# Commande sous contraintes pour des systèmes dynamiques incertains : une approache basée sur l'interpolation

Hoai Nam Nguyen

N° d'ordre : 2012-14-TH

# THÈSE DE DOCTORAT

## DOMAINE : STIC

### SPECIALITE : AUTOMATIQUE

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

*Présentée par :*

**Hoai-Nam NGUYEN**

Sujet:

« Commande sous contraintes pour des systèmes dynamiques incertains : une approche basée sur l'interpolation »

Soutenue le 01/10/2012

devant les membres du jury :

| | | |
|---|---|---|
| **M. Mazen ALAMIR** | Gipsa-lab, Grenoble | Rapporteur |
| **M. Tor Arne JOHANSEN** | NTNU, Trondheim, Norway | Rapporteur |
| **M. Jamal DAAFOUZ** | ENSEM - CRAN, Nancy | Examinateur |
| **M. Didier DUMUR** | SUPELEC | Examinateur |
| **M. Silviu-Iulian NICULESCU** | LSS | Examinateur |
| **M. Sorin OLARU** | SUPELEC | Encadrant |
| **M. Per-Olof GUTMAN** | Tehnion, Israel | Invité |

”Even the best control system cannot make a Ferrari out of a Volkswagen.”

***Skogestad and Postlethwaite***

# Preface

A fundamental problem in automatic control is the control of uncertain plants in the presence of input and state or output constraints. An elegant and theoretically most satisfying framework is represented by optimal control policies which, however, rarely gives an analytical feedback solution, and oftentimes builds on numerical solutions (approximations).

Therefore, in practice, the problem has seen many ad-hoc solutions, such as ***override control***, ***anti-windup***, as well as modern techniques developed during the last decades usually based on state space models. One of the popular example is ***Model Predictive Control*** (MPC) where an optimal control problem is solved at each sampling instant, and the element of the control vector meant for the nearest sampling interval is applied. In spite of the increased computational power of control computers, MPC is at present mainly suitable for low-order, nominally linear systems. The robust version of MPC is conservative and computationally complicated, while the ***explicit*** version of MPC that gives an affine state feedback solution involves a very complicated division of the state space into polyhedral cells.

In this thesis a novel and computationally cheap solution is presented for linear, time-varying or uncertain, discrete-time systems with polytopic bounded control and state (or output) vectors, with bounded disturbances. The approach is based on the interpolation between a stabilizing, outer low gain controller that respects the control and state constraints, and an inner, high gain controller, designed by any method that has a robustly positively invariant set within the constraints. A simple Lyapunov function is used for the proof of closed loop stability.

In contrast to MPC, the new interpolation based controller is not necessarily employing an optimization criterion inspired by performance. In its explicit form, the cell partitioning is simpler that the MPC counterpart. For the implicit version, the on-line computational demand can be restricted to the solution of one linear program or quadratic program.

Several simulation examples are given, including uncertain linear systems with output feedback and disturbances. Some examples are compared with MPC. The control of a laboratory ball-and-plate system is also demonstrated. It is believed that the new controller might see wide-spread use in industry, including the automotive industry, also for the control of fast, high-order systems with constraints.

Place(s), month year

***SUPELEC, Gif sur Yvette***
***October 3, 2012***

# Acknowledgements

This thesis is the result of my interaction with a large number of people, which on a personal and scientific level helped me during these last three years.

I would like first to thank my supervisor, Prof. Sorin Olaru, for his enthusiastic support, and for creating and maintaining a creative environment for research and studies, and making this thesis possible. To him and his close collaborator, Prof. Morten Hovd, I am grateful for their help which helped me to take the first steps into becoming a scientist.

Big thanks and a big hug goes to Per-Olof Gutman, who has been part of almost all the research in the thesis, and has worked as my supervisor, and co-authored most of the papers. Thanks to him, I believe that I learned to think the right way about some problems in control, which I would not be able to approach appropriately otherwise.

I would like to thank to Prof. Didier Dumur, who was my supervisor for the first two years of my PhD. I must thank to him for enabling my research, and to offer helpful suggestions for the thesis.

In the Automatic Control department of Supelec, I would like to thank especially to Prof. Patrick Boucher and Ms. Josiane Dartron for their support. I would also like to thank to all the people I have met in the department which made my stay here interesting and enjoyable. Without giving an exhaustive list, some of them are Hieu, Florin, Bogdan, Raluca, Christina, Ionela, Nikola, Alaa.

Finally, I am grateful to my wife Linh and my son Tung, who trusted and supported me in several ways and made this thesis possible.

# Résumé étendu de la thèse

## Introduction

Un problème fondamental à résoudre en Automatique réside dans la commande des systèmes incertains qui présentent des contraintes sur les variables de l'entrée, de l'état ou la sortie. Ce problème peut être théoriquement résolu au moyen d'une commande optimale. Cependant la commande optimale (en temps minimal) par principe n'est pas une commande par retour d'état ou retour de sortie et offre seulement une trajectoire optimale le plus souvent par le biais d'une solution numérique.

Par conséquent, dans la pratique, le problème peut être approché par de nombreuses méthodes, tels que "commande over-ride" et "anti-windup". Une autre solution, devenu populaire au cours des dernières décennies est la commande prédictive. Selon cette méthode, un problème de la commande optimale est résolu à chaque instant d'échantillonnage, et le composant du vecteur de commande destiné à l'échelon curant est appliquée. En dépit de la montée en puissance des architecture de calcul temps-réel, la commande prédictive est à l'heure actuelle principalement approprié lorsque l'ordre est faible, bien connu, et souvent pour des systèmes linéaires. La version robuste de la commande prédictive est conservatrice et compliquée à mettre en oeuvre, tandis que la version explicite de la commande prédictive donnant une solution affine par morceaux implique une compartimentation de l'état-espace en cellules polyédrales, très compliquée.

Dans cette thèse, une solution élégante et peu coûteuse en temps de calcul est présentée pour des systèmes linéaire, variants dans le temps ou incertains . Les développements se concentre sur les dynamiques en temps discret avec contraintes polyédriques sur l'entrée et l'état (ou la sortie) des vecteurs, dont les perturbations sont bornées. Cette solution est basée sur l'interpolation entre un correcteur pour la région *extérieure* qui respecte les contraintes sur l'entrée et de l'état, et un autre pour la région intérieure, ce dernier plus agressif, conçue par n'importe quelle méthode classique, ayant un ensemble robuste positivement invariant à l'intérieur des contraintes. Une fonction de Lyapounov simple est utilisée afin d'apporter la preuve de la stabilité en boucle fermée.

Contrairement à la commande prédictive, la nouvelle commande interpolée n'est pas nécessairement fondée sur un critère d'optimisation. Dans sa forme explicite, la partition de l'espace d'état est plus simple que celle de la commande prédictive. Pour la version implicite, la demande de calcul en ligne peut se limiter à la solution d'un ou de deux programmes linéaires.

On donne plusieurs exemples de simulation, y compris pour les systèmes linéaires incertains avec retour de sortie et les perturbations. On donne quelques à titre de comparaison avec la commande prédictive. Une application de ce type de commande a été commande expérimentée pour un système de positionnement d'une bille sur une plaque.

Nous pensons que la nouvelle commande peut être largement utilisée dans l'industrie, y compris l'industrie automobile, ainsi que pour la commande de systèmes d'ordres élevées avec contraintes.

## Contraintes

Les contraintes sont présentes dans tous les systèmes dynamiques du monde réel. Elles rendent plus complexe la synthèse de lois des commandes, non seulement en théorie, mais aussi dans le domaine des applications pratiques. Du point de vue conceptuel, les contraintes peuvent être de nature différente. Fondamentalement, il existe deux types de contraintes imposées par les limitations physiques et/ou la performance souhaitée.

- Les contraintes physiques sont dues aux limitations physiques de la partie mécanique, électrique, biologique, actionneur, etc. La principale préoccupation ici est la stabilité du système en présence des contraintes sur les variables d'entrée et de sortie ou sur l'état. Les variables d'entrée et de sortie doivent rester à l'intérieur des contraintes afin d'éviter la surexploitation ou de dommages. En outre, la violation des contraintes peut conduire à une dégradation de performance, à des oscillations ou même à l'instabilité.
- Les contraintes de performance sont introduites lors de la synthèse afin de satisfaire les exigences de performance, par exemple le temps de montée, le temps de premier maximum, la tolérance aux pannes, la longévité des équipements et des problèmes environnementaux, etc.

### *Contraintes sur l'entrée*

Une classe de contraintes couramment imposées tout au long de cette thèse sont les contraintes sur l'entrée considérées dans le but d'éviter la saturation

- Contraintes sur la norme Euclidienne de la commande

$$\|u(k)\|_2 \leq u_{max} \tag{0.1}$$

où $u(k) \in \mathbb{R}^m$ est la commande du système.
– Contraintes polyédrales sur la commande

$$u(k) \in U, U = \{u \in \mathbb{R}^m : F_u u(k) \leq g_u\} \tag{0.2}$$

où le matrice $F_u$ et le vecteur $g_u$ sont supposés être constants avec $g_u > 0$ de sorte que l'origine est contenue à l'intérieur de $U$.

## *Contraintes sur la sortie*

Une autre classe de contraintes présentes dans ce manuscrit sont les contraintes sur la sortie.
– Contraintes sur la norme Euclidienne de la sortie

$$\|y(k)\|_2 \leq y_{max} \tag{0.3}$$

où $y(k) \in \mathbb{R}^p$ est la sortie du système.
– Contraintes polyédrales sur la sortie

$$y(k) \in Y, Y = \{y \in \mathbb{R}^p : F_y y(k) \leq g_y\} \tag{0.4}$$

où le matrice $F_y$ et le vecteur $g_y$ sont supposés être constants avec $g_y > 0$ de sorte que l'origine est contenue à l'intérieur de $Y$.

## *Contraintes sur l'état*

Une dernière classe de contraintes présentes dans ce mémoire sont les contraintes sur l'état.
– Contraintes sur la norme Euclidienne de l'état

$$\|x(k)\|_2 \leq x_{max} \tag{0.5}$$

où $x(k) \in \mathbb{R}^n$ est l'état du système.
– Contraintes polyédrales sur l'état

$$x(k) \in X, X = \{x \in \mathbb{R}^n : F_x x(k) \leq g_x\} \tag{0.6}$$

où le matrice $F_x$ et le vecteur $g_x$ sont supposés être constants avec $g_x > 0$ de sorte que l'origine est contenue à l'intérieur de $X$.

## Incertitudes

Le problème de la commande sous contraintes peut devenir encore plus difficile en présence d'incertitudes de modèle, ce qui est inévitable dans la pratique [143], [2]. Les incertitudes du modèle apparaissent dans certains cas spécifiques, par exemple quand un modèle linéaire est obtenu par une approximation d'un système non linéaire autour d'un point de fonctionnement. Même si ce processus est assez bien représenté par un modèle linéaire, les paramètres du modèle pourraient être variants dans le temps ou pourraient changer en raison d'un changement des points de fonctionnement. Dans ces cas, la cause et la structure des incertitudes du modèle sont assez bien connus. Néanmoins, même si le processus réel est linéaire, il y a toujours une certaine incertitude associée, par exemple, à des paramètres physiques, qui ne sont jamais connus exactement. En outre, les processus réels sont généralement affectés par des perturbations et il est nécessaire de les prendre en compte dans la conception de la lois de commande.

Il est généralement admis que la principale raison de l'asservissement est de diminuer les effets de l'incertitude, qui peut apparaître sous différentes formes telles que les incertitudes paramétriques, les perturbations additives insuffisances dans les modèles utilisés servant à la conception de l'asservissement. L'incertitude du modèle et sa robustesse ont été un thème central dans le développement de lois de commande en automatique [9].

## Une perspective historique

Une manière simple, qui permet de stabiliser un système sous contraintes est de réaliser la conception de la lois de commande sans tenir compte des contraintes. Puis, une adaptation de la loi de commande est considérée en considérant la saturation d'entrée. Une telle approche est appelée *anti-windup* [79], [152], [150], [158], plus récemment traitée dans [142].

Au cours des dernières décennies, la recherche concernant le sujet de la commande sous contraintes n'a pu déboucher que dans la mesure où ces contraintes ont pu être prises en compte lors de la phase de synthèse. Par son principe, la commande prédictive révèle toute son importance dans le traitement des contraintes [32], [100], [28], [129], [47], [96], [48]. Dans l'approche de la *commande prédictive*, une séquence de valeurs prédites de commande sur un horizon de prédiction finie est calculée afin d'optimiser la performance du système en boucle fermée, exprimée en termes d'une fonction de coûts [3]. La commande utilise un modèle mathématique interne qui, compte tenu des mesures actuelles, prédit le comportement futur du système réel en fonction du changement des entrées de commande. Une fois que la séquence optimale d'entrées de la commande a été calculé, seul le premier élément est effectivement appliquée au système et l'optimisation est répétée à l'instant suivant avec la nouvelle mesure de l'état [5], [100], [96].

Dans la commande prédictive classique, l'action d'entrée à chaque instant est obtenue en résolvant en ligne le problème de commande optimale en boucle ouverte [126], [33]. Avec un modèle linéaire, des contraintes polyédrales, et un coût quadratique, le problème d'optimisation est un programme quadratique. La résolution du programme quadratique peut être coûteuse en temps de calcul, surtout quand l'horizon de prédiction est grand, ce qui a traditionnellement limitée la commande prédictive aux applications avec période d'échantillonnage relativement faible [4].

Dans la dernière décennie, des tentatives ont été faites pour utiliser la commande prédictive aux processus rapides. Dans [122], [121], [20], [154] il a été montré que la commande prédictive sous contraintes est équivalente à un problème d'optimisation multi-paramétrique, où l'état joue le rôle d'un vecteur de paramètres. La solution est une fonction affine par morceaux de l'état sur une partition polyédrale de l'espace d'état, et l'effort de calcul de la commande prédictive est déplacé hors-ligne. Cependant, la commande prédictive explicite a aussi des inconvénients. L'obtention de la solution optimale explicite nécessite de résoudre un problème hors-ligne d'optimisation paramétrique, qui est généralement un problème NP-difficile. Bien que le problème soit traitable et pratiquement résoluble pour plusieurs applications de intéressantes de l'automatique, l'effort hors-ligne de calcul croît *exponentiellement* plus vite même que l'augmentation de la taille du problème [84], [85], [83], [64], [65]. C'est le cas d'une grand horizon de prédiction, d'un grand nombre de contraintes et de systèmes de grande dimension.

Dans [157], les auteurs montrent que le calcul en ligne est préférable dans le cas des systèmes de grande dimension où la réduction significative de la complexité de calcul peut être obtenue en exploitant la structure particulière du problème d'optimisation à partir d'une solution obtenue à l'étape précédente de temps. La même référence mentionne que pour les modèles de plus de cinq dimensions, la solution explicite pourrait ne pas être pratique. Il convient de mentionner que les solutions approchées explicites ont été étudiées pour aller au-delà de cette limitation ad-hoc [19], [62], [140].

Notez que, comme son nom l'indique, les commandes prédictives implicites et explicites classiques sont basées sur des modèles mathématiques qui, invariablement, présentent un décalage par rapport aux systèmes physiques. La commande prédictive robuste est conçue pour couvrir à la fois l'incertitude des modèles et des perturbations. Cependant, le robuste MPC présente un grand conservatisme et/ou grand complexité de de calcul [78], [87].

L'utilisation de l'interpolation dans la commande sous contraintes permettent d'éviter des procédures très complexes de synthèse est bien connue dans la littérature. Il y a une longue tradition de développements sur ces sujets étroitement liés à la commande prédictive, voir par exemple [11], [132], [133], [131]. En effet, l'interpolation entre les séquences d'entrée, l'état des trajectoires, les gains de correcteurs et/ou des ensembles associés aux ensembles invariants, peut être déterminée.

## Formulation du problème

Considérons le problème de la régulation d'un système linéaire variant dans le temps ou incertain

$$x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \tag{0.7}$$

où $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $w(k) \in \mathbb{R}^d$ sont respectivement, le vecteur d'état, le vecteur des entrées et le vecteur des perturbations. Les matrices du système $A(k) \in \mathbb{R}^{n \times n}$, $B(k) \in \mathbb{R}^{n \times m}$ et $D(k) \in \mathbb{R}^{n \times d}$ satisfont

$$\begin{cases} A(k) = \sum\limits_{i=1}^{q} \alpha_i(k) A_i, \quad B(k) = \sum\limits_{i=1}^{q} \alpha_i(k) B_i, \quad D(k) = \sum\limits_{i=1}^{q} \alpha_i(k) D_i \\ \sum\limits_{i=1}^{q} \alpha_i(k) = 1, \quad \alpha_i(k) \geq 0 \end{cases} \tag{0.8}$$

où les matrices $A_i$, $B_i$ et $D_i$ sont donnés.

