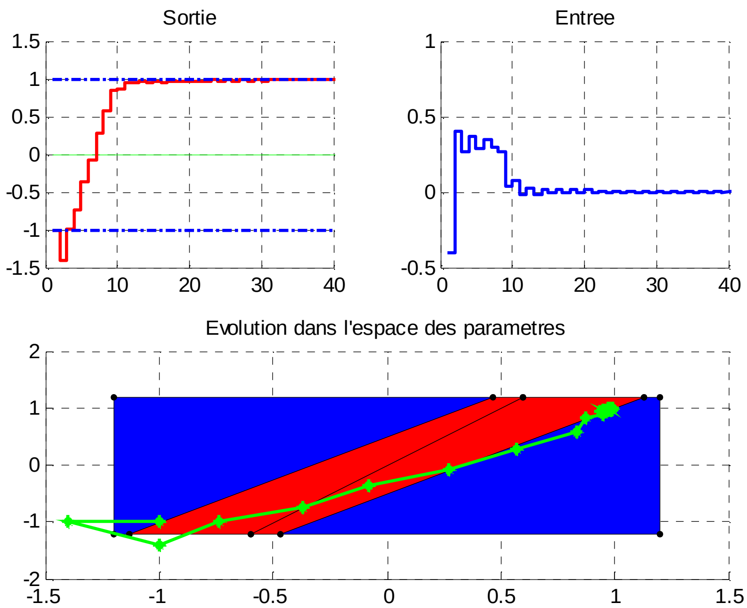



**Figure 6.13 : a) et b) Simulation temporelle (incluant l'action des perturbations) de la loi GPC synthétisée pour le modèle nominal c) Découpage de l'espace des paramètres correspondant à la formulation explicite du GPC dans le cas nominal et trajectoire pendant la simulation**

Avec une géométrique similaire, la formulation explicite de la loi GPC robuste basée sur le problème 'min-max' :

$$\{\Delta u_t, \Delta u_{t+1}\} = \min_{\Delta u_t, \Delta u_{t+1}} \max_{v_t, v_{t+1}} J \tag{6.64}$$

peut être construite. La solution se trouve sur les sommets paramétrés d'un polyèdre de dimension 7 avec 2 paramètres. Il y a 94 sommets paramétrés non dégénérés. Après l'étape d'élimination, seuls 10 d'entre eux sont retenus et la loi de commande explicite est exprimée par une loi affine par morceaux, Figure 6.14c.

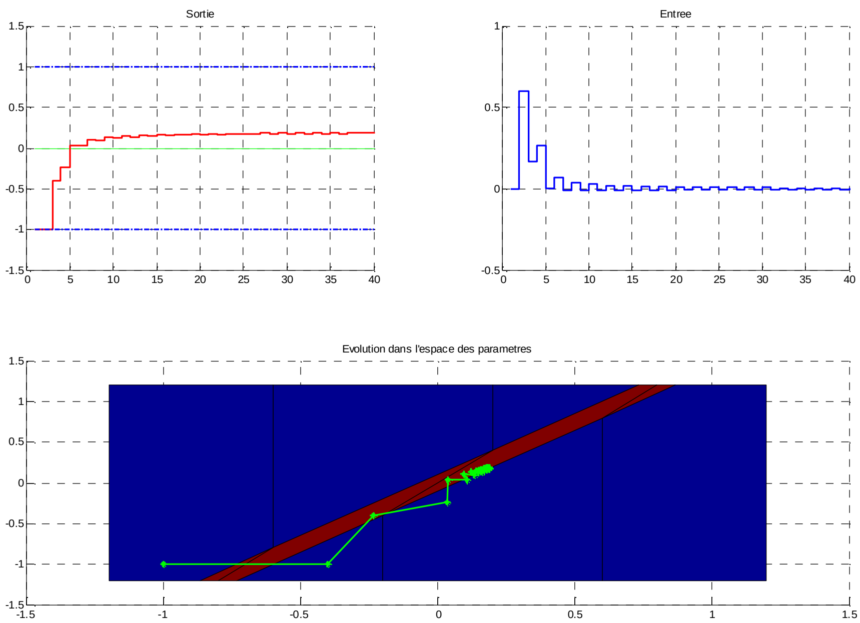



**Figure 6.14 : a) et b) Simulation temporelle du GPC robuste c) Découpage dans l'espace des paramètres correspondant au GPC robuste et trajectoire pendant la simulation**

La Figure 6.14a-b illustre la réponse temporelle du système soumis à cette loi de commande robuste. La satisfaction robuste des contraintes est mise en évidence, et ce dès les premiers instants pour lesquels la loi synthétisée nominale pour le modèle nominal avait échoué. La sortie ne revient pas à 0 par manque d'action intégrale dans la loi de commande. Cependant, si des perturbations plus violentes doivent être prises en considération, cette stratégie 'en boucle ouverte' peut ne pas offrir assez de degrés de liberté pour la commande robuste du système, imposant une approche 'en boucle fermée' :

$$\{\Delta u_t, \Delta u_{t+1}\} = \min_{\Delta u_t} \max_{v_t} \min_{\Delta u_{t+1}} \max_{v_{t+1}} U \tag{6.65}$$

La formulation explicite est mise en œuvre via un problème 'min-max' imbriqué. Pour l'exemple précédent ceci signifie :

$$\begin{aligned} \Delta\tilde{u}_{t+1} &= \arg\min_{\Delta u_{t+1}} \max_{v_{t+1}} U = \\ &= \begin{cases} 0 & \text{pour } -0{,}6 \le y_{t+1} \le 0{,}6 \\ -2\Delta u_t - v_t + (-3+2q^{-1})y_t + 0{,}6 & \text{pour } 0{,}6 \le y_{t+1} \\ -2\Delta u_t - v_t + (-3+2q^{-1})y_t - 0{,}6 & \text{pour } y_{t+1} \le -0{,}6 \end{cases} \end{aligned} \tag{6.66}$$

soit l'incrément d'entrée à appliquer effectivement :

$$\Delta\tilde{u}_t = \arg\min_{\Delta u_t} \max_{v_t} U(\Delta u_{t+1} = \Delta\tilde{u}_{t+1}) \tag{6.67}$$

Pour notre exemple, le découpage de la solution explicite est similaire à celui de la Figure 6.14c.

## *6.6 Conclusions*

Ce chapitre a proposé une étude des méthodes de synthèse de régulateurs prédictifs robustes à partir de critères de type 'min-max' résolus en tenant compte de leur équivalence avec des problèmes de programmation linéaire multiparamétrique. Des formulations en ce sens ont été décrites pour le cas de modèles CARIMA (commande prédictive généralisée) et pour des modèles d'état.

Les résultats obtenus ont permis de conclure que la robustesse des algorithmes de commande prédictive dans le cas sous contraintes doit tenir compte en premier lieu des aspects de satisfaction robuste de contraintes puis juger du compromis vis-à-vis des performances dynamiques (le cas sans contraintes se résumait à ce seul deuxième aspect). En effet, les problèmes d'infaisabilité deviennent primordiaux en présence de perturbations et d'incertitudes de modélisation pour les systèmes soumis à des limitations.

L'originalité apportée par ce chapitre consiste en la formulation explicite de la loi de commande prédictive robuste élaborée à partir d'arguments géométriques. Si le domaine faisable des problèmes d'optimisation linéaire peut toujours être décrit par un polyèdre paramétré, la solution explicite est basée sur l'expression des sommets paramétrés de ce polyèdre. En conséquence, la loi prédictive explicite étant décrite par des sommets paramétrés ou par des combinaisons de sommets paramétrés a la forme d'une loi linéaire affine par morceaux. Cependant, dès lors que la solution explicite du problème d'optimisation n'est pas unique, il est préférable de sélectionner d'un point de vue pratique une fonction continue linéaire par morceaux.

La recherche dans le domaine de la commande prédictive robuste est très active car les problèmes de faisabilité [Ker00] ou de complexité de la loi explicite résultante [BBM01], [KM02] sont sources de sujets très intéressants. Mentionnons par exemple les méthodes basées sur des expressions explicites approximatives [JG02] qui peuvent diminuer la finesse des partitions dans l'espace des paramètres.

# 7. Application de la commande prédictive sous contraintes à un problème de positionnement

Ce chapitre envisage l'application des structures de commande prédictive sous contraintes développées lors des chapitres précédents à un système électromécanique. Le problème choisi consiste à commander en position une machine asynchrone, processus réputé sensible du point de vue des performances dynamiques. L'approche se sert des modèles existants pour aboutir à la construction de lois de type GPC de faible complexité, qui sont ensuite validées à l'aide d'un simulateur non linéaire puis sur plate-forme expérimentale. Le problème de positionnement de la machine asynchrone envisagé ici s'intègre dans la problématique dite 'transitique rapide' et implique pour le banc considéré des périodes d'échantillonnage faibles de sorte que l'application des lois de commande basées sur la résolution totale en ligne des problèmes d'optimisation s'avère déconseillée, leur complexité ne permettant pas la construction de la commande en temps réel.

La présence de contraintes et leurs influences sur le fonctionnement et sur la structure de la loi de commande constituent les principaux points abordés. L'analyse et la synthèse de la loi de commande prédictive, pour l'exemple choisi, reprennent les résultats théoriques décrits lors des chapitres précédents. On retrouve ainsi les trois types de contraintes rencontrés le plus souvent dans la pratique : les contraintes de fonctionnement (caractérisées dans notre cas par des contraintes sur le signal de commande, image du couple sur la machine), les contraintes de performance (illustrées ici par des contraintes sur l'erreur de positionnement) ainsi que des contraintes de stabilité (contraintes à la fin de l'horizon de prédiction). Soulignons que, dans la pratique, la commande des machines asynchrones est assurée par des lois classiques et qu'en présence de contraintes, ces schémas sont très souvent invalidés ou nécessitent des mécanismes de supervision (souvent difficiles à régler) pour empêcher la violation des contraintes. La philosophie de commande prédictive offre en ce sens une formulation temporelle qui permet l'inclusion des contraintes dès le début du processus de synthèse.

L'analyse géométrique des domaines faisables va nous permettre d'identifier les degrés de liberté des différentes lois, les limites du domaines faisable en fonction de l'évolution des paramètres de contexte, ainsi que l'expression explicite des lois de commande prédictive par des fonctions linéaires par morceaux.

Ce chapitre débute par une introduction proposant une description rapide de l'application considérée et du modèle utilisé, ce dernier constituant l'élément primordial pour la construction des lois prédictives. Après un rappel des schémas de commande classiquement implantés sur cette machine et qui serviront de référence en terme de performance, les techniques de commande prédictive sont ensuite élaborées en soulignant les avantages offerts durant la phase de synthèse. Les premières limitations introduites seront induites par des contraintes terminales permettant l'utilisation d'horizons de prédiction plus faibles. Les contraintes de type inégalité considérées ensuite seront placées sur la commande et sur la sortie du système, mettant en évidence le phénomène d'infaisabilité. L'influence de bruit de mesure sera également examinée. Pour chaque cas, les lois de commande explicites associées seront identifiées.

Les résultats décrits dans ce chapitre sont le fruit de la collaboration entre SUPELEC et le LGEP (Laboratoire de Génie Electrique de Paris), laboratoire où est implanté le banc considéré. Ils ont donné lieu aux publications [OD05c], [OD05g].

## *7.1 Introduction*

L'application choisie se classe dans la catégorie des machines électriques, et plus particulièrement des machines asynchrones, type de moteurs électriques très répandu de nos jours, avec des versions variant de quelques watts jusqu'aux mégawatts. Même si la machine asynchrone est très appréciée pour ses qualités de robustesse, de fiabilité et de coût, les variateurs de vitesse ont parfois posé problème (alimentation à fréquence variable), des réponses acceptables n'ayant été fournies qu'après l'émergence de la technologie à semi-conducteurs [39].
Les progrès de l'électronique de puissance ont ouvert la voie à l'application de stratégies de commande avancées une effervescence a ainsi été constatée depuis plusieurs années en ce qui concerne les sujets de recherche liés à la commande d'entraînements électriques, concrétisée au niveau théorique par les travaux de [Can00], mais aussi par la mise en place de plates-formes d'essais permettant des tests comparatifs de lois de commande par exemple [MB02]. Dans la suite des développements, la présentation du banc d'essais ainsi que la partie de modélisation nécessaire pour l'application des techniques prédictives sont largement inspirées des travaux de [MB02] inclus dans un numéro du JESA (Journal européen des systèmes automatisés) dédié à ce sujet.

[39] Rappelons que les machines asynchrones ont été inventées dès la fin du $19^{ème}$ siècle.

La structure de l'actionneur asynchrone regroupe les blocs suivants (Figure 7.1) :

- L'alimentation en tension constante
- Un onduleur triphasé de tension
- Des capteurs de courant
- La machine asynchrone en elle-même
- Le capteur de position
- Une carte interface entre les éléments mentionnés ci-dessus et l'ordinateur
- L'ordinateur

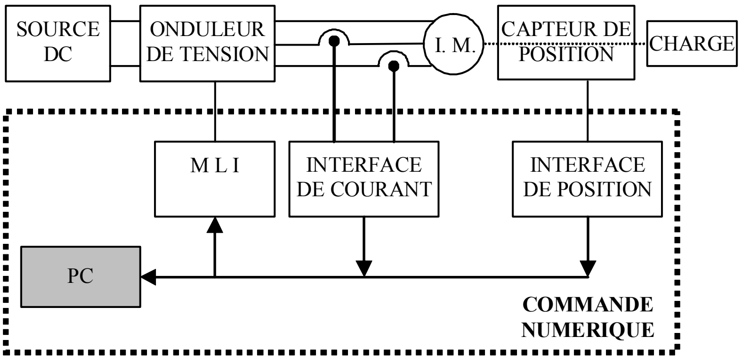



**Figure 7.1 : Structure du banc d'essai**

La machine asynchrone triphasée (à cage de 1,1kW ayant 2 paires de pôles, couplées en étoile et délivrant un couple de 7Nm) est alimentée par un onduleur de tension triphasé fonctionnant en modulation de largeur d'impulsion (MLI) à une fréquence de 13,05kHz (76,6µs). L'onduleur à son tour est alimenté par la source en tension constante (300V) et délivre un courant maximum par phase de 10A, qui permet d'obtenir un couple maximum de 21Nm. La charge de la machine est constituée par un frein à poudre. La mesure de la position mécanique est effectuée par l'intermédiaire d'un codeur incrémental, offrant une résolution de 14400 points par tour mécanique. Cette mesure de la position, ainsi que les deux courants de phase de la machine (à l'aide de sondes à effet Hall) et la tension d'alimentation de l'onduleur représentent les grandeurs disponibles pour le processus de commande. Les courants et la tension sont filtrés, avant échantillonnage, par des filtres passe-bas de type Butterworth d'ordre 4 de fréquence de coupure de 6,12 kHz. Il est possible à partir de ces mesures d'estimer l'état magnétique de la machine de sorte que la commande de la position mécanique peut se faire à une période d'échantillonnage de 153,2 µs. En revanche, la commande en tension de la machine doit être calculée à chaque période de l'onduleur, soit 76,6 µs.

La machine asynchrone (Figure 7.2) est pilotée par une structure de commande de type cascade, les boucles internes de couple et de flux étant conçues autour d'une commande vectorielle à flux orienté direct (DFOC, de 'Direct field oriented control'), la boucle externe de position incluant la loi prédictive qui nous intéresse ici, exprimée sous forme RST ou RST par morceaux. Plus précisément, la boucle interne est réalisée dans le repère du flux rotorique (pour lequel l'angle et le flux sont estimés) et regroupe deux boucles de courant imposant les courants $i_{sd}$ et $i_{sq}$ (courants donnés par les équations statoriques), contrôlés par des régulateurs PI. Le bloc RC-MLI (Régulation de courant et Modulation de Largeur d'impulsions), Figure 7.2, représente l'asservissement des ces courants $i_{sd}$ et $i_{sq}$ et la génération des signaux de commande en modulation de largeur d'impulsion. Une multitude de références peut être trouvée dans la littérature [LEO01], [MB02], relative à la commande vectorielle au niveau de la boucle interne des machines asynchrones. La boucle externe comprend le correcteur de position (fournissant le couple de référence à la boucle interne) ainsi qu'un module de désexcitation nécessaire lorsque la vitesse du moteur dépasse la vitesse nominale.

Les simulations et les essais effectués se sont focalisés autour de cette boucle extérieure de position, la boucle interne étant assimilée à une boite noire.

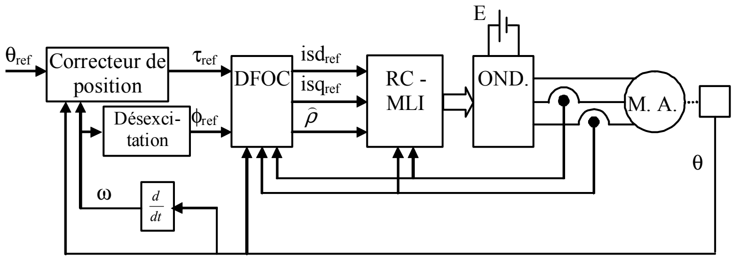



**Figure 7.2 : Structure de commande de la machine asynchrone**

La commande est appliquée via un PC fonctionnant avec le système ‘temps réel’ RT-Linux. La programmation de la commande s’effectue en langage C. On dispose par ailleurs d’un simulateur du fonctionnement du système, écrit également en langage C. Ce simulateur est composé d’un modèle non linéaire du système, très proche de la réalité, et permet de simuler la commande avant son implantation réelle sur le banc d’essai. Ainsi, avec le système sous RT-Linux, le programme de commande développé en simulation est directement utilisable sur le système expérimental.

## *7.2 Modélisation*

La construction des lois de commande prédictive nécessite l’existence d’un modèle de comportement du moteur issu d’une étape d’identification expérimentale. Pour la plate-forme considérée, l’étape de modélisation effectuée par le LGEP révèle les éléments caractéristiques suivants.

**Modèle du système électrique (commande en couple)**

La partie électrique est représentée par une fonction de transfert d’ordre un du couple de référence vers le couple effectif. Elle englobe la boucle de courant, la commande à flux orienté, la MLI et la dynamique de l’onduleur. Cette fonction fait donc intervenir une constante de temps $\tau_e$ :

$$\frac{C}{C^*} = \frac{1}{1+\tau_e\, s} \qquad s \text{ variable de Laplace} \tag{7.1}$$

où $C$ est le couple électromagnétique et $C^*$ la consigne en couple.

**Modèle du système mécanique**

La partie mécanique du moteur est caractérisée par l’inertie du moteur et de la charge ramenée sur l’arbre moteur, $J$, le coefficient de frottement visqueux $f$ et le couple de charge $\Gamma$. La constante $\tau_e$ représentant la partie électrique est négligeable en comparaison avec la constante de temps mécanique équivalente. La relation entre le couple électromécanique $C$ et la vitesse $\Omega$ est donc de la forme :

$$\left.\begin{aligned} C &= J\frac{d\Omega}{dt} + f\Omega \\ C &= (Js+f)\Omega \end{aligned}\right\} \Rightarrow \frac{\Omega}{C} = \frac{1}{Js+f} \tag{7.2}$$

**Facteur de transformation des unités**

La vitesse fournie en [rad/s] nécessite l’introduction d’un facteur de proportionnalité pour déterminer son expression en [tour/min] :

$$\Omega[rad/s] \xrightarrow{K=\frac{60}{2\pi}} \tilde{\Omega}[tour/\min] \tag{7.3}$$

**Intégration de la vitesse**

La position est liée à la vitesse par une simple intégration :

$$\left.\begin{array}{l}\tilde{\Omega} = \dfrac{d\varphi}{dt} = \dot{\theta} \\ \tilde{\Omega} = s\theta\end{array}\right\} \frac{\theta}{\tilde{\Omega}} = \frac{1}{s} \tag{7.4}$$

Ces éléments permettent de construire un modèle complet du système à commander, utilisé lors de la synthèse des lois prédictives. La Figure 7.3 résume ce modèle, reprenant les parties mécanique et électrique ainsi que le bloqueur et l’échantillonneur, caractérisant l’interface entre le système analogique et le PC.

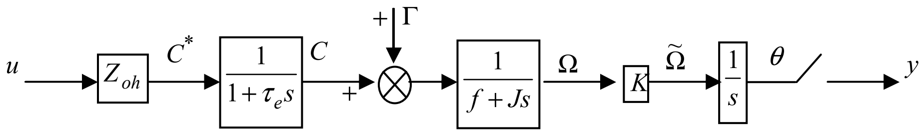



**Figure 7.3 : Modèle de la machine asynchrone. $C^*$ - consigne en couple; $C$ - couple électromagnétique ; $\Gamma$ - charge ; $\Omega$ - vitesse angulaire ; $\theta$ - position.**

Finalement, le modèle utilisé est représenté par une fonction de transfert discrète (pour une période d’échantillonnage de $T_e = 14 \times 76{,}6\,\mu\text{s} = 1{,}0724\text{ms}$) entre le couple électromécanique et l’écart angulaire :

$$\frac{\theta(q^{-1})}{u(q^{-1})} = \frac{q^{-1}B(q^{-1})}{A(q^{-1})} = \frac{10^{-4}(0.821q^{-1} + 0.8206q^{-2})}{(1-q^{-1})(1-0.998q^{-1})}. \tag{7.5}$$

La validation des performances des différentes lois de commande, est réalisée via un scénario ‘transitique rapide’ pour le positionnement [MB02]. La trajectoire considérée est classique pour les applications de type machine-outil ou robotique, la référence de vitesse monte jusqu’à la vitesse nominale et revient à zéro selon différents profils. La Figure 7.4 présente la trajectoire de position, la trajectoire de position filtrée effectivement suivie (filtre du deuxième ordre de coefficient d’amortissement $\zeta = 1$ et fréquence de coupure $\omega_0 = 25\text{rad/s}$). Ce cycle permet d’examiner le comportement de l’actionneur à basses vitesses (pour $t \in [2{,}6s;4{,}6s]$, la position de référence varie entre $20\pi$ et $21\pi$ radians avec une vitesse de référence de $\pi/2$ rad/s), à vitesse nulle avec changement du couple de charge, à vitesse nominale. Une autre point sensible est la poursuite d’une trajectoire de position non linéaire (dans ce cas parabolique). L’état magnétique de la machine est maintenu à sa valeur nominale.

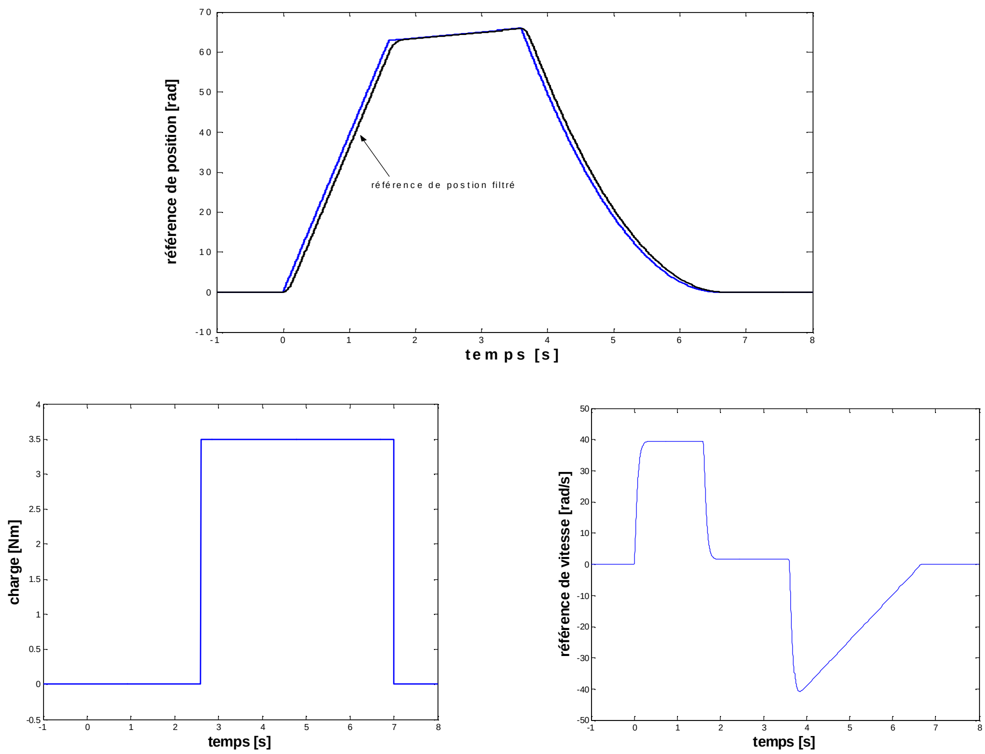



**Figure 7.4 : Scénario pour le problème de positionnement de la machine asynchrone a) Position b) Couple de charge c) Vitesse**

## 7.3 *Commande linéaire*

En se basant sur le modèle de la machine asynchrone décrit précédemment, des stratégies de commande classiques ou avancées peuvent être mises en place. Ce paragraphe présente les lois de commande qui peuvent être implémentées par des régulateurs de type RST (Figure 7.5).

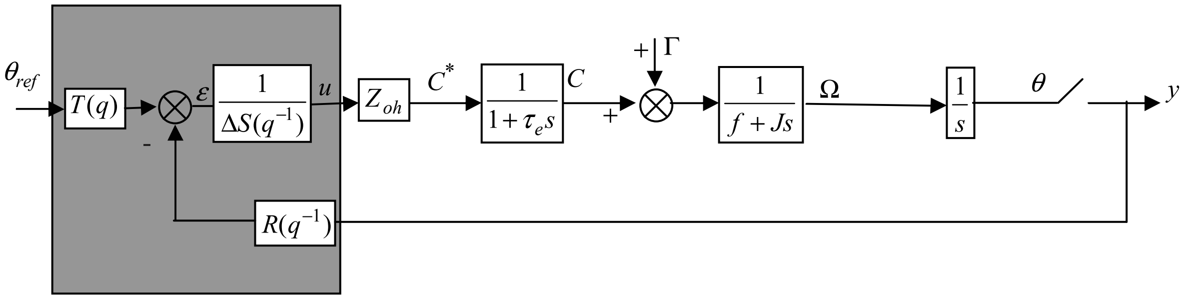



**Figure 7.5 : Schéma bloc pour la commande basé sur des régulateurs RST**

Notons que la restriction à cette classe de lois de commande, de faible complexité, implique que la famille des contraintes qui peuvent être prises en compte lors de la synthèse des algorithmes prédictifs est restreint à des contraintes de type égalité.

### 7.3.1 Commande classique par régulateur PID

Ce paragraphe présente succinctement, avant l'étude des régulateurs prédictifs la stratégie de commande classiquement implantée sur ce banc, à base de PID (Figure 7.6), qui servira de référence lors des comparaisons à venir avec les régulateurs prédictifs.

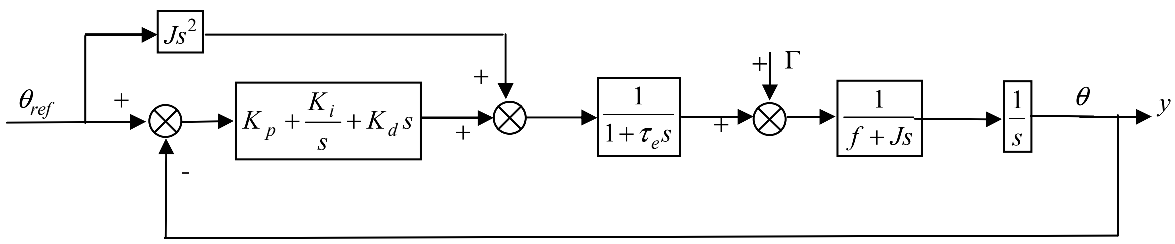



**Figure 7.6 : Schéma utilisé pour la synthèse du PID**

Notons que cette stratégie est relativement sophistiquée, puisqu'elle fait intervenir un régulateur PID avec anticipation et filtre de l'action dérivée (non représenté ici). Le choix des paramètres du régulateur PID s'effectue à partir d'un modèle simplifié pour lequel la constante de temps électrique est négligée par rapport à la constante mécanique, de même pour le terme de frottement visqueux dans le transfert mécanique. Avec ce modèle simplifié (en fait un double intégrateur), la fonction de transfert entre la perturbation de couple $\Gamma$ et la sortie $\theta$ est :

$$\frac{\theta}{\Gamma} = \frac{\frac{1}{J}s}{s^3 + \frac{K_d}{J}s^2 + \frac{K_p}{J}s + \frac{K_i}{J}} \tag{7.6}$$

qui est identifiée à l'équation caractéristique :

$$(s+\omega_0)(s^2 + 2\xi\omega_0 s + \omega_0^2) \tag{7.7}$$

soit :

$$K_p = \omega_0^2(1+2\xi)J; K_i = \omega_0^3 J; K_d = \omega_0(1+2\xi)J \tag{7.8}$$

L'objectif est ainsi de fixer la dynamique du rejet de perturbation, la dynamique de poursuite fixée par $\omega_0$ dépend en fait du terme d'anticipation. Tous les paramètres de réglage sont linéairement dépendants de l'inertie et de cette façon, la valeur de $J$ peut être utilisée comme un potentiomètre pour ajuster le PID sur le système réel. Pour $J = 0{,}007\,\mathrm{kgm}^2, \xi = 1$ et $\omega_0 = 50\,\mathrm{rad/s}$, on obtient :

$$K_p = 52{,}5\,\mathrm{Nm\,s/rad};\ K_i = 875\,\mathrm{Nm\,s/rad};\ K_d = 1{,}05\,\mathrm{Nm\,s/rad} \tag{7.9}$$

Ce régulateur PID peut être implanté de façon numérique en utilisant la transformation d'Euler avec $T_e = 1{,}0724\,\mathrm{ms}$ et un filtre pour l'action dérivée (fréquence de coupure de $1500\,\mathrm{rad/s}$). Pour faciliter l'analyse, la structure d'un tel régulateur PID peut s'exprimer sous la forme RST polynomiale équivalente :

$$\Delta S(q^{-1})C^*(t) = T(q^{-1})\theta_{ref}(t) - R(q^{-1})\theta(t) \tag{7.10}$$

donnant ici :

$$\begin{aligned} S(q^{-1}) &= 1 + 1{,}217q^{-1} + 0{,}370q^{-2} \\ R(q^{-1}) &= 2586{,}983 - 5054{,}545q^{-1} + 2489{,}435q^{-2} - 19{,}446q^{-3} \\ T(q^{-1}) &= 7065{,}84q - 11564{,}44 - 2740{,}86q^{-1} + 9798{,}46q^{-2} - 302{,}09q^{-3} - 2254{,}49q^{-4} \end{aligned} \tag{7.11}$$

Le polynôme $T$ contient un terme non causal dû au terme d'anticipation utilisé lors de la synthèse du régulateur PID (Figure 7.6).

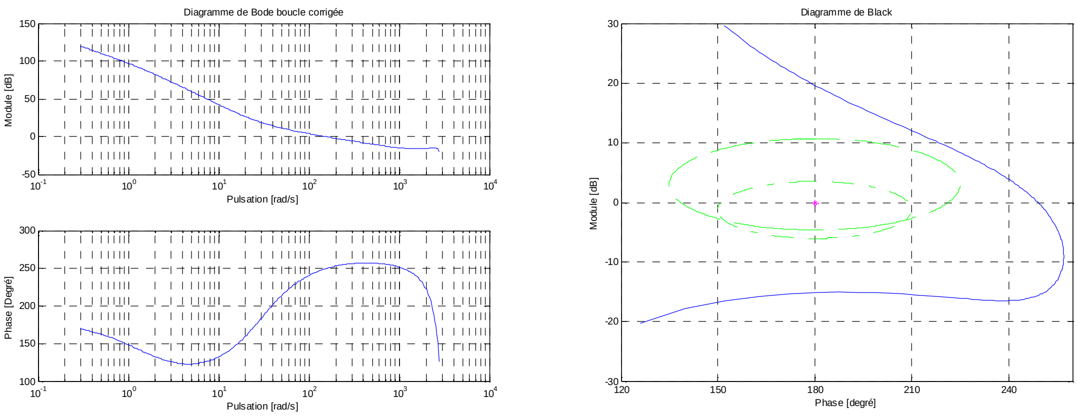



**Figure 7.7 Diagramme de Bode et de Black de la boucle ouverte corrigée par le correcteur PID**

La Figure 7.8 illustre les résultats obtenus avec ce correcteur PID implémenté dans le simulateur non linéaire. Si l'erreur est acceptable, on observe que le bruit induit sur la mesure de la position, se fait ressentir de façon sensible sur la commande en couple.

