

N° D'ORDRE : 9686



**THÈSE DE
DOCTORAT**

SPECIALITE : PHYSIQUE

*Ecole Doctorale « Sciences et Technologies de l'Information des
Télécommunications et des Systèmes »*

Présentée par :

Hichem BENLAOUKLI

Sujet :

**Méthodes géométriques pour la construction des ensembles invariants
Application à la faisabilité des lois de commande prédictive.**

Soutenue le 10 décembre 2009 devant les membres du jury :

Mme. P. BENDOTTI (Ingénieur, Département OSIRIS R32, EDF R&D), invitée
M. Y. CHITOUR (Professeur, Université Paris XI, Président du Jury)
M. D. LEFEBVRE (Professeur, Université le Havre) Rapporteur
M. S. MAMMAR (Professeur, Université d'Evry) Rapporteur
M. G. DUC (Professeur, Supélec, Directeur de thèse)
M. S. OLARU (Enseignant-chercheur, Supélec) Encadrant

Table des matières

Liste des notations et acronymes	4
Liste des publications	6
1 Introduction	9
1.1 Repère historique sur la commande prédictive et la commande basée sur optimisation temps réel	9
1.2 Géométrie de la commande prédictive	10
1.3 Analyse des système PWA - Invariance positive	11
1.4 Contribution	12
1.5 Organisation	12
2 Commande Prédictive	13
2.1 Introduction	13
2.2 Les bases de la commande prédictive	14
2.2.1 Modèle de prédiction à temps discret	14
2.2.2 Modèle de prédiction avec intégrateur	15
2.2.3 Séquence de commande optimale	16
2.2.4 Commande à horizon glissant	17
2.2.5 Commande MPC avec contraintes	19
2.3 Commande prédictive et optimisation paramétrique	23
2.3.1 Optimisation paramétrique	23
2.3.2 Méthodes de construction de la solution explicite	25
2.3.3 Analyse des solutions explicites	25
2.3.4 Commande prédictive explicite	26
2.3.5 Implémentation de la commande explicite	27
2.4 Invariance	27
2.4.1 Ensembles invariants	27
2.4.2 Ensemble invariant maximal par rapport à la commande	31
2.5 Stabilité de la commande prédictive	32
2.5.1 Faisabilité	32
2.5.2 Loi MPC avec garantie de stabilité	33
2.6 Conclusion	34

3	Systèmes affines par morceaux	35
3.1	Introduction	35
3.2	Systèmes hybrides	35
3.2.1	Système PWA (Piecewise Affine Systems)	36
3.2.2	Systèmes MLD (<i>Mixed Logical Dynamical systems</i>)	36
3.2.3	Systèmes à complémentarité linéaire LC	36
3.2.4	Systèmes à complémentarité linéaire étendu ELC	37
3.2.5	Systèmes MMPS (<i>min-max plus scaling</i>)	37
3.2.6	Équivalence entre les différents modèles	37
3.3	Exemples illustratifs	38
3.3.1	Système affine par morceaux : véhicule avec plusieurs dynamiques	38
3.3.2	Système hybride : convertisseur DC-DC	39
3.3.3	Système hybride : balle bondissante	40
3.3.4	Systèmes LTI en boucle fermée avec une loi prédictive explicite	42
3.4	Analyse des systèmes PWA	43
3.5	Stabilité des systèmes LTI et PWA	45
3.5.1	Notions de stabilité pour systèmes dynamiques	46
3.5.2	Notions de stabilités pour systèmes LTI et avec incertitudes	46
3.5.3	Notions de stabilités pour les systèmes PWA	49
3.6	Conception de la commande pour système PWA	52
3.6.1	À partir d'une fonction de Lyapunov quadratique	52
3.6.2	À partir d'une fonction de Lyapunov quadratique par morceaux	53
3.7	Conclusion	53
4	Calcul des ensembles invariants	55
4.1	Introduction	55
4.2	Calcul et approximation des ensembles invariants	55
4.2.1	Calcul de la borne supérieure du MPI	56
4.2.2	Calcul de la borne inférieure du MPI	59
4.2.3	Phénomène de cycle limite	64
4.2.4	Convergence des constructions et le rapport avec l'ensemble MPI	68
4.3	Amélioration du temps de calcul	70
4.3.1	Analyse par intervalles	70
4.4	Analyse par graphe de transitions	79
4.4.1	Propriétés des graphes de transitions	81
4.5	Calcul basé sur les graphes de transitions	82
4.5.1	Calcul en nombre d'itérations fini	85
4.6	Conclusion	89
5	Extension de la commande prédictive	91
5.1	Introduction	91
5.2	MPC robuste et les ensembles invariants	91
5.2.1	Commande prédictive robusteifiée à base de LMI	92
5.2.2	Réglage du MPC pour garantir la robustesse autour de l'équilibre	94

5.2.3	Ensemble invariant robuste maximal	95
5.3	Comparaison des ensembles invariants	98
5.3.1	Commande prédictive basée sur les LMI	98
5.3.2	MPC désensibilisée	99
5.3.3	Illustration des constructions invariantes robustes	100
5.3.4	Application du MPC robuste pour système LTI affecté par un retard variable à l'entrée	103
5.3.5	Formulation du problème	105
5.4	Problèmes de faisabilité dans le suivi de trajectoire	111
5.5	Conclusion	118
6	Commande et suivi de trajectoire	121
6.1	Introduction	121
6.2	Description de la vallée hydraulique	121
6.3	Problématique 1 : Pente de rendement constant	122
6.4	Modélisation	122
6.5	Commande prédictive	122
6.5.1	Formulation de la commande	122
6.5.2	Résolution et simulation	125
6.5.3	Synthèse MPC	126
6.5.4	Utilisation du domaine viable (invariance)	127
6.6	Problématique 2 : Pente de rendement différencié pompage/turbinage	130
6.7	Problématique 3 : tube de viabilité	130
6.8	Conclusion	132
7	Conclusion	139
7.1	Contribution	139
7.2	Directions et perspectives	140
A	Calcul géométrique sur les polytopes	141
A.1	Opérations sur les ensembles polyédraux	141
A.2	Enveloppe convexe d'un ensemble de polytopes	143
A.3	Image et image inverse d'un polytope	143
A.4	Distance de Hausdorff	145
B	Comparaison de temps de calcul	147

Liste des notations et acronymes

Notations

\mathbb{Z}	Nombres entiers
\mathbb{N}	Nombres entiers positifs
\mathbb{N}^*	Nombres entiers positifs non nuls
\mathbb{R}	Nombres réels
$\lambda(A)$	Valeurs propres de la matrice A
I	Matrice identité

Acronymes

ELC	Extended Linear Complementarity
GPC	Generalized Predictive Control
KKT	Karush-Kuhn-Tucker
LC	Linear Complementarity
LDI	Linear Differential Inclusion system
LMI	Linear Matrix Inequality
LTI	Linear Time Invariant
MPI	Maximal Positively Invariant
MMPS	Max-Min-Plus-Scaling
MLD	Mixed Logical Dynamical
MPC	Model Predictive Control
PWA	Piecewise Affine

Liste des publications

Conférences internationales avec actes

- H. Benlaoukli, S. Oлару, "On the computation and bounding of invariant sets", *ICCAS 2007*, Séoul, Corée de sud, 17-20 octobre 2007.
- H. Benlaoukli, S. Oлару, "Computation and bounding of robust invariant sets for uncertain systems", *IFAC World Congress 2008*, Séoul, Corée de sud.
- H. Benlaoukli, S. Oлару, S.-I. Niculescu, P. Boucher, "Commande prédictive des systèmes avec l'entrée affectée par retard variable", *CIFA, 2008*, Roumanie.
- H. Benlaoukli, S. Oлару, "Model Predictive Control - Numerical Methods for the Invariant Sets Approximation", *NAA, 2008*, Bulgarie.
- H. Benlaoukli, M. Hovd, S. Oлару, P. Boucher, "Invariant sets for PWA systems using transition graph", *ICCA, 2009*, Nouvelle Zélande.
- H. Benlaoukli, S. Oлару, "Feasible Tracking Reference set computation using the principle of invariant set construction", *ICCA, 2009*, Nouvelle Zélande.
- S. Oлару, H. Benlaoukli, S. Niculescu, "MPC for systems with variable time-delay, Robust positive invariant set approximations", *ICINCO, 2008*, Portugal.

Chapitres de livres

- H. Benlaoukli, S. Oлару, "Model predictive control - numerical methods for the invariant sets approximation", *LNCS 5434 Numerical Analysis and Its Applications : 4th International Conference*, Berlin, Heidelberg : Springer-Verlag, pp. 192-199, 2009.
- H. Benlaoukli, S. Oлару, S.-I. Niculescu, S. Marinkovic, "Commande prédictive des systèmes avec l'entrée affectée par retard variable", *Automatique Avancée Et Informatique Appliquée 2009*, Édition de l'Académie Roumaine, pp. 55-61, 2009.

Articles de Journal

- H. Benlaoukli, M. Hovd, S. Oлару, "Geometrical characterization of robust predictive control strategies", *CEAI*, Vol. 10, No. 4, pp. 13-19, 2008.

Chapitre 1

Introduction

1.1 Repère historique sur la commande prédictive et la commande basée sur optimisation temps réel

Il existe dans la littérature récente des monographies qui jalonnent les développements de la commande basée sur l'optimisation en temps réel et qui conduit généralement (par l'utilisation d'un *modèle de prédiction* et l'application du *principe de l'horizon glissant* pour le choix de la trajectoire) vers ce qu'on appelle dans le langage courant de l'automaticien **la commande prédictive**.

Un des premiers ouvrages sur le sujet est celui de M. Gevers et coauteurs [1] qui pose les premières questions sur la base théorique de cette méthodologie (émergente à cette époque là - 1990) mais qui, paradoxalement, montre un succès certain dans le monde industriel¹. Il est intéressant à remarquer dans cette première référence le lien établi avec la commande adaptative qui place les racines de la commande prédictive dans une perspective historique. Une monographie qui apparaît dans le moment de plein essor de la théorie MPC est celle de E. Camacho et C. Bordons [3] qui traite en détail la mise en équation du problème d'optimisation sur un horizon fini à base d'un modèle de prédiction en représentation par fonction de transfert ou en représentation d'état. Agrémenté par des exemples ponctuels qui soutiennent l'adaptation de cette méthodologie à différentes classes de systèmes (instables, phase non-minimale, affectées par un retard), le volume devient très populaire et il est réédité (sans doute à cause de son succès et de développements ultérieurs qui n'étaient pas traités dans la première édition) au début des années 2000, [4]. L'école anglaise qui a eu une contribution majeure dans le développement de la commande prédictive (voir les travaux des pionniers de la formulation GPC -Generalized predictive control- [5]) a proposé à son tour par J. A. Rossiter [6] et J. M. Maciejowski [7] deux livres de références. Si [6] reste à un niveau descriptif et est fort indiqué pour ceux qui veulent faire la connaissance du domaine, on retrouve dans [7] une analyse poussée de l'état de l'art au niveau théorique et méthodologique avec un effort apprécié en ce qui concerne la prise en compte des contraintes et les limitations associées à ce type de synthèse de lois de commande.

Les travaux de recherche provenant des centres de recherche d'Australie sont marquants

1. Qin et Badgwell [2] ont recensé les versions industrielles de la commande prédictive en détaillant tous les domaines d'application de cette technologie.

dans la généalogie MPC (le titre déjà mentionné [1] étant un exemple dans ce sens). A part les travaux des pionniers, il faut remarquer plus récemment une référence significative pour l'analyse des problèmes d'optimisation et des conditions d'optimalité dans le cadre prédictif [8] avec une mention particulière pour la présence des chapitres dédiés au traitement du problème dual d'estimation à horizon glissant.

On laisse pour la fin de ce tour d'horizon non exhaustif l'école française avec deux références pour la commande prédictive linéaire [9] (et sa réédition en version anglaise [10]) et [11]. Ces deux monographies qui se distinguent par l'aspect méthodologique de la présentation vont pousser l'analyse jusqu'aux derniers détails de l'implémentation et permettent le passage à l'implémentation algorithmique. On note que dans une perspective historique la formulation PFC -Predictive Functional Control- représente un repère notable par l'identification des éléments de base pour la commande à base de prédiction [12] à la fin des années '70. De date plus récente, le volume collectif [13] dresse l'état de l'art, avec les dernières avancées de la commande prédictive et indique les pistes de recherche avec les thèmes de recherches actives.

Tous ces repères bibliographiques sont pertinents pour la commande prédictive linéaire (basée sur modèles de prédiction linéaires et contraintes linéaires), cadre dans lequel sont placés aussi les travaux de thèse décrits par la suite. La commande prédictive non linéaire représente un sujet de recherche actif, qui a ses propres références (comme [14]) mais représente un cadre plus large pour les problèmes traités dans le présent manuscrit.

Les problèmes ouverts ainsi que les dernières avancées de la commande prédictive sont recueillis dans une série éditée par *Springer* en relation avec le cycle de conférences *Assessment and Future Directions of Nonlinear Model Predictive Control* [15, 16, 17].

1.2 Géométrie de la commande prédictive

Les avancées au niveau de la mise en forme des problèmes de commandes basées sur la prédiction mettent en évidence à la fin des années 90 et au début des années 2000 le fait qu'une révolution conceptuelle peut avoir lieu seulement en exploitant la structure des problèmes d'optimisation et notamment la géométrie du domaine faisable. Dans cette période les approches basées sur les LMI ont suscité de l'intérêt puisqu'elles promettaient d'apporter des solutions intéressantes par la manipulation de contraintes de façon systématique ainsi que pour la prise en compte des incertitudes.

Un vrai renouvellement au moins du point de vue de la perception de la commande prédictive est venue par les formulations explicites des lois de commandes prédictives en évitant donc la résolution d'un problème d'optimisation à chaque pas d'échantillonnage. Plusieurs approches ont émergé simultanément et on mentionne ici :

- la programmation multiparamétrique [18] [19],
- la dualité des représentations géométriques des contraintes [20], [8]
- l'utilisation des polyèdres paramétrés [21], [22], [13]
- la programmation dynamique [23]

Le travail de cette thèse se place dans la continuité de ces travaux récents sur les formulations explicites par l'intérêt qu'on porte sur l'analyse structurelle des résultats avec des implications sur la complexité et par l'attention donnée à la reformulation des problèmes de commande pour

tirer profit de la formulation analytique des solutions optimales associées.

On remarque que la construction de ces solutions explicites est basée dans tous les cas de figure sur des opérations géométriques qui intrinsèquement présentent une complexité exponentielle en fonction des paramètres d'entrées (nombre des contraintes, nombre des points extrêmes du domaine faisable, dimension du vecteur des paramètres). Le fait que tous ces traitements géométriques se font hors-ligne peut nous conduire vers la conclusion que le temps de traitement est virtuellement infini. Cette conclusion est bien sûr amendable du point de vue pratique et au delà de ça il faut souligner que la complexité de la solution explicite sera aussi directement affectée par la dimension des paramètres d'entrées pour le problème d'optimisation multiparamétrique associé à un problème de commande prédictive. Les implications de cette complexité de la loi de commande impactent les performances de la commande en temps réel qui devra évaluer une fonction de commande par retour d'état dans la limite d'un pas d'échantillonnage.

Concrètement, en partant de ces observations, l'objectif de ce travail de recherche est d'ignorer dans la phase de conception les contraintes qui garantissent *a priori* la stabilité de la boucle. Une telle construction vise bien sûr un allègement de la formulation, le prix à payer sera la perte des garanties de stabilité, laquelle devra être étudiée *a posteriori* en utilisant les concepts d'invariance positive.

1.3 Analyse des système PWA - Invariance positive

Structurellement, une solution explicite d'un problème de commande prédictive dans le cadre linéaire prend la forme d'un retour d'état affine par morceaux (polyédral) dans l'espace d'état et donc globalement le système en boucle fermée est décrit par un système dynamique affine par morceaux. Il faut souligner les avantages structurels hérités de la synthèse MPC tels que la distribution disjointe des régions par rapport à la loi de commande affine associée et éventuellement leur continuité (si le critère optimisé est quadratique).

L'opération d'analyse d'une solution explicite MPC sera réduite donc à l'analyse de la stabilité des systèmes PWA. Notons au passage qu'il s'agit des systèmes PWA en temps discret pour lesquels la stabilité ne peut pas se réduire au traitement de la région de frontière du domaine faisable car les trajectoires se caractérisent par des sauts et l'incidence de la trajectoire avec la frontière ne peut pas profiter de la continuité de la dynamique. L'objectif principal devient donc la construction des ensembles positifs invariants (voir les travaux de F. Blanchini et ses coauteurs pour une familiarisation avec ces concepts classiques et leur déclinaison dans le cas des dynamiques linéaires [24], [25]) pour les systèmes PWA en temps discret vus dans leur ensemble. Les propriétés contractives de la dynamique directe ou expansive de la dynamique en temps inverse seront les éléments de base pour notre approche qui veut identifier les ensembles invariants comme des points fixes pour une dynamique définie sur un ensemble compact de l'espace d'état. Au passage on mentionne que les aspects de convergence en temps fini des itérations sur les ensembles se rapprochent de la théorie classique des ensembles maximaux admissibles par rapport aux contraintes sur les sorties [26]. D'autre part, l'analyse d'atteignabilité [27] et la théorie de la viabilité [28] peuvent être aussi considérés comme des domaines proche de la problématique traitée.

1.4 Contribution

La thèse retrace les principes généraux des formulations explicites pour la commande prédictive et souligne la structure linéaire par morceaux de la dynamique.

Les contributions principales de ce travail de thèse résident dans la construction des ensembles invariants pour les systèmes affines par morceaux. Les résultats se concentrent sur le traitement des systèmes PWA (affines par morceaux) définis sur une partition polyédrale de l'espace d'état même si les principes sont applicables dans un cadre plus général.

Trois constructions remarquables peuvent être mises en évidence :

- la construction expansive,
- la construction contractive,
- la construction basée sur les graphes des transitions.

Au niveau méthodologique toutes ces constructions étant basées sur la dynamique directe ou en temps inverse des ensembles de l'espace d'état, elles impliquent un traitement géométrique au moins dans la partie de comparaison avec le domaine faisable qui peut s'avérer gourmand en temps de calcul. Une solution innovante a été proposée en exploitant l'analyse par intervalles.

Il est intéressant d'observer que la construction des ensembles invariants ouvre la voie au post-traitement des lois de commandes prédictives en vue de la maximisation de leur domaine de fonctionnement avec garantie de sûreté. Des comparaisons sont faites entre les différentes structures MPC avec d'une part les formulations qui bénéficient du renforcement de l'invariance dès la phase de synthèse et d'autre part les formulations explicites qui bénéficient de la post-analyse pour la caractérisation des domaines viables.

En outre, le présent travail fait état des extensions MPC à base de ces méthodes géométriques pour le suivi de trajectoire, pour la prise en compte des incertitudes paramétriques ou d'un retard variable à l'entrée du système. Une grande partie de ces développements théoriques sont illustrés par des exemples au fur et à mesure de leur introduction. Le mémoire contient aussi l'étude d'un problème de suivi de trajectoire et de faisabilité/viabilité d'un certain profil, avec application pour la production d'électricité en conjonction avec la caractérisation d'une vallée hydraulique.

1.5 Organisation

Le présent manuscrit est organisé en 5 parties. Le chapitre suivant est dédié aux généralités de la commande prédictive et à quelques définitions concernant l'invariance et la stabilité. Le chapitre 3 regardera en détail les propriétés des systèmes dynamiques affines par morceaux en plaçant cette classe de système dans la famille des systèmes hybrides et en balayant quelques méthodes classiques d'analyse de stabilité. Le chapitre 4 contient les principaux résultats concernant la construction des ensembles invariants ; l'utilité de ces résultats théoriques et méthodologiques sera illustrée dans le cinquième chapitre où les extensions de la commande prédictive sont détaillées pour mettre en lumière les caractéristiques liées à l'invariance positive. Le chapitre 6 traite un problème industriel de viabilité de fonctionnement d'un système dynamique sous contraintes ; une partie des techniques géométriques proposées dans cette thèse voit son illustration dans le calcul des trajectoires faisables (viables).

Chapitre 2

Commande Prédictive

2.1 Introduction

La commande prédictive basée sur modèle (MPC) est une méthodologie de commande basée sur l'optimisation (sous contraintes) à chaque pas d'échantillonnage et sur un horizon glissant d'un certain critère de coût qui prend en compte de façon générale l'écart entre la trajectoire prédite par le modèle du système et la consigne de référence. Une fois le problème d'optimisation résolu (séquence de commande calculée) on applique la première commande (le premier élément de la séquence optimale), ensuite on considère la mise à jour de l'état du système pour faire glisser la fenêtre de prédiction et ainsi réitérer la procédure d'optimisation.

Parmi les avantages de la commande MPC on peut citer :

- MPC a l'avantage de considérer les contraintes sur la commande, les variables d'états et de sorties dès la phase de conception,
- MPC fournit une loi de commande en boucle fermée à cause de la résolution à chaque pas d'échantillonnage d'un problème d'optimisation et de l'application de la première partie de la séquence de commande seulement,
- MPC possède de bonnes performances en poursuite de trajectoire parce qu'elle utilise à chaque pas d'échantillonnage une fenêtre (qui se déplace) du signal de référence, qui peut ainsi exploiter toute information non causale disponible au moment de la synthèse.

Parmi les inconvénients de la commande MPC on peut citer :

- MPC requiert un certain temps de calcul pour résoudre le problème d'optimisation. Ce temps de calcul doit être inférieur à la période d'échantillonnage du système, donc la commande MPC doit gérer des contraintes temps réel pour les systèmes avec une dynamique rapide. Ce problème a reçu une attention spécifique dans les dix dernières années par le calcul des contrôleurs explicites hors-ligne.

2.2 Les bases de la commande prédictive

2.2.1 Modèle de prédiction à temps discret

Soit un système LTI continu :

$$\begin{cases} \dot{x}_c(t) = A_c x_c(t) + B_c u_c(t) \\ y_c(t) = C_c x_c(t) \end{cases} \quad (2.1)$$

avec $y(k) = y_c(kT_s)$ et $u_c(t) = u_k$ pour $kT_s + \tau \leq t < (k+1)T_s + \tau$, avec τ le retard dû à l'échantillonnage. D'après la figure 2.1 $\dot{x}_c(t)$ s'écrit comme :

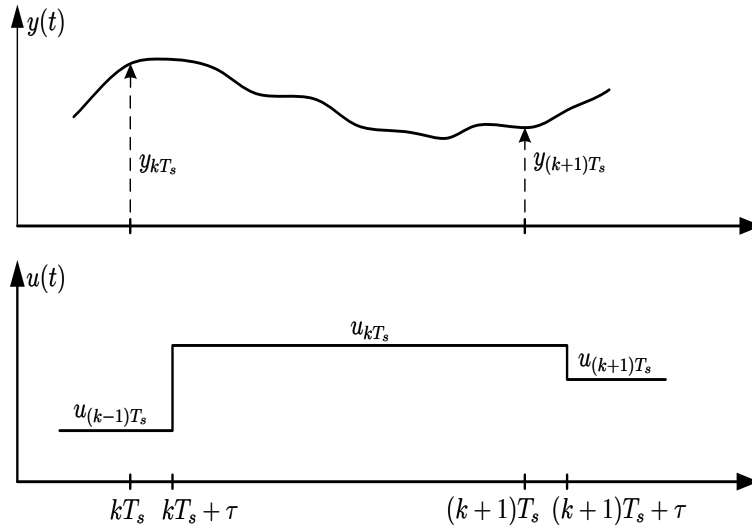


FIGURE 2.1 – Temps de mesure et temps d'application du signal de commande.

$$\dot{x}_c(t) = \begin{cases} A_c x_c(t) + B_c u_{k-1} & \text{pour } kT_s \leq t < kT_s + \tau \\ A_c x_c(t) + B_c u_k & \text{pour } kT_s + \tau \leq t < (k+1)T_s \end{cases} \quad (2.2)$$

La solution de l'équation (2.1) est :

$$x_c(t) = e^{A_c(t-t_0)} x_c(t_0) + \int_{t_0}^t e^{A_c(t-\theta)} B_c u_c(\theta) d\theta \quad (2.3)$$

avec t_0 l'instant initial et $x_c(t_0)$ l'état initial. On a donc :

– pour $kT_s \leq t < kT_s + \tau$:

$$x_c(kT_s + \tau) = e^{A_c \tau} x_c(kT_s) + \left[\int_{kT_s}^{kT_s + \tau} e^{A_c(kT_s + \tau - \theta)} d\theta \right] B_c u_{k-1} \quad (2.4)$$

En posant $\eta = kT_s + \tau - \theta \Rightarrow d\eta = -d\theta$ alors :

$$\int_{kT_s}^{kT_s + \tau} e^{A_c(kT_s + \tau - \theta)} d\theta = \int_0^\tau e^{A_c \eta} d\eta = \Gamma_1 \quad (2.5)$$

$$\Rightarrow x_c(kT_s + \tau) = e^{A_c\tau} x_c(kT_s) + \Gamma_1 B_c u(k-1) \quad (2.6)$$

– pour $kT_s + \tau \leq t \leq (k+1)T_s$:

$$\begin{aligned} x_c((k+1)T_s) &= e^{A_c(T_s-\tau)} x_c(kT_s + \tau) + \left[\int_{kT_s+\tau}^{(k+1)T_s} e^{A_c((k+1)T_s-\theta)} d\theta \right] B_c u_k \\ &= e^{A_c(T_s-\tau)} [e^{A_c\tau} x_c(kT_s) + \Gamma_1 B_c u_{k-1}] + \left[\int_0^{T_s-\tau} e^{A_c\eta} d\eta \right] B_c u_k \\ &= e^{A_c T_s} x_c(kT_s) + e^{A_c(T_s-\tau)} \Gamma_1 B_c u_{k-1} + \left[\int_0^{T_s-\tau} e^{A_c\eta} d\eta \right] B_c u_k \end{aligned}$$

En notant $x_k = x_c(kT_s)$, on peut donc écrire :

$$x_{k+1} = Ax_k + B_1 u_{k-1} + B_2 u_k \quad (2.7)$$

avec : $A = e^{A_c T_s}$, $B_1 = e^{A_c(T_s-\tau)} \Gamma_1 B_c$, $B_2 = \left[\int_0^{T_s-\tau} e^{A_c\eta} d\eta \right] B_c$.

Pour la mise en forme standard, un changement de variable peut être utilisé en posant

$\xi_k = \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}$, on obtient :

$$\begin{cases} \xi_{k+1} = \begin{bmatrix} A & B_1 \\ 0 & 0 \end{bmatrix} \xi_k + \begin{bmatrix} B_2 \\ I \end{bmatrix} u_k \\ y_k = \begin{bmatrix} C & 0 \end{bmatrix} \xi_k \end{cases} \quad (2.8)$$

Notons que si $\tau = 0$, alors $B_1 = 0$ et si $\tau = T_s$, alors $B_2 = 0$.

2.2.2 Modèle de prédiction avec intégrateur

La méthodologie de la commande prédictive est basée sur un modèle qui permet de prédire le comportement du système dans le futur sur une fenêtre de longueur donnée. La fiabilité du modèle est primordiale pour la qualité de la prédiction, donc il faut prendre toutes les précautions nécessaires dans la phase de modélisation.

En partant des principes classiques de la théorie des systèmes linéaires et régulation, il est bien connu le fait que pour une erreur stationnaire nulle, la fonction de transfert en boucle ouverte doit contenir un intégrateur. La plus simple façon d'obtenir cette garantie dans le contexte prédictif est de construire un modèle de prédiction dans l'incrément de la commande qui contiendra donc implicitement l'intégrateur de la chaîne directe. Pour un système LTI discret comme précédemment :

$$\begin{cases} \tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}u_k \\ y_k = \tilde{C}\tilde{x}_k \end{cases} \quad (2.9)$$

avec $\tilde{x}_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ et $y_k \in \mathbb{R}^q$ l'état, la commande et la sortie du système respectivement, si on voudrait changer (augmenter) le modèle (2.9) pour qu'il inclut dans sa représentation un intégrateur, on peut poser :

$$\tilde{x}_{k+1} - \tilde{x}_k = \tilde{A}(\tilde{x}_k - \tilde{x}_{k-1}) + \tilde{B}(u_k - u_{k-1}) \quad (2.10)$$

En notant $\Delta\tilde{x}_k = \tilde{x}_k - \tilde{x}_{k-1}$ et $\Delta u_k = u_k - u_{k-1}$ on obtient le modèle augmenté suivant :

$$\left\{ \begin{array}{l} \underbrace{\begin{bmatrix} \Delta\tilde{x}_{k+1} \\ y_{k+1} \end{bmatrix}}_{x_{k+1}} = \underbrace{\begin{bmatrix} \tilde{A} & 0_m^T \\ \tilde{C}\tilde{A} & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} \Delta\tilde{x}_k \\ y_k \end{bmatrix}}_{x_k} + \underbrace{\begin{bmatrix} \tilde{B} \\ \tilde{C}\tilde{B} \end{bmatrix}}_B \Delta u_k \\ y_k = \underbrace{\begin{bmatrix} 0_m & 1 \end{bmatrix}}_C \begin{bmatrix} \Delta\tilde{x}_k \\ y_k \end{bmatrix} \end{array} \right. \quad (2.11)$$

donc le système augmenté (2.11) sera commandé par la variation ou bien l'incrément de la commande Δu_k et non pas par u_k .

2.2.3 Séquence de commande optimale

La construction de la séquence de commande optimale sur un horizon fini en partant de l'information sur l'état à instant k revient à la résolution d'un problème d'optimisation dont la fonction de coût à optimiser a la forme générale suivante :

$$J_k = \sum_{i=1}^N \|y(k+i|k) - r(k+i|k)\|_Q^2 + \sum_{i=0}^{N_u-1} \|\Delta u(k+i|k)\|_R^2 \quad (2.12)$$

avec :

N : horizon de prédiction.

N_u : horizon de commande, et pour $i \geq N_u$, $\Delta u(k+i|k) = 0$.¹

$Q = Q^T$: matrice de pondération sur l'erreur de sortie par rapport à la consigne r_k , avec $Q \succ 0$.

$R = R^T$: matrice de pondération sur la variation de la commande Δu , avec $R \succcurlyeq 0$.

Le facteur de pondération R pénalise la norme des variations de la commande Δu_k , compris comme l'effort (énergie) de commande. Dans certain cas le terme $\|\Delta u(k+i|k) - U_0\|_R^2$ peut être ajouté pour pénaliser l'écart entre Δu et une valeur constante U_0 . Tous les paramètres N , N_u , Q , R et la trajectoire de référence r_k affectent le comportement du processus (en boucle fermée) par le mécanisme de prédiction. Ces paramètres (spécialement les matrices Q et R) doivent être ajustés pour satisfaire les performances dynamiques voulues pour le système à commander en choisissant le compromis adéquat entre l'erreur de suivi et l'énergie dépensée par la commande.

La fonction de coût (2.12) peut être développée comme suit :

$$J_k = (Y_{ref} - Y)^T \bar{Q} (Y_{ref} - Y) + \Delta U^T \bar{R} \Delta U, \quad (2.13)$$

avec :

$$\left\{ \begin{array}{l} Y_{ref} = \begin{bmatrix} r_{k+1} & r_{k+2} & r_{k+3} & \dots & y_{k+N} \end{bmatrix}^T \\ Y = \begin{bmatrix} y_{k+1} & y_{k+2} & y_{k+3} & \dots & y_{k+N} \end{bmatrix}^T \end{array} \right. \quad (2.14)$$

1. A noter que la valeur de $\Delta u(k+i|k)$ pour $i \geq N_u$ peut avoir une autre expression prédéfini, par exemple dans une forme générale $\Delta u(k+i|k) = Kx(k+i|k)$. Ce choix s'avère intéressante pour les modèles instables en boucle ouverte. Le choix $\Delta u(k+i|k) = 0$ reste toutefois la plus courante pour modèles stable en boucle ouverte.

$$\Delta U = [\Delta u_k \quad \Delta u_{k+1} \quad \Delta u_{k+2} \quad \dots \quad \Delta u_{k+N_u-1},]^T \quad (2.15)$$

$$\bar{R} = \text{diag}(R) \in \mathbb{R}^{mN_u \times mN_u} \quad \bar{Q} = \text{diag}(Q) \in \mathbb{R}^{qN \times qN} \quad (2.16)$$

et

$$Y = Fx_k + \Phi \Delta U \quad (2.17)$$

$$\text{avec : } F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^N \end{bmatrix}; \Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \dots & CA^{N-N_u}B \end{bmatrix}.$$

Pour identifier le minimum de la fonction de cout (2.13), la satisfaction des conditions nécessaire peut être utilisée :

$$\frac{\partial J_k(x_k, \Delta U, Y_{ref})}{\partial \Delta U} = 0 \quad (2.18)$$

en ayant plus précisément pour le critère de coût (2.13) :

$$J_k = (Y_{ref} - Y)^T \bar{Q} (Y_{ref} - Y) + \Delta U^T \bar{R} \Delta U \quad (2.19)$$

$$= Y_{ref}^T \bar{Q} Y_{ref} - 2Y_{ref}^T \bar{Q} Y + Y^T \bar{Q} Y + \Delta U^T \bar{R} \Delta U \quad (2.20)$$

$$= Y_{ref}^T \bar{Q} Y_{ref} - 2Y_{ref}^T \bar{Q} (Fx_k + \Phi \Delta U) \quad (2.21)$$

$$+ (Fx_k + \Phi \Delta U)^T \bar{Q} (Fx_k + \Phi \Delta U) + \Delta U^T \bar{R} \Delta U \quad (2.22)$$

$$= \Delta U^T (\Phi^T \bar{Q} \Phi + \bar{R}) \Delta U - 2(Y_{ref} - Fx_k)^T \bar{Q} \Phi \Delta U + Y_{ref}^T \bar{Q} Y_{ref} \quad (2.23)$$

La solution optimale pour la séquence de commande dans le cas sans contraintes, ΔU est obtenue pour :

$$\Delta U = (\Phi^T \bar{Q} \Phi + \bar{R})^{-1} \Phi^T \bar{Q} (Y_{ref} - Fx_k), \quad (2.24)$$

avec la supposition que $(\Phi^T \bar{Q} \Phi + \bar{R})^{-1}$ est bien définie (hypothèse pour laquelle une condition suffisante est $R \succ 0$).

2.2.4 Commande à horizon glissant

Pour la commande MPC le problème d'optimisation (2.13) est résolu à chaque pas d'échantillonnage pour trouver une solution ΔU optimale en fonction de l'évolution de x_k et Y_{ref} . Après l'application de la première commande (le premier élément de ΔU), on acquiert le nouveau vecteur d'état et on réitère la procédure selon le principe de l'horizon glissant illustré dans la figure 2.2.

Le premier élément de ΔU peut être obtenu :

$$\begin{aligned} \Delta u_k(Y_{ref}, x_k) &= \underbrace{[1 \quad 0 \quad 0 \quad \dots \quad 0]}_{N_u} (\Phi^T \bar{Q} \Phi + \bar{R})^{-1} \Phi^T \bar{Q} (Y_{ref} - Fx_k) \\ &= K_y Y_{ref} - K_{MPC} x_k \end{aligned} \quad (2.25)$$

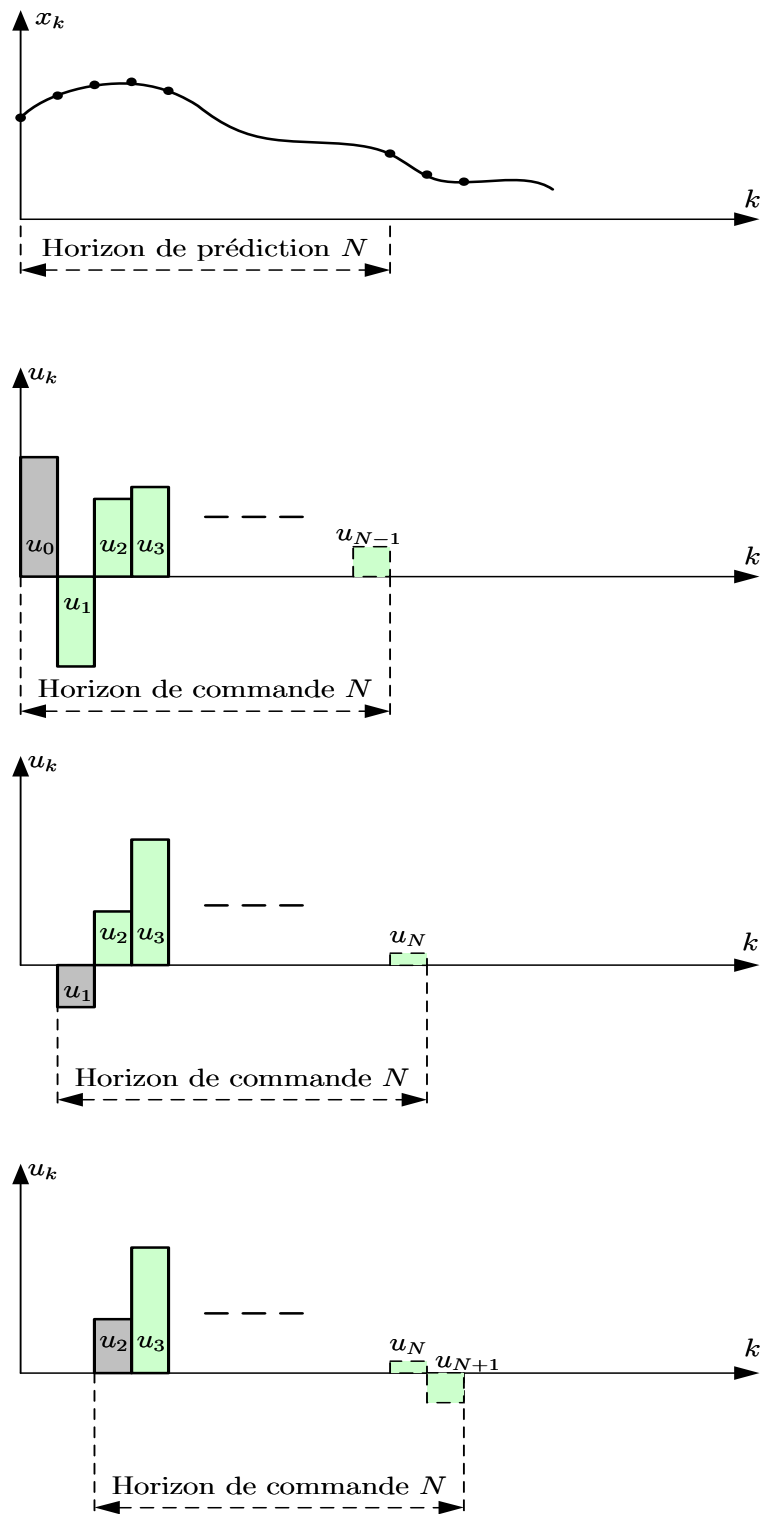


FIGURE 2.2 – Principe de commande prédictive sur un horizon fixe et glissant.

avec K_y la première ligne de $(\Phi^T \bar{Q} \Phi + \bar{R})^{-1} \Phi^T \bar{Q}$, et K_{MPC} la première ligne de $(\Phi^T \bar{Q} \Phi + \bar{R})^{-1} \Phi^T \bar{Q} F$.

En appliquant la commande MPC sans contrainte pour le système (2.11), la dynamique en boucle fermée devient :

$$\begin{aligned} x_{k+1} &= Ax_k + B\Delta u_k \\ &= Ax_k + B(K_y Y_{ref} - K_{\text{MPC}} x_k) \\ &= (A - BK_{\text{MPC}})x_k + BK_y Y_{ref} \end{aligned} \quad (2.26)$$

où on retrouve donc la composante de rétroaction et la composante de la commande par anticipation. La figure 2.3 montre l'implantation du système (2.26).

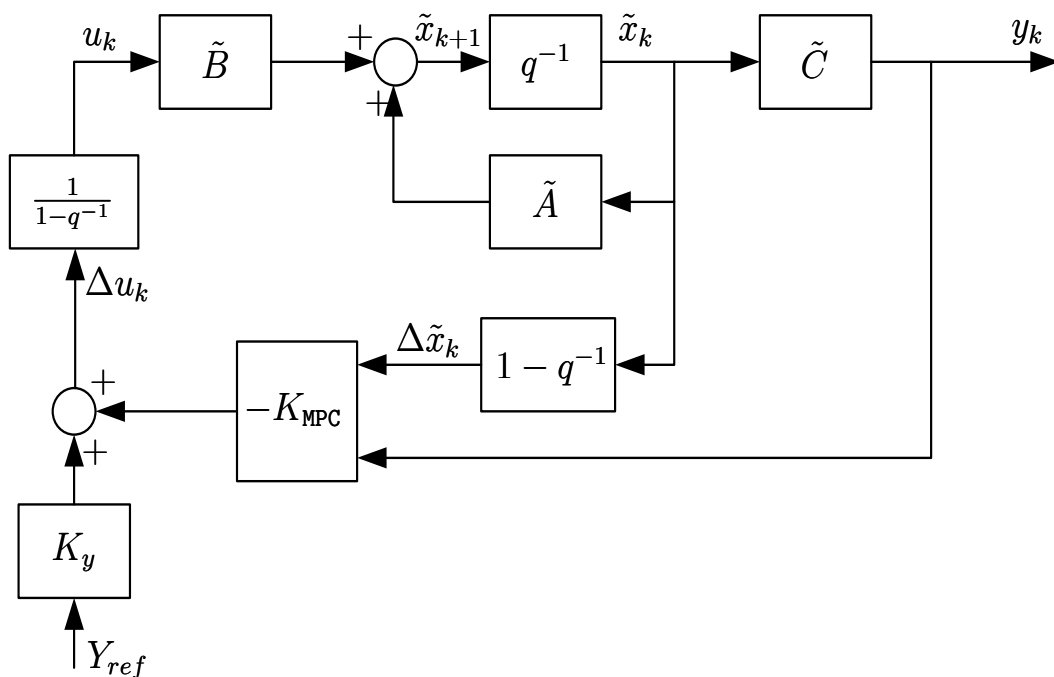


FIGURE 2.3 – Schéma de principe d'une commande prédictive.

2.2.5 Commande MPC avec contraintes

En fonction du modèle de prédiction choisi et des objectifs de commande, différentes contraintes peuvent être ajoutées à la formulation de la commande prédictive. Parmi les plus usuelles on retrouve :

- contraintes sur l'incrément de la commande Δu_k ,
- contraintes sur la commande u_k ,
- contraintes sur la sortie du système y_k .

Une écriture compacte de l'ensemble de ces contraintes peut être obtenue dans le cas linéaire. Sachant que les variables de décision de la fonction de coût (2.12) sont ΔU , il est clair qu'on devrait écrire toutes les contraintes en fonction de ΔU et des variables de contexte (l'état actuel notamment mais aussi la trajectoire future).

Pour l'incrément de la commande il est facile d'écrire :

$$\underline{\Delta U} \leq \Delta U \leq \overline{\Delta U} \quad (2.27)$$

qui peut être réécrite :

$$\Delta U \leq \overline{\Delta U} \quad (2.28)$$

$$-\Delta U \leq -\underline{\Delta U} \quad (2.29)$$

ou bien sous forme plus compacte :

$$\underbrace{\begin{bmatrix} -\mathbb{I} \\ \mathbb{I} \end{bmatrix}}_{M_1} \Delta U \leq \underbrace{\begin{bmatrix} -\underline{\Delta U} \\ \overline{\Delta U} \end{bmatrix}}_{N_1} \quad (2.30)$$

Par la suite pour exprimer les contraintes sur la commande, u_k s'écrit en fonction de ΔU :

$$\begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N_u-1} \end{bmatrix} = \underbrace{\begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \end{bmatrix}}_{C_1} u_{k-1} + \underbrace{\begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ I & I & 0 & \cdots & 0 \\ I & I & I & \cdots & 0 \\ \vdots & & & & \\ I & I & I & \cdots & I \end{bmatrix}}_{C_2} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \Delta u_{k+2} \\ \vdots \\ \Delta u_{k+N_u-1} \end{bmatrix} \quad (2.31)$$

donc les contraintes sur la commande peuvent être écrites :

$$-(C_1 u_{k-1} + C_2 \Delta U) \leq -\underline{U} \Leftrightarrow -C_2 \Delta U \leq -\underline{U} + C_1 u_{k-1} \quad (2.32)$$

$$(C_1 u_{k-1} + C_2 \Delta U) \leq \overline{U} \Leftrightarrow C_2 \Delta U \leq \overline{U} - C_1 u_{k-1} \quad (2.33)$$

ou bien sous une forme plus compacte :

$$\underbrace{\begin{bmatrix} -C_2 \\ C_2 \end{bmatrix}}_{M_2} \Delta U \leq \underbrace{\begin{bmatrix} -\underline{U} + C_1 u_{k-1} \\ \overline{U} - C_1 u_{k-1} \end{bmatrix}}_{N_2} \quad (2.34)$$

avec : \overline{U} et \underline{U} des vecteurs colonnes qui contiennent les valeurs maximale et minimale pour la commande.

Pour les contraintes sur la sortie du système on a :

$$\underline{Y} \leq F x_k + \Phi \Delta U \leq \overline{Y} \quad (2.35)$$

$$\Phi \Delta U \leq \overline{Y} - F x_k \quad (2.36)$$

$$-\Phi \Delta U \leq -\underline{Y} + F x_k \quad (2.37)$$

ou sous forme compacte :

$$\underbrace{\begin{bmatrix} -\Phi \\ \Phi \end{bmatrix}}_{M_3} \Delta U \leq \underbrace{\begin{bmatrix} -\underline{Y} + F x_k \\ \overline{Y} - F x_k \end{bmatrix}}_{N_3} \quad (2.38)$$

Finalement on peut regrouper toutes les contraintes sous cette forme :

$$\underbrace{\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix}}_M \Delta U \leq \underbrace{\begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix}}_\gamma \quad (2.39)$$

En rappelant que le critère de coût (2.13) peut être écrit :

$$J_k = \Delta U^T \underbrace{(\Phi^T \bar{Q} \Phi + \bar{R})}_H \Delta U - 2 \underbrace{(Y_{ref} - F x_k)^T \bar{Q} \Phi}_{f^T} \Delta U + Y_{ref}^T \bar{Q} Y_{ref} \quad (2.40)$$

En ignorant la partie constante qui n'influe pas sur l'argument optimal, le critère de coût sous contrainte à optimiser s'écrit alors :

$$\begin{aligned} \min & 0.5 \Delta U^T H \Delta U + f^T \Delta U \\ \text{sujet à : } & M \Delta U \leq \gamma \end{aligned} \quad (2.41)$$

La résolution de ce problème fait l'objet du paragraphe 2.3. Auparavant l'exemple 1 montre l'intérêt de considérer les contraintes d'un système à commander dès la phase de conception.

Exemple 1. Considérons la commande classique LQG d'un système linéaire du deuxième ordre à phase non minimale :

$$sys = \frac{s-1}{s^2-0.1s} \quad (2.42)$$

La commande est soumise à la contrainte $|u_k| \leq 2$. Une loi de commande LQG est calculée de telle sorte qu'elle n'atteint jamais ces limites. La fonction de coût utilisée pour le calcul de cette commande est :

$$V = \sum_{i=0}^{\infty} x_k^T Q x_k + R u_k^2 \quad (2.43)$$

avec $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ et $R = 650$. La loi de commande obtenue est :

$$u_k = [0.32985 \quad -0.29662] x_k \quad (2.44)$$

La figure 2.5(a) nous montre l'évolution de la sortie et permet de vérifier le respect de contraintes sur la commande. Maintenant on veut rendre la réponse du système plus rapide, pour cela on réduit la pondération sur la commande pour qu'elle devienne $R = 0.1$. Le retour d'état est :

$$K = [1.7927 \quad -0.98221] \quad (2.45)$$

et la commande appliquée au système $u_k = K x_k$ ne respecte plus les contraintes imposées comme on peut le voir sur figure 2.5(b) (à noter que le système devient plus rapide, voir l'échelle de temps). Donc pour éviter la violation des contraintes sur la commande on ajoute un élément de saturation **sat** défini comme :

$$\text{sat}(u_k) = \begin{cases} 2 & \text{si } u_k > 2 \\ u_k & \text{si } |u_k| \leq 2 \\ -2 & \text{si } u_k < -2 \end{cases} \quad (2.46)$$

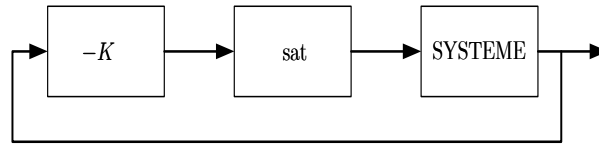


FIGURE 2.4 – Le système avec loi de commande et élément de saturation pour éviter la violation des contraintes.

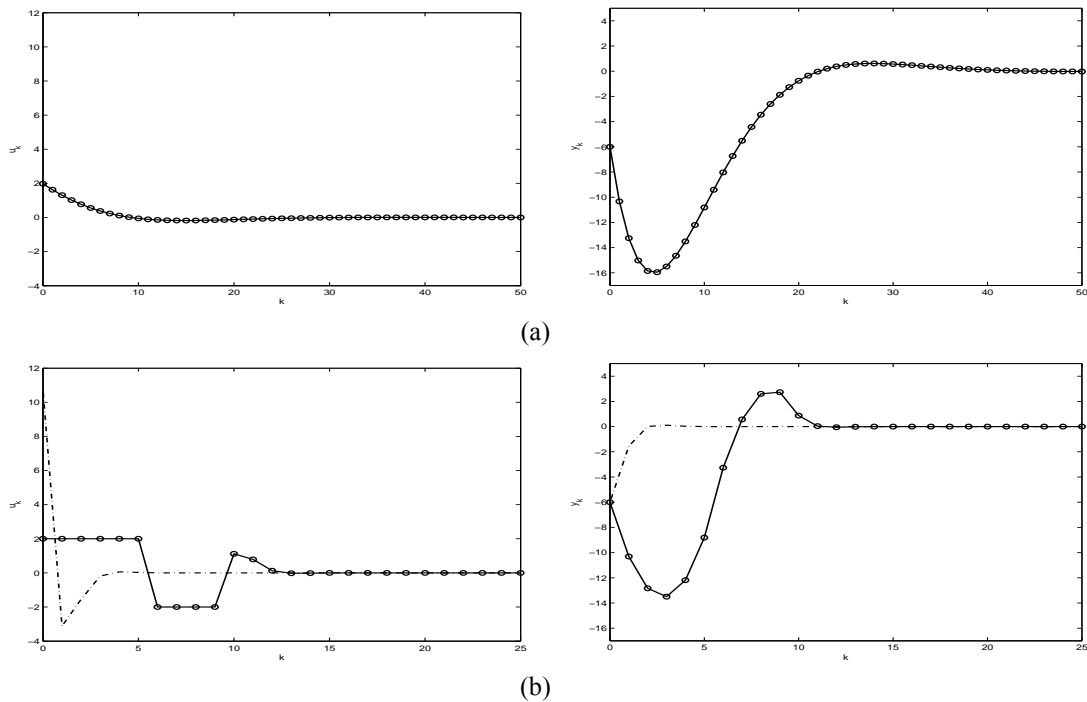


FIGURE 2.5 – (a) Simulation du système avec une loi de commande initiale qui satisfait les contraintes implicitement. (b) Simulation du système avec une loi de commande sans puis avec saturation.

La figure 2.4 illustre le schéma bloc de la commande avec élément de saturation. Le fait d'ajouter cet élément de saturation aveugle dégrade les performances du système (figure 2.5(b)).

Maintenant on considère une loi de commande prédictive qui prend en considération les contraintes dès la phase de conception, avec les paramètres suivants :

- horizon de prédiction $N = 5$,
- matrice de pondération sur l'état $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$,
- pondération sur la commande $R = 0.1$,
- contrainte sur la commande $|u| \leq 2$

La figure 2.6 montre l'évolution temporelle de la sortie du système. Le fait de considérer les contraintes en phase de la conception de la loi de commande permet d'obtenir les performances en adaptant les actions de commandes d'une fenêtre de prédiction par rapport à l'activation des contraintes.

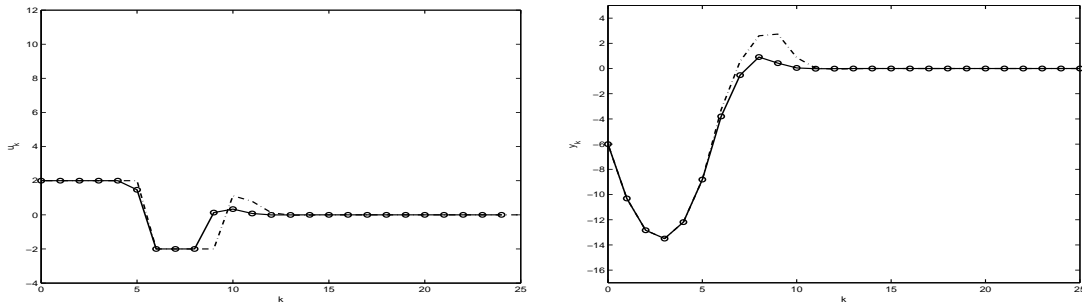


FIGURE 2.6 – Comparaison de la loi de commande précédente avec saturation et la loi de commande prédictive.

La figure 2.7 résume le principe de la commande prédictive dans le cas général d'un problème de commande optimale sous contraintes sur un horizon glissant.

2.3 Commande prédictive et optimisation paramétrique

2.3.1 Optimisation paramétrique

Une solution numérique pour un problème d'optimisation est dite soit explicite soit implicite [21]. 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 en mettant en évidence des quantités connues, le calcul est alors *explicite*.

On considère la formulation générale d'un problème d'optimisation quadratique sous contraintes :

$$\begin{aligned} \arg \min_U & 0.5U^T H U + p^T F U \\ \text{sous : } & A U \leq B p + b \end{aligned} \quad (2.47)$$

où p est le vecteur des paramètres d'optimisation. La solution générale du problème (2.47) peut être écrite sous forme compacte [21] :

$$U^*(p) = \left[(H^{-1} \bar{A}^T (\bar{A} H^{-1} \bar{A}^T) \bar{A} H^{-1} - H^{-1}) F + H^{-1} \bar{A}^T (\bar{A} H^{-1} \bar{A}^T)^{-1} \bar{B} \right] p + H^{-1} \bar{A}^T (\bar{A} H^{-1} \bar{A}^T)^{-1} \bar{b} \quad (2.48)$$

où \bar{A} est une matrice (de rang plein) formée avec un sous ensemble de A correspondant aux contraintes inégalité saturées par la valeur optimale, et \bar{B}, \bar{b} sont définis de façon similaire pour la partie droite des contraintes. Cette solution n'est pas globalement linéaire en le vecteur de paramètres p et en plus, localement elle présente une dépendance affine. Les matrices ont une structure variable en fonction des contraintes saturées, donc on ne peut pas les utiliser pour un calcul direct de la solution.