Le système est soumis à des contraintes sur l'état, la commande et la perturbation

$$\begin{cases} x(k) \in X, \quad X = \{x \in \mathbb{R}^n : F_x x \leq g_x\} \\ u(k) \in U, \quad U = \{u \in \mathbb{R}^m : F_u u \leq g_u\} \\ w(k) \in W, \quad W = \{w \in \mathbb{R}^d : F_w w \leq g_w\} \end{cases} \tag{0.9}$$

où les matrices $F_x$ et $F_u$, $F_w$ et les vecteurs $g_x$, $g_u$ et $g_w$ sont supposés être constants avec $g_x > 0$, $g_u > 0$ et $g_w > 0$ tel que l'origine est contenue à l'intérieur de $X$, $U$ et $W$. Les inégalités sont valable sur tous ces éléments.

## Ensembles invariants

Avec la théorie de Lyapunov introduite dans le cadre des systèmes régis par des équations différentielles ordinaires, la notion d'ensemble invariant a été utilisée dans de nombreux problèmes concernant l'analyse et la commande des systèmes dynamiques. Une motivation importante ayant conduit à introduire les ensembles invariants est venu du besoin d'analyser l'influence des incertitudes sur le système. Deux types de systèmes seront examinées dans la présente section, à savoir, des systèmes linéaires incertains à temps discret (0.7) et des systèmes autonomes

$$x(k+1) = A_c(k)x(k) + D(k)w(k) \tag{0.10}$$

**Définition 0.1. Ensemble positif invariant robuste** L'ensemble $\Omega \subseteq X$ est dit positif invariant robuste pour le système(0.10) si et seulement si

$$x(k+1) = A_c(k)x(k) + D(k)w(k) \in \Omega$$

pour tout $x(k) \in \Omega$ et pour tout $w(k) \in W$.

Par conséquent, si le vecteur d'état du système (0.10) atteint un ensemble positivement invariant, il restera dans l'ensemble en dépit de la perturbation $w(k)$. Le terme *positivement* fait référence au fait que seulement les évolutions du système (0.10) en temps direct sont considérées. Cet attribut sera omis dans les sections à venir par souci de brièveté.

Étant donné un ensemble borné $X \subset \mathbb{R}^n$, l'ensemble invariant robuste maximal $\Omega_{max} \subseteq X$ est un ensemble invariant robuste, qui contient tous les ensembles invariants robustes contenues dans $X$.

**Définition 0.2. Ensemble contractif robuste** Pour un certain scalaire $\lambda$ avec $0 \leq \lambda \leq 1$, l'ensemble $\Omega \subseteq X$ est $\lambda$-contractif robuste pour le système (0.10) si et seulement si

$$x(k+1) = A_c(k)x(k) + D(k)w(k) \in \lambda\Omega$$

pour tout $x(k) \in \Omega$ et pour tout $w(k) \in W$.

**Définition 0.3. Ensemble invariant robuste contrôlée** L'ensemble $C \subseteq X$ est invariante robuste contrôlée pour le système (0.7) si pour tout $x(k) \in C$, il existe une valeur de commande $u(k) \in U$ tel que

$$x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \in C$$

pour tout $w(k) \in W$.

Étant donné un ensemble borné $X \subset \mathbb{R}^n$, l'ensemble invariant robuste contrôlée maximal $C_{max} \subseteq X$ est un ensemble invariant robuste contrôlée, qui contient tous les ensembles invariants robustes contrôlée contenues dans $X$.

**Définition 0.4. Ensemble contractif robuste contrôlée** Pour un certain scalaire $\lambda$ avec $0 \leq \lambda < 1$, l'ensemble $C \subseteq X$ est robuste contractif contrôlée pour le système (0.7) si pour tout $x(k) \in C$, il existe une valeur de commande $u(k) \in U$ tel que

$$x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \in \lambda C$$

pour tout $w(k) \in W$.

De toute évidence, si le facteur contractif $\lambda = 1$, la question des concepts de l'invariance robuste et l'invariance robuste contrôlée se repose.

Pour les système (0.7) et (0.10) deux types d'ensembles convexes sont largement utilisés pour caractériser le domaine d'attraction. La première classe est celle des ensembles ellipsoïdaux ou ellipsoïdes. Les ensembles ellipsoïdaux sont les plus couramment utilisés dans l'analyse de stabilité robuste et la synthèse des correcteurs des systèmes sous contraintes. Leur popularité est due à l'efficacité des calculs grâce à l'utilisation de formulations LMI et du fait que leur complexité est fixe par rapport à la dimension de l'espace d'état [29], [137]. Cette approche, cependant, peut conduire à des résultats entachés de *conservatisme*.

La seconde classe est celle des ensembles polyédrales. Avec des contraintes linéaires sur les variables d'état et de commande, les ensembles invariants polyédrales

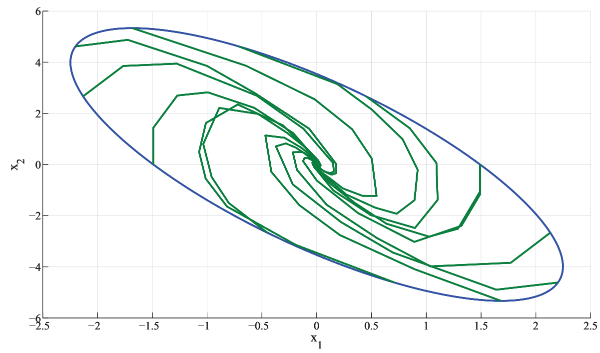



**Fig. 0.1** Ensemble invariant ellipsoïdale.

sont préférés aux ensembles invariants ellipsoïdaux, car ils offrent une meilleure approximation du domaine d'attraction [35], [55], [21]. Leur principal inconvénient est que la complexité de la représentation n'est pas fixée par la dimension de l'espace.

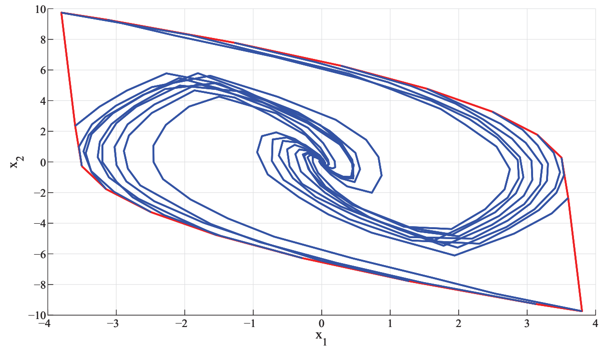
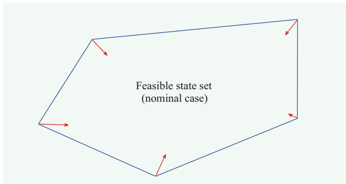


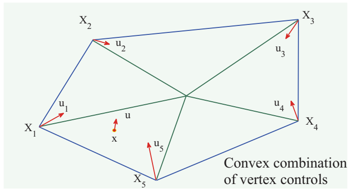

**Fig. 0.2** Ensemble invariant polyédrique.

## Commande aux sommets

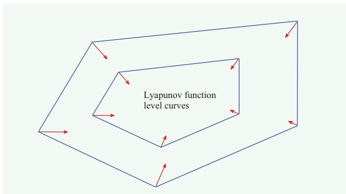
La solution est une extension de la commande aux sommets, développé par Gutman et Cwikel (1986) et prolongée par Blanchini (1992) dans le cas des systèmes incertains. Les conditions nécessaires et suffisantes de stabilisation à l'origine du système (0.7), (0.9) sont que, à chaque sommet de l'ensemble invariant, il existe une commande faisable qui amène l'état de l'intérieur respectif à l'ensemble invariant pour tous les cas d'incertitude ou de variation dans le temps se produisant dans les matrices du système.



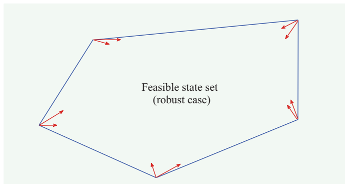

(a) Ensemble invariant faisable




(b) Commande aux sommets




(c) Courbes de niveau des fonctions de Lyapunov




(d) Ensemble invariant faisable - Cas robuste

**Fig. 0.3** Commande sommet.

Le principal avantage du système de la commande aux sommets est la taille du domaine d'attraction, c'est-à-dire l'ensemble $C_N$. Il est clair que l'ensemble invariant contrôlée $C_N$ et le domaine faisable pour la commande, pourrait être aussi grand que tout autres correcteur sous contraintes pourraient avoir. Toutefois, une faiblesse de la commande aux sommets réside dans ce que l'action de commande maximale est appliqué seulement à la frontière de l'ensemble faisable, avec une amplitude de l'action de commande diminué lorsque l'état se rapproche de l'origine. Une solution à ce problème consiste à basculer vers un correcteur local plus agressif qui doit être complété, par exemple, par un mécanisme de type hystérésis afin d'éviter la vibration. Une faiblesse de lois de commande par commutation, c'est que le signal de commande peut changer brusquement.

## Commande interpolée basée sur la programmation linéaire

### *Solution implicite*

Une solution aux faiblesses mentionnées ci-dessus est la loi de commande aux sommets améliorée qui réalise une interpolation entre le signal de commande externe et un correcteur interne plus agressif. Supposons que une loi de commande interne stabilisant a été conçu dont l'ensemble invariant robuste maximal $\Omega_{max}$ est un sous-ensemble de $C_N$.

Rappelons que, le MAS $\Omega_{max}$ est l'ensemble polyédrale maximal pour le quel la loi de commande choisie donne un signal de commande admissible $u_o$ tel que $x(k)$ reste dans $\Omega_{max}$.

Soit $x(k) \in C_N$ décomposé comme

$$x(k) = c(k)x_v(k) + (1 - c(k))x_o(k) \tag{0.11}$$

où $x_v(k) \in C_N$, $x_o \in \Omega_{max}$ et $0 \leq c \leq 1$. Considérons la loi de commande suivante

$$u(k) = c(k)u_v(k) + (1 - c(k))u_o(k) \tag{0.12}$$

où $u_v(k)$ est obtenu en appliquant la loi de commande aux sommets, et $u_o(k) = Kx_o(k)$ est la loi de commande optimale localement, et qui est faisable dans $\Omega_{max}$.

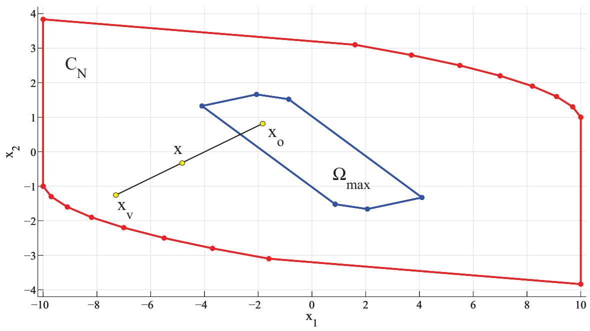



**Fig. 0.4** Commande interpolée. Tout l'état $x(k)$ peut être exprimée comme une combinaison convexe de $x_v(k) \in C_N$ et $x_o(k) \in \Omega_{max}$.

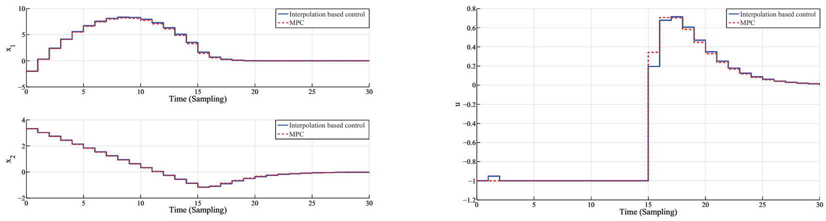
À chaque instant, considérons le problème d'optimisation non linéaire suivant

$$c^* = \min_{c, x_v, x_o} \{c\} \tag{0.13}$$

sujet à

$$\begin{cases} x_v \in C_N, \\ x_o \in \Omega_{max}, \\ cx_v + (1 - c)x_o = x, \\ 0 \leq c \leq 1 \end{cases}$$

On montre que le problème d'optimisation non linéaire (0.13) peut être converti en un problème de programmation linéaire. Il sera prouvé que la loi de commande (0.11), (0.12), (0.13) est faisable et stabilise asymptotiquement le système en boucle fermée avec garanties de robustesse. Le minimum de $c(k)$ est le meilleur choix du point de vue de la commande, car il donne la mesure de l'action de commande, qui se rapproche autant que possible de l'action de commande optimale. Il est en outre montré que le coefficient d'interpolation $c^*$ est une fonction de Lyapunov.




(a) Trajectoires d'etat (b) Trajectoires d'entrée

**Fig. 0.5** Une simulation de la commande interpolée et de la commande prédictive.

## *Solution explicite*

Pour le problème d'optimisation (0.13), les propriétés suivantes peuvent être exploitées :

– Pour tout $x \in \Omega_{max}$, le résultat du problème d'interpolation optimale est la solution triviale $c^* = 0$ et donc $x_o^* = x$ dans (0.13).

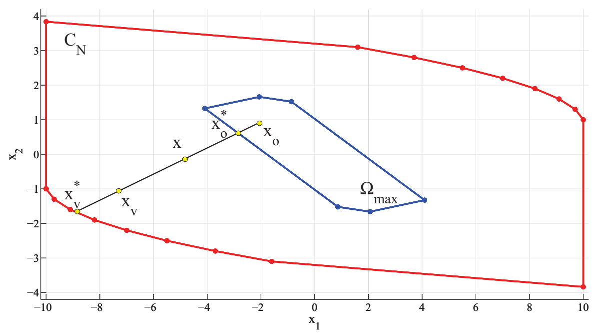



**Fig. 0.6** Illustration graphique. Pour tout $x \in C_N \setminus \Omega_{max}$, la solution optimale du problème (0.13) est atteinte si et seulement si $x$ est écrit comme une combinaison convexe de $x_v$ et $x_o$ où $x_v \in \mathrm{Fr}(C_N)$ et $x_o \in \mathrm{Fr}(\Omega_{max})$.

– Soit $x \in C_N \setminus \Omega_{max}$ avec une combinaison particulière convexe $x = cx_v + (1-c)x_o$ où $x_c \in C_N$ et $x_o \in \Omega_{max}$. Si $x_o$ est strictement à l'intérieur de $\Omega_{max}$, on peut définir
$$\widetilde{x}_o = \mathrm{Fr}(\Omega_{max}) \cap \overline{x, x_o}$$
$\widetilde{x}_o$ comme l'intersection entre la frontière de $\Omega_{max}$ et le segment reliant $x$ et $x_o$. En utilisant des arguments de convexité, on a
$$x = \widetilde{c}x_v + (1-\widetilde{c})\widetilde{x}_o$$
où $\widetilde{c} < c$. D'une manière générale, pour tout $x \in C_N \setminus \Omega_{max}$ l'interpolation optimale (0.13) conduit à une solution $\{x_v^*, x_o^*\}$ avec $x_o^* \in \mathrm{Fr}(\Omega_{max})$.

– D'autre part, si $x_v$ est strictement à l'intérieur de $C_N$, on peut exprimer
$$\widehat{x}_v = \mathrm{Fr}(C_N) \cap \overline{x, x_v}$$
$\widehat{x}_v$ à l'intersection entre la frontière de $C_N$ et le rayon de raccordement de $x$ et $x_v$. On obtient
$$x = \widehat{c}\widehat{x}_v + (1-\widehat{c})x_o$$
avec $\widehat{c} < c$, ce qui conduit à la conclusion que la solution optimale $\{x_v^*, x_o^*\}$, il estime que $x_v^* \in \mathrm{Fr}(C_N)$.