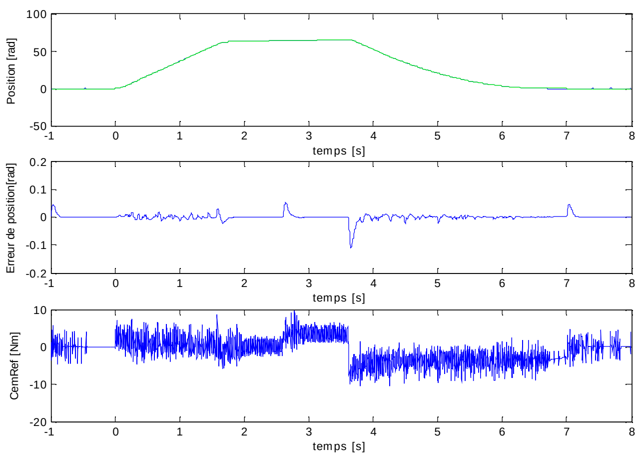



**Figure 7.8 : Réponses temporelles pour le correcteur PID décrit par (7.11)**
**(de haut en bas : Position, erreur, couple)**

On a choisi pour le régulateur PID présenté ci-dessus une période d'échantillonnage $T_e = 1{,}0724\text{ms}$ pour garder la même formulation que celle utilisée par la suite pour la synthèse prédictive, et qui nécessite un horizon plus large de prédiction. Néanmoins, le correcteur PID peut fonctionner à des périodes d'échantillonnage plus faibles allant jusqu'à $T_e = 76{,}6\mu\text{s}$, la période de l'onduleur. En effet, en choisissant cette valeur limite de période d'échantillonnage, les performances de l'asservissement de position sont améliorées (Figures 7.9 et 7.10). Dans les études effectuées par [Rod03] on peut retrouver d'autres choix pour le filtrage de la position ($T_e = 153{,}2\mu\text{s}$).

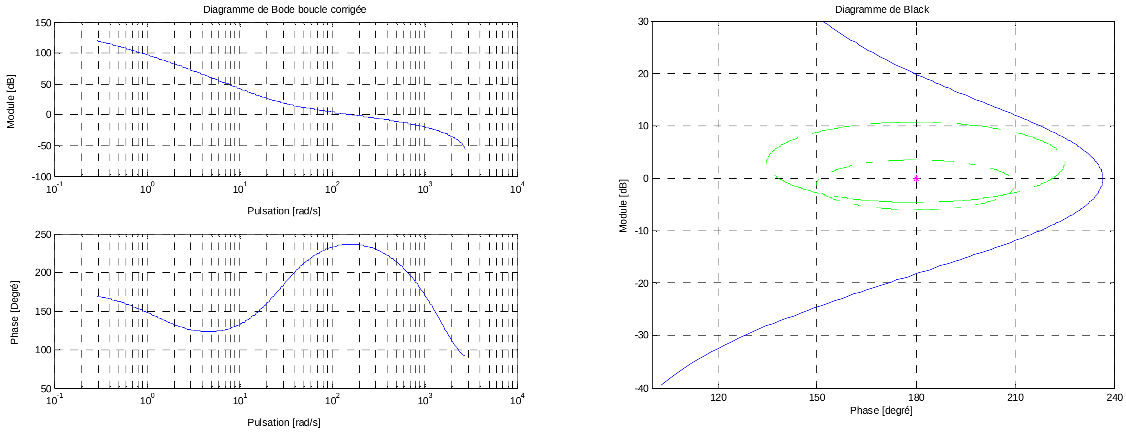



**Figure 7.9 : Diagrammes de Bode et de Black de la boucle ouverte corrigée par PID, $T_e = 76{,}6\mu s$**

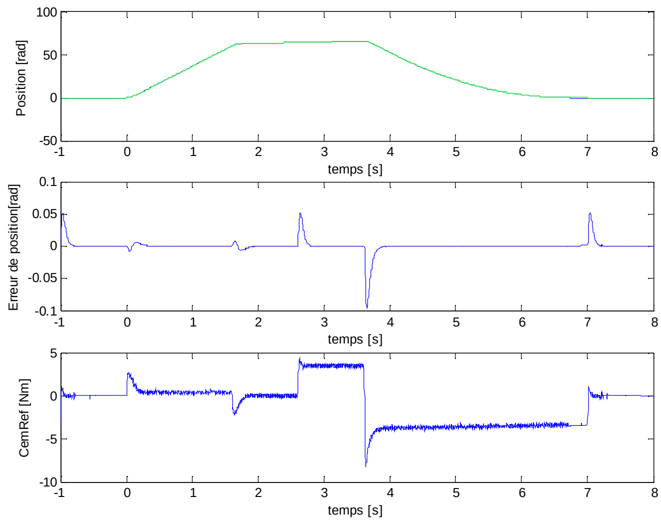



**Figure 7.10 : Réponses temporelles pour le correcteur PID avec $T_e = 76{,}6\mu s$ (de haut en bas : Position, erreur, couple).**

### 7.3.2 Commande prédictive sans contraintes

L'application de la loi de commande basée sur le régulateur classique PID montre les difficultés liées au choix des éléments d'anticipation et des différents paramètres de façon optimale. L'objectif des développements suivants est de montrer que ces desiderata peuvent être réunis au sein d'un problème de commande prédictive proposant un régulateur numérique sous forme polynomiale RST.

Partons du modèle du système à commander donné par la Figure 7.3, pour lequel la partie électrique est négligée par rapport à la partie mécanique décrite par $J = 0{,}007\text{kgm}^2$ et $f = 0{,}01\text{Nm(rad/s)}^{-1}$, aboutissant au modèle (7.5) pour un période d'échantillonnage $T_e = 1{,}0724\text{ms}$. En considérant les règles de stabilité et de robustesse données dans [BD96], un correcteur GPC peut être synthétisé avec le jeu de paramètres :

$$N_1 = 1; N_2 = 16, N_u = 1; \lambda = 0{,}0001 \tag{7.12}$$

On obtient les polynômes :

$$
\begin{aligned}
&S(q^{-1}) = 1 + 0{,}508q^{-1} \\
&R(q^{-1}) = 7087{,}442 - 13157{,}886q^{-1} + 6141{,}279q^{-2} \\
&T(q) = 0{,}047q + 0{,}19q^2 + 0{,}43q^3 + 0{,}76q^4 + 1{,}19q^5 + 1{,}71q^6 + 2{,}32q^7 + \\
&\qquad + 3{,}03q^8 + 3{,}84q^9 + 4{,}74q^{10} + 5{,}73q^{11} + 6{,}82q^{12} + 8q^{13} + 9{,}27q^{14} + \\
&\qquad + 10{,}63q^{15} + 12{,}09q^{16}
\end{aligned}
\tag{7.13}
$$

Les caractéristiques fréquentielles de la boucle ouverte corrigée avec ce régulateur sont données Figure 7.11.

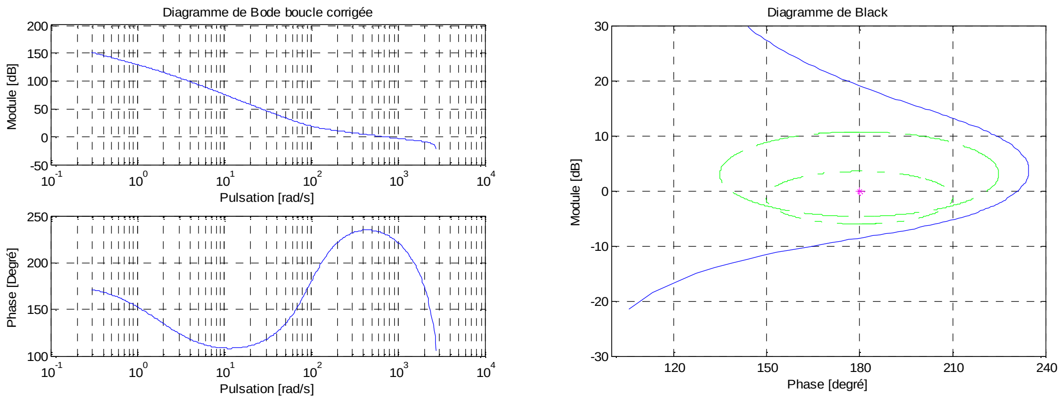



**Figure 7.11 : Diagramme de Bode et de Black de la boucle ouverte corrigée avec le correcteur GPC**

L'application de la loi prédictive pour le positionnement de la machine asynchrone confirme les bonnes performances en poursuite, même en présence de variation de la charge (Figure 7.12).

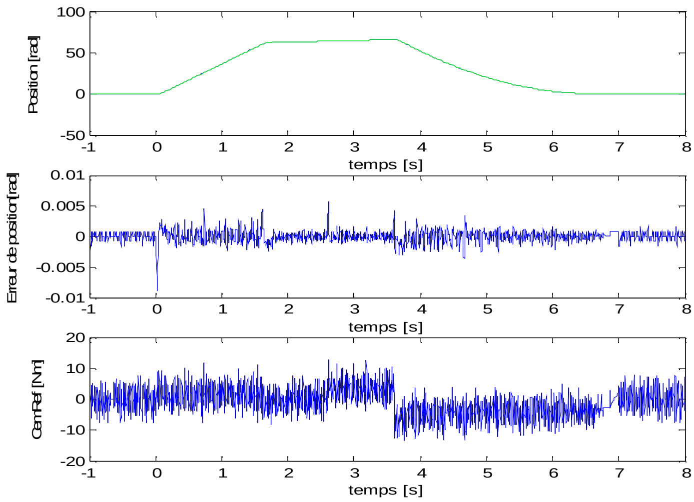



**Figure 7.12 : Réponses temporelles pour le correcteur GPC**
**(de haut en bas : Position, erreur, couple)**

On observe néanmoins que les bruits de mesure se répercutent fortement sur le signal de commande, induisant des variations autour des valeurs utiles de la commande qui peuvent endommager les actionneurs. Une solution á ce problème est offerte par la stratégie de robustification basée sur la paramétrisation Youla, présentée au chapitre 6. Sachant que l'on possède un régulateur GPC initial sous forme RST, l'application de la méthode présentée dans [Rod03] ne soulève aucune difficulté. La robustification impose de choisir un gabarit acceptable pour le rejet d'une perturbation de type échelon (correspondant à la charge du moteur pour notre application) et fournit au final le paramètre de Youla optimal. Pour un gabarit présenté Figure 7.13, on obtient le paramètre sous la forme d'une fonction de transfert :

$$
Q(q^{-1}) = \frac{-4824 + 2341q^{-1}}{1 - 0{,}9532\, q^{-1} + 0{,}4507\, q^{-2}} \tag{7.14}
$$

Ensuite, en appliquant les relations (6.8), on définit parmi la famille des régulateurs stabilisants, un régulateur RST moins sensible aux bruits de mesure et qui préserve de plus le comportement entrée-sortie :

$$
\begin{aligned}
&S(q^{-1}) = 1 - 0{,}049q^{-1} + 0{,}172q^{-2} + 0{,}0357q^{-3} \\
&R(q^{-1}) = 2263{,}19 - 3109{,}48q^{-1} + 405{,}99q^{-2} - 43{,}84q^{-3} + 431{,}7q^{-4} \\
&T(q^{-1}) = 0{,}021q + 0{,}04q^{2} + 0{,}06q^{3} + 0{,}125q^{4} + 0{,}24q^{5} + 0{,}4q^{6} + 0{,}6q^{7} + \\
&\qquad + 0{,}9q^{8} + 1{,}16q^{9} + 1{,}5q^{10} + 1{,}9q^{11} + 2{,}35q^{12} + 2{,}83q^{13} + 3{,}37q^{14} + \\
&\qquad + 3{,}95q^{15} + 4{,}58q^{16} + 0{,}89q^{17} + 12{,}09q^{18}
\end{aligned}
\tag{7.15}
$$

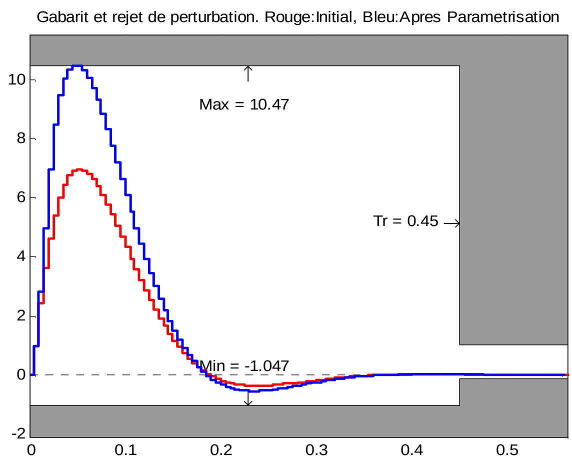



**Figure 7.13 : Gabarit utilisé lors de la robustification du correcteur GPC**

Une analyse fréquentielle (Figure 7.14) permet de vérifier que des caractéristiques satisfaisantes de la boucle ouverte corrigée avec le nouveau correcteur sont préservées. Il faut tout de même mentionner qu'un compromis entre le degré de robustesse face aux bruits de mesure et les caractéristiques en régulation du système corrigé doit être respecté. On a choisi ici de rester dans les limites acceptables pour les caractéristiques en régulation, on pourrait aller plus loin en imposant une robustesse renforcée par rapport au bruit de mesure et paramètres incertains [Rod03].

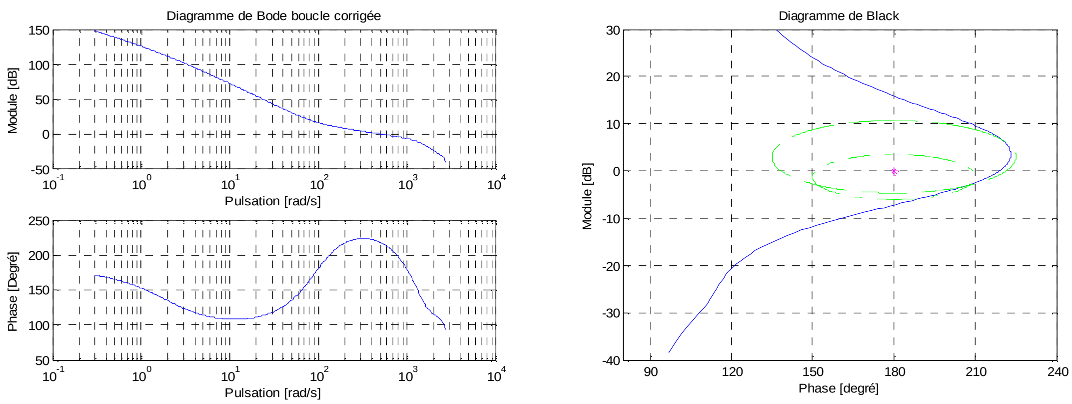



**Figure 7.14 : Diagramme de Bode et de Black de la boucle ouverte corrigée avec le correcteur GPC robustifié**

En appliquant cette loi de commande prédictive robustifiée, les résultats temporels de la Figure 7.15 mettent en évidence une erreur de poursuite toujours de niveau faible (ce qui est normal puisque le comportement entrée-sortie de la boucle fermée est préservé) et un signal de commande subissant moins l'influence du bruit de mesure (conséquence de la robustification à l'aide du paramètre de Youla).

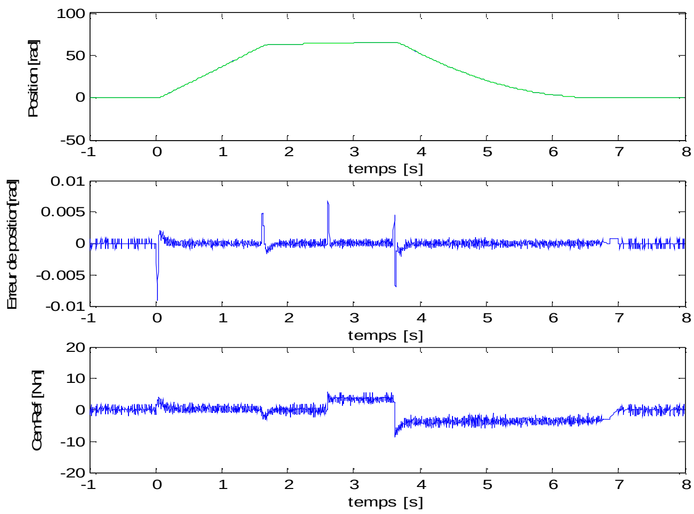



**Figure 7.15 : Réponses temporelles pour le correcteur GPC robustifié (de haut en bas : Position, erreur, couple)**

### 7.3.3 Contraintes égalité

Examinons à ce stade les procédures de synthèse des lois prédictives en présence de contraintes terminales de type égalité :

$$\begin{aligned}&\min_{\{\Delta u_t,\cdots,\Delta u_{t+N_u-1}\}} \sum_{j=N_1}^{N_2}\left[w_{t+j}-\hat{y}_{t+j}\right]^2+\lambda\sum_{j=1}^{N_u}\left[\Delta u_{t+j-1}\right]^2\\ &y_{t+j}=w_{t+N_2+j},\, N_2\le j\le N_2+N_c\end{aligned} \tag{7.16}$$

Ce type de contraintes, qui se manifeste à la fin de l'horizon de prédiction, enlève des degrés de liberté à la séquence de commande. Dès lors une règle intuitive consiste à choisir l'horizon de commande supérieur au nombre de contraintes terminales, $N_u > N_c$.

Les bonnes performances des lois prédictives synthétisées dans le cas sans contraintes se conservent également en présence de contraintes égalité. Ainsi pour $N_u = 2; N_c = 1; N_2 = 10, \lambda = \mathrm{trace}(G^T G)$ (avec $G$ matrice de réponse libre de système [BD96] intervenant dans la construction des prédicteurs optimaux) la loi polynomiale obtenue possède les caractéristiques fréquentielles décrites Figure 7.16.

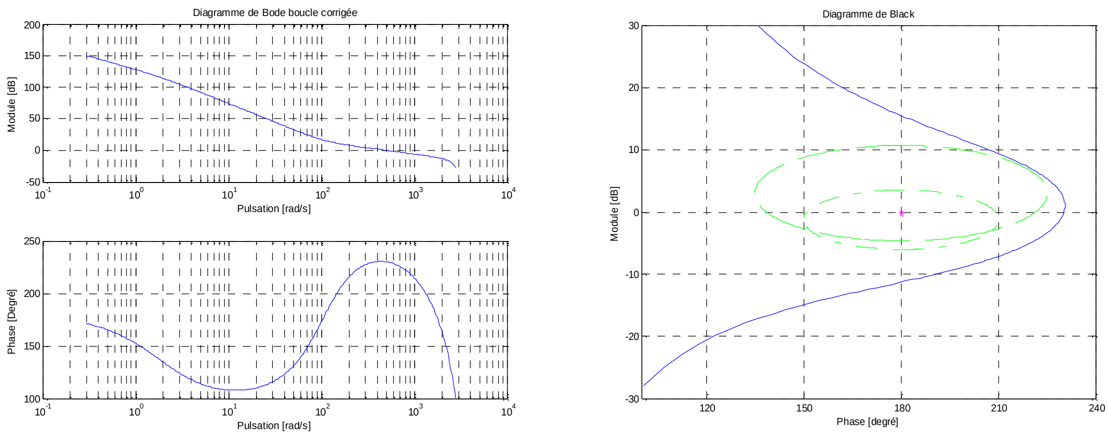



**Figure 7.16 : Diagramme de Bode et de Black de la boucle ouverte corrigée avec le correcteur GPC robustifié, contraintes égalité**

Cette loi possède une structure tout à fait similaire à celle sans contrainte (structure RST), l'implémentation sur la machine asynchrone ne pose donc aucun problème supplémentaire. Les résultats sont présentés Figure 7.17, a) étant la configuration incluant le régulateur RST proposé par la solution analytique du problème d'optimisation quadratique associé à (7.16), b) la configuration avec le régulateur robustifié selon la procédure de [Rod03] sans contraintes.

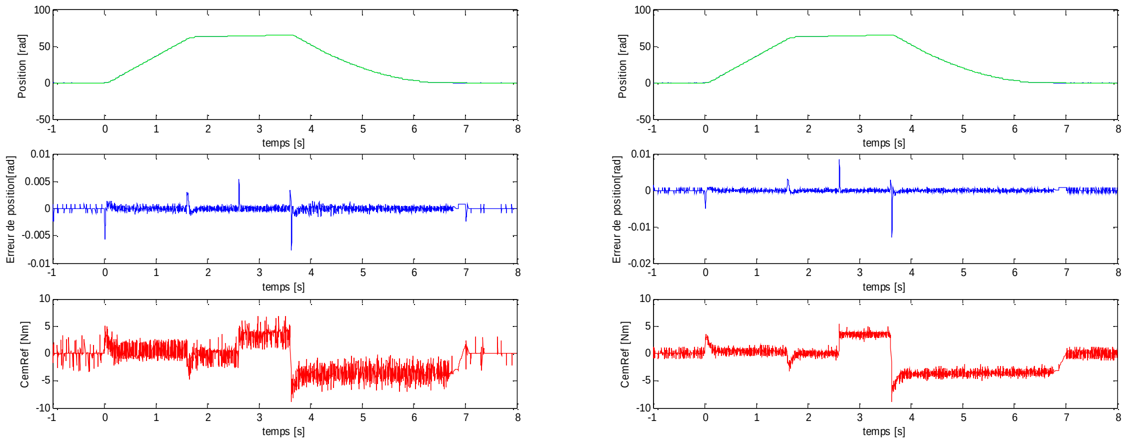



**Figure 7.17 : Réponses temporelles pour le correcteur GPC avec contraintes de type égalité**
**a) non robustifié b) robustifié (de haut en bas : Position, erreur, couple)**

On peut alors conclure de cette première prise en compte de contraintes pour les lois prédictives que les horizons de prédiction nécessaires pour garantir les performances de la boucle fermée changent. En effet, pour ce système particulier, il est possible de diminuer l'horizon de prédiction et donc de retrouver une loi RST ayant un polynôme $T$ de faible complexité. Dans le but de déterminer la limite inférieure de cet horizon, les simulations suivantes montrent que le régulateur RST obtenu avec un l'horizon de prédiction égal à 3 pas d'échantillonnage (Figure 7.18a) peut assurer un bonne poursuite mais ses caractéristiques en rejet de perturbation et vis-à-vis des bruits sur la commande sont affectées. En présence d'incertitude paramétriques (variations de l'inertie du moteur par exemple), la loi de commande pourrait s'avérer instable.

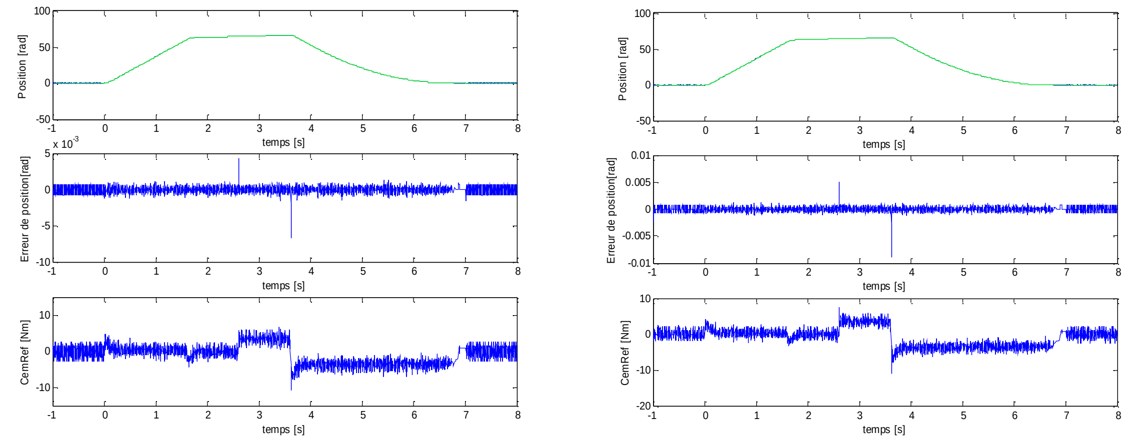



**Figure 7.18 : Réponses temporelles pour le correcteur GPC robustifié**
**(de haut en bas : Position, erreur, couple)**
**a) :** $Nu = 2; N_2 = 3; N_c = 1; \lambda = \mathrm{trace}(G^T G)$ **b) :** $Nu = 10; N_2 = 8; N_c = 2; \lambda = \mathrm{trace}(G^T G)$

D'autres choix pour l'horizon de commande peuvent être considérés, comme par exemple un horizon de commande qui couvre tout l'horizon de prédiction (Figure 7.18b). Un autre aspect qui doit être pris en compte lors de la considération de contraintes terminales est l'apparition d'erreurs statiques non nulles pour des consignes différentes de l'échelon dans le cas où le critère utilisé est de la forme :

$$
\begin{aligned}
&\min_{\{\Delta u_t, \cdots, \Delta u_{t+N_u-1}\}} \sum_{j=N_1}^{N_2} \left[w_{t+j} - \hat{y}_{t+j}\right]^2 + \lambda \sum_{j=1}^{N_u} \left[\Delta u_{t+j-1}\right]^2 \\
&y_{t+j} = w_{t+N_2}, N_2 \le j \le N_2 + N_c
\end{aligned}
\tag{7.17}
$$

En effet pour $N_u = 10; N_2 = 8; N_c = 2; \lambda = \text{trace}(G^T G)$ et avec (7.17), on obtient les résultats décrits Figure 7.19.

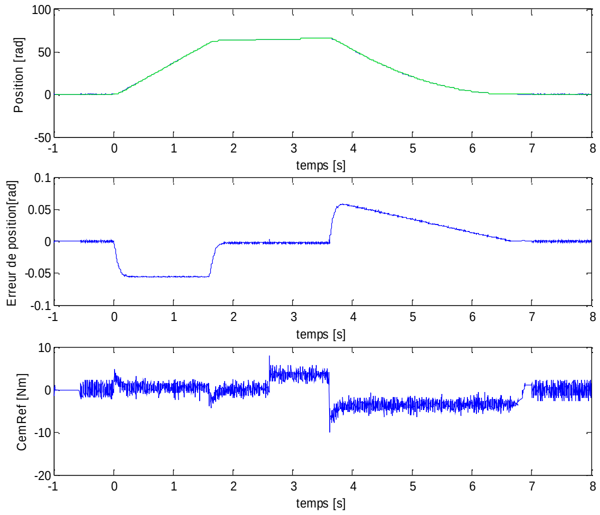



**Figure 7.19 : Réponses temporelles pour le correcteur GPC caractérisé par une erreur stationnaire non nulle**

## *7.4 Commande prédictive avec contraintes sur la commande*

Ce paragraphe se focalise désormais sur les conséquences de la présence de contraintes de type inégalité dans la formulation de la loi prédictive, en étudiant plus particulièrement l'association de contraintes sur la commande et de contraintes terminales. Cette association est réputée pour sa sensibilité au niveau des problèmes de faisabilité. En effet, l'ajout de contraintes terminales fait que les commandes appliquées au système sont plus nerveuses afin d'améliorer la dynamique de suivi.

Un autre point intéressant concerne la formulation explicite de la loi de commande dans ce cas. Dans un contexte de travail basé sur un modèle CARIMA, la loi explicite GPC est donnée par des régulateurs RST affines par morceaux.

### 7.4.1 Prise en compte de contraintes

On considère un ensemble de contraintes sur l'incrément de commande (décrit par les limitations $\Delta u_{\min} = -2{,}3$ et $\Delta u_{\max} = 2{,}3$), et sur la sortie (avec les limitations $y_{\min} = -0{,}05$ et $y_{\max} = 67$) venant s'ajouter aux contraintes terminales déjà décrites précédemment. L'ensemble des contraintes se résume donc à :

$$\begin{cases} \Delta u_{\min} \le \Delta u(t+k) \le \Delta u_{\max}, k = 1..N_u \\ y_{\min} \le \hat{y}(t+k) \le y_{\max}, k = N_1..N_2 \\ \hat{y}(t+N_2+k) = w(t+N_2+j), k = 1..N_c \end{cases} \tag{7.18}$$

soit sous forme matricielle l'ensemble des égalités et inégalités linéaires :

$$\begin{cases} \mathbf{A}_{in}\mathbf{k}_u \le \mathbf{b}_{in} + \mathbf{B}_{in}\, p(t) \\ \mathbf{A}_{eq}\mathbf{k}_u = \mathbf{b}_{eq} + \mathbf{B}_{eq}\, p(t) \end{cases} \tag{7.19}$$

paramétré par le vecteur de contexte :

$$p(t) = [\Delta u_{t-1} \ \cdots \ \Delta u_{t-n_b} \,|\, y_t \ \cdots \ y_{t-n_b} \,|\, w_{t+N_1} \ \cdots \ w_{t+N_2+N_c}]^{\mathrm{T}} \tag{7.20}$$

avec $n_a = 2$ et $n_b = 1$ comme on peut le déduire du modèle décrit par (7.5).

On peut se demander dans quelle mesure ces contraintes influencent le comportement du système bouclé. En analysant la réponse du système obtenu avec le régulateur synthétisé en ignorant les contraintes inégalité, on peut tirer une première conclusion à partir de la simulation Matlab concernant le fonctionnement de la commande prédictive pour $N_u = 2; N_1 = 1; N_2 = 16; N_c = 1; \lambda = 0.001$.

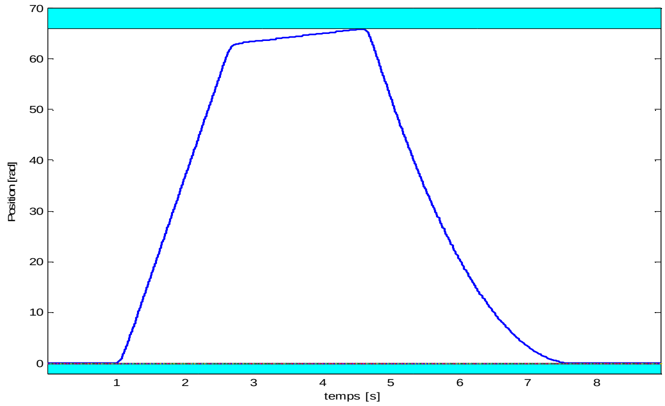



**Figure7.20 : Réponse en suivi de position (loi GPC) avec illustration des contraintes sur la sortie**

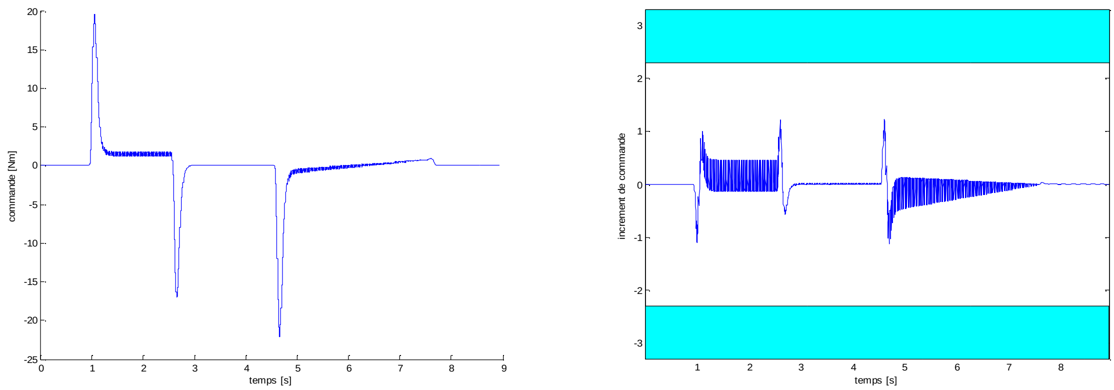



**Figure 7.21 : a) Commandes appliquées selon le principe de l'horizon glissant**
**b) Incréments de commande calculés par GPC sans tenir compte des contraintes**

Les Figures 7.20 et 7.21 illustrent le comportement du système bouclé auquel on applique la séquence optimale des deux commandes selon le principe de l'horizon glissant. On constate un comportement tout à fait satisfaisant par rapport aux contraintes, ignorées lors du calcul de la séquence optimale. En effet, les contraintes sur la sortie et l'incrément ne sont pas violées. Malgré tout, les contraintes sur les incréments de commande, même si elles paraissent inutiles, peuvent influencer le signal de commande. En fait, la cause de cette fausse impression devient claire si l'on considère la distribution des séquences optimales de commande (Figure 7.22b pour les incréments). On observe en effet que les contraintes sur l'incrément de commande ne sont effectivement pas violées pour le premier incrément, mais les contraintes correspondant au deuxième incrément dépassent la région admissible. En conclusion, les séquences de commande sont maintenues dans la région faisable en se basant sur les incréments de commande calculés et non appliqués au delà des limites admissibles.