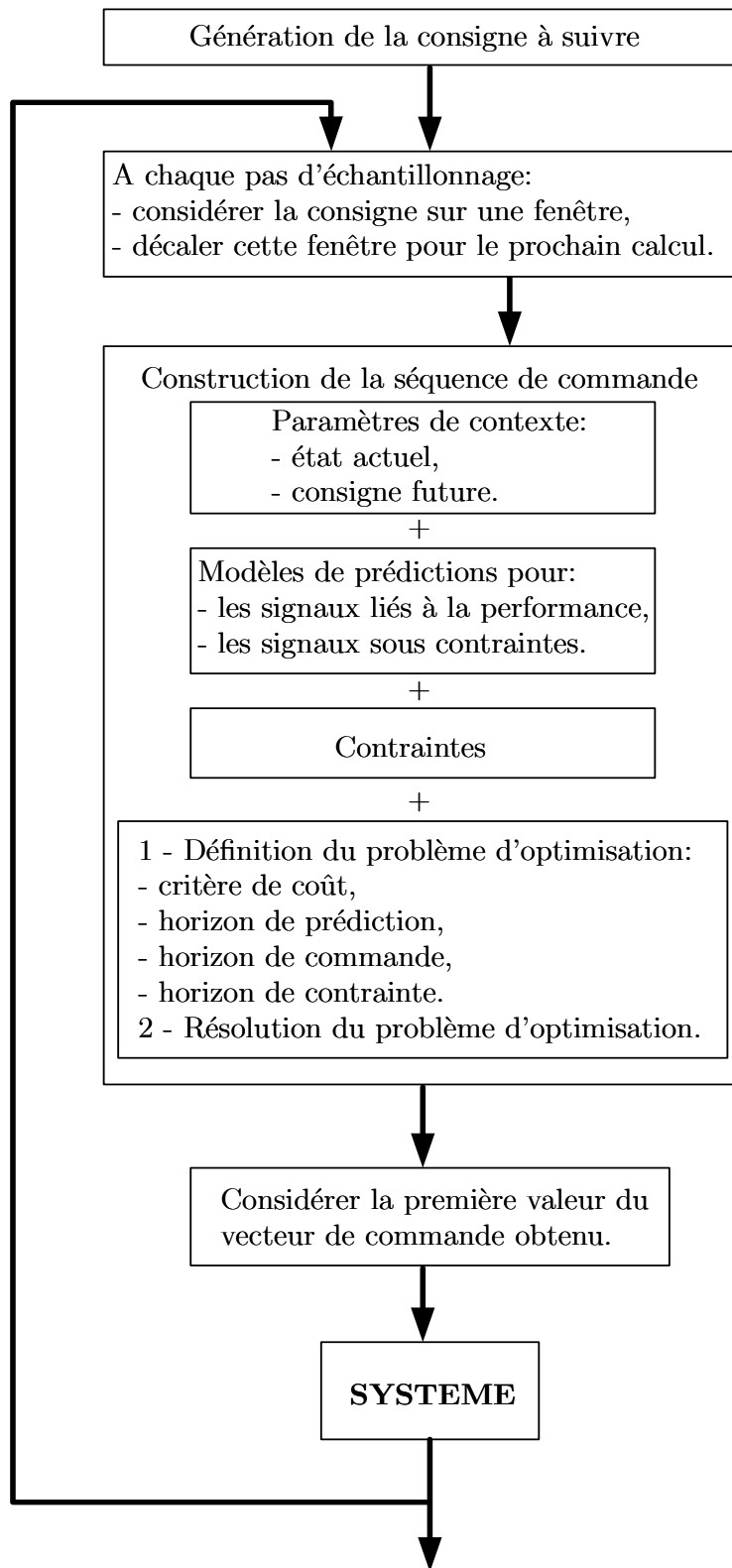


FIGURE 2.7 – Schéma de principe de la commande prédictive.

2.3.2 Méthodes de construction de la solution explicite

Plusieurs méthodes pour la construction des solutions explicites pour les problèmes d'optimisation multiparamétriques peuvent être trouvées dans la littérature [19, 29, 21]. Les paragraphes suivants présentent succinctement la méthode développée en [19].

Méthode basée sur les conditions KKT

En considérant le problème d'optimisation (2.47) et en appliquant la transformation :

$$Z = U + H^{-1}Fp \quad (2.49)$$

la paramétrisation de la fonction de coût est transférée exclusivement vers la paramétrisation des contraintes. En effet le problème s'écrit sous la forme :

$$\begin{aligned} \min_Z \quad & 0.5Z^T H Z \\ \text{sous : } \quad & AZ \leq Bp + b \end{aligned} \quad (2.50)$$

la méthode part de la caractérisation de la solution en un point spécifique que $U^*(p_0)$ et avec les conditions KKT :

$$\begin{cases} HZ + A^T \lambda = 0, \lambda \in \mathbb{R}^r, \\ \lambda \geq 0, \\ \lambda^T (AZ - b - Bp_0) = 0, \\ AZ \leq Bp_0 + b. \end{cases} \quad (2.51)$$

on obtient la caractérisation locale de la solution optimale au voisinage de p_0 :

$$\mathcal{R}_0 : \begin{cases} AH^{-1}\bar{A}^T(\bar{A}H^{-1}\bar{A}^T)^{-1}(\bar{b} + \bar{B}p) \leq Bp + b, \\ (\bar{A}H^{-1}\bar{A}^T)^{-1}(\bar{b} + \bar{B}p) \leq 0. \end{cases} \quad (2.52)$$

avec \bar{A} , \bar{B} et \bar{b} identifiant les inégalités saturées pour $\{p_0, Z^*(p_0)\}$. Une fois cette région critique non vide déterminée, le reste de l'espace des paramètres est exploré en quête de nouvelles régions faisables.

La solution optimale $Z(p)^*$, obtenue localement sur \mathcal{R}_0 (où toute autre région définie comme en (2.52)) peut être mise sous la forme (2.48) et en utilisant la transformé inverse de (2.49) on obtient une solution explicite affine par morceaux.

2.3.3 Analyse des solutions explicites

Le Théorème 1 donne une caractérisation importante concernant la continuité et la convexité des problèmes d'optimisation paramétrique rencontrés dans la synthèse de lois de commande prédictives.

Théorème 1. Soit un problème d'optimisation quadratique multiparamétrique :

$$\begin{aligned} \min_U \quad & U^T H U + 2x_k^T F U + x_k^T G x_k \\ \text{sous : } \quad & A_{in} U \leq B_{in} x_k + b_{in} \end{aligned} \quad (2.53)$$

avec $H \succ 0$, $U \in \mathbb{R}^p$ et $x_k \in \mathbb{R}^n$ et pour lequel l'ensemble des paramètres faisables est $\mathcal{R} \subset \mathbb{R}^n$. La fonction explicite qui représente l'argument optimal de (2.53) : $U^* : \mathcal{R} \rightarrow \mathbb{R}^p$ est continue et linéaire aff ne par sous domaine polyédraux dans \mathcal{R} .

La solution optimale :

$$J^*(x_k) = U^{*T} H U^* + 2x_k^T F U^* + x_k^T G x_k \quad (2.54)$$

est continue, quadratique par sous-domaine polyédraux dans \mathcal{R} et convexe.

2.3.4 Commande prédictive explicite

D'une façon générale la loi de commande prédictive explicite pour un problème de suivi de trajectoire dans le cas d'un modèle de prédiction linéaire :

$$x_{k+1} = A x_k + B u_k \quad (2.55)$$

avec des contraintes linéaires conduit vers un problème de type (2.47) qui doit être résolu à chaque pas d'échantillonnage selon la variation du vecteur de paramètres p (qui contient l'état actuel x_k et les références futures sur l'horizon de prédiction $r(k+i)$, $i = 1, \dots, N$). En utilisant la solution explicite du problème d'optimisation paramétrique, la loi de commande est représentée en fait par la première composante de la solution optimale, vue comme une fonction explicite (linéaire aff ne par morceaux) de l'état actuel du système ou dans une forme plus générale de l'état courant du système et des futures références de consigne [30] :

$$u_k = F_i^{FB} x_k + F_i^{FF} \begin{bmatrix} r(k+1) \\ \vdots \\ r(k+N) \end{bmatrix} + G_i \quad (2.56)$$

definie sur un ensemble de régions polytopiques :

$$\mathbb{X} = \bigcup_{i=1}^N \mathcal{R}_i \quad (2.57)$$

avec l'intersection $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset \forall i \neq j$. F^{FB} représente le gain de la contre réaction alors que F^{FF} représente le *feedforward* qui multiplie la future consigne de référence considérée comme une donnée sur l'horizon de prédiction.

Si l'ensemble \mathbb{X} est convexe alors (2.57) représente la partition faisable de l'espace d'état et (2.55) devient :

$$x_{k+1} = (A + B F_i^{FB}) x_k + B F_i^{FF} \begin{bmatrix} r(k+1) \\ \vdots \\ r(k+N) \end{bmatrix} + B G_i \quad (2.58)$$

Quand la consigne de référence est nulle, la solution générale (2.56) est réduite à une solution d'un problème de régulation [19] :

$$u_k = F_i x_k + G_i, \quad x_k \in \mathcal{R}_i \subseteq \mathbb{X} \quad (2.59)$$

et les dynamiques en boucle fermée sont données par :

$$x_{k+1} = (A + B F_i) x_k + B G_i, \quad x_k \in \mathcal{R}_i \subseteq \mathbb{X} \quad (2.60)$$

2.3.5 Implémentation de la commande explicite

L'expression d'une loi de commande explicite est une fonction linéaire par morceaux. Cette expression 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 (le vecteur d'état du système) et comme information correspondante les fonctions linéaires associées (la loi de commande). A l'aide de cette table, l'implémentation en ligne de la commande prédictive explicite doit suivre les étapes suivantes :

- Mesurer (ou estimer) le vecteur des paramètres courant x_k ,
- Si x_k n'appartient pas à la région couverte par la solution explicite, arrêter la procédure, le problème est infaisable (éventuellement faire appel à des méthodes de récupération de la faisabilité),
- Rechercher dans la table la région critique contenant x_k ,
- 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. Cette identification se fait donc en ligne et peut engendrer une charge informatique importante si le nombre des partitions est élevé. Plusieurs méthodes ont été développées pour déterminer quelle est la région qui contient le vecteur de mesure [31, 32].

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.

Par la suite, notre travail utilise pour le calcul des solutions explicites la bibliothèque MPT développé à ETH Zurich [33]. Cette boîte à outils 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 [34].

2.4 Invariance

2.4.1 Ensembles invariants

Le concept de l'invariance joue un rôle important dans le domaine de la théorie de la commande (synthèse de commande sous contraintes, analyse de stabilité et de robustesse...etc). Étant donné un système dynamique, un sous-ensemble de l'espace d'état est dit **positivement invariant** s'il vérifie la propriété suivante : pour tout x_k appartenant à cet ensemble à un instant donné, la trajectoire future reste dans cet ensemble à tout instant [24]. Un sous-ensemble de l'espace d'état est dit **invariant** si l'appartenance à cet ensemble à un moment donné implique son inclusion dans le futur et dans le passé (donc pour une évolution en direct et en inverse également) [24]. Il y a une relation entre la notion d'ensemble positivement invariant et le concept de fonction de Lyapunov, cette relation vient du fait que si on considère une telle fonction pour l'énergie d'un système dynamique alors les courbes de niveaux de cette fonction représentent les limites ou les frontières de ces ensembles positivement invariants.

Exemple 2. On considère un pendule qui oscille autour de son axe avec un certain frottement visqueux (figure 2.8). On relâche le pendule à partir d'une position initiale θ_0 , le pendule va dissiper de l'énergie à cause du frottement, donc la trajectoire du pendule va converger vers sa position d'équilibre ($\theta_{eq} = 0$). L'intervalle défini par $-\theta_0 \leq \theta \leq \theta_0$ est un ensemble positivement invariant. Par contre si on considère que le pendule oscille sans frottement visqueux, alors l'oscillation restera entre $-\theta_0$ et θ_0 , et cet ensemble est appelé invariant.

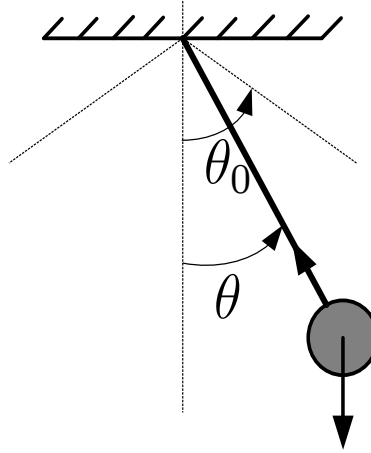


FIGURE 2.8 – Pendule oscillant autour de son axe.

On donne par la suite quelques définitions formelles [24, 25] liées à l'invariance.

Définition 1. L'ensemble $\Phi^{\text{PI}} \subseteq \mathbb{X} \subset \mathbb{R}^n$ est positivement invariant pour le système :

$$x_{k+1} = f(x_k), \quad x_k \in \mathbb{X} \quad (2.61)$$

si pour tout $x_0 \in \Phi^{\text{PI}}$, la solution $x_k \in \Phi^{\text{PI}}$ pour tout $k > 0$. Si $x_0 \in \Phi^{\text{PI}}$ implique $x_k \in \Phi^{\text{PI}}$ pour tout $k \in \mathbb{Z}$, alors on dit que Φ^{PI} est invariant.

Définition 2. L'ensemble $\Phi^{\text{RPI}} \subseteq \mathbb{X} \subset \mathbb{R}^n$ est positivement invariant de manière robuste pour le système :

$$x_{k+1} = f(x_k, w_k), \quad x_k \in \mathbb{X}, \quad w_k \in \mathbb{W} \subset \mathbb{R}^n \quad (2.62)$$

si pour tout $x_0 \in \Phi^{\text{RPI}}$ et pour tout $w_k \in \mathbb{W}$, la solution est telle que $x_k \in \Phi^{\text{RPI}}$ pour tout $k > 0$.

Définition 3. L'ensemble $\Phi^{\text{RCI}} \subseteq \mathbb{X} \subset \mathbb{R}^n$ est invariant de manière robuste par rapport à la commande pour la dynamique :

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k), \quad x_k \in \mathbb{X}, \quad w_k \in \mathbb{W} \subset \mathbb{R}^n \\ y_k &= g(x_k) \end{aligned} \quad (2.63)$$

s'il existe une loi de commande en boucle fermée :

$$u_k = \Omega(y_k) \quad (2.64)$$

qui assure l'existence d'une solution unique pour $x_0 \in \Phi^{\text{RCI}}$ et $w_k \in \mathbb{W}$ pour tout $k \geq 0$ tel que l'ensemble Φ^{RCI} est positivement invariant de manière robuste pour le système en boucle fermée.

Définition 4. L'ensemble $\mathcal{R} \subset \mathbb{R}^n$ est contractif pour le système :

$$x_{k+1} = f(x_k, u_k, w_k), \quad x_k \in \mathcal{R}, \quad w_k \in \mathbb{W} \subset \mathbb{R}^n \quad (2.65)$$

s'il existe une loi de commande $u_k(x_k)$ et un réel positif $\lambda < 1$ tels que :

$$x_k \in \mathcal{R} \Rightarrow x_{k+1} \in \lambda \mathcal{R}, \quad \forall w_k \in \mathbb{W} \subset \mathbb{R}^n \quad (2.66)$$

On présente dans ce qui suit deux grandes familles d'ensembles invariants ainsi que leurs caractéristiques à savoir les ensembles invariants ellipsoïdaux et les ensembles invariants polyédraux.

Ensembles invariant ellipsoïdaux

Considérons le système discret :

$$x_{k+1} = Ax_k \quad (2.67)$$

Soit une fonction de Lyapunov pour le système (2.67) :

$$V(x_k) = x_k^T P x_k \text{ avec } A^T P A - P = -Q \quad (2.68)$$

où $P = P^T \succ 0$ et $Q = Q^T \succ 0$ sont des matrices symétriques et définies positives, alors l'ensemble ellipsoïdal \mathcal{E} :

$$\mathcal{E} = \{x \in \mathbb{R}^n : x^T P x \leq 1\} \quad (2.69)$$

est un ensemble invariant pour le système (2.67).

Ensembles invariants polyédraux

Considérons un ensemble polyédral \mathcal{R} défini par ses contraintes :

$$\mathcal{R} = \{x \in \mathbb{R}^n | Hx \leq \bar{1}\} \quad (2.70)$$

avec $H \in \mathbb{R}^{r \times n}$ and $\bar{1} = [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^r$. Le polyèdre \mathcal{R} peut être représenté par ses sommets :

$$\mathcal{R} = \left\{ x = T\alpha, \quad \bar{1}^T \alpha = \sum_{i=1}^s \alpha_i = 1, \quad \alpha \geq 0 \right\} \quad (2.71)$$

avec $T \in \mathbb{R}^{n \times s}$ et $\alpha \in \mathbb{R}^s$. Dans la représentation (2.71) chaque colonne de la matrice T est un sommet de l'ensemble \mathcal{R} .

Le théorème suivant [24] établit les conditions de la positive invariance pour l'ensemble \mathcal{R} défini par (2.70).

Théorème 2. Les conditions suivantes sont équivalentes :

- (i) l'ensemble polyédral contractif \mathcal{R} est positivement invariant pour le système discret

$$x_{k+1} = Ax_k$$

(ii) il existe un réel positif $\lambda \leq 1$, et une matrice non négative F telle que :

$$H\bar{1} \leq \lambda\bar{1}$$

$$HF = FA$$

(iii) il existe un réel positif $\lambda \leq 1$, et une matrice non négative P telle que :

$$\bar{1}^T P \leq \lambda\bar{1}^T$$

$$TP = AT$$

La condition (ii) est basée sur le lemme suivant :

Lemme 1. Étant donnés deux polyèdres

$$\mathcal{R}_1 = \{x \in \mathbb{R}^n : H_1 x \leq k_1\}$$

et

$$\mathcal{R}_2 = \{x \in \mathbb{R}^n : H_2 x \leq k_2\}$$

alors $\mathcal{R}_1 \subseteq \mathcal{R}_2$ si et seulement s'il existe une matrice non-négative H tel que :

$$HH_1 = H_2 \tag{2.72}$$

et

$$Hk_1 \leq Hk_2 \tag{2.73}$$

L'ensemble \mathcal{R} est positivement invariant si et seulement si $x \in \mathcal{R}$ implique $Ax \in \lambda\mathcal{R}$, ainsi $FAx \leq \lambda\bar{1}$. Cela veut dire que le polyèdre \mathcal{R} doit être inclus dans le polyèdre $\{x \in \mathbb{R}^n : FAx \leq \lambda\bar{1}\}$. L'application du lemme 1 nous conduit à la condition (ii).

Pour la démonstration de la condition (iii) on considère un vecteur $x \in \mathcal{R}$. Ce vecteur peut être exprimé par la combinaison convexe des sommets de \mathcal{R} ; ce qui veut dire $x = T\alpha$. Donc le polyèdre \mathcal{R} est invariant si et seulement si :

$$\forall x_j \in \text{vert}\{\mathcal{R}\}, Ax_j \subseteq \lambda\mathcal{R} \tag{2.74}$$

ou bien de façon équivalente :

$$\forall x_j \in \text{vert}\{\mathcal{R}\}, Ax_j = Tp_j \tag{2.75}$$

avec p_j vecteur non-négatif tel que :

$$\sum_{i=1}^s p_{ij} = \bar{1}^T p_j \leq \lambda \leq 1 \tag{2.76}$$

avec p_{ij} est la $i^{\text{ème}}$ composante de p_j . En posant $P = [p_{ij}]$ on obtient la condition (iii).

On peut généraliser ce résultat comme suit :

L'ensemble \mathcal{R} est invariant par rapport à la commande pour le système :

$$x_{k+1} = Ax_k + Bu_k \quad (2.77)$$

si et seulement s'il existe un positif $\lambda \leq 1$, une matrice $P \in \mathbb{R}^{s \times s} \geq 0$, et une matrice $U \in \mathbb{R}^{m \times s} \geq 0$ telles que :

$$AT + BU = TP$$

et

$$\bar{1}P \leq \lambda \bar{1}^T$$

Les colonnes de U sont les commandes u_k associées aux colonnes de T c'est-à-dire les sommets x_k de \mathcal{R} .

2.4.2 Ensemble invariant maximal par rapport à la commande

Considérons le système LTI suivant :

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases} \quad (2.78)$$

tel que : $x_k \in \mathbb{X}$, $u_k \in \mathbb{U}$ et $y_k \in \mathbb{Y}$, $k = 0, \dots, \infty$.

L'ensemble invariant maximal par rapport à la commande Ω_∞ est calculé de la façon contractive suivante :

- $S_0 = \mathbb{X}$,
- $\Phi(S_0) = A^{-1}(S_0 \oplus (-BU)) \cap S_0$,
- $S_{i+1} = \Phi(S_i)$.

On réitère la procédure telle que $S_{k+1} = \Phi(S_k)$ jusqu'à obtenir $S_{k^*+1} = S_{k^*}$ qui est l'ensemble invariant maximal par rapport à la commande $\Omega_\infty = S_{k^*}$. Cet ensemble est convexe du fait que le système et les contraintes sont linéaires.

Exemple 3. Considérons le problème de commande d'un système double intégrateur :

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u_k \quad (2.79)$$

tel que $x_k \in \mathcal{X}$ et la commande est soumise à la contrainte $-0.5 \leq u_k \leq 0.5$. \mathcal{X} est un polytope contenant l'origine et défini comme le montre la figure 2.9 en rouge. L'ensemble invariant maximal par rapport à la commande Ω_∞ est représenté en vert.

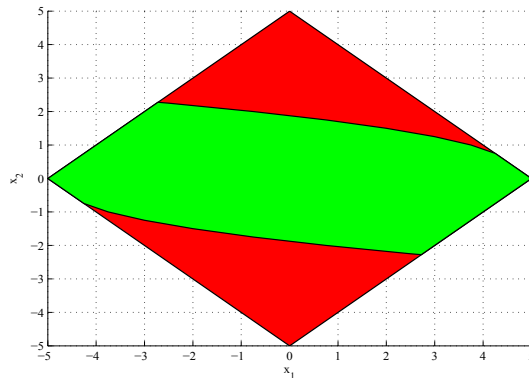


FIGURE 2.9 – (en rouge) Espace de définition de l'état. (vert) L'ensemble invariant maximal par rapport à la commande.

2.5 Stabilité de la commande prédictive avec contraintes et coûts terminaux

2.5.1 Faisabilité

On a vu que la commande prédictive est basée sur la résolution à chaque pas d'échantillonnage d'un problème d'optimisation sous contrainte et sur un horizon fini, mais aucune garantie n'est donnée pour la faisabilité (l'existence d'une solution) de ce problème d'optimisation ou de cette loi de commande, l'étude de la faisabilité devient donc primordiale avant implémentation effective. Pour un système physique, on ne peut pas admettre que son régulateur prédictif affiche un message `Problème infaisable`, mais des précautions doivent être prises pour surmonter cette situation.

On peut se contenter alors de trouver des lois de commande offrant des garanties de stabilité locale (sans contraintes). Ceci est donc une première limitation que la commande prédictive va devoir subir.

Les causes principales qui peuvent engendrer le phénomène d'infaisabilité d'une loi de commande prédictive sont les suivantes :

- la description du problème de commande sous contraintes est intrinsèquement infaisable (on dit que le jeu de contraintes est incompatible),
- une variation soudaine sur la consigne peut conduire à l'infaisabilité. Si le changement ne peut pas se faire avec des commandes à énergie limitée, les contraintes deviennent temporairement incompatibles.
- l'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 n'a pas été prise en considération quand la loi de commande a été synthétisée conduit à l'infaisabilité.
- une différence entre le modèle et le système réel induit l'infaisabilité.
- un mauvais conditionnement du problème d'optimisation de la loi MPC la rend infaisable.

La construction d'une loi de commande prédictive explicite ne garantit pas la faisabilité de cette loi de commande de l'état initial à l'état final puisqu'on peut perdre la faisabilité après quelques pas d'échantillonnage. Ce phénomène a été étudié avec intérêt dans les années 90, les solutions recherchées visant une construction de type : *“la faisabilité à l'instant 0 garantit la faisabilité à tout instant futur qui est étroitement lié au concept d'invariance positive”*. En effet on peut garantir la faisabilité de la loi de commande en essayant toujours de choisir l'état initial du système à l'intérieur de l'ensemble invariant calculé à partir de la loi de commande explicite.

Soit \mathbb{X} l'espace d'état, une contrainte terminale est définie par un ensemble $\mathcal{X} \subset \mathbb{X}$ que l'état du système doit atteindre à la fin de l'horizon de prédiction $x_N \in \mathcal{X}$. Cette contrainte est ajoutée aux contraintes du problème d'optimisation de la loi de commande prédictive.

Si on parle d'un coût terminal $x_N^T P x_N$ ajouté à la fonction de coût du critère d'optimisation sur un horizon fini, c'est pour que cette fonction devienne une fonction de Lyapunov par l'extrapolation de la prédiction à un horizon infini [35], [8], [21].

2.5.2 Loi MPC avec garantie de stabilité

L'étude de la stabilité pour une commande calculée par optimisation sur un horizon glissant n'est pas triviale. L'optimisation d'un critère de coût ne garantit pas l'obtention d'une commande stabilisante (en supposant que le problème d'optimisation soit faisable) mais plutôt une commande qui minimisera cette fonction de coût.

Pour garantir la stabilité de la commande prédictive sous contraintes pour les systèmes linéaires invariant dans le temps, une des directions les plus utilisées se base sur l'utilisation des ensembles invariants comme ensemble objectif pour la trajectoire du système. Il doit être atteint pour un horizon de prédiction donné.

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 [6].

Globalement, la liaison entre faisabilité, ensembles invariants et stabilité de la commande prédictive sous contraintes est résumée par [35].

Théorème 3. Pour une loi de commande prédictive $K : \mathbb{X} \rightarrow \mathbb{R}^m$ satisfaisant les conditions :

- $\mathcal{X}_f \subset \mathbb{X}$ et $0 \in \mathcal{X}_f$, avec \mathcal{X}_f ensemble terminal,
- $\exists K_f(x_k) \in \mathbb{U}$, $\forall x \in \mathcal{X}_f$, avec $K_f(x_k)$ une loi de commande associée à l'ensemble terminal satisfaisant les contraintes sur la commande,
- $\forall x_k$, l'état x_{k+1} résultant de l'application de la commande $K_f(x_k)$ est tel que $x_{k+1} \in \mathcal{X}_f$, avec \mathcal{X}_f l'ensemble positif invariant par rapport à la loi de commande $K_f(x_k)$,
- $\exists F : \mathbb{R}^n \rightarrow \mathbb{R}$ tel que : $F(x_{k+1}) - F(x_k) + l(x_k, K_f(x_k)) \leq 0$ où $l(x_k, K_f(x_k))$ représente l'apport de l'état x_k à la fonction de coût (il existe une fonction de Lyapunov définie sur l'ensemble \mathcal{X}_f).

la stabilité de la loi de commande est assurée à condition que la loi $K : \mathbb{X} \rightarrow \mathbb{R}^m$ amène l'état dans l'ensemble terminal \mathcal{X}_f à la fin de l'horizon de prédiction (donc que la loi soit faisable).

Ce résultat fournit des conditions suffisantes de stabilité.

Exemple 4. Considérons un système double intégrateur :

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u_k \quad (2.80)$$

soumis à des contraintes sur les commandes $-1 \leq u_k \leq 1$ et 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 $P = \begin{bmatrix} 1.808 & 0.231 \\ 0.231 & 2.648 \end{bmatrix}$ pour le coût terminal (correspondant à la solution de l'équation de Riccati associée). La figure 2.10 montre la loi explicite (calculée en utilisant [33]), sa projection sur l'espace d'état ainsi qu'une simulation de trajectoire pour l'état initial $x_0 = [-4 \ 6]^T$.

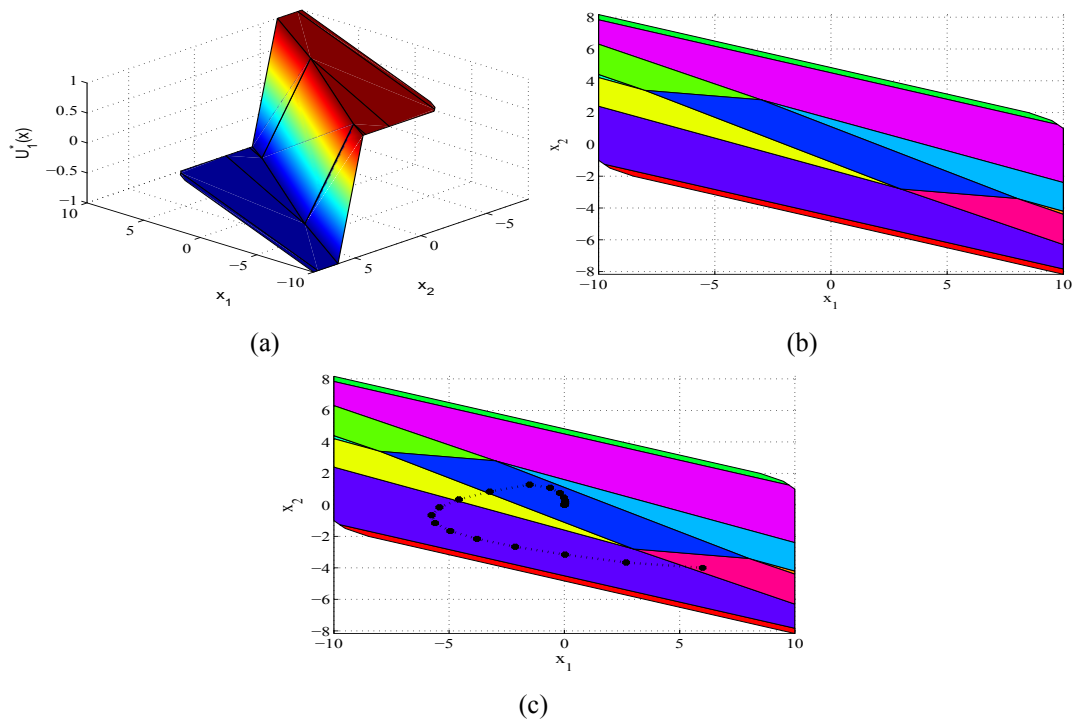


FIGURE 2.10 – (a) Loi explicite. (b) Projection de la loi explicite sur l'espace d'état. (c) Simulation de trajectoire.

2.6 Conclusion

Ce chapitre a été consacré à la commande prédictive, sa philosophie, aux notions de stabilité ainsi qu'à l'aspect géométrique des solutions analytiques. Le chapitre suivant sera dédié aux systèmes affines par morceaux pour compléter la partie d'analyse des solutions explicites mais aussi parce que, dans un cadre plus général, les algorithmes développés dans ce travail pour le calcul des ensembles invariants sont utilisables pour tout système PWA défini sur une partition de l'espace d'état.

Chapitre 3

Systemes affines par morceaux

3.1 Introduction

Ce chapitre est dédié à l'étude des systèmes affines par morceaux. Cette classe de systèmes dynamiques est importante par sa capacité de modéliser (ou approcher) une large catégorie de processus avec un caractère non linéaire, ainsi que les systèmes dont le modèle change selon l'évolution d'une région à une autre dans l'espace d'état. On verra comment certaines sous-classes de systèmes hybrides peuvent être écrites sous forme de système PWA.

La complexité de l'étude des systèmes PWA ne vient pas forcément de la complexité des dynamiques locales mais plutôt des implications que la structure des partitions peut avoir sur le comportement global. En effet, il existe des exemples de systèmes PWA instables, qui deviennent stables en changeant la structure de leurs partitions ou bien l'inverse. Même si on discute des dynamiques linéaires localement, l'étude de stabilité par l'intermédiaire d'une fonction de Lyapunov quadratique globale pour les systèmes PWA n'est pas bien adaptée. D'autres fonctions quadratiques ou linéaires définies par morceaux, qui tiennent compte de la structure des partitions, doivent être considérées. La synthèse des lois de commande pour ce type de systèmes sera évoquée brièvement en soulignant particulièrement le cas de la commande prédictive.

3.2 Systèmes hybrides

Un système hybride est un système qui fait intervenir dans son modèle d'évolution des variables continues et discrètes. Si la construction des méthodes générales pour le traitement de ce type de systèmes est un sujet d'actualité (on trouve des monographies récentes qui se fixent cet objectif [36]), il existe néanmoins quelques catégories de systèmes hybrides qui ont fait l'objet d'études pour plusieurs chercheurs à savoir [37, 38, 39] pour les modèles hybrides MLD, [40] pour les critères de stabilité des modèles PWA définis en temps continu et [41, 42] pour les modèles hybrides LC. Dans ce qui suit, on rappellera ces sous-classes de systèmes hybrides. Par la suite, des équivalences [43] entre les différents modèles vont être mentionnées pour souligner le fait que dans la pratique l'utilisation d'une méthode de modélisation pour un système physique permettra ensuite la conversion vers un modèle équivalent si nécessaire pour l'analyse ou pour la synthèse d'une loi de commande. On reprend dans cette section les

descriptions et l'équivalence telles qu'elles sont discutées en [43].

3.2.1 Système PWA (Piecewise Affine Systems)

D'une façon générale un système discret affine par morceaux peut être défini comme suit :

$$x_{k+1} = A_i x_k + a_i \quad (3.1)$$

tel que : $x_k \in \mathcal{R}_i \subset \mathbb{R}^n$ et $n \in \mathbb{N}^+$ sont respectivement l'état du système et son ordre. \mathcal{R}_i définit les différentes régions correspondant à chaque dynamique dont la partie linéaire est décrite par la matrice $A_i \in \mathbb{R}^{n \times n}$ et la partie affine par le vecteur $a_i \in \mathbb{R}^{n \times 1}$.

Si on restreint notre attention à des régions n'ayant pas d'intersection entre elles, définies par des hyperplans dans l'espace d'état $x \in \mathbb{R}^n$, leur représentation est :

$$\mathcal{R}_i = \{x \in \mathbb{R}^n | H_i x \leq h_i\} \quad (3.2)$$

tel que $H_i \in \mathbb{R}^{m \times n}$ et $h_i \in \mathbb{R}^{m \times 1}$ définissent des contraintes linéaires pour chaque région. Cette représentation par hyperplans peut être traduite dans une description duale des générateurs (sommets de la région polyédrale). Le système est défini dans une région de l'espace d'état tant que $x_k \in \mathbb{X} = \bigcup_{i=1}^l \mathcal{R}_i$. Il est à noter que même si les régions \mathcal{R}_i sont convexes, leur union \mathbb{X} ne l'est pas nécessairement.

On note que dans cette description aucune hypothèse n'est faite à propos de la continuité de la dynamique à la frontière entre les régions.

3.2.2 Systèmes MLD (Mixed Logical Dynamical systems)

Comme leur nom l'indique, les modèles MLD intègrent dans leur représentation des variables logiques, des dynamiques et des contraintes de fonctionnement. Cette structure a été proposée par [37, 38]. La description d'un modèle MLD s'écrit :

$$x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k, \quad (3.3)$$

$$y_k = Cx_k + D_1 u_k + D_2 \delta_k + D_3 z_k, \quad (3.4)$$

et pour les contraintes :

$$E_1 x_k + E_2 u_k + E_3 \delta_k + E_4 z_k \leq g, \quad (3.5)$$

tel que, $x_k = [x_r^T \ x_b^T]^T$ avec $x_r \in \mathbb{R}^{n_r}$ et $x_b \in \{0, 1\}^{n_b}$, y_k et u_k ont la même structure logique/réel que x_k . $z_k \in \mathbb{R}^{r_c}$ et $\delta_k \in \{0, 1\}^{r_b}$ représente des variables auxiliaires.

3.2.3 Systèmes à complémentarité linéaire LC

Parmi les travaux qui ont étudié ce genre de système, on cite [41, 42]. La description d'un modèle LC se résume par :

$$x_{k+1} = Ax_k + B_1 u_k + B_2 w_k, \quad (3.6)$$

$$y_k = Cx_k + D_1 u_k + D_2 w_k, \quad (3.7)$$

$$v_k = E_1 x_k + E_2 u_k + E_3 w_k + g, \quad (3.8)$$

$$0 \leq v_k \perp w_k \geq 0 \quad (3.9)$$

tel que, $v_k, w_k \in \mathbb{R}^s$. L'équation (3.9) exprime le fait que les vecteurs v_k et w_k sont perpendiculaires, $v_k^T w_k = 0$ d'où la notion de complémentarité.

3.2.4 Systèmes à complémentarité linéaire étendu ELC

Une autre classe de systèmes hybrides qui a connu une certaine reconnaissance dans la littérature est celle qui peut être modélisée par la représentation étendue ELC [43]. Le modèle mathématique ELC s'écrit sous la forme suivante :

$$x_{k+1} = Ax_k + B_1 u_k + B_2 d_k, \quad (3.10)$$

$$y_k = Cx_k + D_1 u_k + D_2 d_k, \quad (3.11)$$

$$E_1 x_k + E_2 u_k + E_3 d_k \leq g, \quad (3.12)$$

$$\sum_{i=1}^p \prod_{j \in \Phi_j} (g - E_1 x_k - E_2 u_k - E_3 d_k)_j = 0, \quad (3.13)$$

3.2.5 Systèmes MMPS (*min-max plus scaling*)

Une expression MMPS de la fonction f de variables x_1, \dots, x_n , est définie par la grammaire :

$$f := x_i |\alpha| \max(f_k, f_l) | \min(f_k, f_l) | f_k + f_l | \beta f_k \quad (3.14)$$

avec $i \in \mathbb{N}^*$, $\alpha, \beta \in \mathbb{R}$, et f_k, f_l , sont des expressions MMPS.

Un système MMPS est décrit par :

$$\begin{cases} x_{k+1} = \mathcal{M}_x(x_k, u_k, d_k), \\ y_k = \mathcal{M}_y(x_k, u_k, d_k), \\ \mathcal{M}_c(x_k, u_k, d_k) \leq c, \end{cases} \quad (3.15)$$

où $\mathcal{M}_x, \mathcal{M}_y$ et \mathcal{M}_c , sont des expressions MMPS en fonction de l'état x_k , de la commande u_k et d'une variable auxiliaire d_k .

3.2.6 Équivalence entre les différents modèles

Il a été montré dans [43] qu'il y a, sous certaines conditions, des équivalences entre les différents modèles de représentation de classes de systèmes hybrides, à savoir les MLD, LC, ELC et MMPS. La figure 3.1 montre les passages qui peuvent se faire entre ces différents modèles.¹ Par la suite dans le mémoire, la représentation qui sera utilisée est celle des systèmes affines par morceaux PWA mais on retiendra que les méthodes développées sont transférables par le biais de ces équivalences à d'autres représentations des systèmes hybrides.

1. Il existe des boîtes à outils [33] qui permettent de générer d'une façon automatique le modèle PWA à partir d'un modèle MLD par exemple.

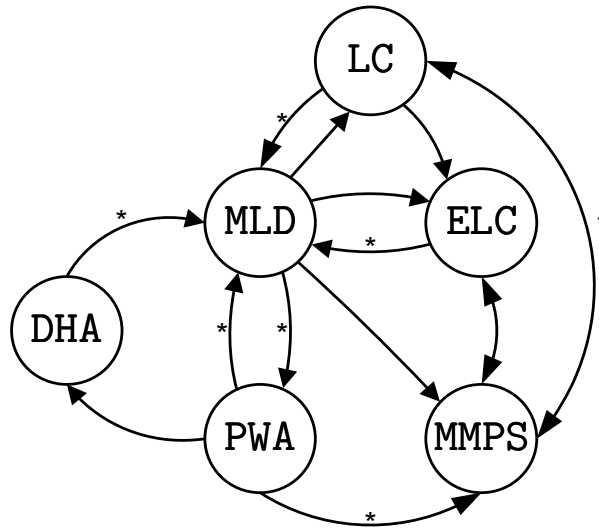


FIGURE 3.1 – Les différents liens entre les différentes classes de systèmes hybrides. Les liens avec étoile expriment l’existence de conditions [43] pour établir le lien.

3.3 Exemples illustratifs

Avant de continuer avec l’analyse des systèmes PWA, on présente quelques exemples de systèmes, dont leur modèle peut être décrit par un système affine par morceaux. Le premier exemple présente un système PWA par nature, le deuxième système est un convertisseur électronique DC-DC qui peut être modélisé par un modèle hybride, le troisième est un système électrique avec saturation (non linéarité) et le dernier un système linéaire en boucle fermée avec un correcteur prédictif explicite.

3.3.1 Système affine par morceaux : véhicule avec plusieurs dynamiques

L’exemple d’une voiture qui se déplace horizontalement (axe x) sur une voie à plusieurs pentes [33], comme illustré sur la figure 3.2, peut être modélisé par plusieurs dynamiques à savoir sa position sur les différentes partitions. Le modèle dynamique de déplacement de la voiture s’écrit d’une façon générale en continu :

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u - \begin{bmatrix} 0 \\ g \sin \alpha \end{bmatrix} \quad (3.16)$$

La vitesse du véhicule ne dépend que de la force appliquée u et de la pente de la voie α . Donc on peut modéliser le comportement du véhicule par quatre dynamiques suivant sa position à savoir :

- premier tronçon : $x \leq x_1 \Rightarrow \alpha = \alpha_1$
- deuxième tronçon : $x_1 \leq x \leq x_2 \Rightarrow \alpha = \alpha_2 = 0$
- troisième tronçon : $x_2 \leq x \leq x_3 \Rightarrow \alpha = \alpha_3$
- quatrième tronçon : $x \geq x_3 \Rightarrow \alpha = \alpha_4 = 0$

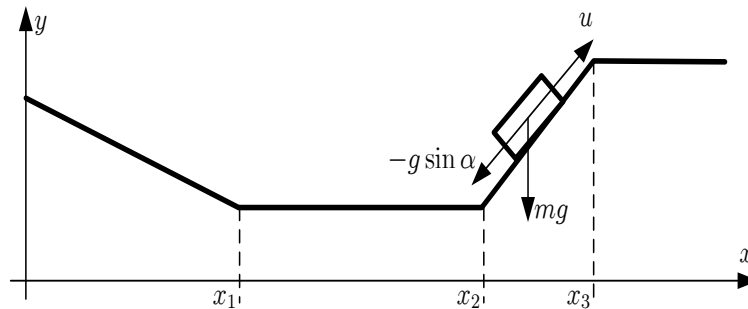


FIGURE 3.2 – Véhicule se déplaçant sur plusieurs pentes.

En partant de cette description on retrouve quatre dynamiques linéaires affectées par des termes constants, leur validité étant donnée par des inégalités linéaires. On dispose donc d'un système PWA. En utilisant des techniques classiques de type Euler ou alternatives on peut retrouver un modèle correspondant PWA en temps discret.

3.3.2 Système hybride : convertisseur DC-DC

Les convertisseurs DC-DC sont des circuits électroniques qui permettent de convertir une tension d'entrée continue en une tension de sortie continue par emmagasinage temporaire de l'énergie électrique à l'entrée, et la libération de cette énergie en sortie sous une tension différente. Ces convertisseurs utilisent des composants électroniques qui fonctionnent comme des interrupteurs commandés ou non par un signal exogène, donc le convertisseur peut avoir plusieurs configurations de fonctionnement selon l'ouverture ou la fermeture de ses interrupteurs. Ce genre de systèmes peut être modélisé par des modèles hybrides. On donne comme exemple une structure simplifiée d'un type de convertisseur de tension appelé **Buck** (abaisseur de tension), qui est schématisé par la figure 3.3.

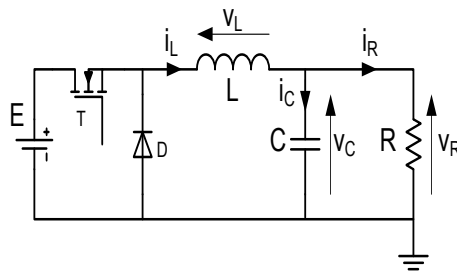


FIGURE 3.3 – Schéma électrique simplifié d'un abaisseur de tension.

Ce circuit est un système hybride parce que les différents modèles mathématiques qui régissent son comportement dépendent d'une part, d'une variable logique exogène (qui permet de bloquer ou de saturer le transistor), et aussi de certaines conditions liées à ses dynamiques pour saturer ou bloquer la diode. Dépendant de l'annulation ou non du courant de la bobine i_L entre deux cycles consécutifs de la saturation du transistor T, deux modes de fonctionnement du convertisseur sont à considérer à savoir :

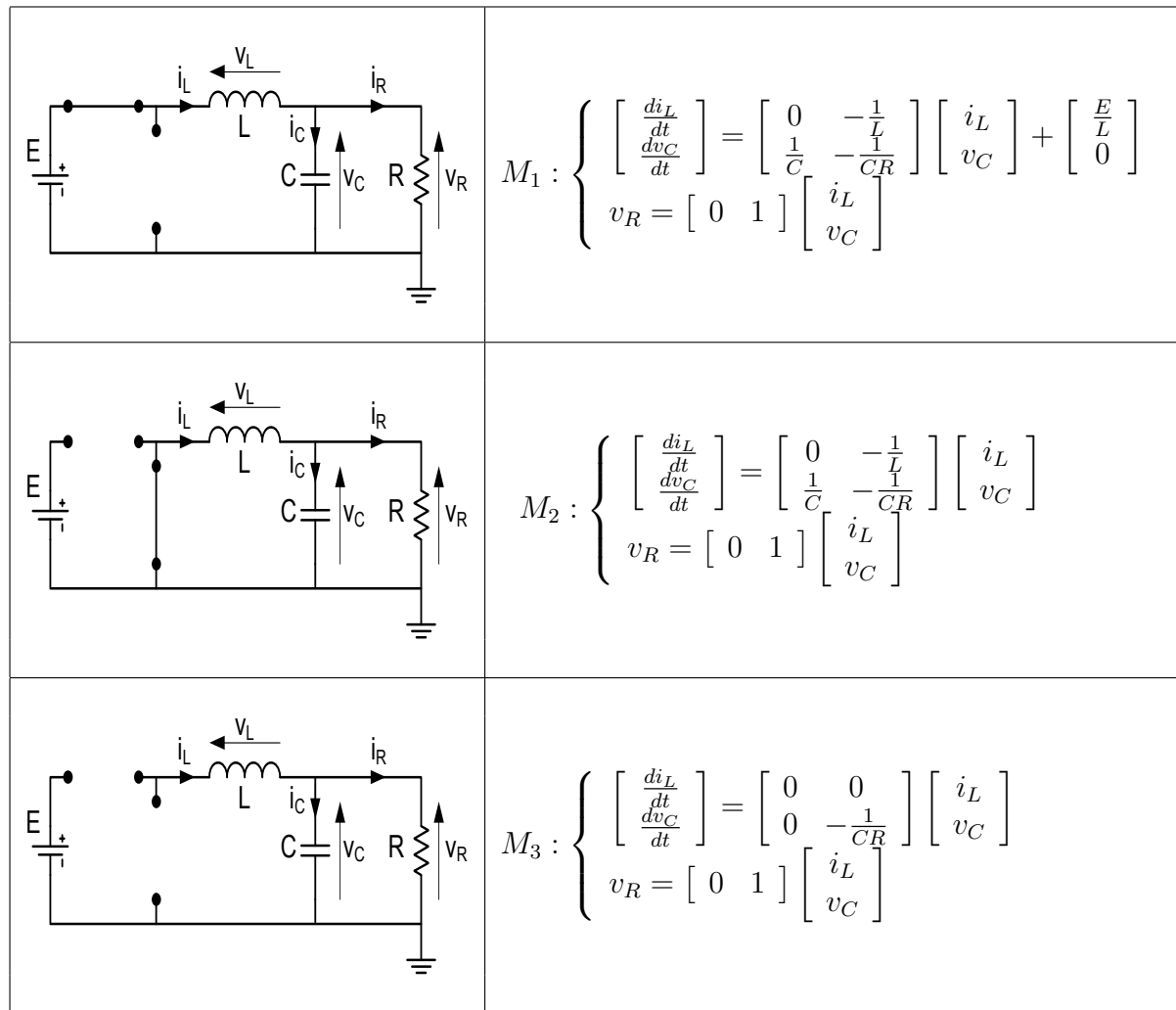


FIGURE 3.4 – Modèles correspondant aux différentes configurations des interrupteurs, à savoir l'ouverture ou la fermeture du transistor T et de la diode D.

- fonctionnement en CCM (Continuous Current Mode), le courant i_L ne s'annule jamais au moment de l'ouverture du transistor
- fonctionnement en DCM (Discontinuous Current Mode), le courant i_L s'annule un instant après l'ouverture du transistor.

La figure 3.4, montre les différentes configurations (modèles) possibles du système ainsi que les conditions de passage d'un modèle à un autre. La figure 3.5 illustre ces transitions.

3.3.3 Système hybride : balle bondissante

On considère une balle bondissante de masse m , lâchée d'une hauteur $x(t_0) = h$. Cette balle perd de l'énergie à chaque rebondissement, donc elle n'atteint jamais la hauteur initiale h .

Avant le toucher du sol, cette balle subit une force gravitationnelle mg , donc son accélération est :

$$\ddot{x}(t) = -g, \quad \text{pour } x(t) > 0 \quad (3.17)$$

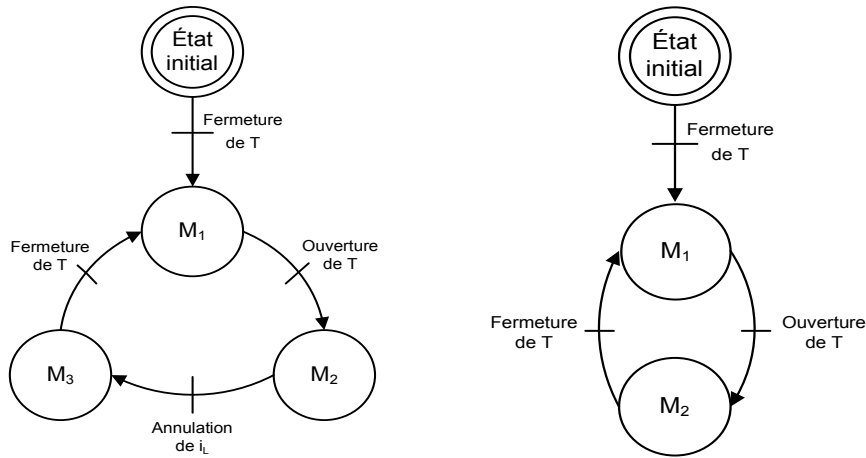


FIGURE 3.5 – Condition de passage d’un modèle à un autre. à gauche fonctionnement en DCM. à droite fonctionnement en CCM.

et quand elle touche le sol :

$$\dot{x}(t^+) = -(1 - \alpha)\dot{x}(t^-), \text{ pour } x(t) \leq 0 \tag{3.18}$$

avec α coefficient d’élasticité de la balle tel que pour $\alpha = 0$, la balle est élastique, et pour $\alpha = 1$ la balle est complètement inélastique. La figure 3.6 illustre les deux configurations de la balle, à savoir si elle touche ou non le sol. On discrétise le modèle de la balle pour une période d’échantillonnage T_s .

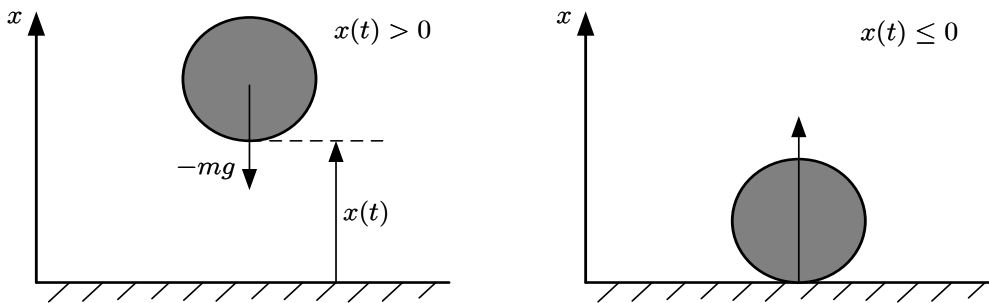


FIGURE 3.6 – Balle bondissante.

Pour $x(t) > 0$, la vitesse et l’accélération de la balle peuvent être approximées par :

$$\begin{cases} \dot{x}(t) \approx v_k = (x_k - x_{k-1})/T_s \\ -g = \ddot{x}(t) \approx (v_k - v_{k-1})/T_s \end{cases} \tag{3.19}$$

ce qui nous mène à l’équation dynamique suivante :

$$\begin{cases} v_{k+1} = v_k - T_s g \\ x_{k+1} = x_k + T_s v_k - T_s^2 g \end{cases} \tag{3.20}$$

Pour $x(t) \leq 0$, la vitesse change de direction dès que la balle touche le sol, donc l'équation dynamique qui régit le système est :

$$\begin{cases} v_{k+1} &= -(1 - \alpha)v_k \\ x_{k+1} &= x_k + T_s v_k \end{cases} \quad (3.21)$$

Ce système peut être modélisé comme un système hybride avec deux modèles dynamiques tel que le passage entre les deux modèles se fait par la vérification de la condition $x_k > 0$, voir la figure 3.7.

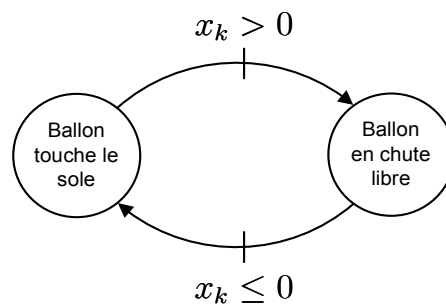


FIGURE 3.7 – Balle rebondissante : condition de basculement entre les deux modèles.

3.3.4 Systèmes LTI en boucle fermée avec une loi prédictive explicite

On a vu au chapitre 2 comment on peut obtenir un système PWA issu d'une commande prédictive par optimisation multiparamétrique.

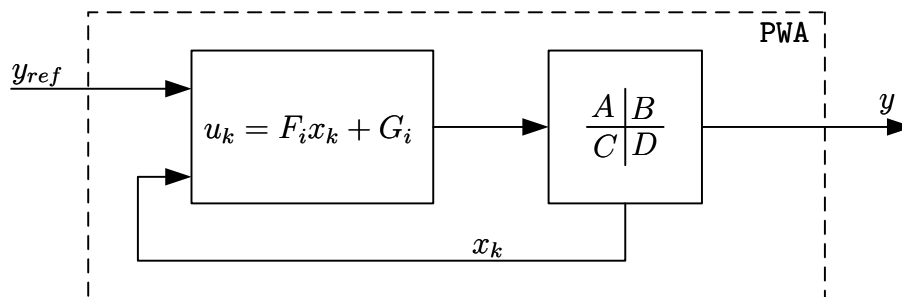


FIGURE 3.8 – Système PWA construit à partir d'un système LTI et sa loi explicite en boucle fermée issue d'une commande prédictive.

Il est à noter que pour un système LTI possédant un point d'équilibre à l'origine, une loi de commande explicite provenant d'une formulation de commande prédictive possède la caractéristique de continuité entre les régions du contrôleur ainsi qu'une région centrale qui contient l'origine.

3.4 Analyse des systèmes PWA

La complexité d'analyse pour les systèmes PWA réside dans le fait que plusieurs phénomènes complexes peuvent apparaître, parmi lesquels :

- la non convexité de leur domaine de définition global,
- des trajectoires complexes, dans le sens où elles ne peuvent pas être générées à partir de surfaces de niveaux d'une fonction analytique (quadratique par exemple),
- l'existence de discontinuités,
- le danger d'instabilité d'un point d'équilibre même si les dynamiques propres à chaque partition sont stables,
- la possibilité d'apparition de surfaces de glissement entre les partitions de l'espace d'état du système.

Trajectoires complexes

Les systèmes PWA peuvent avoir, en passant d'une région à une autre, des trajectoires complexes. Ce phénomène est dû aux différences entre les dynamiques ainsi qu'aux discontinuités qui peuvent exister entre les différentes partitions. L'exemple suivant illustre ce phénomène.

Exemple 5. Trajectoires complexes [44]. Considérons le système PWA (3.22-3.23) défini sur quatre régions, dont les dynamiques correspondantes sont linéaires et génèrent des trajectoires ellipsoïdales convergeant vers l'origine.

$$x_{k+1} = A_i x_k, \quad x_k \in \mathcal{R}_i \quad (3.22)$$

avec

$$A_1 = A_3 = \begin{bmatrix} 0.99 & 0.1 \\ -0.5 & 0.99 \end{bmatrix}, \quad A_2 = A_4 = \begin{bmatrix} 0.99 & 0.5 \\ -0.1 & 0.99 \end{bmatrix}, \quad (3.23)$$

La figure 3.9 montre les partitions et un exemple de trajectoire complexe pour laquelle on ne peut pas trouver de fonction de Lyapunov quadratique.

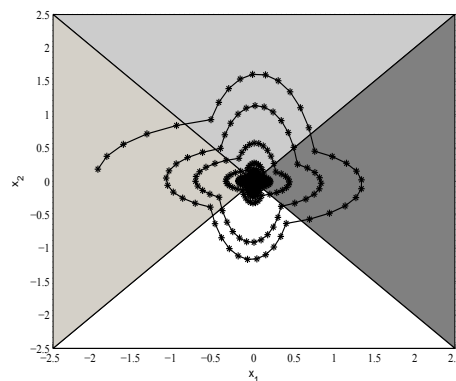


FIGURE 3.9 – Trajectoire non convexe pour le système (3.22) défini sur quatre régions.

Changement de partition qui rend le système instable

Malgré le fait que le système (3.22) est décrit par des dynamiques stables selon les critères appliquées aux systèmes LTI, le comportement global PWA devient instable si on change les partitions comme il est montré sur la figure 3.10.