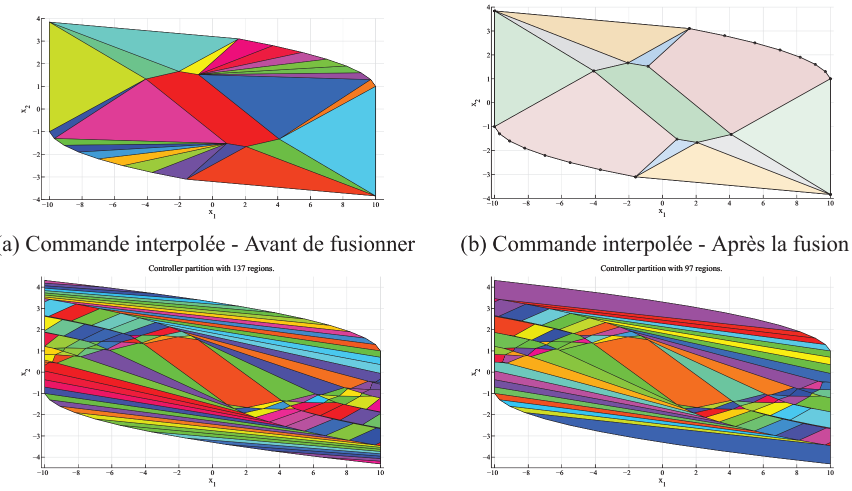
De la remarque précédente, nous concluons que pour tout $x \in C_N \setminus \Omega_{max}$ le coefficient d'interpolation $c$ atteint un minimum en (0.13) si et seulement si $x$ est écrit sous la forme d'une combinaison convexe de deux points, l'un appartenant à la frontière de $\Omega_{max}$ et l'autre étant sur la frontière de $C_N$.

Il est en outre montré que, si $x \in C_N \setminus \Omega_{max}$, la plus petite valeur de $c$ sera atteint lorsque la région $C_N \setminus \Omega$ est décomposé en polytopes avec leurs sommets situés, soit sur la frontière de $\Omega_{max}$ ou sur la frontière de $C_N$. Ces polytopes peuvent être décomposée en simplexes, formées chacune par $r$ sommets de $C_N$ et $n-r+1$ sommets de $\Omega_{max}$ ou $1 \leq r \leq n$. Nous allons également prouver que la commande d'interpolation de base donne une solution explicite avec les lois de commande affines par morceaux dans $C_N \setminus \Omega_{max}$ partitionné en simplexes (solution similaire, mais plus simple que cell de la commande prédictive explicite). A l'intérieur de l'ensemble $\Omega_{max}$, la commande interpolée s'avère être la commande optimale sans contrainte $u(k) = Kx(k)$.

## Commande interpolée basée sur la programmation quadratique

### *L'interpolation entre les régulateurs linéaires*

La commande interpolée basée sur l'interpolation (0.11), (0.12), (0.13) se réduit à l'utilisation de la programmation linéaire, qui est extrêmement simple. Toutefois, le principal problème en ce qui concerne la mise en ouvre de l'algorithme (0.11),




(a) Commande interpolée - Avant de fusionner (b) Commande interpolée - Après la fusion

(c) Commande prédictive - Avant de fusionner (d) Commande prédictive - Après la fusion

**Fig. 0.7** Partition de l'espace état de la commande interpolée et la commande prédictive.

(0.12), (0.13) est la non-unicité de la solution. Plusieurs optimas ne sont pas souhaitables, car ils pourraient conduire à une commutation rapide entre les différentes actions de commande optimale lorsque le problème LP (0.13) est résolu en ligne. Traditionnellement, la commande prédictive a été formulé en utilisant un critère quadratique [100]. Donc, pour la commande à base d'interpolation, cela vaut la peine de se tourner vers l'utilisation de la programmation quadratique.

Avant d'introduire une formulation QP, notons que l'idée d'utiliser des formulations QP pour la commande interpolée n'est pas nouvelle. Dans [11], [132], la théorie de Lyapunov est utilisée pour calculer une borne supérieure de la fonction de coût à horizon infini.

$$J = \sum_{k=0}^{\infty} \left\{ x(k)^T Q x(k) + u(k)^T R u(k) \right\} \tag{0.14}$$

où $Q \succeq 0$ et $R \succ 0$ sont les matrices d'état et d'entrée. À chaque instant, l'algorithme dans [11] utilise une décomposition en ligne de l'état actuel, avec chaque composant se trouvant dans un ensemble distinct invariant. Après quoi le dispositif de commande correspondant est appliqué à chaque composant séparément, de maniére à calculer l'action de commande. Les polytopes qu'on utilise comme ensembles candidats sont invariants. Par conséquent, le problème d'optimisation en ligne peut être formulé comme un problème QP. Cependant, les résultats de [11], [132] ne permettent pas d'imposer une priorité parmi les lois de contrôle d'interpolation.

Dans ce manuscrit, nous fournissons une contribution à cet direction de recherche en tenant compte dans l'interpolation, du fait que une des commandes aura la plus grande priorité, tandis que les autres gains joueront le rôle de degrés de liberté de manière à élargir le domaine d'attraction. Cette approche alternative peut fournir un cadre approprié pour la conception de lois de commande sous contraintes qui s'appuie sur la commande optimale sans contraintes (généralement avec un gain élevé) et par la suite on pourra règle le facteur d'interpolation pour faire face à des contraintes et des limitations (par interpolation avec les contrôleurs adéquats à faible gain).

On suppose que l'utilisation des résultats établis dans la théorie de la commande, on obtient un ensemble de correcteurs sans contraintes asymptotiquement stabilisés $u(k) = K_i x(k)$, $i = 1, 2, \ldots, r$ tel que pour les matrices d'état et d'entrée, le problème d'optimisation suivant

$$(A_j + B_j K_i)^T P_i (A_j + B_j K_i) - P_i \preceq -Q_i - K_i^T R_i K_i, \forall j = 1, 2, \ldots, s$$

est faisable par rapport à la variable $P_i \in \mathbb{R}^{n \times n}$.

Notons $\Omega_i \subseteq X$ un ensemble invariant maximal pour chaque correcteur $K_i$, et $\Omega$ dans une enveloppe convexe de $\Omega_i$. De la convexité de $X$, il s'ensuie' que $\Omega \subseteq X$. Le correcteur de gain élevé dans cette énumération jouera le rôle d'un candidat prioritaire, tandis que les autres correcteurs de gain faible seront utilisés dans le schéma d'interpolation pour élargir le domaine d'attraction.

Tout l'état $x(k) \in \Omega$ peut être décomposé comme suit

$$x(k) = \lambda_1 \widehat{x}_1 + \lambda_2 \widehat{x}_2 + \ldots + \lambda_r \widehat{x}_r \tag{0.15}$$

où $\widehat{x}_i \in \Omega_i$ pour tout $i = 1, 2, \ldots, r$ et

$$\sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \geq 0$$

Considérons la loi de commande suivante

$$u(k) = \lambda_1 K_1 \widehat{x}_1 + \lambda_2 K_2 \widehat{x}_2 + \ldots + \lambda_r K_r \widehat{x}_r \tag{0.16}$$

où $u_i(k) = K_i \widehat{x}_i$ est la loi de commande, associé à la construction de l'ensemble invariant $\Omega_i$.

À chaque instant, considère le problème d'optimisation suivant

$$\min_{x_i, \lambda_i} \left\{ \sum_{i=2}^{r} \widehat{x}_i^T P_i \widehat{x}_i + \lambda_i^2 \right\} \tag{0.17}$$

sujet aux contraintes

$$\begin{cases} \widehat{x}_i \in \Omega_i, \forall i = 1, 2, \ldots, r \\ \sum_{i=1}^{r} \lambda_i \widehat{x}_i = x \\ \sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \geq 0 \end{cases}$$

Nous insistons sur le fait que la fonction objectif est construite sur les indices $\{2,\ldots r\}$, ce qui correspond aux correcteurs de plus faible priorité.

Il sera démontré que le problème d'optimisation non linéaire (0.17) peut être converti en un problème d'optimisation quadratique. Il sera en outre montré que, la commande d'interpolation basée sur (0.15), (0.16), (0.17) garantit la faisabilité récursive et la stabilité asymptotique robuste du système en boucle fermée.

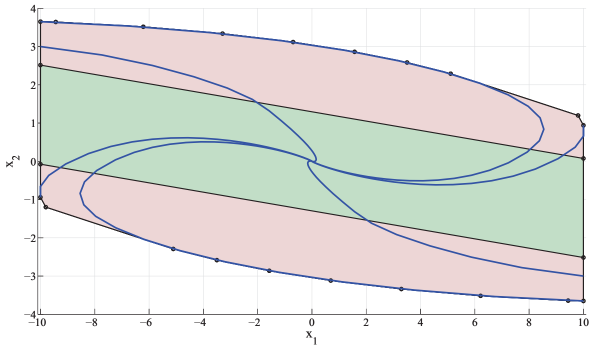
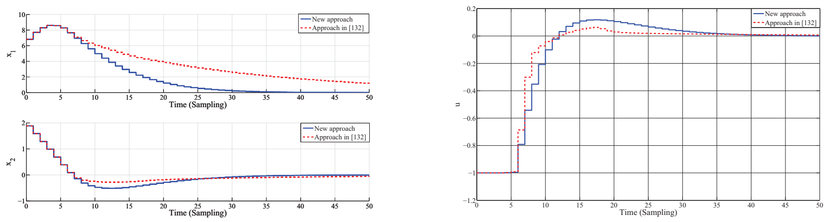



**Fig. 0.8** Ensembles invariant et des trajectoires d'état.




(a) Trajectoires d'état (b) Trajectoires d'entrée

**Fig. 0.9** Commande interpolée basée sur la programmation quadratique.

Il est évident que lorsque $x \in \Omega_1$, le problème d'optimisation (0.17) admet une solution triviale, on a

$$\begin{cases} x_i = 0, \\ \lambda_i = 0 \end{cases}$$

pour tout $i = 2,3,\ldots,r$. D'où $x_1 = x$ et $\lambda_1 = 1$ ou dans une autre perspective, pour tout $x \in \Omega_1$, la commande interpolée s'avère être la commande optimal sans contrainte.

### *Commande interpolée entre les correcteurs saturés*

Afin d'utiliser le complet potentiel des actionneurs et de satisfaire aux contraintes d'entrée sans avoir à manipuler une commande inutilement complexe, fondée sur l'optimisation, une fonction de saturation à l'entrée sera considérée. La saturation permettra de garantir que les contraintes sur l'entrée du système soient satisfaites. Dans notre conception, nous exploitons le fait que la commande linéaire saturée linéaire peut être exprimée comme une combinaison convexe d'un ensemble de lois linéaire selon Hu et al. [59]. Ainsi, les lois de commande disponibles dans l'enveloppe convexe plutôt que la loi de commande optimale, vont gérer les contraintes sur les signaux d'entrée.

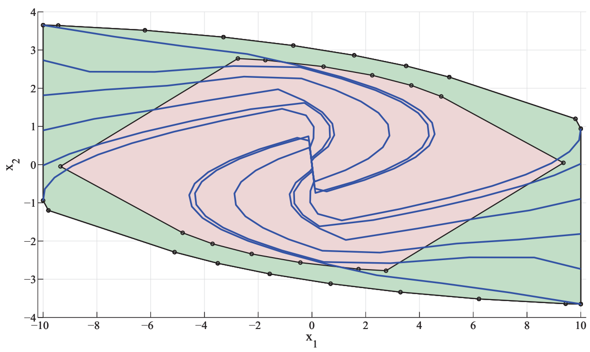
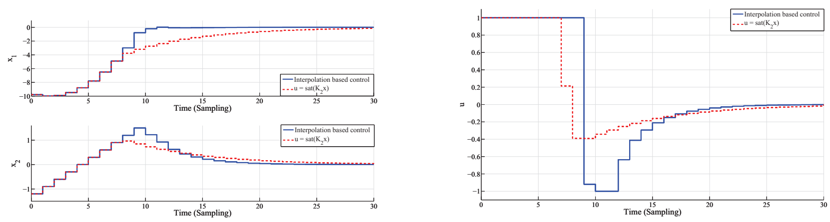



**Fig. 0.10** Ensembles invariant et trajectoires d'état.




(a) Trajectoires d'état (b) Trajectoires d'entrée

**Fig. 0.11** Interpolation entre les correcteurs saturés.

## Commande interpolée basée sur LMI

Pour les systèmes de dimension élevée, les méthodes fondées sur les ensembles polyédrales pourraient ne pouvoir s'appliquer, puisque le nombre de sommets ou de demi-espaces peuvent conduire à une complexité exponentielle. Dans ces cas, les ellipsoïdes semblent être une classe appropriée d'ensembles candidats pour l'interpolation. Dans ce manuscrit, l'enveloppe convexe d'une famille d'ellipsoïdes est utilisé pour estimer le domaine de stabilité pour un système de la commande sous contraintes. Ceci est motivé par des problèmes liés à l'estimation du domaine d'attraction de façon à l'agrandir. Afin de décrire brièvement la classe des problèmes, supposons qu'un ensemble d'ellipsoïdes invariants et un ensemble associé de lois *saturées* soient disponibles. Notre objectif est de savoir si l'enveloppe convexe de l'ensemble de ces ellipsoïdes est invariant par la commande et la façon de construire une loi de commande pour cette région .

Il est supposé que les contraintes sur l'état $X$ et les contraintes sur l'entrée $U$ sont symétriques. Il est également supposé qu'un ensemble de correcteurs $K_i \in \mathbb{R}^{m\times n}$ pour $i = 1,2,\ldots,r$ sont disponibles tels que les ensembles ellipsoïdales invariants $E(P_i)$

$$E(P_i) = \left\{x \in \mathbb{R}^n : x^T P_i^{-1} x \le 1\right\} \tag{0.18}$$

sont non-vide pour $i = 1,2,\ldots,r$. Rappelons que pour tout $x(k) \in E(P_i)$, il s'ensuit que $sat(K_i x) \in U$ et $x(k+1) = Ax(k) + Bsat(K_i x(k)) \in X$. On note par la suite $\Omega_E \subset \mathbb{R}^n$ comme une enveloppe convexe de $E(P_i)$ pour tout $i$. Il s'ensuit que $\Omega_E \subseteq X$, depuis $E(P_i) \subseteq X$.

Tout état $x(k) \in \Omega_E$ peut être décomposé comme suit

$$x(k) = \sum_{i=1}^{r} \lambda_i \widehat{x}_i(k) \tag{0.19}$$

avec $\widehat{x}_i(k) \in E(P_i)$ et $\lambda_i$ sont les coefficients d'interpolation, qui satisfont

$$\sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \ge 0$$

Considérons la loi de commande suivante

$$u(k) = \sum_{i=1}^{r} \lambda_i sat(K_i \widehat{x}_i(k)) \tag{0.20}$$

où $sat(K_i \widehat{x}_i(k))$ est la loi de commande saturée, ce qui est faisable dans $E(P_i)$.

Le premier correcteur de gain élevé sera utilisé afin de garantir la performance et sera considéré comme prioritaire, tandis que le reste des correcteurs disponibles (à faible gain) seront utilisés pour élargir le domaine d'attraction. Pour l'état courant donné $x$, considérer la fonction objective suivante

$$\min_{\widehat{x}_i,\lambda_i} \sum_{i=2}^{r} \lambda_i \tag{0.21}$$

sujet à

$$\begin{cases} \widehat{x}_i^T P_i^{-1} \widehat{x}_i \leq 1, \forall i = 1,2,\ldots,r \\ \sum\limits_{i=1}^{r} \lambda_i \widehat{x}_i = x \\ \sum\limits_{i=1}^{r} \lambda_i = 1 \\ \lambda_i \geq 0, \forall i = 1,2,\ldots,r \end{cases}$$

Il sera montré que le problème d'optimisation non linéaire (0.21) peut être reformulée comme un problème d'optimisation LMI. Il sera en outre montré que la commande d'interpolation basée sur l'utilisation d'une solution du problème d'optimisation (0.21) garantit la faisabilité et la stabilité récursive robuste et asymptotique du système en boucle fermée.

Il est clair que pour tout $x \in E(P_1)$, le problème d'optimisation (0.21) admet une solution triviale, c'est

$$\begin{cases} \widehat{x}_i = 0, \\ \lambda_i = 0 \end{cases} \quad \forall i = 2,3,\ldots,r$$

pour laquelle $x_1 = x$ et $\lambda_1 = 1$. Ou en d'autres termes, la commande d'interpolation s'avère être la commande optimal e de haut gain élevé $u(k) = sat(K_1 x)$.