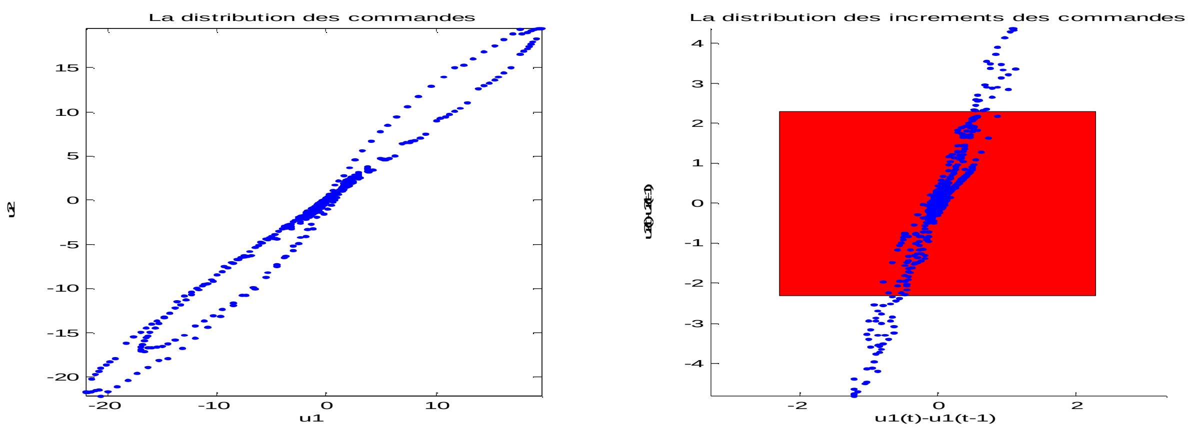



**Figure 7.22 : Distributions des séquences optimales de commande calculées par la loi GPC (sans tenir compte des contraintes), a) Commande b) Incréments de commande**

Ce comportement de la loi de commande peut s'avérer dangereux pour le fonctionnement du système bouclé et la solution idéale serait évidemment de résoudre le problème d'optimisation liée à la conception de la séquence optimale en tenant compte de l'ensemble complet des contraintes. Si l'on simule le comportement du système commandé par de telles séquences optimales appliquées selon le principe de l'horizon fuyant, on obtient un signal de commande (Figure 7.23) légèrement différent du précédent (Figure 7.21).

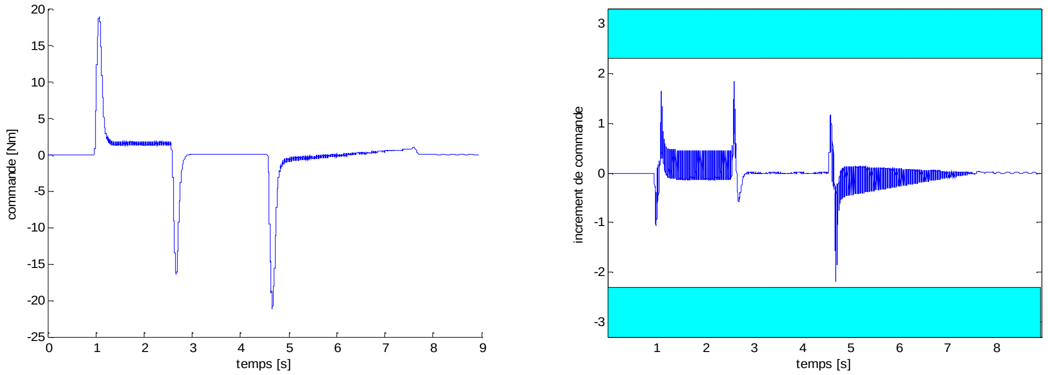



**Figure 7.23 : a) Commandes appliquées selon le principe de l'horizon glissant**
**b) Incréments de commande calculés par GPC en tenant compte des contraintes**

La distribution des séquences optimales à la base de l'application effective de la loi prédictive montre que les contraintes sont désormais prises en compte (Figure 7.24), comparée à la Figure 7.22.

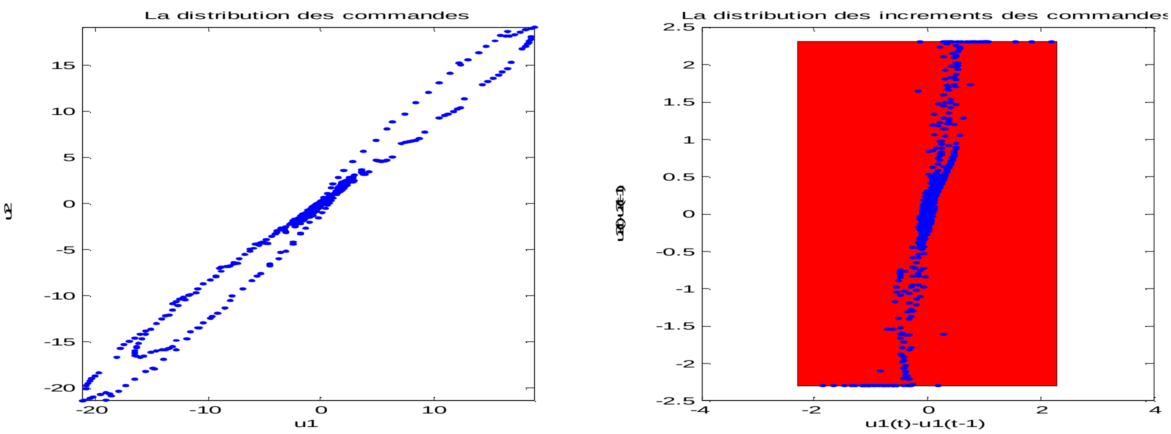



**Figure 7.24 : Distributions des séquences optimales de commande calculées par la loi GPC (en tenant compte des contraintes), a) Commande b) Incréments de commande**

Cette analyse a montré que l'utilisation de la solution du problème d'optimisation complet (en considérant l'ensemble de contraintes (7.18)) est préférable pour garantir la validité des contraintes sur tout l'horizon de prédiction. Le paragraphe suivant s'intéresse alors à la rechercher des paramètres de la loi prédictive qui préservent la faisabilité, et examine d'autre part les moyens permettant d'implémenter effectivement une loi prédictive avec contraintes.

### 7.4.2 Faisabilité

L'analyse précédente a montré que la faisabilité du problème d'optimisation résolu à chaque pas d'échantillonnage constitue l'un des aspects sensibles lors de l'implémentation de la loi prédictive. Les paramètres (principalement les horizons de commande et de prédiction, et les contraintes terminales) doivent être choisis afin d'éviter l'infaisabilité.
L'analyse géométrique apporte également des informations au sujet de la topologie du domaine polyédral des commandes admissibles. Ainsi, les contraintes sur les incréments de commande restreignent le domaine faisable à une hypercube à $2^{N_u}$ sommets indépendants des paramètres de contexte (car ils ne dépendent pas de l'évolution de la dynamique du système). Le domaine faisable est ensuite construit par intersection de cet hypercube avec les contraintes telles que celles décrites par les limitations sur la sortie. L'impact de ces contraintes sur le domaine faisable est en relation directe avec les valeurs des paramètres de contexte, car ils

influencent la prédiction. Pour illustrer cette influence, considérons le cas de la commande GPC avec les horizons $N_u = 3$, $N_1 = 1$, $N_2 = 16$. Evaluons ensuite deux cas extrêmes des paramètres de contexte : l'état stationnaire initial quand la commande et la sortie ont des valeurs proches de zéro et le paramètre de contexte correspondant à une sortie stabilisée autour de 67 rad (ce qui représente la limitation supérieure pour le scénario de trajectoire considéré). Les polyèdres correspondants sont présentés Figure 7.25. On constate que les combinaisons extrêmes des commandes sont affectées, mais sans provoquer de problèmes d'infaisabilité. Par exemple pour la valeur stationnaire zéro, l'application de trois incréments négatifs consécutifs peut induire une réponse qui viole la limitation inférieure sur la sortie.

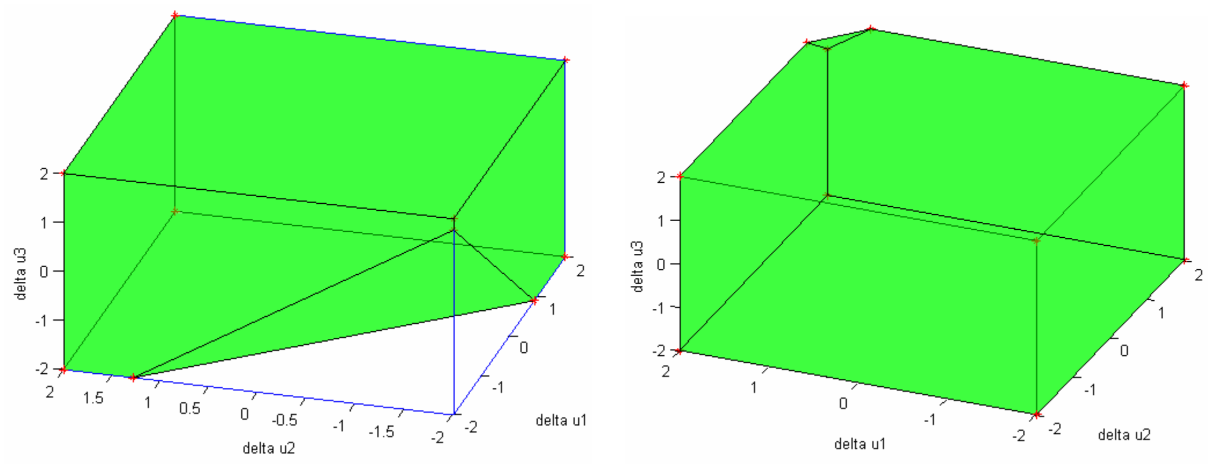



**Figure 7.25 : Domaines polyédraux faisables. a) pour une position stable autour de zéro b) pour des positions stationnaires autour des limites supérieures admissibles**

Quand les contraintes terminales égalité sont prises en compte, certains degrés de liberté dans le choix des incréments de commande n'existent plus comme on peut le constater Figure 7.26, qui décrit les domaines faisables pour les paramètres de contexte pour les situations mentionnées précédemment, mais auxquels on a ajouté des contraintes terminales. Pour une seule contrainte égalité, le domaine faisable est réduit à un polytope de dimension 2, puis pour deux égalités à un segment.

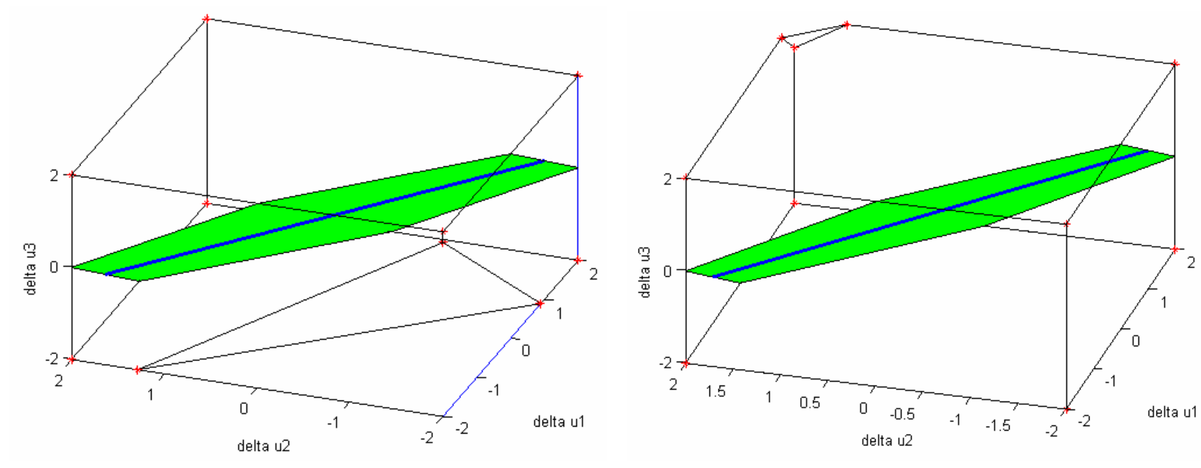



**Figure 7.26 : Domaines faisables après l'ajout des contraintes terminales**

> **Remarque 7.1 :** On peut observer que l'ajout des contraintes terminales en respectant la relation $N_u = N_c + 1$ fait que les contraintes sur la sortie deviennent redondantes. Elles peuvent être alors ignorées dans l'analyse de faisabilité et dans la formulation explicite de la loi prédictive.

Les développements suivants visent à déterminer les horizons de prédiction de commande assurant la faisabilité, tout en considérant $N_c = N_u - 1$ contraintes terminales, garantissant que le domaine faisable est un polyèdre paramétré pour lequel seuls deux sommets paramétrés restent valides à chaque pas d'échantillonnage.

Comme il a déjà mentionné au chapitre 5, la faisabilité doit être analysée en relation avec la consigne à suivre. Pour le scénario 'transitique rapide' envisagé, la consigne est connue à l'avance et l'on peut identifier les zones induisant une infaisabilité potentielle, Figure 7.27. Ainsi, ayant considéré des contraintes sur les incréments de commande, les transitions de la zone A vers la zone B et de la zone B vers la zone C peuvent conduire à des situations d'infaisabilité.

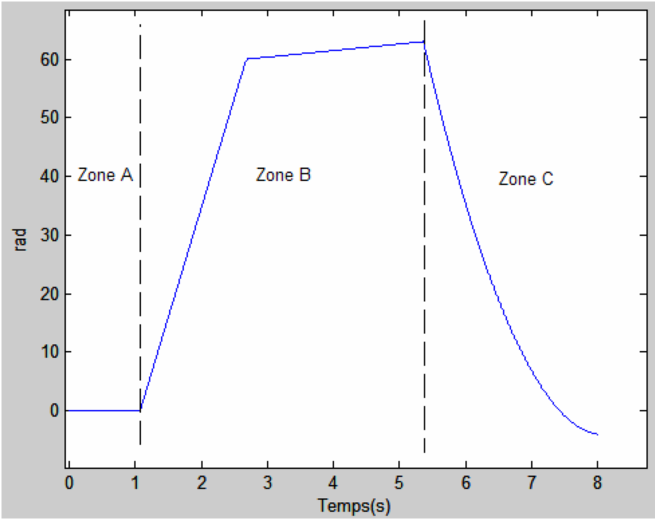



**Figure 7.27 : Partitionnement de la consigne à suivre en zones pour lesquelles la transition peut provoquer des situations d'infaisabilité**

Le modèle étant à déphasage minimal, la résolution des problèmes de programmation linéaires pour les transitions peut déterminer les familles de consignes susceptibles d'être suivies. Ces familles se caractérisent par les pentes (dérivée première) admissibles pour le signal de référence. Cette démarche a fourni les résultats présentés Figure 7.28 pour des jeux de paramètres tels que $N_c = 1 \cdots 3$ et $N_2 = 1 \cdots 39$. $N_u$ n'a pas été utilisé comme paramètre de réglage étant fixé à $N_u = N_c + 1$.

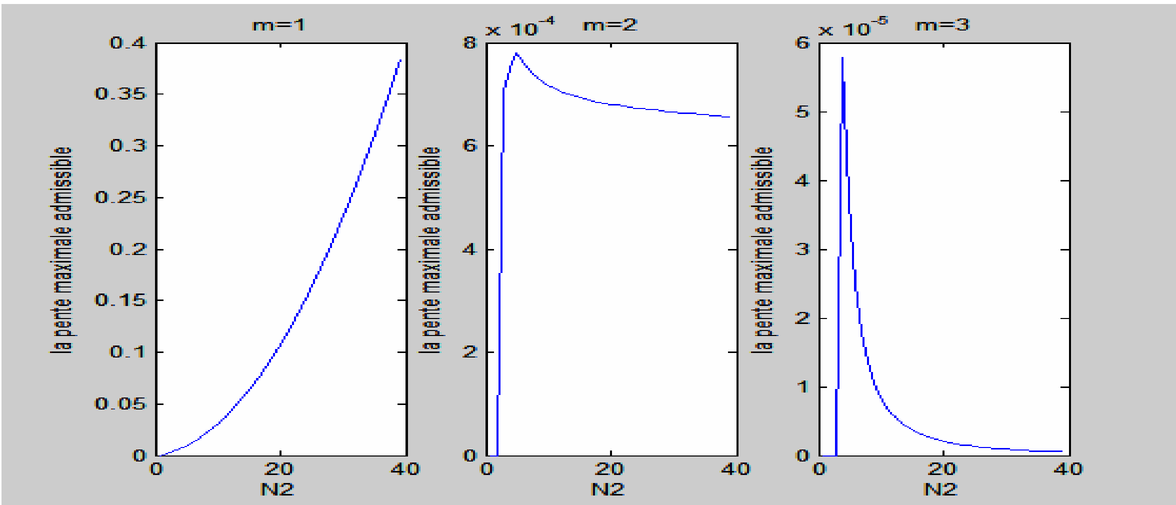



**Figure 7.28 : Pentes maximales admissibles pour le signal de référence lors du passage de la zone A vers la zone B (Nc=1..3; N2=1..39)**

La conclusion pour notre application nécessitant un passage de la zone A vers la zone B avec une pente de $0{,}04$ rad/pas d'échantillonnage est que seul le choix $N_c = 1$ peut assurer la faisabilité. Cette restriction en ce qui concerne les contraintes terminales est normale si l'on considère leur incompatibilité avec les contraintes implicites $\Delta u(t + N_u + k) = 0, k > 0$, surtout pour le suivi d'une consigne rampe. Par ailleurs, la stabilité peut être assurée en imposant $N_c > \deg\text{ré}(A)$ ou avec $N_2$ plus grand que le temps d'établissement. Le premier choix étant écarté pour cause d'infaisabilité, il faut alors s'orienter vers un horizon de prédiction plus grand.

Si l'on poursuit l'analyse de la consigne à suivre, on retrouve dans la zone B un point pour lequel la pente change sans engendrer de problème d'infaisabilité car le changement se fait vers un effort de commande moins important. En ce qui concerne le passage de la zone B vers la zone C (ce dernier passage correspond à une erreur stationnaire non nulle du fait de l'allure parabolique), la faisabilité est jugée en terme d'une borne inférieure pour la trajectoire de consigne désirée. En effet si l'on considère la dérivée première de la parabole qui décrit la zone C et si l'on assure qu'il existe une pente admissible ayant une inclinaison plus grande que celle de la dérivée, alors on peut conclure que le suivi de la parabole est garanti.

En choisissant $N_c = 1$ et $N_2 > 13$, la pente satisfait les conditions de faisabilité pour la partie non linéaire de la trajectoire à poursuivre comme on peut l'observer Figure 7.29. Les horizons de prédiction $N_2 = \{15;16\}$ seront sélectionnés désormais afin de préserver une marge de faisabilité. Globalement, cette partie d'analyse de faisabilité peut se résumer par le choix des paramètres suivants pour la loi GPC :

$$N_c = 1; N_u = 2; N_1 = 1; N_2 = \{15;16\} \tag{7.21}$$

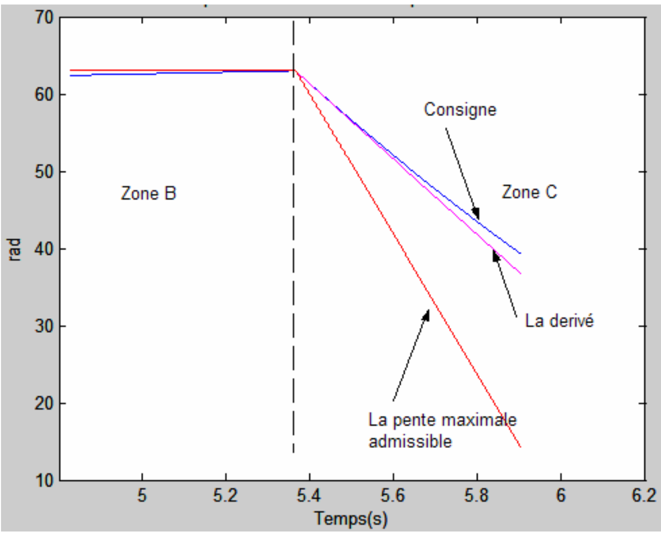



**Figure 7.29 : Pentes pour la partie décroissante de la trajectoire**

Une fois ce jeu de paramètres choisi, il est intéressant d'examiner la famille de consignes autour de la consigne donnée qui peut être suivie, tout en restant faisable du point de vue de l'optimisation en ligne. Une façon d'illustrer toutes ces trajectoires possibles consiste à recourir à une simulation permettant à chaque pas d'échantillonnage de calculer, en plus de la sortie du système, les valeurs des consignes qui peuvent être atteintes en respectant les contraintes de type égalité à la fin de l'horizon de prédiction. Autrement dit, on suit la trajectoire initiale mais on calcule aussi les déviations locales autorisées pendant l'évolution.

La Figure 7.30 illustre cette simulation. On y retrouve la trajectoire initiale et un certain intervalle 'de confiance' pour lequel la loi GPC peut garantir les contraintes terminales. Cet intervalle est finalement assez réduit, particulièrement autour des points de transition entres zones induisant une modification de l'allure (changement de pente ou de profil). On retrouve cet aspect dans la structure des prédicteurs reportés Annexe 4. On constate en effet que les coefficients de la réponse indicielle qui déterminent la structure de la matrice **G** sont assez faibles impliquant une faible influence des incréments de commande sur la sortie.

### 7.4.3 Lois explicites

Comme dans le cas avec contraintes, la loi de commande solution d'un problème d'optimisation à chaque pas ne peut s'implanter pour ce système avec un faible pas d'échantillonnage en se basant sur les procédures itératives habituelles. En conséquence, une formulation explicite de la solution peut s'avérer utile lors de l'implémentation. Considérons pour cela le problème d'optimisation quadratique sous les contraintes suivantes :

$$\begin{cases} \Delta u_{\min} \le \Delta u(t+k) \le \Delta u_{\max}, & k = 0,1 \\ y(t+N_2+1) = w(t+N_2+1) \end{cases} \Leftrightarrow \begin{cases} \mathbf{1}\,\Delta u_{\min} \le \mathbf{k}_u(t) \le \mathbf{1}\,\Delta u_{\max} \\ A_{eq}\,\mathbf{k}_u(t) = B_{eq}\,\boldsymbol{\theta}(t) \end{cases} \tag{7.22}$$

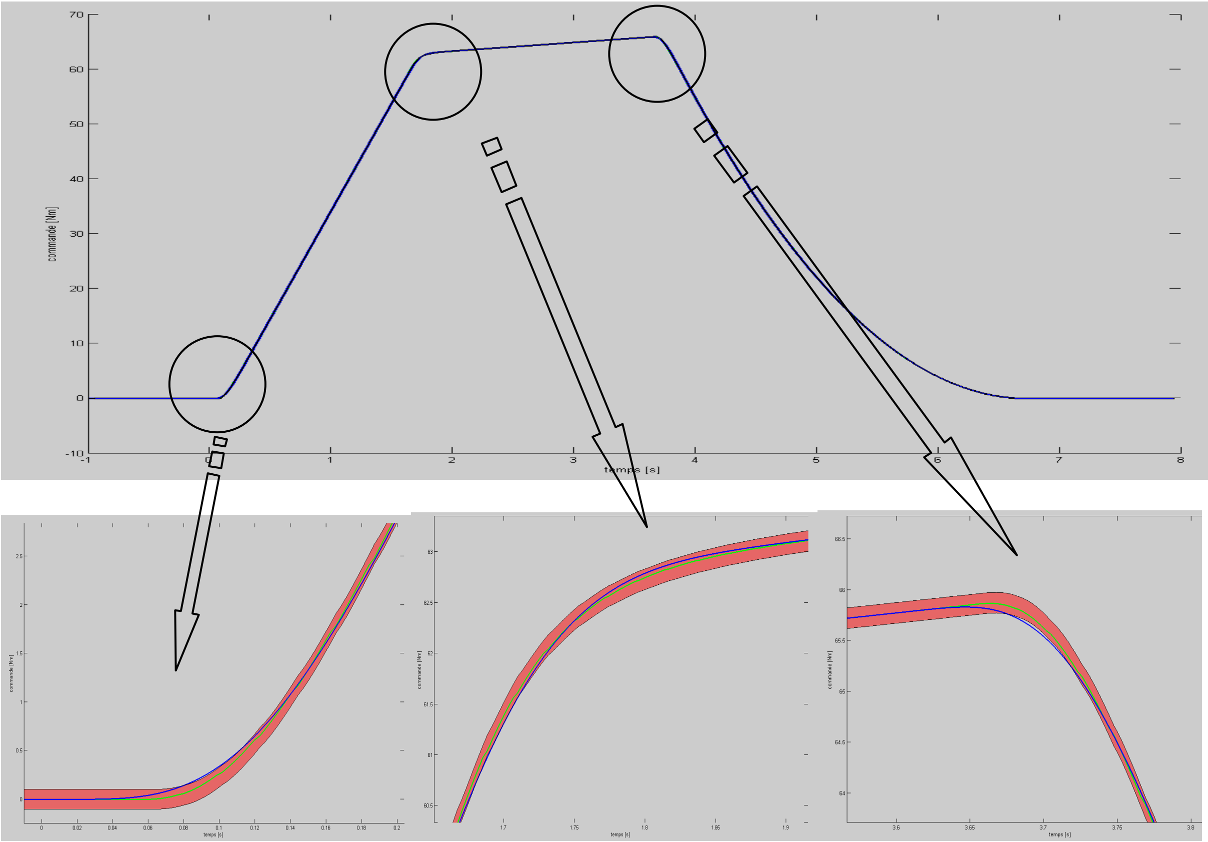



**Figure 7.30 : Détails des familles de consignes de position faisables pour le jeu de paramètres choisi**

Un horizon de prédiction $N_u = 2$ implique que l'espace de recherche pour la séquence optimale est bi-dimensionel, mais l'existence de la contrainte terminale enlève un degré de liberté. On retrouve alors un espace faisable donné par un segment (Figure 7.31a) qui sera affecté par l'évolution des paramètres de contexte. Sachant que l'on n'applique effectivement que la première valeur de la séquence optimale, et que la droite de support pour le domaine faisable dépend des paramètres, on peut exprimer de façon explicite la loi GPC à l'aide d'un polyèdre paramétré (qui sera ensuite décrit dans un espace étendu, Figure 7.31b).

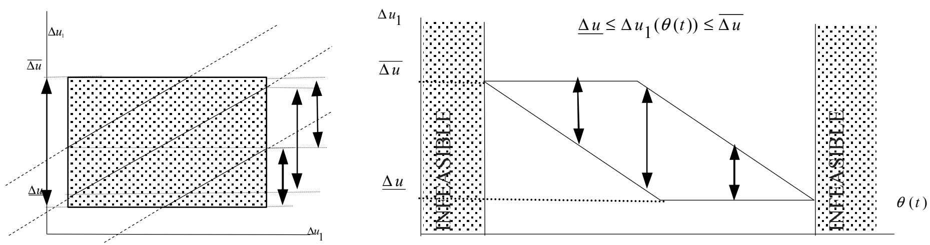



**Figure 7.31 : Intervalle de variation de $\Delta u_1$ en tant que polyèdre paramétré**

> **Remarque 7.2 :** Le phénomène d'infaisabilité est associé aux situations pour lesquelles la droite de support décrite par la contrainte égalité possède une intersection nulle avec le polyèdre matérialisé par les limitations des incréments. Des telles combinaisons de paramètres de contexte peuvent être évitées en choisissant judicieusement les paramètres de la loi prédictive selon la démarche du paragraphe précédent.

En se basant sur le fait que la valeur de la commande se trouve dans un intervalle dont les limites varient linéairement avec les paramètres, on va essayer par la suite de construire une formulation explicite de ces limitations, ce qui correspond à une implémentation de la forme :

$$\underline{\Delta u}(\mathbf{p}(t)) \leq \Delta u \leq \overline{\Delta u}(\mathbf{p}(t)) \tag{7.23}$$

Pour clarifier le mécanisme, la Figure 7.32 décrit les limitations des incréments de commande pour une simulation d'un cycle de positionnement. On observe que pour des contraintes de type $-2{,}3 \le \Delta u \le 2{,}3$ l'application effective est loin de se limiter à la simple saturation, car l'activation des contraintes pour les valeurs prédites de l'incrément de commande (mais non appliquées) se répercute sur les limitations de l'incrément de commande appliqué à l'instant *t*.

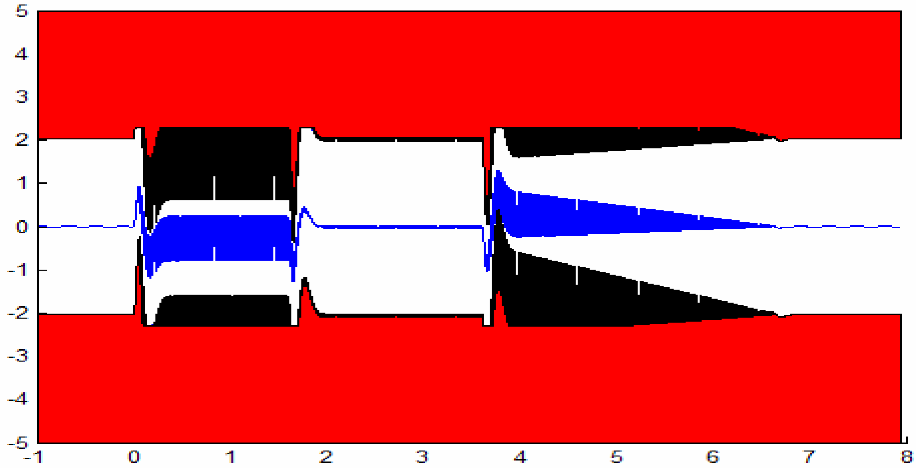


**Figure 7.32 : Incréments de commande et évolution de l'intervalle faisable durant la simulation $\underline{\Delta u}, \Delta u, \overline{\Delta u}$**

On retrouve également les implications de ces limitations sur la valeur de la commande effective, comme présenté Figure 7.33. On observe que l'intervalle faisable est assez faible, d'où une perte de performances potentielle en suivi de trajectoire, mais en même temps une limitation de la sensibilité de la commande face aux bruits de mesure.

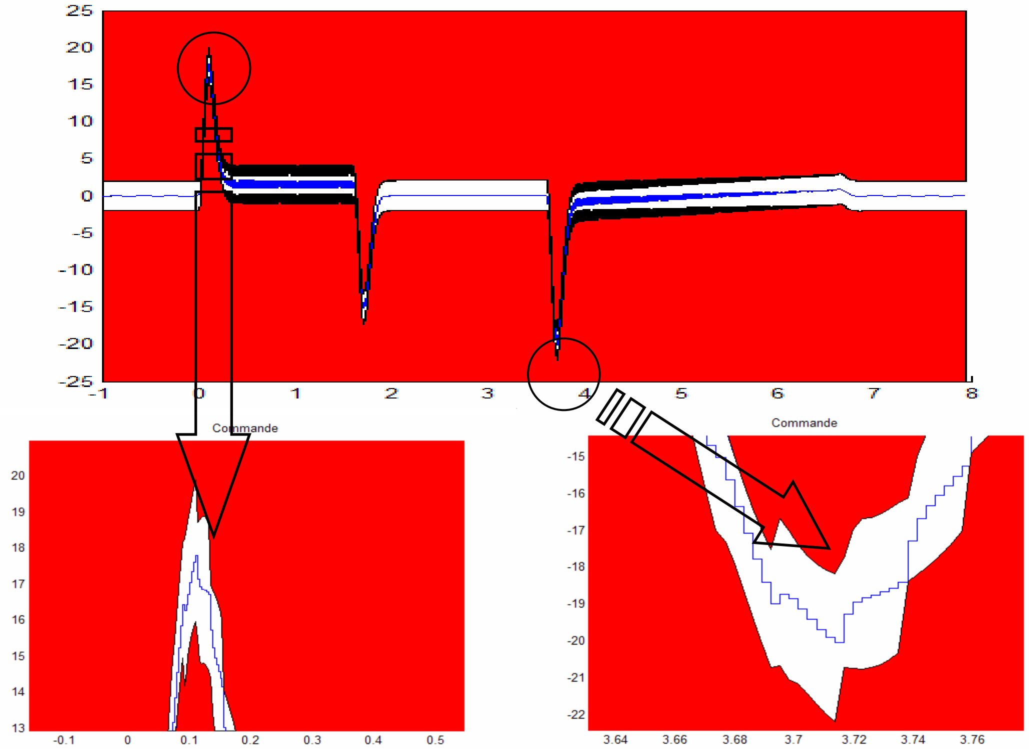



**Figure 7.33 : a) Commandes et évolution de l'intervalle faisable durant la simulation b) et c) Détails pour deux positions sensibles – départ de zéro et point de début de décroissance**

Si l'on part de la description des contraintes (7.23), la variation de la limitation en fonction des valeurs des paramètres de contexte peut être élaborée en exploitant le fait que $N_u = 2$ et donc :

$$\mathbf{k}_u^{\ *} = \{\Delta u(t), \Delta u(t+1)\} \tag{7.24}$$

En plus des saturations explicites $\Delta u_{\min} \le \Delta u(t) \le \Delta u_{\max}$, les limitations de l'incrément effectivement appliqué ( $\Delta u(t)$ ) sont données par les équations :

$$
\begin{gathered}
\underbrace{\begin{bmatrix} A_{eq} \\ 0\;1 \end{bmatrix}}_{\overline{A}} \begin{bmatrix} \overline{\Delta u}(t) \\ \overline{\Delta u}(t+1) \end{bmatrix} = \begin{bmatrix} B_{eq}\mathbf{p}(t) \\ \Delta u_{\max} \end{bmatrix} \\
\underbrace{\begin{bmatrix} A_{eq} \\ 0\;1 \end{bmatrix}}_{\overline{A}} \begin{bmatrix} \underline{\Delta u}(t) \\ \underline{\Delta u}(t+1) \end{bmatrix} = \begin{bmatrix} B_{eq}\mathbf{p}(t) \\ \Delta u_{\min} \end{bmatrix}
\end{gathered}
\tag{7.25}
$$

car le système est à déphasage minimal (et donc le premier coefficient de $A_{eq}$ est positif). Ainsi, en se basant sur le fait que l'inverse de $\overline{A}$ garde une structure donnée par : $\overline{A}^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix}$, on peut écrire :

$$
\begin{aligned}
\underline{\Delta u}(t) &= a_{11}\mathbf{p}(t) + a_{12}\Delta u_{\min} = \\
&= \underbrace{a_{11}w(t+N_2+1)}_{T(q)w(t)} - \underbrace{a_{11}if_c(q^{-1})y(t)}_{R(q^{-1})y(t)} - \underbrace{a_{11}ih_c(q^{-1})\Delta u(t-1)}_{\tilde{S}(q^{-1})\Delta u(t)} + \underbrace{a_{12}\Delta u_{\min}}_{ct} \\
\overline{\Delta u}(t) &= a_{11}\mathbf{p}(t) + a_{12}\Delta u_{\max} = \\
&= \underbrace{a_{11}w(t+N_2+1)}_{T(q)w(t)} - \underbrace{a_{11}if_c(q^{-1})y(t)}_{R(q^{-1})y(t)} - \underbrace{a_{11}ih_c(q^{-1})\Delta u(t-1)}_{\tilde{S}(q^{-1})\Delta u(t)} + \underbrace{a_{12}\Delta u_{\max}}_{ct}
\end{aligned}
\tag{7.26}
$$

Il est intéressant d'observer que ces quantités ne diffèrent que par leur partie affine, il en résulte une évaluation simplifiée dès lors que la partie linéaire est calculée une seule fois à partir d'une formulation polynomiale RST.