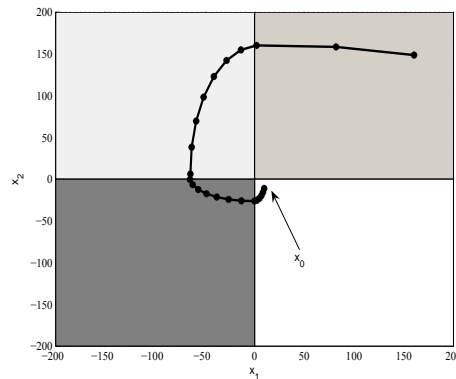


FIGURE 3.10 – Système (3.22) devenu instable en changeant ses partitions.

Remarque 1. Le partitionnement de l'espace d'état des systèmes PWA joue un rôle important dans la détermination de la stabilité ou l'instabilité du système. Les fonctions de Lyapunov quadratiques, si elles ne prennent pas en considération les informations structurelles sur les partitions des systèmes PWA, seront mal adaptées pour l'analyse de stabilité de ce genre de systèmes.

Système avec surface de glissement

Le phénomène de glissement pour les systèmes continus [44, 45] peut apparaître à la frontière des deux régions de la partition. Les trajectoires seront "piégées" entre deux dynamiques, une qui conduit le système à traverser la frontière vers une région voisine, alors que la dynamique correspondant à cette région le repousse vers la région d'origine. Ce phénomène apparaît aussi pour les systèmes discrets, l'exemple suivant montre un système PWA avec le phénomène de glissement.

Exemple 6. Phénomène de glissement.

Considérons le système PWA suivant :

$$x_{k+1} = A_i x_k, \quad x_k \in \mathcal{R}_i \quad (3.24)$$

avec

$$A_1 = A_3 = \begin{bmatrix} 0.98 & -0.02 \\ 0.04 & 1 \end{bmatrix}, \quad A_2 = A_4 = \begin{bmatrix} 0.98 & 0.02 \\ -0.04 & 1 \end{bmatrix}, \quad (3.25)$$

et la partition décrite dans la figure 3.11. La frontière entre les deux régions représente une surface de glissement, qui conduit dans ce cas à l'instabilité du système. Dans la figure 3.12 il est montré que le changement du comportement du système par l'apparition d'une surface de glissement stable peut être obtenu en changeant les partitions.

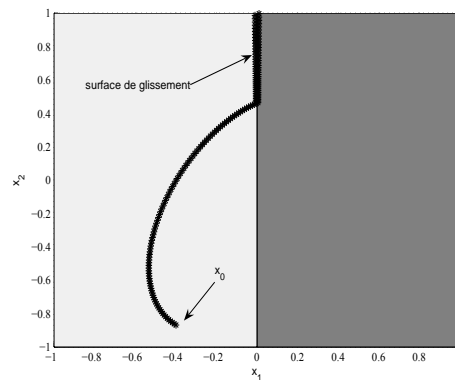


FIGURE 3.11 – Système instable avec phénomène de glissement qui capte la trajectoire du système entre deux régions.

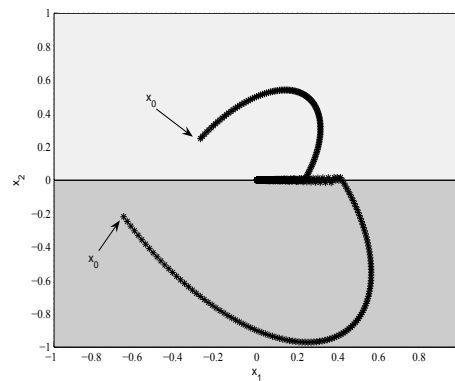


FIGURE 3.12 – Système stable avec phénomène de glissement qui capte la trajectoire du système entre deux régions.

Dans ce qui suit on rappelle quelques résultats utiles pour l'analyse de stabilité asymptotique des systèmes PWA.

3.5 Stabilité des systèmes LTI et PWA

L'analyse de la stabilité des systèmes dynamiques est une phase très importante pour la compréhension de son fonctionnement ainsi que pour la conception des lois de commande. Les méthodes d'analyse qui seront présentées ici sont des méthodes basées sur les fonctions de Lyapunov que ce soit pour les systèmes linéaires autonomes ou affines par morceaux. Dans une deuxième phase la construction de fonctions de Lyapunov quadratiques affines par morceaux mieux adaptées aux systèmes définis par morceaux sera rappelée. Du point de vue pratique, ces fonctions de Lyapunov seront calculées par la résolution de problèmes d'optimisation convexe.

3.5.1 Notions de stabilité pour systèmes dynamiques

Considérons un système dynamique autonome :

$$\dot{x}(t) = f(x) \quad (3.26)$$

avec f une application définie dans un domaine $D \in \mathbb{R}^n$. On considère $x_e = 0 \in D$ comme point d'équilibre du système qui satisfait $f(x_e) = 0$. On considère $x(t)$ comme étant la solution du système (3.26) qui démarre de l'état initial x_0 à l'instant t_0 . On rappelle quelques définitions mathématiques sur la notion de stabilité [46].

Stabilité au sens de Lyapunov

Le point d'équilibre x_e du système (3.26) est stable au sens de Lyapunov, si pour tout $t_0 \geq 0$ et $\epsilon > 0$, il existe $\delta(t_0, \epsilon)$ tel que :

$$\|x_0\| < \delta(t_0, \epsilon) \Rightarrow \|x(t)\| < \epsilon \quad \forall t \geq t_0. \quad (3.27)$$

Stabilité asymptotique

Le point d'équilibre x_e du système (3.26) est asymptotiquement stable si :

- x_e est un point d'équilibre stable,
- x_e est un point attractif, c'est-à-dire, pour tout $t_0 \geq 0$, il existe un $\delta(t_0)$ tel que :

$$\|x_0\| < \delta \Rightarrow \lim_{t \rightarrow \infty} \|x(t)\| = 0. \quad (3.28)$$

La stabilité asymptotique assure que le système tend vers le point d'équilibre x_e (état de repos) après une phase transitoire.

Stabilité exponentielle

Ce type de stabilité fait intervenir la notion de taux de convergence. On dit que le point d'équilibre stable x_e du système (3.26) est exponentiellement stable s'il existe $m, \alpha > 0$ tel que :

$$\|x(t)\| \leq m e^{-\alpha(t-t_0)} \|x_0\|, \quad (3.29)$$

pour tout $x_0 \in \mathbb{B}_h$, $t \geq t_0 \geq 0$. La constante α exprime le taux de convergence. La stabilité exponentielle assure que le système converge vers le point d'équilibre stable x_e avec une certaine vitesse de convergence.

3.5.2 Notions de stabilités pour systèmes LTI et avec incertitudes

Étant donné un système linéaire discret LTI :

$$x_{k+1} = Ax_k, \quad A \in \mathbb{X} \subseteq \mathbb{R}^{n \times n} \quad (3.30)$$

les concepts de stabilité pour systèmes discrets [47] sont largement connus.

Définition 5. Le point d'équilibre $x_e = 0$ du système (3.30) est stable au sens de Lyapunov si, pour tout $\epsilon > 0$, il existe $\delta(\epsilon) > 0$, tel que :

$$\|x_0\| < \delta(\epsilon) \Rightarrow \|x_k\| < \epsilon, \quad \forall k > 0. \quad (3.31)$$

Si en plus

$$\lim_{k \rightarrow +\infty} \|x_k\| = 0 \quad (3.32)$$

alors le point d'équilibre $x_e = 0$ est asymptotiquement stable.

Définition 6. Le système (3.30) est dit exponentiellement stable si toutes ses solutions approchent exponentiellement 0 (pour $k \rightarrow \infty$). Le système est dit stable si toutes ses solutions sont bornées. S'il existe des solutions non bornées, le système est dit instable et s'il existe des solutions approchant ∞ exponentiellement, l'instabilité est exponentielle.

Théorème 4. Pour la stabilité exponentielle de (3.30) il faut et il suffit que les valeurs propres de A soient strictement à l'intérieur du disque unité du plan complexe. Pour la stabilité il faut et il suffit que les valeurs propres de A soient à l'intérieur du disque et que les valeurs propres sur le cercle unité soient simple. L'instabilité est exponentielle si au moins une valeur propre se trouve à l'extérieur du disque unité.

Ensemble invariant ellipsoïdal pour un système LTI

Le domaine d'attraction pour un système linéaire LTI (3.30), ou pour une inclusion différentielle linéaire LDI introduite plus loin en (3.40), est défini par l'ensemble des valeurs initiales pour lesquelles les trajectoires convergent vers le point d'équilibre x_e .

Pour un système LTI (3.30), défini dans un domaine polytopique contenant l'origine :

$$x \in \mathbb{X} = \{x \in \mathbb{R}^n \mid h_k^T x \leq 1, \quad k = 1, \dots, p\} \quad (3.33)$$

on peut affirmer que l'origine est exponentiellement stable si et seulement si le problème d'optimisation convexe (3.34)[48] :

$$\begin{aligned} & \text{minimiser} && \log \det Q^{-1} \\ & \text{tel que} && Q \succ 0 \\ & && A^T Q A - Q \prec 0 \\ & && h_k^T Q h_k \leq 1 \quad k = 1, \dots, p \end{aligned} \quad (3.34)$$

est faisable (il existe une solution). En plus, le plus grand domaine d'attraction ellipsoïdal $\mathbb{D} \subseteq \mathbb{X}$ est défini par :

$$\mathbb{D} = \{x \in \mathbb{R}^n \mid x^T Q^{-1} x \leq 1\}. \quad (3.35)$$

Les vecteurs h_k avec $k = 1, \dots, p$, définissent des contraintes linéaires pour lesquelles le domaine \mathbb{X} est défini.

Exemple 7. On considère le système autonome suivant :

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x, \quad (3.36)$$

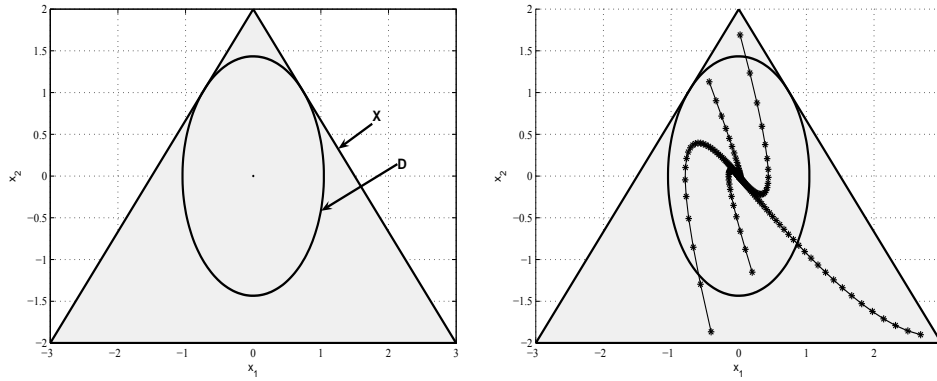


FIGURE 3.13 – (gauche) Région \mathbb{X} pour laquelle le système (3.37) est défini, avec le plus grand domaine estimé par fonction de Lyapunov quadratique ; (droite) trajectoire du système pour quelques points initiaux.

On discrétise ce système à une période d'échantillonnage $T_s = 1s$:

$$x_{k+1} = \begin{bmatrix} 0.60 & 0.232 \\ -0.465 & -0.097 \end{bmatrix} x_k, \quad (3.37)$$

On définit la région de l'espace d'état $\mathbb{X} \subset \mathbb{R}^2$ tel que :

$$\mathbb{X} = \left\{ x \in \mathbb{R}^2 \mid \begin{bmatrix} 0.6667 & 0.5 \\ 0 & -0.5 \\ -0.6667 & 0.5 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}, \quad (3.38)$$

En résolvant le problème d'optimisation convexe (3.34) en utilisant la toolbox *YALMIP* [49], on trouve la plus grande ellipse contenue en \mathbb{X} (figure 3.13), qui peut être décrite comme un domaine d'attraction pour l'origine. Cette ellipse est caractérisée par la matrice symétrique $Q = Q^T \succ 0$:

$$Q = \begin{bmatrix} 1.125 & 0 \\ 0 & 2 \end{bmatrix} \quad (3.39)$$

Remarque 2. Par ces contraintes structurelles, la fonction de Lyapunov quadratique ne peut conduire qu'à une partie de l'ensemble invariant maximal dans l'espace \mathbb{X} , ce qui explique l'existence de trajectoires qui convergent vers l'origine et qui ne sont pas captées par l'ellipse. On verra dans le chapitre suivant comment on peut calculer ou approximer l'ensemble invariant maximal pour ces systèmes définis dans une région restreinte de l'espace d'état.

Inclusion différentielle linéaire LDI

Une inclusion différentielle linéaire et polytopique LDI [48] est définie comme :

$$x_{k+1} = A_{\Delta} x_k, \quad A_{\Delta} \in \mathbb{R}^{n \times n} \quad (3.40)$$

avec $A_\Delta \in \mathbf{Co}\{A_1, A_2, \dots, A_L\}$, où la notation $\mathbf{Co}\{\cdot\}$ désigne l'enveloppe convexe. Pour garantir la stabilité quadratique d'une inclusion différentielle linéaire polytopique, en suivant les principes du cas LTI, il suffit de trouver une fonction de Lyapunov $V(x) = x^T P x$, tel que :

$$\begin{cases} P = P^T \succ 0, \\ A_i^T P A_i - P \prec 0, \quad i = 1, \dots, L. \end{cases} \quad (3.41)$$

3.5.3 Notions de stabilités pour les systèmes PWA

Dans ce qui suit on rappelle [50] différents résultats sur l'analyse de stabilité pour les systèmes discrets affines par morceaux :

$$x_{k+1} = A_i x_k + a_i, \quad \text{si } x_k \in \mathcal{R}_i \quad (3.42)$$

avec $\mathcal{R}_i \subset \mathbb{X} \subset \mathbb{R}^n$. L'union $\bigcup_{i=1}^s \mathcal{R}_i$ est une partition polyédrale de \mathbb{X} contenant l'origine et définissant l'espace de validité du système (3.42), avec :

$$\mathcal{R}_i \cap \mathcal{R}_j = \emptyset, \quad \forall i \neq j,$$

Les a_i sont des vecteurs constants. Les régions \mathcal{R}_i sont définies comme :

$$\mathcal{R}_i = \{x \in \mathbb{R}^n \mid H_i x \leq h_i\} \quad (3.43)$$

On appelle $\mathcal{I} = \{1, \dots, s\}$ l'ensemble des indices des partitions dans l'espace d'état. On considère $x_e = 0$ comme point d'équilibre du système, donc les partitions qui contiennent l'origine ont leur partie affine $a_i = 0$ ($f(x_e) = 0$). On considère x_k comme étant la trajectoire du système (3.42) qui démarre de l'état initial x_0 à l'instant $k = 0$.

Fonction de Lyapunov quadratique

Un premier pas dans l'étude de stabilité du point d'équilibre $x_e = 0$ est de considérer la fonction de Lyapunov quadratique suivante :

$$V(x_k) = x_k^T P x_k, \quad (3.44)$$

avec $P = P^T \in \mathbb{R}^n$, une matrice symétrique définie positive $P \succ 0$. Si on peut trouver une telle matrice P avec :

$$A_i^T P A_i - P \prec 0, \quad \forall i \in \mathcal{I} \quad (3.45)$$

alors (3.44) est une fonction de Lyapunov quadratique commune à toutes les dynamiques $\{A_1, \dots, A_s\}$, et le système (3.42) est asymptotiquement stable. Il est à noter que trouver une fonction de Lyapunov quadratique (3.44), et satisfaire (3.45), reste une condition suffisante pour la stabilité asymptotique du système discret PWA (3.42). En même temps elle peut s'avérer très conservative car elle réduit le système (3.42) à une inclusion différentielle linéaire de type (3.40).

Pourquoi une fonction de Lyapunov définie par morceaux ?

Les exemples précédents nous conduisent à la conclusion que l'analyse de stabilité par fonctions de Lyapunov quadratiques a des limitations à savoir :

- elle ne considère aucune information structurelle concernant les partitions de l'espace d'état d'un système PWA,
- les dynamiques n'ont pas de partie affine (puisqu'on considère des dynamiques de type $x_{k+1} = A_i x_k$, et non pas de type $x_{k+1} = A_i x_k + a_i$), et des transformations sont nécessaires pour traiter le cas général,
- Les systèmes PWA peuvent avoir des trajectoires non convexes, ce qui entraîne l'impossibilité de trouver une fonction de Lyapunov quadratique commune.

Ces limitations ont poussé plusieurs chercheurs à travailler sur la construction de fonctions de Lyapunov quadratiques par morceaux ou linéaires par morceaux [44, 50].

Fonction de Lyapunov quadratique par morceaux

La fonction de Lyapunov quadratique par morceaux peut être calculée en considérant une matrice symétrique définie positive $P_i \succ 0$ pour chaque partition \mathcal{R}_i définissant le système (3.42). On définit cette fonction de Lyapunov comme :

$$V(x_k) = x_k^T P_i x_k, \quad \forall x \in \mathcal{R}_i \quad (3.46)$$

Pour garantir la stabilité il suffit que la variation de $V(x)$ soit négative pour toute trajectoire dans \mathbb{X} :

$$\Delta V(x_k, x_{k-1}) = V(x_k) - V(x_{k-1}) < 0 \quad (3.47)$$

donc si on suppose que $x_k \in \mathcal{X}_i$ et $x_{k-1} \in \mathcal{X}_j$, on obtient les LMI à satisfaire :

$$P_i \succ 0, \quad \forall i \in \mathcal{I} \quad (3.48)$$

$$A_j^T P_i A_j - P_j \prec 0, \quad \forall (j, i) \in \mathcal{I}_{all} \quad (3.49)$$

où $\mathcal{I}_{all} = \mathcal{I} \times \mathcal{I}$ est l'ensemble de toutes les combinaisons possibles de passage d'une région à une autre. Il est à noter qu'on n'est pas obligé de prendre toutes les possibilités théoriques de passage d'une région à une autre, mais seulement les passages ou transitions possibles entre les régions ; l'étude de l'atteignabilité du système permet de calculer $\mathcal{S} \subseteq \mathcal{I}_{all}$ tel que :

$$\mathcal{S} = \{(j, i) : j, i \in \mathcal{I} \text{ et } \exists k \in \mathbb{N}, \text{ tel que } x_k \in \mathcal{R}_i \text{ et } x_{k-1} \in \mathcal{R}_j\} \quad (3.50)$$

Théorème 5. L'origine du système PWA (3.42) est asymptotiquement stable pour toute condition initiale dans \mathbb{X} , s'il existe s matrices P_i telles que les LMI suivantes soient satisfaites :

$$P_i \succ 0, \quad \forall i \in \mathcal{I} \quad (3.51)$$

$$A_j^T P_i A_j - P_j \prec 0, \quad \forall (j, i) \in \mathcal{S} \quad (3.52)$$

On relève les remarques suivantes :

- si le système PWA possède des trajectoires au sein d'une même région, autrement dit, \mathcal{S} possède des paires (i, i) , alors, pour garantir la stabilité asymptotique il faut que les dynamiques A_i de ces régions soient stables, ce qui est exigé par les LMI (3.51, 3.52),
- si \mathcal{S} possède des paires $\{(i, j), (j, i)\}$, alors, pour que le système PWA soit asymptotiquement stable, il faut satisfaire les LMI (3.51, 3.52), avec comme conséquence que toutes les dynamiques définies par $A_i A_j$, $A_j A_i$ sont stables (leurs valeurs propres sont à l'intérieur du cercle unité),
- si le système PWA possède des trajectoires cycliques, c'est-à-dire qu'on démarre d'une région i_1 , on parcourt i_2, i_3, \dots , et on revient à i_1 , alors, si la fonction de Lyapunov (3.46) existe avec satisfaction de (3.51, 3.52), alors toutes les matrices \tilde{A} doivent être stables (valeurs propres à l'intérieur du cercle unité) avec :

$$\tilde{A} = \cup_{l=2}^{\infty} \{A : A = \prod_{v=1}^l A_{i_v}, (i_1, \dots, i_l) \in \mathcal{S}_l\} \quad (3.53)$$

avec

$$\mathcal{S}_l = \{(i_1, \dots, i_l) \in \mathcal{I}^l : \{(i_1, i_2), \dots, (i_{l-1}, i_l), (i_l, i_1)\} \subset \mathcal{S} \text{ avec } i_j \text{ tous différents}\} \quad (3.54)$$

donc \mathcal{S}_l regroupe toutes les trajectoires cycliques de longueur l , et \tilde{A} regroupe toutes les multiplications des dynamiques correspondant à tous les groupement de trajectoires dans \mathcal{S}_l .

Fonction de Lyapunov quadratique par morceaux avec relaxation

Pour relaxer le conservatisme des LMI (3.51, 3.52) lié à l'utilisation d'une fonction de Lyapunov quadratique par morceaux, une autre méthode a été développée en considérant les informations liées aux structures des partitions [50]. On réécrit la définition d'une région \mathcal{R}_i (3.2) comme :

$$\mathcal{R}_i = \left\{ x \in \mathbb{R}^n \mid \bar{H}_i \begin{bmatrix} x \\ 1 \end{bmatrix} \geq 0 \right\} \quad (3.55)$$

avec $\bar{H}_i = [H_i \quad -h_i]$. Si on peut trouver des matrices f_i et $g_{i,j}$ telles que :

$$\begin{cases} x^T f_i x \geq 0 & \text{si } x \in \mathcal{R}_i \\ x^T f_i x \leq 0 & \text{si } x \notin \mathcal{R}_i \\ x^T g_{i,j} x \geq 0 & \text{si } x \in \mathcal{R}_i \cup \mathcal{R}_j \\ x^T g_{i,j} x \leq 0 & \text{si } x \notin \mathcal{R}_i \cup \mathcal{R}_j \end{cases} \quad (3.56)$$

alors la nouvelle condition de stabilité asymptotique relaxée du système PWA (3.42) s'écrit :

$$P_i - f_i \succ 0, \quad \forall i \in \mathcal{I} \quad (3.57)$$

$$A_j^T P_i A_j - P_j + g_{i,j} \prec 0, \quad \forall (j, i) \in \mathcal{S} \quad (3.58)$$

La satisfaction des LMI (3.57,3.58) implique la satisfaction des LMI (3.51, 3.52).

Les choix de f_i et $g_{i,j}$ se font comme suit [50] :

$$f_i = \bar{H}_i^T U_i \bar{H}_i \quad (3.59)$$

et

$$g_{i,j} = \begin{cases} \bar{H}_i^T W_i \bar{H}_i & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \quad (3.60)$$

avec U_i et W_i des matrices d'éléments non négatifs.

Lemme 2. Si les LMI suivantes :

$$P_i - \bar{H}_i^T U_i \bar{H}_i \succ 0, \quad \forall i \in \mathcal{I} \quad (3.61)$$

$$A_j^T P_i A_j - P_j \prec 0, \quad \forall (j, i) \in \mathcal{S}, \quad i \neq j \quad (3.62)$$

$$A_j^T P_i A_j - P_j + \bar{H}_i^T W_i \bar{H}_i \prec 0, \quad \forall (i, i) \in \mathcal{S} \quad (3.63)$$

admettent une solution en $P_i, P_j \succ 0, U_i \geq 0$ et $W_i \geq 0$, alors l'origine du système PWA (3.42) est asymptotiquement stable en \mathbb{X} .

3.6 Conception de la commande pour système PWA

Étant donné le système PWA suivant :

$$x_{k+1} = A_i x_k + a_i + B_i u_k \quad (3.64)$$

pour $x_k \in \mathcal{R}_i \subset \mathbb{R}^n$ avec \mathcal{R}_i des polyèdres convexes et $u_k \in \mathbb{R}^m$ le vecteur de commande. Les matrices A_i, a_i, B_i sont constantes de dimensions appropriées.

L'objectif est de calculer une loi de commande affine par morceaux qui stabilise l'origine (considérée comme point d'équilibre) pour le système PWA (3.64). Si la loi de commande est de type :

$$u_k = K_i x_k, \quad \forall x_k \in \mathcal{R}_i \quad (3.65)$$

alors le système (3.64) en boucle fermée s'écrit :

$$x_{k+1} = (A_i + B_i K_i) x_k + a_i, \quad \forall x_k \in \mathcal{R}_i \quad (3.66)$$

Dans ce qui suit, on se base sur la construction de la fonction de Lyapunov quadratique ou quadratique affine par morceaux pour calculer la commande (3.65).

3.6.1 À partir d'une fonction de Lyapunov quadratique

L'origine est un point d'équilibre stable pour le système (3.66) si on peut trouver une matrice symétrique $P \succ 0$ telle que :

$$P \succ 0 \quad (3.67)$$

$$(A_i + B_i K_i)^T P (A_i + B_i K_i) - P \prec 0, \quad \forall i \in \mathcal{I} \quad (3.68)$$

En posant $Q = P^{-1}$ et en utilisant le complément de Schur qui stipule que :

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \succ 0 \Leftrightarrow \begin{cases} R \succ 0 \\ Q - S R^{-1} S^T \succ 0 \end{cases} \quad (3.69)$$

ou encore de façon équivalente :

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \succ 0 \Leftrightarrow \begin{cases} Q \succ 0 \\ R - S^T Q^{-1} S \succ 0 \end{cases} \quad (3.70)$$

on peut réécrire les conditions de stabilité (3.67, 3.68) comme :

$$Q \succ 0 \quad (3.71)$$

$$Q - (A_i + B_i K_i) Q (A_i + B_i K_i)^T \succ 0, \quad \forall i \in \mathcal{I} \quad (3.72)$$

donc on écrit les LMI à résoudre (Q et Y_i inconnues) pour garantir la stabilité asymptotique du système (3.66) comme suit :

$$Q \succ 0 \quad (3.73)$$

$$\begin{pmatrix} Q & (A_i Q + B_i Y_i) \\ (A_i Q + B_i Y_i)^T & Q \end{pmatrix} \succ 0 \quad (3.74)$$

avec $K_i = Y_i Q^{-1}$ les gains de retour d'état de commande (pour une loi de commande $u_i = K_i x_k$ avec $x_k \in \mathcal{R}_i$) définis pour chaque région dans l'espace d'état qui font converger le système (3.66) vers l'origine.

On peut définir $\mathbb{X}_0 \subset \mathbb{X}$ comme étant la projection sur \mathbb{X} de la plus grande surface de niveau $\mathbb{X}_0(\epsilon) \subset \mathbb{X}$ définie par $\mathbb{X}_0(\epsilon) = \{x : x^T P x \leq \epsilon\}$, avec $\epsilon > 0$. Toute trajectoire démarrant à l'intérieur de \mathbb{X}_0 convergera vers l'origine sans sortir de \mathbb{X}_0 .

3.6.2 À partir d'une fonction de Lyapunov quadratique par morceaux

Suivant le même raisonnement utilisé dans la section 3.6.1 en utilisant des fonctions de Lyapunov affines par morceaux et en utilisant le complément de Schur, on obtient les LMI à résoudre (en fonction de Q_i et W_j) :

$$Q_i = P_i^{-1} \succ 0 \quad \forall i \in \mathcal{I} \quad (3.75)$$

$$\begin{pmatrix} Q_i & (A_j Q_j + B_j W_j) \\ (A_j Q_j + B_j W_j)^T & Q_j \end{pmatrix} \succ 0 \quad \forall (j, i) \in \mathcal{S} \quad (3.76)$$

avec $W_j = K_j P_j^{-1}$.

On peut définir $\mathbb{X}_0 \subset \mathbb{X}$ comme étant la projection sur \mathbb{X} de la plus grande surface de niveau $\mathbb{X}_0(\epsilon) \subset \mathbb{X}$ définie par $\mathbb{X}_0(\epsilon) = \{x : V(x) \leq \epsilon\}$, avec $\epsilon > 0$ et $V(x) = x^T P_i x$, pour $x \in \mathcal{R}_i$.

3.7 Conclusion

Ce chapitre a été consacré aux systèmes linéaires affines par morceaux PWA avec leur définition et les équivalences par rapport à certaines classes de systèmes hybrides. Quelques éléments liés à la complexité de l'étude de ces systèmes ont été rappelés, comme par exemple l'influence de la topologie de la partition de son domaine de définition. La stabilité des systèmes PWA a été évoquée en insistant sur le fait que les fonctions de Lyapunov quadratiques peuvent ne pas

être bien adaptées à ce genre de système. Finalement le problème de commande pour système PWA a été introduit en utilisant des notions d'optimisation convexe et de commande optimale à horizon glissant.

Le chapitre suivant sera consacré au calcul des ensembles invariants pour les systèmes PWA, problématique qui joue un rôle important pour garantir la stabilité d'un système dynamique dans certains domaines de l'espace d'état. Ainsi on présentera des algorithmes numériques pour leur calcul en gardant à l'esprit qu'ils sont applicables pour les systèmes PWA en général même si notre objectif déclaré reste la caractérisation du domaine de faisabilité d'une politique de commande prédictive pour des systèmes linéaires en présence de contraintes.

Chapitre 4

Calcul des ensembles invariants

4.1 Introduction

Le chapitre 2 a introduit les définitions des ensembles invariants et les principes généraux de la commande prédictive. Dans ce chapitre basé sur les articles [51, 52, 53, 54, 55], seront développées trois stratégies de construction des ensembles invariants pour un système PWA défini sur une union de régions polytopiques dans l'espace d'état. Les approches seront basées sur :

- la construction contractive, utilisant la dynamique directe,
- la construction expansive, exploitant la dynamique en temps inverse,
- l'utilisation des graphes de transitions entre les régions.

Considérons le système linéaire PWA :

$$x_{k+1} = f_{\text{PWA}}(x_k) = A_i x_k + a_i \quad i = 1, \dots, n_r \quad (4.1)$$

défini dans un ensemble compact de l'espace d'état $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i \subset \mathbb{R}^n$ avec $A_i \in \mathbb{R}^{n \times n}$ et $a_i \in \mathbb{R}^{n \times 1}$. Ce système est défini sur n_r régions. À chaque région correspond une dynamique unique A_i et une partie affine a_i . Si le système PWA provient d'une synthèse MPC, la continuité de la dynamique entre les différentes régions est assurée, et pour la région centrale \mathcal{R}_1 qui contient l'origine, la partie affine correspondante est nulle $a_1 = 0$. Cela assure que l'origine est un point d'équilibre pour le système PWA¹. Les régions \mathcal{R}_i sont des polytopes (polyèdres bornés) dont la définition et la caractérisation sont définies en annexe 1.

4.2 Calcul et approximation des ensembles invariants

L'objectif qu'on s'impose ici est de déterminer le plus grand sous-ensemble de \mathbb{X} caractérisé par une propriété de positive invariance par rapport à la dynamique définie dans (4.1). Dans un cadre plus général il faut mentionner que cette propriété est liée à la stabilité des systèmes PWA.

1. Les méthodes détaillées par la suite dans ce chapitre se positionnent différemment par rapport à cette hypothèse. Les algorithmes contractifs et ceux basés sur les graphes de transitions sont indépendants de cette hypothèse, et leur utilisation peut être envisagées pour les systèmes PWA dans le cas le plus général. Quant aux algorithmes expansifs, cette hypothèse, offre des conditions suffisantes pour sa mise en œuvre.

En effet si le système PWA admet une fonction de Lyapunov, alors des ensembles invariants peuvent être décrits par les courbes de niveaux de la fonction de Lyapunov (incluses dans le domaine de définition \mathbb{X}). On note par rapport à une telle construction que la détermination du plus grand ensemble positif invariant reste un objectif difficile à atteindre par le fait que la fonction de Lyapunov n'est pas unique et que le choix doit tenir compte de la topologie des contraintes définies par \mathbb{X} .

En plus on note que la description des ensembles invariants à base de fonctions de Lyapunov exclut les trajectoires décrivant des cycles limites. Tous ces motifs nous ont poussé vers la construction des ensembles invariants à base de méthodes ensemblistes, notamment par itérations géométriques.

Ces classes de problèmes restent néanmoins très vastes sans une caractérisation précise de la famille des ensembles considérés. En partant de la description polyédrale de la partition PWA, ainsi que des avantages numériques du calcul polyédral, le présent chapitre sera concentré sur la classe des ensembles convexes et compacts polyédraux (bornés).

Au delà de ces avantages, les descriptions polyédrales vont relever quelques défis structurels. Par exemple il est évident qu'un ensemble ellipsoïdal ne pourrait être qu'approché par un ensemble polyédral (voir le problème classique 'squaring the circle'-[56]) en effet dans le cas général des constructions des ensembles invariants, ce problème sera primordial et nous conduira souvent à des approximations.

Deux contributions seront décrites par la suite en détail : un premier algorithme qui opère par contraction, et le deuxième par expansion. Ces algorithmes utilisent intensivement des opérations géométriques sur des polytopes.

4.2.1 Calcul de la borne supérieure du MPI

Considérons un système dynamique autonome et affine par morceaux, défini par l'équation (4.1). L'algorithme 1 calcule d'une façon itérative et par contraction, la borne supérieure

$$\Phi^+ \supseteq \Phi^{\text{MPI}}$$

de l'ensemble invariant maximal Φ^{MPI} du système PWA (4.1). On rappelle que la partition $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i$, et les dynamiques associées peuvent être obtenues à partir d'une formulation explicite [33] d'une loi de commande prédictive MPC.

Principe de calcul

À l'initialisation, $iter = 0$, on considère $\mathcal{X}^{iter} = \mathbb{X}$ et on parcourt toutes les régions de l'ensemble initial pour éliminer les régions ou sous-régions qui sortent de \mathcal{X}^{iter} en un seul pas, ce qui veut dire qu'on remplace chaque région $\mathcal{R}_i \subset \mathcal{X}^{iter}$, par $Poly = preImag(\mathcal{R}_i, \mathcal{X}^1)$ ². L'ensemble $Poly$ peut être un ensemble vide si $image(\mathcal{R}_i)$ sort complètement de \mathcal{X}^{iter} , donc $image(\mathcal{R}_i) \not\subset \mathcal{X}^{iter}$. Au fur et à mesure que des régions ou sous-régions sont éliminées, l'ensemble \mathcal{X}^{iter} est mis à jour. Le résultat obtenu (ensemble des régions polytopiques) est considéré comme ensemble initial et la procédure est répétée. Pour que l'on puisse comparer deux

2. Pour la définition des opérateurs $imag$ et $preImag$ voir l'annexe 1.

ensembles obtenus pour deux itérations successives, on utilise un autre ensemble intermédiaire $\mathcal{F} = \mathcal{X}^{iter}$ de telle sorte que les traitements ne se feront pas sur \mathcal{X}^{iter} mais sur \mathcal{F} .

La figure 4.1 illustre graphiquement le principe de calcul de l'algorithme 1.

Algorithme 1 Calcul de la borne supérieure Φ^+ de l'ensemble invariant Φ^{MPI} .

ENTRÉES: Système PWA (4.1) et le nombre d'itérations maximal $iterMax$

```

1:  $iter \leftarrow 0$ 
2:  $\mathcal{X}^{iter} \leftarrow \mathbb{X}$ 
3: tantque  $iter \leq iterMax$  faire
4:    $\mathcal{F} \leftarrow \mathcal{X}^{iter}$ 
5:    $L \leftarrow \text{cardinal de } \mathcal{F}$ 
6:    $i \leftarrow 1$ 
7:   tantque  $i \leq L$  faire
8:      $Poly \leftarrow \text{preImage}(\mathcal{F}(i), \mathcal{F})$ 
9:      $\mathcal{F} \leftarrow \{\mathcal{F}(1, \dots, i-1), \mathcal{M}, \mathcal{F}(i+1, \dots, L)\}$ 
10:     $m \leftarrow \text{cardinal de } Poly$ 
11:     $i \leftarrow i + m$ 
12:     $L \leftarrow L + m - 1$ 
13:   fin tantque
14:    $\mathcal{X}^{iter+1} \leftarrow \mathcal{F}$ 
15:   if  $\mathcal{X}^{iter+1} = \mathcal{X}^{iter}$  then
16:      $iter \leftarrow iterMax + 1$ 
17:   else
18:      $iter \leftarrow iter + 1$ 
19:   end if
20: fin tantque

```

SORTIES: $\Phi^+ \leftarrow \mathcal{X}^{iter+1}$.

Remarque 3. Si l'ensemble \mathcal{X}^{iter} ne change pas entre deux itérations successives (étape 15) (ce qui veut dire que $\text{image}(\mathcal{X}^{iter}) \subseteq \mathcal{X}^{iter}$), alors la construction s'arrête (étape 16), et le résultat obtenu est exactement l'ensemble invariant $\Phi^+ = \Phi^{\text{MPI}}$, sinon, il s'arrête quand il arrive au nombre maximal $iterMax$ des itérations.

Remarque 4. Quand le cardinal de \mathcal{X}^{iter} augmente, l'algorithme 1 devient très gourmand en temps de calcul. La section 4.3 présentera une technique basée sur l'analyse par intervalles et les graphes de transitions pour améliorer le temps de calcul.

Remarque 5. Le résultat de cette construction est un système PWA défini sur une sous-partition qui peut être non convexe. Si le système PWA est caractérisé par une dynamique continue et un point d'équilibre unique³, alors l'ensemble MPI est connecté même si il est non convexe.

3. Cas des systèmes en boucle fermée avec une loi MPC explicite

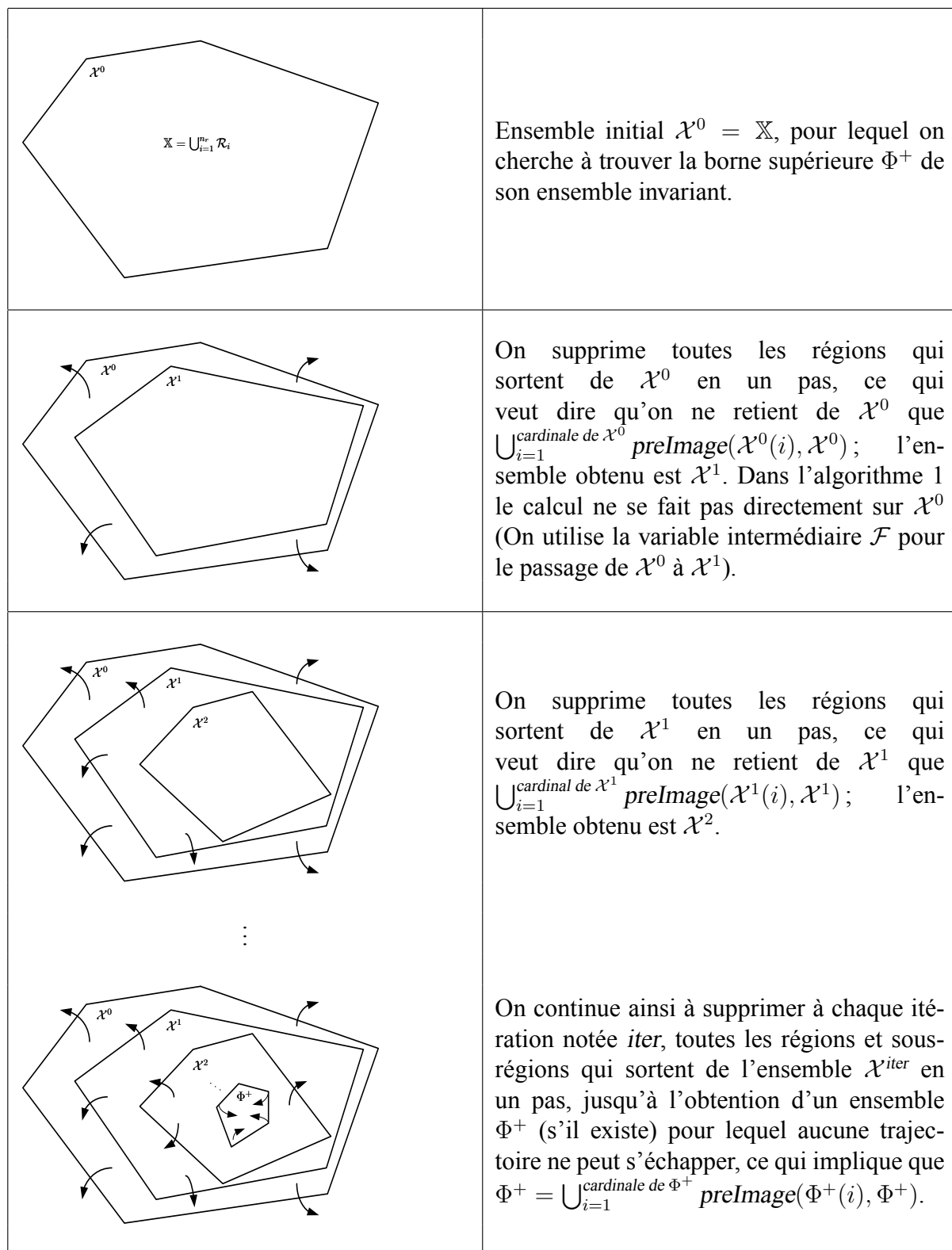


FIGURE 4.1 – Principe de calcul de l'ensemble invariant par contraction, en utilisant l'algorithme 1. Pour la clarté du dessin, on a évité de tracer les régions et les sous-régions.

Exemple 8. Ensemble invariant non convexe.

Considérons le système discret suivant :

$$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} x_k + \begin{bmatrix} 2 \\ 2 \end{bmatrix} u_k \quad (4.2)$$

avec les contraintes :

$$\begin{bmatrix} \mathbb{I}_2 \\ \mathbb{I}_2 \end{bmatrix} x_k \leq 0.1 \begin{bmatrix} \mathbb{1}_{2,1} \\ -\mathbb{1}_{2,1} \end{bmatrix}$$

On calcule une loi de commande prédictive pour le système (4.2), avec les horizons de prédiction et de contrôle $N = 2$, $N_u = 2$. Les matrices de pondérations sont $Q = \mathbb{I}_2$ et $R = 1$. On obtient une solution explicite [33] qui transforme le système (4.2) en un système PWA autonome. La figure 4.2 illustre graphiquement le calcul de l'ensemble invariant. L'algorithme 1 exécute 6 itérations pour trouver l'ensemble invariant exact.

Exemple 9. Ensemble invariant convexe.

Considérons le système discret suivant :

$$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 0.5 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.5 \\ 0.125 \end{bmatrix} u_k \quad (4.3)$$

avec les contraintes :

$$\begin{bmatrix} \mathbb{I}_2 \\ \mathbb{I}_2 \end{bmatrix} x_k \leq 5 \begin{bmatrix} \mathbb{1}_{2,1} \\ -\mathbb{1}_{2,1} \end{bmatrix}$$

On calcule une loi de commande prédictive pour le système (4.3), avec les horizons de prédiction et de contrôle $N = 2$, $N_u = 2$. Les matrices de pondérations sont $Q = \mathbb{I}_2$ et $R = 1$. On obtient une solution explicite et le système PWA associé. La figure 4.3 illustre graphiquement le calcul de l'ensemble invariant. L'algorithme 1 exécute 27 itérations pour trouver l'ensemble invariant exact.

4.2.2 Calcul de la borne inférieure du MPI

Le calcul par construction expansive opère d'une façon inverse (en itérant sur la dynamique inverse du système) par rapport à l'algorithme 1. Le résultat visé est la description d'une borne inférieure Φ^- de l'ensemble invariant Φ^{MPI} .

Principe de calcul

En partant de l'hypothèse que l'origine est placée à l'intérieur d'une région de la partition $\mathcal{R}_1 \subset \mathbb{X}$ et que la dynamique associée à cette région est stable⁴, en appliquant l'algorithme 1, on obtient un ensemble invariant⁵ initial \mathcal{X}^1 . Par la suite, l'idée est de calculer toutes les régions $\mathcal{X}^2 \subseteq \mathbb{X}$ qui transitent en un pas vers \mathcal{X}^1 . Alternativement $\mathcal{X}^{\text{iter}}$ est actualisé pour contenir

4. Cette hypothèse peut être relaxée à l'existence d'un point d'équilibre stable placé strictement à l'intérieur d'une région quelconque de la partition PWA

5. Qui est en même temps la région maximale admissible (*Maximal output admissible set*) [26].

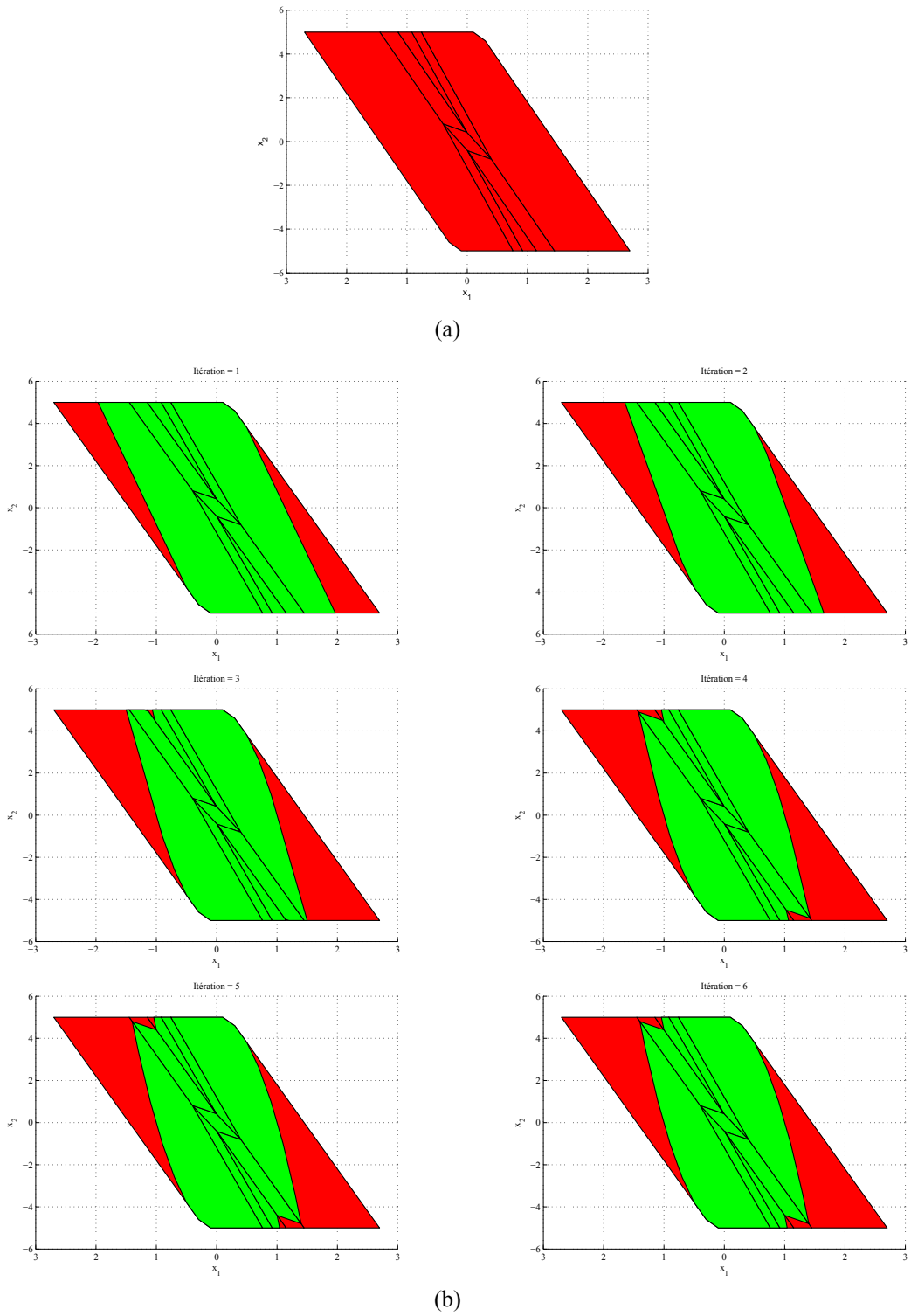


FIGURE 4.2 – Illustration graphique de la construction contractive de l'ensemble invariant de l'exemple 1.

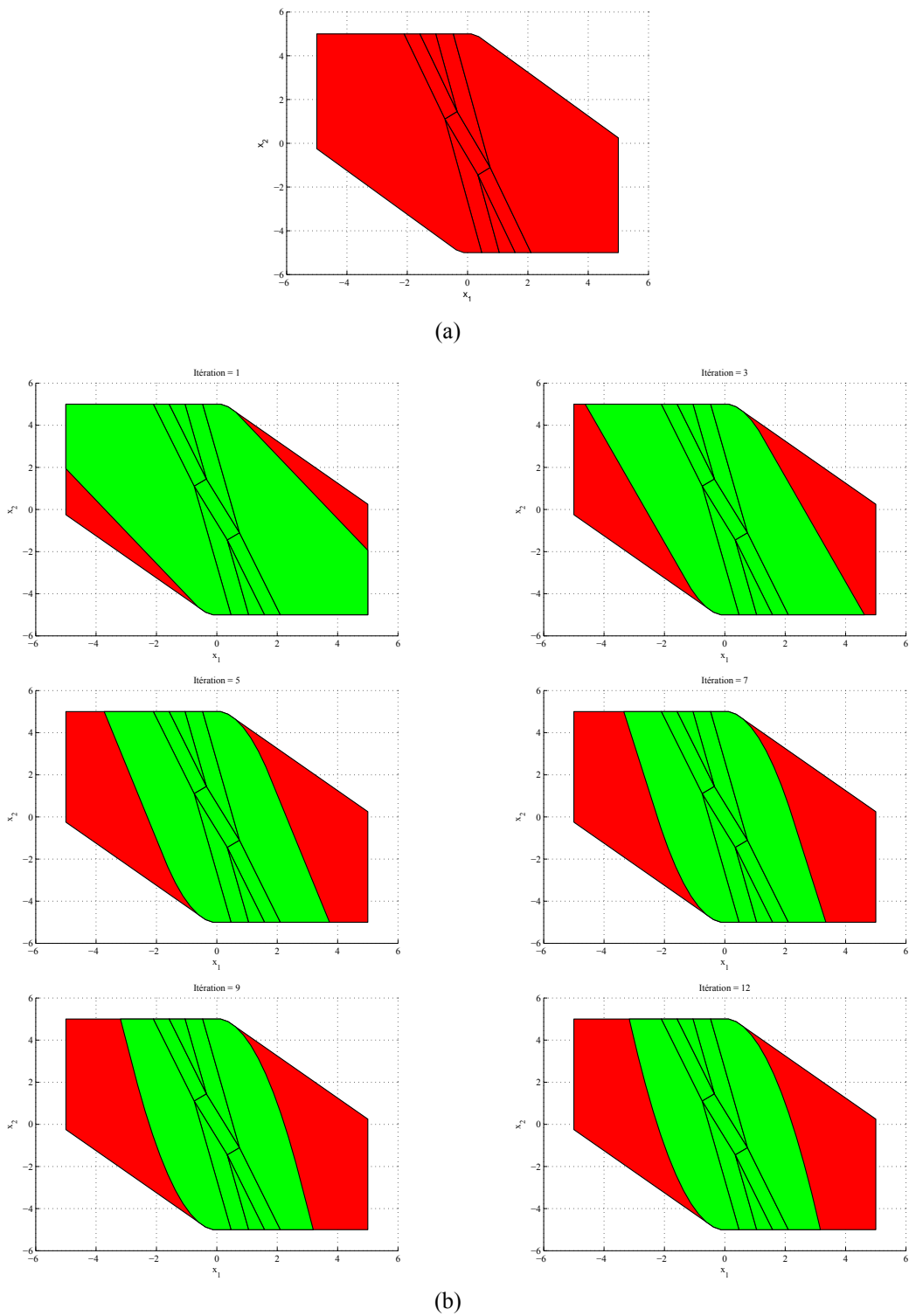


FIGURE 4.3 – Illustration graphique de la construction contractive de l'ensemble invariant de l'exemple 2.

tous les ensembles qui permettent d'atteindre l'ensemble invariant en un pas. Il peut être montré que \mathcal{X}^{iter} se dilate (car les itérations successives décrivent des ensembles avec relation d'inclusion) pour converger vers Φ^{MPI} compris dans ce cas comme le bassin d'attraction du point d'équilibre, et constitue ainsi une borne inférieure $\Phi^- \subseteq \Phi^{MPI}$ de l'ensemble invariant maximal.

La figure 4.4 illustre graphiquement le principe du calcul.

Algorithme 2 Calcul de la borne inférieure Φ^- de l'ensemble invariant Φ^{MPI} .

ENTRÉES: Le système PWA (4.1) et le nombre d'itérations maximale $iterMax$

```

1: Utiliser l'algorithme 1 pour calculer la région invariante de  $\mathcal{R}_1$  qui contient l'origine. Consi-
   dérante que le résultat est  $\mathcal{X}^1 \neq \emptyset$ 
2:  $\mathcal{S} \leftarrow \mathbb{X} \setminus \mathcal{X}^1$ 
3:  $\mathcal{Y}^1 \leftarrow \mathcal{X}^1$ 
4:  $L \leftarrow$  cardinal de  $\mathcal{S}$ 
5:  $iter \leftarrow 1$ 
6: tantque  $iter \leq iterMax$  faire
7:    $\mathcal{M} \leftarrow \emptyset$ 
8:    $i \leftarrow 1$ 
9:   tantque  $i \leq L$  faire
10:     $\mathcal{M} \leftarrow \mathcal{M} \cup preImage(\mathcal{S}(i), \mathcal{X}^{iter})$ 
11:     $i \leftarrow i + 1$ 
12:   fin tantque
13:   if  $\mathcal{M} = \emptyset$  then
14:      $iter \leftarrow iterMax + 1$ 
15:   else
16:      $\mathcal{X}^{iter+1} \leftarrow \mathcal{M}$ 
17:      $\mathcal{Y}^{iter+1} \leftarrow \mathcal{Y}^{iter} \cup \mathcal{M}$ 
18:      $iter \leftarrow iter + 1$ 
19:   end if
20: fin tantque

```

SORTIES: $\Phi^- \leftarrow \mathcal{Y}^{iter}$.

Remarque 6. À l'étape 1 de l'algorithme 2, \mathcal{X}^1 ne devrait pas être un ensemble vide, sinon, la borne inférieure et l'ensemble invariant sont $\Phi^- = \Phi^{MPI} = \emptyset$.

Si la dynamique n'est pas stable ou si le point d'équilibre stable se trouve sur la frontière d'une région de la partition PWA, alors une méthode alternative de construction de l'ensemble invariant initial $\mathcal{X}^1 \subset \Phi^{MPI}$ doit être proposée lors de l'initialisation de l'algorithme expansif. On note tout de même que dans le cas de la commande MPC pour les systèmes LTI, les hypothèses considérées précédemment sont vérifiées.