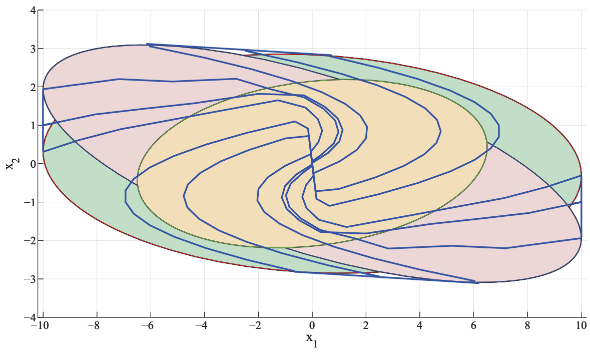
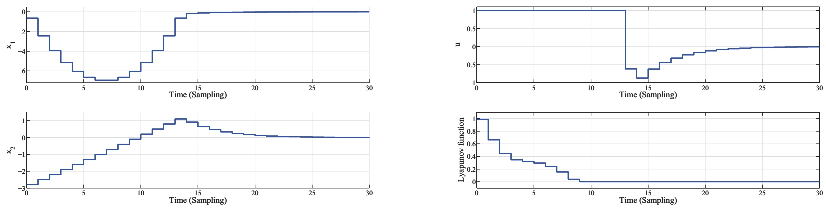



**Fig. 0.12** Ensembles invariant et les trajectoires de l'état.

## Commande par retour de sortie

Jusqu'à présent, les problèmes d'asservissement dans l'espace d'état ont été pris en compte. Cependant, dans la pratique, l'information directe ou la mesure de l'état




(a) Trajectoires de l'état (b) trajectoires de l'entreé

**Fig. 0.13** Commande interpolée basée sur LMI.

complet des systèmes dynamiques peuvent ne pas être disponibles. Dans ce cas, un observateur pourrait éventuellement être utilisée afin d'estimer l'état. Un inconvénient majeur est l'erreur de l'observateur que l'on doit inclure dans l'incertitude. En outre, lorsque les contraintes se manifestent, la non-linéarité domine la structure du système de commande et on ne peut s'attendre que le principe de séparation soit toujours valide. En outre, il n'existe aucune garantie que les trajectoires en boucle fermée satisfassent les contraintes.

Nous reviendrons sur le problème de la reconstruction de l'état grâce à la mesure et le stockage des mesures précédentes appropriées. Même si ce modèle pourrait être *non-minimal* du point de vue de la dimension, il est directement mesurable et fournira un modèle approprié pour la conception de la commande avec des garanties de satisfaction de contraintes. Enfin, il sera montré comment les principes de la commande interpolée peut conduire à une commande par retour de sortie.

## *Formulation du problème*

Considérons le problème de la régulation à l'origine en temps discret pour un système linéaire variant dans le temps ou incertain, décrite par la relation d'entrée-sortie

$$\begin{aligned} y(k+1)+E_1y(k)+E_2y(k-1)+\ldots+E_sy(k-s+1) \\ =N_1u(k)+N_2u(k-1)+\ldots+N_ru(k-r+1)+w(k) \end{aligned} \tag{0.22}$$

où $y(k)\in\mathbb{R}^p$, $u(k)\in\mathbb{R}^m$ et $w(k)\in\mathbb{R}^p$ sont respectivement la sortie, l'entrée et le vecteur de perturbation. Les matrices $E_i$ for $i=1,\ldots,s$ et $N_i$ pour $i=1,\ldots,r$ doivent avoir des dimensions appropriées.

Pour plus de simplicité, il est supposé que $s=r$. Les matrices $E_i$ et $N_i$ pour $i=1,2,\ldots,s$ satisfont

$$\Gamma=\begin{bmatrix} E_1 & E_2 & \ldots & E_s \\ N_1 & N_2 & \ldots & N_s \end{bmatrix}=\sum_{i=1}^{q}\alpha_i(k)\Gamma_i \tag{0.23}$$

où $\alpha_i(k) \geq 0$ et $\sum_{i=1}^{q} \alpha_i(k) = 1$ et

$$\Gamma_i = \begin{bmatrix} E_1^i & E_2^i & \dots & E_s^i \\ N_1^i & N_2^i & \dots & N_s^i \end{bmatrix}$$

sont les réalisations extrêmes d'un modèle polytopique.

Le système est soumis à des contraintes sur la sortie, la commande

$$\begin{cases} y(k) \in Y, \quad Y = \{y \in \mathbb{R}^p : F_y y \leq g_y\} \\ u(k) \in U, \quad U = \{u \in \mathbb{R}^m : F_u u \leq g_u\} \end{cases} \tag{0.24}$$

où $Y$ et $U$ sont des ensembles convexes et compactes. Il est supposé que la perturbation $w(k)$ est inconnue, additive et se trouvent dans le polytope $W$, c'est-à-dire $w(k) \in W$, où $W = \{w \in \mathbb{R}^p : F_w w \leq g_w\}$ est un C-ensemble.

## *Cas nominal*

Nous considérons le cas où les matrices $E_j$ et $N_j$ pour $j = 1, 2, \dots, s$ sont connues et fixes. Le cas où $E_j$ et $N_j$ pour $j = 1, 2, \dots, s$ sont inconnus ou variables dans le temps sera traitée dans la section suivante.

Une représentation d'état sera construite selon les principes de [153]. Toutes les étapes de la construction sont détaillés tels que la présentation des résultats soit autonomes. L'état du système est choisi comme un vecteur de dimension $p \times s$ avec les composants suivants

$$x(k) = \left[ x_1(k)^T \; x_2(k)^T \; \dots \; x_s(k)^T \right]^T \tag{0.25}$$

où

$$\begin{cases} x_1(k) = y(k) \\ x_2(k) = -E_s x_1(k-1) + N_s u(k-1) \\ x_3(k) = -E_{s-1} x_1(k-1) + x_2(k-1) + N_{s-1} u(k-1) \\ x_4(k) = -E_{s-2} x_1(k-1) + x_3(k-1) + N_{s-2} u(k-1) \\ \vdots \\ x_s(k) = -E_2 x_1(k-1) + x_{s-1}(k-1) + N_2 u(k-1) \end{cases} \tag{0.26}$$

Il sera démontré que le modèle d'état est alors définie sous une forme compacte d'équation à différence linéaire comme suit

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + Dw(k) \\ y(k) = Cx(k) \end{cases} \tag{0.27}$$

où

$$A = \begin{bmatrix} -E_1 & 0\;0 \dots 0\; I \\ -E_s & 0\;0 \dots 0\;0 \\ -E_{s-1} & I\;0 \dots 0\;0 \\ -E_{s-2} & 0\;I \dots 0\;0 \\ \vdots & \vdots\;\vdots\;\ddots\;\vdots\;\vdots \\ -E_2 & 0\;0 \dots I\;0 \end{bmatrix}, \quad B = \begin{bmatrix} N_1 \\ N_s \\ N_{s-1} \\ N_{s-2} \\ \vdots \\ N_2 \end{bmatrix}, \quad D = \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$
$$C = \begin{bmatrix} I\;0\;0\;0 \dots 0 \end{bmatrix}$$

On note

$$z(k) = [y(k)^T \dots y(k-s+1)^T \; u(k-1)^T \dots u(k-s+1)^T]^T \tag{0.28}$$

Il sera en outre montré que le vecteur d'état $x(k)$ est liée au vecteur $z(k)$ comme suit

$$x(k) = Tz(k) \tag{0.29}$$

où

$$T = [T_1 \quad T_2]$$
$$T_1 = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ 0 & -E_s & 0 & \dots & 0 \\ 0 & -E_{s-1} & -E_s & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -E_2 & -E_3 & \dots & -E_s \end{bmatrix}, \quad T_2 = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ N_s & 0 & 0 & \dots & 0 \\ N_{s-1} & N_s & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ N_2 & N_3 & N_4 & \dots & N_s \end{bmatrix}$$

A tout instant $k$, le vecteur de variables d'état est disponible uniquement si la mesure et le stockage des mesures précédentes est assuré.

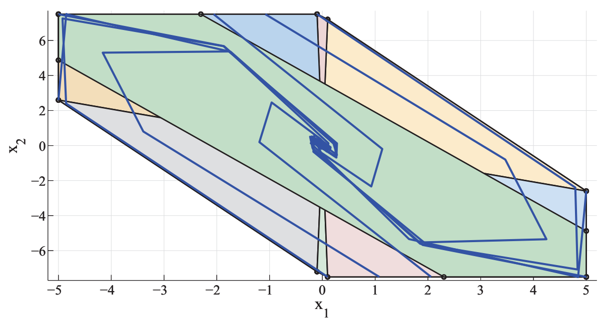



**Fig. 0.14** Ensembles invariant et trajectoires de l'état.

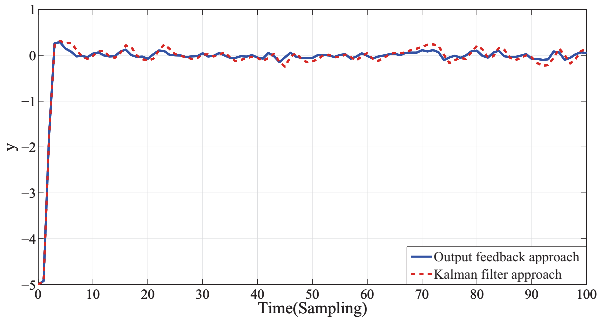



**Fig. 0.15** Trajectoires de sortie. Une simulation de la commande interpolée par retour de sortie comparée à l'utilisation du filtre de Kalman.

## *Cas robuste*

Une faiblesse de l'approche ci-dessus est que la mesure d'état est disponible si et seulement si les paramètres du système sont connus. Pour le système incertain ou variant dans le temps, ce n'est pas le cas. Dans cette section, nous proposons une autre méthode pour construire les variables d'état, qui n'utilisent pas les informations sur les paramètres du système. En utilisant l'entrée mesurée, la sortie et leurs dernières valeurs mesurées, l'état du système est choisi en tant que

$$x(k) = [y(k)^T \ \dots \ y(k-s+1)^T \ u(k-1)^T \ \dots \ u(k-s+1)^T]^T \qquad (0.30)$$

Le modèle espace d'état est alors définie comme suit

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) + Dw(k) \\ y(k) &= Cx(k) \end{cases} \qquad (0.31)$$

où

$$A = \begin{bmatrix} -E_1 & -E_2 & \dots & -E_s & N_2 & \dots & N_{s-1} & N_s \\ I & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & O & I & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & O & 0 & \dots & I & 0 \end{bmatrix}, B = \begin{bmatrix} N_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix}, D = \begin{bmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} I \ 0 \ 0 \dots 0 \ 0 \ 0 \dots 0 \end{bmatrix}$$

Bien que la représentation obtenue soit non-minimale, elle a le mérite de transformer le problème de la commande par retour de sortie pour des systèmes incertains en un problème retour d’état, où les matrices $A$ et $B$ sont dans le polytope sans aucune incertitude supplémentaire et toute commande de retour d’état conçue pour cette représentation sous la forme $u = Kx$ peut être traduit en un correcteur par retour de sortie dynamique.

# Contents

# Notation

The conventions and the notations used in the thesis are classical for the control literature. A short description is provided in the following.

**Sets**

| | |
|---|---|
| $\mathbb{R}$ | Set of real number |
| $\mathbb{R}_+$ | Set of nonnegative real number |
| $\mathbb{R}^n$ | Set of real vectors with $n$ elements |
| $\mathbb{R}^{n\times m}$ | Set of real matrices with $n$ rows and $m$ columns |

**Algebraic Operators**

| | |
|---|---|
| $A^T$ | Transpose of matrix $A$ |
| $A^{-1}$ | Inverse of matrix $A$ |
| $A \succ (\succeq) 0$ | Positive (semi)definite matrix |
| $A \prec (\preceq) 0$ | Negative (semi)definite matrix |

**Set Operators**

| | |
|---|---|
| $P_1 \cap P_2$ | Set intersection |
| $P_1 \oplus P_2$ | Minkowski sum |
| $P_1 \ominus P_2$ | Pontryagin difference |
| $P_1 \subseteq P_2$ | $P_1$ is a subset of $P_2$ |
| $P_1 \subset P_2$ | $P_1$ is a strict subset of $P_2$ |
| $P_1 \supseteq P_2$ | $P_1$ is a superset of $P_2$ |
| $P_1 \supset P_2$ | $P_1$ is a strict superset of $P_2$ |
| $\mathrm{Fr}(P)$ | The frontier of $P$ |
| $\mathrm{Int}(P)$ | The interior of $P$ |
| $Proj_x(P)$ | The orthogonal projection of the set $P$ onto the $x$ space |

**Others**

**I** Identity matrix
**1** Matrix of ones of appropriate dimension
**0** Matrix of zeros of appropriate dimension

**Acronyms**

| | |
|---|---|
| LMI | Linear Matrix Inequality |
| LP | Linear Programming |
| QP | Quadratic Programming |
| LQR | Linear Quadratic Regulator |
| LTI | Linear Time Invariant |
| LPV | Linear Parameter Varying |
| PWA | PieceWise Affine |

# Chapter 1
# Introduction

## 1.1 Constrained uncertain systems

Constraints are encountered in practically all real-world control problems. The presence of constraints leads to high complexity control problems, not only in control theory, but also in practical applications. From the conceptual point of view, constraints can have different nature. Basically, there are two types of constraints imposed by physical limitation and/or performance desiderata.

*Physical constraints* are due to the physical limitations of the mechanical, electrical, biological, etc controlled system. The main concern here is the stability in the presence of input and output or state constraints. The input and output variables must remain inside the constraints to avoid over-exploitation or damage. In addition, the constraint violation may lead to degraded performance, oscillations or even instability.

*Performance constraints* are introduced by the designer for guaranteeing performance requirements, for example transient time, transient overshoot, etc, fault tolerance, equipment longevity and environmental problems.

The constrained control problem can become even more challenging in the presence of model uncertainties, which are unavoidable in practice [142], [2]. Model uncertainties are appear e.g. when a linear model is obtained as an approximation of a nonlinear system around the operating point. Even if the underlying process is quite accurately represented by a linear model[1], the parameters of the model could be time-varying or could change due to a change in the operating points. In these cases, the cause and structure of the model uncertainties are rather well known. Nevertheless, even when the real process is linear, there is always some uncertainty associated, for example, with physical parameters, which are never known exactly. Moreover, the real processes are usually affected by disturbances and it is required to consider them in control design.

It is generally accepted that a key reason of using feedback is to diminish the effects of uncertainty, which may appear in different forms as parametric uncertainties

[1] Which is actually asking a lot.

or as additive disturbances or as other inadequacies in the models used to design the feedback law. Model uncertainty and robustness have been a central theme in the development of the field of automatic control [9].

A straightforward way to stabilize a constrained system is to perform the control design disregarding the constraints. Then an adaptation of the control law is considered with respect to input saturation, such an approach is called *anti-windup* [79], [151], [149], [157].

Over the last decades, the research on constrained control topics has developed to the degree that constraints can be taken into account during the synthesis phase. By its principle, model predictive control (MPC) approach shows its importance on dealing with constraints [32], [100], [28], [129], [47], [96], [48]. In the MPC approach, a sequence of predicted optimal control values over a finite prediction horizon is computed for optimizing the performance of the controlled system, expressed in terms of a cost function [3]. MPC approach uses an internal mathematical model which, given the current measurements, predicts the future behavior of the real system with respect to changes in the control inputs. Once the sequence of optimal control inputs has been calculated, only the first element of this sequence is actually applied to the system and the entire optimization is repeated at the next time instant with the new state measurement [5], [100], [96].

In classical MPC, the control action at each time instant is obtained by solving an on-line open-loop finite optimal control problem [126], [33]. With a linear model, polyhedral constraints, and a quadratic cost, the resulting optimization problem is a quadratic program. Solving the quadratic program can be computationally costly, specially when the prediction horizon is large, and this has traditionally limited MPC to applications with relatively low complexity/sampling interval ratio [4].

In the last decade, attempts have been made to use predictive control in fast processes. In [122], [121], [20], [153] it was shown that the constrained linear MPC is equivalent to a multi-parametric optimization problem, where the state plays the role of a vector of parameters. The solution is a piecewise affine function of the state over a polyhedral partition of the state space, and the computational effort of the MPC is moved off-line. However, explicit MPC implementation approaches also have disadvantages. Obtaining the explicit optimal MPC solution requires to solve an off-line parametric optimization problem, which is generally an NP-hard problem. Although the problem is tractable and practically solvable for several interesting control applications, the off-line computational effort grows *exponentially* fast as the problem size increases [84], [85], [83], [64], [65]. This is the case for long prediction horizon, large number of constraints and high dimensional systems.