> **Remarque 7.3 :** Dans le cas général, la matrice $\overline{A}$ peut être singulière, dans ce cas les limitations correspondantes doivent être ignorées puisque la droite de support pour $\mathbf{k}_u$ n'a pas d'intersection avec les limites de la région faisable.

Une fois la lois sans contraintes obtenue, ainsi que la formulation par des polynômes RST des limitations $\underline{\Delta u}(t), \overline{\Delta u}(t)$ la procédure en ligne se contente de vérifier si :

$$
\max(\underline{\Delta u}(t), \Delta u_{\min}) \le \Delta u(t) \le \min(\overline{\Delta u}(t), \Delta u_{\max}) \tag{7.27}
$$

et si les limites sont dépassées, de remplacer l'incrément effectivement appliqué par la limite atteinte.

Pour le modèle de la machine asynchrone et les paramètres GPC (7.21) et $N_2 = 16$, on obtient :

$$
\begin{aligned}
&S(q^{-1})\Delta u(t) = T(q)w(t) - R(q^{-1})y(t) \\
&\overline{\Delta u}(t) = -\tilde{S}_c(q^{-1})\Delta u(t) + T(q)w(t) - R(q^{-1})y(t) + tl \cdot \Delta u_{\max}(t) \\
&\underline{\Delta u}(t) = -\tilde{S}_c(q^{-1})\Delta u(t) + T(q)w(t) - R(q^{-1})y(t) + tl \cdot \Delta u_{\min}(t)
\end{aligned}
\tag{7.28}
$$

avec :

$$
\begin{aligned}
&S(q^{-1}) = 1 + 0{,}297q^{-1} \\
&R(q^{-1}) = 4042{,}36 - 7625{,}35q^{-1} + 3607{,}69q^{-2} \\
&T(q^{-1}) = 0{,}012q + 0{,}036q^2 + 0{,}057q^3 + 0{,}075q^4 + 0{,}089q^5 + 0{,}099q^6 + 0{,}107q^7 + 0{,}111q^8 + \\
&+ 0{,}112q^9 + 0{,}109q^{10} + 0{,}103q^{11} + 0{,}094q^{12} + 0{,}082q^{13} + 0{,}066q^{14} + 0{,}047q^{15} + 0{,}025q^{16} + 23{,}48q^{17}
\end{aligned}
$$

et :

$$\begin{aligned}
&\widetilde{S}_c(q^{-1}) = 0{,}529q^{-1} \\
&R_c(q^{-1}) = 7201{,}63 - 13593{,}98q^{-1} + 6434{,}94q^{-2} \\
&T_c(q^{-1}) = 42{,}59q^{17} \\
&tl = 1{,}1281
\end{aligned}$$

La Figure 7.34 présente les résultats obtenus lors de l'application de la loi prédictive avec contraintes en utilisant le mécanisme de faible complexité décrit ci-dessus. La version sans contraintes est donnée à titre de comparaison. On observe une dégradation des performances en suivi lors de la prise en compte des contraintes, mais en contrepartie l'activation des contraintes a permis la diminution de la sensibilité de la commande. La Figure 7.35 présente les incréments de commande calculés à chaque pas d'échantillonnage, en confirmant la satisfaction des contraintes. Cette loi de commande GPC a été implantée sur le système réel donnant les résultats de la Figure 7.36.

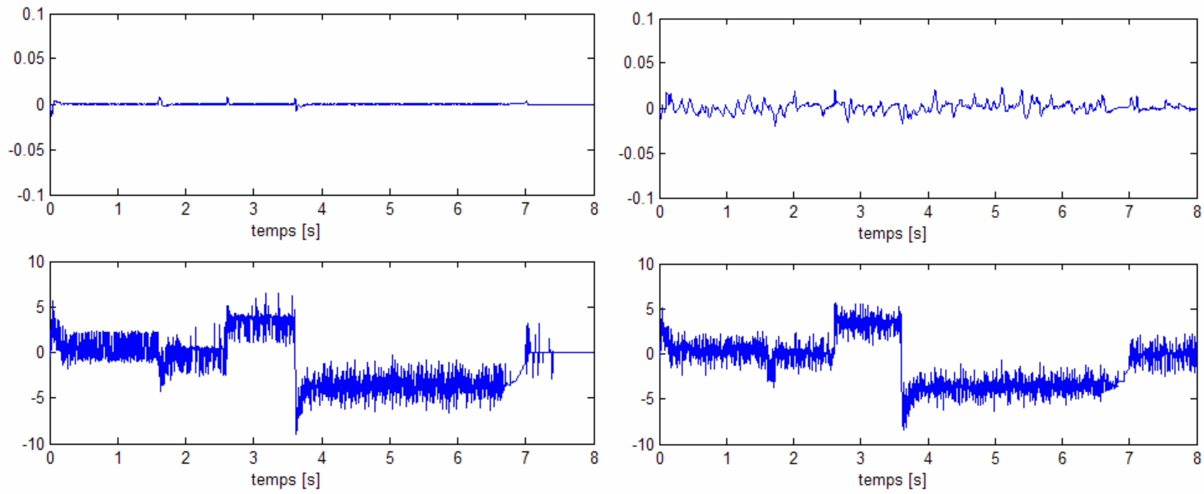



**Figure 7.34 : a) Erreur et commande pour le cas sans contraintes**
**b) Erreur et commande pour le cas avec contraintes**

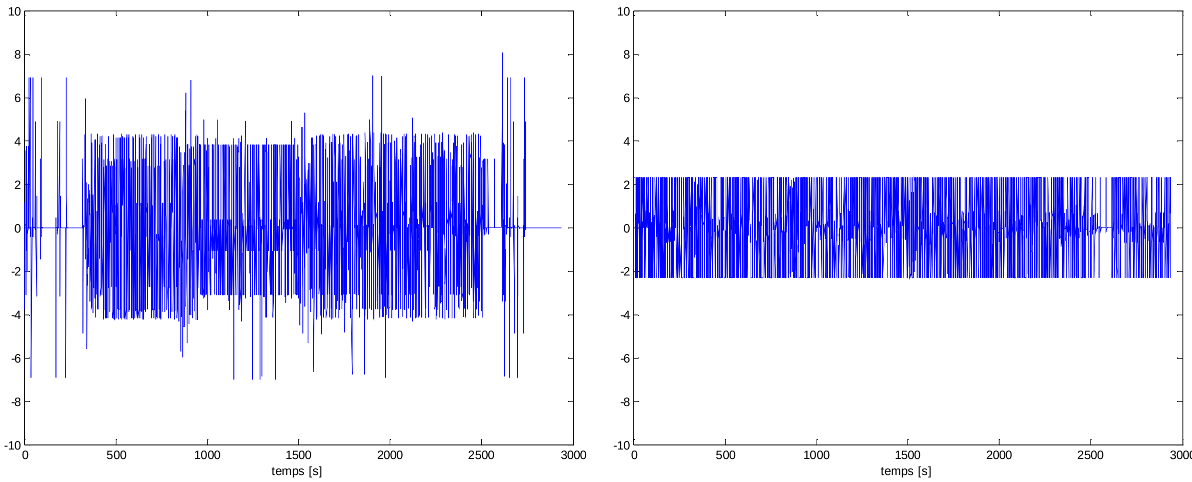



**Figure 7.35 : Incréments de commande. a) Cas sans contraintes ; b) la loi GPC avec contraintes**

On constate donc que les limitations de l'incrément de commande peuvent être vues comme un outil permettant de diminuer la sensibilité du signal de commande. Poursuivant cette idée, des résultats intéressants peuvent être mis en évidence pour le positionnement, comme ceux de la Figure 7.37, obtenus pour :

$$\begin{aligned}
&N_u = 1; N_1 = 1; N_2 = 16; N_c = 0 \\
&-0.8 \le \Delta u(t+k) \le 0.8, \quad k = 0,..., N_u
\end{aligned} \tag{7.29}$$

On peut observer lors de l'application de la loi explicite sur le système affecté par des bruits de mesures que les limitations (données sous la forme de régulateurs polynomiaux RST) seront affectées. Comme il a été mentionné au chapitre 6 sur la robustesse, les procédures classiques de robustification ne sont plus applicables, la solution dans ce cas étant la reconsidération du critère de coût pour prendre en compte les perturbations lors de l'étape de synthèse de la loi de commande. Sur le banc, cette procédure n'est pas possible car elle est incompatible avec la considération des contraintes terminales de type égalité (puisque leur satisfaction pour toutes les réalisations de la perturbation n'est pas garantie). Néanmoins, les contraintes inégalité se prêtent à de telles réalisations et le paragraphe suivant présente une loi prédictive robuste synthétisée dans le cas des contraintes de type inégalité sur la sortie prédite.

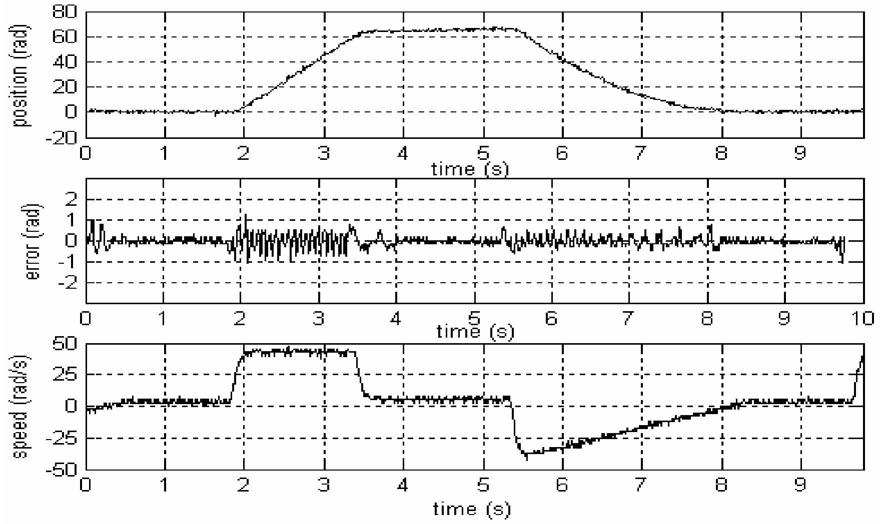



**Figure 7.36 : Résultats obtenus lors de l'implémentation temps réel (position, erreur et vitesse)**

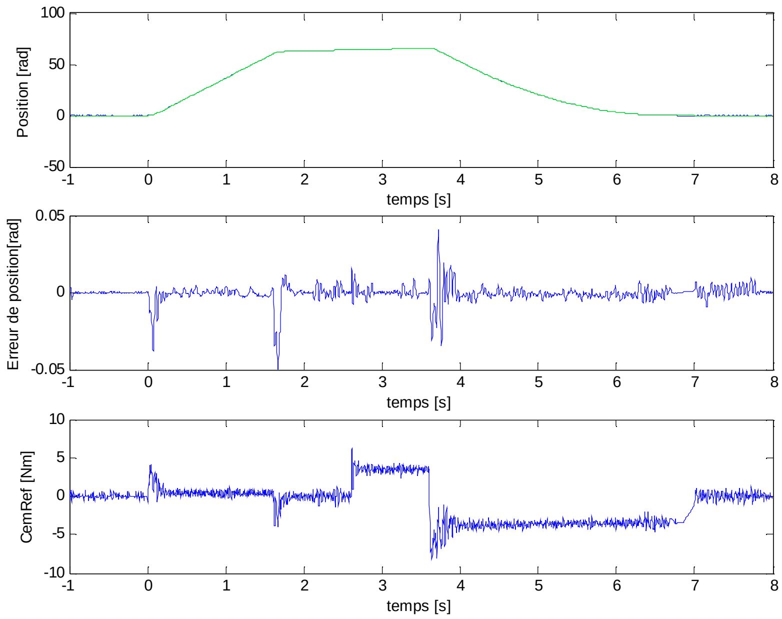



**Figure 7.37 : Réponses temporelles pour le correcteur GPC (7.29) avec contraintes renforcées sur l'incrément de commande**

## *7.5 Commande prédictive avec contraintes sur la sortie*

On considère maintenant le modèle (7.5) pour lequel les perturbations interviennent explicitement :

$$y(t) = q^{-1}\frac{B(q^{-1})}{A(q^{-1})}u(t) + v(t) \Rightarrow y(t) = \frac{10^{-4}(0.821q^{-1} + 0.8206q^{-2})}{(1-q^{-1})(1-0.998q^{-1})}u(t) + v(t) \tag{7.30}$$

Le but est ici de mettre en œuvre une méthode conduisant à la conception de lois de commande qui assure la satisfaction des contraintes sur la sortie. On impose alors pour le suivi de trajectoire un comportement sans dépassement :

$$\hat{y}(t+k) \leq w(t+k), \quad k = 0, ..., N_2 \tag{7.31}$$

Essayons tout d'abord de traiter le problème de façon classique. La procédure démarre par la recherche d'une loi de commande GPC classique (critère quadratique) en se basant sur le modèle nominal (7.5) puis continue avec la recherche de régulateurs robustifiés. Ainsi on considère un choix de paramètres $N_u = 2; N_1 = 1; N_2 = 4$ et une contrainte de type égalité conduisant à un comportement comme celui décrit Figure 7.38a.

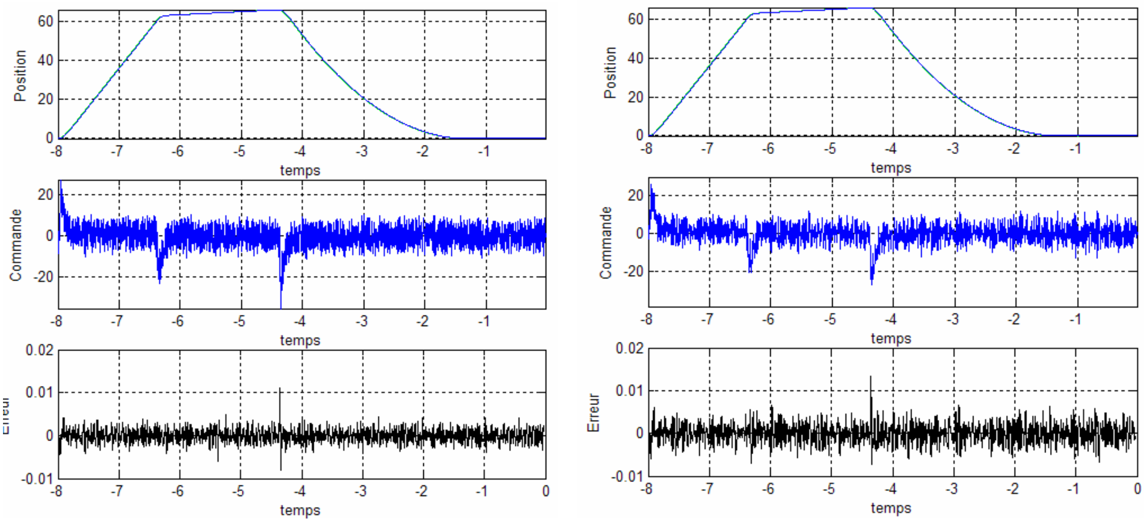



**Figure 7.38 : Réponses temporelles pour les correcteurs GPC obtenus à partir du modèle nominal et critère quadratique. a) Loi GPC initiale - la présence des perturbations se fait sentir sur le signal de commande et sur l'erreur de poursuite. b) Version de loi GPC robustifiée**

La procédure de robustification du régulateur RST équivalent donne les résultats de la Figure 7.38b. Les difficultés surviennent lors de la prise en considération des contraintes (7.31). Si l'on tient compte des contraintes lors de la synthèse, le problème d'optimisation multiparamétrique résultant peut être résolu comme expliqué au chapitre 4 aboutissant à une formulation multi-RST. La Figure 7.39a montre alors le comportement obtenu pour la boucle fermée avec cette loi de commande, les effets des perturbations se font ressentir et les contraintes ne sont pas satisfaites. Ce fait se vérifie plus clairement encore sur la Figure 7.39b.

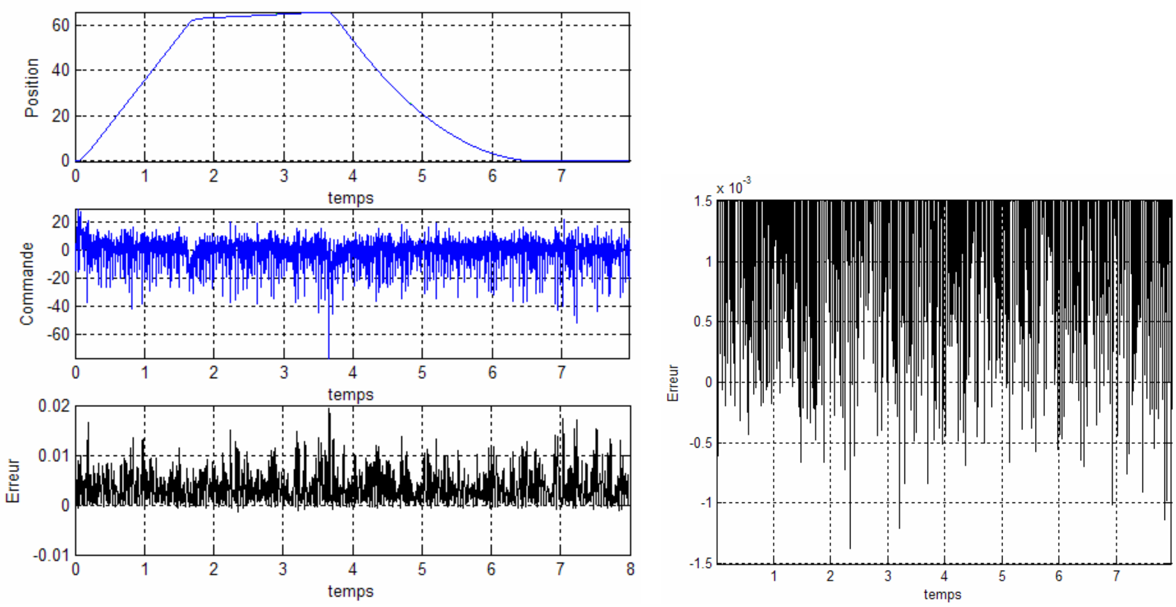



**Figure 7.39 : a) Réponses temporelles pour la loi GPC obtenues à partir du modèle nominal avec prise en compte de contraintes ; b) détail sur l'erreur de poursuite**

On peut espérer qu'un processus de robustification effectué après cette prise en compte des contraintes puisse redonner les performances satisfaisantes et satisfaire les contraintes. Mais comme il a été mentionné au chapitre 6, l'utilisation des paramètres de Youla pour les régulateurs polynomiaux RST définis par morceaux dans l'espace des paramètres de contexte introduit des problèmes de continuité de la loi de commande (Figure 7.40).

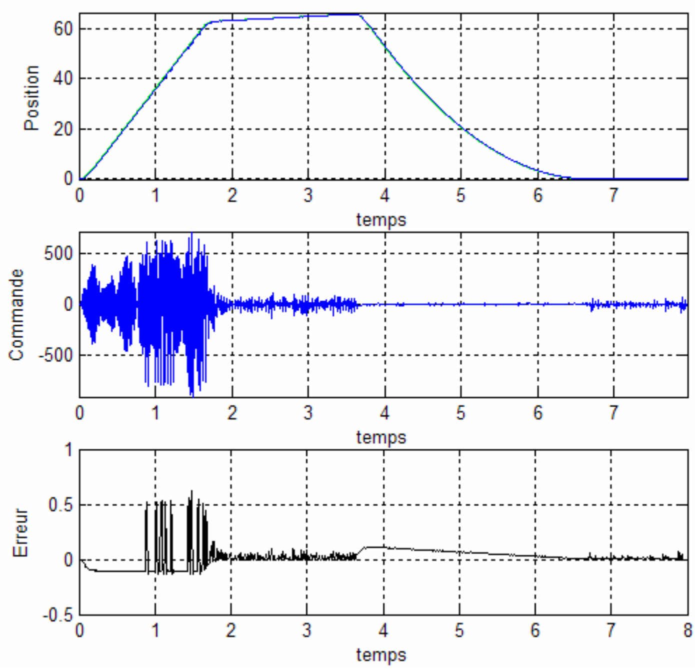



**Figure 7.40 : Simulation du comportement du système bouclé avec la loi GPC, pour laquelle les régulateurs RST par morceaux ont été robustifiés**

On se retrouve donc dans la situation pour laquelle le seul moyen disponible pour la satisfaction robuste des contraintes est la prise en compte de l'action de perturbation dans la phase de synthèse de la loi de commande. Il est ainsi nécessaire de résoudre à chaque pas d'échantillonnage un problème d'optimisation de type 'min-max' avec une fonction de coût exprimée par une somme de termes de norme infinie prenant en compte l'erreur de poursuite et l'effort de commande :

$$\begin{aligned}
&\min_{\{\Delta u(t+k)\}} \max_{\{v(t+k)\}} \sum_{k=1}^{N_2} \left| w(k+t) - y(k+t) \right| + \sum_{k=0}^{N_u-1} \left| \Delta u(k+t) \right| \\
&-0.001 \le v(t+k) \le 0.001; k = 1,..., N_2 \\
&y(t+k) \le w(t+k); k = 1,..., N_2
\end{aligned} \tag{7.32}$$

La vérification de la validité de la démarche pour la vérification robuste des contraintes est effectuée en considérant une version de complexité réduite avec un horizon de prédiction $N_2 = 4$ et une seule contrainte sur la première sortie prédite. Le résultat est illustré Figure 7.41a. On observe que la présence des perturbations ne compromet pas les objectifs de commande mais on peut aussi se rendre compte de la sensibilité du signal de commande vis-à-vis des perturbations présentes.

La sensibilité de la commande est améliorée en considérant les contraintes sur tout l'horizon de prédiction, ce que l'on constate avec le comportement décrit Figure 7.41b. En suivant le même principe, et en augmentant l'horizon de prédiction et de contraintes, on arrive à des lois de commande encore plus robustes, tout en respectant les limitations. Un exemple dans ce sens est la simulation présentée Figure 7.42 pour un horizon de prédiction $N_2 = 14$. Néanmoins, on peut observer le compromis présent entre la robustification et la performance en suivi du système corrigé.

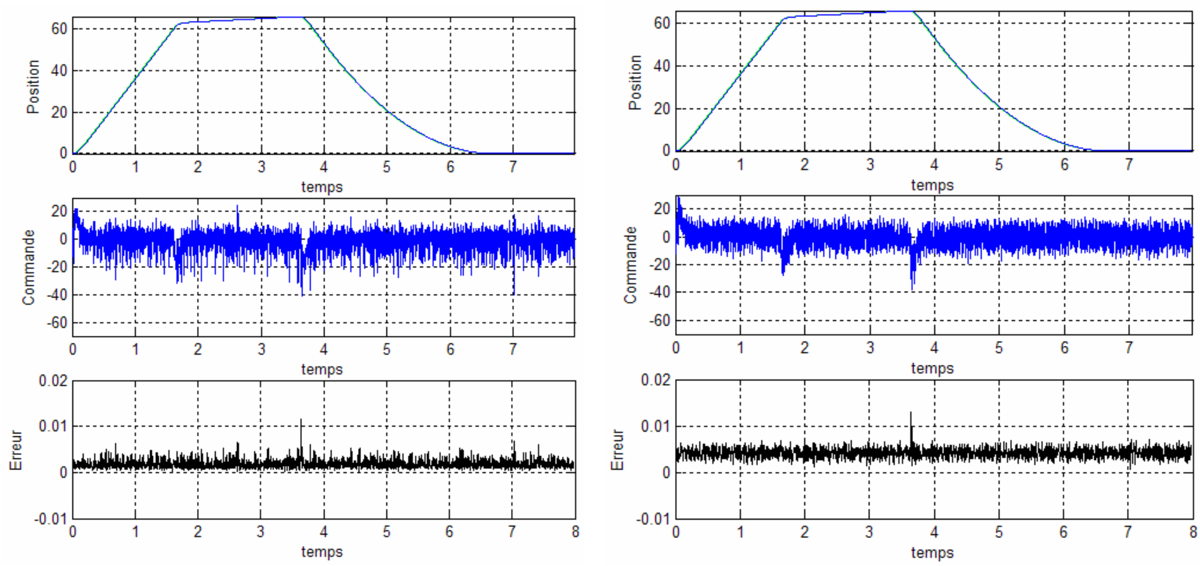



**Figure 7.41 : a) Réponses temporelles pour la loi GPC basée sur un critère min-max, $N_2 = 4$ et contraintes de dépassement sur le premier pas d'échantillonnage**
**b) loi GPC avec $N_2 = 4$ et contraintes de dépassement pour chaque prédiction**

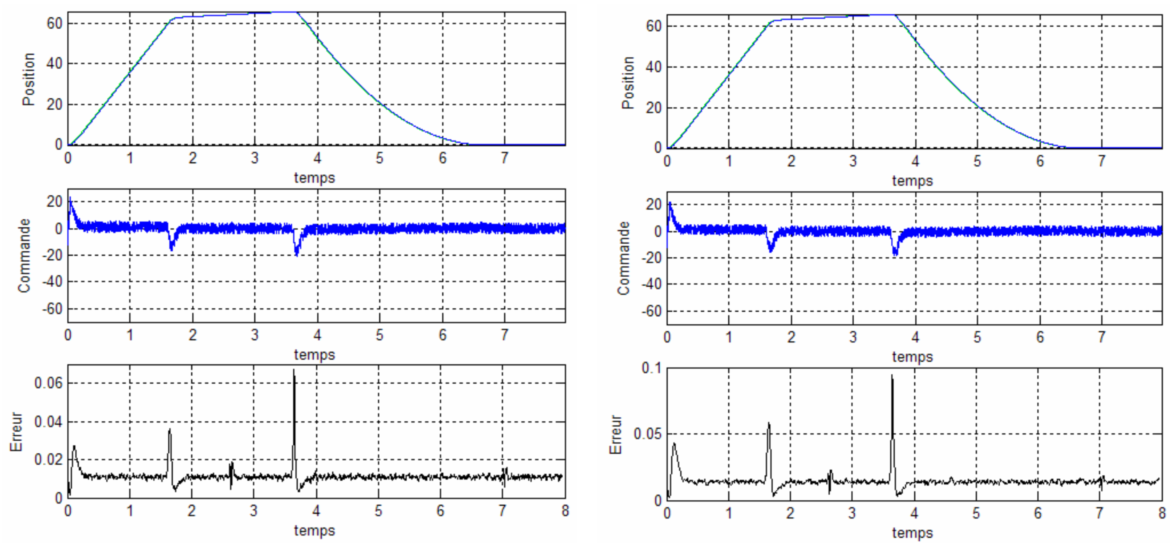



**Figure 7.42 : Loi GPC basée sur un critère min-max avec $N_2 = 14; N_u = 1$. a) Contraintes sur les dix premières sorties prédites ; b) contraintes sur toutes les sortie prédites**

L'augmentation de l'horizon de prédiction (et de contrainte) a cependant un impact sur la complexité de la formulation explicite de la loi de commande prédictive. Par exemple, un horizon de prédiction 14 et une contrainte sur le dépassement sur les 10 premières sorties prédites prend la forme d'un tableau avec 160 régulateurs polynomiaux RST. Ainsi le processus de positionnement en ligne devient problématique du point de vue ressource mémoire mais aussi programmation. Pour réduire la complexité de la solution explicite, une solution consiste à réduire l'horizon de prédiction et ajouter des contraintes sur les incréments de commande (Figure 7.43), le grand défi étant dans ce cas d'éviter l'infaisabilité.

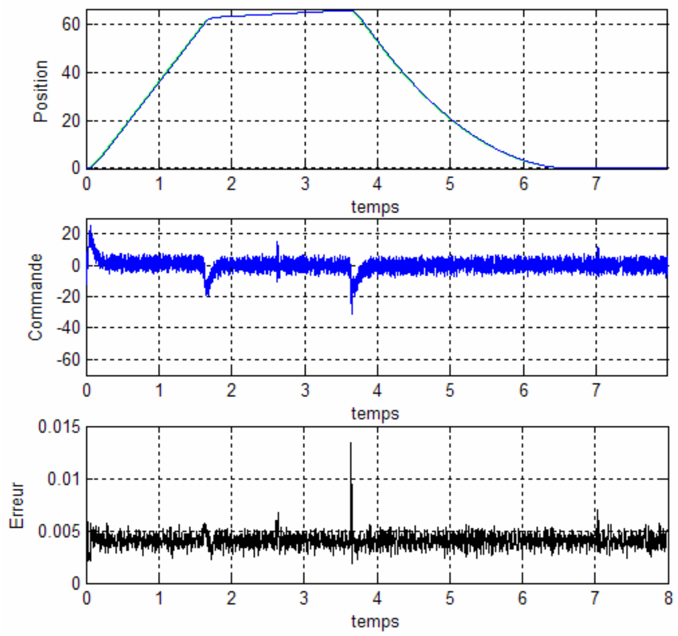



**Figure 7.43 : Loi GPC avec contraintes sur la sortie et sur la commande ($-15 \leq \Delta u(t) \leq 15$) se basant sur un critère min-max pour $N_2 = 4; N_u = 1$**

## *7.6 Conclusions*

Ce chapitre a été dédié à l'application de lois de commandes prédictives à un problème de positionnement d'un système électromécanique. On a pu ainsi illustrer une famille entière de problèmes qui peuvent être traités dans un contexte prédictif, de la loi sans contraintes, puis avec contraintes terminales de type égalité et ensuite en analysant les conséquences de la présence des contraintes de type inégalité, comme cela a été le cas pour les limitations de l'incrément de commande ou les contraintes sur les performances en suivi.

Dans tous les cas, la solution des problèmes d'optimisation multiparamétriques associés à la synthèse des lois prédictives est construite de sorte que l'expression explicite des lois de commande résultantes prenne la forme d'une loi linéaire par morceaux.