Remarque 7. On note que dans le cas des systèmes PWA avec continuité entre les régions, les procédures itératives décrites dans les deux algorithmes 1 et 2 impliquent des ensembles intermédiaires compacts. Cela veut dire qu'au lieu de traiter toutes les régions de la partition \mathbb{X} , seulement le voisinage de l'ensemble courant \mathcal{Y}^{iter} ou bien les régions de la frontière de \mathcal{X}^{iter}

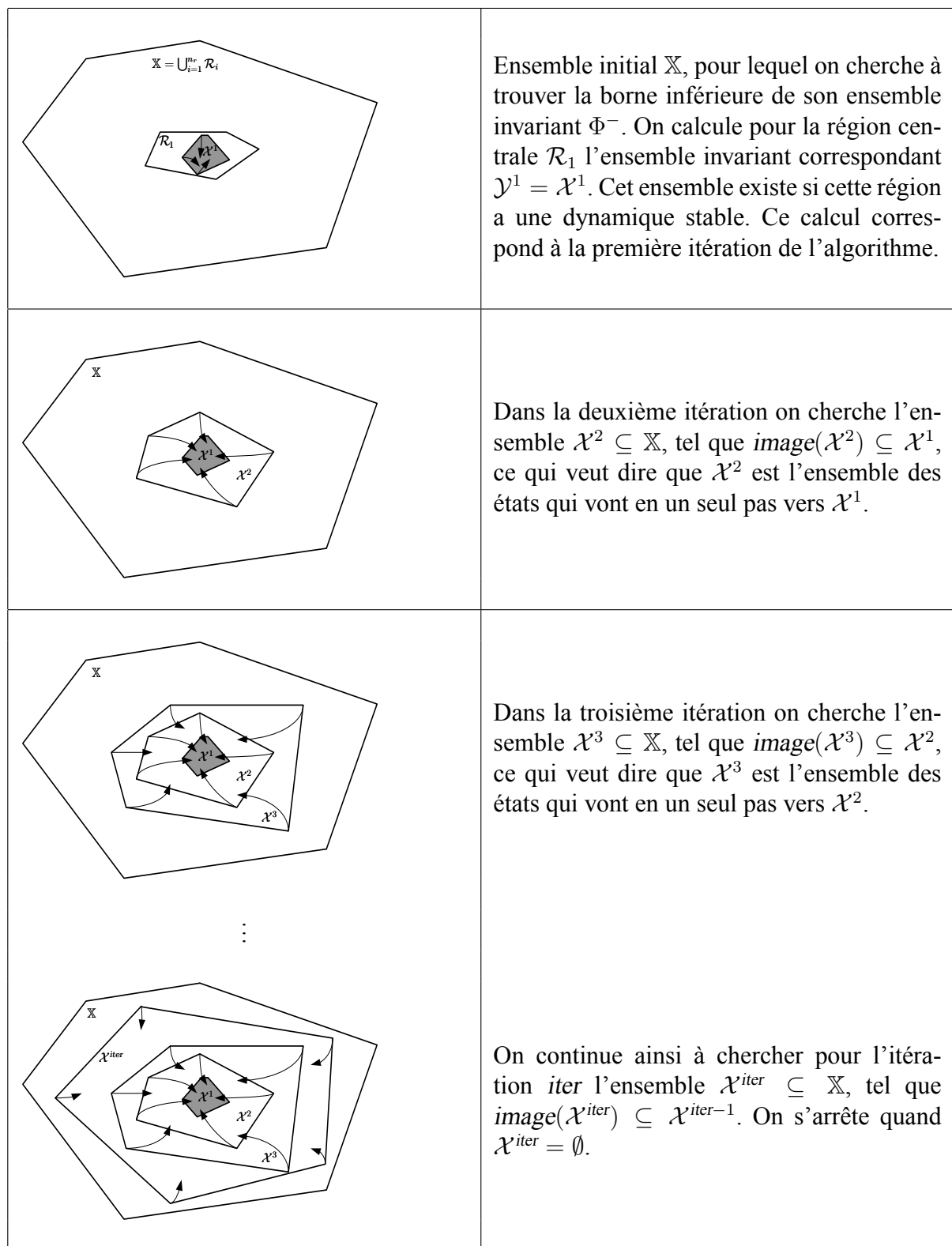


FIGURE 4.4 – Principe de calcul de l'ensemble invariant par expansion, en utilisant l'algorithme 2. Pour la clarté du dessin, on a évité de tracer les régions et les sous-régions.

sont à traiter. En plus de ça, si le nombre de régions à traiter diminue, l'effort de calcul par itération est aussi diminué. Il y a néanmoins le problème de décider quelles régions sont sur la frontière, une solution efficace utilisant l'analyse par intervalles et les graphes de transitions sera utilisée dans la section 4.3.

Remarque 8. Si $\mathcal{M} = \emptyset$ à l'étape 13 (ce qui veut dire qu'il n'y a aucun ensemble dans $\mathbb{X} \setminus \mathcal{Y}^{iter}$ qui peut atteindre \mathcal{X}^{iter-1}), alors l'algorithme 2 s'arrêtera, et le résultat est l'ensemble invariant maximal $\Phi^- = \Phi^{\text{MPI}}$. Sinon l'algorithme s'arrête après l'exécution du nombre maximal $iterMax$ des itérations.

Exemple 10. Ensemble invariant non convexe.

On considère la solution explicite correspondant au système (4.2). La figure 4.5 illustre quelques itérations pour le calcul de l'ensemble invariant utilisant la construction expansive par l'algorithme 2. Cet algorithme trouve l'ensemble invariant exact après 22 itérations. On observe une augmentation de la complexité de la partition \mathcal{X}^{iter} qui doit être par la suite simplifiée après rassemblement des différentes régions.

Exemple 11. Ensemble invariant convexe.

On considère la solution explicite correspondant au système (4.3). La figure 4.6, illustre quelques itérations pour le calcul de l'ensemble invariant utilisant la construction expansive par l'algorithme 2. L'ensemble invariant exact est construit en 25 itérations.

4.2.3 Phénomène de cycle limite

Pour un système PWA la présence de cycles limites⁶ va affecter différemment les constructions expansives/contractives introduites précédemment. L'algorithme 2 qui opère par construction expansive peut être bloqué par ce phénomène car leur convergence peut trouver comme limite la frontière du bassin d'attraction⁷ du cycle limite. En revanche l'algorithme 1 capte ces cycles limites car il procède par contraction et il traite exclusivement les régions extérieures à l'ensemble invariant maximal. L'exemple 12 illustre ce phénomène.

Exemple 12. On considère un système LTI discret, qui consiste à faire une rotation de $\frac{\pi}{2}$ dans le plan de phase :

$$x_{k+1} = \begin{bmatrix} 0 & -1.5 \\ 1.5 & 0 \end{bmatrix} x_k + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u_k \quad (4.4)$$

avec les contraintes :

$$\begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \\ 0.7071 & 0.7071 \\ 0.7071 & -0.7071 \end{bmatrix} x_k \leq 3.5355 \mathbf{1}_{4,1}, \quad \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \\ 0.7071 & 0.7071 \\ 0.7071 & -0.7071 \end{bmatrix} u_k \leq 0.3536 \mathbf{1}_{4,1},$$

6. Les cycles limites représentent des trajectoires fermées d'un système dynamique. Dans le cas des systèmes en temps continu, les cycles limites décrivent des courbes fermées dans le plan des phases [47]

7. C'est la région dans l'espace d'état pour laquelle les trajectoires convergent vers le point d'équilibre.

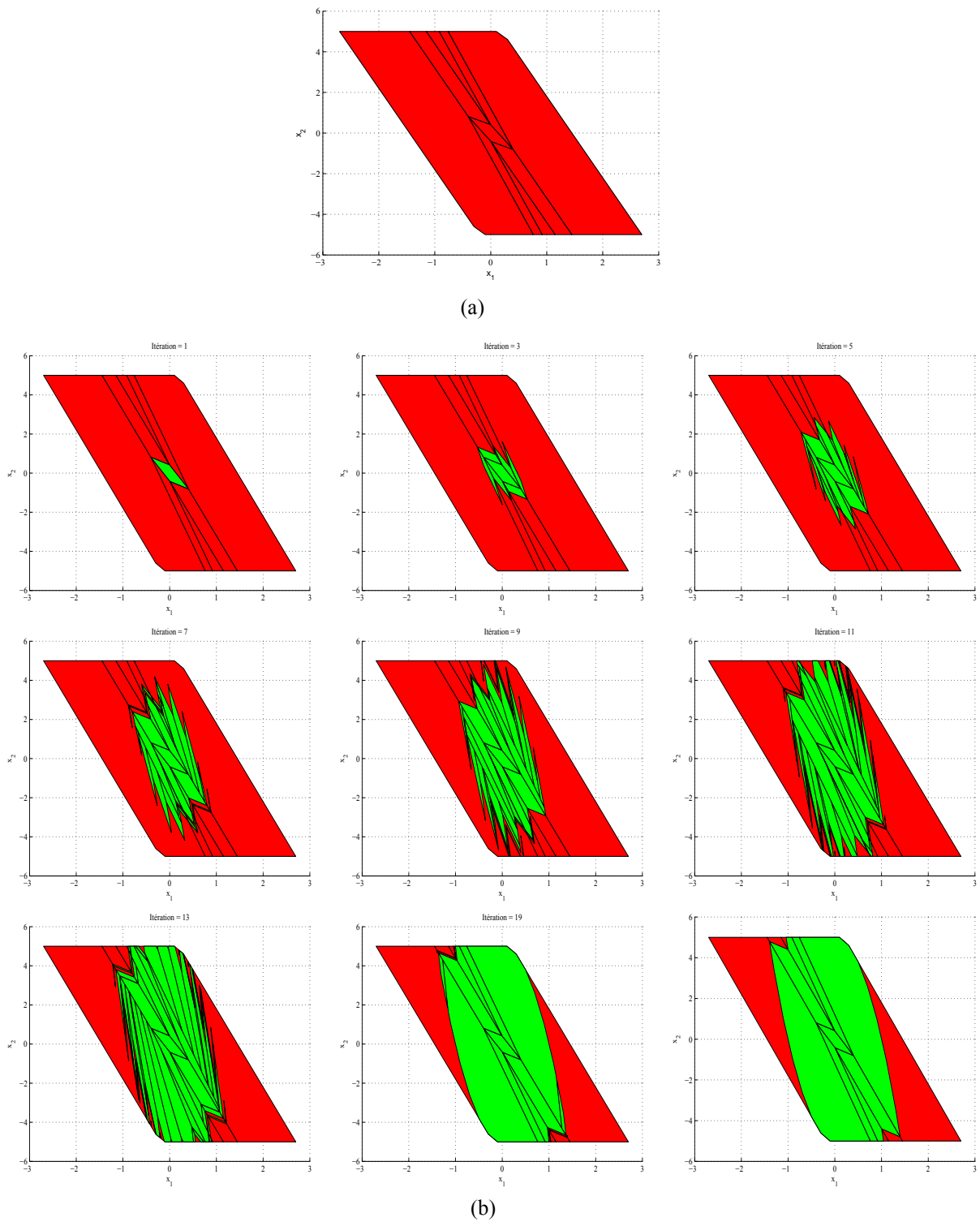


FIGURE 4.5 – Illustration graphique de la construction expansive de l'ensemble invariant de l'exemple 3.

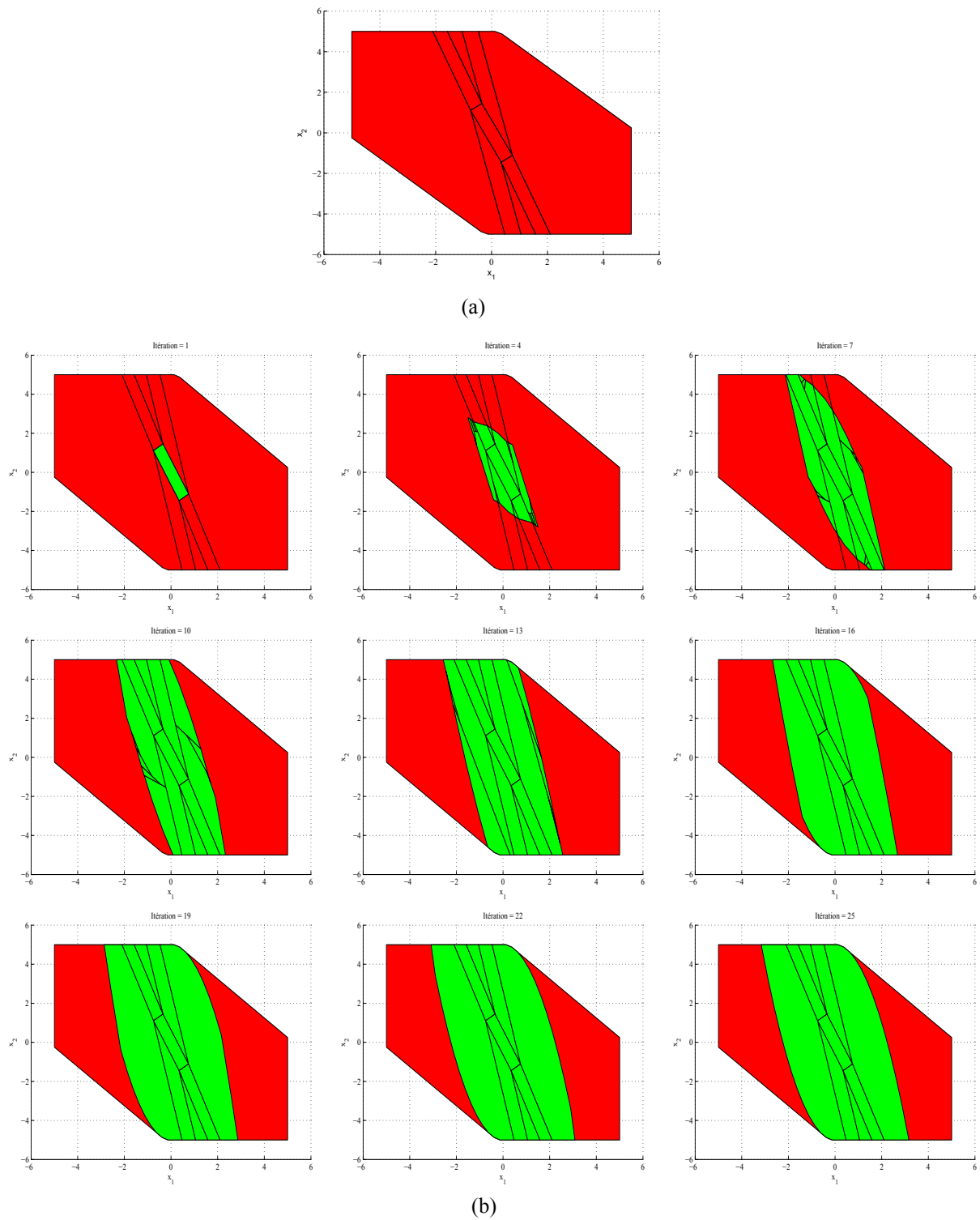


FIGURE 4.6 – Illustration graphique de la construction expansive de l'ensemble invariant de l'exemple 4.

On calcule une loi de commande prédictive pour le système (4.4), avec les horizons de prédiction et de contrôle $N = 3$, $N_u = 3$. Les matrices de pondération sont $Q = \mathbb{I}_2$ et $R = \begin{bmatrix} 1 & 0 \\ 0 & 1000 \end{bmatrix}$. On obtient une solution explicite et le système PWA associé. La figure 4.7 illustre un phénomène de cycle limite attractif.

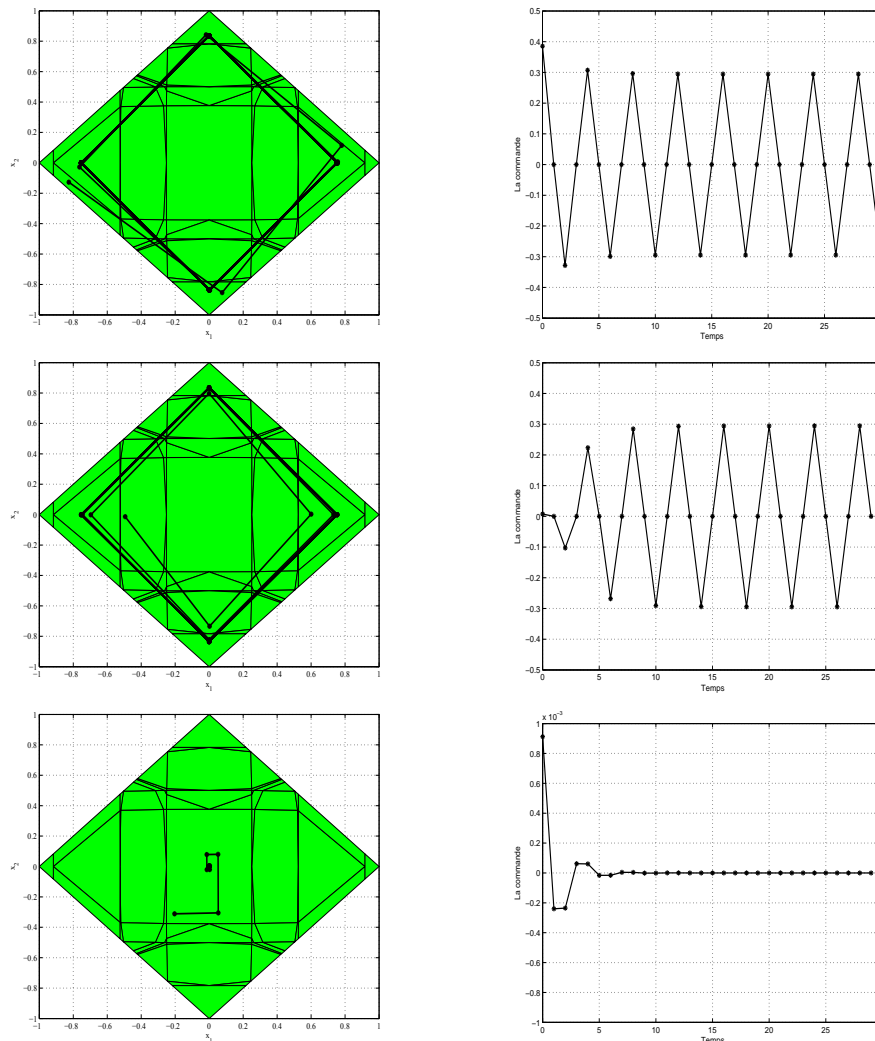


FIGURE 4.7 – Illustration graphique du phénomène de cycle limite pour certaines conditions initiales.

Le calcul de l'ensemble invariant par la construction contractive est fait en une seule itération, parce que la solution explicite est invariante, par contre, la construction expansive ne peut pas calculer complètement l'ensemble invariant à cause du phénomène de cycle limite. La figure 4.8 montre à quel niveau la construction expansive s'arrête en identifiant le bassin d'attraction de l'origine mais en ignorant l'invariance globale du système PWA.

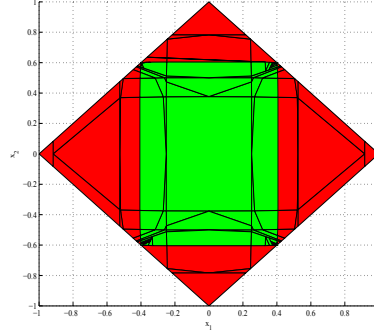


FIGURE 4.8 – La construction expansive (en vert) est bloquée par les cycles limites.

4.2.4 Convergence des constructions et le rapport avec l'ensemble MPI

Dans cette partie on introduit deux propositions sur la convergence de la construction de l'ensemble invariant par contraction (algorithme 1) et par expansion (algorithme 2).

On considère le système PWA (4.1) défini dans une partition de n_r régions. On suppose que \mathcal{R}_1 est la région centrale qui contient l'origine.

Proposition 1. En utilisant l'algorithme 2 à l'itération $iter$, la construction de l'ensemble invariant vérifie :

$$\mathcal{Y}^{iter} \subseteq \mathcal{Y}^{iter+1} \subseteq \Phi^{MPI} \quad (4.5)$$

avec Φ^{MPI} l'ensemble invariant maximal.

Démonstration. L'algorithme 2 est initialisé avec $\mathcal{X}^1 \subseteq \mathcal{R}_1$, l'ensemble invariant pour la région centrale \mathcal{R}_1 . On rappelle que pour \mathcal{R}_1 il n'y a pas de partie affine ($a_0 = 0$, parce que 0 est le point d'équilibre du système (4.1)), ainsi la théorie du *-maximal output admissible sets-* [26], peut être utilisée pour la construction de l'ensemble initial \mathcal{X}^1 . L'algorithme 1 est utilisé dans l'étape 1 de l'algorithme 2 pour calculer \mathcal{X}^1 dans un nombre d'itérations fini tant que toutes les valeurs propres $\lambda(A_i)$ sont à l'intérieur du cercle unité. Il est aussi évident que puisque $\mathcal{X}^1 \subseteq \Phi^{MPI}$ est l'ensemble invariant pour la région centrale, il est aussi ensemble invariant pour toute la partition \mathbb{X} du système PWA car $\Phi^{MPI} \in \mathbb{X}$. En partant de l'ensemble invariant initial $\mathcal{X}^1 \neq \emptyset$, on définit \mathcal{X}^2 comme l'ensemble des points qui transitent en un seul pas vers \mathcal{X}^1 , alors :

$$\mathcal{X}^2 = \bigcup_{i=1}^{n_r} \Psi_i \quad (4.6)$$

tel que :

$$\Psi_i = \left\{ x \in \mathcal{R}_i \mid \begin{array}{l} H_i x \leq h_i \\ H_1 A_i x \leq h_1 - H_1 a_i \end{array} \right\} \quad (4.7)$$

et tel que les $[H_i \ h_i]$ définissent les contraintes de \mathcal{R}_i . Sachant que \mathcal{X}^1 est un ensemble invariant, tous les points $x \in \mathcal{X}^1$ satisfont l'équation (4.6). On a donc l'inclusion $\mathcal{X}^1 \subseteq \mathcal{X}^2$ et la positive

invariance de \mathcal{X}^2 . Φ^{MPI} étant l'ensemble invariant maximal, pour que $\mathcal{X}^2 = \Phi^{\text{MPI}}$ il faut que $\bigcup_{i=1}^{n_r} \Psi_i = \emptyset$.

L'équation (4.5) est obtenue par déduction dans le cas général en suivant le même raisonnement. \square

Proposition 2. En utilisant l'algorithme 1 à l'itération $iter$, la construction de l'ensemble invariant vérifie :

$$\Phi^{\text{MPI}} \subseteq \mathcal{X}^{iter+1} \subseteq \mathcal{X}^{iter} \quad (4.8)$$

où Φ^{MPI} est l'ensemble invariant maximal.

Démonstration. L'algorithme 1 est initialisé en considérant simplement la partition $\mathcal{X}^1 = \mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i \subset \mathbb{R}^n$.

L'ensemble \mathcal{X}^2 est défini par la différence entre \mathcal{X}^1 et toutes les régions polytopiques dans \mathcal{X}^1 qui transitent en un pas à l'extérieur de lui-même, alors :

$$\mathcal{X}^2 = \mathcal{X}^1 \setminus \bigcup_{i=1}^{n_r} \Omega_i \quad (4.9)$$

tel que :

$$\Omega_i = \left\{ x \in \mathcal{R}_i \left| \begin{array}{l} H_i x \leq h_i; \\ \exists j \in \{1, \dots, n_r\}, \text{ tel que } \\ H_j A_i x \geq h_j - H_j a_i \end{array} \right. \right\} \quad (4.10)$$

Par la définition de soustraction des ensembles, l'équation (4.9) prouve l'inclusion $\mathcal{X}^2 \subseteq \mathcal{X}^1$. L'équivalence

$$\mathcal{X}^2 = \mathcal{X}^1 = \Phi^{\text{MPI}}$$

est garantie si et seulement si $\bigcup_{i=1}^{n_r} \Omega_i = \emptyset$. C'est le cas pour lequel la procédure de construction s'arrête avec le résultat exact. Le cas général de l'équation (4.8) est obtenue par déduction en suivant le même raisonnement. \square

On note que la construction expansive fournit à chaque itération un ensemble invariant alors que la construction par contraction fournit à chaque pas d'itération un ensemble qui n'est pas invariant mais qui contient toujours l'ensemble invariant maximal. On donne la proposition suivante qui fournit un degré d'approximation pour l'exécution en parallèle de ces deux procédures.

Proposition 3. Soit le système PWA (4.1) avec un point fixe stable à l'origine $\forall iter \in \mathbb{N}^*$, $\exists \epsilon_{iter}^{min} \in \mathbb{R}$, tel que $\mathcal{X}^{iter} \subset \mathcal{Y}^{iter} \oplus \mathbb{B}(\epsilon)$, $\forall \epsilon \geq \epsilon_{iter}^{min}$. De plus, ϵ_{iter}^{min} est décroissant pour $iter \rightarrow \infty$.

Démonstration. L'ensemble \mathcal{X}^1 est borné parce qu'on considère toujours que \mathbb{X} est borné. Alors il existe ϵ_1^{min} tel que $\mathcal{Y}^1 \subset \mathcal{X}^1 \oplus \mathbb{B}(\epsilon)$, $\forall \epsilon \geq \epsilon_1^{min}$. Alors utilisant les propositions 1 et 2, le comportement décroissant de ϵ_{iter}^{min} est directement obtenu. En plus, utilisant (4.5) et (4.8) on a $\epsilon \geq 0$ du moment que $\mathcal{Y}^{iter} \subseteq \Phi^{\text{MPI}} \subseteq \mathcal{X}^{iter}$. \square

Proposition 4. Soit le système PWA 4.1 avec un point fixe stable à l'origine et sans autre solutions périodiques (cycles limites) ou pseudo-périodiques, $\forall iter \in \mathbb{N}^*$, $\exists \epsilon_{iter}^{min} \in \mathbb{R}$, tel que $\mathcal{X}^{iter} \subset \mathcal{Y}^{iter} \oplus \mathbb{B}(\epsilon)$, $\forall \epsilon \geq \epsilon_{iter}^{min}$. De plus, $\epsilon_{iter}^{min} \rightarrow 0$ pour $iter \rightarrow \infty$.

Remarque 9. Les deux séquences d'ensembles Φ^- et Φ^+ convergent vers l'ensemble invariant maximal Φ^{MPI} tel que, $\Phi^- \subseteq \Phi^{\text{MPI}} \subseteq \Phi^+$, donc on peut utiliser les deux algorithmes 1 et 2 en parallèle, et considérer la distance de Hausdorff entre Φ^- et Φ^+ comme mesure de l'approximation. Celle ci peut être utilisée comme critère d'arrêt en plus du nombre maximal d'itérations *iterMax*.

Exemple 13. La distance de Hausdorff (voir annexe A) entre les ensembles obtenus par la construction contractive et expansive peut montrer la convergence des deux constructions, et nous donner une mesure de rapprochement et ainsi être utilisée comme critère d'arrêt. Reprenant les exemples 8 et 10, les figures 4.9 et 4.10 montrent les évolutions des différentes distances de Hausdorff entre les ensembles des constructions expansive et contractive.

4.3 Amélioration du temps de calcul

L'algorithme 1 (étapes 7 ~ 13), est basé sur le balayage de toutes les régions de \mathbb{X} pour calculer les $\text{preImage}(\mathcal{F}(i), \mathcal{F})$. Ces opérations nécessitent des calculs géométriques, et cette démarche peut s'avérer très gourmande en temps de calcul à cause de la multiplication des régions polyédrales. On note que dans certains cas une partie de ces calculs est inutile car le preImage est vide. Pour surmonter ce problème, et au lieu de calculer $\text{preImage}(\mathcal{F}(i), \mathcal{F})$ pour tout l'ensemble \mathbb{X} (étapes 7 ~ 13), l'analyse par intervalles va nous permettre de trouver les polytopes candidats pour lesquels on calcule $\text{preImage}(\mathcal{F}(i), \mathcal{F})$.

On propose dans cette section une méthode de diminution du temps de calcul des algorithmes 1 et 2 en utilisant l'analyse par intervalles et le graphe de transitions.

4.3.1 Analyse par intervalles

La recherche par arbre d'intervalles [57] est un algorithme qui permet d'identifier d'une façon efficace tous les intervalles, dans une structure prédéfinie, qui chevauchent un intervalle ou un point donné. Pour ce faire, deux aspects importants sont à traiter, à savoir, en premier lieu la structure des données et l'arbre des intervalles, et deuxièmement l'algorithme qui le parcourt.

Recherche par intervalles

Considérons l'ensemble polytopique :

$$\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i$$

Chaque région est définie par ses sommets tels que :

$$\text{sommets}(\mathcal{R}_i) = \{[s_{i,1} \ s_{i,2} \ \dots \ s_{i,m_i}] \mid s_{i,tt} = \begin{bmatrix} s_{i,tt}^1 \\ s_{i,tt}^2 \\ \vdots \\ s_{i,tt}^n \end{bmatrix} \in \mathbb{R}^{n \times 1}, \ tt = 1, \dots, m_i, \ m_i \in \{3, 4, \dots\}\}$$

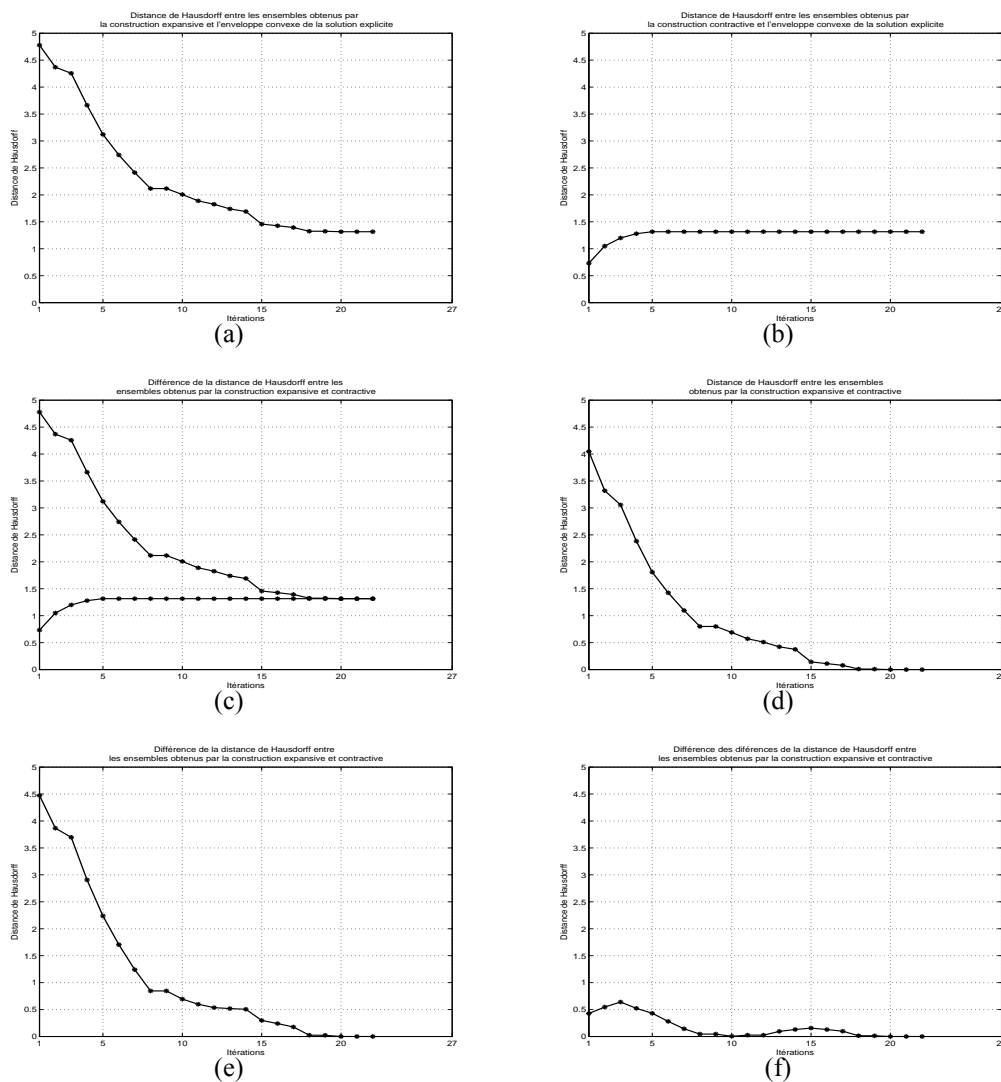


FIGURE 4.9 – Illustration graphique de la convergence entre les ensembles obtenus par les constructions expansive et contractive. (a) la distance de Hausdorff entre les ensembles de la construction expansive et la coque convexe de l'ensemble \mathbb{X} . (b) la distance de Hausdorff entre les ensembles de la construction contractive et la coque convexe de l'ensemble \mathbb{X} . (c) superposition de (a) et (b). (d) la différence entre les distances (a) et (b). (e) la distance de Hausdorff entre les ensembles des constructions expansive et contractive. (f) différence entre (d) et (e) explicable par l'application de la relation de triangle pour la distance de Hausdorff.

La projection des sommets de \mathcal{R}_i , $i = 1, \dots, n_r$, sur chaque dimension $j = \{1, \dots, n\}$ de l'espace \mathbb{R}^n (voir la figure 4.11 pour une illustration), est donnée par :

$$\text{projection}_j(\mathcal{R}_i) = [s_{i,1}^j \ s_{i,2}^j \ \dots \ s_{i,m_i}^j] \quad (4.11)$$

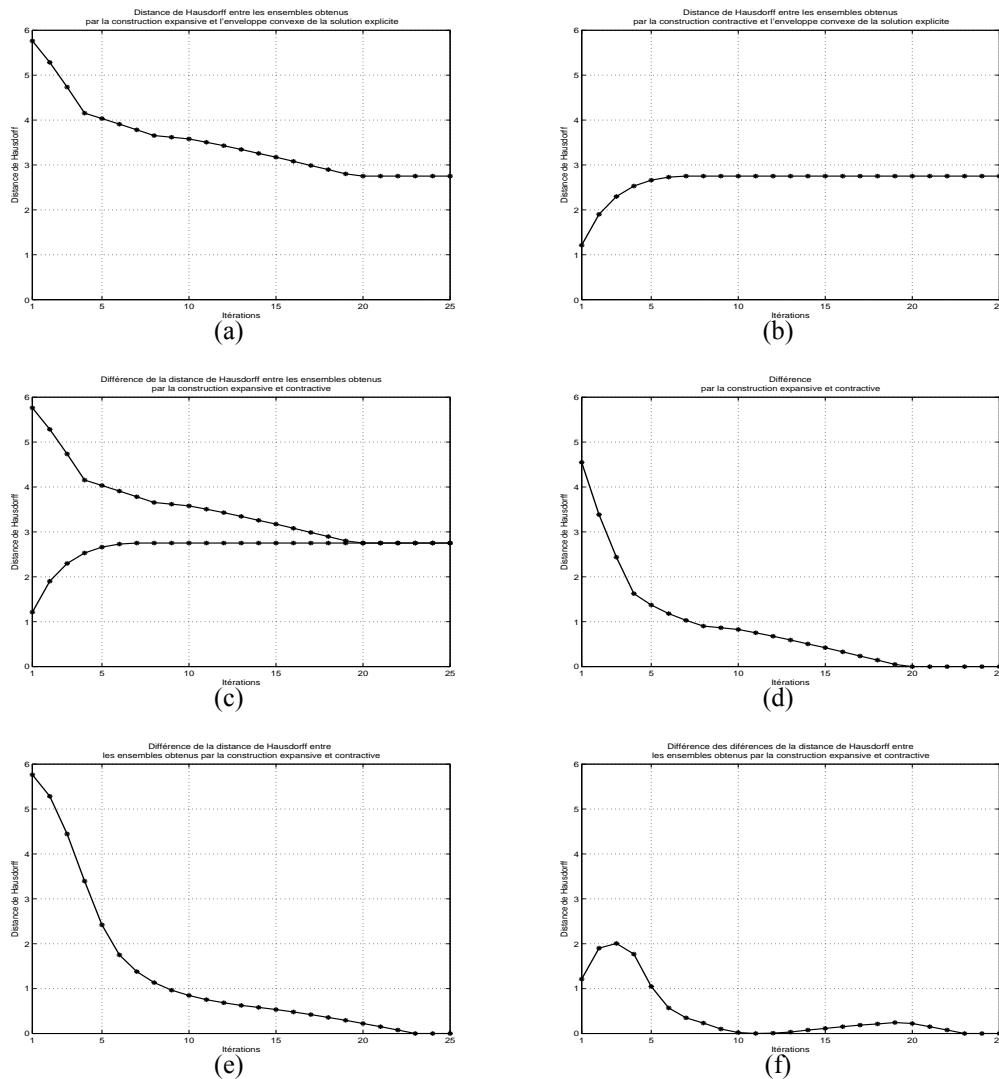


FIGURE 4.10 – Illustration graphique de la convergence entre les ensembles obtenus par les constructions expansive et contractive. (a) la distance de Hausdorff entre les ensembles de la construction expansive et la coque convexe de l'ensemble \mathbb{X} . (b) la distance de Hausdorff entre les ensembles de la construction contractive et la coque convexe de l'ensemble \mathbb{X} . (c) superposition de (a) et (b). (d) la différence entre les distances (a) et (b). (e) la distance de Hausdorff entre les ensembles des constructions expansive et contractive. (f) différence entre (d) et (e).

On définit le max et le min de $projection_j(\mathcal{R}_i)$ comme :

$$\begin{aligned} \max(projection_j(\mathcal{R}_i)) &= \max\{s_{i,1}^j, s_{i,2}^j, \dots, s_{i,m_i}^j\}, \\ \min(projection_j(\mathcal{R}_i)) &= \min\{s_{i,1}^j, s_{i,2}^j, \dots, s_{i,m_i}^j\} \end{aligned}$$

et on définit les intervalles correspondant à \mathcal{R}_i sur chaque dimension j comme :

$$intervalle_j(\mathcal{R}_i) = [\min(projection_j(\mathcal{R}_i)), \max(projection_j(\mathcal{R}_i))] \quad (4.12)$$

Pour illustrer ces définitions, on prend comme exemple la figure 4.11. La région $\mathcal{R}_i \in \mathbb{R}^2$ est définie par 4 sommets :

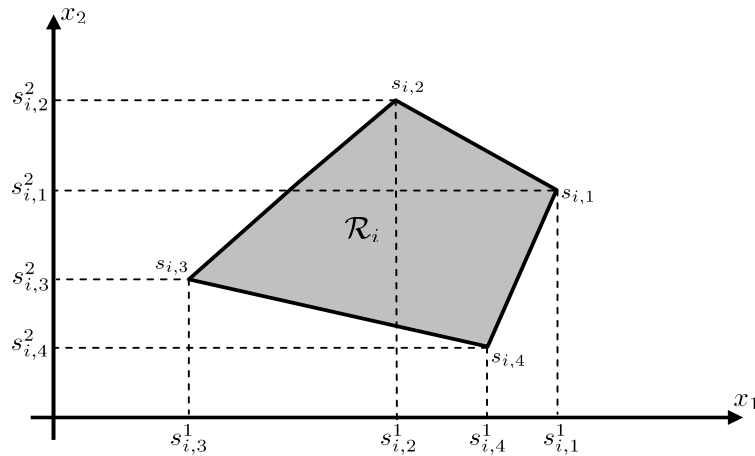


FIGURE 4.11 – Région \mathcal{R}_i et projection de ses sommets.

$$\mathcal{R}_i = [s_{i,1} \ s_{i,2} \ s_{i,3} \ s_{i,4}] = \begin{bmatrix} s_{i,1}^1 & s_{i,2}^1 & s_{i,3}^1 & s_{i,4}^1 \\ s_{i,1}^2 & s_{i,2}^2 & s_{i,3}^2 & s_{i,4}^2 \end{bmatrix}$$

Les projections de leurs sommets sur les deux dimensions sont :

$$\begin{aligned} \text{projection}_1(\mathcal{R}_i) &= [s_{i,1}^1 \ s_{i,2}^1 \ s_{i,3}^1 \ s_{i,4}^1] \\ \text{projection}_2(\mathcal{R}_i) &= [s_{i,1}^2 \ s_{i,2}^2 \ s_{i,3}^2 \ s_{i,4}^2] \end{aligned}$$

Les max, min de ces projections sont donnés par :

$$\begin{aligned} \max(\text{projection}_1(\mathcal{R}_i)) &= \max\{s_{i,1}^1, s_{i,2}^1, s_{i,3}^1, s_{i,4}^1\} = s_{i,1}^1 \\ \min(\text{projection}_1(\mathcal{R}_i)) &= \min\{s_{i,1}^1, s_{i,2}^1, s_{i,3}^1, s_{i,4}^1\} = s_{i,3}^1 \\ \max(\text{projection}_2(\mathcal{R}_i)) &= \max\{s_{i,1}^2, s_{i,2}^2, s_{i,3}^2, s_{i,4}^2\} = s_{i,2}^2 \\ \min(\text{projection}_2(\mathcal{R}_i)) &= \min\{s_{i,1}^2, s_{i,2}^2, s_{i,3}^2, s_{i,4}^2\} = s_{i,4}^2 \end{aligned}$$

et les intervalles correspondant pour chaque dimension sont :

$$\text{intervalle}_1(\mathcal{R}_i) = [s_{i,3}^1, s_{i,1}^1]$$

$$\text{intervalle}_2(\mathcal{R}_i) = [s_{i,4}^2, s_{i,2}^2]$$

Maintenant pour $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i$, on définit l'ensemble :

$$\Delta(\mathbb{X}) : \left\{ \begin{array}{l} \Delta_1(\mathbb{X}) = \begin{bmatrix} \text{intervalle}_1(\mathcal{R}_1) \\ \text{intervalle}_1(\mathcal{R}_2) \\ \vdots \\ \text{intervalle}_1(\mathcal{R}_{n_r}) \end{bmatrix} \\ \Delta_2(\mathbb{X}) = \begin{bmatrix} \text{intervalle}_2(\mathcal{R}_1) \\ \text{intervalle}_2(\mathcal{R}_2) \\ \vdots \\ \text{intervalle}_2(\mathcal{R}_{n_r}) \end{bmatrix} \\ \vdots \\ \Delta_n(\mathbb{X}) = \begin{bmatrix} \text{intervalle}_n(\mathcal{R}_1) \\ \text{intervalle}_n(\mathcal{R}_2) \\ \vdots \\ \text{intervalle}_n(\mathcal{R}_{n_r}) \end{bmatrix} \end{array} \right. \quad (4.13)$$

qui correspond à la collection d'intervalles qui caractérisent l'ensemble $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i$. Deux fonctions sont définies :

$$\text{arbre}(\Delta_j(\mathbb{X})) \quad (4.14)$$

qui permet de créer un arbre d'intervalles [57] à partir d'un ensemble d'intervalles $\Delta_j(\mathbb{X})$ et

$$\text{chercher}(\text{arbre}(\Delta_j(\mathbb{X})), \text{intervalle}_j(\mathcal{R}_i)) \quad (4.15)$$

qui permet de chercher les intervalles dans $\text{arbre}(\Delta_j(\mathbb{X}))$ qui chevauchent $\text{intervalle}_j(\mathcal{R}_i)$ et retourne les indices correspondant.

Dans les algorithmes 1 et 2, les étapes (7~13) et (9~12) sont basées sur un balayage de toutes les régions de l'ensemble \mathbb{X} dans la recherche d'un intervalle non vide. Dans ce contexte, la recherche par intervalle peut nous fournir une amélioration importante en calculant $\text{preImage}(\mathcal{F}(i), \mathcal{F})$ (pour l'algorithme 1) et $\text{preImage}(\mathcal{S}(i), \mathcal{X}^{iter})$ (pour l'algorithme 2) non pas pour tout l'ensemble \mathcal{F} ou \mathcal{X}^{iter} , mais seulement avec des **régions candidates**. Les régions candidates \mathcal{R}_{can} pour une région donnée \mathcal{R}_i , sont données par :

$$\mathcal{R}_{can} = \bigcap_{j=1}^n \text{chercher}(\text{arbre}(\Delta_j(\mathbb{X})), \text{intervalle}_j(\mathcal{R}_i)). \quad (4.16)$$

Pour illustrer le principe de l'utilisation de l'analyse par intervalles, en partant d'une partition comme celle de la figure 4.12, la question qui se pose est : *Quelles sont les régions candidates dans $\mathbb{X} = \bigcup_{i=1}^8 \mathcal{R}_i$ qui peuvent intersecter $\text{image}(\mathcal{R}_8)$?*

La méthode consiste à chercher pour la dimension 1 les régions candidates :

$$\{1, 2, 3, 4, 8, 9\} = \text{chercher}(\text{arbre}(\Delta_1(\mathbb{X})), \text{intervalle}_1(\text{image}(\mathcal{R}_8)))$$

et pour la dimension 2 :

$$\{1, 2, 3, 4, 6, 7, 8\} = \text{chercher}(\text{arbre}(\Delta_2(\mathbb{X})), \text{intervalle}_2(\text{image}(\mathcal{R}_8)))$$

et ainsi les indices des régions candidates sont :

$$\{1, 2, 3, 4, 8\} = \bigcap_{j=1}^2 \text{chercher}(\text{arbre}(\Delta_j(\mathbb{X})), \text{intervalle}_j(\text{image}(\mathcal{R}_8))).$$

Ceci permet de déterminer les régions candidates $\{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4, \mathcal{R}_8\}$ qui peuvent intersecter $\text{image}(\mathcal{R}_8)$.

Remarque 10. Une région candidate pour \mathcal{R}_i ne garantit pas une intersection non nulle. En effet l'intersection des projections reste une condition nécessaire. Une transformation des coordonnées peut être considérée pour améliorer la séparation des projections. Comme on peut le voir sur la figure 4.12, les régions candidates $\{\mathcal{R}_1, \mathcal{R}_4\}$ n'intersectent pas $\text{image}(\mathcal{R}_8)$, ce qui veut dire que :

$$\text{preImage}(\mathcal{R}_8, \mathcal{R}_1) = \text{preImage}(\mathcal{R}_8, \mathcal{R}_4) = \emptyset$$

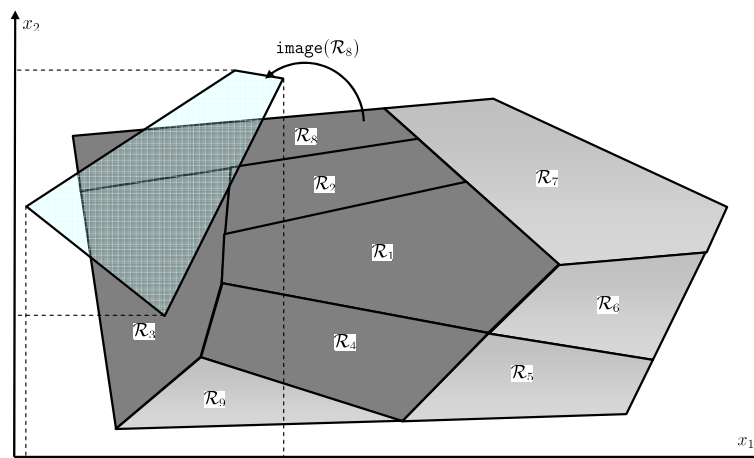


FIGURE 4.12 – En gris foncé, les régions candidates qui peuvent intersecter $\text{image}(\mathcal{R}_8)$.

Pour l'instant seule la recherche par intervalles a été utilisée pour améliorer le temps de calcul des algorithmes. Les paragraphes suivants développent comment créer cet arbre et comment le parcourir pour chercher les intervalles qui chevauchent un intervalle donné.

Création de l'arbre d'intervalles

Pour expliquer le principe de la création d'un arbre d'intervalles, considérons l'exemple de la figure 4.13 avec un ensemble de 7 intervalles $\{I_1, I_2, I_3, I_4, I_5, I_6, I_7\}$. La construction de l'arbre suit les séquences suivantes :

- calculer la médiane x_{mid} en considérant toutes les extrémités de tous les intervalles ;

- sauvegarder dans le nœud $\mathcal{A}_{mid} = \{I_3, I_4, I_5\}$ tous les intervalles qui intersectent la médiane x_{mid} , dans le nœud $\mathcal{A}_d = \{I_6, I_7\}$ tous les intervalles qui se situent complètement à droite de x_{mid} , et dans le nœud $\mathcal{A}_g = \{I_1, I_2\}$ tous les intervalles qui se situent complètement à gauche de x_{mid} ;
- considérer les nœuds \mathcal{A}_d et \mathcal{A}_g comme des ensembles initiaux, et réitérer les mêmes opérations, jusqu'à ce que tous les intervalles aient été parcourus ;
- organiser pour tous les nœuds trouvés (dans notre exemple \mathcal{A}_{mid} , \mathcal{A}_d et \mathcal{A}_g) leur contenu par ordre ascendant en considérant les extrémités gauches de chaque intervalle, et par ordre descendant en considérant les extrémités droites de chaque intervalle, ce qui donne $\{I_3, I_4, I_5\} \leftarrow \mathcal{A}_{mid} \rightarrow \{I_5, I_3, I_4\}$, $\{I_6, I_7\} \leftarrow \mathcal{A}_d \rightarrow \{I_7, I_6\}$ et $\{I_1, I_2\} \leftarrow \mathcal{A}_g \rightarrow \{I_1, I_2\}$.

L'algorithme 3 permet la création de l'arbre d'intervalles.

Algorithme 3 Création de l'arbre d'intervalle $\mathcal{A} = \text{arbre}(\Delta_j(\mathbb{X}))$

ENTRÉES: Ensemble d'intervalles $\Delta_j(\mathbb{X})$

- 1: calculer x_{mid} la médiane en considérant toutes les extrémités de l'ensemble des intervalles $\Delta_j(\mathbb{X})$
- 2: $\mathcal{A}_{mid} \leftarrow$ les intervalles qui intersectent x_{mid}
- 3: organiser le contenu de \mathcal{A}_{mid} par ordre ascendant on considérant les extrémités gauches de chaque intervalle, et par ordre descendant on considérant les extrémités droites de chaque intervalle.
- 4: $\mathcal{A}_d \leftarrow$ les intervalles qui se trouvent complètement à droite de x_{mid}
- 5: $\mathcal{A}_g \leftarrow$ les intervalles qui se trouvent complètement à gauche de x_{mid}
- 6: $\mathcal{A}_d(\mathcal{A}_{mid}) \leftarrow \text{arbre}(\mathcal{A}_d)$
- 7: $\mathcal{A}_g(\mathcal{A}_{mid}) \leftarrow \text{arbre}(\mathcal{A}_g)$

SORTIES: $\mathcal{A} = \text{arbre}(\Delta_j(\mathbb{X}))$.

Parcourir l'arbre des intervalles

Cette section est dédiée à l'utilisation d'un arbre d'intervalles pour la génération d'une liste des régions candidates pour l'intersection. Pour chaque région la recherche va se faire sur les intervalles donnés par les projections des régions en cause.

Si un intervalle I_1 intersecte un intervalle I_2 , alors uniquement 4 configurations sont possibles concernant les positions de I_1 par rapport à I_2 ; la figure 4.14 montre ces 4 possibilités.

Si l'une des extrémités de I_1 est à l'intérieur de I_2 , alors I_1 intersecte I_2 . Donc il suffit de considérer les extrémités de I_1 et chercher dans l'arbre des intervalles, les intervalles qui l'intersectent. L'algorithme 4 effectue cette recherche. Pour chercher les intervalles qui intersectent un point x_0 par exemple on suit les étapes suivantes :

- pour chaque nœud constituant l'arbre des intervalles on compare son point médian x_{mid} avec x_0 .

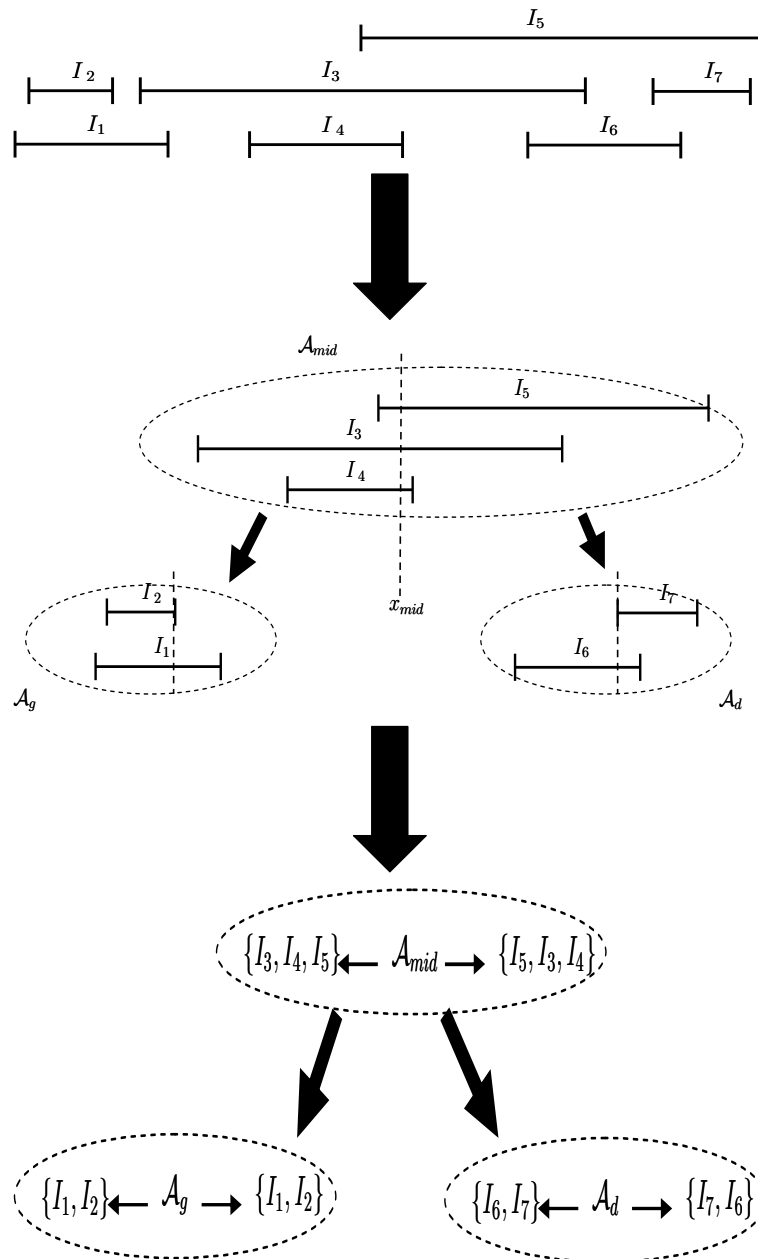


FIGURE 4.13 – Les 3 phases principales pour la création des nœuds centraux, de droite et de gauche.

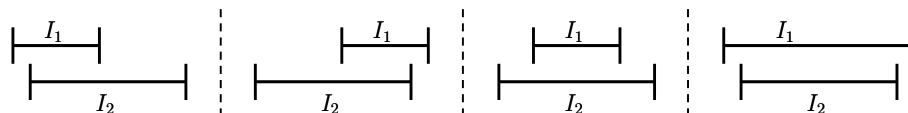


FIGURE 4.14 – Les 4 positions possible de I_1 par rapport à I_2 si ils s'intersectent.

– si x_0 est plus grand que x_{mid} alors seulement les intervalles de droite seront considérés, de

plus, on scrute les intervalles de \mathcal{A}_{mid} qui sont sortis par ordre descendant en commençant par le point le plus à gauche, et on s'arrête quand le point x_0 n'est plus inférieur aux extrémités gauches. On continue de la même façon jusqu'au parcours de tous les intervalles.

L'algorithme 4 permet de trouver les intervalles dans \mathcal{A} qui chevauchent un point x_0 .

Algorithme 4 $chercher(\mathcal{A}, x_0)$, cherche les intervalles dans \mathcal{A} qui intersectent le point x_0

ENTRÉES: Arbre d'intervalles \mathcal{A} et le point x_0

- 1: **tantque** on n'a pas parcouru l'arbre \mathcal{A} **faire**
- 2: **if** $x_0 > x_{mid}$ **then**
- 3: scruter et sauvegarder les indices des intervalles de \mathcal{A}_{mid} qui sont sortis par ordre descendant en commençant par le point le plus à gauche, et on s'arrête quand le point x_0 n'est plus inférieur aux extrémités gauches.
- 4: $chercher(\mathcal{A}_d, x_0)$
- 5: **else**
- 6: scruter et sauvegarder les indices des intervalles de \mathcal{A}_{mid} qui sont sortis par ordre ascendant en commençant par le point le plus à droite, et on s'arrête quand le point x_0 n'est plus supérieur aux extrémités droites.
- 7: $chercher(\mathcal{A}_g, x_0)$
- 8: **end if**
- 9: **fin tantque**

SORTIES: indices des intervalles qui intersectent le point x_0 .

Avec les procédures introduites, il est possible de disposer des éléments nécessaires pour la construction d'une liste des régions candidates à l'intersection avec une région donnée \mathcal{R} . Soit $I = projection_i(\mathcal{R})$ et l'arbre des intervalles $arbre(\Delta_i(\mathbb{X}))$ associé à la projection $projection_i(\mathbb{X})$, la construction passe par les étapes décrites par l'algorithme 5.

Algorithme 5 $chercher(\mathcal{A}, I)$ cherche les intervalles dans \mathcal{A} qui sont candidats à intersecter I

ENTRÉES: L'arbre \mathcal{A} et l'intervalle I

- 1: **if** $x_{mid}(\mathcal{A}) \in I$ **then**
- 2: retenir les intervalles de \mathcal{A}_{mid} comme candidats
- 3: ajouter aux candidats les intervalles résultant de $chercher(\mathcal{A}_d, I)$
- 4: ajouter aux candidats les intervalles résultant de $chercher(\mathcal{A}_g, I)$
- 5: **else**
- 6: $x_0 \leftarrow$ les extrémités de l'intervalle I
- 7: $chercher(\mathcal{A}, x_0)$ en utilisant l'algorithme 4
- 8: **end if**

SORTIES: les intervalles candidats.

Exemple 14. Temps de calcul avec et sans analyse par intervalles.