In [156], the authors show that the on-line computation is preferable for high dimensional systems where significant reduction of the computational complexity can be achieved by exploiting the particular structure of the optimization problem as well as by early stopping and warm-starting from a solution obtained at the previous time-step. The same reference mentions that for models of more than five dimensions the explicit solution might be impractical. It worth mentioning that approximate explicit solutions have been investigated to go beyond this ad-hoc limitation [19], [62], [140].

Note that as its name says, most traditional implicit and explicit MPC approaches are based on mathematical models which invariably present a mismatch with respect to the physical systems. The robust MPC is meant to address both model uncertainty and disturbances. However, the robust MPC presents great conservativeness and/or on-line computational burden [78], [123], [87].

The use of interpolation in constrained control in order to avoid very complex control design procedures is well known in the literature. There is a long line of developments on these topics generally closely related to MPC, see for example [11], [132], [133], [131], where interpolation between input sequences, state trajectories, different feedback gains and/or associated invariant sets can be found.

The vertex control can be considered also as an interpolation control approach based on the explicit control values, assumed to be available for the extreme points of a certain region in the state space [54], [22]. A weakness of vertex control is that the full control range is exploited only on the border of the feasible positive invariant set in the state space, and hence the time to regulate the plant to the origin is much longer than e.g. by time-optimal control. A way to overcome this shortcoming is to switch to another, more aggressive, local controller, e.g. a state feedback controller $u_o = Kx$, when the state reaches the maximal feasible set of the local controller. The disadvantage of such a switching-based solution is that the control action becomes non-smooth [103].

For LTI systems the vertex control Lyapunov level curves are polyhedra parallel with the border of the vertex controller feasible set, and as such we will, without loss of generality, base the new design method on the existence of a polyhedral contractive set for a local control law. This set will be related to the description of the maximal controlled invariant set. Then we point to the existence of a smooth convex interpolation between the vertex control action $u_v$ and the local control action $u_o$ for the current state $x$, in the form $u(x) = c(x)u_v(x) + (1 - c(x))u_o(x)$ with $0 \leq c(x) \leq 1$, whereby $c(x)$ is minimized in order to be as close as possible to the local optimal controller. It is shown that with this objective function, there exists a Lyapunov function for the system controlled by the interpolated controller $u$, and hence stability is proven.

It is shown that from a computational point of view the minimization of the interpolating coefficient $c$ can be done by linear programming. It is further shown that that the minimization can be done off-line, yielding a polyhedral partition of the feasible region, with an affine control law for each polyhedron, while guaranteeing the global continuity of the state feedback. Thus, our controller can be compared from the structural point of view with explicit MPC where the feasible set in the state space is also partitioned into polyhedra, each of which with its own affine state feedback control law.

The interpolation based on an LP (linear programming) problem between the global vertex controller and the local more aggressive controller is the first aim of the thesis. Then as in the traditional MPC approach, which is formulated using a quadratic criterion [100], we will show how an interpolation based control problem for linear systems can be set up as a quadratic program. All the interpolation schemes via LP or QP computations are based on the use of polyhedral sets. For high dimensional systems, the polyhedral based control methods might be impractical, since the number of vertices or half-spaces may lead to an exponential complexity. In these cases, the ellipsoids seem to be the suitable class of sets in the interpolation. It will be shown that the convex hull of a set of invariant ellipsoids is controlled invariant. A continuous feedback control law is constructed based on the solution of an LMI problem at each time instant. For all interpolation optimization based schemes, a proof of recursive feasibility and robust asymptotic stability will be provided.

## 1.2 Organization of the thesis

The thesis (except the present chapter) is partitioned into four parts and appendices

**Part** I contains two chapters introducing the theoretical foundations for the rest of the thesis. In Chapter 2, basic set theory elements are discussed with the accent on the (controlled) invariant set. The advantages as well as disadvantages of different families of sets and their use in control will be considered, which is instrumental for the presentation of the main results of the thesis. Chapter 3 reviews the main approaches to optimal and constrained control with emphasis on vertex control, which is one of the main ingredients of an interpolation based control scenario.

**Part** II consists of three chapters and provides a novel and computationally attractive solution to a constrained control problem. This part presents several original contributions on constrained control algorithms for discrete-time linear systems. Chapter 4 is concerned only with the nominal state-input constrained systems where there are no disturbances and no model mismatch. In this chapter a series of generic interpolation based control schemes via linear programming, quadratic programming or linear matrix inequality are introduced. Further, in Chapter 5, we extend the interpolation technique for the discrete time linear uncertain or time-varying systems subject to bounded disturbances. To complete the presentation, in Chapter 6, the output feedback case is considered. This last feature is very important, since state feedback is rarely used in (constrained control) practice. For all algorithms proposed in this part, the proofs of recursive feasibility and asymptotic stability are given.

**Part** III contains two chapters applying the theoretical results discussed in Part II to one practical application proposed in the literature and one benchmark. In Chapter 7 the interpolation based control via linear programming is used for stabilizing a ball and plate laboratory system. Then in Chapter 8, the explicit interpolation based control approach is implemented on a non-isothermal continuous stirred tank reactor.

**Part** IV consists two sections which completes the thesis with conclusions and future directions.

## 1.3 List of Publications related to the PhD

We provide here the complete list of publications submitted/accepted to various conferences and journals

**Publish journal papers**

- **Hoai-Nam Nguyen**, Sorin Olaru,"Hybrid modeling and constrained control of juggling system", *International Journal of Systems Science*, 2011. [114]
- **Hoai-Nam Nguyen**, Sorin Olaru, Morten Hovd, A patchy approximation of explicit model predictive control, *International Journal of Control*, 2012. [113]

**Submitted journal papers**

- **Hoai-Nam Nguyen**, Per-Olof Gutman, Sorin Olaru, Morten Hovd, Improved vertex control for discrete-time linear time-invariant systems with state and control constraints, submitted for publication (second review round), *Automatica*.

**Book chapters**

- **Hoai-Nam Nguyen**, Sorin Olaru, Per-Olof Gutman, Morten Hovd, Robust output feedback interpolation based control for constrained linear systems, 2012, *Informatics in Control, Automation and Robotics Revised and Selected Papers from the International Conference on Informatics, published by Springer-Verlag.* [109]

**Publish conference papers**

- **Hoai-Nam Nguyen**, Sorin Olaru, Per-Olof Gutman, Morten Hovd, "Interpolation based control for constrained linear time-varying or uncertain systems in the presence of disturbances", *in 4th IFAC Nonlinear Model Predictive Control Conference, August 23 - 27, 2012, NH Conference Center Leeuwenhorst, Noordwijkerhout, NL.* [112]
- **Hoai-Nam Nguyen**, Sorin Olaru, Per-Olof Gutman, Morten Hovd, "Improved vertex control for a ball and plate system", *in 7th IFAC Symposium on Robust Control Design, June 20-22, 2012, Aalborg, Denmark.* [111]
- **Hoai-Nam Nguyen**, Sorin Olaru, Per-Olof Gutman, Morten Hovd, "An improved interpolation based control method in the presence of actuator saturation", *At Itzhack Y. Bar-Itzhack Memorial Symposium on Estimation, Navigation, and Spacecraft Control, October 1417, 2012, Dan Panorama Hotel, Haifa, Israel*

- **Hoai-Nam Nguyen**, Sorin Olaru, Per-Olof Gutman, Morten Hovd, "Constrained interpolation-based control for polytopic uncertain systems", *In 50th IEEE Conference on Decision and Control and European Control Conference, December 12-15, 2011, US.* [110]
- **Hoai-Nam Nguyen**, Per-Olof Gutman, Sorin Olaru, Morten Hovd, "An interpolation approach for robust constrained output feedback", *In 50th IEEE Conference on Decision and Control and European Control Conference, December 12-15, 2011, US.* [107]
- **Hoai-Nam Nguyen**, Sorin Olaru, Florin Stoican, "On maximal robustly positively invariant sets", *In ICINCO 2011: 8th International Conference on Informatics in Control, Automation and Robotics.* [115]
- **Hoai-Nam Nguyen**, Per-Olof Gutman, Sorin Olaru, Morten Hovd, "Explicit constraint control based on interpolation techniques for time varying and uncertain linear discrete time systems", *In 18th IFAC World Congress, Italy, 2011.* [106]
- **Hoai-Nam Nguyen**, Per-Olof Gutman, Sorin Olaru, Morten Hovd, Federic Colledani, "Improved vertex control for uncertain linear discrete time systems with control and state constraints", *In ACC2011, San Francisco, California, USA, June 29 - July 1, 2011.* [108]
- **Hoai-Nam Nguyen**, Sorin Olaru, Morten Hovd, "Patchy approximate explicit model predictive control", *In Proceedings of International Conference on Control, Automation and Systems ICCAS (International Conference on Control, Automation and Systems ICCAS), Gyeonggi-do (Korea), 2010.* [103]
- **Hoai-Nam Nguyen**, Sorin Olaru, "Hybrid modeling and optimal control of juggling systems", *In Proceedings of International Conference on Control, Automation and Systems, Sinaia (Romania), 2010.* [114]

# Part I
# Background

# Chapter 2
# Set Theoretic Methods in Control

The first aim of this chapter is to briefly review some of the set families used in control and to comment on the strengths and weaknesses of each of them. The tools of choice throughout the manuscript will be ellipsoidal and polyhedral sets due to their combination of numerical applicability and flexibility in the representation of generic convex sets. After the geometrical nomenclature, the concept of robustly invariant and robust controlled invariant sets are introduced. Some algorithms are proposed for computing such sets. The chapter ends with an original contribution on estimating the domain of attraction for time-varying and uncertain discrete-time systems with a saturated input.

## 2.1 Set terminology

For completeness, some standard definitions of set terminology will first be introduced in this section. For a detailed reference, the reader is referred to the book [77].

**Definition 2.1. (Closed set)** A set $S$ is *closed* if it contains its own boundary. In other words, any point outside $S$ has a neighborhood disjoint from $S$.

**Definition 2.2. (Closure of a set)** The *closure* of a set $S$ is the intersection of all *closed sets* containing $S$.

**Definition 2.3. (Bounded set)** A set $S \subset \mathbb{R}^n$ is *bounded* if it is contained in some ball $B_R = \{x \in \mathbb{R}^n : \|x\|_2 \leq \varepsilon\}$ of finite radius $\varepsilon > 0$.

**Definition 2.4. (Compact set)** A set $S \subset \mathbb{R}^n$ is *compact* if it is closed and bounded.

**Definition 2.5. (Support function)** The support function of a set $S \subset \mathbb{R}^n$, evaluated at $z \in \mathbb{R}^n$ is defined as

$$\phi_S(z) = \sup_{x \in S} z^T x$$

## 2.2 Convex sets

### *2.2.1 Basic definitions*

The fact that convexity is a more important property than linearity has been recognized in several domains, the optimization theory being maybe the best example [128], [30]. We provide in this section a series of definitions which will be useful in the sequel.

**Definition 2.6. (Convex set)** A set $S \subset \mathbb{R}^n$ is *convex* if for all $x_1 \in S$ and $x_2 \in S$, it holds that

$$\alpha x_1 + (1-\alpha)x_2 \in S, \quad \forall \alpha \in [0,\ 1]$$

The point

$$x = \alpha x_1 + (1-\alpha)x_2$$

where $0 \leq \alpha \leq 1$ is called a *convex combination* of the pair $(x_1, x_2)$. The set of all such points is the segment connecting $x_1$ and $x_2$. In other words a set $S$ is said to be convex if the line segment between any two points in $S$ lies in $S$.

The concept of convex set is closely related to the definition of a convex functions[1].

**Definition 2.7. (Convex function)** A function $f : S \rightarrow \mathbb{R}$ with $S \subseteq \mathbb{R}^n$ is *convex* if and only if the set $S$ is convex and

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2)$$

for all $x_1 \in S$, $x_2 \in S$ and for all $\alpha \in [0,\ 1]$.

**Definition 2.8. (C-set)** A set $S \subset \mathbb{R}^n$ is a $C-$set if it is a convex and compact set, containing the origin in its interior.

**Definition 2.9. (Convex hull)** The *convex hull* of a set $S \subset \mathbb{R}^n$ is the smallest convex set containing $S$.

It is well known [158] that for any finite set $S = \{s_1, s_2, \ldots, s_r\}$ with $r \in \mathbb{N}$, the convex hull of the set $S$ is given as

$$\text{Convex Hull}(S) = \{s \in \mathbb{R}^n : s = \sum_{i=1}^{r} \alpha_i s_i : \forall s_i \in S\}$$

where $\sum\limits_{i=1}^{r} \alpha_i = 1$ and $\alpha_i \geq 0$.

---

[1] One can understand the link straightforwardly by the fact that the epigraph of a convex function is a convex set.

## 2.2.2 Ellipsoidal set

Ellipsoidal sets or ellipsoids are one of the famous classes of convex sets. Ellipsoids represent a large category used in the study of dynamical systems and the control fields due to their simple numerical representation [29], [80]. Next we provide a formal definition for ellipsoidal sets and a few properties.

**Definition 2.10. (Ellipsoidal set)** An ellipsoidal set $E(P,x_0) \subseteq \mathbb{R}^n$ with center $x_0$ and shape matrix $P$ is a set of the form

$$E(P,x_0) = \{x \in \mathbb{R}^n : (x-x_0)^T P^{-1}(x-x_0) \leq 1\} \tag{2.1}$$

where $P \in \mathbb{R}^{n\times n}$ is a positive definite matrix.

If the ellipsoid is centered in the origin then it is possible to write

$$E(P) = \{x \in \mathbb{R}^n : x^T P^{-1} x \leq 1\} \tag{2.2}$$

Define $Q = P^{\frac{1}{2}}$ as the Cholesky factor of matrix $P$, which satisfies $Q^T Q = QQ^T = P$. With the matrix $Q$, it is possible to show an alternative dual representation for an ellipsoidal set

$$D(Q,x_0) = \{x \in \mathbb{R}^n : x = x_0 + Qz\}$$

where $z \in \mathbb{R}^n$ such that $z^T z \leq 1$.

Ellipsoidal sets are probably the most commonly used in the control field since they are associated with powerful tools such as the Lyapunov equation[2] or Linear Matrix Inequalities (LMI) [137], [29]. When using ellipsoidal sets, almost all the optimization problems present in the classical control methods can be reduced to the optimization of a linear function under LMI constraints. This optimization problem is convex and is by now a powerful design tool in many control applications.

A linear matrix inequality is a condition of the type [137], [29]

$$F(x) \succ 0$$

where $x \in \mathbb{R}^n$ is a vector variable and the matrix $F(x)$ is affine in $x$, that is

$$F(x) = F_0 + \sum_{i=1}^{n} F_i x_i$$

with symmetric matrices $F_i \in \mathbb{R}^{m\times m}$.

LMIs can either be understood as feasibility conditions or constraints for optimization problems. Optimization of a linear function over LMI constraints is called semidefinite programming, which is considered as an extension of linear programming. Nowadays, a major benefit in using LMIs is that for solving an LMI problem,

---

[2] A quadratic Lyapunov function can be associated to stable linear dynamics. Consequently, the ellipsoids are natural representations of the quadratic Lyapunov function level sets.

several polynomial time algorithms were developed and implemented in free available software packages, such as LMI Lab [43], YALMIP [94], CVX [49], SEDUMI [147] etc.