Les lois prédictives synthétisées pour les contraintes de type égalité ont pu être validées avec le simulateur non linéaire et sur le banc d'essais dans la continuité des travaux de [Rod03]. Une autre originalité concernant les résultats réside dans le test de la prise en compte des contraintes de type inégalité sur la commande à la fois sur le simulateur et en temps réel, grâce à la formulation des limitations sous une forme multi RST de faible complexité. En ce qui concerne les contraintes sur la sortie, elles ont été testées en simulation, leur implémentation temps réel pourra s'effectuer lors de la mise à niveau du banc du point de vue matériel actuellement en cours, car la complexité des lois linéaires par morceaux obtenues (avec le mécanisme de positionnement associé) nécessite des ressources conséquentes.

# 8. Conclusions

Cette thèse s'est intéressée à la commande prédictive sous contraintes. Celle-ci, grâce à sa formulation temporelle, présente le cadre nécessaire pour la prise en compte explicite de ces contraintes lors de la phase de synthèse. Les travaux de recherche ont traité les aspects théoriques et algorithmiques se situant à la frontière entre la commande prédictive et l'optimisation multiparamétrique.

L'approche développée considère comme axe principal l'ensemble de contraintes, et analyse les lois de commande résultantes par l'intermédiaire de la structure du domaine faisable. Mais, dès lors que la dynamique du système à asservir intervient dans la structure des contraintes, il est nécessaire de faire appel à des ensembles de contraintes paramétrés, qui définissent ensuite des domaines faisables pour les problèmes d'optimisation liés à la commande prédictive. Le point de départ a consisté à dresser un inventaire des méthodes d'implantation basées sur la résolution de ces problèmes à chaque pas d'échantillonnage pour ensuite explorer la possibilité d'utiliser les informations sur la topologie de l'ensemble faisable pour alléger l'effort en ligne.

Des méthodes alternatives ont alors été successivement mises en place pour opérer la migration progressive de cet effort de calcul en ligne vers des procédures hors ligne. Ainsi, des stratégies de passage de problèmes d'optimisation multiparamétriques généraux vers des optimisations multiparamétriques avec contraintes non redondantes par morceaux ont pu être élaborées. Finalement, par l'intermédiaire d'arguments géométriques, les lois de commande explicites ont été développées, équivalentes à l'application de séquences optimales de commande selon la philosophie de l'horizon glissant. Elles prennent la forme de fonctions linéaires affines par morceaux impliquant le partitionnement de l'espace des paramètres de contexte en sous-régions polyédrales. Ces régions sont stockées dans des tableaux avec les lois affines associées, de sorte que, lors de l'application temps réel, il suffise de se positionner dans ce tableau et d'évaluer la loi linéaire affine trouvée.

Ce mémoire a examiné de façon approfondie ces solutions explicites ainsi que leurs liaisons avec la structure des contraintes définissant les domaines faisables, car elles apportent des informations importantes sur les caractéristiques de la boucle fermée (faisabilité, stabilité, robustesse, efficacité d'implémentation). Cette formulation de la solution explicite des problèmes d'optimisation multiparamétriques en liaison avec un choix judicieux des paramètres prédictifs (horizons de commande, de prédiction et de contrainte, et fonction de coût associée) peut assurer les desiderata de faisabilité et stabilité (en utilisant comme ingrédients certaines éléments de la théorie des ensembles invariants) et constitue de plus une solution 'de remplacement' pour l'application aux systèmes rapides.

Pour résumer, l'utilisation des solutions explicites est une alternative intéressante aux stratégies d'optimisation itérative traditionnelles. Les avantages résident dans la structure simple (linéaire par morceaux) de la loi de commande et la diminution de la complexité de l'opération d'évaluation lors de l'implémentation. En effet, cette phase ne nécessite que des évaluations de fonctions linéaires et d'inégalités linéaires, ce qui s'avère simple de mise en œuvre par quelques lignes de code, et donc robuste du point de vue sensibilité numérique. Les inconvénients sont liés principalement à l'augmentation de la complexité du partitionnement (et donc des besoins en termes d'espace de mémoire et de positionnement en ligne), qui demeure exponentielle avec la dimension du problème d'optimisation associé. Cet aspect restreint l'application à des systèmes possédant une faible dimension de l'espace d'état (paramètres de contexte).

## *8.1 Apport scientifique et originalité du travail*

Les travaux relatifs à cette thèse ont fait un usage intensif pour les problèmes d'automatique des concepts liés à la géométrie informatique, comme celles des polyèdres paramétrés et des sommets paramétrés, utilisation n'étant pas clairement mise en œuvre dans les études antérieures. Ces concepts ont permis une représentation intuitive des limites de la région faisable pour les lois de commande basées sur des problèmes d'optimisation.

L'étude approfondie de la géométrie de l'espace faisable pour les problèmes d'optimisation multiparamétriques liés à la commande prédictive a permis tout d'abord de cristalliser certains résultats relatifs à la redondance des ensembles de contraintes. Du point de vue optimisation, ces résultats ont abouti à la description d'un problème d'optimisation multiparamétrique en sous-problèmes basés sur des sous-ensembles de contraintes non redondantes par morceaux. Des études comparatives ont alors démontré que pour des classes de problèmes d'optimisation multiparamétriques, caractérisées par d'importants degrés de redondance (ensembles de contraintes séparables), effectuer une telle étape de pré-découpage de l'espace des paramètres peut s'avérer utile en vue des performances en ligne. Cette phase en elle-même offre une formulation intermédiaire entre les procédures exclusivement basées sur une optimisation itérative en ligne et les méthodes basées sur l'évaluation des solutions explicites. Un avantage qui mérite d'être souligné est le fait que le partitionnement de l'espace des paramètres en vue de l'élimination de la redondance est réutilisable si l'on prend en compte des fonctions de coût différentes, ce qui n'est plus vrai dans le cas général lors de la construction des solutions explicites.

Une méthodologie de construction des solutions explicites, basée exclusivement sur des arguments géométriques, a ensuite été mise en place. Un de ses avantages provient du fait que les problèmes de dégénérescence sont ignorés car ils correspondent à des frontières des domaines de validité pour les sommets paramétrés décrivant la région faisable. Ces algorithmes ont été développés pour la construction des solutions explicites dans les deux cas des problèmes d'optimisation linéaire et quadratique multiparamétriques.

Enfin, au niveau de l'analyse des lois prédictives, une attention spéciale a été portée à la faisabilité, une des conditions de base pour assurer la fiabilité du système bouclé. En utilisant des éléments de la théorie des espaces invariants, des conditions nécessaires et suffisantes on été structurées pour permettre une analyse des jeux de paramètres conduisant à un comportement faisable, avant même l'application effective de la loi de commande. Ainsi ces conditions nécessaires et suffisantes de faisabilité peuvent être envisagées comme un outil d'aide au choix des paramètres de la loi prédictive. Toujours dans ce cadre, les situations d'infaisabilité ont été classifiées et des règles mises en place pour éviter de telles combinaisons. Dans le cas où les signaux exogènes conduisent le système hors de l'ensemble des paramètres de contexte pour lequel la loi de commande est faisable, on a démontré qu'il était possible de concevoir un mécanisme permettant la récupération de la faisabilité.

Les stratégies prédictives sous contraintes faisant appel à des lois explicites conçues à partir d'une démarche géométrique ont été appliquées pour le problème de positionnement d'une machine asynchrone. Les résultats obtenus sont dans la continuation de ceux élaborés dans un cadre sans contraintes dans [Rod03].

Pour conclure, les points précédents ont fait ressortir le rôle primordial des polyèdres paramétrés dans l'élaboration des différentes stratégies. Toutes les solutions ont pu être mises en œuvre grâce à la création d'une bibliothèque de programmes informatiques manipulant les polyèdres paramétrés fonctionnant dans l'environnement Matlab™. Cet outil basé sur la double représentation des polyèdres s'avère en fait très général, n'étant pas réservé uniquement aux applications de l'Automatique.

## *8.2 Perspectives*

Même si la commande prédictive avec contraintes a constitué ces dernières années un domaine de recherche très actif, des développements sont encore envisageables, tout particulièrement en relation avec la programmation multiparamétrique en vue d'une application pour des systèmes rapides. Ainsi, des travaux futurs en relation directe avec les sujets traités dans ce mémoire peuvent se répartir en trois catégories : travaux sur la partie informatique, travaux concernant la mise au point d'architecture matérielle dédiée aux applications de masse, travaux liés aux perspectives d'élargissement des familles de systèmes et de contraintes pour lesquelles on peut concevoir des solutions explicites.

*A) Améliorations informatiques*
Cette catégorie englobe la construction de solutions explicites mais aussi les procédures qui permettront la simplification du processus d'évaluation en ligne :

- L'exploitation du fait que les solutions explicites possèdent une évolution progressive avec l'augmentation de l'horizon de prédiction peut orienter la construction des lois prédictives vers une procédure de synthèse basée sur le principe de la programmation dynamique.
- Sachant que l'évaluation en ligne d'une fonction définie par morceaux conditionne les performances temps réel de la loi prédictive, la conception de mécanismes de positionnement intelligents au sein de la table de régulateurs est nécessaire de façon à réduire la charge informatique. Dans cette direction, le remplacement de la solution optimale (linéaire par morceaux) par une solution sous-optimale, basée sur la concentration des partitions dans l'espace des paramètres, peut faciliter l'évaluation de la solution. Certains travaux existent dans la littérature sur la construction des telles solutions approximatives, mais nécessitent généralement le passage par la solution optimale [RG05], [JG02], [GJ02]. Ainsi un sujet de recherche prometteur concerne la construction *directe* des solutions explicites sous-optimales pour les problèmes d'optimisation multiparamétriques. Il subsiste néanmoins des problèmes en ce qui concerne le degré de sous-optimalité, la complexité d'évaluation de la loi résultante qui à son tour est liée à la structure de la fonction.

- Les solveurs des problèmes d'optimisation multiparamétrique peuvent être améliorés si les procédures géométriques deviennent plus performantes (exploitation de la structure de l'ensemble de contraintes, exploitation de la transformation des sommets paramétrés avec l'évolution des paramètres, migration des solveurs mpQP vers des langages de niveaux plus proches de la machine).
- La complexité de la loi MPC avec contraintes dans le cas des systèmes incertains ou affectés par des perturbations reste un point sensible tout comme le conservatisme des résultats existants. Des résultats théoriques sur la robustesse de la faisabilité seront très recherchés pour les applications sensibles aux défaillances.

*B) Implémentation*

La mise en place d'architectures matérielles spécifiques pour l'implémentation des lois de commande linéaires par morceaux (éventuellement avec structure reconfigurable type FPGA), à l'aide de matériel de faible coût (arithmétique en virgule fixe) est envisageable dans un futur immédiat. Les ressources mémoire nécessaires sont néanmoins importantes, cet aspect doit être étudié avec soin. Une fois ces problèmes résolus, l'application à grande échelle de modules de commande prédictive se fera avec un faible coût de conception.

C) *Extension des solutions explicites à d'autres problèmes de commande basés sur une optimisation*

- Une première extension peut être la formulation des solutions explicites pour les problèmes d'optimisation multiparamétriques continus. Ceci pourra ensuite ouvrir des perspectives vers la façon de choisir une période d'échantillonnage assurant une complexité réduite de la solution explicite pour le système discret.
- Il existe dans la théorie de la commande prédictive des formulations plus souples utilisant des horizons de prédiction variables (pour surpasser les situations d'infaisabilité, pour s'adapter aux changement de configuration …). Ce type de formulations est difficilement traitable par les procédures de synthèse des lois explicites présentées dans le présent mémoire, leur extension constitue donc une direction de travaux futurs.
- La commande de systèmes pour lesquels l'ensemble des commandes admissibles est délimité une fois par les contraintes de type polyédral et ensuite par l'appartenance à un alphabet fini est connu comme un problème qui se réduit à des partitions dans l'espace des variables d'état [QGD04]. L'investigation de l'extension de l'approche par polyèdre paramétré pour ce type de système et généralement vers les systèmes hybrides peut s'avérer une perspective intéressante.
- Des développements sont envisageables pour étendre la formulation explicite vers la loi prédictive dans le cadre non linéaire. La migration de l'effort de calcul en ligne vers des procédures hors-ligne de construction des solutions explicites peuvent être intéressantes mais il ne faut pas oublier que les problèmes d'optimisation seront plus compliqués (optimisations générales).

Finalement, une attention particulière doit être portée au problème dual de la commande sous contraintes, à savoir l'estimation sous contraintes. Le domaine est très intéressant et il a reçu l'attention d'un ouvrage récent [GSD04] présentant certains résultats (en prouvant la richesse du domaine) qui peuvent être étendus.

## 9. Bibliographie


[AM71] B. Anderson, J. Moore, *"Linear Optimal Control"*, Prentice Hall, 1971.

[Bao02] M. Baotic. "An Efficient Algorithm for Multi-Parametric Quadratic Programming*", Technical repport AUT02-04*, Automatic Control Laboratory, ETH Zurich, 2002.

[BBM01] A. Bemporad, F. Borelli, M. Morari, "Robust Model Predictive Control: Piecewise Linear Explicit Solution", *Proceedings of the European Control Conference*, pp. 939-944, Karlsruhe, 2001.

[BD96] P. Boucher, D. Dumur, *"La commande prédictive"*, Editions Technip, Paris, 1996.

[Bea55] E.M.L. Beale. "On minimizing a convex function subject to linear inequalities", Journal of Royal Statistical Society, Ser. B 17, pp.173-184, 1955.

[Bel57] R. Bellman. *"Dynamic Programming"*, Princeton University Press, Princeton, 1957.

[BEM02] A. Bemporad, "Efficient conversion of mixed logical dynamical systems into a equivalent piecewise affine form", IEEE Transactions on Automatic Control, Vol. 49 (5), pp. 832-838, 2004.

[BF01] A. Bemporad, C. Filippi. "Suboptimal explicit MPC via approximate quadratic programming". *Proceedings IEEE Conference Decision and Control,* Orlando. pp. FrP08–5, 2001.

[BFT02] A. Bemporad, K. Fukuda, F.D. Torrisi, "Convexity Recognition of the Union of Polyhedra", *Computational Geometry*, Vol. 18 (3), pp. 141-154, 2002.

[BGW90] R.R. Bitmead, M. Gevers, V.Wertz, *"Adaptive optimal control: The thinking man's GPC"*, Prentice Hall International, 1990

[Bi88] G. Bitsoris, "On the positive invariance of polyhedral sets for discrete-time systems", *System & Control Letters*, Vol. 11, pp.243-248, 1988.

[Bla99] F. Blanchini, "Set invariance in control", *Automatica,* Vol. 35, pp. 1747-1767, 1999.

[BM96] A. Bemporad, E. Mosca, "Fulfiling hard constraints in uncertain linear systems by reference managing", *Automatica*, Vol. 34, No. 4 pp.451-461, 1996.

[BMDP02] A. Bemporad, M. Morari, V. Dua, E. Pistikopoulos "The Explicit Linear Quadratic Regulator for Constrained Systems", *Automatica*, Vol. 38, pp. 3-20, 2002.

[BN01] A. Ben-Tal, A. Nemirovski. "Lectures on Modern Convex Optimization. Analysis, Algorithms, and Engineering Applications", SIAM, 2001.

[BOR03] F. Borelli, *"Constrained Optimal Control of Linear and Hybrid Systems"*, Springer-Verlag, Berlin, 2003.

[BSS93] M. S. Bazaraa, H. D. Sherali, C.M. Shetty. *"Nonlinear programming. Theory and algorithms"* Second edition, John Wiley ans Sons, 1993.

[CAM93] E. F. Camacho, "Constrained Generalized Predictive Control (Constrained receding horizon predictive control)", *IEEE Transactions on Automatic Control*, Vol. 38(2), pp.327-331, 1993.

[Can00] C. Canudas, "Commande des moteurs asynchrones", vol1 et 2, Editions Hermes, 2000.

[CB03] E.F. Camacho, C. Bordons, *"Model Predictive Control",* Springer, 2003.

[CG86] M. Cwikel, P.-O. Gutman, "Convergence of an algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states", *IEEE Transactions on Automatic Control*, Volume 31, Issue 5, pp. 457 - 459, 1986.

[Che64] N.V. Chernikova. "Algorithm for finding a general formula for the non-negative solutions of a system of linear equations", USSR Computational Mathematics and Mathematical Physics, Vol. 4(4), pp. 151-158, 1964.

[CM86] P.J. Campo, M. Morari, "$\infty$ Norm Formulation of Model Predictive Control Problems" *Proceedings of American Control Conference* 1986, pp. 339-343, 1986.

[CM96] D. Chmielewski, V. Manousiouthakis. "On constraints infinite-time linear quadratic optimal control". *Systems and Control Letters*, Vol. 29, pp. 121-129, 1996.

[CMP04] A. Casavola, E. Mosca, M. Papini, "Control under Constraints: An application of the command governor approach to inverted pendulum", *IEEE Transactions on control system technology*, vol. 12, No. 1 pp.193-204, 2004.

[CMT87] D.W Clarke, C. Mohtadi, P.S. Tuffs, "Generalized Predictive Control: Part I The Basic Algorithm, Part II: Extensions and Interpretation", *Automatica*, 23 (2), 137-160, 1987.

[CS91] D. W. Clarke, R. Scattolini, "Constrained receding-horizon predictive control", *IEE Proceedings-D*, Vol. 138 (4), 1991.

[CZ99] L. Chisci, G. Zappa, "Fast algorithm for a constrained infinite horizon LQ problem". *International Journal of Control*, vol. 72(11), pp. 1020-1026, 1999.

[DB98] D. Dumur, P. Boucher, "A Review Introduction to Linear GPC and Applications", *Journal A*, Vol. 39(4), pp.21-35, 1998.

[DeD00] J. DeDona. *"Input Constrained Linear Control"*, PhD thesis, University of Newcastle Australia, 2000.

[DP00] V. Dua, E. N. Pistikopoulos. "An algorithm for the solution of multiparametric mixed integer linear programming problems". Annals of Operations Research, 99, pp. 123-139, 2000.

[EBD96] A. Ehrlinger, P. Boucher, D. Dumur, "Unified Approach of Equality and Inequality Constraints in G.P.C", *Proceedings of the 5th IEEE Conference on Control Applications*, pp.893-899, Dearborn, 1996.

[FLE81] R. Fletcher, *"Practical Methods of Optimization 2: Constrained Optimization"*, John Wiley and Sons, Chichester, 1981.

[FQ88] F. Fernandez, Quinton, P. "Extension of Chernikova's Algorithm for Solving General Mixed Linear Programming Problems". Technical Report 437, IRISA, Rennes, 1988.

[Fuk04] K. Fukuda. "Polyhedral computation FAQ*"*, *http://www.ifor.math.ethz.ch/staff/fukuda,* 2004.

[Fuk05] K. Fukuda. CDD http://www.ifor.math.ethz.ch/~fukuda/CDD_home/cdd.html, 2005.

[GAL94] T. Gal. Postoptimal analyses, *"Parametric Programming, and Related Topics". Second edition.* De Gruyter. Berlin, 1994.

[GBT05] "Geometric Bounding Toolbox (GBT) for Matlab". S. M. Veres; http://www.sysbrain.com.

[GC86] P.-O. Gutman, M. Cwikel, "Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states", *IEEE Transactions on Automatic Control,* Volume 31, Issue 4, pp. 373 - 376, 1986.

[GC87] P.-O. Gutman, M. Cwikel, "An algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states", *IEEE Transactions on Automatic Control*, Volume 32, Issue 3, pp. 251 – 254, 1987.

[GJ02] A. Grancharova, T. A. Johansen. "Approximate explicit model predictive control incorporating heuristics". *Proceedings IEEE Conference on Computer Aided Control Design*, Glasgow, 2002.

[GM82] C.E. Garcia, M. Morari, "Internal Model Control. A unifying review and some new results", *I&EC Process Design and Development*, 21, 308-323, 1982.

[GN72] T. Gal, J. Nedoma, *"Multiparametric Linear Programming"*, Management Science, 18, 1972.

[GS03] A. H. Glattfelder, W. Schaufelderger, "*Control Systems with Input and Output Constraints"*, Springer, 2003.

[GSD04] G. C. Goodwin, M.M. Seron, J.A. De Dona, *"Constrained Control and Estimation",* Springer-Verlag, London, 2004.

[GT91] E.G. Gilbert, K.T. Tan. "Linear systems with state and control constraints: The theory and application of maximal output admissible sets." *IEEE Transactions on Automatic Control*, Vol. 36(9), pp.1008-1020, 1991.

[Hen95] J.C. Hennet, "Discrete time constrained linear systems", *Control and Dynamical Systems*, Vol. 71, pp. 157-213, 1995.

[Hen89] J. C. Hennet. "Une extension du lemme de Farkas et son application au probleme de regulation lineaire sous contraintes". *Comptes Rendus des Seances de l'Academie des Sciences Paris*, No. 308 (Serie I), pp. 415-419, 1989.

[JG02] T. A. Johansen, A. Grancharova. "Approximate explicit model predictive control implemented via orthogonal search tree partitioning". *Preprints, IFACWorld Congress*, Barcelona, 2002.

[JKM04] C. Jones, E. C. Kerrigan, J. M. Maciejowski. "Equality set Projection: A new algorithm for the projection of polytopes in halfspace representation", *Technical Report: CUED/F-INFENG/TR.463,* Department of Engineering, University of Cambridge , 2004.

[KAL60] R.E. Kalman "Contributions to the theory of optimal control", Boletin de la Sociedad Matematica Mexicana, Vol. 5, PP.102-119, 1960.

[KAN95] P.P. Kanjilal, "Adaptive prediction and predictive control", IEE, Peter Peregrinus Ltd., 1995.

[KAR84] N. Karmarkar. “A new polynomial-time algorithm for linear programming”, Combinatorics, Vol. 4,pp. 373-395,1984.

[KBM96] M. V. Kothare, V. Balakrishnan, M. Morari, “Robust Constrained Model Predictive Control using Linear Matrix Inequalities”, *Automatica,* vol. 32, 1996.

[KCM94] M. V. Kothare, P.J. Campo, M. Morari, C. N. Nett. “A unified framework for the study of anti-windup designs”, *Automatica*, vol. 30, pp. 1869–1883, 1994.

[KCR00] B. Kouvaritakis, M. Cannon, J.A. Rossiter, “Stability, feasibility, optimality and the degrees of freedom in constrained predictive control”, Nonlinear Model Predictive Control, F. Allgower and A. Zheng (eds.), Progress in Systems and Control Theory Series, Volume No 26, pp.403-417, Birkhauser Verlag, Bâle.

[Ker00] E. Kerrigan*, “Robust Constraint Satisfaction: Invariant Sets and Predictive Control”*, PhD Thesis, University of Cambridge, 2000.

[Ker04] E. Kerrigan, “Feedback Min-max Model Predictive Control using a Single Linear Program: Robust Stability and the Explicit Solution”, *International Journal of Robust and Nonlinear Control*, vol.14, 2004.

[KG87] S.S. Keerthi, E.G. Gilbert. “Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints”, IEEE Transactions on Automatic Control, Vol. AC-32(5), pp. 432-435, 1987

[KGB04] M. Kvasnica, P. Grieder, M. Baotic, M. Morari, “Multi-parametric Toolbox (MPT)”, http://control.ee.ethz.ch/ mpt/, 2004. (2004)

[KH03] K. Kogiso, K. Hirata,”A reference governor in a piecewise state affine function: The implementation and validation”, *42nd IEEE Conference on Decision and Control*, Maui, 2003.

[KLTZ83] M. H. Karwan, V. Lotfi, J. T. Telgen, and S. Zionts, *“Redundancy in Mathematical Programming: A State of the Art Survey”*, Springer-Verlag, Berlin, 1983.

[KM00] E.C. Kerrigan, J.M. Maciejowski,”Invariant Sets for Constrained Nonlinear Discrete-time Systems with Application to Feasibility in Model Predictive Control”, *39th IEEE Conference on Decision and Control*, Sydney, 2000.

[KM02] E. C. Kerrigan, J.M. Maciejowski, “Feedback Min-max Model Predictive Control using a Single Linear Program: Robust Stability and the Explicit Solution”, *Technical Report Cambridge University Engineering Department*, 2002.

[KS72] H. Kwakernaak, R. Sivan, “*Linear Optimal Control Systems*”, John Wiley & Sons, New-York, 1972.

[Kuc79] V. Kucera, “Discrete linear control: the polynomial equation approach”, John Wiley and Sons, Chichester, 1979.

[LeF94] M.Le Fur. “Parcours de polyèdre paramétré avec l'élimination de Fourier-Motzkin”, Technical Report RR2358, IRISA, France, 1994.

[LEM68] C. E. Lemke *“Mathematics of the decision science”*, chapter on Complementary Pivot Theory. Editeurs G.B. Dantzing et A.F. Veinott, 1968

[LEO01] D. Leonhard, *“Control of electrical drives”*, Springer-Verlag, 2001.

[LEV94] H. Leverge, *“A Note on Chernikova’s Algorithm”*, Technical Report 635, IRISA, France, 1994.

[LY97] J.H. Lee, Z. Yu, “Worst case formulation of Model Predictive Control for Systems with bounded parameters”, Automatica, Vol. 33, pp. 763-781, 1997

[LW97] V. Loechner, D.K. Wilde, “Parameterized Polyhedra and their Vertices”, *International Journal of Parallel Programming*, Vol. 25(6), 1997.

[MAC02] J. Maciejowski, *“Predictive Control with Constraints”*, Prentice Hall, 2002.

[May01] D.Q. Mayne, “Control of Constrained Dynamic Systems”, *European Journal of Control*, Vol. 7, pp. 87-99, 2001.

[MB02] E. Mendes, J.P. Barbot, “Benchmark transitique rapide”, *APII Journal Européen des Systèmes Automatisés*, Vol. 36-5, pp. 701-708, 2002.

[McM70] P. McMullen. The maximum number of faces of a convex polytope. *Mathematika,* XVII:pp. 179-184, 1970.

[MD04] J. Mare, J. De Dona, "Use of Dynamic Programming for the Analytical Solution of Input-Constrained LQR Problems", 5th IEEE Asian Control Conference ASCC'04, Melbourne, Australie, 20-23 Juillet 2004.

[MM93] H. Michalska, D. Q. Mayne. "Robust receiding horizon control of constrained nonlinear systems", *IEEE Transactions on Automatic Control*, vol.38, 1623–1633, 1993.

[MOS95] E. Mosca "Optimal Predictive and Adaptive Control", Prentice Hall, 1995.

[MRRS00] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scockaert, "Constrained model predictive control: Stability and optimality", *Automatica*, Vol.36, pp. 789-814, 2000.

[MRTT53] T. S. Motzkin, H. Raiffa, G.L. Thompson, R.M. Thrall, "The Double Description Method", The Double Description Method, 1953 republié in *Theodore S. Motzkin: Selected Papers*, Birkhauser, Boston, 1983.

[NN94] Y. Nesterov, A. Nemirovskii. "Interior-Point Polynomial Algorithms in Convex Programming", SIAM, 1994.

[OD03] S. Olaru, D. Dumur, "Feasibility analysis of constrained predictive control", *Proceedings of the 14th Conference on Control Systems and Computer Sciences*, pp.164-169, Bucharest, 2003.

[OD04a] S. Olaru, D. Dumur, "Feasibility of Constrained Generalized Predictive Control within Invariant Sets Framework", 5th *IEEE Asian Control Conference ASCC'04, Melbourne*, Australie, 20-23 Juillet 2004. (Best Poster Prize)

[OD04b] S. Olaru, D. Dumur, "Some Feasibility Issues Related to Constrained Generalized Predictive Control", *1st International Conference on Informatics in Control, Automation and Robotics ICINCO'04*, Setubal, Portugal, 25-28 Août 2004.

[OD04c] S. Olaru, D. Dumur, "On Constrained Predictive Control On-line Optimization Routines" *Proceedings of the IEEE Conference on Methods and Models in Automation and Robotics,* Miedzyzdroje, Poland, September, 2004.

[OD04d] S. Olaru, D. Dumur, "A Parameterized Polyhedra Approach for Explicit Constrained Predictive Control", *Proceedings of the 43rd IEEE Conference on Decision and Control*, Bahamas, 2004.

[OD05a] S. Olaru, D. Dumur, "Avoiding Constraints Redundancy in Predictive Control Optimization Routines", à paraître dans *IEEE Transaction on Automatic Control* (Août 2005).

[OD05b] S. Olaru, D. Dumur, "Feasibility of constrained generalized predictive control within invariant sets framework", *Australian Journal of Electrical and Electronics Engineering, Engineers Australia*, 2(1):69-80, 2005.

[OD05c] S. Olaru, D. Dumur, "Constrained predictive control for position tracking of an induction motor", *15th Computer Science and Control Systems International Conference*, Bucarest, Roumanie, 25 – 27 Mai 2005.

[OD05d] S. Olaru, D. Dumur, "Explicit constrained model predictive control. The influence of constraints redundancy", *5th IEEE International Conference on Control and Automation*, Budapest, Hongrie, 26-29 Juin 2005.

[OD05e] S. Olaru, D. Dumur, "Constrained model predictive control. Influence of redundancy in the set of constraints, possible amelioration*", IMACS Conference*, Paris, France, 11-15 Juillet 2005.

[OD05f] S. Olaru, D. Dumur, "Parameterized Polyhedra Approach for Robust Constrained Generalized Predictive Control", *11th International IEEE Conference on Methods and Models in Automation and Robotics, MMAR'05,* Miedzyzdroje, Pologne, Septembre 2005.

[OD05g] S. Olaru, D. Dumur, "Parameterized polyhedra approach for robust constrained generalized predictive control*", IEEE Conference on Control Applications*, Toronto, Canada, Août 2005.

[OD05h] S. Olaru, D. Dumur, "Analysis of MPC feasible domains using a parameterized polyhedra approach", *International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, Freudenstadt-Lauterbad, Allemagne, 26-30 Août 2005.

[OD05i] S. Olaru, D. Dumur, "A parameterized polyhedra approach for the explicit Robust Model Predictive Control", *2nd International Conference on Informatics in Control, Automation and Robotics ICINCO'05*, Barcelone, Espagne, 14-17 Septembre 2005.

[PABC04] D. M. Pena, T. Alamo, A. Bemporad, E.F. Camacho, "A dynamic programming approach for determining the explicit solution of linear MPC controllers" *43rd IEEE Conference on Decision and Control*, 2004. Vol. 3, 14-17, pp. 2479 - 2484 Vol.3, 2004.

[PH04] J. Pekar, V. Havlena,"Design and analysis of model predictive control using MPT Toolbox", http://dsp.vscht.cz/konference matlab/ matlab04/pekar.pdf, 2004.

[PVH96] Y. Peng, D. Vrančić, R. Hanus. "Anti-windup, bumpless and conditioned transfer techniques for PID controllers". *IEEE Control System Magazine*, vol 16, pp. 48–57, 1996.

[PWR03] D. Pannocchia, S.J. Wright, J.B. Rawlings. "Existence and computation of infinite horizon model predictive control with active steady-state input constraints". *IEEE Transactions on Automatic Control*, vol. 48(6), 1002–1006, 2003.

[QB97] J. S. Qin, T. A. Badgwell, "An overview of industrial model predictive control technology". *AIChE Symposium on Chemical Process Control*, Vol. 93, pp. 232–256, 1997.

[QGD04] D.E. Quevedo, G.C. Goodwin, J. A. DeDona, "Finite constraint set receiding horizon quadratic control", International Journal of Robust and Nonlinear Control, Vol. 14 (4), pp. 355-377, 2004.

[RWR98] C. V. Rao, S. J. Wright, and J.B. Rawlings "Efficient interior point methods for model predictive control", *Journal of Optimization Theory and Applications,* 99(3), pp. 723-757, 1998.

[RAW00] J. B. Rawlings. "Tutorial overview of model predictive control". *IEEE Control Systems Magazine*, vol 20, pp. 38–52, 2000.

[RD02] P. Rodriguez, D. Dumur, "Robustification of GPC controlled system by convex optimisation of the Youla parameter", *Procedings of Conference on Control Applications*, pp. 1236-1241, Glasgow, 2002.

[RG05] J.A. Rossiter, P.Grieder. "Using interpolation to improve efficiency of multiparametric predictive control", Automatica, Vol. 41(4), pp. 637-643, 2005.

[RIC93] J. Richalet, *"Pratique de la commande predictive"*, Hermes, 1993.

[RM93] J. B. Rawlings, K. R. Muske. "The stability of constrained receding-horizon control". *IEEE Transactions on Automatic Control ,* Vol. 38, 1512-1516, 1993.

[Rod03] P. Rodriguez, "Robustification de lois de commande prédictive par la parametrisation de Youla", thèse soutenue à Univ. Paris XI, le 25 sept.2003.

[ROS60] J.B. Rosen. "The gradient projection method for nonlinear programming", Part 1, Linear Constraints. SIAM Journal of Applied Mathematics, Vol. 8, pp. 181-217, 1960.