Considérons la construction d'une loi de commande prédictive pour le système suivant :

$$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 0.5 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.5 \\ 0.125 \end{bmatrix} u_k \quad (4.17)$$

avec les contraintes :

$$\begin{bmatrix} \mathbb{I}_2 \\ \mathbb{I}_2 \end{bmatrix} x_k \leq 10 \begin{bmatrix} \mathbb{1}_{2,1} \\ -\mathbb{1}_{2,1} \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} u_k \leq \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

avec les horizons de prédiction et de commande $N = 7$, $N_u = 7$, les matrices de pondération $Q = \mathbb{I}_2$ et $R = 0.1$. La loi de commande explicite possède initialement 99 régions. En utilisant l'algorithme 1 avec et sans analyse par intervalles, pour un calcul de 20 itérations, le tableau 4.1 montre la différence entre ces deux approches.

nombre d'itérations	3	8	10	15	20
nombre de régions	99	110	119	155	189
nbr. d'opérations sans analyse par intervalles	9801	12100	14161	24025	35721
nbr. d'opérations avec analyse par intervalles	2657	2666	3020	4479	6116

TABLE 4.1 – Comparaison entre le nombre d'opérations (*image* et *preImage*), effectués par l'algorithme 1 avec et sans utilisation de l'analyse par intervalles.

4.4 Analyse par graphe de transitions

Considérons le système PWA défini par l'équation (4.1), en notant le complément de \mathbb{X} par rapport à \mathbb{R}^n :

$$\mathcal{C} = \mathbb{R}^n \setminus \mathbb{X} \quad (4.18)$$

Définition 7. Un graphe de transitions associé au système PWA (4.1) est décrit par la paire (Ω, Θ) où Ω est l'ensemble des nœuds, et Θ l'ensemble des arcs. Chaque région \mathcal{R}_i , $\forall i = 1, \dots, n_r$ est associée à un nœud $\mathcal{N}_i \in \Omega$. Par défaut le nœud \mathcal{N}_0 sera associé à la région \mathcal{C} . L'existence d'un arc orienté t_{ij} entre les nœuds \mathcal{N}_i et \mathcal{N}_j est équivalent à l'existence d'au moins une trajectoire qui démarre dans $x \in \mathcal{R}_i$ et satisfait $A_i x_k + a_i \in \mathcal{R}_j$. \square

D'un point de vue pratique le graphe *orienté* associé au système PWA est défini à partir d'opérations géométriques sur les polytopes comme il est illustré sur la figure 4.15.

La procédure de construction d'un graphe de transition vérifie l'intersection de chaque $image(\mathcal{R}_i)$ avec la partition définissant le système. En utilisant les opérateurs *image* et *preImage*, les arcs Θ peuvent être sauvegardés dans une matrice (généralement non symétrique) $\mathcal{M} \in \{0, 1\}^{(n_r+1) \times (n_r+1)}$. C'est ce que fait l'algorithme 6.

Remarque 11. La ligne et la colonne $(n_r + 1)$ correspondent au *nœud de sortie* \mathcal{N}_0 .

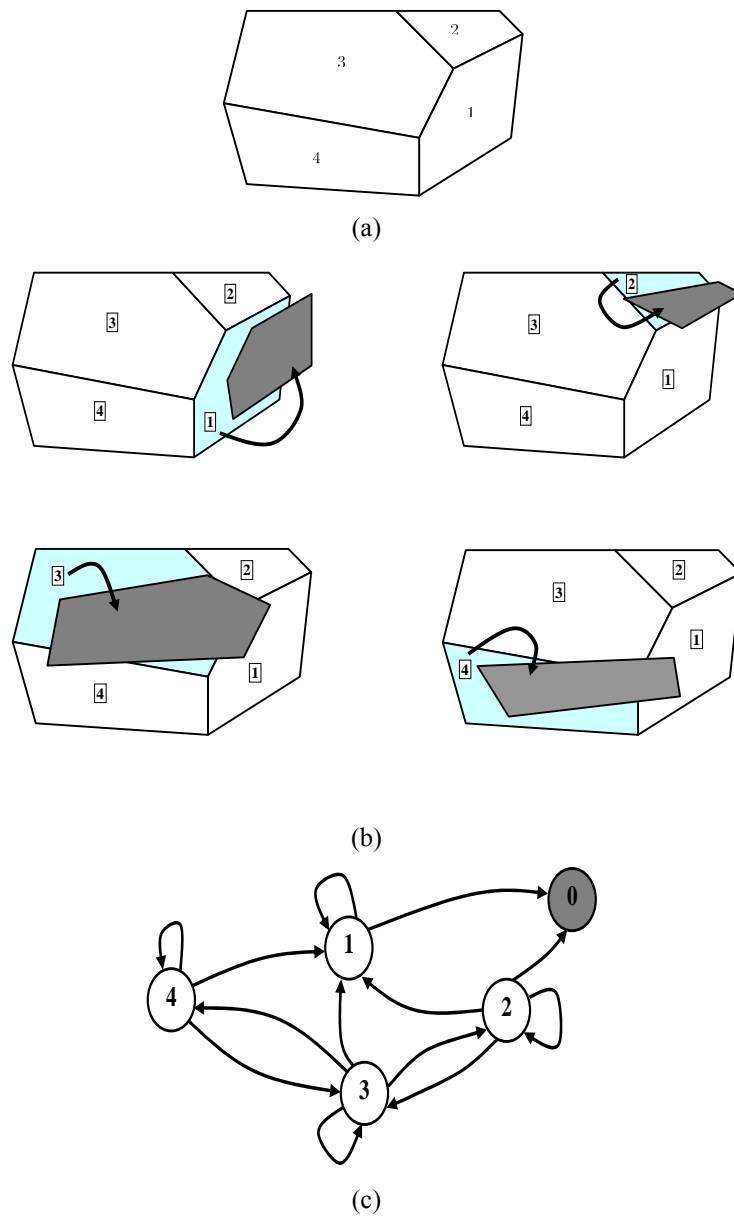


FIGURE 4.15 – Exemple d'un système PWA défini par sa partition (a) et son graphe de transitions (c). Le nœud (0) représente le *nœud de sortie* (il possède seulement des arcs entrants) et correspond à la région complémentaire C par rapport aux régions définissant le système PWA. La construction est obtenue par intersections de l'image de chaque région avec la partition qui définit le système PWA.

Algorithme 6 Calcul de la matrice de transitions $\mathcal{M} \in \{0, 1\}^{(n_r+1) \times (n_r+1)}$

ENTRÉES: système PWA défini dans $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i$

```

1:  $\mathcal{M} \leftarrow 0_{(n_r+1) \times (n_r+1)}$ 
2: pour  $ind_1 = 1$  à  $n_r$  faire
3:    $S_{ind_1} = \emptyset$ 
4:   pour  $ind_2 = 1$  à  $n_r$  faire
5:      $IP = image(\mathcal{R}_i) \cap \mathcal{R}_j$ 
6:     if  $IP = \emptyset$  then
7:        $\mathcal{M}(ind_1, ind_2) \leftarrow 0$ 
8:     else
9:        $\mathcal{M}(ind_1, ind_2) \leftarrow 1$ 
10:    end if
11:     $S_{ind_1} = S_{ind_2} \cup preImage(\mathcal{R}_i, IP)$ 
12:  fin pour
13:  if  $\mathcal{R}_i \setminus S_{ind_1} = \emptyset$  then
14:     $\mathcal{M}(ind_1, n_r + 1) \leftarrow 0$ 
15:  else
16:     $\mathcal{M}(ind_1, n_r + 1) \leftarrow 1$ 
17:  end if
18: fin pour

```

SORTIES: Matrice de transitions \mathcal{M}

L'algorithme 6 calcule n_r^2 opérations d'intersections, il peut être très gourmand en temps de calcul pour des systèmes définis sur un nombre de partitions très élevé. L'analyse par intervalle peut être utilisée pour diminuer considérablement le nombre d'intersections en fournissant les régions candidates, et par voie de conséquence le temps de calcul [52].

4.4.1 Propriétés des graphes de transitions

L'algorithme 6 est basé sur le calcul d'*image*, et chaque ligne et colonne de la matrice \mathcal{M} correspond à un nœud du graphe de transitions tel que si le nœud \mathcal{N}_i est connecté au nœud \mathcal{N}_j , alors, $\mathcal{M}(i, j) = 1$ sinon $\mathcal{M}(i, j) = 0$.

Remarque 12. La matrice de transitions contient sur la ligne et la colonne $n_r + 1$ des informations sur la transition depuis/vers le **nœud de sortie** \mathcal{N}_0 . Seulement les arcs entrant peuvent exister pour \mathcal{N}_0 et par conséquent la ligne $n_r + 1$ de la matrice \mathcal{M} est remplie par des zéros.

Par exemple, dans le cas du système PWA décrit par la figure 4.15, l'ensemble des arcs Θ est complètement décrit par la matrice suivante :

$$\mathcal{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

La matrice des transitions \mathcal{M} offre des informations de voisinage qui ne sont pas transparentes dans la description du système PWA original. Bien sûr, la reconstruction de cette information nécessite des évaluations de la dynamique, tâches gourmandes en temps de calcul.

L'analyse de l'atteignabilité peut être effectuée en suivant les chemins (arcs) dans le graphe de transitions. Pour calculer par exemple tous les nœuds qui atteignent le nœud de sortie \mathcal{N}_0 pour l'exemple de la figure 4.15, il faut calculer $[0 \ 0 \ 0 \ 0 \ 1] \mathcal{M}^T = [1 \ 1 \ 0 \ 0 \ 0]$. Inversement pour connaître quels nœuds sont atteints par le nœud \mathcal{N}_1 pour l'exemple de la figure 4.15, il faut calculer $[1 \ 0 \ 0 \ 0 \ 0] \mathcal{M} = [1 \ 0 \ 0 \ 0 \ 1]$ pour trouver que le nœud \mathcal{N}_1 transite vers lui même et vers le nœud de sortie \mathcal{N}_0 .

Remarque 13. Par la suite dans la structure des données liées à un graphe de transitions il sera considéré que pour chaque nœud $\mathcal{N}_i \in \Omega$, seront mémorisées les régions \mathcal{R}_i ainsi que l'ensemble S_i , calculé dans l'algorithme 6. Ces régions $S_i \subset \mathcal{R}_i$ représentent les ensembles qui transitent en un seul pas en dehors du domaine de validité $image(S_i) \subset \mathcal{C}$.

4.5 Calcul des ensembles invariants basé sur les graphes de transitions

Dans cette section une nouvelle procédure est proposée pour la construction des ensembles invariants pour les systèmes affines par morceaux PWA. Une focalisation sera faite sur la possibilité de transition vers le nœud de sortie \mathcal{N}_0 dans le graphe de transitions.

Proposition 5. Considérons le domaine de définition $\mathbb{X} = \cup_{i=1}^{n_r} \mathcal{R}_i$ pour un système PWA, soit (Ω, Θ) le graphe de transitions pour le même système. Si le nœud \mathcal{N}_0 est isolé, alors l'ensemble \mathbb{X} est invariant à l'égard des dynamiques du système PWA.

Démonstration : L'invariance du domaine \mathbb{X} vis-à-vis d'une dynamique donnée implique qu'il n'y a aucune trajectoire en dehors du domaine. Toutes les trajectoires qui sortent du domaine \mathbb{X} sont décrites par des chemins dans le graphe de transitions qui se terminent par le nœud de sortie \mathcal{N}_0 . Si \mathcal{N}_0 est un nœud isolé, alors aucun chemin ne peut être construit, et ainsi l'invariance du domaine $\mathbb{X} = \cup_{i=1}^{n_r} \mathcal{R}_i$ est garantie. \square

L'algorithme 7 calcule le graphe de transitions pour un système PWA donné, ensuite itère dans chaque région \mathcal{R}_i connectée au nœud de sortie \mathcal{N}_0 afin d'éliminer les sous-régions qui peuvent évoluer vers l'extérieur de \mathbb{X} par la $i^{\text{ème}}$ dynamique correspondante. Une fois toutes les régions connectées à \mathcal{N}_0 traitées, l'ensemble des arcs Θ est mis à jour et la procédure est relancée jusqu'à ce que l'isolation du nœud \mathcal{N}_0 soit achevée.

Hypothèse 1. Pour le système PWA (4.1), il existe un unique $i \in \{1, \dots, n_r\}$ tel que $0 \in \mathcal{R}_i$ \square

Considérons les définitions suivantes :

Définition 8. Pour un domaine donné $\mathcal{R}_i \subset \mathbb{X}$, la sous-région $out\mathcal{R}_i^k$ est définie comme :

$$out\mathcal{R}_i^k = \{x_0 \in \mathcal{R}_i \mid x_l \in \mathcal{R}_i, \forall l = 1, \dots, k-1 \wedge x_k \notin \mathbb{X} \text{ avec} \\ x_l = A_i^l x_0 + (I + \dots + A_i^{l-1}) a_i\} \quad (4.19)$$

ce qui veut dire que le k -image de \mathcal{R}_i sorte de P dans k pas en progressant exclusivement à l'intérieur de \mathcal{R}_i . \square

La figure 4.16 illustre le principe de construction.

Définition 9. Pour une région \mathcal{R}_i et un indice donné $k \in \mathbb{N}^*$ le complément de la sous-région à l'égard de $out\mathcal{R}_i^k$ est :

$$inv\mathcal{R}_i^k = \mathcal{R}_i \setminus \bigcup_{s=1}^k out\mathcal{R}_i^s \quad (4.20)$$

\square

Définition 10. La sous-région $out\mathcal{R}_i^\infty$ est définie comme :

$$out\mathcal{R}_i^\infty = \bigcup_{k=1}^{\infty} out\mathcal{R}_i^k$$

et son complément est : $inv\mathcal{R}_i^\infty = \mathcal{R}_i \setminus out\mathcal{R}_i^\infty$. \square

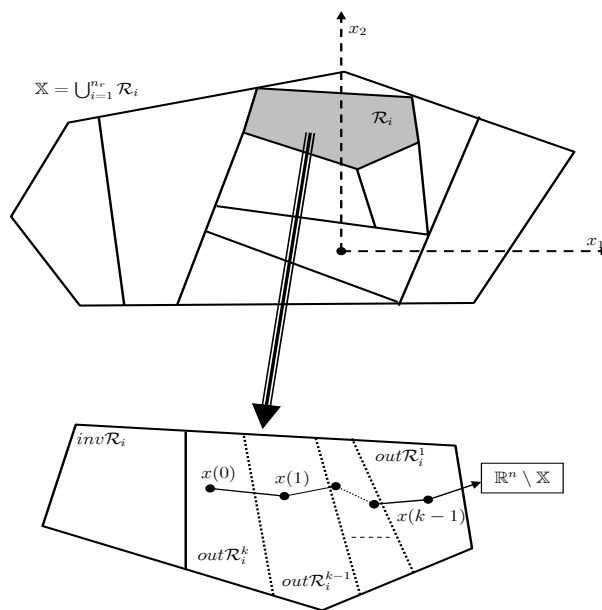


FIGURE 4.16 – Illustration graphique de la partition $\mathcal{R}_i = inv\mathcal{R}_i^k \cup \bigcup_{s=1}^k out\mathcal{R}_i^s$

Pour chaque nœud connecté au nœud de sortie \mathcal{N}_0 , la région correspondante est décomposée comme suit :

$$\mathcal{R}_j = inv\mathcal{R}_j^k \cup out\mathcal{R}_j^k$$

où $out\mathcal{R}_j^k$ est le sous-ensemble qui force la transition vers \mathcal{N}_0 exclusivement à travers sa dynamique définie pour la région \mathcal{R}_j . $inv\mathcal{R}_j^k$ est le complément vis-à-vis la région \mathcal{R}_j . Une question intéressante concerne la détermination du nombre de pas avant la transition effective vers le nœud de sortie \mathcal{N}_0 . Dans la section suivante les conditions seront données pour que le nombre

Algorithme 7 Calcul de l'ensemble invariant utilisant le graphe de transitions.

ENTRÉES: Système PWA (4.1) et le nombre d'itérations maximal $iterMax$

- 1: Calcul du graphe de transitions correspondant au système PWA (4.1) en utilisant l'algorithme 6. En résultent la matrice \mathcal{M} et l'ensemble des paires $(\mathcal{R}_i, \mathcal{S}_i), i = 1, \dots, n_r$ représentant les nœuds.
- 2: $iter \leftarrow 1$
- 3: **tantque** $(\exists i \text{ s.t. } t_{i0} \in \Theta) \wedge (iter \leq iterMax)$ **faire**
- 4: Sélectionner le sous ensemble maximal $\mathbf{J} \subset \{1, \dots, n_r\}$ tel que $t_{i0} \in \Theta, \forall i \in \mathbf{J}$
- 5: **pour** $j \in \mathbf{J}$ **faire**
- 6: $k = 1$
- 7: $out\mathcal{R}_j^k = \mathcal{S}_j$
- 8: $inv\mathcal{R}_j^k = \mathcal{R}_j \setminus \mathcal{S}_j$
- 9: $IP = out\mathcal{R}_j^k$
- 10: **tantque** $(IP \neq \emptyset)$ **faire**
- 11: $k \leftarrow k + 1$
- 12: $IP = preImag_j(Image(inv\mathcal{R}_j^k) \cap out\mathcal{R}_j^k)$
- 13: $out\mathcal{R}_j^k = out\mathcal{R}_j^k \cup IP$
- 14: $inv\mathcal{R}_j^k = inv\mathcal{R}_j^k \setminus IP$
- 15: **fin tantque**
- 16: $\mathcal{R}_j = inv\mathcal{R}_j^k$
- 17: $\mathcal{S}_j = out\mathcal{R}_j^k$
- 18: **fin pour**
- 19: reconstruire le graphe de transitions pour prendre en compte la nouvelle définition du système PWA
- 20: $iter \leftarrow iter + 1$
- 21: **fin tantque**

SORTIES: Ensemble exact ou l'approximation de Φ^+ = l'ensemble final après reconstruction des graphes de transitions durant les étapes (3 ~ 21).

d'itérations soit fini dans l'étape (10 ~ 15) de l'algorithme 7. On note que si cette portion d'algorithme est assurée d'effectuer un nombre fini d'itérations, alors la solution exacte n'est pas garantie à cause de la reconstruction du graphe de transitions dans la boucle (3 ~ 21)⁸. Si le nombre maximal d'itérations est atteint, alors l'ensemble obtenu Φ^+ est une approximation externe de l'ensemble invariant maximal MPI.

L'avantage principal est que chaque ensemble d'opérations effectuées par l'algorithme 7 fournit des informations utiles sur la convergence vers l'ensemble invariant. Au lieu des itérations utilisant la dynamique directe inverse (*image* et *preImage*) sur l'ensemble \mathbb{X} , l'algorithme 7 exécute le nombre nécessaire d'itérations utilisant le calcul par (*image* et *preImage*) pour compléter le calcul pour une seule région. Si l'algorithme s'arrête avant la construction exacte de l'ensemble invariant, il fournit des informations utiles, à savoir la description exacte des

8. Il est connu que dans le cas général, même pour les systèmes linéaires, la détermination en temps fini de l'ensemble invariant n'est pas garantie [58].

régions qui devraient être mise à jour $\mathcal{R}_i \setminus \mathcal{S}_i$ pour renforcer l'invariance.⁹

Remarque 14. La procédure décrite dans l'algorithme 7 fournit une approximation de l'ensemble invariant maximal du système PWA. Chaque itération dans la construction de l'ensemble MPI pour les systèmes linéaires peut être réalisée avec une approximation donnée [59] en déterminant la taille de $out\mathcal{R}_i^k$. Une idée pour contourner le problème de détermination exacte en un nombre d'itérations fini consiste à modifier les dynamiques du système PWA dans les régions de sortie $-\mathcal{R}_i \setminus \mathcal{S}_i$ pour renforcer l'invariance. Il y a deux problèmes liés à cette construction :

- La possibilité de perdre la continuité du système PWA
- Les changements du système PWA pour renforcer l'invariance ne sont pas uniques, ainsi les candidats appropriés vis-à-vis des objectifs de conception de la loi de commande originale doivent être choisis.

Remarque 15. À l'étape 19 de l'algorithme 7, la reconstruction du graphe de transitions doit être faite localement par rapport aux nœuds et arcs ajoutés à l'itération courante.

4.5.1 Calcul en nombre d'itérations fini

Dans la section précédente, un des points clé de la construction de l'ensemble invariant est le calcul de la partition pour chaque nœud $\mathcal{R}_j = inv\mathcal{R}_j^k \cup_{s=1}^k out\mathcal{R}_j^s$. Ce calcul correspond aux étapes 10 ~ 15 de l'algorithme 7. Le théorème suivant donne les conditions pour lesquelles le calcul effectué pour chaque région se fait en un nombre d'itérations fini.

Théorème 6. Soit un système PWA défini sur l'ensemble $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i \subset \mathbb{R}^n$ avec $x_{k+1} = f_{PWA}(x_k) = A_i x_k + a_i$ pour $x_k \in \mathcal{R}_i$. Pour toute région \mathcal{R}_i , une description polytopique finie¹⁰ de la partition est :

$$\mathcal{R}_i = inv\mathcal{R}_i^\infty \cup out\mathcal{R}_i^\infty$$

si $(\mathbb{I} - A_i)^{-1}a_i \notin \mathcal{R}_i$ ou $(\mathbb{I} - A_i)^{-1}a_i \in \mathcal{R}_i$ et $|\lambda(A_i)| < 1$.

Démonstration : Considérons $\mathcal{R}_i = \{x \in \mathbb{R}^n | H_{\mathcal{R}_i}x \leq h_{\mathcal{R}_i}\}$ la sous-partition $\mathcal{R}_i = (inv\mathcal{R}_i^1 \cup out\mathcal{R}_i^1) \subset \mathbb{X}$. Pour tout état initial $x(0) \in inv\mathcal{R}_i^1$, l'image d'ordre k de $x(0)$ à l'intérieur de \mathcal{R}_i est donnée par :

$$x_k = A_i^k x(0) + (\mathbb{I} - A_i)^{-1}(\mathbb{I} - A_i^k)a_i \quad (4.21)$$

qui peut être réécrit comme :

$$x_k = A_i^k \{x(0) - (\mathbb{I} - A_i)^{-1}a_i\} + (\mathbb{I} - A_i)^{-1}a_i. \quad (4.22)$$

Supposons que tout $x_l \in \mathcal{R}_i$ pour $l = 0, \dots, k$. Si ce n'est pas le cas alors pour un k fini, x_k sort de \mathcal{R}_i , et par conséquent il n'y a pas d'intérêt pour le problème de calcul en nombre d'itérations fini.

9. L'algorithme 7 peut être utilisé en conjonction avec l'algorithme 2 pour avoir une certaine mesure (en utilisant la distance de Hausdorff par exemple) de l'approximation de l'ensemble invariant exact MPI.

10. le terme fini comprend ici la représentation polytopique de $inv\mathcal{R}_i^\infty$ et plus précisément sous-entend que ces ensembles doivent être décrits par un nombre fini de contraintes linéaires.

On envisage le problème de l'existence de $n^* \in \mathbb{N}$ tel que $out\mathcal{R}_i^\infty = out\mathcal{R}_i^{n^*}$ en sachant que $\forall x(0) \in \mathcal{R}_i$:

$$x(n^* + 1) = A_i^{n^*}(x(0) - (\mathbb{I} - A_i)^{-1}a_i) + (\mathbb{I} - A_i)^{-1}a_i \notin \mathcal{R}_i. \quad (4.23)$$

En excluant le cas trivial $A_i = \mathbb{I}$, deux cas sont traités : le point $(\mathbb{I} - A_i)^{-1}a_i$ à l'intérieur ou à l'extérieur de \mathcal{R}_i .

- $(\mathbb{I} - A_i)^{-1}a_i \in \mathcal{R}_i$:

En effectuant un changement de variable :

$$z(k) = x_k - (\mathbb{I} - A_i)^{-1}a_i \quad (4.24)$$

l'équation (4.22) est réécrite comme :

$$z(k + 1) = A_i z(k). \quad (4.25)$$

et définie sur le nouveau domaine $D_z = \{z \in \mathbb{R}^n \mid H_{\mathcal{R}_i} z \leq h_{\mathcal{R}_i} - H_{\mathcal{R}_i}(\mathbb{I} - A_i)^{-1}a_i\}$. On note que l'origine est incluse dans D_z . La première étape s'intéresse pour cette région au calcul de l'ensemble maximal admissible par rapport à la sortie O_∞ afin de caractériser son ensemble invariant maximal.

Prenant en compte l'hypothèse du théorème, l'étude est restreinte à l'étude du cas $|\lambda(A_i)| < 1$ (dynamique strictement stable) :

Pour ce cas, l'ensemble O_∞ est complètement déterminé en temps fini. En utilisant le théorème 4.1 dans [26] et ses notations : $O_\infty = O_{n^*} = O_{n^*+1}$ pour certain $n^* \in \mathbb{N}$. Une fois O_∞ construit, il peut être soustrait de la région \mathcal{R}_i pour avoir un nouvel ensemble $new\mathcal{R}_i = \mathcal{R}_i \setminus O_\infty$ dégagé de tout ensemble invariant.

Considérons l'ensemble $new\mathcal{R}_i$ avec $0 \notin new\mathcal{R}_i$ et une dynamique stable associée (4.25). Soit $s > n^*$ et en supposant $out\mathcal{R}_i^{s+1} \neq out\mathcal{R}_i^s$, cela implique que $\exists x(0)$ tel que $x(s) \in new\mathcal{R}_i$ et $x(s+1) \notin new\mathcal{R}_i$. Mais $x(s) \in new\mathcal{R}_i \subset \mathcal{R}_i$ veut dire que $x(s) \in O_s = O_{n^*}$, pourvu que $O_{n^*} = O_{n^*+i}, \forall i > 0$. Ce qui nous mène à une contradiction :

$$\begin{cases} x(s) \in new\mathcal{R}_i = \mathcal{R}_i \setminus O_\infty \\ x(s) \in O_s = O_\infty \end{cases} \quad (4.26)$$

La conclusion est que $out\mathcal{R}_i^\infty$ est complètement déterminé, et de plus, il peut être montré que sa construction peut être accomplie en N itérations.

- $(\mathbb{I} - A_i)^{-1}a_i \notin \mathcal{R}_i$: Pour ce cas spécifique que, il existe trois situations :
- $|\lambda(A_i)| < 1$

Dans ce cas on peut toujours trouver $\delta_1, \delta_2 > 0$ et $\mathbb{B}_2^{\delta_1}((\mathbb{I} - A_i)^{-1}a_i)$, $\mathbb{B}_2^{\delta_2}((\mathbb{I} - A_i)^{-1}a_i)$ tels que $\mathcal{R}_i \cap \mathbb{B}_2^{\delta_1}((\mathbb{I} - A_i)^{-1}a_i) = \emptyset$ et $\mathcal{R}_i \subset \mathbb{B}_2^{\delta_2}((\mathbb{I} - A_i)^{-1}a_i)$, voir la figure 4.18. Avec ces éléments $\exists n^*(\delta_1, \delta_2) \in \mathbb{N}$ tel que :

$$A_i^{n^*} \mathbb{B}_2^{\delta_2}((\mathbb{I} - A_i)^{-1}a_i) \subset \mathbb{B}_2^{\delta_1}((\mathbb{I} - A_i)^{-1}a_i)$$

Par conséquent $\forall x(0) \in \mathcal{R}_i$ il existe $l \leq n^*$ tel que

$$A_i^l(x(0) - (\mathbb{I} - A_i)^{-1}a_i) \notin \mathcal{R}_i,$$

Ce qui signifie que en n^* pas, l'utilisation de la dynamique (4.25) va fournir l'ensemble $out\mathcal{R}_i^{n^*}$. Mais $out\mathcal{R}_i^{n^*} = out\mathcal{R}_i^{n^*+1}$, pourvu que $A_i^{n^*} \mathcal{R}_i \cap \mathcal{R}_i = \emptyset$ et ainsi le calcul en nombre de pas fini est assuré.

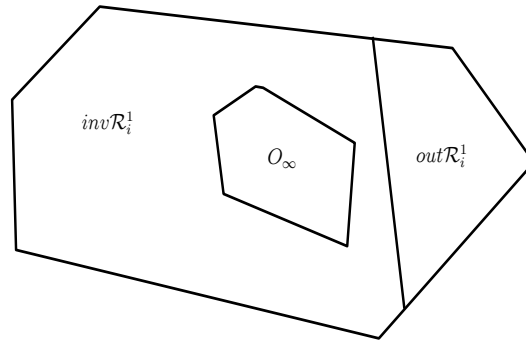


FIGURE 4.17 – L’existence d’un output admissible set pour une région \mathcal{R}_i donnée.

- Une des valeurs propres de A_i est instable.

Dans ce cas on peut toujours trouver une boule $\mathbb{B}_2^\delta((\mathbb{I} - A_i)^{-1}a_i) \supset \mathcal{R}_i$, similaire au cas décrit dans la figure 4.18, tel que $\forall x(0) \in \mathcal{R}_i, \exists n \in \mathbb{N}$ tel que

$$\|x(n) - (\mathbb{I} - A_i)^{-1}a_i\| > \delta,$$

Avec les mêmes arguments que dans le cas strictement stable, on peut conclure sur le calcul en nombre fini de pas.

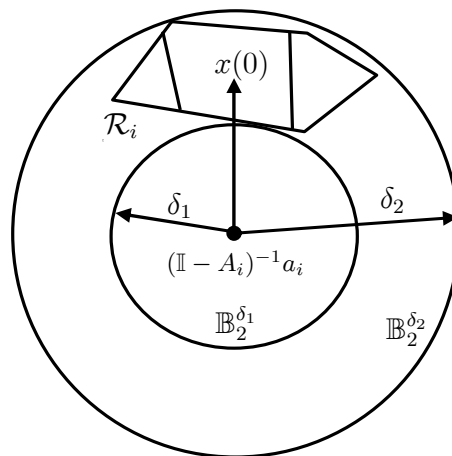


FIGURE 4.18 – L’existence d’une boule pour laquelle $x(n)$ n’est pas située à l’intérieur.

- $|\lambda(A_i)| = 1$:

Pour ce cas, la démonstration est plus compliquée, mais on peut observer que pour $A_i = \mathbb{I}, A_i = -\mathbb{I}$ et la combinaison des valeurs propres réelles sur le disque unitaire, la construction est faite en un seul pas.

En plus, quand on traite les valeurs complexes sur le disque unitaire, on est ramené à traiter des dynamiques comme des rotations. En passant en coordonnées polaires et en exploitant la convexité et l’évolution linéaire dans le plan de phase on peut conclure sur la séparation en un nombre de pas fini entre \mathcal{R}_i et ses images par (4.25). Dans ce cas $A_i \in \mathbb{R}^{2 \times 2}$ avec une paire de pôles complexes et conjugués sur le cercle, on peut obtenir

α comme angle de rotation. \mathcal{R}_i est un polytope et on peut trouver deux directions d_1 et d_2 délimitant \mathcal{R}_i comme montré sur la figure 4.19. Les angles entre $x(0)$, δ_1 et δ_2 sont écrite comme :

$$\cos(\theta_1) = \frac{\langle x(0) - (\mathbb{I} - A_i)^{-1}a_i | \delta_1 \rangle}{\|x(0) - (\mathbb{I} - A_i)^{-1}a_i\| \|\delta_1\|}$$

et

$$\cos(\theta_2) = \frac{\langle x(0) - (\mathbb{I} - A_i)^{-1}a_i | \delta_2 \rangle}{\|x(0) - (\mathbb{I} - A_i)^{-1}a_i\| \|\delta_2\|}$$

Il existe $n^* \in \mathbb{N}$ tel que $\theta_1 - n^*\alpha < 0$ ou $\theta_2 - n^*\alpha < 0$ ce qui signifie que le calcul se termine en nombre de pas fini.

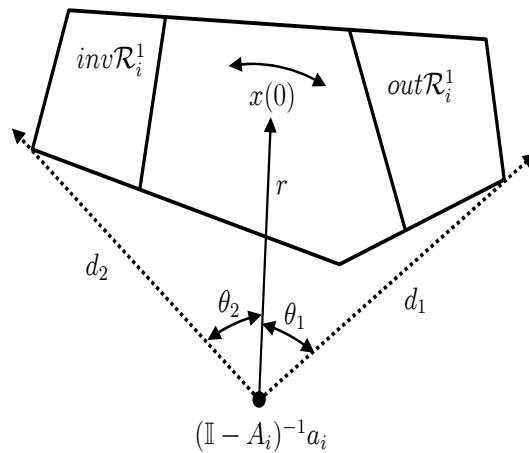


FIGURE 4.19 – L'existence d'une frontière qui délimite \mathcal{R}_i .

□

Remarque 16. L'analyse de la construction dans le cas où A_i correspond à une dynamique instable et $(\mathbb{I} - A_i)^{-1}a_i \in \mathcal{R}_i$ est difficile sans hypothèse sur la topologie de \mathcal{R}_i et la dynamique associée. La même chose peut être observée pour $|\lambda(A_i)| = 1$ cas pour lequel il est connu que la construction du *-maximal output admissible set-* produit des difficultés structurelles (nombre d'itérations infini).

Exemple 15. Cet exemple illustre la différence de calcul de l'ensemble invariant entre l'algorithme 1 et l'algorithme 7 qui utilise le graphe de transitions :

$$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u_k \quad (4.27)$$

Une loi de commande prédictive est construite avec $N = N_u = 5$, $Q = I_2$ et $R = 1$ pour l'ensemble des contraintes :

$$-20 \leq y_k \leq 20; \quad -1 \leq u_k \leq 1 \quad (4.28)$$

Une loi de commande explicite est obtenue et l'ensemble invariant de la dynamique en boucle fermée est calculé en utilisant l'algorithme 1 et la procédure utilisant le graphe de transitions, la démarche est représentée sur la figure 4.20.

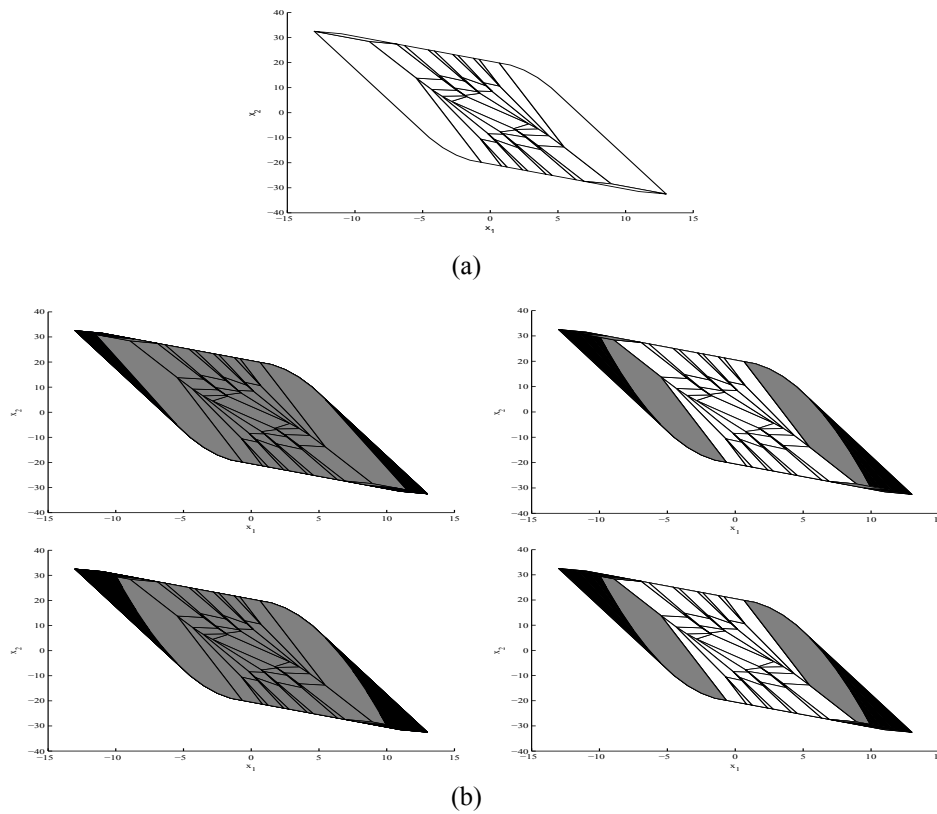


FIGURE 4.20 – (a) Loi de commande explicite. (b) Colonne de droite : deux itérations sont suffisantes pour trouver l'ensemble invariant (en couleur grise et blanche) en utilisant le graphe de transitions. les sous-régions blanches sont des sous-régions non traitées. (b) Colonne de gauche : six itérations (seulement deux sont montrées) sont nécessaires pour obtenir MPI (couleur grise) en utilisant l'algorithme de construction contractive.

4.6 Conclusion

Dans ce chapitre trois stratégies ont été présentées pour le calcul des ensembles invariant à savoir, l'algorithme contractif, l'algorithme expansif et l'algorithme basé sur les graphes de transitions. Une technique a aussi été proposée, pour améliorer le temps de calcul de ces algorithmes, en se basant sur l'analyse par intervalles. Les méthodes développées sont valables pour tout système PWA ce qui inclut le cas des systèmes linéaires en boucle fermée avec une loi de commande prédictive explicite.

Chapitre 5

Extension de la commande prédictive

5.1 Introduction

Ce chapitre basé sur les articles [60, 61, 62, 63, 64] se présente comme une extension des chapitres 2 – 4. Il sera consacré à l’application des lois de commande prédictive, avec leurs formulations analytiques, aux systèmes linéaires à paramètres incertains, aux systèmes affectés par des retards ainsi qu’aux adaptations de structure pour le suivi de trajectoire. Plus spécifiquement on s’intéressera à la version robuste des procédures décrites dans le chapitre précédent pour le calcul des ensembles invariants pour les systèmes PWA non affectés par des incertitudes.

Les points traités dans ce chapitre sont :

- la prise en compte (post-synthèse) des incertitudes de modèle dans l’application de la commande prédictive,
- le calcul des ensembles invariants pour systèmes PWA incertains,
- l’application de la commande prédictive pour systèmes linéaires affectés par un retard variable dans le signal de commande,
- l’utilisation des ensembles invariants pour garantir la faisabilité de la loi de commande pour le suivi de trajectoire.

5.2 MPC dans une formulation robuste et les ensembles invariants associés

La commande MPC est basée sur la résolution à chaque pas d’échantillonnage d’un problème d’optimisation sous contrainte sur un horizon fuyant. Cette stratégie de commande basée sur l’optimisation ne garantit pas a priori la stabilité du système en boucle fermée. En effet si le système est instable ou bien à phase non minimale, un choix défectueux des horizons de commande/prédiction peut conduire à des séquences de commande optimale pour une fonction de coût inappropriée qui fait que le système en boucle fermée est instable. La présence des contraintes rend la synthèse encore plus compliquée à cause des phénomènes d’infaisabilité (voir [65], [66] pour des exemples concrets). On a vu dans le chapitre précédent que des méthodes d’analyse des lois de commande peuvent offrir des certificats de faisabilité et décrire les régions de l’espace d’état avec certaines propriétés de convergence.

Si le modèle du système utilisé pour la prédiction de son comportement n'est pas parfait (ce qui est le cas dans la pratique), ou si le système est atteint par des perturbations, alors trouver un optimum d'une fonction de coût construite sur le modèle nominal peut ne pas être suffisant pour fournir une commande robuste pour le système réel.

En même temps, les fondements de l'optimisation imposent la faisabilité comme une donnée cruciale puisqu'elle représente le principal ingrédient pour la stabilité de toute la boucle fermée [35]. Dans ce contexte la positive invariance est importante puisqu'elle assure que si à l'instant initial la faisabilité est garantie, alors elle sera préservée pour tous les instants futurs. Par voie de conséquence, l'utilisation de contraintes terminales et la positive invariance représentent les méthodologies courantes qui garantissent une stabilité robuste.

Pour un système nominal LTI et un ensemble de contraintes :

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ Cx_k + Du_k \leq E \end{cases} \quad (5.1)$$

le point d'équilibre $\{x, u\} = \{0, 0\}$ est considéré comme admissible par rapport aux contraintes (en plus il sera considéré comme faisant partie de l'intérieur strict du domaine admissible pour éviter les problèmes de commandes sous contraintes).

On considère une fonction de coût à minimiser sur un horizon de prédiction fini et une pénalité sur l'état final. La loi de commande MPC applique la première composante de la séquence de commande optimale :

$$\begin{aligned} \min_{u_k, u_{k+1}, \dots, u_{k+N-1}} & \sum_{i=1}^{N-1} \|x_{k+i}\|_Q^2 + \sum_{i=0}^{N-1} \|u_{k+i}\|_R^2 + \|x_{k+N}\|_P^2 \\ \text{sujet à : } & \begin{cases} x_{k+i+1} = Ax_{k+i} + Bu_{k+i} \\ Cx_{k+i} + Du_{k+i} \leq E \end{cases} \end{aligned} \quad (5.2)$$

tel que N est l'horizon de prédiction, $Q = Q^T \geq 0$ et $R = R^T \succ 0$ sont les matrices de pondérations sur l'état et la commande et la norme $\|\cdot\|_2$ est employée pour garantir un optimum unique.¹

La stratégie de robustification, décrite par la suite, vise à remplacer dans une première phase le coût terminal $\|x_{k+N}\|_{P_{LQ}}^2$ par un terme quadratique correspondant à une loi stabilisante robuste par rapport à des incertitudes de modèle. La garantie de l'existence d'une région invariante sera obtenue au niveau théorique en se basant sur le principe de l'optimalité inverse (voir les travaux de [67]). Au niveau de la mise en œuvre, les adaptations des algorithmes introduites dans le cas MPC nominal seront détaillées.

5.2.1 Commande prédictive robustifiée à base de LMI

Dans le cas où le système est affecté par des incertitudes, on peut retenir le modèle de prédiction nominal comme dans (5.1) et considérer le modèle affecté par des incertitudes polytopiques :

$$x_{k+1} = A_\Delta x_k + B_\Delta u_k \quad (5.4)$$

1. Dans la construction classique, P est la solution de l'équation de Riccati :

$$P_{LQ} = Q + A^T P_{LQ} A - A^T P_{LQ} B [R + B^T P_{LQ} B]^{-1} B^T P_{LQ} A \quad (5.3)$$

et le contrôleur stabilisant correspondant est : $K_{LQ} = [R + B^T P_{LQ} B]^{-1} B^T P_{LQ} A$.

décrit par d réalisations extrêmes tel que $\exists \alpha_i \in [0; 1]$, $i = 1, \dots, d$ et :

$$\begin{cases} (A_\Delta, B_\Delta) = \sum_{i=1}^d \alpha_i (A_i, B_i) \\ \sum_{i=1}^d \alpha_i = 1. \end{cases} \quad (5.5)$$

Il est supposé que le modèle nominal se trouve à l'intérieur de cette description polytopique $(A, B) \in (A_\Delta, B_\Delta)$.

La construction de la loi de commande robuste qui stabilise (5.4) peut être formulée en considérant le pire cas (dynamiques extrêmes); à chaque pas d'échantillonnage elle peut être obtenue si la fonction de coût suivante présente un optimum borné :

$$\min_{u_k, u_{k+1}, \dots} \max_{(A_\Delta, B_\Delta)} J_\infty(k) = \min_{u_k, u_{k+1}, \dots} \max_{(A_\Delta, B_\Delta)} \sum_{i=0}^{\infty} (x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i}) \quad (5.6)$$

Il est montré dans [68] que ce problème peut être ramené à un problème de programmation semi-définie en utilisant un majorant :

$$V(x) = x^T P_{\text{LMI}} x \quad (5.7)$$

qui représente une fonction de Lyapunov commune pour le modèle polytopique (5.5). En utilisant la formulation LMI [68] la recherche de P_{LMI} se résume à la résolution de (5.8) :

$$\begin{cases} \min_{\gamma, S, Y} \gamma \\ S > I \\ \begin{bmatrix} S & S A_j^T + Y^T B_j^T & S Q^{0.5} & Y^T R^{0.5} \\ A_j S + B_j Y & S & 0 & 0 \\ Q^{0.5} S & 0 & \gamma I & 0 \\ R^{0.5} Y & 0 & 0 & \gamma I \end{bmatrix} \geq 0, \text{ pour } j = 1 \dots d \end{cases} \quad (5.8)$$

Étant données les hypothèses faites en (5.7) sur la structure de la fonction de Lyapunov, la faisabilité devient un problème sensible. Si (5.8) est faisable, alors une loi de commande robuste est donnée sous la forme d'un retour d'état :

$$K_{\text{rob}} = Y S^{-1} \quad (5.9)$$

et la fonction de Lyapunov associée est construite comme une fonction quadratique en utilisant :

$$P_{\text{rob}} = \gamma S^{-1} \quad (5.10)$$

En revenant à la synthèse MPC classique (5.2), fondée sur le modèle de prédiction nominal, la présence des incertitudes (5.4) rend la robustesse critique. En utilisant de la matrice de coût terminal P_{rob} obtenue à partir de la loi de commande stabilisante robuste basée sur les LMI (5.10), la résolution du problème :

$$\begin{aligned} & \min_{u_k, u_{k+1}, \dots, u_{k+N-1}} \sum_{i=1}^{N-1} x_{k+i}^T Q x_{k+i} + \sum_{i=0}^{N-1} u_{k+i}^T R u_{k+i} + x_N^T P_{\text{rob}} x_N \\ & \text{sujet à : } \begin{cases} x_{k+i+1} = A x_{k+i} + B u_{k+i} \\ C x_{k+i} + D u_{k+i} \leq E \end{cases} \end{aligned} \quad (5.11)$$

peut fournir un régulateur PWA explicite moins sensible aux incertitudes de modèle mais qui ne garantit pas l'existence d'un ensemble invariant (qui contient l'origine) et pour lequel les contraintes soient satisfaites. Ceci est dû au fait que dans la région centrale $K_0 \neq K_{\text{LMI}}$.

5.2.2 Réglage du MPC pour garantir la robustesse autour de l'équilibre

Considérons à nouveau le problème de commande min-max sur un horizon infini pour le système polytopique (5.5) :

$$\min_{u_k, u_{k+1}, \dots} \max_{(A_\Delta, B_\Delta)} J_\infty \quad (5.12)$$

où

$$J_\infty = \sum_{i=0}^{\infty} x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i} \quad (5.13)$$

$$u_k = K_{rob} x_k \quad (5.14)$$

où $Q \succ 0$, $R \succ 0$ sont des matrices de pondération convenables *a priori* et avec K_{rob} , le gain de retour, jouant le rôle d'argument d'optimisation.

En notant comme précédemment $P_{rob} = \gamma S^{-1}$ et $Y = K_{rob} S$, pour $S \geq I$, la loi de commande stabilisante robuste est donnée par $K_{rob} = Y S^{-1}$ où Y , S et le scalaire γ sont les solutions du problème LMI suivant :

$$\left\{ \begin{array}{l} \min_{\gamma, S, Y} \gamma \\ S > I \\ \left[\begin{array}{cccc} S & S A_j^T + Y^T B_j^T & S Q^{0.5} & Y^T R^{0.5} \\ A_j S + B_j Y & S & 0 & 0 \\ Q^{0.5} S & 0 & \gamma I & 0 \\ R^{0.5} Y & 0 & 0 & \gamma I \end{array} \right] \geq 0, \text{ pour } j = 1 \dots d \end{array} \right. \quad (5.15)$$

Le retour d'état $u_k = K_{rob} x_k$ représente une commande robuste stabilisante pour le système sans contraintes. La procédure de réglage des paramètres MPC proposée est la suivante :

1. Utiliser les matrices A_Δ , B_Δ et Q , R pour obtenir Y , S en résolvant (5.15)
2. En utilisant la paire (A, B) correspondant au modèle nominal, les matrices $\bar{Q} \geq 0$ et $\bar{R} > 0$ (et indirectement $\bar{P} = P_{rob} \geq 0$) seront construites telles que la solution optimale du problème sans contraintes (5.11) soit :

$$\mathbf{k}_u^* = \begin{bmatrix} Y S^{-1} \\ Y S^{-1} (A + B Y S^{-1}) \\ \vdots \\ Y S^{-1} (A + B Y S^{-1})^{N-1} \end{bmatrix} x_k \quad (5.16)$$

La solution (non-unique) (\bar{Q}, \bar{R}) doit satisfaire² :

$$\bar{Q} = \bar{P} - A^T \bar{P} A + \{Y S^{-1}\}^T (\bar{R} + B^T \bar{P} B) Y S^{-1} \quad (5.18)$$

2. Ce problème peut être résolu dans le cas général en employant la formulation LMI [67] :

$$\begin{array}{l} \min \alpha \\ \bar{P} - \bar{A}^T \bar{P} \bar{A} + \{Y S^{-1}\}^T (\bar{R} + \bar{G}^T \bar{P} \bar{G}) Y S^{-1} \succ 0 \\ \left[\begin{array}{c} Z \\ \bar{R} Y S^{-1} + B^T \bar{P} B Y S^{-1} + B^T \bar{P} A \\ * \\ I \end{array} \right] \succ 0 \\ Z \prec \alpha I, \quad \bar{P} \succ 0 \end{array} \quad (5.17)$$

$$\bar{R}Y S^{-1} + B^T \bar{P} B Y S^{-1} + B^T \bar{P} A = 0 \quad (5.19)$$

3. La loi de commande prédictive robuste sera obtenue en appliquant, selon le principe de l'horizon glissant la séquence de commande optimale selon (5.11).

Théorème 7. La loi de commande MPC nominale synthétisée selon la procédure décrite ci-dessus en se basant sur un index de performance obtenu par optimalité inverse en tenant compte d'une loi de commande linéaire robuste, est stabilisante avec robustesse dans un voisinage (non dégénéré) de l'origine pour le système (5.5) en présence des contraintes.

La démonstration est constructive et suit les arguments décrits dans cette section en exploitant le fait que la loi de commande dans une formulation explicite est définie sur des régions polytopiques $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i$:

$$u_k^{MPC} = F_i x_k + G_i, \text{ pour } x_k \in \mathcal{R}_i \quad (5.20)$$

et le système (5.1) peut être écrit comme un système autonome :

$$x_{k+1} = (A + B F_i) x_k + B G_i = f_{PWA}^i(x_k) \quad (5.21)$$

présentant dans la région centrale un gain statique $F_0 = K_{rob}$ et $G_0 = 0$. À ce correcteur linéaire on peut associer une fonction de Lyapunov quadratique et donc dans un voisinage de l'origine la loi de commande prédictive est stabilisante en dépit des incertitudes du modèle.

5.2.3 Ensemble invariant robuste maximal

Le problème de synthèse prédictive robuste étant résolu, on dispose d'une loi de commande supposée stabiliser le système dans une région non-dégénérée en dépit des incertitudes. La question qui se pose est : *quel est l'ensemble invariant maximal pour le système en boucle fermée ?*

Par la suite on va utiliser le fait qu'une approximation peut être obtenue par la construction de l'ensemble invariant robuste (MRPI) pour un système aff ne par morceaux (PWA) affecté d'incertitudes. En effet, un système PWA (avec incertitudes polytopiques) est obtenu par l'association du système LTI (affecté par des incertitudes polytopiques) en boucle fermée avec la loi de commande aff ne par morceaux :

$$\begin{aligned} x_{k+1} &= f_{PWA}(x_k) = (A_\Delta + B_\Delta F_i^{MPC}) x_k + G_i^{MPC} \quad \text{pour } x_k \in \mathcal{R}_i \\ (A_\Delta, B_\Delta) &\in Co\{(A_1, B_1), (A_2, B_2), \dots, (A_d, B_d)\} \end{aligned} \quad (5.22)$$

où \mathcal{R}_i sont des partitions polytopiques $\mathbb{X} = \bigcup_i \mathcal{R}_i$.

Les dynamiques liées à une réalisation extrême des incertitudes polytopiques du PWA seront décrites par :

$$\begin{aligned} x_{k+1} &= f_{PWA}^j(x_k) = (A_j + B_j F_i^{MPC}) x_k + G_i^{MPC} \\ &\text{pour } x_k \in \mathcal{R}_i, j \in \{0, 1, \dots, d\} \end{aligned} \quad (5.23)$$

L'invariance de cette loi de commande explicite sur toute la partition de l'espace d'état $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i$ n'est pas garantie, en conséquent pour une caractérisation de la région de confiance le besoin se manifeste de calculer (ou approximer) la région invariante maximale. Les algorithmes qui seront présentés par la suite suivent les principes détaillés dans le chapitre précédent en s'adaptant au cas des dynamiques PWA affectées par des incertitudes polytopiques.

Calcul du MRPI par itérations contractives

Le principe de la construction contractive est expliqué au chapitre 4 pour une dynamique PWA. La philosophie restant la même, pour les systèmes linéaires PWA incertains, on devra prendre en considération les dynamiques extrêmes de façon exhaustive.

L'algorithme 8 calcule la borne supérieure $\Phi^+ \supseteq \Phi^{\text{MRPI}}$ où Φ^{MRPI} est l'ensemble invariant maximal et robuste pour le système (5.4).

Algorithme 8 Calcul de la borne supérieure Φ^+ de l'ensemble invariant maximal Φ^{MRPI} .

ENTRÉES: Système PWA (5.4) avec d dynamiques extrêmes, et le nombre d'itérations maximal

$iterMax$

```

1:  $iter \leftarrow 0$ 
2:  $\mathcal{X}^{iter} \leftarrow \mathbb{X}$ 
3: tantque  $iter \leq iterMax$  faire
4:    $\mathcal{F} \leftarrow \mathcal{X}^{iter}$ 
5:    $L \leftarrow \text{cardinal de } \mathcal{F}$ 
6:    $i \leftarrow 1$ 
7:   pour  $j = 1$  à  $d$  faire
8:     tantque  $i \leq L$  faire
9:        $Poly \leftarrow preImage_j(\mathcal{F}(i), \mathcal{F})$  {operation qui implique l'utilisation de la dynamique
           $j$  pour le calcul de  $preImage$ }
10:       $\mathcal{F} \leftarrow \{\mathcal{F}(1, \dots, i-1), Poly, \mathcal{F}(i+1, \dots, L)\}$ 
11:       $m \leftarrow \text{cardinal de } Poly$ 
12:       $i \leftarrow i + m$ 
13:       $L \leftarrow L + m - 1$ 
14:     fin tantque
15:   fin pour
16:    $\mathcal{X}^{iter+1} \leftarrow \mathcal{F}$ 
17:   if  $\mathcal{X}^{iter+1} = \mathcal{X}^{iter}$  then
18:      $iter \leftarrow iterMax + 1$ 
19:   else
20:      $iter \leftarrow iter + 1$ 
21:   end if
22: fin tantque

```

SORTIES: $\Phi^+ \leftarrow \mathcal{X}^{iter+1}$.

Remarque 17. Les boucles 7 – 15 et 8 – 14 sont interchangeable sans changer la philosophie de la construction mais avec des résultats différents sur la convergence de l'algorithme. Une analyse des dynamiques affines et de la topologie des régions peut être utilisée pour accélérer la construction en adaptant le parcours des régions et des dynamiques.

Calcul du MRPI par itérations expansives

La différence entre l'algorithme expansif pour le calcul du MRPI et celui qui calcule selon les mêmes principes la borne inférieure du MPI, est de considérer à chaque itération tous les sous-ensembles invariants (par rapport aux dynamiques extrêmes) au lieu des sous-ensembles intermédiaires dans l'algorithme qui calcule la borne inférieure du MPI.

Algorithme 9 Calcul de la borne inférieure Φ^- de l'ensemble invariant Φ^{MRPI} .

ENTRÉES: Système PWA (5.4) avec d dynamiques extrêmes, et le nombre d'itérations maximale $iterMax$

- 1: Utiliser l'algorithme 8 pour calculer la région invariante robuste de \mathcal{R}_1 qui contient l'origine. La suite de l'algorithme part de l'hypothèse que le résultat $\mathcal{X}^1 \neq \emptyset$, sinon **fin**.
- 2: $\mathcal{S} \leftarrow \mathbb{X} \setminus \mathcal{X}^1$
- 3: $\mathcal{Y}^1 \leftarrow \mathcal{X}^1$
- 4: $L \leftarrow \text{cardinal de } \mathcal{S}$
- 5: **iter** $\leftarrow 1$
- 6: **tantque** $iter \leq iterMax$ **faire**
- 7: $\mathcal{M}_j \leftarrow \emptyset$ avec $j = 1, \dots, d$
- 8: **pour** $j = 1$ à d **faire**
- 9: $i \leftarrow 1$
- 10: **tantque** $i \leq L$ **faire**
- 11: $\mathcal{M}_j \leftarrow \mathcal{M}_j \cup \text{preImage}_j(\mathcal{S}(i), \mathcal{X}^{iter})$ {**preImage** $_j(\mathcal{F}(i), \mathcal{F})$ veut dire, utiliser la dynamique j pour le calcul de **preImage**}
- 12: $i \leftarrow i + 1$
- 13: **fin tantque**
- 14: **fin pour**
- 15: **if** $\bigcap_{j=1}^d \mathcal{M}_j = \emptyset$ **then**
- 16: **fin**
- 17: **else**
- 18: $\mathcal{X}^{iter+1} \leftarrow \bigcap_{j=1}^d \mathcal{M}_j$
- 19: $\mathcal{Y}^{iter+1} \leftarrow \mathcal{Y}^{iter} \cup \bigcap_{j=1}^d \mathcal{M}_j$
- 20: **iter** $\leftarrow iter + 1$
- 21: **end if**
- 22: **fin tantque**

SORTIES: $\Phi^- \leftarrow \mathcal{Y}^{iter}$.