The Schur complement formula is a very useful tool for manipulating matrix inequalities. The Schur complement states that the nonlinear conditions of the special forms

$$\begin{cases} P(x) \succ 0 \\ P(x) - Q(x)^T R(x)^{-1} Q(x) \succ 0 \end{cases} \tag{2.3}$$

or

$$\begin{cases} R(x) \succ 0 \\ R(x) - Q(x) P(x)^{-1} Q(x)^T \succ 0 \end{cases} \tag{2.4}$$

can be equivalently written in the LMI form

$$\begin{bmatrix} P(x) & Q(x)^T \\ Q(x) & R(x) \end{bmatrix} \succ 0 \tag{2.5}$$

The Schur complement allows one to convert certain nonlinear matrix inequalities into a higher dimensional LMI. For example, it is well known [80] that the support function of the ellipsoid $E(P,x_0)$, evaluated at the vector $f$ is

$$\phi_{E(P,x_0)}(z) = f^T x_0 + \sqrt{f^T P f} \tag{2.6}$$

then it is obvious that the ellipsoid $E(P)$ in (2.2) is a subset of the polyhedral set[3]

$$P(f,1) = \{x \in \mathbb{R}^n : |f^T x| \leq 1\}$$

with $f \in \mathbb{R}^n$ if and only if

$$f^T P f \leq 1$$

or by using the Schur complement this condition can be rewritten as [29], [57]

$$\begin{bmatrix} 1 & f^T P \\ Pf & P \end{bmatrix} \succeq 0 \tag{2.7}$$

Obviously an ellipsoidal set $E(P,x_0) \subset \mathbb{R}^n$ is uniquely defined by its matrix $P$ and by its center $x_0$. Since matrix $P$ is symmetric, the complexity of the representation (2.1) is

$$\frac{n(n+1)}{2} + n = \frac{n(n+3)}{2}$$

The main drawback of ellipsoids is however that having a fixed and symmetrical structure they may be too conservative and this conservativeness is increased by the related operations. It is well known [80][4] that

- The convex hull of of a set of ellipsoids, in general, is not an ellipsoid.

---

[3] A rigorous definition of polyhedral sets will be given in the section.

[4] The reader is referred to [80] for the definitions of operations with ellipsoids.

- The sum of two ellipsoids is not, in general, an ellipsoid.
- The difference of two ellipsoids is not, in general, an ellipsoid.
- The intersection of two ellipsoids is not, in general, an ellipsoid.

### *2.2.3 Polyhedral set*

Polyhedral sets provide a useful geometrical representation for the linear constraints that appear in diverse fields such as control and optimization. In a convex setting, they provide a good compromise between complexity and flexibility. Due to their linear and convex nature, the basic set operations are relatively easy to implement [82], [154]. Principally, this is related to their dual (half-spaces/vertices) representation [101], [31] which allows choosing which formulation is best suited for a particular problem.

This section is started by recalling some theoretical concepts.

**Definition 2.11. (Hyperplane)** A hyperplane $H \subset \mathbb{R}^n$ is a set of the form

$$H = \{x \in \mathbb{R}^n : f^T x = g\} \tag{2.8}$$

where $f \in \mathbb{R}^n$ is a column vector and $g \in \mathbb{R}$ is a scalar.

**Definition 2.12. (Half-space)** A closed half-space $H \subset \mathbb{R}^n$ is a set of the form

$$H = \{x \in \mathbb{R}^n : f^T x \leq g\} \tag{2.9}$$

where $f \in \mathbb{R}^n$ is a column vector and $g \in \mathbb{R}$ is a scalar.

**Definition 2.13. (Polyhedral set)** A convex polyhedral set $P(F,g)$ is a set of the form

$$P(F,g) = \{x \in \mathbb{R}^n : F_i x \leq g_i, \quad i = 1,2,\ldots,n_1\} \tag{2.10}$$

where $F_i \in \mathbb{R}^{1\times n}$ denotes the $i-$th row of the matrix $F \in \mathbb{R}^{n_1 \times n}$ and $g_i$ is the $i-$th component of the column vector $g \in \mathbb{R}^{n_1}$. The inequalities here are element-wise.

A polyhedral set contains the origin if and only if $g \geq 0$, and includes the origin in its interior if and only if $g > 0$.

**Definition 2.14. (Polytope)** A polytope is a *bounded* polyhedral set.

**Definition 2.15. (Dimension of polytope)** A polytope $P \subset \mathbb{R}^n$ is of dimension $d \leq n$, if there exists a $d-$dimension ball with radius $\varepsilon > 0$ contained in $P$ and there exists no $(d+1)-$dimension ball with radius $\varepsilon > 0$ contained in $P$. A polytope is full dimensional if and only if $d = n$.

**Definition 2.16. (Redundant half-space)** For a given polytope $P(F,g)$, a polyhedral set $P(\overline{F},\overline{g})$ is defined by removing the $i-th$ half-space $F_i$ from matrix $F$ and the corresponding component $g_i$ from vector $g$. The facet $(F_i, g_i)$ is *redundant* if and only if

$$\overline{g_i} < g_i \tag{2.11}$$

where

$$\overline{g_i} = \max_x \{F_i x\}$$
$$\text{subject to: } \overline{F} x \leq \overline{g}$$

**Definition 2.17. (Face, facet, vertex, edge)** A $(n-1)-$dimensional face $F_a^i$ of polytope $P(F,g) \subset \mathbb{R}^n$ is defined as a set of the form

$$F_a^i = \{x \in P : F_i x = g_i\} \tag{2.12}$$

and can be interpreted as the intersection between the polytope and a *non-redundant supporting hyperplane*

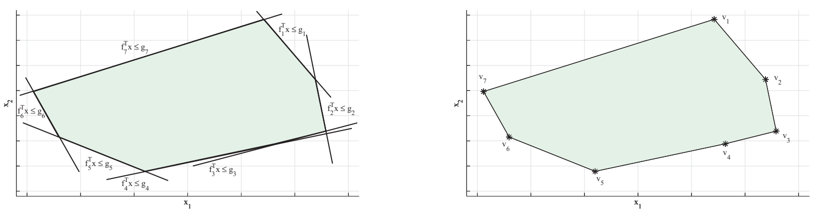
$$F_a^i = P \cap \{x \in \mathbb{R}^n : F_i x = g_i\} \tag{2.13}$$

The non-empty intersection of two faces of dimension $(n-1)$ leads to the description of $(n-2)-$ dimensional face. The faces of the polytope $P$ with dimension 0, 1 and $(n-1)$ are called *vertices*, *edges* and *facets*, respectively.

One of the fundamental properties of polytopes is that it can be presented in half-space representation as in Definition 2.13 or in vertex representation as follows

$$P(V) = \left\{ x \in \mathbb{R}^n : x = \sum_{i=1}^{r} \alpha_i v_i, \quad 0 \leq \alpha_i \leq 1, \quad \sum_{i=1}^{r} \alpha_i = 1 \right\}$$

where $v_i \in \mathbb{R}^{n \times 1}$ denotes the $i-$ column of matrix $V \in \mathbb{R}^{n \times r}$.




(a) Half-space representation (b) Vertex representation

**Fig. 2.1** Exemplification fo the equivalent of half-space and vertex representations of polytopes.

This dual (half-spaces/vertices) representation has very practical consequences in methodological and numerical applications. Due to this duality we are allowed to use either representation in the solving of a particular problem. Note that the transformation from one representation to another may be time-consuming with several well-known algorithms: Fourier-Motzkin elimination [37], CDD [41], Equality Set Projection [63].

Recall that the expression $x = \sum_{i=1}^{r} \alpha_i v_i$ with a given set of vectors $\{v_1, v_2, \ldots, v_r\}$ and

$$\sum_{i=1}^{r} \alpha_i = 1, \quad \alpha_i \geq 0$$

is called *the convex hull* of a set of vectors $\{v_1, v_2, \ldots, v_r\}$ and will be denoted as

$$x = \text{Conv}\{v_1, v_2, \ldots, v_r\}$$

**Definition 2.18. (Simplex)** A simplex $S \in \mathbb{R}^n$ is an $n-$dimensional polytope, which is the convex hull of $n+1$ vertices.

For example, a $2D-$simplex is a triangle, a $3D-$simplex is a tetrahedron, and a $4D-$simplex is a pentachoron.

**Definition 2.19. (Redundant vertex)** For a given polytope $P(V)$, a polyhedral set $P(\overline{V})$ is defined by removing the $i-$th vertex $v_i$ from the matrix $V$. The vertex $v_i$ is *redundant* if and only if

$$p_i < 1 \tag{2.14}$$

where

$$p_i = \min_p \{1^T p\}$$
$$\text{subject to: } \overline{V} p = v_i$$

**Definition 2.20. (Minimal representation)** A half-space or vertex representation of polytope $P$ is *minimal* if and only if the removal of any facet or any vertex would change $P$, i.e. there are no redundant facets or redundant vertices.

Clearly, a minimal representation of a polytope can be achieved by removing from the half-space (vertex) representation all the redundant facets (vertices).

**Definition 2.21. (Normalized representation)** A polytope

$$P(F,g) = \{x \in \mathbb{R}^n : F_i x \leq g_i,\ i = 1, 2, \ldots, n_1\}$$

is in a normalized representation if it has the following property

$$F_i F_i^T = 1$$

A normalized full dimensional polytope has a *unique* minimal representation. This fact is very meaningful in practice, since normalized full dimensional polytopes in minimal representation allow us to avoid any ambiguity when comparing them.

Next, some basic operations on polytopes will be briefly reviewed. Note that although the focus lies on polytopes, most of the operations described here are directly or with minor changes applicable to polyhedral sets. Additional details on polytope computation can be found in [158], [52], [42].

**Definition 2.22. (Intersection)** The intersection of two polytopes $P_1 \subset \mathbb{R}^n$, $P_2 \subset \mathbb{R}^n$ is a polytope

$$P_1 \cap P_2 = \{x \in \mathbb{R}^n : x \in P_1, x \in P_2\}$$

**Definition 2.23. (Minkowski sum)** The Minkowski sum of two polytopes $P_1 \subset \mathbb{R}^n$, $P_2 \subset \mathbb{R}^n$ is a polytope

$$P_1 \oplus P_2 = \{x_1 + x_2 : x_1 \in P_1, x_2 \in P_2\}$$

It is well known [158] that if $P_1$ and $P_2$ are presented in vertex representation, i.e.

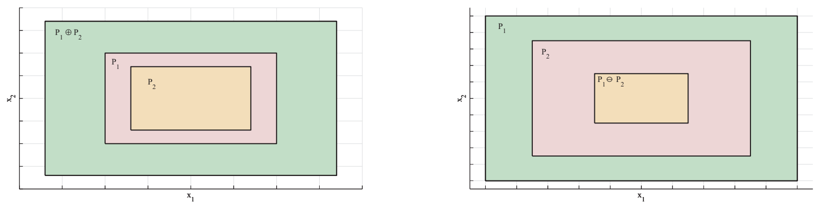
$$P_1 = \text{Conv}\{v_{11}, v_{12}, \dots v_{1p}\},$$
$$P_2 = \text{Conv}\{v_{21}, v_{22}, \dots v_{2q}\}$$

then the Minkowski can be computed as

$$P_1 \oplus P_2 = \text{Conv}\{v_{1i} + v_{2j}\}, \ \forall i = 1, 2, \dots, p, \forall j = 1, 2, \dots, q$$

**Definition 2.24. (Pontryagin difference)** The Pontryagin difference of two polytopes $P_1 \subset \mathbb{R}^n$, $P_2 \subset \mathbb{R}^n$ is a polytope

$$P_1 \ominus P_2 = \{x_1 \in P_1 : x_1 + x_2 \in P_1, \forall x_2 \in P_2\}$$



(a) Minkowski sum $P_1 \oplus P_2$ (b) Pontryagin difference $P_1 \ominus P_2$.

**Fig. 2.2** Minkowski sum and Pontryagin difference of polytopes.

Note that the Pontryagin difference is not the complement of the Minkowski sum. For two polytopes $P_1$ and $P_2$, it holds only that $(P_1 \ominus P_2) \oplus P_2 \subseteq P_1$.

**Definition 2.25. (Projection)** Given a polytope $P \subset \mathbb{R}^{n_1+n_2}$ the orthogonal projection onto the $x_1-$space $\mathbb{R}^{n_1}$ is defined as

$$\text{Proj}_{x_1}(P) = \{x_1 \in \mathbb{R}^{n_1} : \exists x_2 \in \mathbb{R}^{n_2} \text{ such that } [x_1^T \quad x_2^T]^T \in P\}$$

It is well known that the Minkowski sum operation on polytopes in their half-plane representation is complexity-wise equivalent to a projection [158]. Current projection methods for polytopes that can operate in general dimensions can be grouped into four classes: Fourier elimination [69], block elimination [12], vertex based approaches and wrapping-based techniques [63].

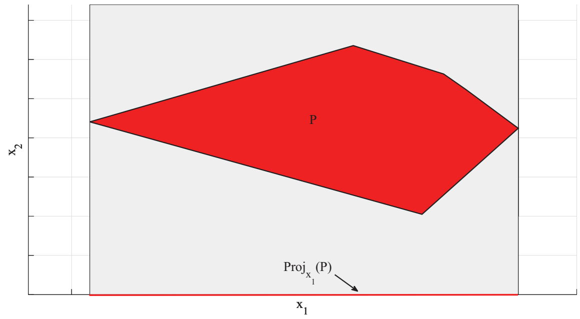



**Fig. 2.3** Projection of a 2-dimensional polytope $P$ onto a line $x_1$.

It is straightforward to see that the complexity of the representation of polytopes is not a function of the space dimension only, but it may be arbitrarily big. For the half-space (or alternatively vertex) representation, the complexity of the polytopes is a linear function of the number of rows of the matrix $F$ (the number of columns of the matrix $V$). As far as the complexity issue concerns, it is worth to be mentioned that none of these representations can be regarded as more convenient. Apparently, one can define an arbitrary polytope with relatively few vertices, however this may nevertheless have a surprisingly large number of facets. This happens, for example when some vertices contribute to many facets. And equally, one can define an arbitrary polytope with relatively few facets, however this may have relatively many more vertices. This happens, for example when some facets have many vertices [42].

The main advantage of the polytopes is their flexibility. It is well known [31] that any convex body can be approximated arbitrarily close by a polytope. Particularly, for a given bounded, convex and closed set $S$ and for a given $\varepsilon$ with $0 < \varepsilon < 1$, then there exists a polytope $P$ such that

$$(1-\varepsilon)S \subseteq P \subseteq S$$

for an inner $\varepsilon-$approximation of the set $S$ and

$$S \subseteq P \subseteq (1+\varepsilon)S$$

for an outer $\varepsilon-$approximation of the set $S$.

## 2.3 Set invariance theory

### 2.3.1 Basic definitions

Set invariance is a fundamental concept in analysis and controller design for constrained systems, since the constraint satisfaction can be guaranteed for all times if and only if the initial states are contained in an invariant set. Two types of systems will be considered in this section, namely, discrete-time uncertain nonlinear systems

$$x(k+1) = f(x(k), w(k)) \tag{2.15}$$

and systems with additional external control inputs

$$x(k+1) = f(x(k), u(k), w(k)) \tag{2.16}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $w(k) \in \mathbb{R}^d$ are respectively the system state, the control input and the unknown disturbance. It is assumed that $f(0,0,0) = 0$ and $f(0,0) = 0$.

The state vector $x(k)$, the control vector $u(k)$ and the disturbance $w(k)$ are subject to constraints

$$\begin{cases} x(k) \in X \\ u(k) \in U \qquad \forall k \geq 0 \\ w(k) \in W \end{cases} \tag{2.17}$$

where the sets $X \subset \mathbb{R}^n$, $U \subset \mathbb{R}^m$ and $W \subset \mathbb{R}^d$ are assumed to be closed and bounded. It is also assumed that the sets $X$, $U$ and $W$ contain the origin their respective interior.

**Definition 2.26. Robust positively invariant set** [23], [70] The set $\Omega \subseteq X$ is robust positively invariant for the system (2.15) if and only if

$$f(x(k), w(k)) \in \Omega$$

for all $x(k) \in \Omega$ and for all $w(k) \in W$.

Hence if the state vector of system (2.15) reaches a robust positively invariant set, it will remain inside the set in spite of disturbance $w(k)$. The term *positively* refers to the fact that only forward evolutions of the system (2.15) are considered and will be omitted in future sections for brevity.

Given a bounded set $X \subset \mathbb{R}^n$, the maximal robustly invariant set $\Omega_{max} \subseteq X$ is a robustly invariant set, that contains all the robustly invariant sets contained in $X$.

**Definition 2.27. Robust contractive set** [23] For a given scalar number $\lambda$ with $0 \leq \lambda \leq 1$, the set $\Omega \subseteq X$ is robust $\lambda-$contractive for the system (2.15) if and only if

$$f(x(k), w(k)) \in \lambda \Omega$$

for all $x(k) \in \Omega$ and for all $w(k) \in W$.