[ROS03] J. A. Rossiter, *"Model – based predictive control. A practical approach"*, CRC Press, ISBN 0-8493-1291-4, 2003.

[SC94a] P.O. M. Scockaert, D. Clarke, "Stability and feasibility in constrained predictive control", *Advances in model-based predictive control*, pp. 217-230, Oxford University Press, 1994.

[SC94b] P. O. M. Scockaert, D. Clarke, "Stabilising properties of constrained predictive control", *IEE Proceedings Control Theory Applications*, Vol. 141(5), pp.295-304, 1994.

[Sch86] A. Schrijver, *"Theory of Linear and Integer Programming"*, John Wiley and Sons, NY, 1986.

[SD87] M. Sznaier, M. J. Damborg. "Suboptimal control of linear systems with state and control inequality constraints". *Proceedings of the 26th IEEE Conference on Decision and Control*, pp. 761-762, 1987.

[SGD02] M. M. Seron, G.C. Goodwin, J.A. De Dona, "Characterisation of Receding Horizon Control for Constrained Linear Systems", *Asian Journal of Control*, Vol. 5(2), pp. 271-286, 2002. (2003).

[SKH01] H. Sunan, T.K. Kiong, L.T.Heng, *"Applied Predictive Control"*, Springer, 2001.

[SLG04] R. Suard, J. Lofberg, P. Grieder, M. Kvasnica, M. Morari, "Efficient Computation of Controller Partitions in Multi-Parametric Programming", *Proceedings of the 43rd IEEE Conference on Decision and Control*, Bahamas, December 2004. pp.

[SM98] P. O. M. Scokaert, D.Q. Mayne, "Min-max Feedback Model Predictive Control for Constrained Linear Systems", *IEEE Transactions on Automatic Control*, Vol. 43(8), 1136-1142, 1998.

[Son90] E. Sontag. "Mathematical Control Theory: Deterministic Finite Dimensional Systems". Springer-Verlag, New York, 1990.

[SR96] J.M.M. Sanchez, J. Rodellar, *"Adaptive Predictive Control. From concepts to plant optimization"*, Prentice Hall, 1996.

[SR98] P.O.M. Scokaert, J.B. Rawlings. "Constrained linear quadratic regulation". *IEEE Transactions on Automatic Control*, vol. 43(8), 1163–1169, 1998.

[STE03] G. Stein. "Respect the unstable", *IEEE Control System Magazine*, Vol. 23 (4), pp. 12-25. (2003).

[SY91] H.J. Sussmann; Y. Yang. "On the stabilizability of multiple integrators by means of bounded feedback controls Proceedings of the 30th IEEE Conference on Decision and Control 1991, vol.1, pp. 70 - 72, 1991.

[TG97] S. Tarbouriech, G. Garcia, *"Control of Uncertain Systems with Bound Inputs"*, Springer, 1997.

[TJB01] P. Tondel, T. Johansen, A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions", *Proceedings of the 40th IEEE CDC*, 2001.

[TJB03] P. Tøndel, T. Johansen, A. Bemporad P. Tøndel, "An algorithm for multiparametric quadratic programming and explicit MPC solutions", *Automatica*, Vol. 39, pp. 489-497, 2003.

[Ton0] P. Tondel, "Constrained Optimal Control via Multiparametric Quadratic Programming", PhD thesis NTNU, 2003.

[VHB88] M. C. Vassilaki, C. Hennet, G. Bitsoris, "Feedback control of linear discrete-time systems punder state and control constraints", *International Journal of Control*, Vol. 47, pp.1727-1735, 1988.

[ZM93] A. Zheng, M. Morari. "Robust Stability of Model Predictive Control" Proceedings of American Control Conference 1993, pp. 379-383, 1993.

[YE89] Y. Ye. "Interior Point Algorithms for quadratic programming", Paper Series No 89-29, Departement of Management Sciences, University of Iowa, Iowa, 1989.

[WIL94] D. K. Wilde, *"A Library for Doing Polyhedral Operations"*, Technical report 785, IRISA-Rennes, France, 1994.

[WOL63] P. Wolfe, "Methods of nonlinear programming", Recent Advances in Mathemathical Programming, editeurs R.L. Graves and P. Wolfe, 1963.

[WRI97a] S. J. Wright. "Applying new optimization algorithms to model predictive control", dans J. Kantor, C. Garcia, and B. Carnahan, editeurs, *Proceedings of 5th Conference on Chemical Process Control*, CACHE, AIChE, Vol. 93, pp. 147-155, 1997.

[WRI97b] S. J. Wright. "*Primal-Dual Interior Point Methods",* SIAM Publications 1997.

# 10. Annexe 1 : Qualification des contraintes

Le problème motivant le travail présenté concerne l'optimisation multiparamétrique devant être résolue à chaque pas d'échantillonnage, pour laquelle $x_t$ représente le vecteur de paramètres de contexte, selon la notion introduite au chapitre 2 :

$$\begin{aligned}&\min_{\mathbf{k}_u} J_t = 0{,}5\,\mathbf{k}_u^T\, H\,\mathbf{k}_u + x_t^T F\mathbf{k}_u \\ &\text{sous les contraintes :} \begin{cases} A_{in}\mathbf{k}_u \le B_{in}x_t + b_{in} \\ A_{eq}\mathbf{k}_u = B_{eq}x_t + b_{eq}\end{cases}\end{aligned} \tag{A1.1}$$

Dans le problème (A1.1), les deux composantes qui par leur structure définissent la solution optimale sont la fonction de coût et le domaine faisable. La fonction de coût peut être décrite géométriquement par le point optimal généré en l'absence de contraintes et par les surfaces isocoût (des hyper-ellipses par exemple pour la relation (A1.1) à cause de la forme quadratique de la fonction de coût). Les contraintes - leur nombre et ensuite la topologie des domaines qu'elles définissent - offrent un capital de complexité virtuellement infinie pour le problème (A1.1). Notons par exemple que la discrimination entre les contraintes de type égalité et inégalité est artificielle (et superflue), car il est évident que chaque contrainte de type égalité peut être considérée comme l'intersection des deux demi-espaces, et en conséquence peut être réécrite par deux inégalités :

$$A_{eq}\mathbf{k}_u = b_{eq} \Leftrightarrow \begin{cases} A_{eq}\mathbf{k}_u \le b_{eq} \\ A_{eq}\mathbf{k}_u \ge b_{eq}\end{cases} \tag{A1.2}$$

Cependant, cette distinction est utile pour souligner le fait que les contraintes de type égalité signifient la suppression d'un degré de liberté et donc le fait que l'éventuel problème d'optimisation doit chercher les solutions optimales dans un espace de dimension restreinte.

## *10.1 Géométrie de la solution optimale*

L'ensemble des contraintes mixtes de (A1.1) définit un domaine convexe fonction de la valeur du vecteur $x_t$ :

$$P(x_t) = \left\{\mathbf{k}_u \middle| A_{in}\mathbf{k}_u \le B_{in}x_t + b_{in}\,;\, A_{eq}\mathbf{k}_u = B_{eq}x_t + b_{eq}\right\} \tag{A1.3}$$

La solution optimale d'un problème d'optimisation quadratique est conditionnée (comme on a pu le voir via le Théorème 2.1) par le fait que l'intersection entre le cône des directions faisables et celui des directions de descente est vide. Ceci peut être transformé par l'intermédiaire des conditions KKT en une condition d'optimalité facile à vérifier géométriquement pour le cas des contraintes linéaires, **à savoir que $-\nabla J(\mathbf{k}_u, x_t)$ doit appartenir au cône des gradients des contraintes actives**.

La Figure A1.1 décrit le problème de décision pour la loi prédictive à l'instant $t$, supposant que le vecteur des paramètres de contexte (l'état du système) $x_t$ est disponible. Deux points ont été illustrés, $\mathbf{k}_{u1}$ et $\mathbf{k}_{u2}$. Les vecteurs $\vec{a}_1$, $\vec{a}_2$ et $\vec{a}_3$ représentent les gradients des contraintes actives pour les deux points considérés. Avec la nature linéaire des contraintes, ils représentent en fait des directions perpendiculaires aux hyperplans :

$$\begin{aligned} a_1\mathbf{k}_u &= \beta_1 + b_1x_t \\ a_2\mathbf{k}_u &= \beta_2 + b_2x_t \\ a_3\mathbf{k}_u &= \beta_3 + b_3x_t \end{aligned} \tag{A1.4}$$

qui sont les frontières des trois demi-espaces, décrites par des inégalités présentes dans l'ensemble initial $A_{in}\mathbf{k}_u \le b_{in} + B_{in}x_t$. Sachant que le point optimal sans contraintes $\mathbf{k}_u^*(x_t)$ se trouve en dehors du domaine faisable, l'algorithme d'optimisation doit trouver le meilleur candidat sur les frontières de la région admissible. Si l'on considère le cas du point $\mathbf{k}_{u1}$, on peut construire le gradient de la fonction de coût en ce point et vérifier son inclusion dans le cône décrit par les vecteurs $\vec{a}_1$ et $\vec{a}_2$. Comme il ne fait pas partie de ce cône, la satisfaction des conditions KKT n'est pas assurée et donc il ne représente pas un point optimal. En revanche pour le point $\mathbf{k}_{u2}$, le gradient de la fonction de coût se trouve à l'intérieur du cône des directions perpendiculaires sur les contraintes actives et donc il n'existe pas de directions admissibles de descente. En conclusion il représente la solution optimale qui respecte les contraintes de la loi prédictive.

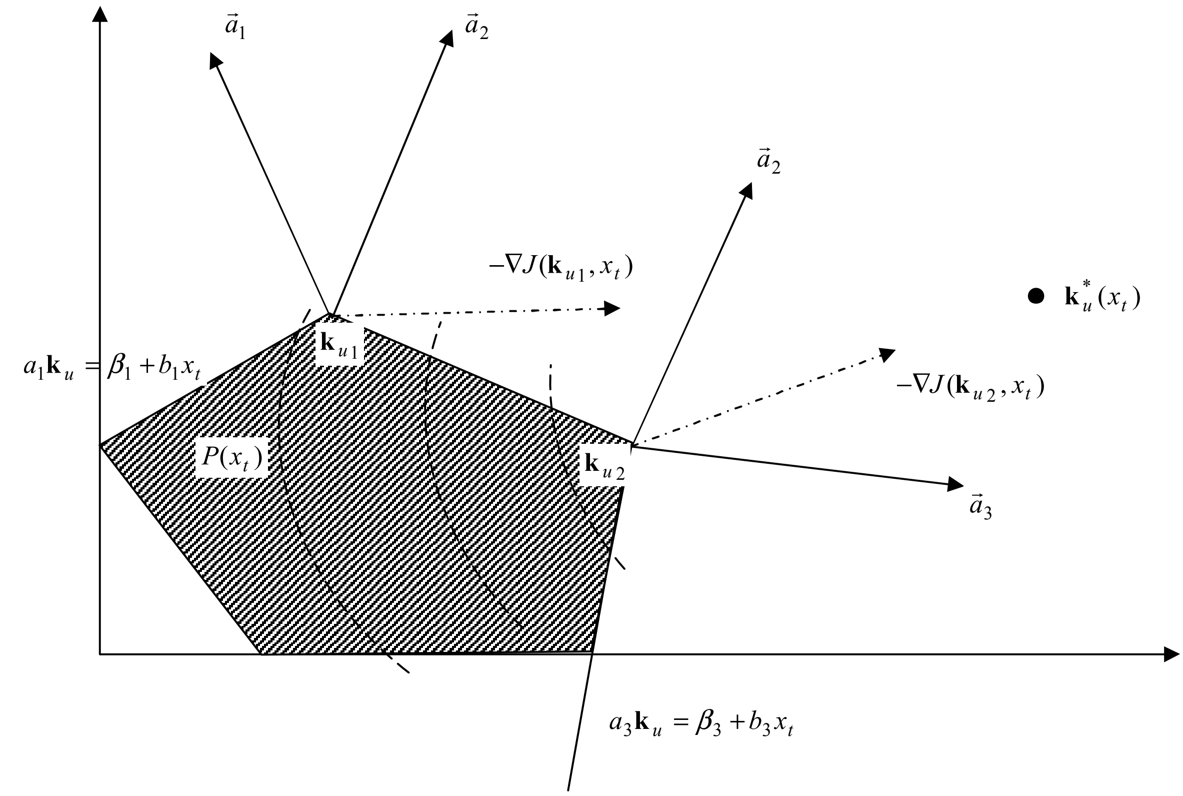



**Figure A1.1 : Illustration géométrique du problème de décision à l'instant *t* pour la loi prédictive.**

## *10.2 Classification des contraintes – Bases de l'interprétation géométrique des conditions d'optimalité*

Les conditions KKT d'optimalité peuvent être en fait élaborées à partir des conditions plus générales de Fritz-John [GSD04] pour lesquelles des propriétés supplémentaires sont établies pour les multiplicateurs correspondant à la fonction de coût. On rappelle ci-dessous les éléments permettant de décrire ces conditions à partir d'une classification des contraintes. Un des résultats préliminaires essentiels pour la compréhension des restrictions induites par un ensemble de contraintes linéaires est le lemme de Farkas.

**Lemme A1.1 (Farkas)[26] :** Soit une matrice $A \in \Re^{r \times p}$ et $c \in \Re^p$. Un seul des problèmes suivants possède une solution :

1. $\exists \mathbf{k}_u \in \Re^p$ tel que $A\mathbf{k}_u \leq 0$ et $c^T\mathbf{k}_u > 0$ (A1.5)

2. $\exists y \in \Re^r$ tel que $A^T y = c$ et $y > 0$ (A1.6)

■

De façon plus classique, étant donné un ensemble de directions représentées par les lignes de $A$ et une autre direction $c$, deux possibilités peuvent survenir, soit $c$ est une combinaison positive des directions données, soit $c$ réside en dehors du cône décrit par ces directions, et donc il existe un hyperplan $\{\mathbf{k}_u | b^T\mathbf{k}_u = 0\}$ qui les sépare. Cet hyperplan contient $t-l$ directions linéaires indépendantes parmi les $l$ initiales, avec $t$ donné par :

$$t = rang\left(\begin{bmatrix} A \\ c^T \end{bmatrix}\right) \tag{A1.7}$$

C'est pour cette raison que ce théorème est souvent dénommé théorème de séparation.

[26] La preuve peut être consultée dans [Sch86]

> **Définition A1.2 :** Soit $P \in \Re^p$ un ensemble non vide et $\mathbf{k}_u \in cl\, P$ (enveloppe frontière du domaine). Le *cône des tangentes* de $P$ en $\mathbf{k}_u$, noté $T(\mathbf{k}_u)$, est l'ensemble de toutes les directions $d$ telles que $d = \lim_{k\to\infty} \lambda_k (z_k - \mathbf{k}_u)$ avec $\lambda_k > 0$, $z_k \in P, \forall k$ et $z_k \to \mathbf{k}_u$. ■

Cette définition s'avère tout à fait générale ; dans le cas de contraintes linéaires, on peut démontrer que les limites du cône des tangentes coïncide sur un certain voisinage avec les frontières de $P$. La caractérisation de l'optimalité d'un point en fonction du cône des tangentes est donnée par la proposition suivante.

> **Proposition A1.3 :** Soit $P \in \Re^p$ un ensemble non vide, $\mathbf{k}_u \in P$, et une fonction de coût différentiable $J(\mathbf{k}_u) : \Re^p \to \Re$. Si $\mathbf{k}_u$ est l'optimum de la fonction $J(\mathbf{k}_u)$ sous les contraintes de type $\mathbf{k}_u \in P$, alors $D(\mathbf{k}_u) \cap T(\mathbf{k}_u) = \text{O}$, avec $D(\mathbf{k}_u) = \{d | \nabla J(\mathbf{k}_u) d < 0\}$ et $T(\mathbf{k}_u)$ est le cône de tangentes de $P$ en $\mathbf{k}_u$. ■

Cette proposition définit comme condition nécessaire d'optimalité le fait que l'intersection du cône des tangentes avec celui des directions de descente soit vide.

En revenant maintenant au cas des contraintes linéaires qui constitue spécifiquement les problèmes d'optimisation liés à la commande prédictive, il existe un résultat très important concernant la qualification des contraintes, constituant le premier maillon vers les conditions d'optimalité KKT. Il s'agit de la *qualification des contraintes d'Abadie*.

> **Lemme A1.4 :** Soit $P = \{\mathbf{k}_u | A_{in}\mathbf{k}_u \le b_{in}; A_{eq}\mathbf{k}_u = b_{eq}\}$ un ensemble non vide. Pour chaque $\mathbf{k}_u \in P$, il est possible de décomposer $A_{in} = \begin{bmatrix} \overline{A}_{in}^T & \widetilde{A}_{in}^T \end{bmatrix}^T$ et $b_{in} = \begin{bmatrix} \overline{b}_{in}^T & \widetilde{b}_{in}^T \end{bmatrix}^T$ selon $\overline{A}_{in}\,\mathbf{k}_u = \overline{b}_{in}$ et $\widetilde{A}_{in}\,\mathbf{k}_u < \widetilde{b}_{in}$. Alors on a $T(\mathbf{k}_u) = G(\mathbf{k}_u) \cap H$, avec $T(\mathbf{k}_u)$ le cône des tangentes de $P$ en $\mathbf{k}_u$, $G(\mathbf{k}_u) = \{d | \overline{A}_{in}\, d \le 0\}$ et $H = \{d | A_{eq} d = 0\}$. ■

On dispose désormais avec ce résultat de tous les éléments pour déduire les conditions nécessaires de type KKT en utilisant exclusivement des arguments géométriques.

> **Théorème (conditions KKT) A1.5 :** Soit $P(x_t) = \{\mathbf{k}_u | A_{in}\mathbf{k}_u \le B_{in}x_t + b_{in}; A_{eq}\mathbf{k}_u = B_{eq}x_t + b_{eq}\}$ un ensemble non vide avec $A_{in} \in \Re^{r\times p}; B_{in} \in \Re^{r\times n}; b_{in} \in \Re^r; A_{eq} \in \Re^{s\times p}; B_{eq} \in \Re^{s\times n}; b_{eq} \in \Re^s$ et le problème de minimisation $J(\mathbf{k}_u, x_t) = 0{,}5\,\mathbf{k}_u^T H\, \mathbf{k}_u + x_t^T F \mathbf{k}_u$ sous les contraintes $\mathbf{k}_u \in P(x_t)$. Si $\overline{\mathbf{k}}_u^*$ est une solution optimale pour le problème sous contraintes, alors il existe un vecteur $\mu(x_t) \in \Re^l$, $\mu(x_t) \ge 0$ avec $l$ le nombre de lignes de $\overline{A}_{in} \in \Re^{l\times p}$ dans la décomposition $A_{in} = \begin{bmatrix} \overline{A}_{in}^T & \widetilde{A}_{in}^T \end{bmatrix}^T$, $B_{in} = \begin{bmatrix} \overline{B}_{in}^T & \widetilde{B}_{in}^T \end{bmatrix}^T$ et $b_{in} = \begin{bmatrix} \overline{b}_{in}^T & \widetilde{b}_{in}^T \end{bmatrix}^T$ selon $\overline{A}_{in}\,\mathbf{k}_u = \overline{B}_{in}\, x_t + \overline{b}_{in}$ et $\widetilde{A}_{in}\,\mathbf{k}_u < \widetilde{B}_{in}\, x_t + \widetilde{b}_{in}$, et $\lambda(x_t) \in \Re^s$, tels que :
>
> $$H\overline{\mathbf{k}}_u^* + F^T x_t + A_{eq}^T \lambda(x_t) + \overline{A}_{in}^T \mu(x_t) = 0 \tag{A1.8}$$

**Preuve :** Si $\overline{\mathbf{k}}_u^*$ est une solution optimale, il vient de la Proposition A1.3 $D(\overline{\mathbf{k}}_u^*) \cap T(\overline{\mathbf{k}}_u^*) = \text{O}$. La classification d'Abadie des contraintes est validée car le domaine est décrit par des égalités et inégalités linéaires, et donc en conséquence du lemme A1.4, on a $T(\overline{\mathbf{k}}_u^*) = G(\overline{\mathbf{k}}_u^*) \cap H$, qui implique ensuite $D(\overline{\mathbf{k}}_u^*) \cap G(\overline{\mathbf{k}}_u^*) \cap H = \text{O}$. Le fait que cette intersection soit vide est équivalent à dire qu'il n'existe pas de solution au système :

$$\begin{bmatrix} \overline{A}_{in} \\ A_{eq} \\ -A_{eq} \end{bmatrix} d \le 0 \text{ et } -(H\mathbf{k}_u^* + F^T x_t)^T d > 0 \tag{A1.9}$$

Par le lemme de Farkas, le fait qu'aucune solution pour (A1.8) n'existe implique l'existence d'une solution pour :

$$\begin{bmatrix} \overline{A}_{in} \\ A_{eq} \\ -A_{eq} \end{bmatrix}^T y = -(H\mathbf{k}_u^* + F^T x_t) \text{ et } y \geq 0 \tag{A1.10}$$

qui finalement peut être réécrit avec $y = \begin{bmatrix} \lambda^T & \mu_1^T & \mu_2^T \end{bmatrix}^T \geq 0$ sous la forme :

$$\begin{aligned} & H\mathbf{k}_u^* + F^T x_t + \overline{A}_{in}^T \lambda + \overline{A}_{eq}^T \mu_1 - \overline{A}_{eq}^T \mu_2 = 0 \Leftrightarrow \\ & \Leftrightarrow H\mathbf{k}_u^* + F^T x_t + \overline{A}_{in}^T \lambda + \overline{A}_{eq}^T \mu = 0 \end{aligned} \tag{A1.11}$$

où l'on a noté $\mu = \mu_1 - \mu_2$. Il faut remarquer qu'un certain point $\mathbf{k}_u^*$ peut vérifier les conditions d'optimalité pour plusieurs combinaisons des paramètres de contexte $x_t$. Même si les conditions d'optimalité sont vérifiées avec le même ensemble d'inégalités saturées, les multiplicateurs associés $\lambda(x_t), \mu(x_t)$ peuvent donc varier en fonction de $x_t$. ■

Un résumé des évolutions caractérisant la construction des conditions d'optimalité est donné Figure A1.2.

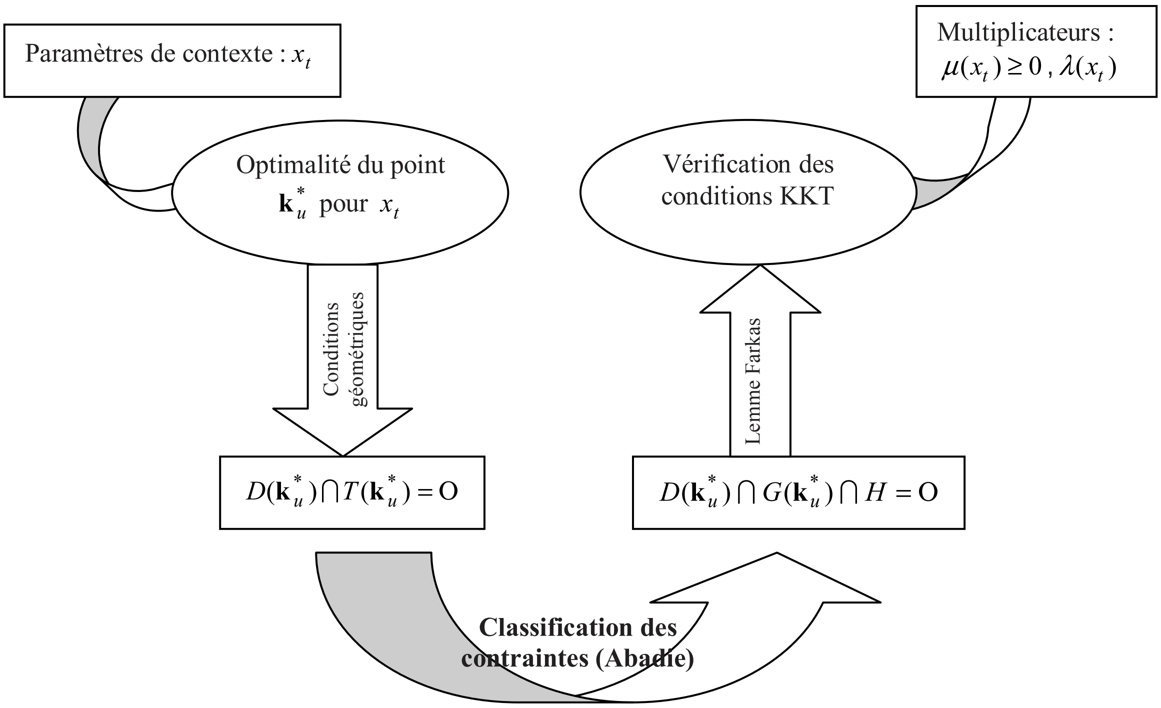



**Figure A1.2 : Schéma bloc des implications géométriques définissant les conditions d'optimalité.**

### *10.3 Influence des paramètres sur la géométrie des solutions*

Le vecteur des paramètres influence les deux éléments cruciaux du problème d'optimisation (A1.1) : la fonction de coût et les contraintes. Géométriquement leurs implications se retrouvent dans :

- L'apparition des paramètres dans la partie linéaire de la fonction de coût, induisant une dépendance linéaire de l'optimum sans contraintes $\mathbf{k}_u^*(x_t) = H^{-1} F^T x_t$. Il est important de noter que les courbes isocoût gardent la même forme car elles sont influencées exclusivement par la forme du Hessien $H$.
- Le domaine faisable correspondant au problème d'optimisation change avec l'évolution des paramètres. La modification de la partie affine d'une inégalité implique un déplacement de la frontière du demi-espace qu'elle définit.

Du point de vue des conditions d'optimalité, la dépendance de la solution optimale envers le vecteur des paramètres est transparente en ce qui concerne la modification des valeurs des multiplicateurs de Lagrange de (A1.11) ou plus spécifiquement dans :

- La décomposition en contraintes saturées $\overline{A}_{in}\mathbf{k}_u = \overline{B}_{in}x_t + \overline{b}_{in}$ et contraintes vérifiées $\widetilde{A}_{in}\mathbf{k}_u < \widetilde{B}_{in}x_t + \widetilde{b}_{in}$ pour un certain point $\mathbf{k}_u$
- La modification du gradient de la fonction de coût $\nabla J(\mathbf{k}_u, x_t) = H\mathbf{k}_u + F^T x_t$

La figure A1.3 propose une illustration des changements qui interviennent avec l'évolution du vecteur des paramètres. Elle met en évidence le déplacement et le changement de la forme du domaine faisable conjointement avec le décalage de l'optimum sans contraintes. Les courbes isocoût et les gradients des contraintes restent inchangés.

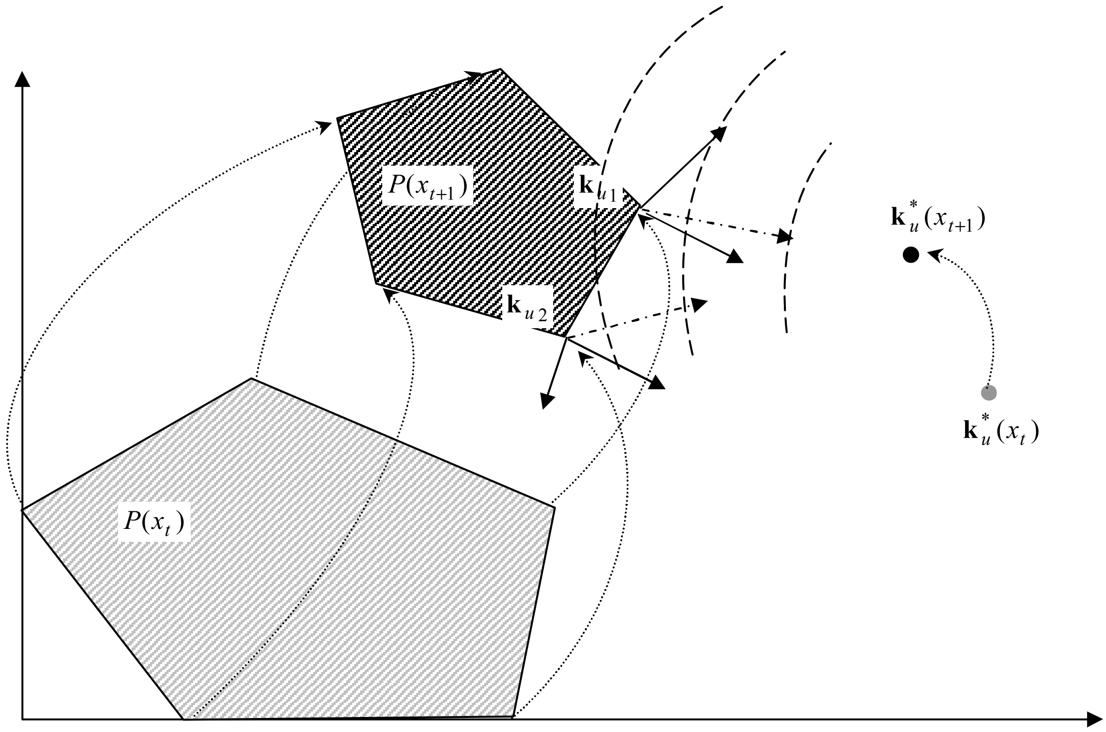



**Figure A1.3 : Illustration géométrique du problème de décision à l'instant $t+1$ pour la loi prédictive.**

Même si l'ensemble de contraintes dépend des valeurs mesurées pour $x_t$, le domaine décrit reste convexe. Ceci constitue un aspect important pour garantir l'existence de la solution optimale, mais les informations qu'il offre sur la topologie effective de l'espace faisable reste limitées. Par exemple, il peut contenir des directions selon lesquelles on peut avancer infiniment dans la quête des points admissibles améliorant la fonction de coût. Ce type de détail peut jouer un rôle important lors de la mise en œuvre d'une loi de commande car l'existence d'une telle direction ne dépend pas de l'évolution des paramètres.

# 11. Annexe 2 : Algorithmes de détermination de la double description

Les développements du chapitre 3 ont montré qu'il était intéressant de construire la double description d'un polyèdre, l'une de ces deux descriptions s'avérant mieux adaptée pour des opérations géométriques particulières. Des algorithmes de passage existent, permettant d'obtenir à partir de la représentation par contraintes la représentation par générateur et inversement. Cette annexe propose après une brève classification de ces algorithmes le squelette de l'algorithme de Chernikova [Che64], l'un des plus classiques pour ce type d'opération.

> **Proposition A2.1 :** Pour un polyèdre $P \subset \Re^p$ représenté dans sa version homogène, on définit le polyèdre polaire par :
>
> $$P^* = \left\{ z \in \Re^p \middle| \forall y \in P, z^T y \geq 0 \right\} \tag{A2.1}$$
>
> avec les propriétés suivantes :
>
> - $P^*$ est un polyèdre
> - $P^{**} = P$
> - il existe une correspondance 1 :1 entre les faces d'ordre $k$ de $P$ et les faces d'ordre $p-k$ de $P^*$.

Dans le processus de passage d'une représentation d'un domaine polyédral à la représentation duale, il ne faut se préoccuper que d'une seule construction (contraintes vers générateurs) car avec la dualité, les mêmes algorithmes peuvent être employés dans le sens inverse. Pour vérifier la dualité, il suffit de vérifier (A2.1) pour :

$$\begin{aligned} C &= \left\{ y \middle| y = L\boldsymbol{\lambda} + R\boldsymbol{\rho}; \quad \boldsymbol{\rho} \geq 0 \right\} \\ C^* &= \left\{ z \middle| z = A_{eq}^T \boldsymbol{\alpha} + A_{in}^T \boldsymbol{\beta}; \quad \boldsymbol{\beta} \geq 0 \right\} \end{aligned} \tag{A2.2}$$

En effet :

$$\begin{aligned} z^T y &= (A_{eq}^T \boldsymbol{\alpha} + A_{in}^T \boldsymbol{\beta})^T (L\boldsymbol{\lambda} + R\boldsymbol{\rho}) = (\boldsymbol{\alpha}^T A_{eq} + \boldsymbol{\beta}^T A_{in})(L\boldsymbol{\lambda} + R\boldsymbol{\rho}) = \\ &= \boldsymbol{\alpha}^T (A_{eq} L\boldsymbol{\lambda} + A_{eq} R\boldsymbol{\rho}) + \boldsymbol{\beta}^T (A_{in} L\boldsymbol{\lambda} + A_{in} R\boldsymbol{\rho}) \geq 0 \end{aligned} \tag{A2.3}$$

et donc $C, C^*$ sont duales comme conséquence de la proposition A2.1.