En ce qui concerne le nombre d'itérations des algorithmes 8-9, les conclusions du cas nominal peuvent être reconduites. Si l'ensemble invariant maximal et robuste MRPI ne peut être calculé en temps fini, alors les deux algorithmes 8-9 peuvent être utilisés ensemble pour fournir une approximation du MRPI.

Remarque 18. : Si $\mathcal{X}^1 = \emptyset$ dans l'étape 1, l'algorithme a comme sortie $\text{MRPI} = \emptyset$ même s'il existe un ensemble invariant robuste non-vide. La construction expansive ne conduit pas vers un ensemble invariant sans un ensemble de départ qui est à son tour invariant robuste. Il mérite

d'être mentionné le fait que la méthode proposée dans la section 5.2.2 garantit la satisfaction d'une telle hypothèse.

5.3 Approches alternatives et comparaison des ensembles invariants

5.3.1 Commande prédictive basée sur les LMI

Une loi de commande prédictive peut être synthétisée par la résolution d'un problème d'optimisation basé sur les LMI (5.8). En considérant une supra-approximation des contraintes sur la commande u_k et la sortie y_k sous la forme des limitations en norme :

$$\|u_{k+i}\|_2 \leq u_{max}, \quad \|y_{k+i}\|_2 \leq y_{max}, \quad \forall i = 1, N$$

le problème de minimisation est défini dans (5.24) :

$$\begin{aligned} & \min_{\gamma, S, Y} \gamma \\ & \left\{ \begin{array}{l} \begin{bmatrix} 1 & x_k^T \\ x_k & Q \end{bmatrix} \geq 0 \\ \begin{bmatrix} S & SA_j^T + Y^T B_j^T & SQ^{0.5} & Y^T R^{0.5} \\ A_j S + B_j Y & S & 0 & 0 \\ Q^{0.5} S & 0 & \gamma I & 0 \\ R^{0.5} Y & 0 & 0 & \gamma I \end{bmatrix} \geq 0 \\ \begin{bmatrix} u_{max}^2 I & Y \\ Y^T & S \end{bmatrix} \geq 0. \\ \begin{bmatrix} S & (A_j S + B_j Y)^T C^T \\ C(A_j S + B_j Y) & y_{max}^2 I \end{bmatrix} \geq 0 \\ \text{pour } j = 1 \dots d. \end{array} \right. \end{array} \quad (5.24)$$

qui prend en compte les contraintes sur la commande et sur la sortie. Si ce problème a une solution à l'instant k , la faisabilité de la loi de commande est garantie à tous les instants futurs. Le contrôleur à chaque pas d'échantillonnage k sera donné par un retour d'état variable dans le temps :

$$u_k = K_k x_k \quad (5.25)$$

avec

$$K_k = Y Q^{-1} \quad (5.26)$$

Ensemble invariant ellipsoïdal

Dans le cas d'une commande prédictive basée sur les LMI dans (5.24), (5.25) et (5.26), il est montré que pour $u_k = K_k x_k$, alors si :

$$x_k^T Q^{-1} x_k \leq 1 \quad (5.27)$$

ou, par équivalence :

$$x_k^T P_{\text{LMI}} x_k \leq \gamma \text{ avec } P_{\text{LMI}} = \gamma Q^{-1} \quad (5.28)$$

alors

$$\max_{A_\Delta, B_\Delta} x_{k+i}^T Q^{-1} x_{k+i} < 1, \quad i \geq 1 \quad (5.29)$$

et

$$\max_{A_\Delta, B_\Delta} x_{k+i}^T P_{\text{LMI}} x_{k+i} < \gamma, \quad i \geq 1 \quad (5.30)$$

Ces ensembles de niveaux représentent des ensembles invariants ellipsoïdaux pour les dynamiques décrites par $[A_\Delta, B_\Delta]$. Considérant l'ensemble des contraintes à satisfaire par (5.1), l'ensemble maximal invariant et robuste est défini par l'union des ellipsoïdes (5.30) donnés par la paire faisable $(P_{\text{LMI}}(x), \gamma(x))$.

5.3.2 MPC désensibilisée

La formulation de la loi de commande MPC désensibilisée DMPC [69] considère la sensibilité de la trajectoire comme un facteur complémentaire dans la fonction de coût (5.2). Cette construction améliore la robustesse de la loi de commande MPC à condition que la variation du modèle soit basée sur une expression analytique de la dynamique des paramètres.

Considérons un système dynamique défini par :

$$x_{k+1} = A_\mu x_k + B_\mu u_k \quad (5.31)$$

où $\mu = [\mu_1, \dots, \mu_m] \in \mathbb{R}^m$ est le vecteur des paramètres qui transforme le modèle de prédiction en un système LPV.

On définit la variation d'une matrice donnée M par rapport à μ par :

$$\nabla_\mu M := \left[\frac{\partial M^T}{\partial \mu_1} \dots \frac{\partial M^T}{\partial \mu_m} \right]^T \quad (5.32)$$

et on considère μ_0 comme vecteur nominal des paramètres. Dans [69], la trajectoire de sensibilité p à μ_0 est définie comme :

$$p_k := \nabla_\mu x_k |_{\mu_0} \quad (5.33)$$

et

$$p_{k+1} = A_\mu x_k + A_b p_k + B_\mu u_k + B_b u_{\mu k} \quad (5.34)$$

où :

$$\begin{cases} A_\mu = \nabla_\mu A_\mu |_{\mu_0}, \\ B_\mu = \nabla_\mu B_\mu |_{\mu_0}, \\ A_b = I_m \otimes A_{\mu_0}, \\ B_b = I_m \otimes B_{\mu_0}, \\ u_{\mu k} = \nabla_\mu u_k |_{\mu_0}. \end{cases}$$

En considérant la trajectoire de sensibilité, la résolution à chaque pas d'échantillonnage du problème :

$$\begin{aligned} & \min_{u_k, u_{k+1}, \dots, u_{k+N-1}} \sum_{i=1}^N \|x_{k+i}\|_Q^2 + \|p_{k+i}\|_Q^2 + \sum_{i=0}^{N-1} \|\Delta u_{k+i}\|_R^2 + \|x_{k+N}\|_P^2 \\ \text{sujet à : } & \begin{cases} x_{k+i+1} = A|_{\mu_0} x_{k+i} + B|_{\mu_0} u_{k+i} \\ p_{k+i+1} = A_\mu x_{k+i} + A_b p_{k+i} + B_\mu u_{k+i} \\ Cx_{k+i} + Du_{k+i} \leq E \end{cases} \end{aligned} \quad (5.35)$$

et l'implémentation de la première partie de la séquence définissent une stratégie prédictive robuste par l'apparition de l'état actuel dans le problème LMI à résoudre à chaque pas k .

Dans la section suivante un exemple illustratif pour la géométrie des ensembles invariants qui peuvent être obtenus pour ces techniques de synthèse est détaillé.

5.3.3 Illustration des constructions invariants robustes

Exemple 16. Considérons le système à paramètre incertain défini par :

$$\begin{cases} x_{k+1} = \begin{bmatrix} 1 & 1 \\ 1 + \mu & 0.5 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k \\ y_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k \end{cases} \quad (5.36)$$

tel que le paramètre incertain $\mu \in [-0.1, +2.8]$, et sa valeur nominale $\mu_0 = 0$. Un simple problème de commande optimale avec un horizon de prédiction $N = 2$, $Q = I_2$ et $R = 1$ est considéré. Les contraintes à satisfaire sont :

$$\begin{cases} [-0.5, -0.5]^T \leq y_k \leq [0.5, 0.5]^T \\ -0.2 \leq u_k \leq 0.2 \end{cases} \quad (5.37)$$

et les dynamiques extrêmes sont définies par :

$$\begin{cases} x_{k+1}^1 = \begin{bmatrix} 1 & 1 \\ 0.9 & 0.5 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k \\ x_{k+1}^2 = \begin{bmatrix} 1 & 1 \\ 3.8 & 0.5 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k \end{cases} \quad (5.38)$$

Dans ce qui suit on considère ces trois stratégies de commande prédictive, à savoir MPC (la commande prédictive nominale), RMPC (commande prédictive robuste avec garantie de l'existence d'un ensemble invariant) et DMPC (la commande prédictive désensibilisée).

Stratégie MPC

Dans ce premier cas, on considère la fonction de coût dans (5.2). $P_{LQ} = \begin{bmatrix} 1.8243 & 0.6275 \\ 0.6275 & 1.6135 \end{bmatrix}$ est la solution de l'équation de Riccati (5.3) qui fournit le coût terminal correspondant au contrôleur stabilisant $K_{LQ} = -[0.8243 \ 0.6275]$ pour la dynamique nominale. Utilisant la Toolbox MPT

[33], un contrôleur explicite est obtenu avec 5 régions. Avec cette loi de commande explicite, l'algorithme contractif est utilisé pour calculer l'ensemble invariant robuste en considérant ses dynamiques extrêmes.

La figure 5.1 montre le contrôleur explicite, dont l'ensemble invariant est vide, parce que le contrôleur K_{LQ} dans la région centrale (qui contient l'origine) ne peut pas stabiliser la seconde dynamique extrême de (5.38).

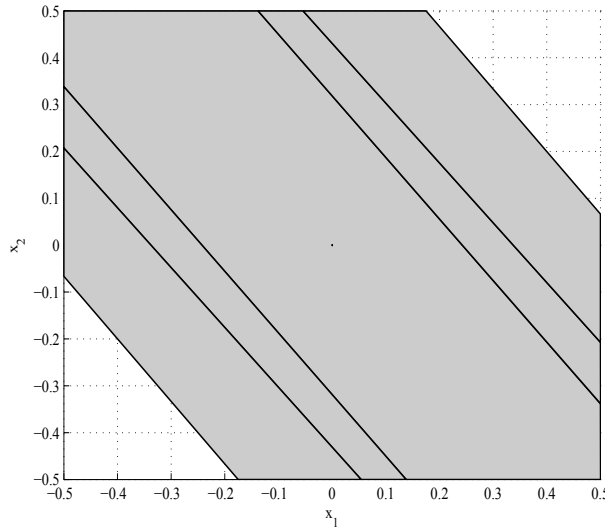


FIGURE 5.1 – Contrôleur explicite avec ensemble invariant vide.

Stratégie RMPC

On utilise la formulation LMI comme il est mentionné dans (5.8) pour calculer le contrôleur robuste qui stabilise le système avec toutes ses dynamiques. Le coût terminal $x_N^T P x_N$ est obtenu en suivant les étapes :

- calcul de $P_{LMI} = \begin{bmatrix} 29.0687 & -2.2891 \\ -2.2891 & 3.1386 \end{bmatrix}$ et de la loi de commande stabilisante

$$K_{rob} = -[1.0280 \ 0.9419]$$

qui stabilisent toutes les dynamiques du système en l'absence de contraintes,

- calcul par optimalité inverse [67] des nouvelles matrices de pondération

$$Q_{inv} = \begin{bmatrix} 0.8900 & -0.6257 \\ -0.6257 & 0.4992 \end{bmatrix} 10^{-4} \text{ et } R_{inv} = 1.8362 \cdot 10^{-7}$$

- calcul du nouveau $P_{LQ} = \begin{bmatrix} 0.8918 & -0.6240 \\ -0.6240 & 0.6658 \end{bmatrix} 10^{-4}$ (en utilisant l'équation de Riccati (5.3)) qui correspond à K_{rob} ainsi que Q_{inv} et R_{inv} ,
- calcul de la loi de commande explicite utilisant la fonction de coût (5.2).

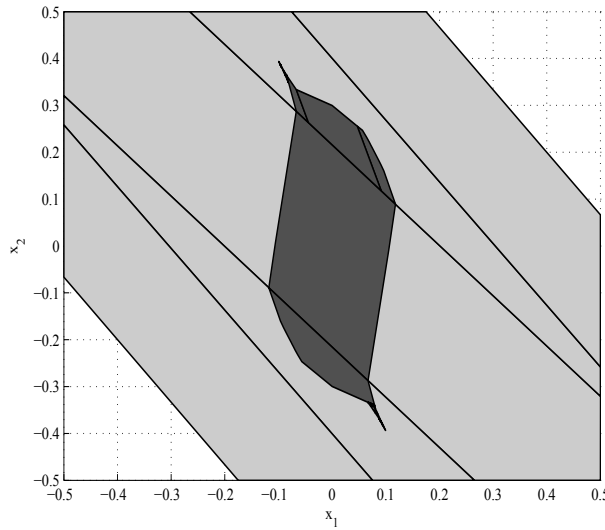


FIGURE 5.2 – Contrôleur explicite (gris clair) et l'ensemble invariant (gris foncé).

La figure 5.2 montre le contrôleur explicite et son ensemble invariant robuste correspondant.

L'ensemble ellipsoïdal (5.30) noté E_1 correspondant à K_{LMI} calculé en absence de contraintes est montré dans la figure 5.3. L'ensemble ellipsoïdal E_2 qui satisfait toutes les contraintes de la région centrale est calculé en contractant E_1 par un facteur $\gamma/10$. Ces ensembles ellipsoïdaux sont montrés dans la figure 5.3.

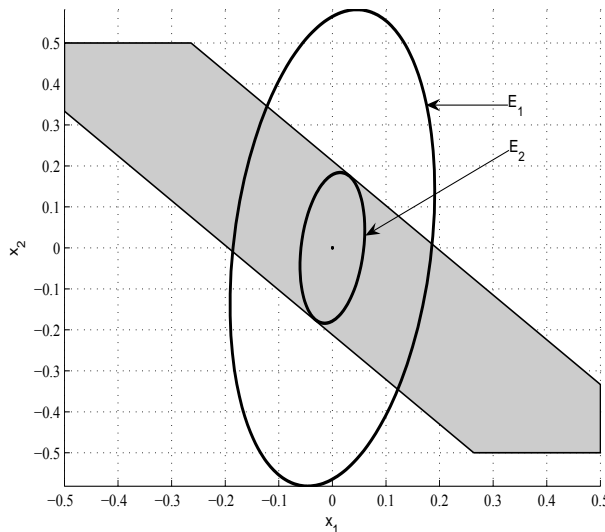


FIGURE 5.3 – Ensemble invariant ellipsoïdal qui satisfait toutes les contraintes E_2 .

La superposition des ensembles ellipsoïdaux et invariants robustes est montrée dans la figure 5.4.

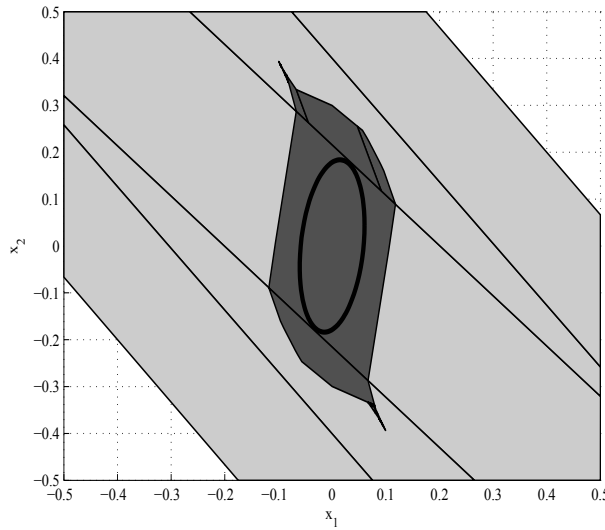


FIGURE 5.4 – Superposition de l'ensemble invariant et l'ensemble invariant ellipsoïdal.

Stratégie DMPC

Dans le cas du DMPC, la dynamique de sensibilité vis-à-vis de μ est considérée comme montré dans (5.35). Tous les coefficients de (5.34) sont :

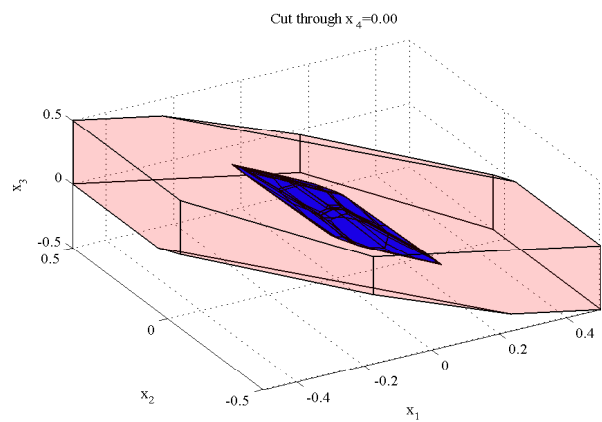
$$A_{\mu} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad A_b = \begin{bmatrix} 1 & 1 \\ 1 & 0.5 \end{bmatrix}, \quad B_{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

En suivant les mêmes étapes que pour la stratégie MPC pour calculer le contrôleur explicite et son ensemble invariant pour le système étendu (5.35), on obtient les solutions décrites dans la figure 5.5.

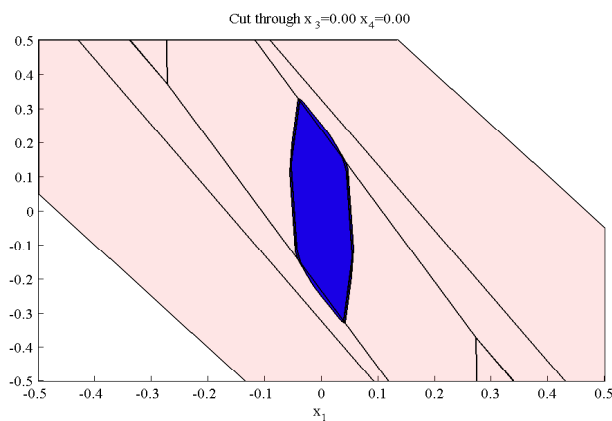
5.3.4 Application du MPC robuste pour système LTI affecté par un retard variable à l'entrée

L'interaction entre un système réel et un processus physique avec des excitations exogènes peut parfois être difficilement caractérisée comme instantanée et une des façons classiques de modéliser ces situations et ces phénomènes est l'utilisation d'un retard. Les retards (variables, distribués ou non) décrivent le couplage entre les dynamiques notamment dans le cas de propagation, les phénomènes de transport, l'hérédité et la compétition dans les dynamiques de population. Plusieurs exemples peuvent être trouvés dans [70], [71], [72]. Il y a un consensus pour considérer le *retard* comme un *paramètre critique* dans l'analyse des comportements dynamiques et des performances des systèmes.

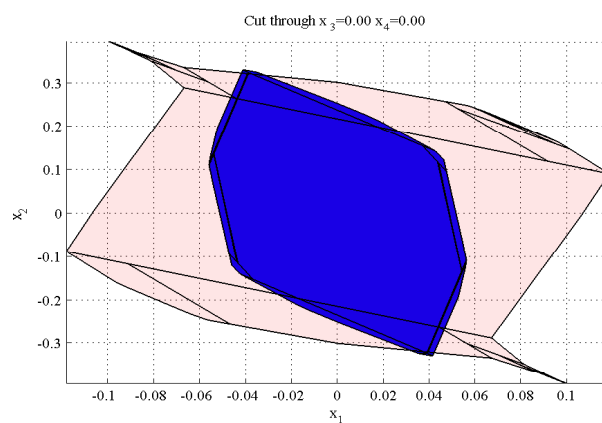
Indépendamment des problèmes mathématiques liés à la représentation, les systèmes à retard sont connus pour leurs problèmes de commande dus à l'instabilité introduite par des actions de commandes différées. Une façon naturelle pour neutraliser l'effet du temps mort est de prédire l'évolution du système avec une attention particulière quant à la sensibilité de prédiction pour des modèles instables.



(a)



(b)



(c)

FIGURE 5.5 – (a) Enveloppe convexe du contrôleur explicite (13 régions en rouge clair) et son ensemble invariant (133 régions en bleu). (b) Section à travers les dimensions $[3 \ 4]$ en considérant $p_k = 0$ pour tous les k , et comparaison entre le contrôleur explicite et l'ensemble invariant correspondant. (c) Comparaison entre l'ensemble invariant pour RMPC (en rouge clair) et le DMPC (en bleu).

La formulation temporelle de la commande prédictive basée sur modèle permet la satisfaction des contraintes dès l'étape de conception [73] et se montre très populaire pour les praticiens. Plusieurs stratégies ont été proposées afin de renforcer la stabilité du MPC [7, 8] avec comme ingrédients principaux : la fonction de coût terminale et les contraintes terminales [35].

Dans cette sous-section, on utilise la loi de commande prédictive obtenue à partir du modèle nominal. La loi de commande affine par morceaux correspondante transformera la dynamique en boucle fermée en un système affine par morceaux. Quant au retard variable, il introduit un certain degré d'incertitude qui doit être analysé pour garantir le fonctionnement global. L'existence d'un ensemble invariant non vide peut être garantie comme on a pu voir dans la section précédente. L'application de ces principes sera détaillée par la suite ainsi que l'approximation des MRPI résultantes.

5.3.5 Formulation du problème

Considérons le système linéaire LTI suivant :

$$\dot{x}(t) = A_c x(t) + B_c u(t - \tau) \quad (5.39)$$

affecté par un retard variable. La variation du retard est supposée bornée :

$$\tau \in [0, \tau_{max}] \quad (5.40)$$

Notons que le système est défini en temps discret :

$$x_k = x(kT_e) \quad (5.41)$$

où T_e est le temps d'échantillonnage. Considérons³ :

$$\begin{aligned} d &= \lceil \bar{\tau}/T_e \rceil \\ \epsilon &= dT_e - \bar{\tau} \end{aligned} \quad (5.42)$$

où $\bar{\tau}$ est la valeur probable du retard (s'il n'y a pas d'information statistique sur la distribution du retard qui permette d'extraire une valeur espérée, la valeur centrale de l'intervalle $[0, \tau_{max}]$ peut être choisie par exemple comme $\bar{\tau} = \tau_{max}/2$). Maintenant on peut décrire le modèle LTI discret affecté par le retard :

$$x_{k+1} = Ax_k + Bu_{k-d} - \bar{\Delta}(u_{k-d} - u_{k-d+1}) \quad (5.43)$$

avec

$$A = e^{A_c T_e} \quad (5.44)$$

$$B = \int_0^{T_e} e^{A_c(T_e-\theta)} B_c d\theta \quad (5.45)$$

$$\bar{\Delta} = \int_{T_e-\epsilon}^{T_e} e^{A_c(T_e-\theta)} B_c d\theta = \int_{-\epsilon}^0 e^{-A_c\theta} B_c d\theta \quad (5.46)$$

3. L'opérateur $\lceil x \rceil$ approxime l'élément x par l'entier le plus proche en direction de $+\infty$.

obtenues en supposant que la commande est maintenue constante entre deux instants d'échantillonnage : $u(t) = u_k, \forall t \in [t_k, t_{k+1})$.

Ce modèle LTI discret (5.43) sera considéré par la suite comme modèle nominal.

Pour tenir compte de l'incertitude due au retard variable qui se traduit par la variation de la limite d'intégration ϵ , on va considérer le modèle polytopique (qui correspond à une variation $0 \leq \epsilon \leq T_e$) :

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_{k-v} - \Delta(u_{k-v} - u_{k-v+1}) \\ \Delta &\in \Omega_x \\ \Omega_x &= Co\{\Delta_0, \Delta_1, \dots, \Delta_n\} \\ v &\in \{0, 1, 2, \dots, h\} \end{aligned} \quad (5.47)$$

où v est une variable entière qui décrit le retard en nombre de pas d'échantillonnage. La valeur maximale du retard est :

$$h = \left\lceil \frac{\tau_{max}}{T_e} \right\rceil \quad (5.48)$$

et les réalisations extrêmes Δ_i sont obtenues en utilisant la forme canonique de Jordan pour la matrice A_c dans (5.46), calculée pour les valeurs extrêmes de ϵ , à savoir 0 et τ_{max} . On peut montrer qu'en utilisant la décomposition de Jordan et l'exponentielle matricielle de chaque bloc de Jordan résultant, le nombre minimum des sommets caractérisant l'incertitude polytopique pour le système original à retard variable est $n + 1$ où n est la dimension du système d'état [74].

En utilisant une description dans un espace d'état étendu fondée sur l'équation (5.43), on obtient le modèle de prédiction nominal :

$$\xi_{k+1} = \bar{F}\xi_k + \bar{G}u_k \quad (5.49)$$

En effet en définissant :

$$\xi_k^T = [x_k^T \quad u_{k-h}^T \quad u_{k-h+1}^T \quad \dots \quad u_{k-d}^T \quad u_{k-d+1}^T \quad \dots \quad u_{k-1}^T] \quad (5.50)$$

les matrices \bar{F} , \bar{G} sont données par les équations (5.51-5.52).

Pour le même vecteur d'état ξ_k , on peut décrire la variation structurelle que provoque la variation du retard par l'existence d'un modèle affecté d'une incertitude paramétrique :

$$\begin{aligned} \xi_{k+1} &= F\xi_k + Gu_k \\ (F, G) &\in \Omega_\xi \\ \Omega_\xi &= Co\{(F_1, G_1), (F_2, G_2), \dots, (F_s, G_s)\} \end{aligned} \quad (5.55)$$

En utilisant A , B et l'incertitude polytopique pour la matrice Δ , avec les réalisations extrêmes $\Delta_i, i = \{1, \dots, n\}$, les matrices de transition seront données par (5.53-5.54).

Les variables $\delta_i, i = 1, \dots, h$ sont telles que :

$$\sum_{i=1}^h \delta_i = 1; \quad \delta_i \in \{0, 1\} \quad (5.56)$$

Le nombre de réalisations extrêmes est $s = h(n + 1)$ (dû au fait qu'il y a h combinaisons de variables logiques admissibles). Cependant, $n - 1$ de ces réalisations, correspondant à $\Delta = 0$

$$\bar{F} = \left[\begin{array}{c|c|c|c} & \overbrace{0 \ 0 \ \dots \ 0}^{h-d} & B - \bar{\Delta} \ \bar{\Delta} & \overbrace{0 \ \dots \ \dots \ 0}^{d-2} \\ \hline A & & & \\ \hline \left. \begin{array}{l} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{array} \right\} h-d & \begin{array}{ccc} 0 & I_m & \ddots \\ & \ddots & \ddots & 0 \\ & & \ddots & I_m \\ & & & 0 \end{array} & \begin{array}{cc} 0 & 0 \\ & \ddots \\ 0 & \ddots \\ I_m & 0 \end{array} & \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \\ \hline \vdots & \ddots & \begin{array}{cc} 0 & I_m \\ \ddots & 0 \end{array} & \begin{array}{cccc} 0 & \ddots & \ddots & \ddots \\ I_m & 0 & \ddots & \ddots \end{array} \\ \hline \left. \begin{array}{l} \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{array} \right\} d-2 & \dots & \dots & \begin{array}{ccc} 0 & I_m & \ddots \\ & \ddots & \ddots & 0 \\ & & \ddots & I_m \\ & & & 0 \end{array} \end{array} \right] ; \quad (5.51)$$

$$\bar{G} = [0 \ | \ 0 \ \dots \ \dots \ \dots \ | \ 0 \ 0 \ | \ \dots \ \dots \ 0 \ I_m]^T \quad (5.52)$$

$$F = \left[\begin{array}{cccccccc} A & \delta_h(B - \Delta) & \delta_{h-1}(B - \Delta) + \delta_h \Delta & \delta_{h-2}(B - \Delta) + \delta_{h-1} \Delta & \dots & \dots & \delta_1(B - \Delta) + \delta_2 \Delta & \\ 0 & 0 & I_m & 0 & \dots & \dots & 0 & \\ \vdots & \ddots & 0 & I_m & \ddots & \ddots & \vdots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & I_m & \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & \end{array} \right] ; \quad (5.53)$$

$$G = [\delta_1 \Delta \ 0 \ \dots \ \dots \ 0 \ I_m] \quad (5.54)$$

sont, en fait, générées par les combinaisons des variables logiques des blocs avoisinant telles que la complexité globale du modèle polytopique est donnée par $s = nh + 1$ réalisations extrêmes.

L'évolution du système satisfera les limitations physiques sur la commande et les restrictions sur la région accessible dans l'espace d'état. Leurs descriptions seront regroupées dans un ensemble d'inégalités linéaires :

$$C\xi_k \leq W \quad (5.57)$$

Dans ce qui suit, l'objectif de la commande est la régulation de l'état ξ_k vers l'origine en satisfaisant les contraintes (5.57) et en utilisant une séquence de commande optimale sur un horizon glissant. Le critère de performance est construit de telle sorte que la loi de commande obtenue garantisse, pour une région non vide au voisinage de l'origine, un comportement robuste en présence d'un retard variable.

Commande prédictive

La stratégie de commande MPC, pour le système à retard considéré ici, construit à chaque pas d'échantillonnage k la séquence de commande optimale :

$$\mathbf{k}_u^* = \{u(k|k), \dots, u(k+N-1|k)\} \quad (5.58)$$

pour l'index de performance

$$\mathbf{J} = \left. \begin{aligned} & \arg \min_{\{u(k|k), \dots, u(k+N-1|k)\}} \{ \xi(k+N|k)^T \bar{P} \xi(k+N|k) \\ & + \sum_{j=1}^N \xi(k+j|k)^T \bar{Q} \xi(k+j|k) + \sum_{j=0}^{N-1} u(k+j|k)^T \bar{R} u(k+j|k) \} \end{aligned} \right\} \quad (5.59)$$

tel que :

$$\begin{cases} \xi(k+j+1|k) = \bar{F} \xi(k+j|k) + \bar{G} u(k+j|k) \\ C \xi(k+j|k) \leq W; \quad j = 1, \dots, N-1 \end{cases}$$

qui évalue la dynamique du système sur un horizon fini $k+1, \dots, k+N$. Il est à remarquer que l'horizon de prédiction devrait être plus grand que le retard : $N \geq d$, pour permettre la propagation des commandes et donc une influence directe sur la sortie du système. Sachant que la prédiction est construite selon un modèle nominal, mais que le système réel peut être affecté par un retard allant jusqu'à h échantillons, on considérera $N \geq h$ pour traiter toutes les variations possibles.

La première composante de \mathbf{k}_u^* est la commande appliquée au système :

$$u(k) = \mathbf{k}_u^*(1) = u(k|k) \quad (5.60)$$

Utilisant les nouvelles mesures, le nouveau vecteur de commande est re-calculé par optimisation, ainsi une commande en boucle fermée est obtenue.

La construction de la loi de commande prédictive sera influencée par le choix de l'horizon de prédiction N , les facteurs de pondérations sur l'état $\bar{Q} = \bar{Q}^T \succ 0$ et sur la commande $\bar{R} = \bar{R}^T \succ 0$. Pour la pénalisation de l'état terminal, la matrice \bar{P} est généralement construite telle que l'horizon de prédiction puisse être étendu à l'infini par l'introduction du terme $\xi(k+N|k)^T \bar{P} \xi(k+N|k)$ dans (5.59). En utilisant la procédure de synthèse discutée dans la sous-section 5.2.2 on obtient les matrices \bar{Q} , \bar{R} et indirectement \bar{P} telles que la loi de commande calculée présente un certain degré de robustesse vis-à-vis de la variation du retard.

Le problème d'optimisation dans (5.59) peut être réécrit sous une forme compacte :

$$\begin{aligned} \mathbf{k}_u^*(\xi(k)) &= \arg \min_{\mathbf{k}_u} 0.5 \mathbf{k}_u^T H \mathbf{k}_u + \mathbf{k}_u^T G \xi(k) \\ \text{sujet à : } & A_{in} \mathbf{k}_u \leq b_{in} + B_{in} \xi(k) \end{aligned} \quad (5.61)$$

où ξ_k est le vecteur des paramètres. La fonction de coût est convexe et l'optimum est unique si $H \succ 0$, l'ensemble des contraintes linéaires décrit des polyèdres paramétrés tel que l'optimum sous contraintes peut être décrit par la combinaison des sommets paramétrés [22]. La solution

explicitite pour la loi MPC peut être obtenue en retenant la première composante de $\mathbf{k}_u^*(\xi(k))$, ainsi la commande prédictive est exprimée en terme d'une loi de commande rétroactive affine par morceaux :

$$u(k) = K_i^{\text{MPC}} \xi + \kappa_i^{\text{MPC}}, \quad \text{avec } i \text{ tel que } \xi \in \mathcal{R}_i, \quad (5.62)$$

pour \mathcal{R}_i , régions polyédrales dans \mathbb{R}^{n+hm} .

Le modèle de prédiction est linéaire, l'origine est un point faisable (dans le cas général placé à l'intérieur du domaine faisable) et représente un point d'équilibre pour le système (5.49). Les problèmes (5.59) et (5.61) sont faisables et de plus l'optimum associé sera sans contraintes. La conséquence directe est que la loi de commande affine correspondant à la région \mathcal{R}_{i_0} contenant l'origine ($0 \in \mathcal{R}_{i_0}$) n'a plus de partie affine ($\kappa_{i_0}^{\text{MPC}} = 0$) et la loi de commande locale correspond à la commande robuste sans contraintes ($K_{i_0}^{\text{MPC}} = K_{rob}$). Si l'ensemble MRPI pour le système en boucle fermée ne peut pas être déterminé en un nombre fini d'itérations, une construction itérative par procédure duale expansive/contractive peut être utilisée pour obtenir une approximation garantie.

Illustration pour un processus physique

L'exemple numérique suivant illustre la procédure de calcul d'une loi de commande prédictive robuste ainsi que son ensemble invariant robuste.

Exemple 17. Considérons le procédé défini dans la référence [75] dont l'objectif est la commande du niveau dans une cuve, le schéma-bloc est présenté dans la figure 5.6.

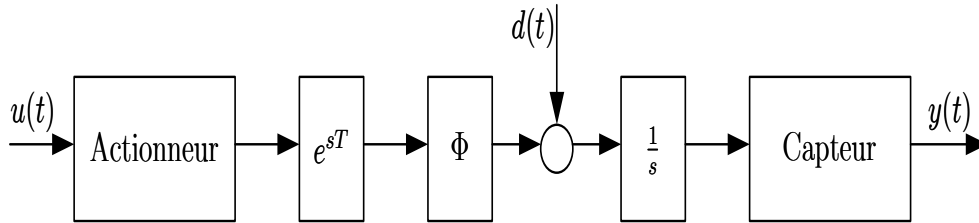


FIGURE 5.6 – Schéma général du procédé à contrôler.

Le système continu à contrôler est considéré comme un double intégrateur avec un retard variable :

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t - \tau), \quad \tau \in [0s, 0.3s] \quad (5.63)$$

Le modèle discret avec $T_e = T_e = 0.1s$ est donné par :

$$x_{k+1} = \begin{bmatrix} 1 & 0 \\ 0.1 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.1 \\ 0.05 \end{bmatrix} u_{k-i} - \Delta(u_{k-v} - u_{k-v+1}) \quad (5.64)$$

avec $v \in \{0, 1, 2\}$

Dans un premier temps on calcule la matrice Δ qui englobe le facteur incertain. Du fait que dans la représentation originale, le vecteur d'état x_k est à deux dimensions, l'incertitude polytopique sera :

$$\Delta \in Co \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.05 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0.013 \end{bmatrix} \right\} \quad (5.65)$$

Dans la représentation d'état étendue, le modèle nominal obtenu pour $\bar{\tau} = 0.3$ est :

$$\xi_{k+1} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0.1 & 1 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xi_k + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_k \quad (5.66)$$

Une commande par retour d'état robuste et stabilisante est obtenue pour le cas sans contraintes par la résolution du problème LMI (5.15) :

$$K = [-1.3188 \quad -0.5408 \quad -0.1292 \quad -0.0157 \quad -0.1511] \quad (5.67)$$

Le problème d'optimalité inverse nous mène après la résolution de (5.17) au réglage de la loi MPC nominale avec les matrices de pondération définies en (5.68) :

$$R = 1,$$

$$Q = \begin{bmatrix} 1.5204 & 0.6985 & -0.0210 & 0.0247 & 0.1837 \\ 0.6985 & 0.5769 & 0.1856 & 0.1393 & 0.1024 \\ -0.0210 & 0.1856 & 0.4259 & 0.0022 & -0.1953 \\ 0.0247 & 0.1393 & 0.0022 & 0.1293 & 0.0488 \\ 0.1837 & 0.1024 & -0.1953 & 0.0488 & 0.2139 \end{bmatrix}, \quad (5.68)$$

$$P = \begin{bmatrix} 27.2258 & 9.1962 & 2.5071 & 2.5913 & 2.4945 \\ 9.1962 & 7.0818 & 1.0028 & 1.1254 & 1.0667 \\ 2.5071 & 1.0028 & 0.6747 & 0.2539 & 0.0309 \\ 2.5913 & 1.1254 & 0.2539 & 0.8035 & 0.2981 \\ 2.4945 & 1.0667 & 0.0309 & 0.2981 & 0.9723 \end{bmatrix}.$$

En imposant un ensemble de contraintes sur la commande et sur l'état :

$$\begin{aligned} -0.1 &\leq u_k \leq 0.1 \\ \begin{bmatrix} -2 \\ -2 \end{bmatrix} &\leq x_k \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix} \end{aligned} \quad (5.69)$$

et en tenant compte que le retard maximal est de 3 instants d'échantillonnages, on choisit un horizon de prédiction $N = 5$ afin d'avoir une solution explicite moins complexe (47 régions dans la partition de l'espace d'état, voir la figure 5.7). Néanmoins, il est à mentionner qu'en général le domaine de faisabilité est d'autant plus grand que l'horizon de prédiction est grand.

Le modèle polytopique dans la représentation d'état étendue qui englobe (en utilisant 7 réalisations extrêmes) le système à retard variable va nous permettre d'employer la procédure contractive pour l'approximation de l'ensemble invariant maximal. Dans la figure 5.8 l'approximation du MRPI est obtenue après 5 itérations.

Finalement dans la figure 5.9 une simulation temporelle avec variation du retard est présentée (l'état initial est $(0; -2)$).

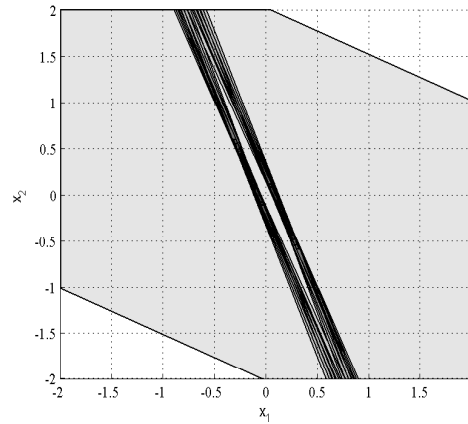


FIGURE 5.7 – Projection de la partition de la solution explicite sur les deux premières composantes de l'espace d'état étendu.

5.4 Problèmes de faisabilité dans le suivi de trajectoire

Dans ce qui suit, on montre comment on peut garantir la faisabilité de la loi de commande prédictive dans la poursuite de trajectoire.

Considérons le système LTI suivant :

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Hx_k \end{aligned} \quad (5.70)$$

avec $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ et $y_k \in \mathbb{R}^q$, l'ensemble des contraintes linéaires décrites par une région polytopique contenant l'origine :

$$Z = \{[x_k, u_k]^T | Cx_k + Du_k \leq E\} \quad (5.71)$$

et une loi de commande explicite correspondant à une formulation de commande prédictive. Cette loi explicite est exprimée par une loi de commande affine par morceaux PWA. Si la contrainte terminale n'est pas renforcée pour garantir la positive invariance de la commande en boucle fermée, alors les algorithmes contractifs/expansifs peuvent être utilisés pour trouver une approximation de l'ensemble maximal positif invariant.

Dans le contexte de suivi de trajectoire, on cherche à ce que la sortie y_k du système (5.70) suive une trajectoire de référence donnée. Dans ce contexte, une séquence complète de points de référence est disponible a priori au moins pour une fenêtre de prédiction de longueur N . Le principe de l'horizon fuyant (voir la figure 5.10 pour illustration) permet d'utiliser les valeurs précédentes et de mettre à jour seulement la dernière composante.

En écrivant la trajectoire de référence dans une forme compacte pour la fenêtre de prédiction on obtient :

$$Y^{ref} = [w_1^T w_2^T \cdots w_N^T]^T \quad (5.72)$$

avec $w_k \in \mathbb{R}^q$ et $Y^{ref} \in \mathbb{R}^{qN}$.

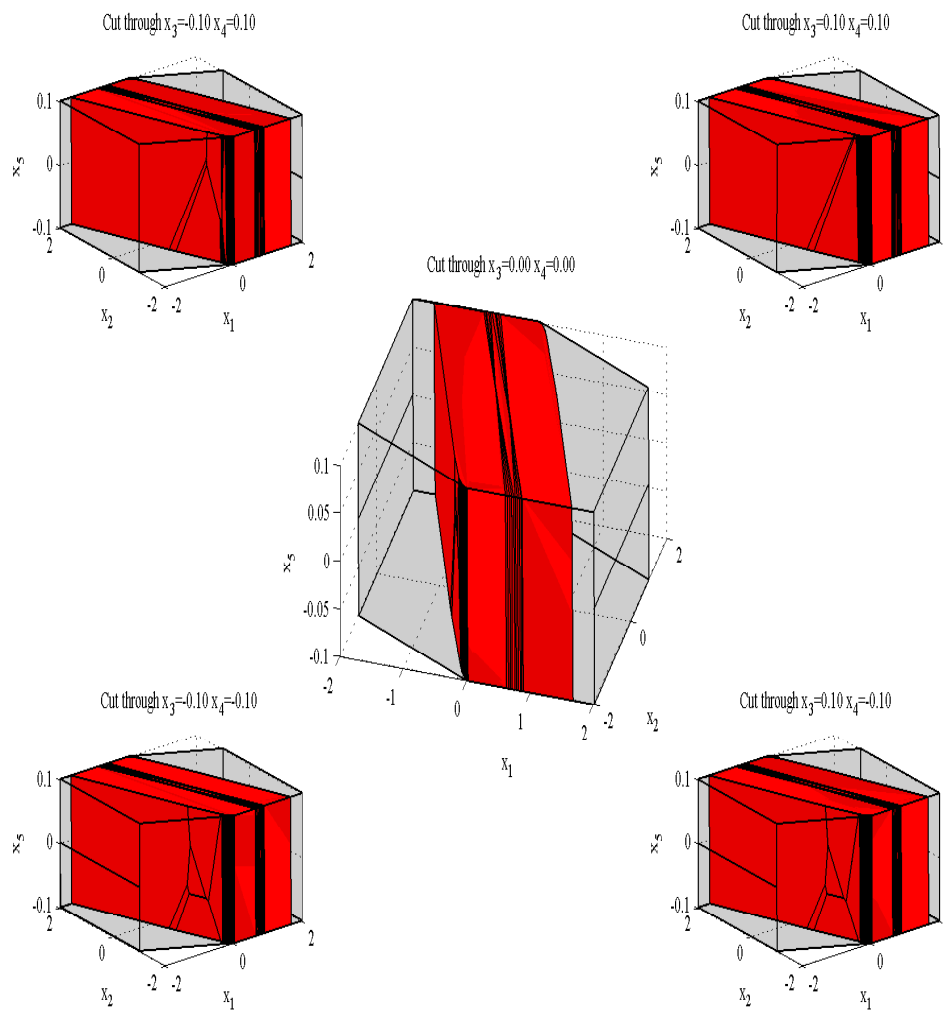
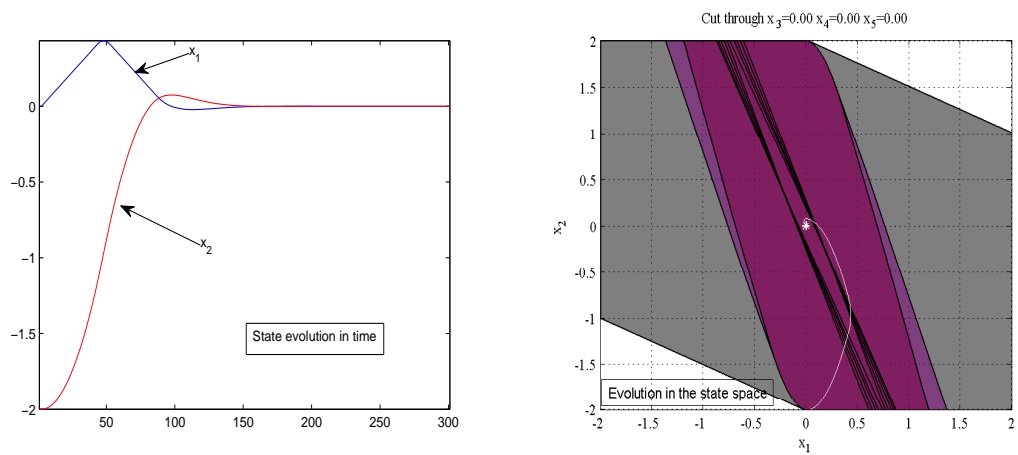
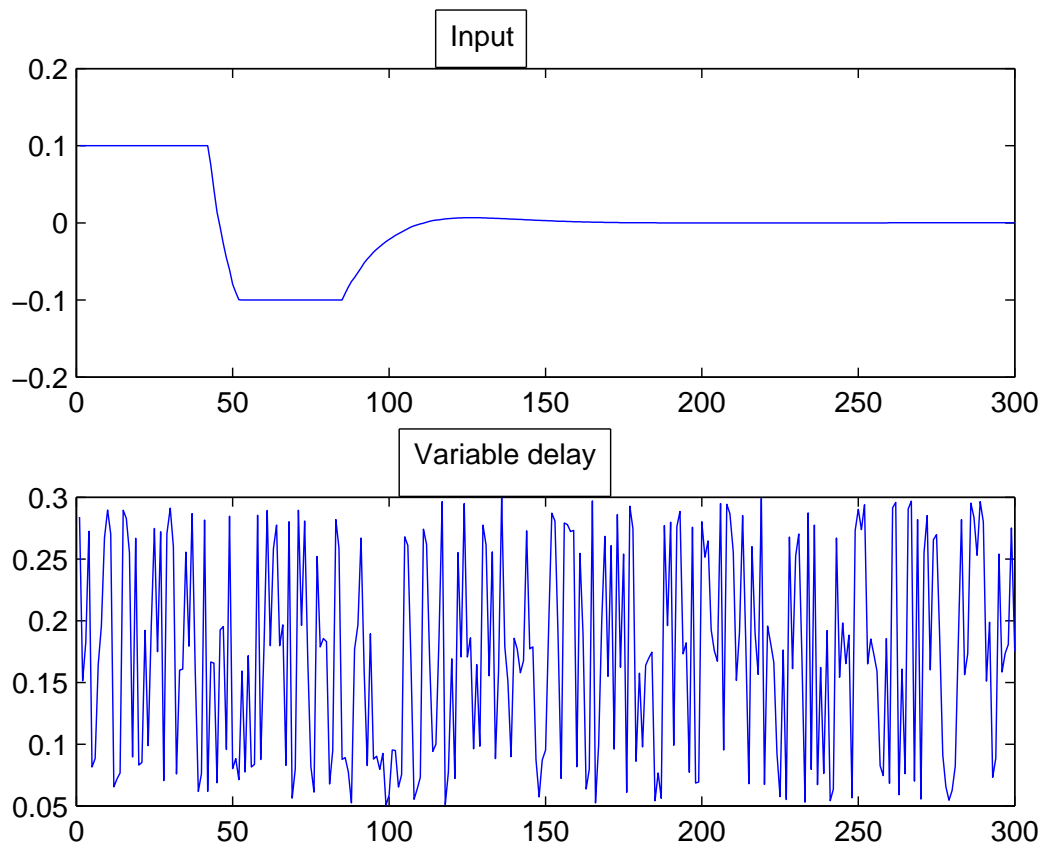


FIGURE 5.8 – La partition de la solution explicite et l'approximation de l'ensemble MRPI.



(a)



(b)

FIGURE 5.9 – (a) Évolution temporelle de l'état du système. (b) Le signal de commande et la variation du retard.

Sachant que les prédictions considèrent le principe de la fenêtre fuyante, on est intéressé par

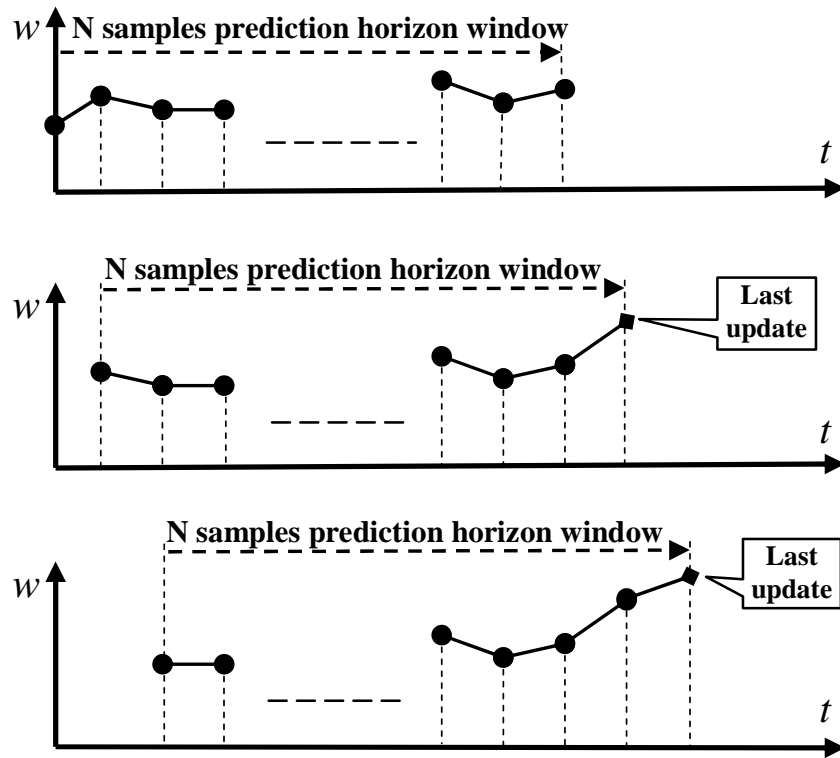


FIGURE 5.10 – Évolution des références de poursuite $w = [w_1^T w_2^T \cdots w_N^T]^T$ par le principe de l'horizon fuyant.

l'équation d'actualisation pour Y^{ref} , deux cas extrêmes sont à distinguer dans ce qui suit :

1. **Actualisation du dernier point de référence :** En considérant seulement le dernier point de référence, w_N est actualisé durant la poursuite de trajectoire, une équation d'évolution est utilisée pour le signal de référence pour la fenêtre de prédiction :

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_N \end{bmatrix}_k + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \nu \end{bmatrix} \quad (5.73)$$

2. **Actualisation de la fenêtre :** Ce deuxième cas est considéré quand tous les points de référence dans (5.72) sont actualisés sans connaître a priori leurs valeurs :

$$\begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_N \end{bmatrix} \quad (5.74)$$

Remarque 19. Les variables ν, ν_1, \dots, ν_N prennent en compte l'actualisation de la référence. En exploitant l'information a priori sur la trajectoire à suivre on imposera des bornes dérivées à

leur tour des contraintes sur les états et/ou les sorties du système. En effet, si la sortie du système est soumise à une contrainte $y_k \in Y$ alors il est naturel d'imposer les mêmes contraintes sur l'actualisation de la référence $\nu \in Y$.

Remarque 20. Les variables ν, ν_1, \dots, ν_N peuvent être interprétées comme des perturbations exogènes et la loi de commande MPC peut être calculée pour qu'elle soit robuste vis-à-vis de leur influence, pour minimiser ainsi l'influence d'éventuelles variations. Dans ce qui suit on se réfère à des approches qui tiennent compte de l'information a priori et assurent une bonne réactivité.

Seulement le premier cas de l'actualisation de Y^{ref} sera traité (le deuxième cas suit le même principe). L'état du système (5.70) est augmenté pour considérer le problème de poursuite :

$$\begin{bmatrix} x_{k+1} \\ w_{k+2} \\ \vdots \\ w_{k+N} \\ w_{k+N+1} \end{bmatrix} = \begin{bmatrix} x \\ w_1 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}_{k+1} = \begin{bmatrix} A & 0 & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ * & * & * & \cdots & * \end{bmatrix} \begin{bmatrix} x \\ w_1 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}_k + \begin{bmatrix} B \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u(k) \quad (5.75)$$

Le symbole * signifie que ces valeurs n'influencent pas la conception de la loi de commande. En effet, si l'équation de sortie peut être modifiée pour exprimer l'erreur de poursuite :

$$\epsilon_k = \begin{bmatrix} C & -I & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_k \\ w_1 \\ \vdots \\ w_N \end{bmatrix} \quad (5.76)$$

alors, le problème de la poursuite de référence pour Y^{ref} est équivalent avec le problème de régulation pour le nouveau vecteur de sortie ϵ_k et il est régulé vers zéro par une formulation MPC. Il est à noter que de par les contraintes sur l'état et la sortie du système, on peut imposer une contrainte de précision sur la poursuite comme :

$$\underline{\epsilon} \leq \begin{bmatrix} y_1^T & \cdots & y_N^T \end{bmatrix}_k^T - Y_k^{ref} \leq \bar{\epsilon} \quad (5.77)$$

laquelle peut être réécrite en termes d'inégalité linéaire en les arguments d'optimisation et le vecteur des paramètres (l'état courant et l'ensemble des points futurs).

En suivant la philosophie prédictive, à chaque pas d'échantillonnage une séquence de commande optimale en boucle ouverte :

$$U^* = \begin{bmatrix} u(k|k)^T & u(k+1|k)^T & \cdots & u(k+N-1|k)^T \end{bmatrix}^T \quad (5.78)$$

est calculée en résolvant le problème d'optimisation suivant :

$$U^* = \arg \min_U \sum_{i=0}^{N-1} \left[\epsilon(k+i+1|k)^T Q \epsilon(k+i+1|k) + u(k+i|k)^T R u(k+i|k) \right] \quad (5.79)$$

où seulement la première composante de la commande optimale U^* est appliquée au système (5.70) et un nouveau problème d'optimisation est formulé et résolu par le principe de l'horizon fuyant.

Avec la notation de la partie inconnue de l'état augmenté :

$$z_k = [x^T \quad w_1^T \quad \cdots \quad w_N^T]_k^T \quad (5.80)$$

le problème d'optimisation dans MPC peut être reformulé :

$$\arg \min_U U^T H U + U^T F z + z^T G z \quad (5.81)$$

sous :

$$Q U \leq W + E z$$

et sa solution est représentée par une fonction continue et affine par morceaux :

$$U^*(z) = F_i z + G_i, \text{ pour } z \in \mathcal{R}_i \quad (5.82)$$

où \mathcal{R}_i sont des régions polyédrales dans \mathbb{R}^{n+qN} et seulement les premières composantes de F_i et G_i sont utilisées pour déterminer la commande à appliquer au système :

$$u(z) = F_i^{(1)} z + G_i^{(1)}. \quad (5.83)$$

En écrivant la description de la dynamique en boucle fermée, on obtient :

$$x_{k+1} = A x_k + B F_i^{(1,x)} x_k + B F_i^{(1,w)} [w_1^T \quad \cdots \quad w_{N-1}^T]_k^T + B G_i^{(1)} \quad (5.84)$$

La résolution du problème d'optimisation (5.81) sans contraintes terminales à la fin de prédiction, ne garantira pas sa faisabilité à n'importe quel moment. Pour surmonter ce problème on définit une référence de poursuite faisable FTR comme un domaine de sécurité dans l'espace de référence $Y^{ref} \in \mathbb{R}^{qN}$ qui garantit la faisabilité du contrôleur explicite (5.83).