**Definition 2.28. Robust controlled invariant set** [23], [70] The set $C \subseteq X$ is robust controlled invariant for the system (2.16) if for all $x(k) \in C$, there exists a control value $u(k) \in U$ such that

$$x(k+1) = f(x(k), u(k), w(k)) \in C$$

for all $w(k) \in W$.

Given a bounded set $X \subset \mathbb{R}^n$, the maximal robust controlled invariant set $C_{max} \subseteq X$ is a robust controlled invariant set and contains all the robust controlled invariant sets contained in $X$.

**Definition 2.29. Robust controlled contractive set** [23] For a given scalar number $\lambda$ with $0 \leq \lambda < 1$ the set $C \subseteq X$ is robust controlled contractive for the system (2.16) if for all $x(k) \in C$, there exists a control value $u(k) \in U$ such that

$$x(k+1) = f(x(k), u(k), w(k)) \in \lambda C$$

for all $w(k) \in W$.

Obviously, in Definition 2.27 and Definition 2.29 if the contraction factor $\lambda = 1$ we will, respectively retrieve the robust invariance and robust controlled invariance.

### 2.3.2 Problem formulation

From this point on, we will consider the problem of computing an invariant set for the following discrete time linear time-varying or uncertain system

$$x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \tag{2.18}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, $w(k) \in \mathbb{R}^d$ are, respectively the state, input and disturbance vectors.

The matrices $A(k) \in \mathbb{R}^{n \times n}$, $B(k) \in \mathbb{R}^{n \times m}$, $D(k) \in \mathbb{R}^{n \times d}$ satisfy

$$\begin{cases} A(k) = \sum_{i=1}^{q} \alpha_i(k) A_i, \quad B(k) = \sum_{i=1}^{q} \alpha_i(k) B_i, \quad D(k) = \sum_{i=1}^{q} \alpha_i(k) D_i \\ \sum_{i=1}^{q} \alpha_i(k) = 1, \quad \alpha_i(k) \geq 0 \end{cases} \tag{2.19}$$

where the matrices $A_i$, $B_i$ and $D_i$ are the extreme realizations of $A(k)$, $B(k)$ and $D(k)$.

*Remark 2.1.* Note that the numbers of the extreme realizations of $A(k)$, $B(k)$ and $D(k)$ can be different as

$$\begin{cases} A(k) = \sum_{i=1}^{q_1} \alpha_i(k) A_i, \quad B(k) = \sum_{i=1}^{q_2} \beta_i(k) B_i, \quad D(k) = \sum_{i=1}^{q_3} \gamma_i(k) D_i \\ \sum_{i=1}^{q_1} \alpha_i(k) = 1, \alpha_i(k) \geq 0, \ \forall i = 1, 2, \ldots, q_1 \\ \sum_{i=1}^{q_2} \beta_i(k) = 1, \beta_i(k) \geq 0, \ \forall i = 1, 2, \ldots, q_2 \\ \sum_{i=1}^{q_3} \gamma_i(k) = 1, \gamma_i(k) \geq 0, \ \forall i = 1, 2, \ldots, q_3 \end{cases} \tag{2.20}$$

In this case the form of (2.20) can be translated into the form of (2.19) as follows. For simplicity we consider here the case when $D(k) = 0, \forall k \geq 0$, but the extension to the case when $D(k) \neq 0$ is straightforward.

$$\begin{aligned} x(k+1) &= \sum_{i=1}^{q_1} \alpha_i(k) A_i x(k) + \sum_{j=1}^{q_2} \beta_j(k) B_j u(k) \\ &= \sum_{i=1}^{q_1} \alpha_i(k) A_i x(k) + \sum_{i=1}^{q_1} \alpha_i(k) \sum_{j=1}^{q_2} \beta_j(k) B_j u(k) \\ &= \sum_{i=1}^{q_1} \alpha_i(k) \left\{ A_i x(k) + \sum_{j=1}^{q_2} \beta_j(k) B_j u(k) \right\} \\ &= \sum_{i=1}^{q_1} \alpha_i(k) \left\{ \sum_{j=1}^{q_2} \beta_j(k) A_i x(k) + \sum_{j=1}^{q_2} \beta_j(k) B_j u(k) \right\} \\ &= \sum_{i=1}^{q_1} \alpha_i(k) \sum_{j=1}^{q_2} \beta_j(k) \left\{ A_i x(k) + B_j u(k) \right\} \\ &= \sum_{i=1}^{q_1} \sum_{j=1}^{q_2} \alpha_i(k) \beta_j(k) \left\{ A_i x(k) + B_j u(k) \right\} \end{aligned}$$

Consider the polytope $P_c$, the vertices of which are given by taking all possible combinations of $\{A_i, B_j\}$ with $i = 1, 2, \ldots, q_1$ and $j = 1, 2, \ldots, q_2$. Since

$$\sum_{i=1}^{q_1} \sum_{j=1}^{q_2} \alpha_i(k) \beta_j(k) = \sum_{i=1}^{q_1} \alpha_i(k) \sum_{j=1}^{q_2} \beta_j(k) = 1$$

it is clear that $\{A(k), B(k)\}$ can be expressed as a convex combination of the vertices of $P_c$.

The state, the control and the disturbance are subject to the following polytopic constraints

$$\begin{cases} x(k) \in X, X = \{x \in \mathbb{R}^n : F_x x \leq g_x\} \\ u(k) \in U, U = \{u \in \mathbb{R}^m : F_u u \leq g_u\} \\ w(k) \in W, W = \{w \in \mathbb{R}^d : F_w w \leq g_w\} \end{cases} \tag{2.21}$$

where the matrices $F_x$, $F_u$, $F_w$ and the vectors $g_x$, $g_u$, $g_w$ are assumed to be constant with $g_x > 0$, $g_u > 0$, $g_w > 0$ such that the origin is contained in the interior of $X$, $U$ and $W$. Recall that the inequalities are element-wise.

### 2.3.3 Ellipsoidal invariant sets

Ellipsoidal sets are the most commonly used for robust stability analysis and controller synthesis of constrained systems. Their popularity is due to computational efficiency via the use of LMI formulations and the complexity is fixed with respect to the dimension of the state space [29], [137]. This approach, however may lead to conservative results.

For simplicity, in this subsection, the case of vanishing disturbances is considered. In other words, the system under consideration is

$$x(k+1) = A(k)x(k) + B(k)u(k) \tag{2.22}$$

Due to the symmetric properties of ellipsoids, it is clear that the ellipsoidal invariant sets are less conservative in the case when the constraints on the state and control vector are symmetric, i.e.

$$\begin{cases} x(k) \in X,\ X = \{x : |F_i x| \leq 1\}, \forall i = 1,2,\ldots,n_1 \\ u(k) \in U,\ U = \{u : |u_i| \leq u_{imax}\}, \forall i = 1,2,\ldots,m \end{cases} \tag{2.23}$$

where $u_{imax}$ is the $i-$component of vector $u_{max} \in \mathbb{R}^{m\times 1}$.

Let us consider now the problem of checking robust controlled invariance. The ellipsoid $E(P) = \{x \in \mathbb{R}^n : x^T P^{-1} x \leq 1\}$ is controlled invariant if and only if for all $x \in E(P)$ there exists an input $u = \Phi(x) \in U$ such that

$$(A_i x + B_i \Phi(x))^T P^{-1} (A_i x + B_i \Phi(x)) \leq 1 \tag{2.24}$$

for all $i = 1,2,\ldots,q$, where $q$ is the cardinal of the set of vertices in $P_c$.

It is well known [27] that for the time-varying or uncertain linear discrete-time system (2.22), it is sufficient to check condition (2.24) only for all $x$ on the boundary of $E(P)$, i.e. for all $x$ such that $x^T P^{-1} x = 1$. Therefore condition (2.24) can be transformed into

$$(A_i x + B_i \Phi(x))^T P^{-1} (A_i x + B_i \Phi(x)) \leq x^T P^{-1} x,\ \forall i = 1,2,\ldots,q \tag{2.25}$$

One possible choice for $u = \Phi(x)$ is a linear state feedback controller $u = Kx$. By denoting $A_{ci} = A_i + B_i K$ with $i = 1,2,\ldots,q$, condition (2.25) is equivalent to

$$x^T A_{ci}^T P^{-1} A_{ci} x \leq x^T P^{-1} x,\ \forall i = 1,2,\ldots,q$$

or

$$A_{ci}^T P^{-1} A_{ci} \preceq P^{-1},\ \forall i = 1,2,\ldots,q$$

By using the Schur complement, this condition can be rewritten as

$$\begin{bmatrix} P^{-1} & A_{ci}^T \\ A_{ci} & P \end{bmatrix} \succeq 0,\ \forall i = 1,2,\ldots,q$$

The condition provided here is not linear in $P$. By using the Schur complement again, one gets

$$P - A_{ci}PA_{ci}^T \succeq 0,\ \forall i = 1,2,\ldots,q$$

or

$$\begin{bmatrix} P & A_{ci}P \\ PA_{ci}^T & P \end{bmatrix} \succeq 0,\ \forall i = 1,2,\ldots,q$$

By substituting $A_{ci} = A_i + B_iK$ with $i = 1,2,\ldots,q$, one obtains

$$\begin{bmatrix} P & A_iP + B_iKP \\ PA_i^T + PK^TB_i^T & P \end{bmatrix} \succeq 0,\ \forall i = 1,2,\ldots,q$$

Though this condition is nonlinear (in fact bilinear since $P$ and $K$ are the unknowns). Still it can be re-parameterized into a linear condition by setting $Y = KP$. The above condition is equivalent to

$$\begin{bmatrix} P & A_iP + B_iY \\ PA_i^T + Y^TB_i^T & P \end{bmatrix} \succeq 0,\ \forall i = 1,2,\ldots,q \tag{2.26}$$

Condition (2.26) is necessary and sufficient for ellipsoid $E(P)$ with linear state feedback $u = Kx$ to be robustly invariant. Concerning the constraint satisfaction (2.23), based on equation (2.7) it is obvious that

- The state constraints are satisfied in closed loop if and only if $E(P)$ is a subset of $X$, hence
$$\begin{bmatrix} 1 & F_iP \\ PF_i^T & P \end{bmatrix} \succeq 0,\ \forall i = 1,2,\ldots,n_1 \tag{2.27}$$
- The input constraints are satisfied in closed loop if and only if $E(P)$ is a subset of a polyhedral set $X_u$ where
$$X_u = \{x \in \mathbb{R}^n : |K_ix| \le u_{imax}\}$$
for $i = 1,2,\ldots,m$ and $K_i$ is the $i-$row of the matrix $K \in \mathbb{R}^{m\times n}$, hence
$$\begin{bmatrix} u_{imax}^2 & K_iP \\ PK_i^T & P \end{bmatrix} \succeq 0,$$
By noticing that $K_iP = Y_i$ with $Y_i$ is the $i-$row of the matrix $Y \in \mathbf{R}^{m\times n}$, one gets
$$\begin{bmatrix} u_{imax}^2 & Y_i \\ Y_i^T & P \end{bmatrix} \succeq 0 \tag{2.28}$$
Define a row vector $T_i \in \mathbb{R}^m$ as follows
$$T_i = [0 \quad 0 \quad \ldots \quad 0 \quad \underbrace{1}_{i-th\ position} \quad 0 \quad \ldots \quad 0 \quad 0]$$
It is clear that $Y_i = T_iY$. Therefore equation (2.28) can be transformed into

$$\begin{bmatrix} u_{imax}^2 & T_i Y \\ Y^T T_i^T & P \end{bmatrix} \succeq 0 \tag{2.29}$$

With all the ellipsoids satisfying invariance condition (2.26) and constraint satisfaction (2.27), (2.28), we would like to choose among them the *largest* ellipsoid. In the literature, the size of ellipsoid $E(P)$ is usually measured by the determinant or the trace of matrix $P$, see for example [150]. Here the trace of matrix $P$ is chosen due to its linearity. The trace of a square matrix is defined to be the sum of the elements on the main diagonal of the matrix. Maximization of the trace of matrices corresponds to the search for the maximal sum of eigenvalues of matrices. With the trace of matrix as the objective function, the problem of choosing the *largest* robustly invariant ellipsoid can be formulated as

$$J = \max_{P,Y} \{trace(P)\} \tag{2.30}$$

subject to

- Invariance condition (2.26)
- Constraints satisfaction (2.27), (2.29)

It is clear that the solution $P$, $Y$ of problem (2.30) may lead to the controller $K = YP^{-1}$ such that the closed loop system with matrices $A_{ci} = A_i + B_i K$, $i = 1, 2, \ldots, q$ is at the stability margin. In other words, the ellipsoid $E(P)$ thus obtained might not be contractive (although being invariant). Indeed, the system trajectories might not converge to the origin. In order to ensure $x(k) \to 0$ as $k \to \infty$, it is required that for all $x$ on the boundary of $E(P)$, i.e. for all $x$ such that $x^T P^{-1} x = 1$, to have

$$(A_i x + B_i \Phi(x))^T P^{-1} (A_i x + B_i \Phi(x)) < 1 \; \forall i = 1, 2, \ldots, q$$

With the same argument as above, one can conclude that the ellipsoid $E(P)$ with the linear controller $u = Kx$ is robust contractive if the following set of LMI conditions is satisfied

$$\begin{bmatrix} P & A_i P + B_i Y \\ P A_i^T + Y^T B_i^T & P \end{bmatrix} \succ 0 \; \forall i = 1, 2, \ldots, q \tag{2.31}$$

### *2.3.4 Polyhedral invariant sets*

The problem of invariance description using polyhedral sets is addressed in this section. With linear constraints on state and control variables, polyhedral invariant sets are preferred to the ellipsoidal invariant sets, since they offer a better approximation of the domain of attraction [35], [55], [21]. To begin, let us consider the case, when the control input is in the form of state feedback $u(k) = Kx(k)$. Then the closed loop system of (2.18) is in the form

$$x(k+1) = A_c(k) x(k) + D(k) w(k) \tag{2.32}$$

where

$$A_c(k) = A(k) + B(k)K = \text{Conv}\{A_{ci}\}$$

with $A_{ci} = A_i + B_i K$, $i = 1,2,\ldots,q$.

The state constraints of the closed loop system are in the form

$$x \in X_c, \quad X_c = \{x \in \mathbb{R}^n : F_c x \le g_c\} \tag{2.33}$$

where

$$F_c = \begin{bmatrix} F_x \\ F_u K \end{bmatrix}, \quad g_c = \begin{bmatrix} g_x \\ g_u \end{bmatrix}$$

The following definition plays an important role in computing robustly invariant sets for system (2.32) with constraints (2.33).

**Definition 2.30. (Pre-image set)** For the system (2.32), the one step admissible *pre-image set* of the set $X_c$ is a set $X_c^1 \subseteq X_c$ such that for all $x \in X_c^1$, it holds that

$$A_{ci}x + D_i w \in X_c$$

for all $w \in W$ and for all $i = 1,2,\ldots,q$.

The pre-image set $Pre(X_c)$ can be defined by [26], [22]

$$X_c^1 = \left\{ x \in X_c : F_c A_{ci} x \le g_c - \max_{w \in W}\{F_c D_i w\} \right\} \tag{2.34}$$

for all $i = 1,2,\ldots,q$.

*Example 2.1.* Consider the following uncertain system

$$x(k+1) = A(k)x(k) + Bu(k) + Dw(k)$$

where

$$A(k) = \alpha(k)A_1 + (1-\alpha(k))A_2$$
$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

with $0 \le \alpha(k) \le 1$ and

$$A_1 = \begin{bmatrix} 1.1 & 1 \\ 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.6 & 1 \\ 0 & 1 \end{bmatrix}$$

The constraints on the state, on the input and on the disturbance (2.21) have the particular realization given by the matrices

$$F_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad g_x = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}$$
$$F_u = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad g_u = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$
$$F_w = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad g_w = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}$$

or equivalently $-3 \leq x_i \leq 3$, $i = 1,2$ and $-2 \leq u \leq 2$ and $-0.2 \leq w_i \leq 0.2$, $i = 1,2$.