Les algorithmes de passage se répartissent en deux grandes catégories : les méthodes par pivot et celles basées sur l'algorithme de la double description de Motzkin [Wil94]. On retrouve dans la première catégorie les algorithmes utilisant les techniques d'élimination de Fourier-Motzkin [Sch86] généralisant l'élimination gaussienne pour les systèmes d'inéquations. Le procédé nécessite des projections successives, stratégies connues pour sa complexité élevée, $O(p^{2^k})$, avec $p$ le nombre de contraintes dans l'ensemble initial et $k$ le nombre de projections [LeF94]. Un autre problème lié à cette méthode est la nécessité d'implémentation d'un mécanisme d'élimination de redondances.

La deuxième catégorie de méthodes est celle qui nous intéresse et qui a été implémentée effectivement sous forme de scripts Matlab. La méthode part d'un cône ne subissant pas de contrainte et induit de façon itérative la transformation de l'ensemble des générateurs vers la solution complète. Un tel algorithme a été proposé initialement par Motzkin [MRTT53] et perfectionné par Chernikova, surtout en ce qui concerne la manière dont les incidences sont stockées. Nous avons implémenté une version améliorée de cet algorithme en tenant compte des observations faites par [FQ88] et plus spécialement par [LEV94], permettant ainsi d'augmenter les performances tout particulièrement en ce qui concerne la prévention de l'apparition de redondance.

L'idée de base est de traiter chaque contrainte qui vient d'être ajoutée en considérant la division induite dans l'ensemble des rayons existants. Si la contrainte est définie dans un espace homogène par un vecteur $c^T$, on distingue :

- $R^+$ les rayons qui vérifient la contrainte $c^T \mathbf{k}_u < 0$

- $R^-$ les rayons qui vérifient la contrainte $c^T \mathbf{k}_u > 0$
- $R^0$ les rayons qui saturent la contrainte $c^T \mathbf{k}_u = 0$

Un nouveau cône est généré avec les éléments de $R^+$, $R^0$ et une combinaison convexe de rayons dans $R^+$ et $R^-$ qui sature la nouvelle contrainte. Une précaution doit être prise pour cette génération de nouveaux rayons car la frontière qui les contient ne doit pas permettre l'inclusion des autres générateurs redondants.

**Squelette d'un algorithme de type Chernikova**

**Données d'entrée :** matrices $A_{eq}$ et $A_{in}$ telles que $P = \{x | A_{eq} x = 0; A_{in} x \leq 0\}$

**Données de sortie :** les ensembles $L = \{l_1, \ldots, l_{d\,L}\}, R = \{r_1, \ldots, r_{d\,R}\}$ des rayons uni et bi-directionels

**Initialisation :** l'ensemble $L = \{e_1, \ldots, e_r\}$ avec $e_i$ les éléments faisant partie d'une base canonique et $R = \{0\}$.

Pour toute contrainte décrite par un vecteur $c^T$ représentant une ligne de $A_{eq}$ ou $A_{in}$ :

**A.** Si la contrainte courante est de type égalité :

**a.** S'il existe $l_k$ tel que $\left|c^T l_k\right| > \varepsilon$ alors les ensembles $R$ et $L$ deviennent :

$\widetilde{R} = \{\widetilde{r}_1, \ldots, \widetilde{r}_{d\,R}\}$ et $L = \{\widetilde{l}_1, \ldots, \widetilde{l}_{k-1}, \widetilde{l}_{k+1}, \ldots, \widetilde{l}_{d\,L}\}$ avec :

$\widetilde{r}_i = \mu r_i + \lambda l_k, \mu > 0, \quad t.q. \quad c^T \widetilde{r}_j = 0, i = 1,.., d_R$

$\widetilde{l}_j = \alpha l_j + \beta l_k, \forall j \neq k, \quad t.q. \quad c^T \widetilde{l}_j = 0, \forall j \neq k$

Après une re-indexation $d_L = d_L - 1$

**b.** Si tout $l_k$ satisfait $\left|c^T l_k\right| \leq \varepsilon$ alors l'ensemble $L$ reste inchangé et $R = R^0 \cup \overline{R}$ avec :

$$\overline{R} = \left\{ y \in \Re^r \,\middle|\, \begin{cases} c^T y = 0, y = \alpha y_1 + \beta y_2, \alpha > 0, y_1 \in R^+, y_2 \in R^-, \\ (y_1, y_2) \text{ sont adjacentes} \\ y \text{ sature au moins } r\text{-}n_L\text{-2 contraintes déjà traitées} \end{cases} \right\}$$

**B.** Si la contrainte courante est de type inégalité :

**a.** S'il existe $l_k$ tel que $\left|c^T l_k\right| > \varepsilon$ alors les ensembles $R$ et $L$ deviennent :

$\widetilde{R} = \{\widetilde{r}_1, \ldots, \widetilde{r}_{d\,R}, \widetilde{l}_k\}$ et $L = \{\widetilde{l}_1, \ldots, \widetilde{l}_{k-1}, \widetilde{l}_{k+1}, \ldots, \widetilde{l}_{d\,L}\}$ avec :

$\widetilde{r}_i = \mu r_i + \lambda l_k, \mu > 0, \quad t.q. \quad c^T \widetilde{r}_j = 0, i = 1,.., d_R$

$\widetilde{l}_k = sign(c^T l_k) l_k$

$\widetilde{l}_j = \alpha l_j + \beta l_k, \forall j \neq k, \quad t.q. \quad c^T \widetilde{l}_j = 0, \forall j \neq k$

Après une re-indexation $d_R = d_R + 1$ et $d_L = d_L - 1$.

**b.** Si tout $l_k$ satisfait $\left|c^T l_k\right| \leq \varepsilon$ alors l'ensemble $L$ reste inchangé et l'ensemble des rayons unidirectionnels devient $R = R^0 \cup \overline{R}$ avec $\overline{R}$ comme auparavant (point A.b.).

# 12. Annexe 3 : Domaines faisables et ensembles invariants

La faisabilité d'une loi de commande MPC/GPC est intimement liée à la garantie de satisfaction de contraintes lors de l'évolution du système asservi. Cet aspect est conceptuellement différent de l'acception classique de ce terme dans le milieu mathématique, où l'on utilise le terme 'infaisabilité' à propos d'un programme d'optimisation sous contraintes. Les paragraphes suivants donnent les définitions et concepts de base utiles pour la mise en œuvre de conditions ultérieures permettant de garantir la faisabilité d'une loi de commande prédictive.

## *12.1 Quelques notations mathématiques*

Pour un problème d'optimisation (PO) avec contraintes :

$$\begin{aligned}&\min f(x)\\&\text{sous les contraintes}\quad Ax \geq b\end{aligned} \tag{A3.1}$$

l'ensemble faisable est donné par :

$$F = \{x | Ax - b \geq 0\} \tag{A3.2}$$

**Définition A3.1 :** (PO) est dit *faisable* si le domaine faisable $F$ est non vide. Un point $x \in F$ est dénommé une *solution faisable* pour (PO).

Pour un problème d'optimisation multiparamétrique (mpPO) :

$$\begin{aligned}&\min f(x, p)\\&\text{sous les contraintes}\quad Ax \geq b + Bp\end{aligned} \tag{A3.3}$$

l'ensemble faisable est donné par :

$$F(p) = \{x | Ax - b - Bp \geq 0\} \tag{A3.4}$$

**Définition A3.2 :** (mpPO) est dit *faisable pour le paramètre p* si le domaine faisable $F(p)$ est non vide. Un point $x \in F(p)$ est dénommé une *solution faisable* pour (mpPO) ■

Même si ces définitions ne peuvent pas s'appliquer sous cette forme pour des lois de commande sous contraintes, leur esprit est important car la conception de lois de commande prédictive est basée sur la résolution d'un problème d'optimisation sous contraintes.

## *12.2 Ensembles invariants*

**Définition A3.3 :** Un domaine $D \subset \Re^n$ est dit *positif invariant* pour un système LTI :

$$x_{k+1} = Ax_k \tag{A3.5}$$

si pour tout $x_0 \in D$ la solution $x_k \in D$ pour $k > 0$. Si $x_0 \in D$ implique $x_k \in D$ pour tout $k \in Z$ alors on dit que $D$ est *invariant.* ■

**Définition A3.4 :** Le domaine $D \subset \Re^n$ est dit *invariant par bouclage* pour un système LTI :

$$x_{k+1} = Ax_k + Bu_k \tag{A3.6}$$

s'il existe une loi de commande :

$$u_k = Kx_k \tag{A3.7}$$

qui assure l'existence de la solution pour $k > 0$ et dans le même temps fait que $D$ est positif invariant pour le système bouclé. ■

**Théorème A3.5 :** Considérons le système LTI :

$$x_{k+1} = Ax_k + Bu_k \tag{A3.8}$$

et le domaine $D \subset \Re^n$ contenant l'origine. $D$ est invariant par rapport à la commande si et seulement si pour tout $x \in v(D)$ (l'ensemble des points extrêmes) il existe $u(x)$ tel que :

$$Ax_k + Bu_k \in D \tag{A3.9}$$

Ce théorème peut être étendu au cas où la commande est contrainte, $u \in U$ , et $U$ est un domaine convexe polyédral contenant l'origine, en remplaçant dans l'énoncé du théorème '$u(x)$' par '$u(x) \in U$ '.

**Remarque fondamentale A3.6 :** On se restreint ici à l'utilisation du concept d'invariance pour les systèmes discrets car ce type de modèles est à la base de la philosophie MPC. Cette restriction représente en fait un avantage déterminant car, pour le cas discret, l'invariance positive d'un ensemble est équivalente à la '$A$-invariance' et peut être exprimée en termes de propriétés de cette matrice[27]. ■

## *12.3 Ensembles contrôlables*

Pour un système sous contraintes donné, avant d'analyser une loi de commande MPC, quelques propriétés intrinsèques peuvent être décrites, notamment vis-à-vis de l'évolution de la commande. Soit le système :

$$\Sigma: \quad x_{k+1} = Ax_k + Bu_k, k \geq 0, x_0 = x^0, u_k \in U, x_k \in X \tag{A3.10}$$

l'origine étant incluse dans les domaines polyédraux $U$ et $X$.

**Définition A3.7 :** Un ensemble $C \subset X$ est dit contrôlable (pour le système $\Sigma$ ) s'il satisfait :

$$x^0 \in C \Rightarrow \exists u_k \in U, \forall k \geq 0 \ \ t.q. \ \text{si } x_0 = x^0 \text{ alors } x_k \in X \ \text{ et } \lim_{k \to \infty} x_k = 0 \tag{A3.11}$$

**Définition A3.8 :** Un ensemble $IC \subset X$ contrôlable est dit $U$-invariant s'il satisfait (A3.11) et :

$$x_k \in IC, \forall k \geq 0 \tag{A3.12}$$

Notons qu'un ensemble compact contrôlable $IC \subset X$ est $U$-invariant par rapport à $X$ si et seulement si chaque sommet de $IC$ peut être amené en temps fini dans $\text{int}(IC)$[28].

**Proposition A3.9 :** Pour tout ensemble contrôlable $C \subset X$ de ( $\Sigma$ ) il existe un ensemble contrôlable et $U$-invariant $IC \subset X$ de ( $\Sigma$ ) tel que $C \subset IC$ ■

**Preuve :** Soit $IC$ l'ensemble de toutes les trajectoires partant de $C$ . Il est immédiat que $C \subset IC \subset X$ . L'invariance peut être démontrée pour les points de $IC$ en tronquant une trajectoire partant de $C$ . ■

**Exemple A3.10 :** Considérons la version discrète d'un double intégrateur [GC86] :

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u_k, k \geq 0,$$

$$u_k \in U = \{u \in \Re \mid -1 \leq u \leq 1\}, \quad x_k \in X = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| -5 \leq x_1 \leq 5; -5 \leq x_2 \leq 5 \right\}$$

La Figure A3.1a représente un domaine $C = conv\left\{ \begin{bmatrix} -2.5 \\ 0 \end{bmatrix}; \begin{bmatrix} 2.5 \\ 0 \end{bmatrix}; \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}; \begin{bmatrix} 0 \\ -0.5 \end{bmatrix} \right\}$ contrôlable mais non invariant, car pour amener par exemple le sommet $[2.5 \quad 0]^T$ à l'origine il faut sortir du domaine. La Figure A3.1b en

[27] Le cas des systèmes continus est plus complexe car l'invariance positive d'un ensemble $S$ est vérifiée ssi $S \subset e^{At} S, \forall t \geq 0$ .

[28] La notation $\text{int}(IC)$ fait référence à l'ensemble des points intérieurs au domaine $IC$ , frontières non comprises.

revanche présente un autre domaine contrôlable $C = conv\left\{\begin{bmatrix}-2\\2\end{bmatrix};\begin{bmatrix}1\\0\end{bmatrix};\begin{bmatrix}2\\-2\end{bmatrix};\begin{bmatrix}-1\\0\end{bmatrix}\right\}$ et invariant pour lequel on peut vérifier que tous les sommets peuvent être amenés à l'origine en temps fini.

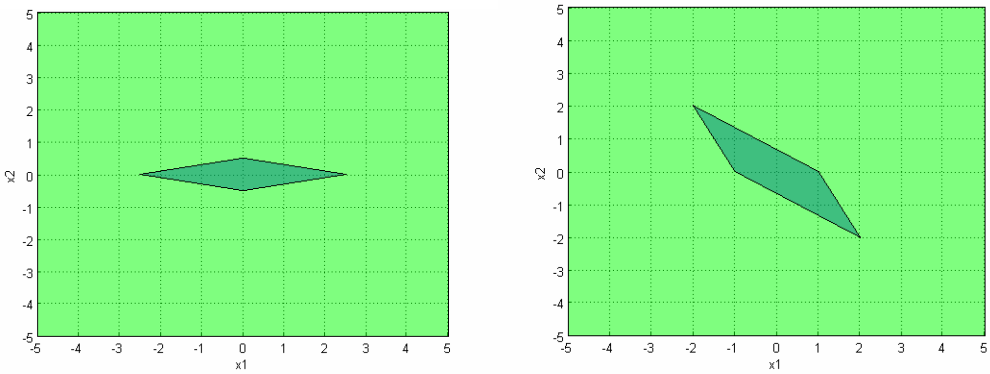



**Figure A3.1 : a) Exemple d'un ensemble contrôlable mais non invariant**
**b) Exemple d'un ensemble contrôlable et invariant**

> **Notation A3.11 :** L'ensemble maximal contrôlable *U*-invariant est noté $IC_{\max}$. Pour tout ensemble contrôlable *U*-invariant $IC \subset X$, on a $IC \subset IC_{\max}$ ■

Il faut remarquer que même si les ensembles $X, U$ sont fermés, $IC_{\max}$ peut être un ensemble ouvert. Cette propriété vient du fait que l'espace contrôlable pour un système avec contraintes sur les conditions initiales est ouvert.

Un premier objectif dans la caractérisation de tout phénomène d'infaisabilité est de définir de façon correcte cet ensemble $IC_{\max}$, qui fournit finalement une mesure de la liberté dont on dispose vis-à-vis des performances d'une loi de commande pour un système sous contraintes. La principale difficulté vient du fait que ces définitions restent théoriques et que les détails ne sont pas vraiment constructifs. Ainsi, une méthode possible consiste à construire par itération des domaines avec l'espoir d'approcher au mieux $IC_{\max}$. Si la matrice $A$ est inversible, les résultats suivants donnent la base d'une construction algorithmique d'une approximation de $IC_{\max}$.

> **Définition A3.12 :** L'ensemble contrôlable pour *t* pas, $IC_t \subset X$, est représenté par l'ensemble de tous les états initiaux qui peuvent être amenés en *t* pas dans un ensemble 'terminal' $IC_0 \subset X$ (qui peut être l'origine) avec des commandes $u_k \in U, \forall \leq k \leq t-1$. ■

Formellement, si $IC_0 = \{0\}$ et la matrice $A$ est inversible, alors on peut écrire :

$$IC_t = \left[A^{-1}(IC_{t-1}) \oplus A^{-1}B(-U)\right] \cap X \tag{A3.13}$$

où $P1 \oplus P2$ représente la somme de Minkowski[29] de deux ensemble $P_1, P_2$ et $M(P) = \{Mp \mid p \in P\}$[30].

**Exemple A3.13 :** Pour le système de l'exemple A3.10, on peut construire itérativement en suivant la définition précédente les ensembles contrôlables (Figure A3.2).

> **Définition A3.14 :** L'ensemble des états qui peuvent être amenés à l'origine en un nombre fini quelconque de pas est :
>
> $$IC_\infty = \bigcup_{t=0}^{\infty} IC_t = \lim_{t\to\infty} IC_t \tag{A3.14}$$

[29] Implémenté dans PPP par la fonction P=mink(P1,P2)

[30] L'opération décrite dans la définition précédente a besoin d'une double représentation des domaines polyédraux. Opérer au niveau des générateurs représente un avantage mais la multiplication de points redondants est un phénomène qui doit être pris en compte. Une double représentation évite alors cet inconvénient. De même, l'intersection successive bénéficie de cette double représentation.

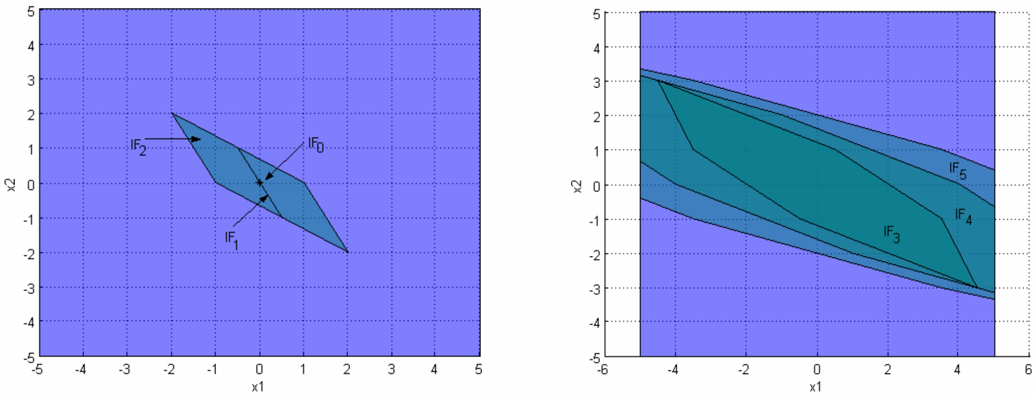



**Figure A3.2 : a) Ensembles contrôlables pour 0, 1 et 2 pas d'échantillonnage**
**b) Ensembles contrôlables pour 3, 4 et 5 pas**

**Proposition A3.15 :** Pour $IC_t, t = 0,1,2,...$, les propriétés suivantes sont vérifiées :

i) $IC_t$ est $U$-invariant et contient l'origine

ii) $IC_t \subset IC_{t+1}$

iii) $\forall t > n, \dim(IC_t) = n$ si la paire $(A, B)$ est contrôlable et $U \subset \Re^m, X \subset \Re^n$ ont des dimensions maximales $\dim(U) = m, \dim(X) = n$

iv) $IC_t \subset IC_\infty \subset IC_{\max}$

v) $IC_\infty = IC_{\max}$ si la paire $(A, B)$ est contrôlable et $U \subset \Re^m, X \subset \Re^n$ ont des dimensions maximales $\dim(U) = m, \dim(X) = n$.

vi) $IC_t = IC_{\max}$ implique que $IC_{\max}$ est un polytope (on rappelle que $U$ et $X$ sont des polytopes et par conséquent $IC_t$ garde cette propriété). ■

**Théorème A3.16 :** Si $IC_t = IC_{t+1}$ alors $IC_{\max} = IC_t$.

**Preuve :**

$$IC_t = IC_{t+1} \Rightarrow IC_{t+2} = \left[A^{-1}(IC_{t+1}) \oplus A^{-1}B(-U)\right] \cap X = \left[A^{-1}(IC_t) \oplus A^{-1}B(-U)\right] \cap X$$

$$\Rightarrow IC_{t+2} = IC_{t+1} \Rightarrow IC_t = IC_{t+1} = IC_{t+2} = \ldots \Rightarrow IC_t = \bigcup_{k=0}^{\infty} IC_k \Rightarrow IC_t = IC_\infty$$ ■

**Exemple A3.17 :** Reprenons l'exemple du double intégrateur pour le jeu de contraintes :

$$u_k \in U = \{u \in \Re \,|\, -1 \leq u \leq 1\}, \qquad x_k \in X = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| -5 \leq x_1 \leq 5; -5 \leq x_2 \leq 5 \right\} \tag{A3.15}$$

Le développement de la série des ensembles contrôlables conduit à $IC_t = IC_{t+1}$ et donc fournit $IC_{\max}$ pour $k = 10$ itérations (Figure A3.3a) (arrêt lorsque deux domaines consécutifs se superposent).

**Exemple A3.18 :** Si les contraintes initiales sont différentes, le domaine maximal sera obtenu bien sûr après un nombre différent d'itérations. Par exemple, pour :

$$u_k \in U = \{u \in \Re \,|\, -1 \leq u \leq 1\}, \qquad x_k \in X = conv\left\{ \pm \begin{bmatrix} 25 \\ 5 \end{bmatrix}; \pm \begin{bmatrix} -25 \\ 5 \end{bmatrix} \right\} \tag{A3.16}$$

on obtient le domaine maximal après $k = 15$ itérations (Figure A3.3b).

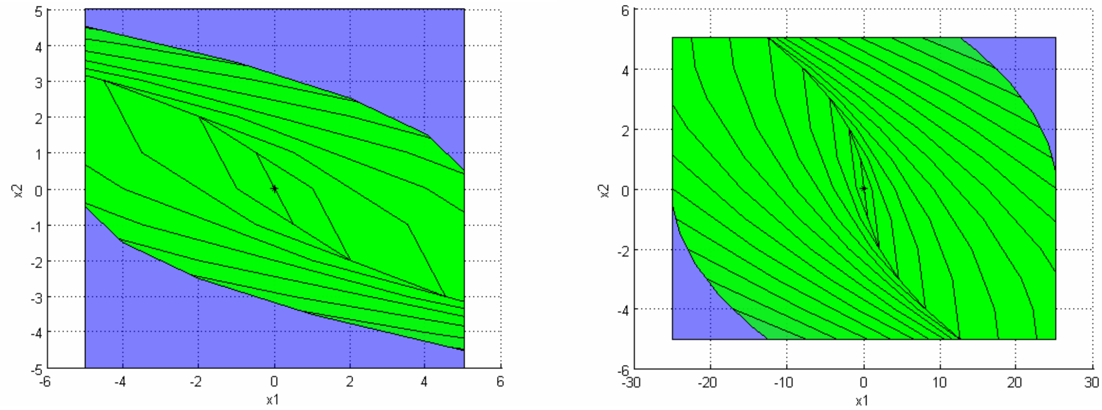



**Figure A3.3 : Ensembles contrôlables compacts pour différentes contraintes**
**a) Exemple 5.22 b) Exemple 5.23**

Les développements précédents ont donné les éléments nécessaires pour l'élaboration d'une procédure itérative de détermination de l'ensemble maximal contrôlable. Ainsi le théorème A3.16 fournit un critère d'arrêt lorsque $IC_{\max}$ est un compact. Malheureusement, comme mentionnée auparavant, $IC_{\max}$ peut être ouvert de sorte que seule une approximation à un $\varepsilon$ près par un ensemble fermé peut être trouvée. Le critère d'arrêt dans le cas général est donné via le théorème suivant.

> **Théorème A3.19 :** Pour tout $\varepsilon \geq 0$ il existe $T = T(\varepsilon)$ tel que pour $\forall t \geq T$ :
>
> $$IC_t \subseteq IC_{\max} \subseteq (1/1-\varepsilon)IC_t \text{ ou de façon équivalente } (1-\varepsilon)IC_{\max} \subseteq IC_t \subseteq IC_{\max} \quad \text{(A3.17)}$$

La preuve peut être consultée dans [CG86][31].

**Exemple A3.20 :** Reprenons toujours le même exemple mais cette fois avec les contraintes :

$$u_k \in U = \{u \in \Re \mid -1 \leq u \leq 1\}, \qquad x_k \in X = conv\left\{\pm\begin{bmatrix}25\\0\end{bmatrix};\pm\begin{bmatrix}0\\5\end{bmatrix}\right\}$$

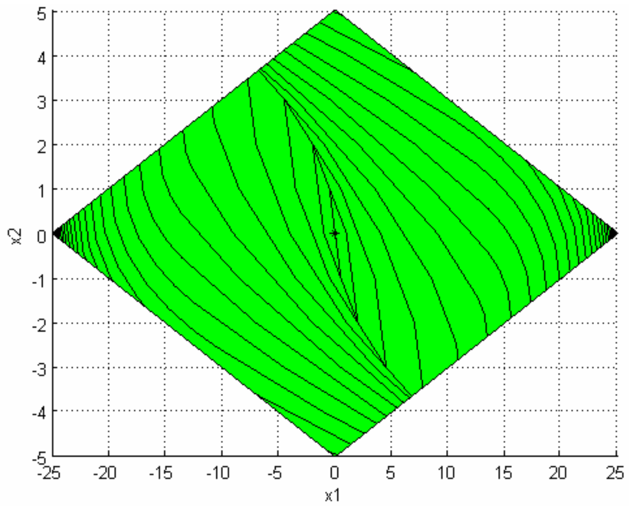



**Figure A3.4 : Ensemble contrôlable ouvert déterminé avec une précision $\varepsilon$**

[31] Dans l'article mentionné, la convergence de $IC_t$ vers $IC_{\max}$ est démontrée et la valeur de $T(\varepsilon)$ peut être choisie par :

$$T(\varepsilon) > \left(1 + \frac{1}{\log(1/\lambda)}\log\left[\frac{d(IC_n, IC_{2n})}{(1-\lambda)\log(1/(1-\varepsilon))}\right]\right)n$$

où $d(IC_n, IC_{2n})$ représente la 'distance' entre deux ensemble compacts et $\lambda$ est une valeur en liaison avec la forme de $X, U$ .

L'ensemble initial $X$ (qui est compact) n'est pas contrôlable et il est impossible d'emmener tous ses sommets dans $\text{int}(X)$ en temps fini (pour les sommets {25,0} ; {-25,0} la seule commande admissible est $u=0$). A l'exception de ces deux points, tous les autres sommets peuvent être amenés en 0 en temps fini et font donc partie de $IC_{\max}$. La Figure A3.4 donne une illustration de la couverture progressive de l'espace admissible.

**Remarque A3.21 :** Tous les développements ont été faits en supposant que la matrice d'état $A$ est inversible, car dans le cas contraire, l'inversion nécessaire pour la construction récursive de l'espace contrôlable devient impossible. Généralement pour des systèmes discrétisés à partir d'un système continu cette condition est satisfaite. De même, les résultats précédents ne sont pas utilisables en présence de contraintes mixtes impliquant entrées passées et états. ■

Pour éliminer cet inconvénient, les développements suivants proposent une construction de l'espace maximal contrôlable lorsque :

$$\begin{aligned} &x_{k+1} = Ax_k + Bu_k, k \geq 0, x_0 = x^0, \\ &Cx_k + Du_k \leq \gamma \end{aligned} \tag{A3.18}$$

Il est évident que le cas précédent $x_k \in X, u_k \in U$ est un cas particulier de ce problème. Ajoutons comme hypothèse que la combinaison $[x_k \quad u_k]^T = 0$ appartient à l'ensemble polyédral $Cx_k + Du_k \leq \gamma$.

L'ensemble :

$$IC_t = \left\{x^0 \middle| \exists u_k, 0 \leq k \leq t-1, \quad t.q. \quad Cx_k + Du_k \leq \gamma \text{ et } x_t = 0\right\} \tag{A3.19}$$

fournit dans l'espace étendu entrée-état l'ensemble des combinaisons accessibles par rapport aux contraintes :

$$Y = \left\{\begin{bmatrix} x \\ u \end{bmatrix} \middle| [C \quad D]\begin{bmatrix} x \\ u \end{bmatrix} \leq \gamma\right\} \tag{A3.20}$$

et également les combinaisons telles que l'état puisse atteindre $IC_t$ en une seule itération :

$$T_{t+1} = \left\{\begin{bmatrix} x \\ u \end{bmatrix} \middle| [A \quad B]\begin{bmatrix} x \\ u \end{bmatrix} \in IC_t\right\} \tag{A3.21}$$

En notant $\text{Pr}_x\{D\}, D \subset \Re^n \times \Re^m$ la projection de $D$ sur $\Re^n$, le résultat suivant présente une méthode constructive permettant la description de l'ensemble contrôlable sur $n$ pas.

**Théorème A3.22 :** $IC_{t+1} = \text{Pr}_x\{T_{t+1} \cap Y\}$[32] [33] (A3.22)

**Preuve :**

Sachant que :

$$\forall x^0 \in IC_{t+1}, \exists k_u = \{u_0, u_1, ..., u_t\}\, t.q.\, x_1 \in IC_t, x_2 \in IC_{t-1}, ... x_{t+1} \in IC_0,$$

alors : $\begin{bmatrix} x_k \\ u_k \end{bmatrix} \in Y, 0 \leq k \leq t$.

Or : $x_1 = [A \quad B]\begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \in IC_t \Rightarrow \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \in T_{t+1}$ et donc $\begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \in T_{t+1} \cap Y$.

Puisque : $\text{Pr}_x\left\{\begin{bmatrix} x_0 \\ u_0 \end{bmatrix}\right\} = x_0$, on peut écrire $IC_{t+1} \subset \text{Pr}_x\{T_{t+1} \cap Y\}$.

[32] Un résultat semblable peut être trouvé dans [KG87] dans un contexte de commande en temps minimal. La différence réside dans la façon d'implémenter l'opération de projection. L'article cité utilise la méthode d'élimination de Fourier-Motzkin.

[33] En utilisant les ensembles $IC_t$, une 'stratégie de commande en temps minimal' peut être développée [KG87]. L'idée sera de trouver le $t$ minimal pour lequel $x^0 \in IC_t$ et ensuite d'appliquer les commandes de sorte que $x_k \in IC_{t-k}, k = 1,...,t$.

$$x^0 \in \Pr_x\{T_{t+1} \cap Y\} \Rightarrow \exists u_0 \; t.q. \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \in T_{t+1} \cap Y$$

D'où :

$$\Rightarrow x_1 = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \in IC_t \Rightarrow \exists \{u_1, u_2, ..., u_t\} \; t.q. \; x_{t+1} = 0$$

qui implique qu'il existe au moins une séquence de commande de dimension $t+1$ qui amène le système à l'origine et donc $IC_{t+1} \subset \Pr_x\{T_{t+1} \cap Y\}$. ■

**Exemple A3.23 :** Reprenons toujours l'exemple d'un double intégrateur :

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u_k,$$

$$\begin{bmatrix} x_k \\ u_k \end{bmatrix} \in conv\left\{ \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \\ -1 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 3 \\ 3 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -3 \\ 1 \end{bmatrix}, \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -3 \\ -3 \\ 1 \end{bmatrix} \right\} \qquad \text{(A3.23)}$$

La Figure A3.5 illustre la succession des ensembles contrôlables, l'ensemble maximal étant atteint après 9 itérations. Pour comparaison, le domaine accessible dans un espace étendu est également représenté.