La construction de la référence de poursuite faisable est similaire à la construction des ensembles invariants détaillés au chapitre 4, mais la partie inconnue de l'évolution de Y^{ref} change la façon dont le calcul a exploité des opérations de projections pour extraire l'information sur l'actualisation du domaine faisable.

Le problème de poursuite est équivalent au calcul des ensembles invariants pour système PWA. Le but de l'analyse de faisabilité est d'obtenir un ensemble de points admissibles $w \in W(z_k)$. $W(z_k)$ sera obtenu par la projection de la solution explicite sur w_N . Structurellement, l'ensemble $W(z_k)$ est un polyèdre paramétré [76]. L'algorithme 10 décrit le principe de cette construction en utilisant les opérateurs :

$$\text{ImaProj}(\mathcal{R}_i) = \left\{ \zeta \in \mathbb{R}^{n+q(N-1)} \mid \exists z \in \mathcal{R}_i \text{ tel que :} \right. \\ \left. \zeta = \left(\begin{bmatrix} A & 0 & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \end{bmatrix} - \begin{bmatrix} B \\ 0 \\ \vdots \\ 0 \end{bmatrix} F_i \right) z + \begin{bmatrix} B \\ 0 \\ \vdots \\ 0 \end{bmatrix} G_i \right\} \quad (5.85)$$

$$\text{preImaProj}(\mathcal{D}, \mathcal{R}_i) = \left\{ \begin{array}{l} z \in \mathcal{R}_i | \exists \zeta \in \mathcal{D} \subset \mathbb{R}^{n+q(N-1)} \text{ tel que :} \\ \zeta = \left(\left[\begin{array}{cccc} A & 0 & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \end{array} \right] - \left[\begin{array}{c} B \\ 0 \\ \vdots \\ 0 \end{array} \right] F_i \right) z + \left[\begin{array}{c} B \\ 0 \\ \vdots \\ 0 \end{array} \right] G_i \end{array} \right\} \quad (5.86)$$

Algorithme 10 Construction de références de poursuite faisables.

ENTRÉES: Système PWA défini sur une partition polyédrale $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i$ et le nombre d'itérations maximal $iterMax$

- 1: $dim \leftarrow n + q(N - 1)$
- 2: $PR \leftarrow \text{Projection}_{dim}(\mathbb{X})$
- 3: **tantque** $iter \leq iterMax$ **faire**
- 4: $L \leftarrow \text{cardinal}(\mathbb{X})$
- 5: $i \leftarrow 1$
- 6: **tantque** $i \leq L$ **faire**
- 7: $\mathcal{F} \leftarrow \emptyset$
- 8: $\mathcal{F} \leftarrow \mathcal{F} \cup (\text{preImaProj}(\text{ImaProj}(\mathcal{R}_i), PR))$
- 9: $m \leftarrow \text{cardinal}(\mathcal{F})$
- 10: $i \leftarrow i + m$
- 11: $\mathbb{X} \leftarrow \{\mathcal{R}_1, \dots, \mathcal{R}_{i-1}, \mathcal{F}, \mathcal{R}_{i+1}, \dots, \mathcal{R}_N\}$
- 12: $\mathcal{F} \leftarrow \text{projection}_{dim}(\mathcal{F})$
- 13: $PR \leftarrow \{PR_1, \dots, PR_{i-1}, \mathcal{F}, PR_{i+1}, \dots, PR_N\}$
- 14: $N \leftarrow \text{cardinal}(\mathbb{X})$
- 15: **fin tantque**
- 16: **fin tantque**

SORTIES: Ensemble de références de poursuite pour préserver la faisabilité.

L'ensemble des références qui préservent la faisabilité offre à chaque pas d'échantillonnage la description de régions faisables pour les futures références. Cet ensemble peut être utilisé pour préserver la faisabilité même si la référence actuelle est en dehors de la région faisable. Par l'implémentation du mécanisme du gouverneur de référence [77], on peut ajuster la trajectoire de référence pour toujours garantir la faisabilité.

Exemple 18. Pour le problème de poursuite de trajectoire, il est important de connaître a priori et à chaque pas d'échantillonnage l'ensemble faisable auquel les futures références à poursuivre devraient appartenir. Pour illustrer comment le domaine de faisabilité est utilisé pour choisir les futures références, on considère le système LTI suivant :

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} -1 \\ 2 \end{bmatrix} u_k; \quad y_k = [1 \quad 0] x_k \quad (5.87)$$

Une solution explicite pour une loi MPC avec $N = 2$, $Q = I_2$ et $R = 1$ est construite pour l'ensemble des contraintes :

$$-15 \leq y_k \leq 15; \quad -20 \leq u_k \leq 20 \quad (5.88)$$

La figure 5.11 présente sur la colonne gauche l'évolution dans le domaine temporel de la sortie et la consigne choisie dans le domaine viable et sur la colonne droite en vert le domaine des consignes faisables en un pas (le domaine qui préserve la faisabilité à tous les instants futurs de façon récursive).

5.5 Conclusion

Dans ce chapitre on a considéré le cas des systèmes PWA incertains. On a développé les deux algorithmes expansifs et contractif pour calculer des ensembles invariants pour les systèmes PWA incertains et dont l'incertitude est considérée comme polytopique. Différentes méthodologies pour calculer une loi de commande prédictive robuste ont été décrites en utilisant les LMI pour garantir un voisinage de l'équilibre avec un comportement stable. Une comparaison avec des méthodes alternatives de désensibilisation a été conduite et la méthode appliquée pour le cas des systèmes linéaires affectés par un retard variable sur l'entrée. On a montré comment traiter le problème de suivi de trajectoire et comment le traitement sur les opérations ensemblistes peut garantir la faisabilité de la loi de commande.

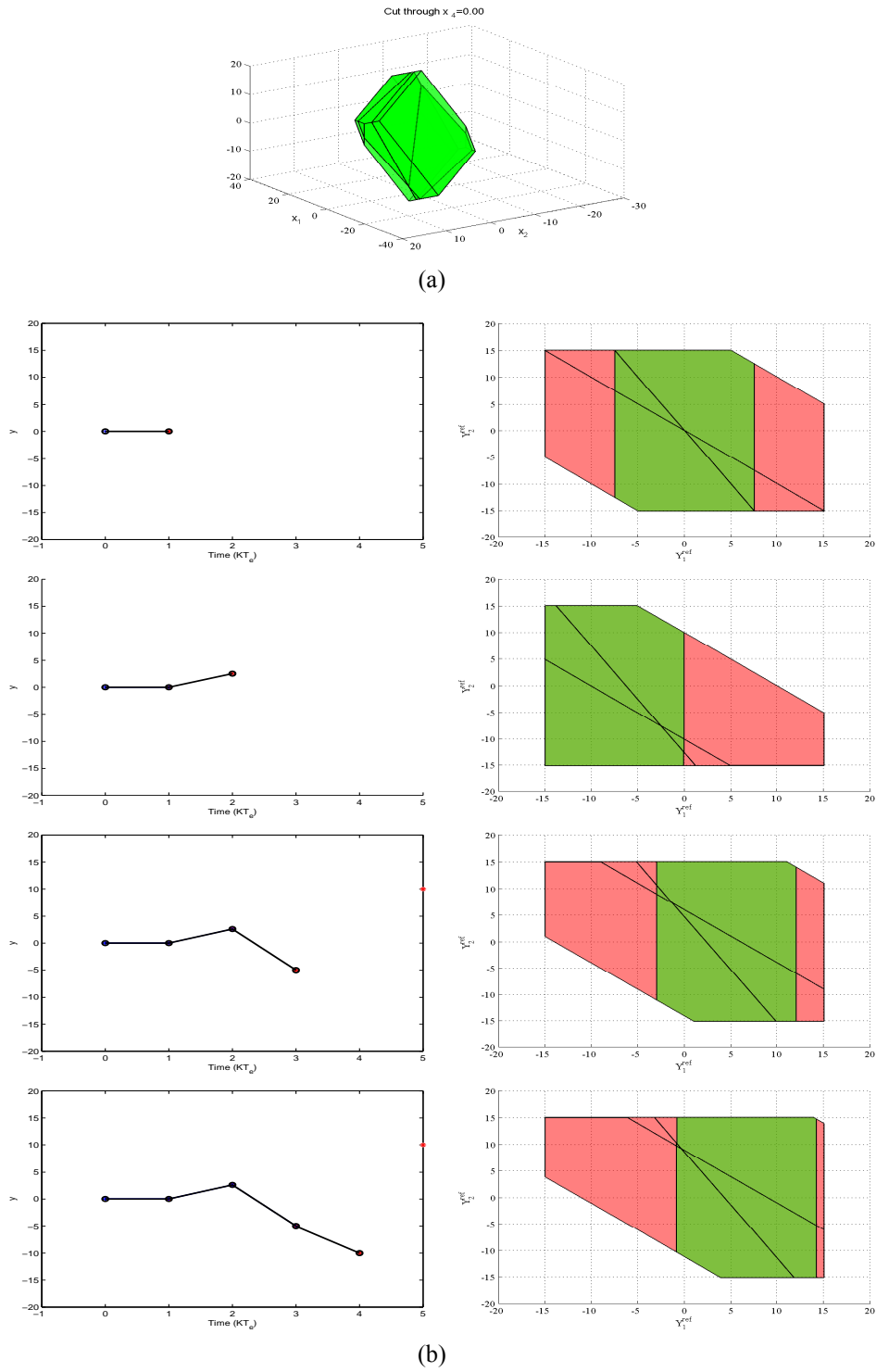


FIGURE 5.11 – Le domaine faisable pour l'évolution dans le domaine temporel.

Chapitre 6

Application : Commande et suivi de trajectoire pour une vallée hydraulique.

6.1 Introduction

La problématique consiste à réguler le niveau d'eau dans les réservoirs d'une vallée hydraulique pour permettre de suivre un certain profil de puissance que le producteur d'énergie génère pour le réseau de distribution. Dans la première partie le fonctionnement du système sera décrit et ensuite la synthèse de la loi de commande prédictive sera détaillée ainsi que la construction des tubes de viabilité qui garantissent la régulation ou la poursuite de trajectoire. La modélisation de ce système sera facilitée par l'utilisation de la formulation LMD et le logiciel Hysdel.

6.2 Description de la vallée hydraulique

La vallée hydraulique est constituée de trois réservoirs d'eau interconnectés par quatre usines U_A , U_{C_1} , U_{C_2} et U_B comme illustré dans la figure 6.1. L'usine U_A peut fonctionner en mode pompe ou en mode turbine. En mode pompe l'usine U_A pompe de l'eau dans le réservoir R_C pour alimenter le réservoir R_A en consommant de l'énergie. En mode turbine l'usine produit de l'énergie, en passant de l'eau du réservoir R_A au réservoir R_C . Les usines U_{C_1} et U_{C_2} fonctionnent en mode turbine, elles alimentent le réservoir R_B en produisant de l'énergie. L'usine U_B est alimenté par le réservoir R_B qui alimente la vallée en produisant de l'énergie.

La puissance globale P_G délivrée par la vallée est donnée par la somme des puissances fournies par chaque usine :

$$P_G = P_A + P_{C_1} + P_{C_2} + P_B \quad (6.1)$$

où P_A , P_{C_1} , P_{C_2} et P_B sont les puissances fournies respectivement par les usines U_A , U_{C_1} , U_{C_2} et U_B . Ces puissances sont définies par :

$$P_A = \alpha_A Q_A, \quad (6.2)$$

avec : $\alpha_A = \frac{760}{222}$ en mode turbine et $\alpha_A = \frac{205}{45}$ en mode pompage.

$$P_{C_1} = \alpha_{C_1} Q_{C_1}, \text{ avec : } \alpha_{C_1} = \frac{43}{96} \quad (6.3)$$

$$P_{C_2} = \alpha_{C_2} Q_{C_2}, \text{ avec : } \alpha_{C_2} = \frac{50}{120} \quad (6.4)$$

$$P_B = \alpha_B Q_B, \text{ avec : } \alpha_B = \frac{19.8}{220} \quad (6.5)$$

où Q_A, Q_B, Q_{C_1} et Q_{C_2} représentent les différents débits des usines U_A, U_B, U_{C_1} et U_{C_2} .

Les contraintes liées à la structure physique de la vallée sont :

- Le volume du réservoir R_A est $V_A \in [0, 17133]$,
- Le volume du réservoir R_B est $V_B \in [0, 1167]$,
- Le volume du réservoir R_C est $V_C \in [0, 16944]$,
- Le débit de l'usine U_A est $Q_A \in [-180, 222]$,
- Le débit de l'usine U_{C_1} est $Q_{C_1} \in [0, 120]$,
- Le débit de l'usine U_{C_2} est $Q_{C_2} \in [0, 96]$,
- Le débit de l'usine U_B est $Q_B \in [0, 220]$.

6.3 Problématique 1 : Pente de rendement constant

On considère une vallée hydraulique simplifiée constituée de trois réservoirs d'eau interconnectés par quatre usines comme illustré dans la figure 6.1 où le coefficient $\alpha_A = \frac{760}{222}$ de l'usine U_A est le même pour le mode de fonctionnement pompage et turbinage, ce qui va permettre par la suite de se placer dans le cadre d'un modèle de prédiction linéaire de dimension réduite (3 états correspondant aux niveaux dans les réservoirs).

6.4 Modélisation

Un réservoir alimenté par une source avec un débit Q_e et qui génère en sortie un débit Q_s peut être modélisé par un intégrateur comme le montre la figure 6.2. Donc le modèle global de la vallée hydraulique sera décrite comme dans la figure 6.3 par l'interconnexion de 3 intégrateurs.

6.5 Commande prédictive

6.5.1 Formulation de la commande

Le modèle du système est écrit comme :

$$\begin{cases} x_{k+1} &= \tilde{A}x_k + \tilde{B}u_k \\ y_k &= \tilde{C}x_k + \tilde{D}u_k \end{cases} \quad (6.6)$$

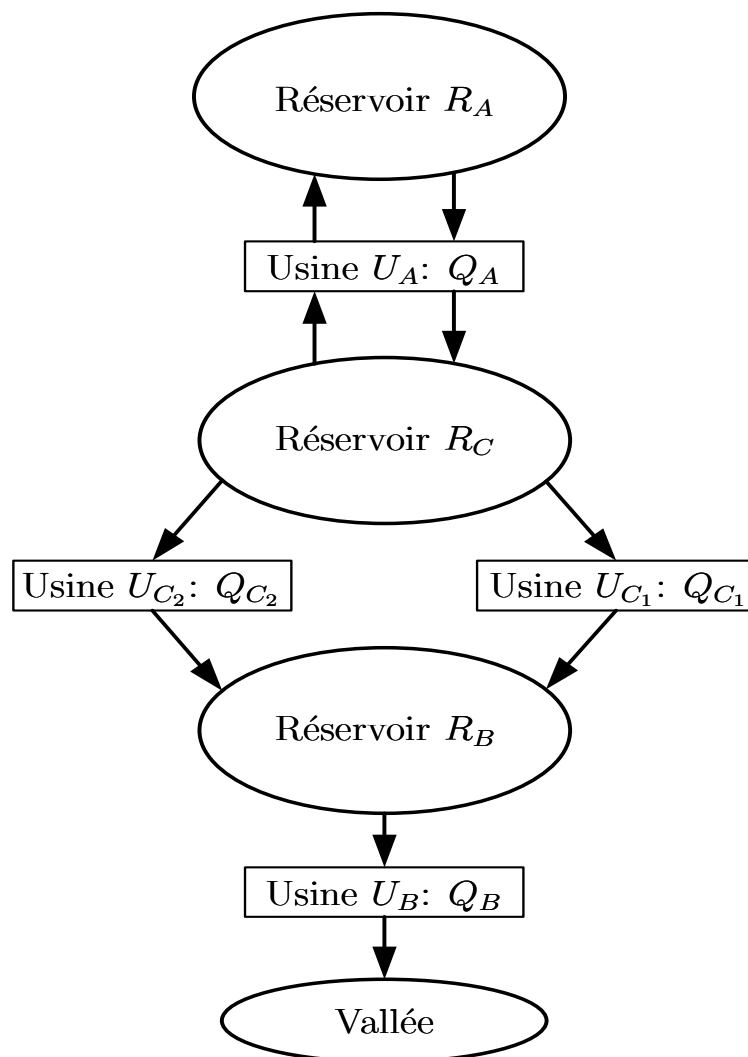


FIGURE 6.1 – Schéma global de la vallée hydraulique (Problématique 1).

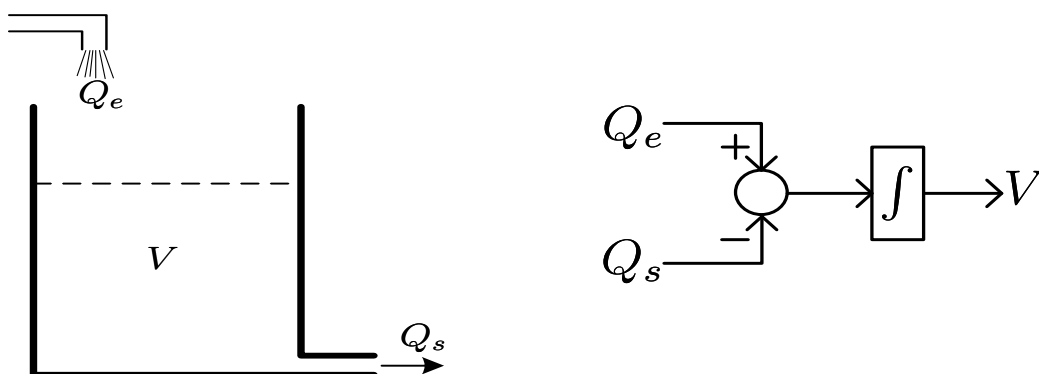


FIGURE 6.2 – Schéma d'un réservoir et son modèle intégrateur.

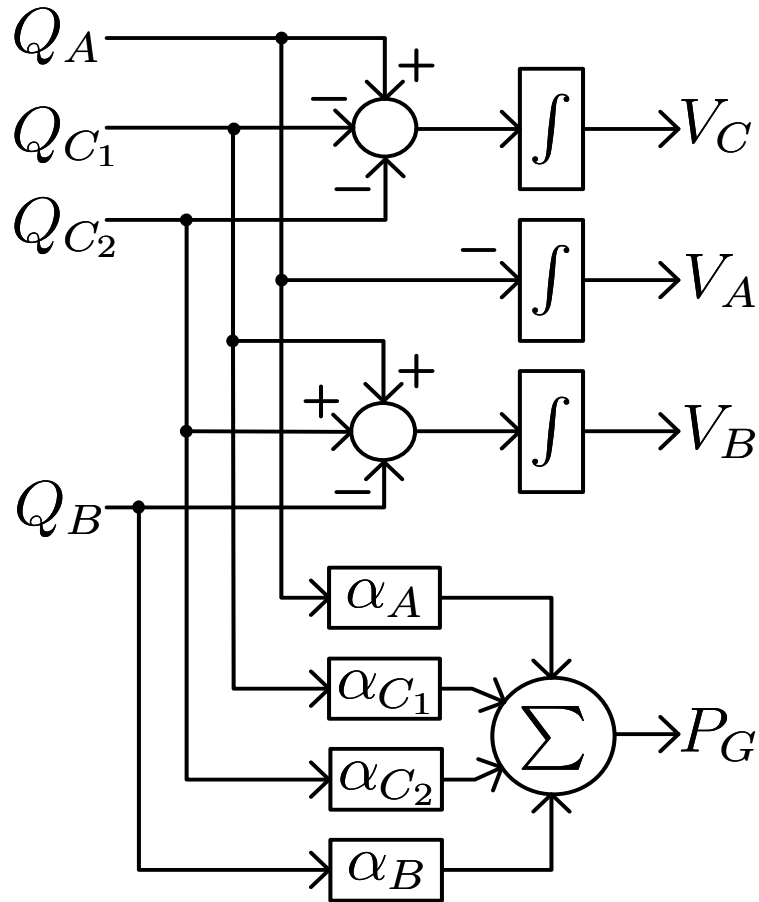


FIGURE 6.3 – Modèle global de la vallée hydraulique.

où :

$$\tilde{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 \end{bmatrix}$$

$$\tilde{C} = [0 \ 0 \ 0] \quad , \quad \tilde{D} = [3.4234 \ 0.0900 \ 0.4479 \ 0.4167]$$

la sortie du système qui est la puissance ne dépend que de la commande (les différents débits) donc $\tilde{C} = 0$ et $\tilde{D} \neq 0$.

L'objectif est de poursuivre un certain profil de puissance qui respecte les différentes contraintes liées à la structure du système et à l'objectif à atteindre (l'erreur de poursuite). Pour ce faire, on augmente l'état du système pour intégrer la consigne à suivre. La référence est définie sur un horizon N :

$$Y_{ref}^T(k) = [w_0^T, w_1^T, \dots, w_{N-1}^T, w_N^T], \quad w_i \in \mathbb{R}^m \quad (6.7)$$

avec ici $m = 1$. La mise à jour de la référence par le principe de l'horizon glissant peut être

écrite comme :

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & I & \cdots & 0 & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & I \\ * & * & \cdots & * & * \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}_k \quad (6.8)$$

L'état augmenté du système (6.6) redevient alors :

$$\underbrace{\begin{bmatrix} x \\ w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}}_{\zeta_{k+1}} = \underbrace{\begin{bmatrix} \tilde{A} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & I & \cdots & 0 & 0 \\ 0 & 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & I \\ * & * & * & \cdots & * & * \end{bmatrix}}_A \underbrace{\begin{bmatrix} x \\ w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}}_{\zeta_k} + \underbrace{\begin{bmatrix} \tilde{B} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_B u_k \quad (6.9)$$

et la sortie du système augmenté devient :

$$\epsilon_k = \underbrace{\begin{bmatrix} \tilde{C} & -I & 0 & \cdots & 0 \end{bmatrix}}_C \zeta_k + \tilde{D}u_k \quad (6.10)$$

donc le nouveau système s'écrit :

$$\begin{cases} \zeta_{k+1} = A\zeta_k + Bu_k \\ \epsilon_k = C\zeta_k + \tilde{D}u_k \end{cases} \quad (6.11)$$

La poursuite de trajectoire du système (6.6) revient à concevoir une loi de commande prédictive pour réguler la sortie (erreur) ϵ_k du système augmenté (6.11) vers 0 :

$$\begin{aligned} \min_U \sum_{i=0}^N (\epsilon_i^T Q_y \epsilon_i + u_i^T R u_i) \\ \text{sujet à : } \begin{cases} H_x x_i \leq K_x \\ H_y y_i \leq K_y \\ H_u u_i \leq K_u \\ H_\epsilon \epsilon_i \leq K_\epsilon \end{cases} \end{aligned} \quad (6.12)$$

avec Q_y et R les matrices de pondérations sur l'erreur ϵ_i et la commande u_i . Les couples de matrices (H_x, K_x) , (H_y, K_y) , (H_u, K_u) et (H_ϵ, K_ϵ) représentent les contraintes sur l'état, la sortie, la commande et l'erreur de poursuite.

6.5.2 Résolution et simulation

On calcule une loi de commande prédictive en-ligne, donc à chaque pas d'échantillonnage on résout un problème d'optimisation défini par le critère de coût (6.12) avec les données suivantes :

$$N = 2, R = \mathbb{I}_4, Q_y = 1, 10, 100$$

Les contraintes sur l'état du système (les différents volumes des réservoirs), la commande et l'erreur de poursuite sont données par :

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} x_k \leq \begin{bmatrix} 17133 \\ 0 \\ 1167 \\ 0 \\ 16944 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \leq \begin{bmatrix} 800 \\ 800 \\ 800 \\ 800 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} u_k \leq \begin{bmatrix} 222 \\ 180 \\ 220 \\ 0 \\ 120 \\ 0 \\ 96 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} \epsilon_k \leq \begin{bmatrix} 800 \\ 800 \end{bmatrix}$$

où les contraintes sur x_k , w_1 , w_2 et u_k résultent des données et des valeurs extrêmes indiquées au paragraphe 6.2. Pour la contrainte sur ϵ_k elle est relaxée.

On considère l'état initial du système $x_0 = [15000, 100, 3666]^T$, les différentes simulations avec différents facteurs de pondérations sur l'erreur Q_y sont données par les figures 6.4, 6.5 et 6.6.

Les différentes simulations montrent bien la diminution de l'erreur de poursuite chaque fois que le facteur de pondération Q_y augmente et qui se traduit aussi par une augmentation de l'amplitude du débit (commande) Q_A en respectant toujours les contraintes.

6.5.3 Synthèse MPC

Une loi de commande prédictive explicite est calculée avec les mêmes données que dans la section précédente. La solution obtenue pour $N = 2$ et $Q_y = 1$ (c'est-à-dire dans un espace d'état de dimension 5) est de 156 régions. La loi de commande obtenue est de la forme :

$$u_k = F_i x_k + G_i, \text{ avec } x_i \in R_i \quad (6.13)$$

On peut simplifier cette loi de commande en considérant que seule la première composante de la commande est possible et la nouvelle loi de commande ne contient que 39 régions de dimension 5. La figure 6.7 montre le temps nécessaire pour calculer une loi de commande d'un contrôleur en-ligne et d'un contrôleur explicite avec un certain avantage pour ce dernier (au moins pour des faibles horizons de prédiction).

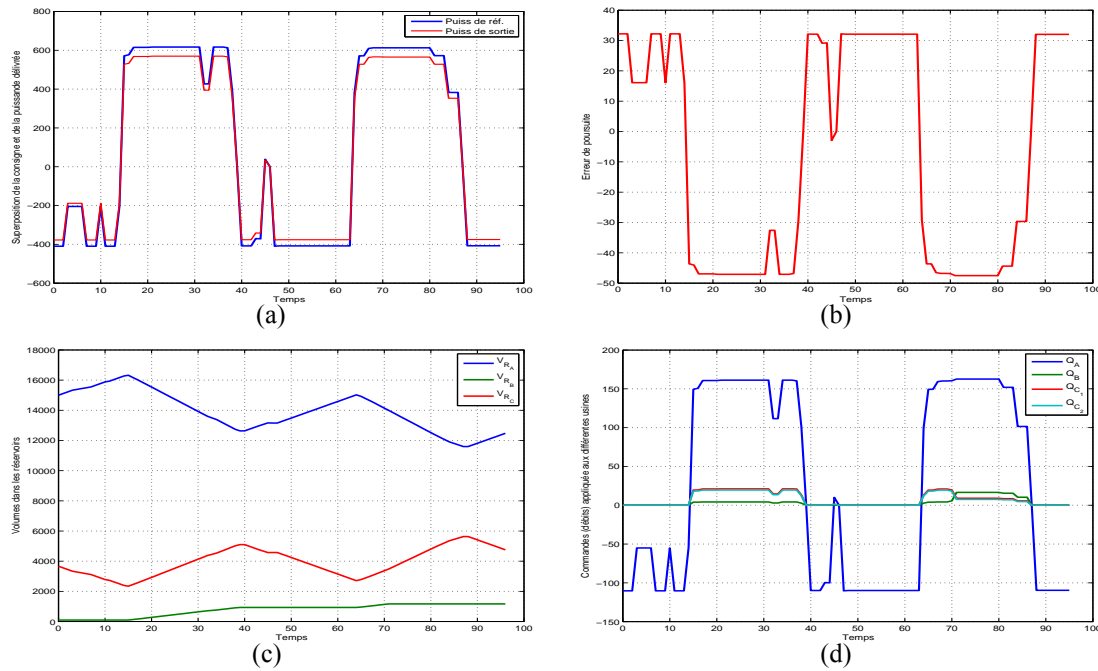


FIGURE 6.4 – Facteur de pondération sur l'erreur $Q_y = 1$. (a) Visualisation de la consigne et de la sortie. (b) Érreur de poursuite. (c) Évolution de l'état du système (volumes des réservoirs). (d) Commande (débits) appliquée au système.

Calcul du domaine viable

Le calcul du domaine viable par l'algorithme 1 du chapitre 5 donne une solution avec plus de 10000 régions après seulement 3 itérations. Cette complexité de calcul nous pousse à utiliser l'analyse par intervalles mentionnée dans le chapitre 4 avec l'algorithme 1 du chapitre 5. La solution obtenue possède 286 régions de dimension 5 après 10 itérations de calcul. On voit bien l'intérêt d'utiliser l'analyse par intervalles pour diminuer la complexité de calcul et ainsi le temps de calcul.

nombre d'itérations	1	2	3	4	5	6	7	8	9	10
nombre de régions	420	345	308	307	307	302	287	287	287	286
nombre d'opérations	9	6	6	6	6	6	6	6	6	6

TABLE 6.1 – Résumé des nombres de régions et nombres d'opérations effectuées par l'algorithme 1 du chapitre 4 avec utilisation de l'analyse par intervalles.

6.5.4 Utilisation du domaine viable (invariance)

Scénario 1 : suivi de trajectoire

Cette simulation consiste à choisir à chaque pas d'échantillonnage une référence à suivre, choisie dans le domaine viable qui permet de garantir la faisabilité de la loi de commande au

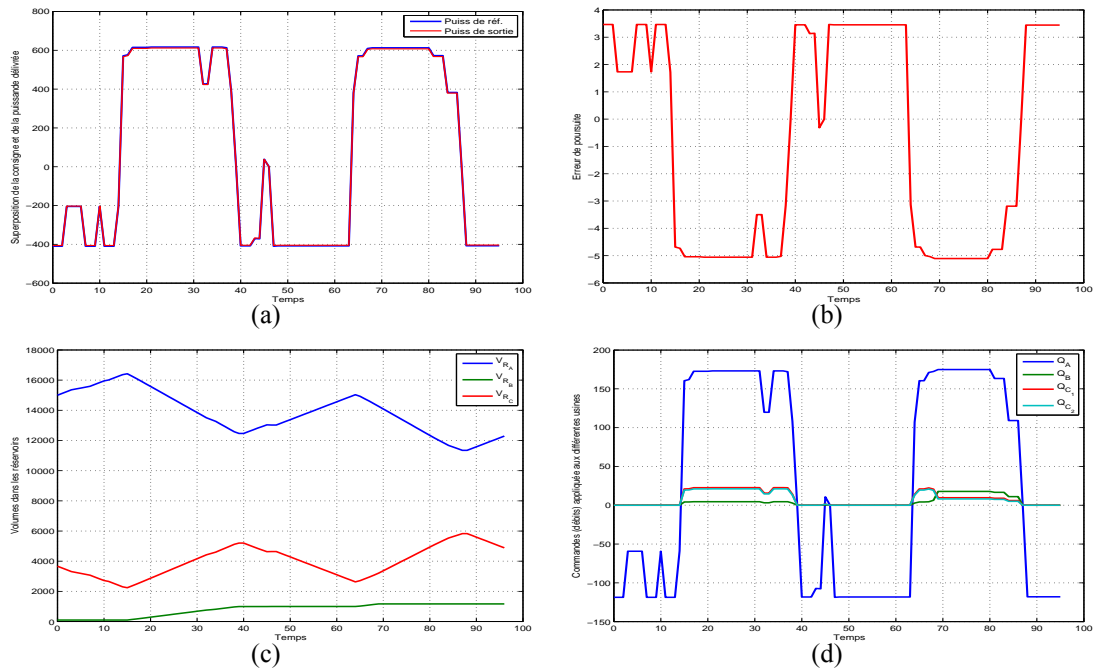


FIGURE 6.5 – Facteur de pondération sur l’erreur $Q_y = 10$. (a) Visualisation de la consigne et de la sortie. (b) Erreur de poursuite. (c) Évolution de l’état du système (volumes des réservoirs). (d) Commande (débits) appliquée au système.

prochain pas. On a choisi ici d’introduire une contrainte plus sévère : $-40 \leq \epsilon_k \leq 40$. La figure 6.8(a) montre la superposition du contrôleur explicite et le domaine viable à chaque pas d’échantillonnage. On voit bien que le contrôleur explicite ne garantit pas la faisabilité de la loi de commande à n’importe quel moment donc les références de puissance à suivre devraient être à l’intérieur du domaine viable (loi de commande) et non pas dans la région initiale de la formulation explicite.

Scénario 2 : aspect dissipatif du système

Dans ce scénario on voudrait montrer l’aspect dissipatif du système quel que soit le niveau d’eau des réservoirs. On a généré une référence de puissance aléatoire (uniformément distribuée dans l’intervalle $[-800, 800]$) choisie à l’intérieur du domaine viable, on voit que le niveau d’eau des réservoirs diminue jusqu’à ce qu’il redevienne vide (figure 6.9(c)), ce qui est en relation directe avec le fonctionnement en conditions réelles qui profite des apports extérieurs pour prolonger la viabilité de la politique de production.

En changeant la pondération sur la commande Q_B pour qu’elle redevienne $R_{Q_B} = 0.1$ on favorise plus l’évacuation du réservoir R_B . Les figures 6.10(a), 6.10(b) et 6.10(c) montrent bien cet aspect par rapport à la précédente simulation.

Pour la simulation de la figure 6.11 on a construit un contrôleur MPC explicite mais en relaxant la contrainte sur l’erreur de poursuite, avec une marge d’erreur $\epsilon_k \in [-800, 800]$ qui rend la commande insensible aux variations de la référence.

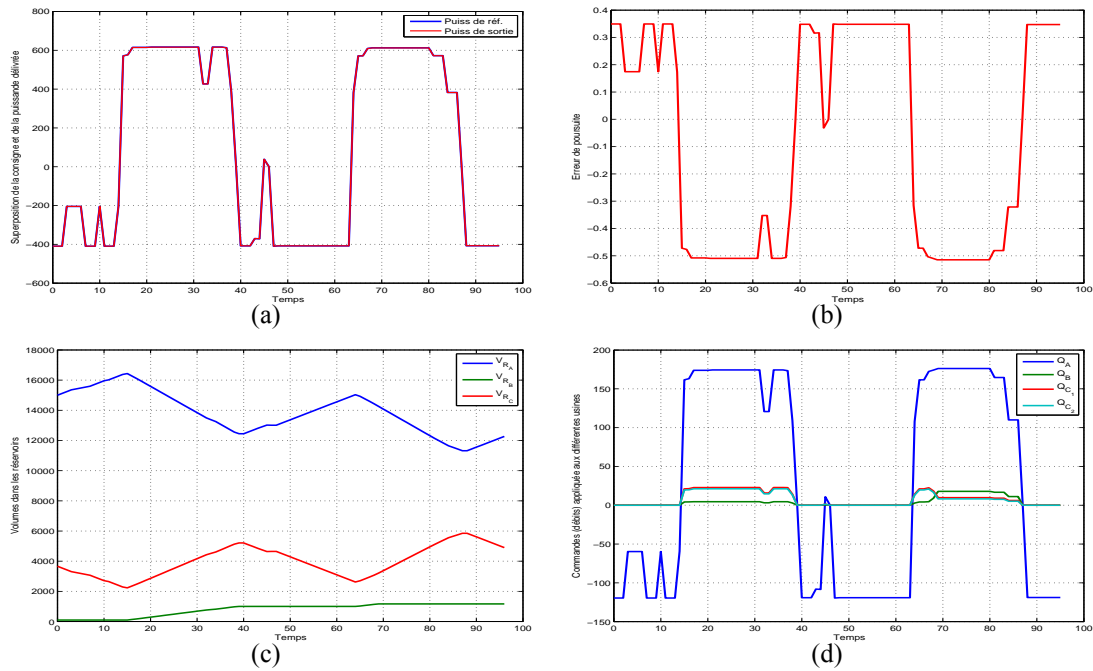


FIGURE 6.6 – Facteur de pondération sur l’erreur $Q_y = 100$. (a) Visualisation de la consigne et de la sortie. (b) Érrer de poursuite. (c) Évolution de l’état du système (volumes des réservoirs). (d) Commande (débits) appliquée au système.

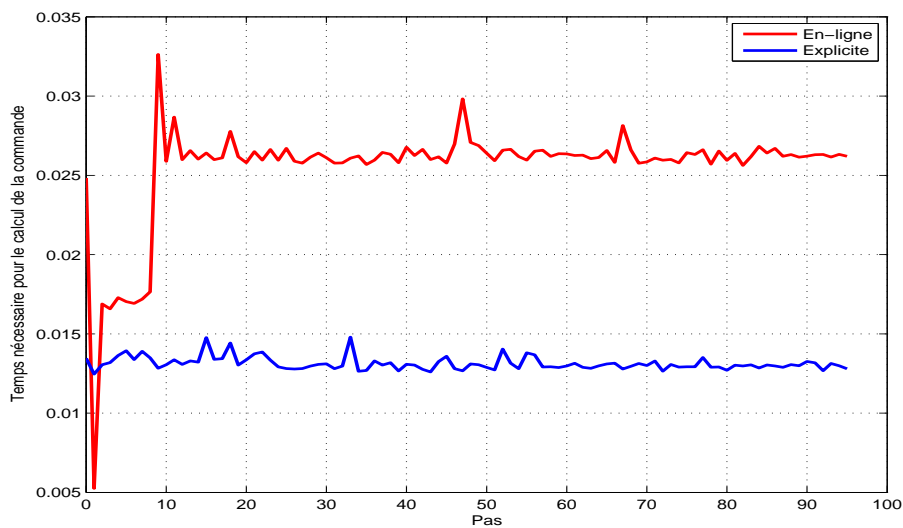


FIGURE 6.7 – Temps de calcul de la loi de commande en-ligne et explicite.

Scénario 3 : Cycles viables

Dans cette simulation on vise à calculer combien de fois on peut appliquer le profil de la puissance (figure 6.4(a)) sur la vallée avant que la commande ne devienne infaisable (impossibilité de suivre la trajectoire avec la marge d’erreur imposée). Le nombre de profils qu’on peut

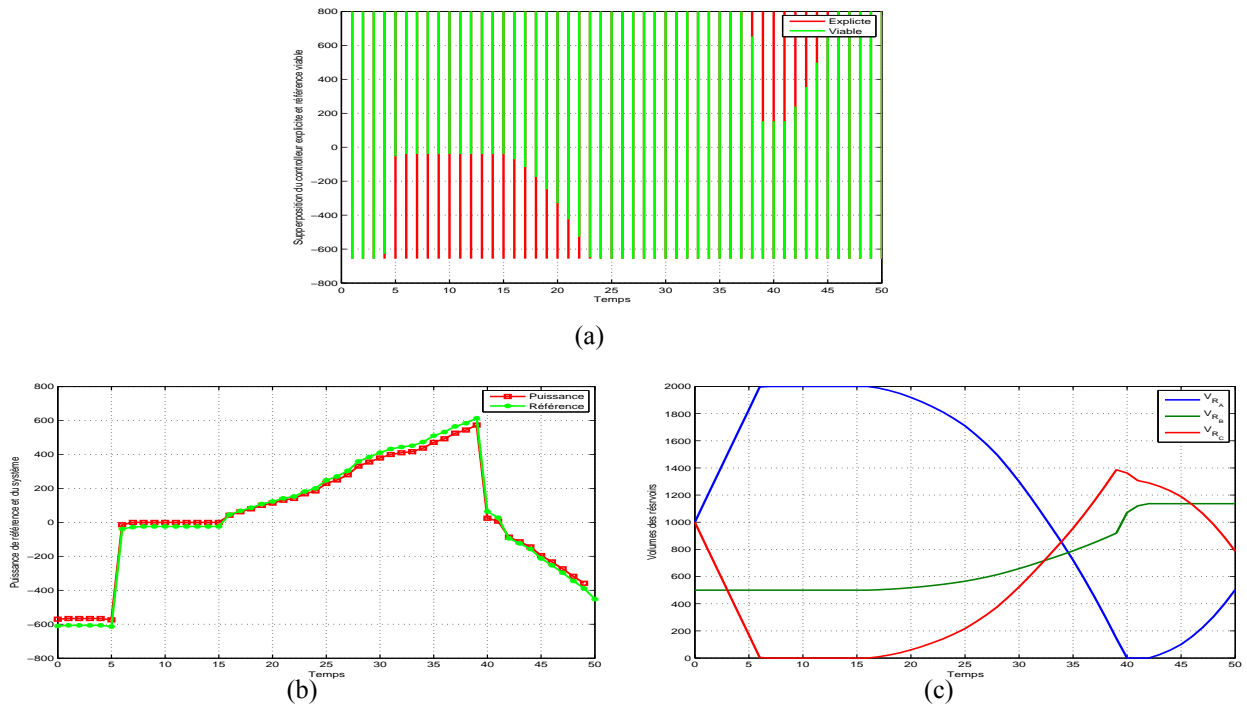


FIGURE 6.8 – (a) Superposition du contrôleur explicite et référence viable. (b) Puissance de référence et puissance fournie. (c) Volumes des réservoirs.

appliquer périodiquement est de 4 (figure 6.12). On refait la simulation en diminuant la pondération sur la commande (le débit du réservoir R_B), la nouvelle pondération étant $R_{Q_B} = 0.1$. La longueur de profil qu'on peut appliquer est de 5 cycles (figure 6.13).

6.6 Problématique 2 : Pente de rendement différencié pompage/turbinage

Dans cette problématique on a remplacé l'usine U_A qui fonctionnait en pompage et turbinage par deux usines U_{A_1} et U_{A_2} indépendantes telles que la première fonctionne en turbinage avec un débit $Q_{A_1} \in [0, 222]$ et la deuxième en mode pompage avec un débit $Q_{A_2} \in [-188, 0]$. Le système voit la dimension du vecteur d'entrée augmentée et le schéma global du modèle est montré dans la figure 6.14. La loi de commande explicite possède 305 régions réduites à 61 régions en ne considérant que la première composante de la loi de commande (principe de l'horizon glissant). Résultat sur la figure 6.15.

6.7 Problématique 3 : tube de viabilité

L'objectif de cette problématique est de trouver un tube de viabilité autour d'un programme de référence donné de telle sorte que si on est dans ce tube à un instant donné alors la viabilité du suivi du programme de référence est assurée dans l'instant futur.

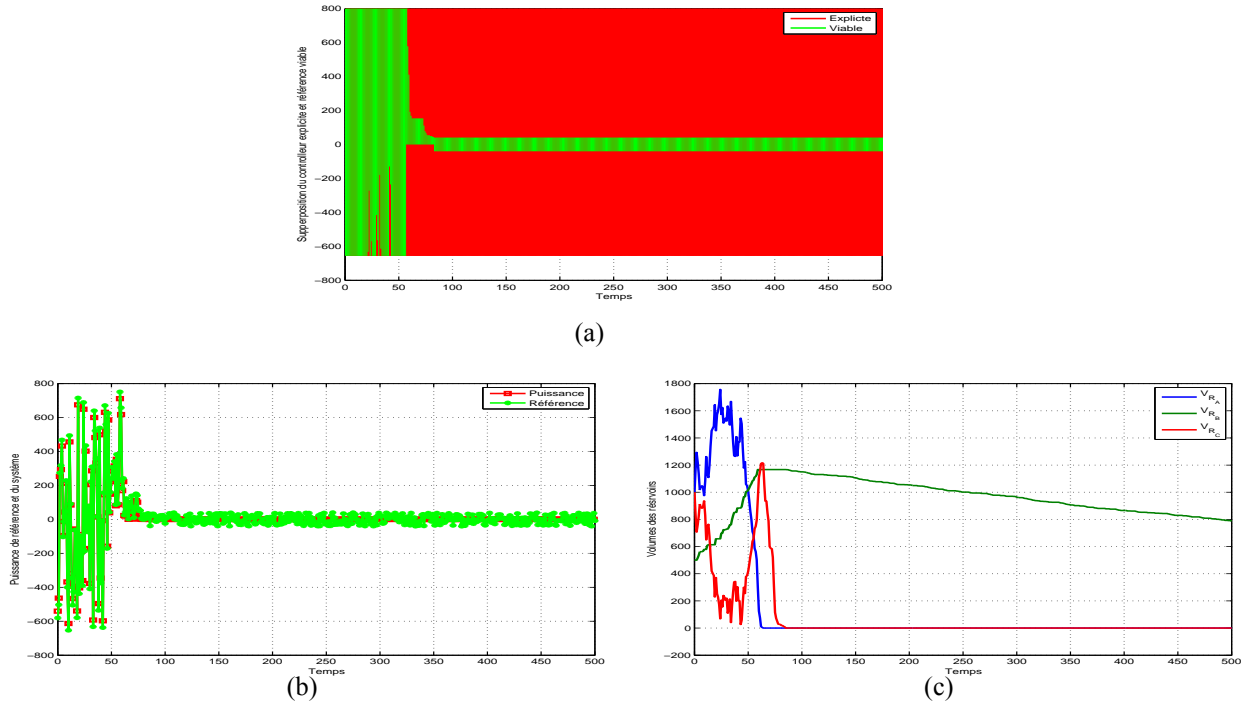


FIGURE 6.9 – (a) Superposition du contrôleur explicite et référence viable. (b) Puissance de référence et puissance fournie. (c) Volumes des réservoirs.

Le principe est de partir de l'instant final T_f puis de remonter le temps jusqu'à l'instant initial T_i en suivant le profil de puissance de référence.

Les valeurs de la puissance du programme de référence à l'instant T_i sont données par W_{T_i} , le principe de construction du tube de viabilité en remontant le temps est le suivant :

- calculer l'ensemble viable X_N à l'instant final T_N :

$$X_N = \text{projection}_x \left\{ \begin{array}{l} x \in \mathbb{R}^3, u \in \mathbb{R}^4 \parallel \begin{bmatrix} E & F \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq G \\ \begin{bmatrix} 0 & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = W_N \end{array} \right\} \quad (6.14)$$

- calculer l'ensemble viable X_{N-1} à l'instant précédent T_{N-1} :

$$X_{N-1} = \text{projection}_x \left\{ \begin{array}{l} x \in \mathbb{R}^3, u \in \mathbb{R}^4 \parallel \begin{bmatrix} E & F \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq G \\ \begin{bmatrix} 0 & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = W_{N-1} \\ \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \in X_N \end{array} \right\} \quad (6.15)$$

La figure 6.16 montre l'évolution du domaine viable en fonction du temps (porté sur l'axe vertical), en le projetant sur les différentes dimensions de l'état x pour la clarté des dessins, sachant qu'on a utilisé le profil de puissance 6.4(a). Dans la figure 6.17 la simulation a considéré

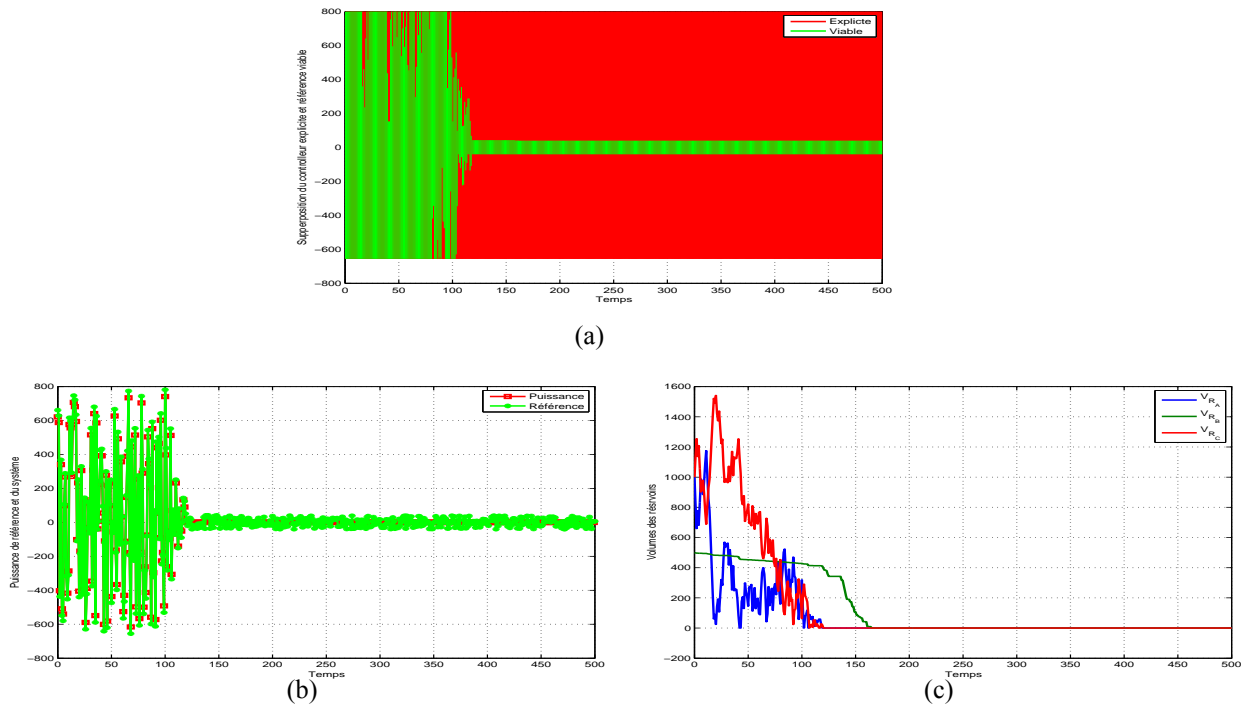


FIGURE 6.10 – (a) Superposition du contrôleur explicite et référence viable. (b) Puissance de référence et puissance fournie. (c) Volumes des réservoirs.

le profil de la puissance 6.4(a) multiplié par un facteur de 1.5 pour bien montrer le besoin d'avoir les réservoirs avec plus de quantité d'eau en remontant le temps.

6.8 Conclusion

Dans cette application industrielle on a calculé une loi de commande prédictive et explicite pour la poursuite de profils de puissance. Le système étudié est un système linéaire soumis à des contraintes linéaires qui ont permis la mise en œuvre des algorithmes pour la construction des ensembles viables pour la poursuite de trajectoire. On a montré l'intérêt d'utiliser l'analyse par intervalles pour alléger la complexité des calculs due à l'explosion exponentielle des nombres de régions à chaque itération.

Plusieurs scénarios ont été mis en scène pour comparer les différents comportements du système pour quelques configurations extrêmes de son état à savoir le volume des réservoirs et aussi pour des valeurs extrêmes des puissances de référence à poursuivre. On a vu aussi comment calculer les tubes de viabilité pour une trajectoire donnée en se basant sur la programmation dynamique. La technique considère le dernier point de profil de référence et calcule le domaine autour de la trajectoire pour lequel on peut rejoindre le profil de puissance en un seul pas quel que soit l'état du système dans le domaine respectif.

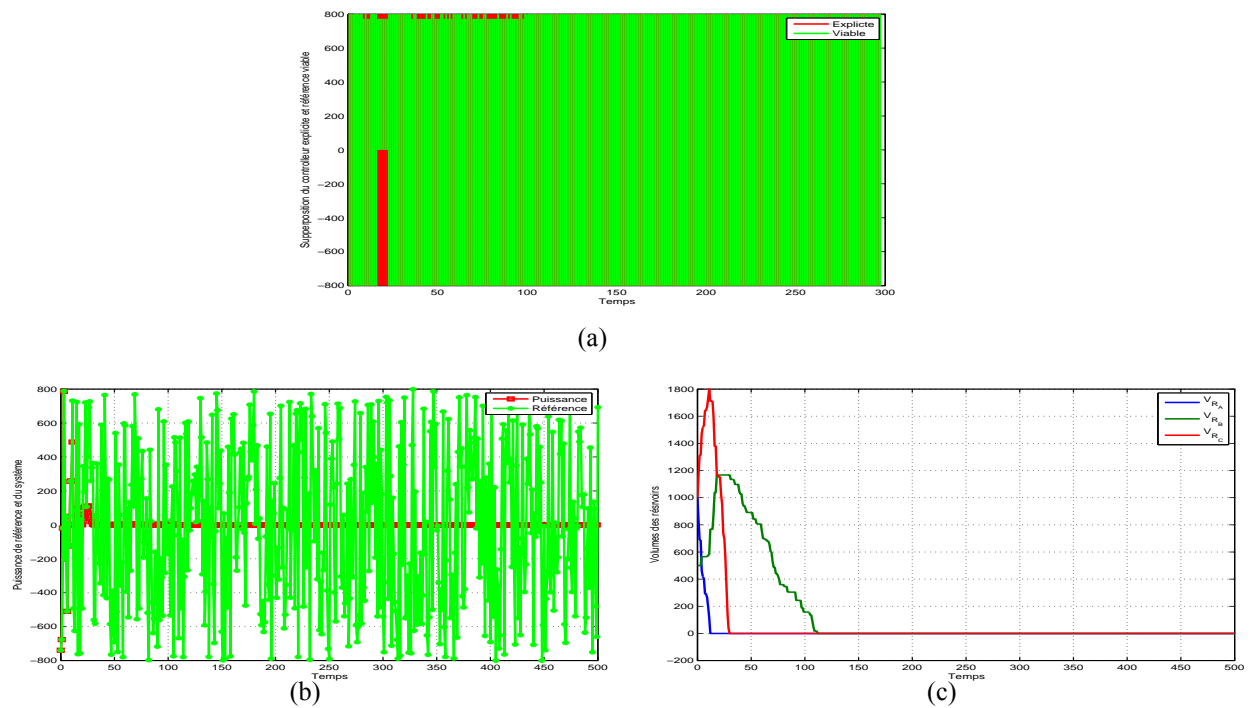


FIGURE 6.11 – (a) Superposition du contrôleur explicite et référence viable. (b) Puissance de référence et puissance fournie. (c) Volumes des réservoirs.

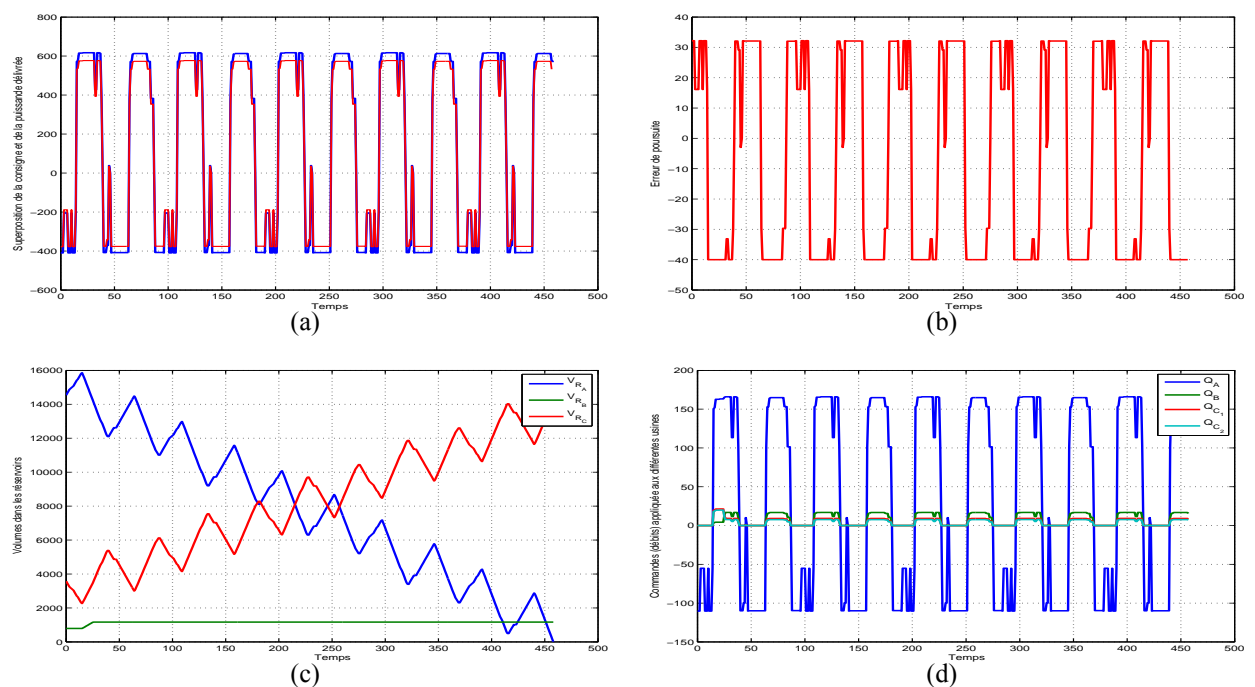


FIGURE 6.12 – (a) Visualisation de la consigne et de la sortie. (b) Erreur de poursuite. (c) Évolution de l'état du système (volumes des réservoirs). (d) Commande (débits) appliquées au système.

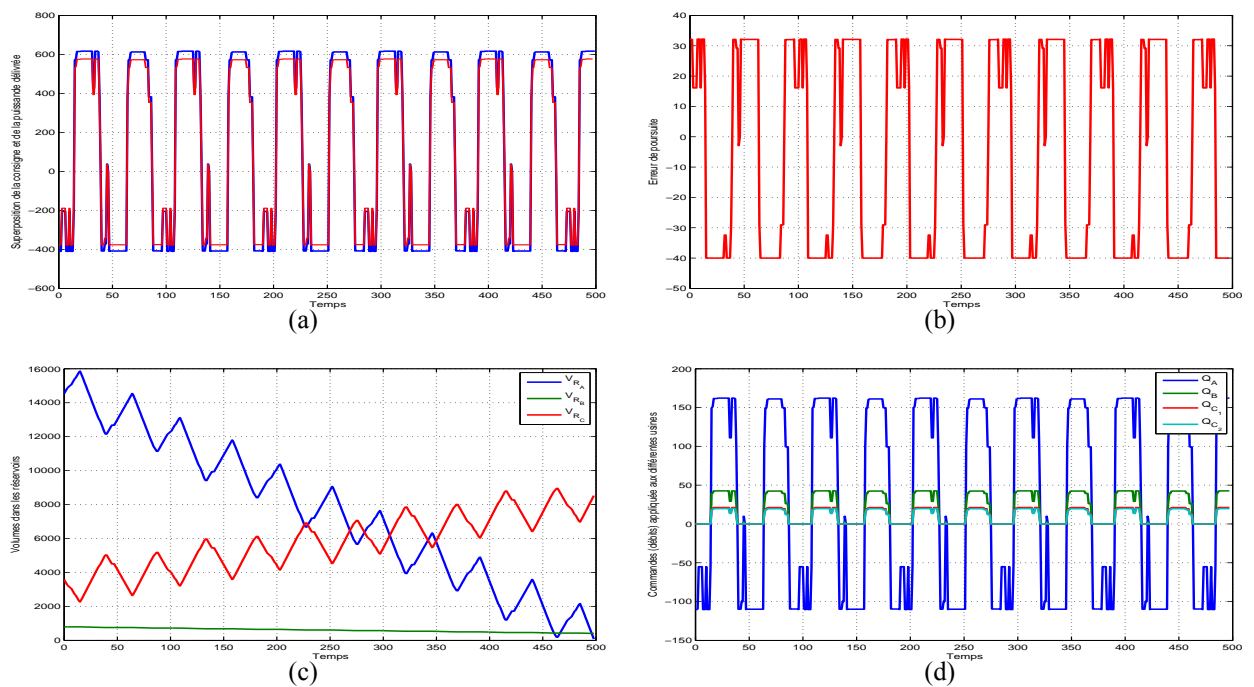


FIGURE 6.13 – (a) Visualisation de la consigne et de la sortie. (b) Écart de poursuite. (c) Évolution de l'état du système (volumes des réservoirs). (d) Commande (débits) appliquées au système.

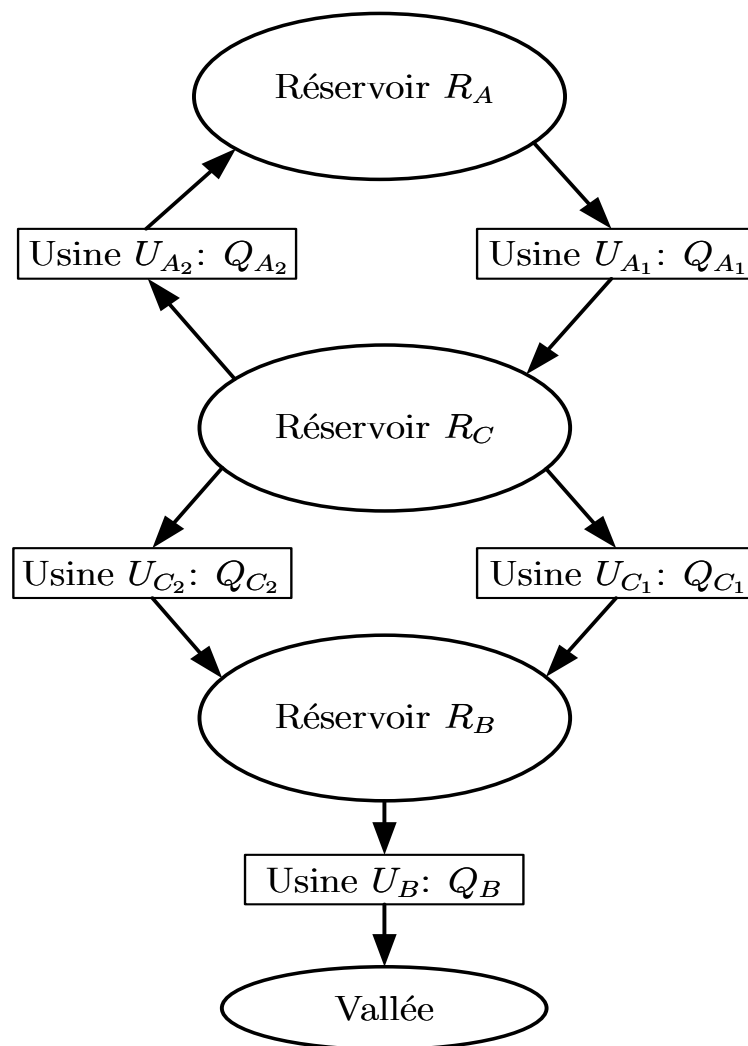


FIGURE 6.14 – Schéma global de la vallée hydraulique (Problématique 2).

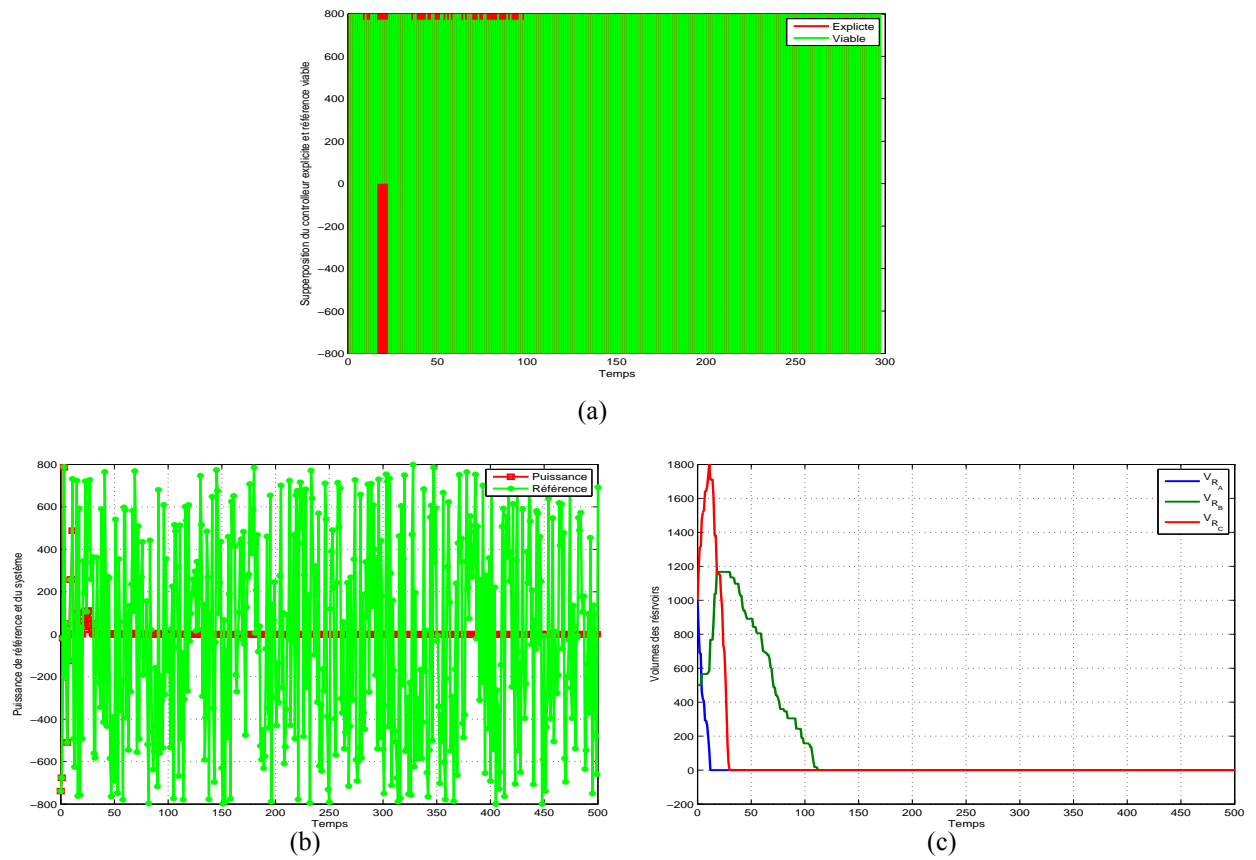


FIGURE 6.15 – (a) Superposition du contrôleur explicite et référence viable. (b) Puissance de référence et puissance fournie. (c) Volumes des réservoirs.

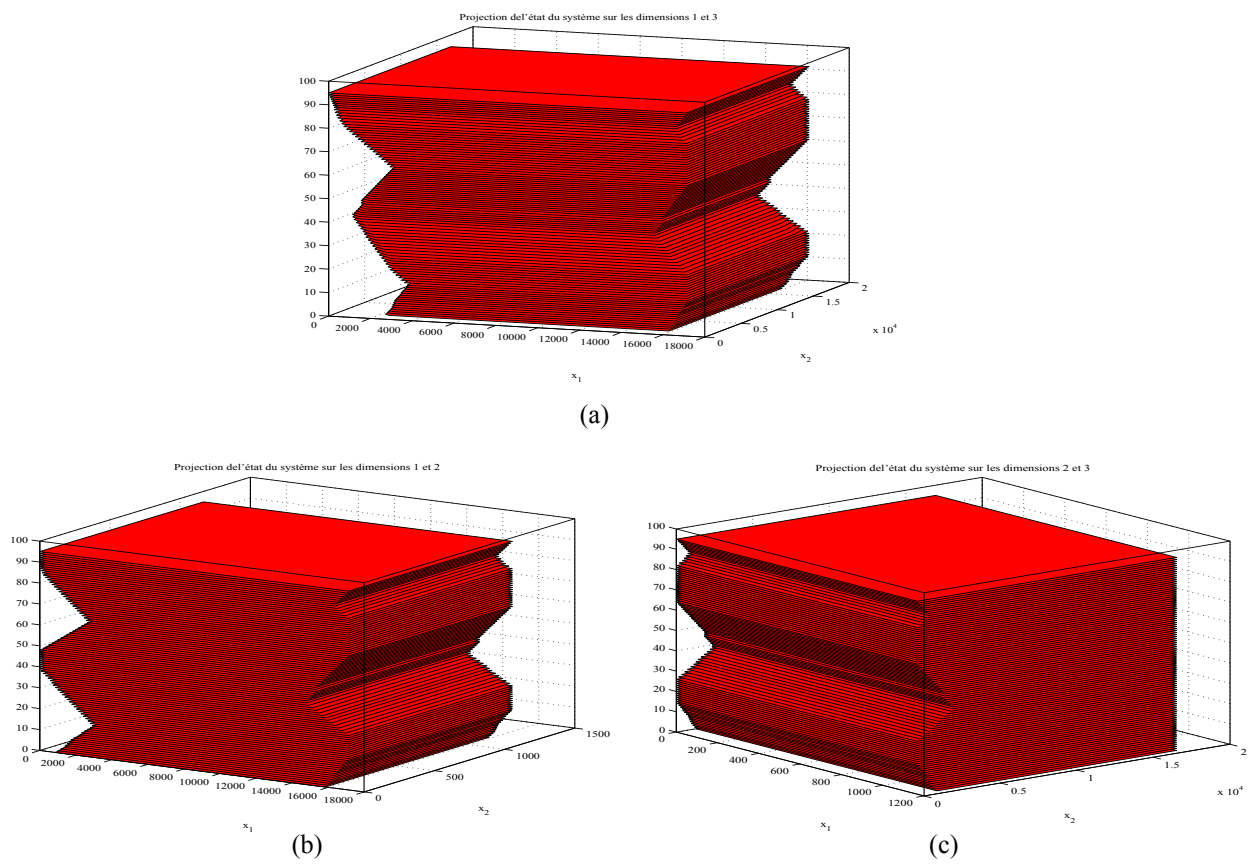


FIGURE 6.16 – (a) Évolution du domaine viable sur les dimensions $[1, 3]$ de l'état x . (b) Évolution du domaine viable sur les dimensions $[1, 2]$ de l'état x . (c) Évolution du domaine viable sur les dimensions $[2, 3]$ de l'état x .

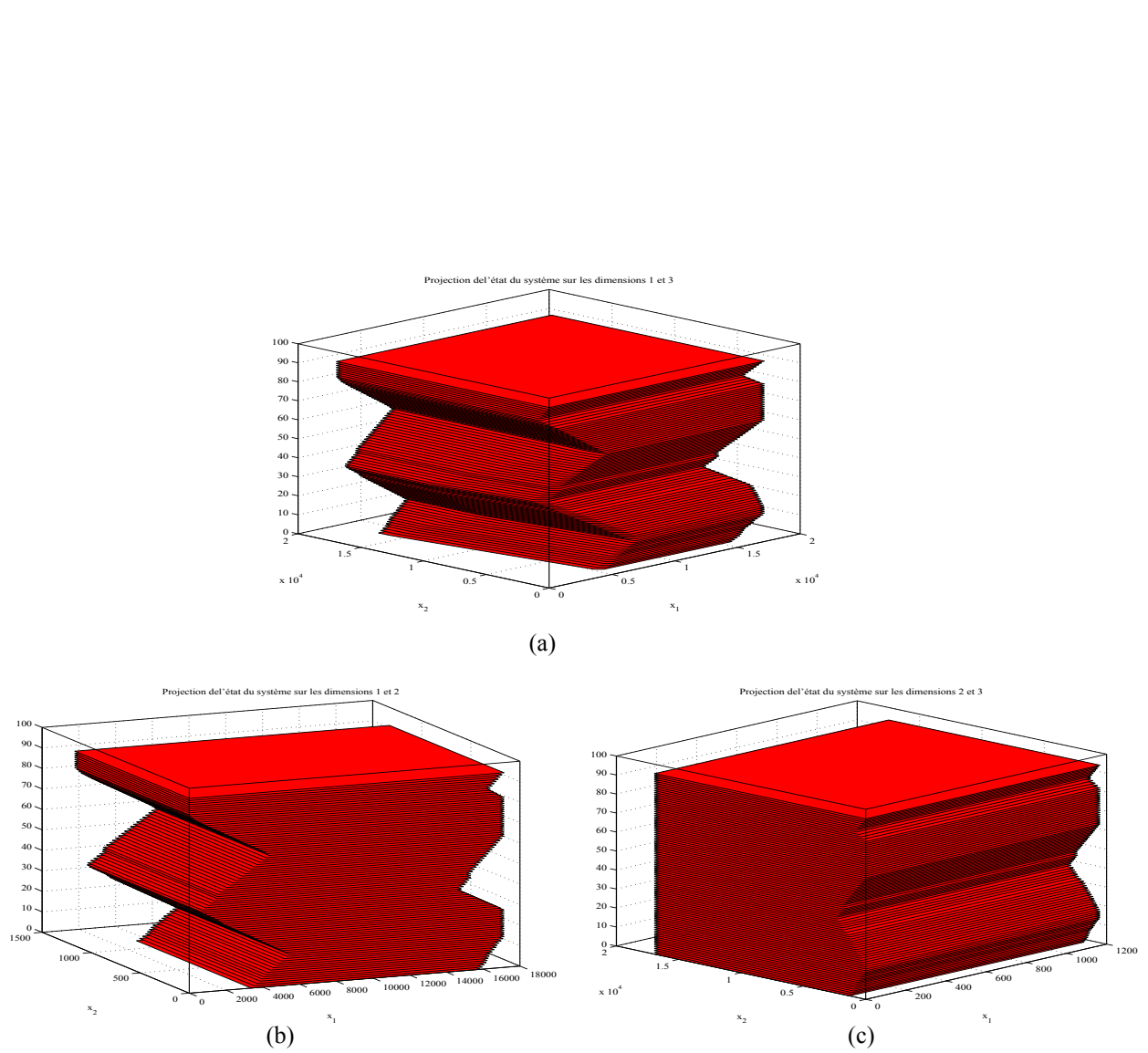


FIGURE 6.17 – (a) Évolution du domaine viable sur les dimensions $[1, 3]$ de l'état x . (b) Évolution du domaine viable sur les dimensions $[1, 2]$ de l'état x . (c) Évolution du domaine viable sur les dimensions $[2, 3]$ de l'état x .

Chapitre 7

Conclusion

7.1 Contribution

Cette thèse s'est intéressée à la construction des ensembles invariants pour les systèmes PWA qui représente le cadre général d'analyse des systèmes linéaires en boucle fermée avec une loi de commande prédictive sous contraintes.

Les contributions principales de ce travail résident dans la construction des ensembles invariants pour les systèmes affines par morceaux. Les techniques sont détaillées pour le cas des systèmes PWA définis sur une partition polyédrale de l'espace d'état avec trois approches différentes :

- la construction expansive,
- la construction contractive,
- la construction basée sur les graphes des transitions.

Au niveau méthodologique toutes ces constructions étant basées sur la dynamique directe et en temps inverse des ensembles de l'espace d'état, elles impliquent un traitement géométrique dans la partie de comparaison des régions résultant des différentes transformations linéaires induites par la dynamique. Cette opération peut s'avérer complexe du point de vue de la charge de calcul et pour éviter l'explosion combinatoire, une solution innovante a été proposée en exploitant l'analyse par intervalles. Il est intéressant d'observer que la construction des ensembles invariants ouvre la voie au post-traitement des lois de commandes prédictives en vue de la maximisation de leur domaine de fonctionnement avec garantie de sûreté. Des comparaisons ont été faites entre les différentes structures MPC avec d'une part les formulations qui bénéficient du renforcement de l'invariance dès la phase de synthèse et d'autre part les formulations explicites qui bénéficient de la post analyse pour la caractérisation des domaines viables.

La construction des ensembles invariants profite principalement à la commande prédictive et le présent travail fait état des extensions de ces méthodes pour le suivi de trajectoire, les incertitudes paramétriques et la prise en compte d'un retard variable à l'entrée du système.

7.2 Directions et perspectives

Une des perspectives immédiates ouverte par le présent travail est la possibilité de synthétiser une commande prédictive et d'obtenir une version de cette loi de commande avec garantie de l'invariance par la modification locale des dynamiques dans les sous régions affectées par l'infaisabilité en temps fini. Une discussion intéressante peut être conduite à propos du prix à payer pour un tel raffinement, une solution naïve étant la relaxation de la continuité de la loi de commande pour ces régions critiques. D'autres approches plus complexes, qui passent par l'introduction d'un sous-partitionnement, peuvent être envisagées.

Un problème qui n'a pas été traité en détail dans le présent travail est l'identification de la présence d'autres points fixes que ceux du système nominal. On fait référence ici aux points d'équilibre autres que l'origine, les cycles limites, les solutions pseudo périodiques et leurs bassins d'attractions.

Pour élargir le cadre de la présente étude, la synthèse de lois de commande prédictive dans un cadre non linéaire doit être considérée. Une extension naturelle des résultats présentés dans cette thèse est la commande prédictive des systèmes hybrides pour lesquels il est connu que les formulations explicites prennent la forme de lois linéaires par morceaux [31].

Annexe A

Calcul géométrique sur les polytopes

Étant donnés deux polytopes \mathcal{R}_1 et \mathcal{R}_2 définis par :

$$\mathcal{R}_i = \{x \in \mathbb{R}^n | H_i x \leq h_i\}. \quad (\text{A.1})$$

avec $H_i \in \mathbb{R}^{l_i \times n}$ et $h_i \in \mathbb{R}^{l_i \times 1}$, définissant les hyperplans de support. Les polytopes \mathcal{R}_i peuvent aussi être définis dans une représentation duale par leurs sommets (générateurs) :

$$\text{sommets}(\mathcal{R}_i) = \{[s_{i,1} \ s_{i,2} \ \dots \ s_{i,m_i}] | s_{i,tt} = \begin{bmatrix} s_{i,tt}^1 \\ s_{i,tt}^2 \\ \vdots \\ s_{i,tt}^n \end{bmatrix} \in \mathbb{R}^{n \times 1}, tt = 1, \dots, m_i, m_i \in \{3, 4, \dots\}\} \quad (\text{A.2})$$

Avant de faire un rappel sur quelques définitions sur le calcul géométrique sur l'espace des polytopes, on introduit les définitions des boules $\mathbb{B}_1^\epsilon(c)$, $\mathbb{B}_2^\epsilon(c)$ et $\mathbb{B}_\infty^\epsilon(c)$ définies dans \mathbb{R}^n , centrées en un point $c \in \mathbb{R}^n$ et de rayon ϵ ,

$$\mathbb{B}_1^\epsilon(c) = \{x \in \mathbb{R}^n | \|x - c\|_1 = |x_1 - c_1| + \dots + |x_n - c_n| \leq \epsilon\} \quad (\text{A.3})$$

$$\mathbb{B}_2^\epsilon(c) = \left\{x \in \mathbb{R}^n | \|x - c\|_2 = \sqrt{|x_1 - c_1|^2 + \dots + |x_n - c_n|^2} \leq \epsilon\right\} \quad (\text{A.4})$$

$$\mathbb{B}_\infty^\epsilon(c) = \{x \in \mathbb{R}^n | \|x - c\|_\infty = \max(|x_1 - c_1|, \dots, |x_n - c_n|) \leq \epsilon\} \quad (\text{A.5})$$

Dans la figure A.1 on trouve une illustration géométrique de ces définitions.

A.1 Opérations sur les ensembles polyédraux

La somme de Minkowski entre deux polytopes, \mathcal{R}_1 et \mathcal{R}_2 , est définie par :

$$\mathcal{R}_1 \oplus \mathcal{R}_2 = \{x_1 + x_2 | x_1 \in \mathcal{R}_1, x_2 \in \mathcal{R}_2\} \quad (\text{A.6})$$

La figure A.2 illustre un exemple d'addition entre deux polytopes \mathcal{R}_1 et \mathcal{R}_2 .

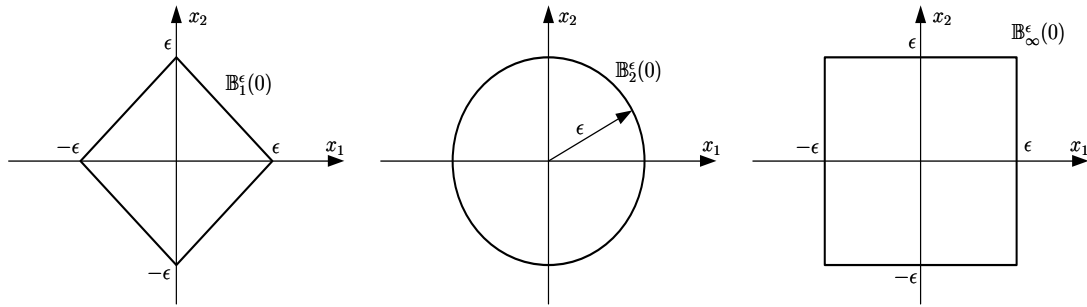


FIGURE A.1 – Illustration graphique dans l'espace \mathbb{R}^2 des différentes boules.

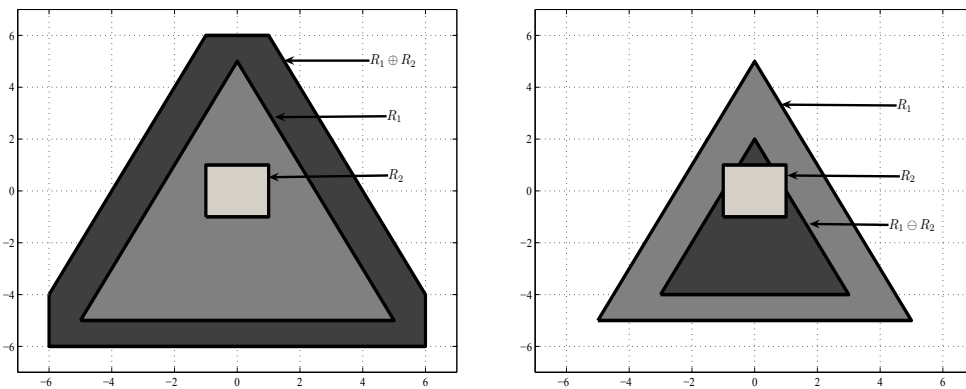


FIGURE A.2 – Illustration géométrique de l'addition et de la soustraction dans \mathbb{R}^2 .

Soustraction de Minkowski

La soustraction de Minkowski (Pontryagin) entre deux polytopes, \mathcal{R}_1 et \mathcal{R}_2 , est définie par :

$$\mathcal{R}_1 \ominus \mathcal{R}_2 = \{x \in \mathcal{R}_1 \mid x + w \in \mathcal{R}_1, \forall w \in \mathcal{R}_2\} \tag{A.7}$$

On note que $\mathcal{R}_1 \ominus \mathcal{R}_2 \oplus \mathcal{R}_2 \subseteq \mathcal{R}_1$, avec équivalence si \mathcal{R}_1 et \mathcal{R}_2 ont la même forme. La figure A.3 illustre ces deux cas.

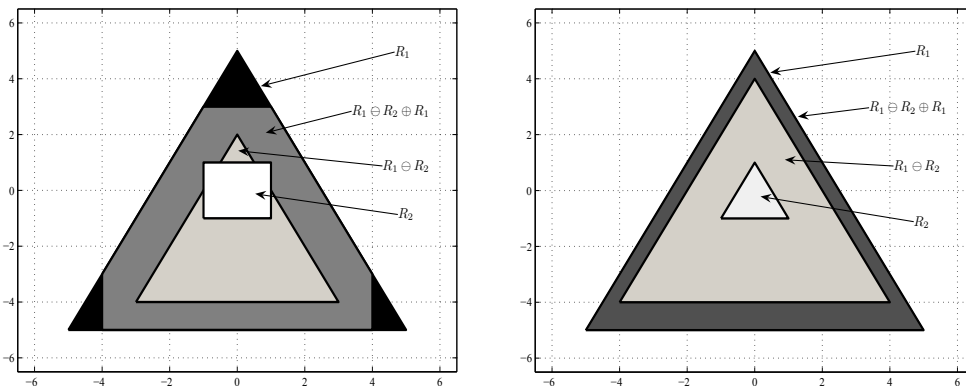


FIGURE A.3 – Illustration géométrique de la formule $\mathcal{R}_1 \ominus \mathcal{R}_2 \oplus \mathcal{R}_2 \subseteq \mathcal{R}_1$ et pour l'équivalence dans \mathbb{R}^2 .

Soustraction entre deux polytopes

La soustraction entre deux polytopes, \mathcal{R}_1 et \mathcal{R}_2 , est définie par :

$$\mathcal{R}_1 \setminus \mathcal{R}_2 = \{x | x \in \mathcal{R}_1 \wedge x \notin \mathcal{R}_2\} \quad (\text{A.8})$$

Intersection et union de deux polytopes

L'intersection et l'union de deux polytopes, \mathcal{R}_1 et \mathcal{R}_2 , sont définies respectivement par :

$$\mathcal{R}_1 \cap \mathcal{R}_2 = \{x | x \in \mathcal{R}_1 \wedge x \in \mathcal{R}_2\} \quad (\text{A.9})$$

$$\mathcal{R}_1 \cup \mathcal{R}_2 = \{x | x \in \mathcal{R}_1 \vee x \in \mathcal{R}_2\} \quad (\text{A.10})$$

A.2 Enveloppe convexe d'un ensemble de polytopes

On définit l'enveloppe convexe (nommée `hull`) de l'union de n_r polytopes comme :

$$\text{hull}\left(\bigcup_{i=1}^{n_r} \mathcal{R}_i\right) = \left\{x \mid x = \sum_{i=1}^{n_r} \alpha_i x_i, x_i \in \mathcal{R}_i, 0 \leq \alpha_i \leq 1, \sum_{i=1}^{n_r} \alpha_i = 1\right\} \quad (\text{A.11})$$

Il contient le plus petit ensemble convexe qui contient les régions \mathcal{R}_i .

A.3 Image et image inverse d'un polytope

Considérons le système linéaire PWA défini sur n_r régions :

$$x(k+1) = f_{\text{PWA}}(x(k)) = A_i x(k) + a_i \quad x(k) \in \mathcal{R}_i \quad \text{avec } i = 1, \dots, n_r \quad (\text{A.12})$$

défini dans un ensemble compact de l'espace d'état $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i \subset \mathbb{R}^n$ avec $A_i \in \mathbb{R}^{n \times n}$ et $a_i \in \mathbb{R}^{n \times 1}$.

On définit l'image d'une région \mathcal{R}_i par sa dynamique i comme :

$$\text{image}(\mathcal{R}_i) = \{A_i x + a_i \mid x \in \mathcal{R}_i\} \quad (\text{A.13})$$

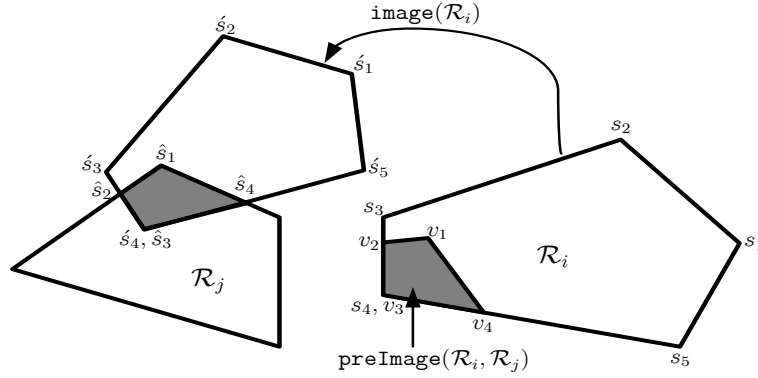
Pour un système PWA défini sur $\mathbb{X} = \bigcup_{i=1}^{n_r} \mathcal{R}_i$ on définit :

$$\text{image}(\mathbb{X}) = \text{image}\left(\bigcup_{i=1}^{n_r} \mathcal{R}_i\right) = \bigcup_{i=1}^{n_r} \text{image}(\mathcal{R}_i) \quad (\text{A.14})$$

L'image inverse d'une région \mathcal{R}_j par la dynamique valide dans la région \mathcal{R}_i est donnée par :

$$\text{preImage}(\mathcal{R}_i, \mathcal{R}_j) = \{x \in \mathcal{R}_i \mid (A_i x + a_i) \in \mathcal{R}_j\} \quad (\text{A.15})$$

L'algorithme 11 permet de calculer le polytope $\text{preImage}(\mathcal{R}_i, \mathcal{R}_j)$. L'inconvénient de cet algorithme est qu'il utilise, à l'étape 5, l'inverse de la matrice A ce qui peut poser problème si

FIGURE A.4 – Illustration géométrique de l'image et pré-Image de \mathcal{R}_i et \mathcal{R}_j .**Algorithme 11** Calcul de $preImage(\mathcal{R}_i, \mathcal{R}_j)$ **ENTRÉES:** polytopes $\mathcal{R}_i, \mathcal{R}_j$ et la dynamique qui correspond à \mathcal{R}_i à savoir $A_i x + a_i$

- 1: $[\hat{s}_1 \hat{s}_2 \dots \hat{s}_{m_i}] \leftarrow A_i [s_1 s_2 \dots s_{m_i}] + [a_i a_i \dots a_i] \quad \{\mathbf{image}(\mathcal{R}_i)\}$
- 2: $Poly_1 \leftarrow$ polytope construit à partir des sommets $[\hat{s}_1 \hat{s}_2 \dots \hat{s}_{m_i}]$
- 3: $Poly_2 \leftarrow \mathcal{R}_j \cap Poly_1$
- 4: $[\hat{s}_1 \hat{s}_2 \dots \hat{s}_s] \leftarrow$ les sommets du polytope $Poly_2$
- 5: $[v_1 v_2 \dots v_s] \leftarrow A_i^{-1}([\hat{s}_1 \hat{s}_2 \dots \hat{s}_s] - [a_i a_i \dots a_i])$

SORTIES: $preImage(\mathcal{R}_i, \mathcal{R}_j) =$ polytope construit à partir des sommets $[v_1 v_2 \dots v_s]$

cette matrice est mal conditionnée. Pour surmonter cet inconvénient, on utilise la formulation suivante pour calculer $preImage(\mathcal{R}_i, \mathcal{R}_j)$.

L'ensemble $preImage(\mathcal{R}_i, \mathcal{R}_j) \neq \emptyset$ vérifie :

$$x(k) \in \mathcal{R}_i \Rightarrow x(k+1) \in \mathcal{R}_j \quad (\text{A.16})$$

$$H_i x(k) \leq h_i \Rightarrow H_j x(k+1) \leq h_j \quad (\text{A.17})$$

$$\Rightarrow H_j (A_i x(k) + a_i) \leq h_j \quad (\text{A.18})$$

$$\Rightarrow H_j A_i x(k) \leq h_j - H_j a_i \quad (\text{A.19})$$

donc on peut définir $preImage(\mathcal{R}_i, \mathcal{R}_j)$ comme :

$$preImage(\mathcal{R}_i, \mathcal{R}_j) = \left\{ x \mid \begin{bmatrix} H_i \\ H_j A_i \end{bmatrix} x \leq \begin{bmatrix} h_i \\ h_j - H_j a_i \end{bmatrix} \right\} \quad (\text{A.20})$$

et on définit aussi

$$preImage(\mathcal{R}_i, \mathbb{X}) = preImage(\mathcal{R}_i, \bigcup_{j=1}^{n_r} \mathcal{R}_j) = \bigcup_{j=1}^{n_r} preImage(\mathcal{R}_i, \mathcal{R}_j) \quad (\text{A.21})$$

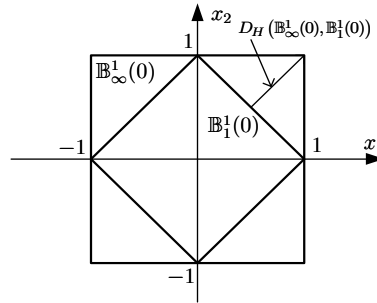


FIGURE A.5 – Distance de Hausdorff entre $\mathbb{B}_\infty^1(0)$ et $\mathbb{B}_1^1(0)$

A.4 Distance de Hausdorff

La distance de Hausdorff D_H entre deux ensembles \mathbb{X}_1 et \mathbb{X}_2 est définie par :

$$D_H(\mathbb{X}_1, \mathbb{X}_2) = \max(dh(\mathbb{X}_1, \mathbb{X}_2), dh(\mathbb{X}_2, \mathbb{X}_1)) \quad (\text{A.22})$$

avec $dh(\cdot, \cdot)$ représentant la distance de Hausdorff orientée :

$$dh(\mathbb{X}_1, \mathbb{X}_2) = \max_{x_1 \in \mathbb{X}_1} d(x_1, \mathbb{X}_2) \quad (\text{A.23})$$

d représente la distance entre le point x_1 et l'ensemble \mathbb{X}_2 . On utilise la norme 2 pour calculer cette distance :

$$dh(\mathbb{X}_1, \mathbb{X}_2) = \max_{x_1 \in \mathbb{X}_1} \min_{x_2 \in \mathbb{X}_2} \|x_1, \mathbb{X}_2\|_2 \quad (\text{A.24})$$

$$dh(\mathbb{X}_1, \mathbb{X}_2) = \max_{x_1 \in \mathbb{X}_1} \min_{x_2 \in \mathbb{X}_2} \frac{1}{2} x_2^T \mathbb{I}_n x_2 - x_1^T x_2 \quad (\text{A.25})$$

Si \mathbb{X}_1 et \mathbb{X}_2 sont deux polytopes, alors le terme \max dans (A.23) est remplacé par les coordonnées des sommets de \mathbb{X}_2 . Par exemple la distance de Hausdorff entre les deux ensembles $\mathbb{B}_\infty^1(0)$ et $\mathbb{B}_1^1(0)$ comme il est montré sur la figure A.5 est $D_H(\mathbb{B}_\infty^1(0), \mathbb{B}_1^1(0)) = 0.7071$.

On rappelle les propriétés d'une distance (de Hausdorff) :

- $D_H(\mathbb{X}, \mathbb{X}) = 0$,
- si $\mathbb{X}_1 \subseteq \mathbb{X}_2$, alors $dh(\mathbb{X}_1, \mathbb{X}_2) = 0$ et $D_H(\mathbb{X}_2, \mathbb{X}_1) \geq 0$, et $D_H(\mathbb{X}_1, \mathbb{X}_2) \geq 0$
- si $d_1 = D_H(\mathbb{X}_1, \mathbb{X})$ et $d_2 = dh(\mathbb{X}_2, \mathbb{X})$, alors $d_3 = D_H(\mathbb{X}_1, \mathbb{X}_2) \leq d_1 + d_2$.

Par la suite, l'étude portera sur le cas particulier $\mathbb{X}_1 \subseteq \mathbb{X}_2$. Dans le cas où les ensembles $\mathbb{X}_1 = \bigcup_{i=1}^{n_r} \mathcal{R}_i$ et $\mathbb{X}_2 = \bigcup_{i=1}^{m_r} \tilde{\mathcal{R}}_i$ sont définis par l'union de polytopes, le calcul de la distance de Hausdorff sera effectué par l'algorithme 12.

Algorithme 12 Calcul $D_H(\mathbb{X}_1, \mathbb{X}_2)$, avec $\mathbb{X}_1 = \bigcup_{i=1}^{n_r} \mathcal{R}_i$, $\mathbb{X}_2 = \bigcup_{i=1}^{n_r} \tilde{\mathcal{R}}_i$, et $\mathbb{X}_1 \subseteq \mathbb{X}_2$.

ENTRÉES: \mathbb{X}_1 et \mathbb{X}_2 .

- 1: $v \leftarrow$ ensemble de tous les sommets des polytopes \mathcal{R}_i
- 2: $res_2 \leftarrow []$
- 3: **pour** $ind_1 = 1$ à cardinal de v **faire**
- 4: $res_1 \leftarrow []$
- 5: **pour** $ind_2 = 1$ à m_r **faire**
- 6: Pour chaque région $\tilde{\mathcal{R}}(ind_2)$, on calcule la distance qui la sépare du sommet $v(ind_1)$.
On utilise la norme 2, pour calculer cette distance, ce qui revient à résoudre un problème d'optimisation quadratique : $d \leftarrow \min_{x \in \tilde{\mathcal{R}}(ind_2)} \frac{1}{2} x^T \mathbb{I}_n x - v^T(ind_1)x$.
- 7: $res_1 \leftarrow [res_1, d]$
- 8: **fin pour**
- 9: $res_2 = [res_2, \min(res_1)]$
- 10: **fin pour**

SORTIES: $D_H(\mathbb{X}_1, \mathbb{X}_2) = \max(res_2)$.

Annexe B

Comparaison de temps de calcul avec et sans analyse par intervalles

Dans cet annexe, on présente un exemple pour voir la différence entre l'utilisation de la construction contractive avec et sans analyse par intervalles.

Exemple 19. Considérons le système discret suivant :

$$x(k+1) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k) \quad (\text{B.1})$$

avec les contraintes :

$$\begin{bmatrix} \mathbb{I}_2 \\ \mathbb{I}_2 \end{bmatrix} x(k) \leq 10 \begin{bmatrix} \mathbb{1}_{2,1} \\ -\mathbb{1}_{2,1} \end{bmatrix}$$
$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} u(k) \leq \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

On calcule une loi de commande prédictive pour le système (B.1), avec les horizons de prédiction et de contrôle $N = 5$, $N_u = 5$. Les matrices de pondérations sont $Q = \mathbb{I}_2$ et $R = 0.1$. On obtient une solution explicite [33] qui transforme le système (B.1) en un système PWA autonome. La figure B.1 illustre graphiquement le calcul de l'ensemble invariant. L'algorithme 1 avec et sans utilisation d'analyse par intervalles met 16 itérations pour trouver l'ensemble invariant exact. En utilisant l'analyse par intervalles l'algorithme 1 nécessite 139.1s pour trouver l'ensemble invariant exact alors que sans analyse par intervalles il nécessite 372.33s.

Le tableau B.1 illustre les différents temps de calculs ainsi que le nombre d'opérations effectuées pour quelques itérations.

On observe dans le tableau B.1 que l'analyse par intervalles diminue le nombre d'opérations géométriques et par conséquent entraîne une diminution du temps de calcul.

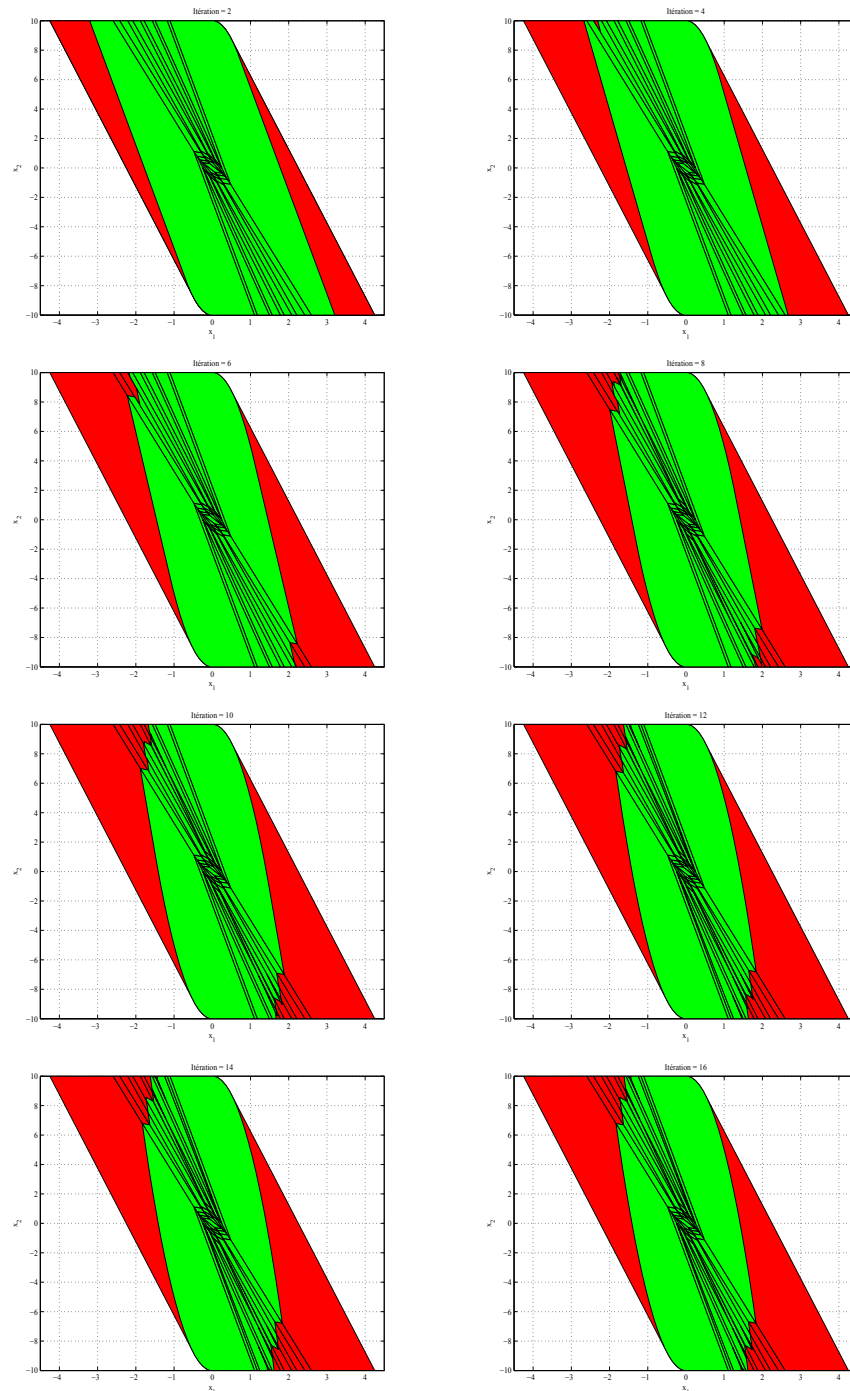


FIGURE B.1 – Illustration graphique de la construction contractive de l'ensemble invariant.

nombre d'itérations	1	8	11	12	16
nombre de régions	51	63	72	75	75
nbr. d'opérations sans analyse par intervalles	2601	3969	5184	5625	5625
nbr. d'opérations avec analyse par intervalles	1127	1243	1409	1584	1665

TABLE B.1 – Comparaison entre le nombre d'opérations (*image* et *preImage*) effectuées par l'algorithme 1 avec et sans utilisation de l'analyse par intervalles.

Références

- [1] M. Gevers, V. Wertz, and R. Bitmead, *Adaptive Optimal Control : The Thinking Man's Gpc*. Prentice Hall, 1990.
- [2] S. Qin and T. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.
- [3] E. F. Camacho and C. A. Bordons, *Model predictive control in the process industry*. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1997.
- [4] ———, *Model Predictive Control*. Springer Verlag, 2nd edition, 2004.
- [5] D. Clarke, C. Mohtadi, and P. Tuffs, "Generalized predictive control, parts 1 and 2," *Automatica*, vol. 23, no. 2, pp. 137–160, 1987.
- [6] J. A. Rossiter, *Model-based predictive control. A paractical approach*. CRC Press, 2003.
- [7] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.
- [8] G. C. Goodwin, M. M. Seron, and J. A. D. Dona, *Constrained Control and Estimation*. Springer, 2004.
- [9] J. Richalet, *Pratique de la commande prédictive*. Hermes, 1993.
- [10] J. Richalet and D. O'Donovan, *Predictive Functional Control*. Springer London, 2009.
- [11] P. Boucher and D. Dumur, *La commande prédictive*. Editions TECHNIP, 1996.
- [12] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control : Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.
- [13] P. Boucher and D. Dumur, *La commande prédictive avancées et perspectives*. Hermes Science, 2006.
- [14] B. Kouvaritakis and M. Cannon, *Nonlinear predictive control : theory and practice*. IEE, London, UK, 2001.
- [15] R. Findeisen, F. Allgöwer, and L. Biegler, *Assessment and Future Directions of Nonlinear Model Predictive Control*. Springer, 2007.
- [16] L. Magni, D. Raimondo, and F. Allgöwer, *Nonlinear Model Predictive Control*. Springer, 2009.
- [17] F. Allgöwer and A. Zheng, *Nonlinear Model Predictive Control*. Springer, 2000.
- [18] V. Dua, E. N. Pistikopoulos, and M. C. Georgiadis, *Multi-Parametric Model-Based Control : Theory and Applications*. Germany : Wiley-VCH Verlag, 2007.

- [19] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [20] M. Seron, G. Goodwin, and J. Doná, “Characterisation of receding horizon control for constrained linear systems,” *Asian Journal of Control*, vol. 5, no. 2, pp. 271–286, 2003.
- [21] S. Olaru, “Contribution à l’étude de la commande prédictive sous contraintes par approche géométrique,” Ph.D. dissertation, Supélec, France, 2005.
- [22] S. Olaru and D. Dumur, “Avoiding constraints redundancy in predictive control optimization routines,” *IEEE Transactions on Automatic Control*, vol. 50(9), pp. 1459–1466, 2005.
- [23] J. Mare and J. D. Dona, “Use of dynamic programming for the analytical solution of input-constrained LQR problems,” in *5th IEEE Asian Control Conference ASCC*, vol. 4, 2004, pp. 20–23.
- [24] F. Blanchini, “Set invariance in control,” *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.
- [25] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Birkhäuser, 2008.
- [26] E. G. Gilbert and K. T. Tan, “Linear systems with state and control constraints, the theory and application of maximal output admissible sets,” *IEEE Transaction on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.
- [27] D. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995.
- [28] J. Aubin, *Viability Theory*. Birkhäuser Boston, Basel, Berlin, 2006.
- [29] V. Dua and E. N. Pistikopoulos, “An algorithm for the solution of multiparametric mixed integer linear programming problems,” in *Annal of Operations Research*, vol. 99, no. 1-4, 2000, pp. 123–139.
- [30] J. Mare and J. D. Dona, “Analytical solution of constrained reference tracking problems by dynamic programming,” in *Proceedings of the IEEE Conference on Decision and Control and European Control Conference*, 2005.
- [31] F. Borelli, *Constrained Optimal Control of Linear and Hybrid Systems*. Springer-Verlag, 2003.
- [32] P. Tøndel, T. A. Johansen, and A. Bemporad, “An algorithm for mpqp and explicit mpc solutions,” *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.
- [33] M. Kvasnica, P. Grieder, and M. Baotić, “Multi-parametric toolbox (mpt),” 2004. [Online]. Available : <http://control.ee.ethz.ch/mpt/>
- [34] C. N. Jones, E. C. Kerrigan, and M. Maciejowski, “Equality set projection : A new algorithm for the projection of polytopes in halfspace representation,” Department of Engineering, University of Cambridge, Tech. Rep., 2004.
- [35] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control : Stability and optimality,” *Automatica*, vol. 36, pp. 789–814, 2000.
- [36] D. Liberzon, *Switching in systems and control*. Birkhäuser, 2003.
- [37] A. Bemporad and M. Morari, “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

- [38] A. Bemporad, G. Ferrari-Trecate, and M. Morari, “Observability and controllability of piecewise affine and hybrid systems,” *IEEE Transaction on Automatic Control*, vol. 45, no. 10, pp. 1864–1876, 2000.
- [39] A. Bemporad, F. D. Torrisi, and M. Morari, *Optimization-based verification and stability characterization of piecewise affine and hybrid systems*, ser. Hybrid systems : Computation and control, LNCS. Springer-Verlag, 2000, vol. 1790, pp. 45–58.
- [40] M. Johansson and A. Rantzer, “Computation of piecewise quadratic lyapunov functions for hybrid systems,” *IEEE Transaction on Automatic Control*, vol. 43, no. 4, pp. 555–559, 1998.
- [41] W. Heemels, J. M. Schumacher, and S. Weiland, “Linear complementarity systems,” *SIAM Journal of Applied Mathematics*, vol. 60, no. 4, pp. 1234–1269, 2000.
- [42] A. J. V. der Schaft and J. M. Schumacher, “Complementarity modelling of hybrid systems,” *IEEE Transaction on Automatic Control*, vol. 43, no. 4, pp. 483–490, 1998.
- [43] W. Heemels, B. D. Schutter, and A. Bemporad, “Equivalence of hybrid dynamical models,” *Automatica*, vol. 37, no. 7, pp. 1085–1091, 2001.
- [44] M. Johansson, *Piecewise Linear Control Systems, A Computational Approach*, ser. Lecture Notes in Control and Information Sciences 284. Springer-Verlag, 2003.
- [45] A. F. Filippov and F. M. Arscott, *Differential Equations with Discontinuous Right Sides : Control Systems*. Springer, 1988.
- [46] S. Sastry, *Nonlinear Systems Analysis, Stability, and Control*. Springer-Verlag, 1999.
- [47] V. Rasvan and R. Stefan, *Systèmes non linéaires - théorie et applications*. Hermes, 2007.
- [48] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics, 1994.
- [49] J. Löfberg, “Yalmip : A toolbox for modeling and optimization in matlab,” in *Proceedings of the CACSD Conference*, 2004. [Online]. Available : <http://control.ee.ethz.ch/~joloef/yalmip.php>
- [50] D. Mignone, G. Ferrari-Trecate, and M. Morari, “Stability and stabilization of piecewise affine and hybrid systems : An lmi approach,” in *Proceedings of the 39th IEEE Conference on Decision and Control*, 2000, pp. 504–509.
- [51] H. Benlaoukli and S. Oлару, “On the computation and bounding of invariant sets,” in *International Conference on Control, Automation and Systems ICCAS*, Seoul, Korea, 2007.
- [52] —, “Computation and bounding of robust invariant sets for uncertain systems,” in *Proceedings of the IFAC World Congress*, Seoul, Korea, 2008.
- [53] —, “Model predictive control - numerical methods for the invariant sets approximation,” in *Numerical Analysis and Its Applications*, Lozentsz, Bulgaria, 2008.
- [54] H. Benlaoukli, M. Hovd, S. Oлару, and P. Boucher, “Invariant sets for PWA systems using transition graph,” in *7th IEEE International Conference on Control and Automation ICCA*, Nouvelle Zélande, Décembre 2009.
- [55] H. Benlaoukli and S. Oлару, *Model Predictive Control - Numerical Methods for the Invariant Sets Approximation*. Berlin, Heidelberg : Springer-Verlag, 2009, pp. 192–199.

- [56] A. Alessio, M. Lazar, A. Bemporad, and W. Heemels, “Squaring the circle : An algorithm for generating polyhedral invariant sets from ellipsoidal ones,” *Automatica*, vol. 43, no. 12, pp. 2096–2103, 2007.
- [57] M. deBerg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry, Algorithms and Applications*. Springer, 2000.
- [58] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, “Invariant approximations of the minimal robust positively invariant set,” *IEEE Transaction on Automatic Control*, vol. 50, pp. 406–410, 2005.
- [59] S. Oлару, J. A. Dedona, and M. Seron, “Positive invariant sets for fault tolerant multisensor control schemes,” in *Proceedings of the IFAC World Congress*, Seoul, Korea, 2008.
- [60] H. Benlaoukli, S. Oлару, S. Niculescu, and P. Boucher, “Commande prédictive des systèmes avec l’entrée affectée par retard variable,” in *CIFA*, Roumanie, 2008.
- [61] H. Benlaoukli and S. Oлару, “Feasible tracking reference set computation using the principle of invariant set construction,” in *7th IEEE International Conference on Control and Automation ICCA*, Nouvelle Zélande, Décembre 2009.
- [62] S. Oлару, H. Benlaoukli, and S. Niculescu, “MPC for systems with variable time-delay, robust positive invariant set approximations,” in *ICINCO*, Portugal, 2008.
- [63] H. Benlaoukli, S. Oлару, S. Niculescu, and S. Marinkovic, *Commande prédictive des systèmes avec l’entrée affectée par retard variable*. Bucuresti : Academiei Romane, 2009, pp. 55–61.
- [64] H. Benlaoukli, M. Hovd, and S. Oлару, “Geometrical characterization of robust predictive control strategies,” *CEAI*, vol. 10, no. 4, pp. 13–19, 2008.
- [65] P. O. M. Scockaeter and D. Clarke, *Advances in model-based predictive control*. Oxford University Press, 1994, ch. Stability and feasibility in constrained predictive control, pp. 217–230.
- [66] S. Oлару and D. Dumur, “Feasibility analysis of constrained predictive control,” in *Proceedings of the 14th Conference on Control Systems and Computer Sciences*, Bucharest, 2003, pp. 164–169.
- [67] V. B. Larin, “About the inverse problem of optimal control,” *Journal of Applied and Computational Mathematics*, vol. 2, pp. 90–97, 2003.
- [68] M. V. Kothare, V. Balakrishnan, and M. Morari, “Robust constrained model predictive control using linear matrix inequalities,” *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.
- [69] C. Lana and M. Rotea, “Desensitized model predictive control,” in *Proceedings of the 15th Mediterranean Conference on Control and Automation*, Athens, Greece, 2007.
- [70] S. I. Niculescu, *Delay effects on stability. A robust control approach*. Heidelberg : Springer, 2001.
- [71] K. Gu, V. L. Kharitonov, and J. Chen, *Stability of Time-Delay Systems*. Boston, USA : Birkhäuser, 2003.
- [72] W. Michiels and S. I. Niculescu, *Stability and stabilization of time-delay systems. An eigenvalue based approach*. Philadelphia, USA : SIAM, 2007.

- [73] J. E. Normey-Rico and E. F. Camacho, *Control of Dead-time Processes*. London : Springer, 2007.
- [74] S. Olaru and S. I. Niculescu, “Predictive control for linear systems with delayed input subject to constraints,” in *Proceedings of the IFAC World Congress*, 2008.
- [75] C. Furtmueller and L. del Re, “Disturbance suppression for an industrial level control system with uncertain input delay and uncertain gain,” in *Proceeding of the IEEE Conference on Control Applications*, 2006, pp. 3206–3211.
- [76] S. Olaru and D. Dumur, “A parameterized polyhedra approach for explicit constrained predictive control,” in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2, 2004, pp. 1580–1585.
- [77] A. Bemporad, “Reference governor for constrained nonlinear systems,” *IEEE Transaction on Automatic Control*, vol. 43, no. 3, pp. 415–419, 1998.