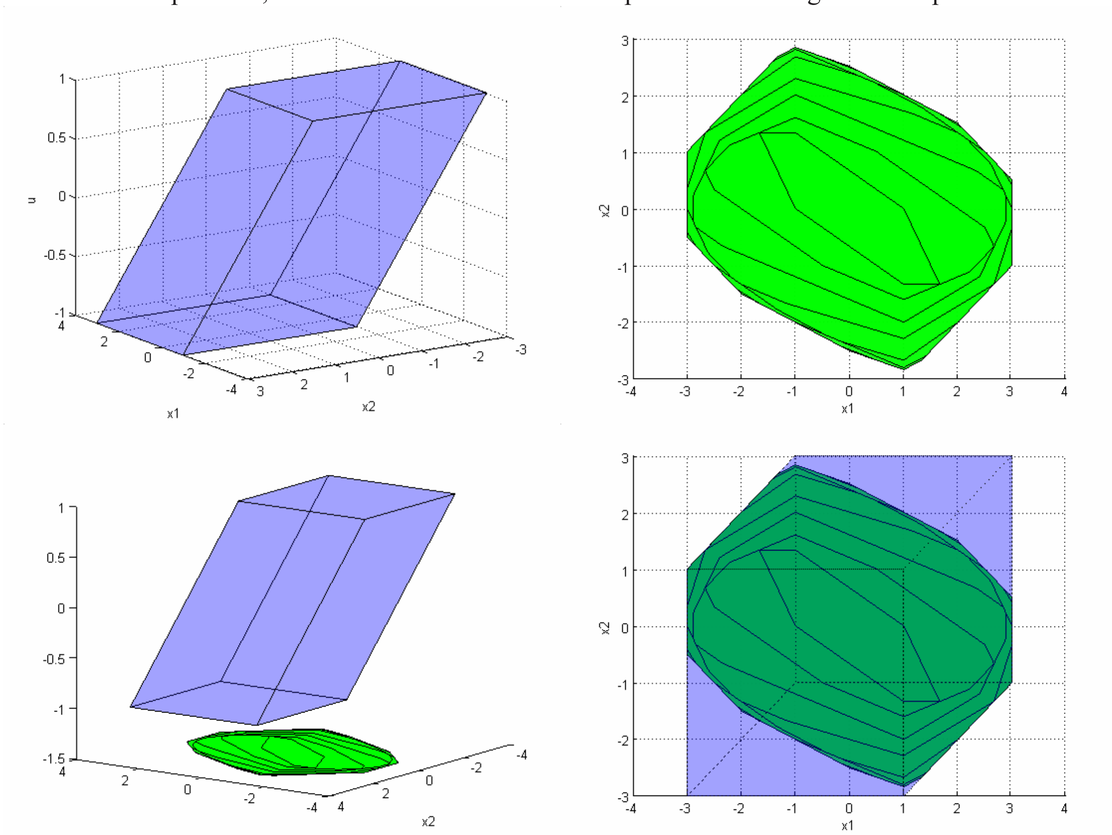



**Figure A3.5 : a) Ensemble contraint $Y$ b) Ensemble contrôlable**
**c) Superposition dans l'espace étendu d) superposition dans l'espace d'état**

**Remarque A3.24 :** Dans tous les développements antérieurs, l'ensemble objectif $IC_0$ a été considéré comme étant l'origine. En fait cet ensemble peut être n'importe quel autre ensemble invariant $IC \subset \Re^n$. Cet aspect est intéressant pour les systèmes stabilisables mais qui contiennent des parties non contrôlables (qui ne peuvent bien sûr pas être amenées à l'origine an temps fini).

**Remarque A3.25 :** Si l'on renonce à l'hypothèse que l'ensemble objectif est invariant, on peut choisir n'importe quel autre $C_0 \subset \Re^n$. La procédure récursive de construction reste valable, mais les propriétés de la série $\{C_0, C_1, C_2, ....., C_t, ....\}$ décrites dans la proposition A3.15 sont perdues. En premier lieu, la propriété d'invariance positive de $C_t$ sera perdue dans le cas général. Malgré tout, s'il existe un ensemble $C_t$ positif invariant, toute la série $C_{t+k}, k>0$ hérite de cette propriété.

## *12.4 Ensembles maximaux admissibles*

La description de zones admissibles dans le cas de contraintes mixtes est en général complexe (Remarque A3.21), mais présente un intérêt certain pour la synthèse de loi MPC. Considérons donc le système :

$$\mathrm{x}_{\mathrm{k}+1} = Ax_k + Bu_k$$

et les contraintes définies par des combinaisons linéaires des entrées et des états, pouvant être considérées comme des sorties de système :

$$\mathrm{z}_{\mathrm{k}} = Cx_k + Du_k \in Z$$

où $Z$ est un domaine polyédral contenant l'origine. Le but est toujours de caractériser tous les états initiaux $x^0 \in \Re^n$ pour lesquels la trajectoire du système peut satisfaire les contraintes. Pour une loi de commande fixée $u_k = Kx_k$ et en considérant les notations : $A+BK \to A, C+DK \to C$ le problème initial devient :

$$x_{k+1} = Ax_k, x_0 = x^0$$
$$z_k = Cx_k \in Z$$

**Définition A3.26 :** *Le domaine maximal admissible [GT91]* est donné par l'ensemble des conditions initiales pour lequel la dynamique du système, $x_k$, satisfait les contraintes sur la sortie :

$$O_\infty = \left\{x^0 \in \Re^n \middle| CA^k x^0 \in Z, \forall k \in \Im_+\right\} \tag{A3.24}$$

**Définition A3.27 :** Un *domaine maximal admissible sur N pas* peut être défini par :

$$O_N = \left\{x^0 \in \Re^n \middle| CA^k x^0 \in Z, \forall 0 \le k \le N\right\} \tag{A3.25}$$

**Proposition A3.28 :** Pour $O_\infty, O_N$, les propriétés suivantes sont vérifiées :

i) Si la paire $(C, A)$ est observable et $Z$ borné, alors $O_\infty$ est borné.

ii) $O_\infty \subset O_{k_2} \subset O_{k_1}, \forall k_1, k_2 \in \Im_+, k_1 \le k_2$

iii) $O_\infty$ est positif invariant.

iv) si $A$ est stable au sens de Lyapunov ( $|\lambda_i(A)| \le 1, i = 1..n,$ et $|\lambda_i(A)| = 1$ sont simples ( $\lambda_i$ valeurs propres de A)) et $0 \in \operatorname{int} Z$ alors $0 \in \operatorname{int} O_\infty$.

v) $O_\infty(A, C, Z) = UO_\infty(\widetilde{A}, \widetilde{C}, Z)$ si $\widetilde{A} = U^{-1}AU, \widetilde{C} = CU$ et par conséquent en choisissant la matrice non singulière $U \in \Re^{n\times n}$ adéquate :

- si $(A, C)$ est non-observable et $(A_1, C_1)$ définit le sous-espace observable de dimension $n-m$ alors :

$$O_\infty(A, C, Z) = O_\infty(A_1, C_1, Z) \times \Re^{n-m}$$

- si $(A, C)$ contient des modes divergents et si par changement de coordonnées on peut retrouver la paire $(A_1, C_1)$ stable au sens de Lyapunov avec $A \in \Re^m$, alors :

$$O_\infty(A, C, Z) = O_\infty(A_1, C_1, Z) \times \{0\}^{n-m}$$

vi) si $O_k = O_{k+1}$ alors $O_\infty = O_k$ [34] ■

[34] Ce résultat est le plus commun dans la littérature, car il revient au principe d'invariance positive de $O_\infty$

**Théorème A3.29 :** En supposant que :

i) $A$ est asymptotiquement stable,

ii) la paire $(C, A)$ est observable,

iii) $Z$ est borné,

iv) $0 \in \operatorname{int} Z$ ,

alors $O_\infty$ peuvent être déterminé par un nombre fini d'itérations.

**Preuve :** En construisant récursivement $O_1, O_2,...$ et en se basant sur la forme compacte des domaines sous contraintes et la continuité des fonctions qui définissent les contraintes, le nombre fini d'itérations peut être démontré [GT91][35,36] ■

**Exemple A3.30 : C**onsidérons le système :

$$\begin{cases} x_{k+1} = \begin{bmatrix} 0.9 & 2 \\ -1 & 1.9 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k \\ z_k = \begin{bmatrix} -2 & 1.5 \end{bmatrix} x_k + u_k \end{cases} \tag{A3.26}$$
$$z_k \in Z = [-0.5; 0.5]$$

Si l'on s'intéresse à l'espace maximal admissible pour ce système bouclé par le correcteur :

$$u_k = \begin{bmatrix} 1 & -1 \end{bmatrix} x_k$$

on se rend compte que les conditions du théorème A3.29 sont satisfaites, et en construisant récursivement les ensembles $O_i, i = 1,2,...$ (Figure A3.6a) on détecte la superposition de deux ensembles consécutifs au 12[ème] pas, d'où la détermination de $O_\infty$ (Figure A3.6b).

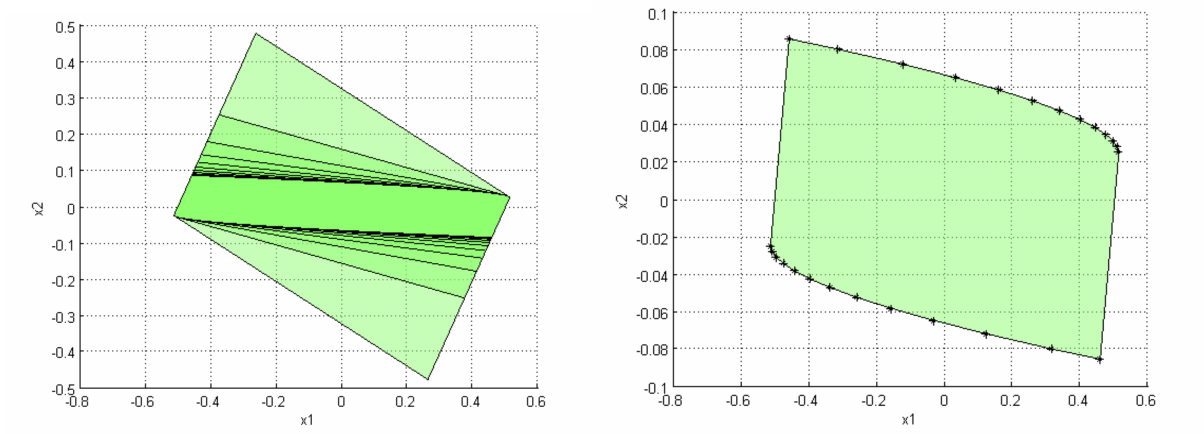



**Figure A3.6 : a)** $0_1, \ldots, 0_{12}$ **pour l'exemple A3.30 b)** $0_{12} = O_\infty$ **pour l'exemple A3.30**

Si dans le cas d'ensembles commandables on laisse la liberté à la commande d'avoir n'importe quelle structure dès l'instant qu'elle respecte les contraintes, en revanche, dans le cas de l'ensemble maximale admissible, on impose une structure de commande particulière, et on détermine le domaine correspondant aux exigences requises. Cet ensemble maximal admissible est intéressant du point de vue de la commande MPC car il représente le domaine à partir duquel la loi sans contraintes n'activera plus les contraintes et sert donc au choix des ensembles terminaux.

---

[35] Il est également démontré dans l'article mentionné que l'hypothèse de stabilité asymptotique de la matrice $A$ peut être remplacée par la stabilité au sens de Lyapunov de cette même matrice. Dans ces conditions (évidemment moins contraignantes) l'ensemble maximal admissible peut être déterminé par un nombre fini d'itérations.

[36] Le développement récursif des espaces admissibles est basé sur une procédure de calcul polyédral insistant sur la partie liée à l'élimination de la redondance. On trouve à ce propos dans la littérature des méthodes employant à chaque itération des solveurs de problèmes de programmation linéaire et utilisant le fait que la redondance est structurellement instable. On a utilisé dans le même but une représentation duale des ensembles pour laquelle la redondance est très naturellement traitée à l'aide des matrices de saturation. Une telle méthode est vivement souhaitable au moins pour les systèmes de faibles dimensions.

## *12.5 Ensemble faisable et commande optimale*

Le paragraphe précédent a défini les ensembles admissibles dans un contexte général, ainsi que des notions faisant intervenir le système bouclé par une loi de commande donnée. Pour le même type de modèle :

$$(\Sigma)\begin{cases} x_{k+1} = Ax_k + Bu_k \\ z_k = Cx_k + Du_k \in Z \end{cases} \tag{A3.27}$$

notons $X_{\max}$ l'ensemble des points initiaux pour lesquels il existe une séquence de commande (potentiellement infinie) qui amène l'état à l'origine tout en respectant les contraintes. Formellement :

$$X_{\max} = \left\{x_0 \middle| \exists u_k, \forall \mathrm{k} \geq 0 \text{ t.q. } z_k \in Z, \text{et } \lim_{\mathrm{k}\to\infty} (x_k^T x_k) = 0\right\}$$

Pour le système $(\Sigma)$, un critère d'optimalité peut être défini par :

$$\phi(x^0, u_0, u_1, u_2, \ldots) = \sum_{\mathrm{k}=0}^{\infty} x_k^T Q x_k + u_k^T R u_k < \infty \tag{A3.28}$$

L'ensemble des conditions initiales pour lesquelles la fonction de coût a une valeur finie sera noté $X_0$ :

$$X_0 = \left\{x_0 \in \Re^n \middle| \exists u_k, \forall k \geq 0 \text{ t.q. } z_k \in Z, \text{et } \sum_{\mathrm{k}=0}^{\infty} x_k^T Q x_k + u_k^T R u_k < \infty\right\} \tag{A3.29}$$

**Lemme A3.31 :** $X_0 = X_{\max}$

**Preuve :**

D'une part $X_{\max} \supseteq X_0$ parce que $\sum_{\mathrm{k}=0}^{\infty} \left\{x_k^T Q x_k + u_k^T R u_k\right\} < \infty \Rightarrow x_k^T Q x_k + u_k^T R u_k \underset{k\to\infty}{\to} 0 \Rightarrow x_k^T x_k \underset{k\to\infty}{\to} 0$.

D'autre part, $\forall x^0 \in X_{\max}$ il existe un trajectoire $x_k^T x_k \underset{k\to\infty}{\to} 0$ qui implique que la paire $(A, B)$ est stabilisable et donc il existe une loi $u_k = Kx_k$ et un ensemble maximal admissible $O_\infty$ associé. Mais $0 \in \operatorname{int} O_\infty$ (Proposition 5.34iv) et donc la trajectoire $\{x_k\}$ va rentrer en temps fini dans $O_\infty$. A ce stade, on détient une séquence faisable qui nous assure que $\sum_{\mathrm{k}=0}^{\infty} \left\{x_k^T Q x_k + u_k^T R u_k\right\} < \infty$. ■

On dispose désormais de tous les éléments permettant d'introduire les notions directement liées à la commande optimale dans les cas sans contraintes ($\mathrm{P}^{\mathrm{U}}$), avec contraintes sur un horizon fini ($\mathrm{P}^{\mathrm{N}}$) ou infini (P).

**(P)** $$\Phi(x^0) = \inf_{k_u} \sum_{k=0}^{\infty} \left\{x_k^T Q x_k + u_k^T R u_k\right\}$$

$$\text{sous les contraintes } \begin{aligned} & x_{k+1} = Ax_k + Bu_k, k \geq 0, x_0 = x^0, \\ & z_k = Cx_k + Du_k, z_k \in Z \end{aligned} \tag{A3.30}$$

(H1) $Q > 0, R > 0$

(H2) $Z$ est un ensemble polyédral

(H3) $0 \in \operatorname{int} Z$

(H4) $x_0 \in X_0 = \left\{x_0 \in \Re^n \middle| \exists u_k \text{t.q. } z_k \in Z, \text{et } \sum_{\mathrm{k}=0}^{\infty} x_k^T Q x + u_k^T R u_k < \infty\right\}$

**Notation A3.32 :** La solution de (P) est $k_u(x^0)=\{u_0,u_1,u_2,...\}$ induisant une trajectoire $\chi(x^0)=\{x_1,x_2,x_3,...\}$. Dans la suite, la dépendance en $x^0$ ne sera plus mentionnée explicitement et donc $k_u(x^0)=k_u$ et $\chi(x^0)=\chi$. Le *ième* élément de chaque énumération sera $k_u(i)$, respectivement $\chi(i)$ ■

**(P$^U$)** $$\Phi^U(x^0)=\inf_{k_u}\sum_{k=0}^{\infty}\left\{x_k^T Q x_k + u_k^T R u_k\right\}$$

$$\text{avec } x_{k+1}=Ax_k+Bu_k,\, k\geq 0, x_0=x^0 \tag{A3.31}$$

(H1)+ (H4)

**Notation A3.33 :** La solution de (P$^U$) sera notée $k_u^U(x^0)$ et la trajectoire $\chi^U(x^0)$.

**Remarque A3.34 :** (P$^U$) admet la solution unique $u_k=-Kx_k$ où :

$$K_{LQ}=-(R+B^T PB)^{-1}B^T PA$$

est construit à partir de l'unique solution positive définie $P$ de l'équation algébrique de Riccati :

$$P=(A+BK_{LQ})^T P(A+BK_{LQ})+K_{LQ}^T RK_{LQ}+Q.$$

**(P$^N$)** $$\Phi^N(x^0)=\inf_{k_u} x_N^T P x_N+\sum_{k=0}^{N-1}\left\{x_k^T Q x_k + u_k^T R u_k\right\}$$

$$\begin{aligned}\text{sous les contraintes } & x_{k+1}=Ax_k+Bu_k,\, k\geq 0, x_0=x^0, \\ & z_k=Cx_k+Du_k,\, z_k\in Z,\quad 0\leq k\leq N-1\end{aligned} \tag{A3.32}$$

(H1)+ (H4)

**Notation A3.35 :** La solution de (P$^N$) sera notée $k_u^N(x^0)$ et la trajectoire correspondante $\chi^N(x^0)$.

La série des $\Phi^N$ construites pour des horizons de contrainte croissants est non décroissante et bornée de sorte que la limite suivante existe :

$$\Phi^\infty=\lim_{N\to\infty}\Phi^N \tag{A3.33}$$

*Le domaine maximal admissible* $O_\infty$ pour le système $(\Sigma)$ bouclé par la loi de commande $u_k=K_{LQ}x_k$ regroupe l'ensemble des conditions initiales pour lequel la solution du problème d'optimisation à horizon infini sans contraintes est équivalente à celui avec contraintes. Ce résultat et ces implications sont résumés par le lemme suivant.

**Lemme A3.36 :** i) $x^0\in O_\infty \Rightarrow \Phi^\infty(x^0)=\Phi^U(x^0)=\Phi^N(x^0), \forall N\geq 0$

ii) [CM96] Si $\exists N<\infty$ t.q. $\chi^N(N)\in O_\infty$ alors $\Phi^N=\Phi$

**Preuve :**
i) Soit la définition :

$$O_\infty=\left\{x\in\Re^n \,\middle|\, (A+BK_{LQ})^k x\in \overline{Z}; k\geq 0\right\}$$

où $\overline{Z}=\left\{x\in\Re^n \,\middle|\, (C+DK_{LQ})x\in Z\right\}$ et $K_{LQ}$ est le gain optimal (construit à l'aide de la solution de l'équation de Riccati) pour le problème LQ à horizon infini comme précédemment. $O_\infty$ est positif invariant et à partir de sa définition on constate que les contraintes sont vérifiées pour $\forall k \geq 0$ et donc l'optimum pour le cas avec contraintes et sans contraintes coïncide $\Phi^{\infty}(x^0) = \Phi^U(x^0)$. Le résultat est obtenu en se basant sur le fait que $\Phi^{\infty}(x^0) \geq \Phi^N(x^0) \geq \Phi^U(x^0), \forall N \geq 0$.

ii) Si $\chi^N(N) \in O_\infty$ alors $k_u^N$ est une solution faisable pour (P) et donc $\Phi^N \geq \Phi$. Mais puisque $\Phi^N \leq \Phi^\infty \leq \Phi$ alors $\Phi^N = \Phi$. ■

Notons que $O_\infty$ est positif invariant et dépend uniquement de l'ensemble $Z$, du modèle $(A, B)$ et du gain optimal $K_{LQ}$. $O_\infty$ est donc un domaine facile à décrire hors-ligne. Dès que la commande permet d'atteindre ce domaine, les risques d'infaisabilité sont écartés.

Soient les scalaires suivants caractérisant $O_\infty$ :

$$0 \leq p \leq \inf_{x \notin O_\infty} \{x^T P x\} \tag{A3.34}$$

$$0 \leq q \leq \inf_{x \notin O_\infty} \{x^T Q x\} \tag{A3.35}$$

$$U \geq \sup_{x \in Z} \Phi \tag{A3.36}$$

[37]

**Théorème A3.37 [CM96] :** $\forall x^0 \in X_0, \exists N < \infty$ t.q. $\chi^N(N) \in O_\infty$

**Preuve :**

$$U \geq \Phi^N = \chi^N(N)^T P \chi^N(N) + \sum_{k=0}^{N-1} \left\{ \chi^N(k)^T Q \chi^N(k) + k_u^N(k)^T Q k_u^N(k) \right\} \geq$$

$$\geq \chi^N(N)^T P \chi^N(N) + \sum_{k=0}^{N-1} \chi^N(k)^T Q \chi^N(k) \geq$$

$$\geq p + \sum_{k=0}^{N-1} \{q\} = Nq + p$$

En raisonnant par l'absurde, supposons que $\chi^N(N) \notin O_\infty$ soit $\Phi^N \geq Nq + p \underset{N \to \infty}{\to} \infty$ ce qui contredit l'hypothèse que $x^0 \in X_0$. Donc pour $\forall N > (U - p)/q$ on a $\chi^N(N) \in O_\infty$ ■

**Remarque A3.38 :** La limite supérieure de la fonction de coût $U$ est liée au domaine associé. Il est évident que ce domaine diminue avec l'évolution du système et en conséquence $U$ peut aussi diminuer.

[37] Toutes ces valeurs peuvent être trouvées en évaluant les quantités respectives des sommets des domaines polyédraux correspondants.

## 13. Annexe 4 : Résultats complémentaires liés à l'application sur banc

La structure du prédicteur optimal pour le modèle de la machine asynchrone utilisé pour la problématique de positionnement est la suivante :

$$\hat{y}_{t+j} = \underbrace{F_j(q^{-1})y_t + H_j(q^{-1})\Delta u_{t-1}}_{\text{l=réponse libre}} + \underbrace{G_j(q^{-1})\Delta u_{t+j-1}}_{\text{réponse forcée}} \tag{A4.1}$$

avec, pour $N_u = 3; N_1 = 1; N_2 = 16$ (notations introduites au chapitre 2) :

$$\mathbf{if} = \begin{bmatrix} 2.9980 & -2.9960 & 0.9980 \\ 5.9920 & -7.9840 & 2.9920 \\ 9.9800 & -14.9600 & 5.9800 \\ 14.9601 & -23.9201 & 9.9601 \\ 20.9301 & -34.8603 & 14.9301 \\ 27.8883 & -47.7766 & 20.8883 \\ 35.8325 & -62.6650 & 27.8325 \\ 44.7608 & -79.5217 & 35.7608 \\ 54.6713 & -98.3426 & 44.6713 \\ 65.5620 & -119.1239 & 54.5620 \\ 77.4308 & -141.8617 & 65.4308 \\ 90.2760 & -166.5520 & 77.2760 \\ 104.0954 & -193.1909 & 90.0954 \\ 118.8872 & -221.7745 & 103.8872 \\ 134.6495 & -252.2989 & 118.6495 \\ 151.3802 & -284.7603 & 134.3802 \end{bmatrix}; \mathbf{ih} = \begin{bmatrix} 0.0001 \\ 0.0002 \\ 0.0005 \\ 0.0008 \\ 0.0012 \\ 0.0017 \\ 0.0023 \\ 0.0029 \\ 0.0037 \\ 0.0045 \\ 0.0054 \\ 0.0064 \\ 0.0074 \\ 0.0085 \\ 0.0098 \\ 0.0110 \end{bmatrix} \qquad \text{(A4.2°}$$

$$\mathbf{G} = \begin{bmatrix} 0.0001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0003 & 0.0001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0007 & 0.0003 & 0.0001 & 0 & 0 & 0 & 0 & 0 \\ 0.0013 & 0.0007 & 0.0003 & 0.0001 & 0 & 0 & 0 & 0 \\ 0.0020 & 0.0013 & 0.0007 & 0.0003 & 0.0001 & 0 & 0 & 0 \\ 0.0029 & 0.0020 & 0.0013 & 0.0007 & 0.0003 & 0.0001 & 0 & 0 \\ 0.0040 & 0.0029 & 0.0020 & 0.0013 & 0.0007 & 0.0003 & 0.0001 & 0 \\ 0.0052 & 0.0040 & 0.0029 & 0.0020 & 0.0013 & 0.0007 & 0.0003 & 0.0001 \\ 0.0066 & 0.0052 & 0.0040 & 0.0029 & 0.0020 & 0.0013 & 0.0007 & 0.0003 \\ 0.0082 & 0.0066 & 0.0052 & 0.0040 & 0.0029 & 0.0020 & 0.0013 & 0.0007 \\ 0.0099 & 0.0082 & 0.0066 & 0.0052 & 0.0040 & 0.0029 & 0.0020 & 0.0013 \\ 0.0117 & 0.0099 & 0.0082 & 0.0066 & 0.0052 & 0.0040 & 0.0029 & 0.0020 \\ 0.0138 & 0.0117 & 0.0099 & 0.0082 & 0.0066 & 0.0052 & 0.0040 & 0.0029 \\ 0.0160 & 0.0138 & 0.0117 & 0.0099 & 0.0082 & 0.0066 & 0.0052 & 0.0040 \\ 0.0183 & 0.0160 & 0.0138 & 0.0117 & 0.0099 & 0.0082 & 0.0066 & 0.0052 \\ 0.0208 & 0.0183 & 0.0160 & 0.0138 & 0.0117 & 0.0099 & 0.0082 & 0.0066 \end{bmatrix} \tag{A4.3}$$

Les courbes suivantes sont ajoutées en complément du paragraphe 7.5 pour le cas de contraintes sur le dépassement et un horizon de prédiction $N_u = 10$.

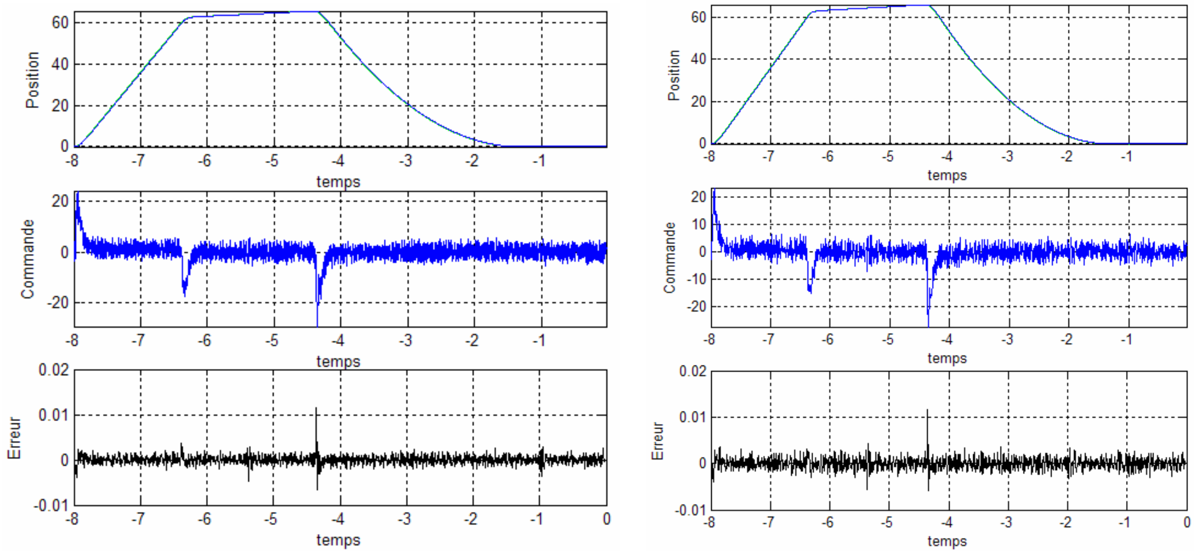



**Figure A4.1 : Réponses temporelles pour les correcteurs GPC obtenus à partir du modèle nominal et critère quadratique a) Loi GPC initiale - la présence des perturbations se fait sentir sur le signal de commande et sur l'erreur de poursuite. b) version de loi GPC robustifiée.**

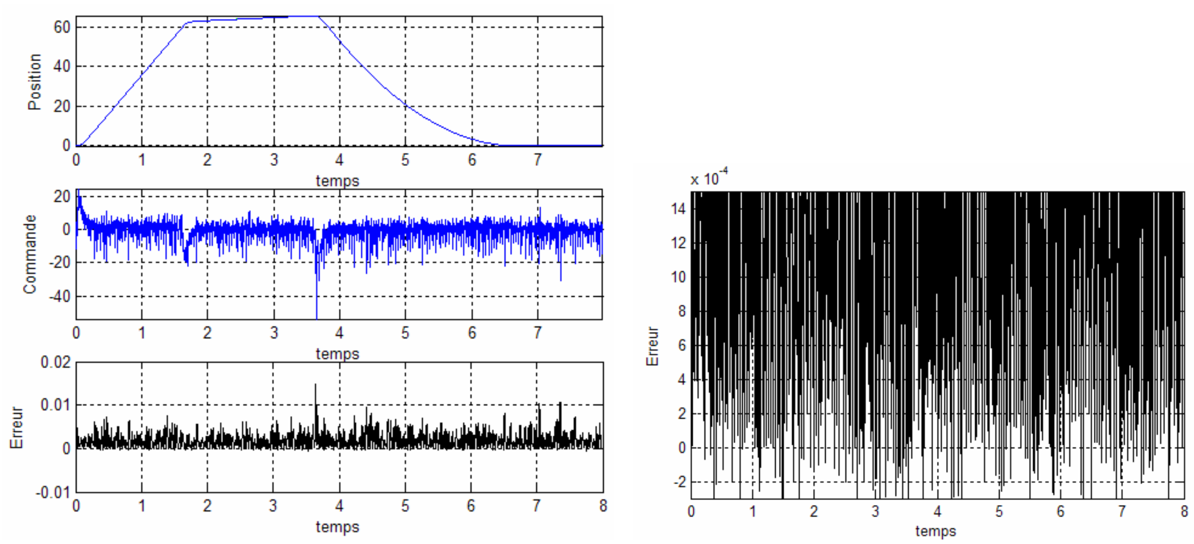



**Figure A4.2 : a) Réponses temporelles pour la loi GPC obtenues à partir du modèle nominal avec prise en compte de contraintes b) détail de l'erreur de poursuite indiquant que les perturbations compromettent la satisfaction de contraintes.**

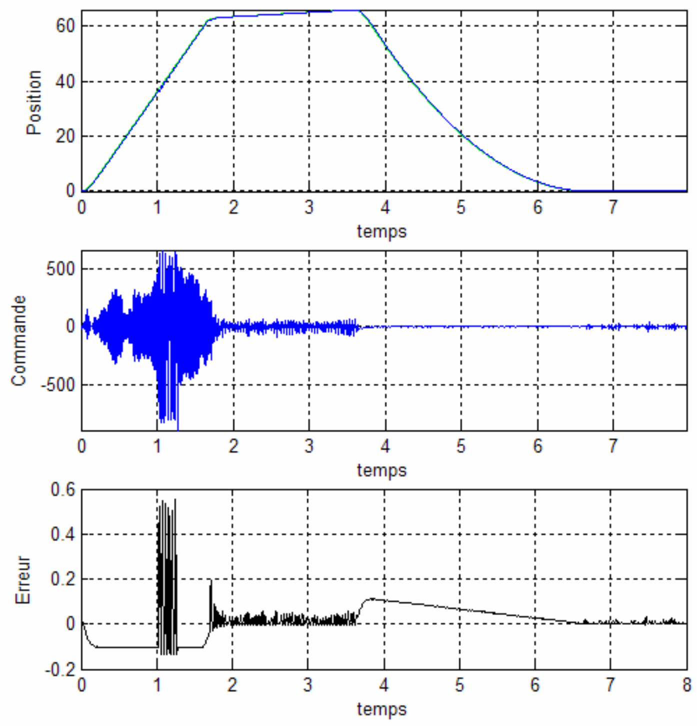



**Figure A4.3 : Simulation du comportement du système bouclé avec la loi GPC, pour laquelle les régulateurs RST par morceaux ont été modifiés via le paramètre de Youla correspondant aux loi sans contraintes**

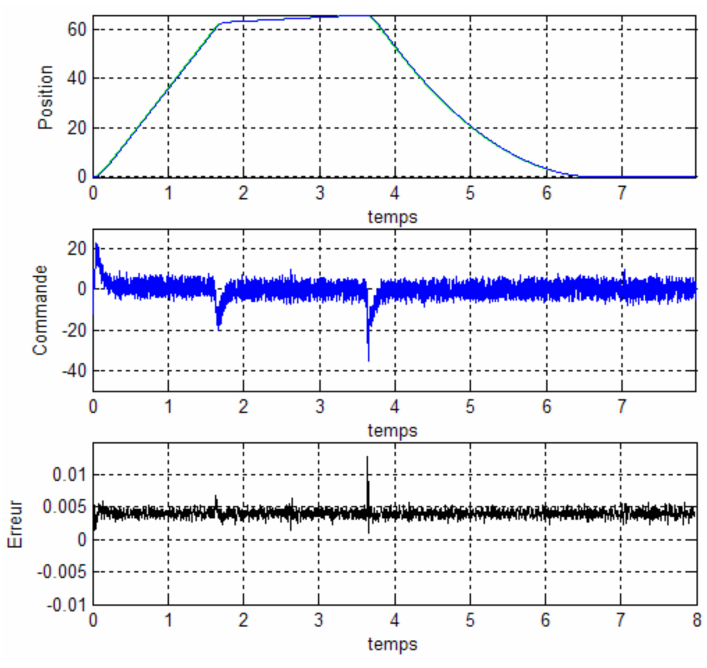



**Figure A4.4 : Réponses temporelles pour la loi GPC basée sur un critère 'min-max', horizon de prédiction de 4**