The robust stabilizing feedback controller is chosen as

$$K = [-0.3856 \quad -1.0024]$$

With this feedback controller the closed loop matrices are

$$A_{c1} = \begin{bmatrix} 1.1000 & 1.0000 \\ -0.3856 & -0.0024 \end{bmatrix}, \quad A_{c2} = \begin{bmatrix} 0.6000 & 1.0000 \\ -0.3856 & -0.0024 \end{bmatrix}$$

The state constraint set $X_c$ is

$$X_c = \left\{ x \in \mathbb{R}^2 : F_c x \leq g_c \right\}$$

where

$$F_c = \begin{bmatrix} 1.0000 & 0 \\ 0 & 1.0000 \\ -1.0000 & 0 \\ 0 & -1.0000 \\ -0.3856 & -1.0024 \\ 0.3856 & 1.0024 \end{bmatrix}, \quad g_c = \begin{bmatrix} 3.0000 \\ 3.0000 \\ 3.0000 \\ 3.0000 \\ 2.0000 \\ 2.0000 \end{bmatrix}$$

By solving the LP problem (2.11), it follows that the half-spaces $[0 \quad 1]x \leq 3$ and $[0 \quad -1]x \leq 3$ are redundant. After eliminating these redundant half-spaces, the state constraint set $X_c$ is presented in minimal normalized half-space representation as

$$X_c = \left\{ x \in \mathbb{R}^2 : \widehat{F}_c x \leq \widehat{g}_c \right\}$$

where

$$\widehat{F}_c = \begin{bmatrix} 1.0000 & 0 \\ -1.0000 & 0 \\ -0.3590 & -0.9333 \\ 0.3590 & 0.9333 \end{bmatrix}, \quad \widehat{g}_c = \begin{bmatrix} 3.0000 \\ 3.0000 \\ 0.9311 \\ 0.9311 \end{bmatrix}$$

Based on equation (2.34), the one step admissible pre-image set $X_c^1$ of the set $X_c$ is defined as

$$X_c^1 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} \widehat{F}_c \\ \widehat{F}_c A_1 \\ \widehat{F}_c A_2 \end{bmatrix} x \le \begin{bmatrix} \widehat{g}_c \\ \widehat{g}_c - \max\limits_{w \in W}\{\widehat{F}_c w\} \\ \widehat{g}_c - \max\limits_{w \in W}\{\widehat{F}_c w\} \end{bmatrix} \right\} \tag{2.35}$$

After removing redundant inequalities, the set $X_c^1$ is represented in minimal normalized half-space representation as

$$X_c^1 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.0000 & 0 \\ -1.0000 & 0 \\ -0.3590 & -0.9333 \\ 0.3590 & 0.9333 \\ 0.7399 & 0.6727 \\ -0.7399 & -0.6727 \\ 0.3753 & -0.9269 \\ -0.3753 & 0.9269 \end{bmatrix} x \le \begin{bmatrix} 3.0000 \\ 3.0000 \\ 0.9311 \\ 0.9311 \\ 1.8835 \\ 1.8835 \\ 1.7474 \\ 1.7474 \end{bmatrix} \right\}$$

The sets $X$, $X_c$ and $X_c^1$ are depicted in Figure 2.4.

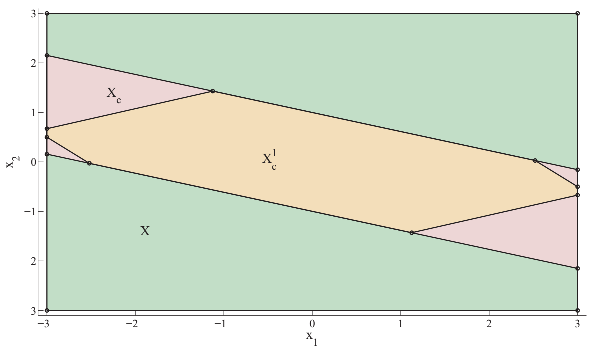



**Fig. 2.4** One step pre-image set for example 2.1.

It is clear that the set $\Omega \subseteq X_c$ is robustly invariant if it equals to its one step admissible pre-image set, that is for all $x \in \Omega$ and for all $w \in W$, it holds that

$$A_i x + D_i w \in \Omega$$

for all $i = 1, 2, \ldots, q$. Based on this observation, the following algorithm can be used for computing a robustly invariant set for system (2.32) with respect to constraints (2.33)

**Procedure 2.1: Robustly invariant set computation [45], [71]**

- **Input:** The matrices $A_{c1}, A_{c2}, \ldots, A_{cq}$, $D_1, D_2, \ldots, D_q$ and $X_c = \{x \in \mathbb{R}^n : F_c x \le g_c\}$ and the set W.
- **Output:** The robustly invariant set $\Omega$.

1. Set $i = 0$, $F_0 = F_c$, $g_0 = g_c$ and $X_0 = \{x \in \mathbb{R}^n : F_0 x \le g_0\}$.
2. Set $X_i = X_0$.
3. Eliminate redundant inequalities of the following polytope

$$P = \left\{ x \in \mathbb{R}^n : \begin{bmatrix} F_0 \\ F_0 A_{c1} \\ F_0 A_{c2} \\ \vdots \\ F_0 A_{cq} \end{bmatrix} x \le \begin{bmatrix} g_0 \\ g_0 - \max\limits_{w \in W} \{F_0 D_1 w\} \\ g_0 - \max\limits_{w \in W} \{F_0 D_2 w\} \\ \vdots \\ g_0 - \max\limits_{w \in W} \{F_0 D_q w\} \end{bmatrix} \right\}$$

4. Set $X_0 = P$ and update consequently the matrices $F_0$ and $g_0$.
5. If $X_0 = X_i$ then stop and set $\Omega = X_0$. Else continue.
6. Set $i = i + 1$ and go to step 2.

The natural question for procedure 2.1 is that if there exists a finite index $i$ such that $X_0 = X_i$, or equivalently if procedure 2.1 terminates after a finite number of iterations.

In the absence of disturbances, the following theorem holds [24].

**Theorem 2.1.** *[24] Assume that the system (2.32) is robustly asymptotically stable. Then there exists a finite index $i = i_{max}$, such that $X_0 = X_i$ in Procedure* 2.1.

*Remark 2.2.* In the presence of disturbances, a necessary and sufficient condition for the existence of a finite index $i$ is that a minimal robustly invariant set[5] [76], [127], [116] is a subset of $X_c$. We will come back to this problem later in Chapter 6, when we deal with a peak to peak controller.

Apparently the sensitive part of procedure 2.1 is step 5. Checking the equality of two polytopes $X_0$ and $X_i$ is computationally demanding, i.e. one has to check $X_0 \subseteq X_i$ and $X_i \subseteq X_0$. Note that if at the step $i$ of procedure 2.1 the set $\Omega$ is invariant then the following set of inequalities

[5] The set $\Omega$ is minimal robustly invariant if it is a robustly invariant set and is a subset of any robustly invariant set.

$$\begin{bmatrix} F_0A_{c1} \\ F_0A_{c2} \\ \vdots \\ F_0A_{cq} \end{bmatrix} x \leq \begin{bmatrix} g_0 - \max\limits_{w\in W}\{F_0D_1w\} \\ g_0 - \max\limits_{w\in W}\{F_0D_2w\} \\ \vdots \\ g_0 - \max\limits_{w\in W}\{F_0D_qw\} \end{bmatrix}$$

is redundant with respect to the set $\Omega$

$$\Omega = \{x \in \mathbb{R}^n : F_0x \leq g_0\}$$

Hence the procedure 2.1 can be modified for computing a robustly invariant set as follows

**Procedure 2.2: Robustly invariant set computation**

---

- **Input:** The matrices $A_{c1}$,$A_{c2}$,...,$A_{cq}$, $D_1$,$D_2$,...,$D_q$, the set $X_c = \{x \in \mathbb{R}^n : F_cx \leq g_c\}$ and the set $W$.
- **Output:** The robustly invariant set $\Omega$.

1. Set $i = 0$, $F_0 = F_c$, $g_0 = g_c$ and $X_0 = \{x \in \mathbb{R}^n : F_0x \leq g_0\}$.
2. Consider the following polytope

$$P = \left\{ x \in \mathbb{R}^n : \begin{bmatrix} F_0 \\ F_0A_{c1} \\ F_0A_{c2} \\ \vdots \\ F_0A_{cq} \end{bmatrix} x \leq \begin{bmatrix} g_0 \\ g_0 - \max\limits_{w\in W}\{F_0D_1w\} \\ g_0 - \max\limits_{w\in W}\{F_0D_2w\} \\ \vdots \\ g_0 - \max\limits_{w\in W}\{F_0D_qw\} \end{bmatrix} \right\}$$

   and iteratively check the redundancy of the subsets starting from the following set of inequalities

$$\{x \in \mathbb{R}^n : F_0A_{cj}x \leq g_0 - \max_{w\in W}\{F_0D_jw\}\}$$

   with $j = 1,2,\ldots,q$.
3. If all of the inequalities are redundant with respect to $X_0$, then stop and set $\Omega = X_0$. Else continue.
4. Set $X_0 = P$
5. Set $i = i+1$ and go to step 2.

It is well known [45], [76], [27] that the set $\Omega$ resulting from procedure 2.1 or procedure 2.2, turns out to be the maximal robustly invariant set for for system (2.32) with respect to constraints (2.21), that is $\Omega = \Omega_{max}$.

*Example 2.2.* Consider the uncertain system in example 2.1 with the same constraints on the state, on the input and on the disturbance. Applying procedure 2.2, the maximal robustly invariant set is obtained after 5 iterations as

$$\Omega_{max} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.3590 & -0.9333 \\ 0.3590 & 0.9333 \\ 0.6739 & 0.7388 \\ -0.6739 & -0.7388 \\ 0.8979 & 0.4401 \\ -0.8979 & -0.4401 \\ 0.3753 & -0.9269 \\ -0.3753 & 0.9269 \end{bmatrix} x \leq \begin{bmatrix} 0.9311 \\ 0.9311 \\ 1.2075 \\ 1.2075 \\ 1.7334 \\ 1.7334 \\ 1.7474 \\ 1.7474 \end{bmatrix} \right\}$$

The sets $X$, $X_c$ and $\Omega_{max}$ are depicted in Figure 2.5.

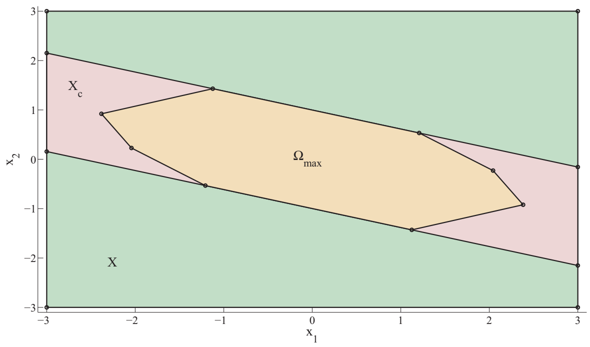



**Fig. 2.5** Maximal robustly invariant set $\Omega_{max}$ for example 2.2.

**Definition 2.31. (One step robust controlled set)** Given the polytopic system (2.18), the one step robust controlled set of the set $C_0 = \{x \in \mathbb{R}^n : F_0 x \leq g_0\}$ is given by all states that can be steered in one step in to $C_0$ when a suitable control action is applied. The one step robust controlled set denoted as $C_1$ can be shown to be [26], [22]

$$C_1 = \left\{ x \in \mathbb{R}^n : \exists u \in U : F_0(A_i x + B_i u) \leq g_0 - \max_{w \in W}\{F_0 D_i w\} \right\} \tag{2.36}$$

for all $w \in W$ and for all $i = 1, 2, \ldots, q$

*Remark 2.3.* If the set $C_0$ is robustly invariant, then $C_0 \subseteq C_1$. Hence $C_1$ is a robust controlled invariant set.

Recall that the set $\Omega_{max}$ is a maximal robustly invariant set with respect to a predefined control law $u(k) = Kx(k)$. Define $C_N$ as the set of all states, that can be steered to $\Omega_{max}$ in no more than $N$ steps along an admissible trajectory, i.e. a trajectory satisfying control, state and disturbance constraints. This set can be generated recursively by the following procedure:

**Procedure 2.3: Robust N-step controlled invariant set computation**

- **Input:** The matrices $A_1, A_2, \ldots, A_q$, $D_1, D_2, \ldots, D_q$ and the sets $X$, $U$, $W$ and the maximal robustly invariant set $\Omega_{max}$
- **Output:** The N-step robust controlled invariant set $C_N$

1. Set $i = 0$ and $C_0 = \Omega_{max}$ and let the matrices $F_0$, $g_0$ be the half space representation of the set $C_0$, i.e. $C_0 = \{x \in \mathbb{R}^n : F_0 x \leq g_0\}$
2. Compute the expanded set $P_i \subset \mathbb{R}^{n+m}$

$$P_i = \left\{ (x,u) \in \mathbb{R}^{n+m} : \begin{bmatrix} F_i(A_1 x + B_1 u) \\ F_i(A_2 x + B_2 u) \\ \vdots \\ F_i(A_q x + B_q u) \end{bmatrix} \leq \begin{bmatrix} g_i - \max\limits_{w \in W}\{F_i D_1 w\} \\ g_i - \max\limits_{w \in W}\{F_i D_2 w\} \\ \vdots \\ g_i - \max\limits_{w \in W}\{F_i D_q w\} \end{bmatrix} \right\}$$

3. Compute the projection of $P_i$ on $\mathbb{R}^n$

$$P_i^n = \{x \in \mathbb{R}^n : \exists u \in U \text{ such that } (x,u) \in P_i\}$$

4. Set

$$C_{i+1} = P_i^n \cap X$$

and let the matrices $F_{i+1}$, $g_{i+1}$ be the half space representation of the set $C_{i+1}$, i.e.

$$C_{i+1} = \{x \in \mathbb{R}^n : F_{i+1} x \leq g_{i+1}\}$$

5. If $C_{i+1} = C_i$, then stop and set $C_N = C_i$. Else continue.
6. If $i = N$, then stop else continue.
7. Set $i = i+1$ and go to step 2.

Since $\Omega_{max}$ is a robustly invariant set, it follows that for each $i$, $C_{i-1} \subseteq C_i$ and therefore $C_i$ is a robust controlled invariant set and a sequence of nested polytopes.

Note that the complexity of the set $C_N$ does not have an analytic dependence on $N$ and may increase without bound, thus placing a practical limitation on the choice of $N$.

*Example 2.3.* Consider the uncertain system in example 2.1. The constraints on the state, on the input and on the disturbance are the same.

Using procedure 2.3, one obtains the robust controlled invariant sets $C_N$ as shown in Figure 2.6 with $N = 1$ and $N = 7$. The set $C_1$ is a set of all states that can be steered in one step in $\Omega_{max}$ when a suitable control action is applied. The set $C_7$ is a set of all states that can be steered in seven steps in $\Omega_{max}$ when a suitable control action is applied. Note that $C_7 = C_8$, therefore $C_7$ is the maximal robust controlled invariant set.

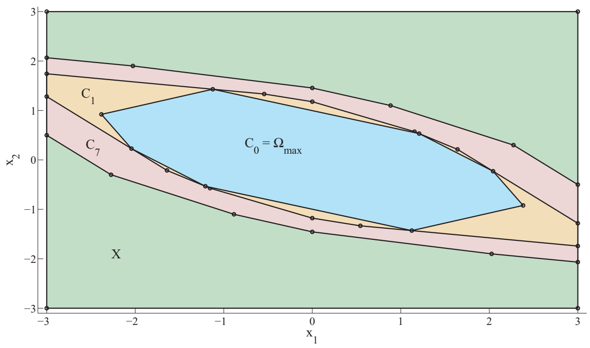



**Fig. 2.6** Robust controlled invariant set for example 2.3.

The set $C_7$ is presented in minimal normalized half-space representation as

$$C_7 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.3731 & 0.9278 \\ -0.3731 & -0.9278 \\ 0.4992 & 0.8665 \\ -0.4992 & -0.8665 \\ 0.1696 & 0.9855 \\ -0.1696 & -0.9855 \\ 0.2142 & 0.9768 \\ -0.2142 & -0.9768 \\ 0.7399 & 0.6727 \\ -0.7399 & -0.6727 \\ 1.0000 & 0 \\ -1.0000 & 0 \end{bmatrix} x \leq \begin{bmatrix} 1.3505 \\ 1.3505 \\ 1.3946 \\ 1.3946 \\ 1.5289 \\ 1.5289 \\ 1.4218 \\ 1.4218 \\ 1.8835 \\ 1.8835 \\ 3.0000 \\ 3.0000 \end{bmatrix} \right\}$$

## 2.4 On the domains of attraction

This section presents an original contribution on estimating the domain of attraction for uncertain and time-varying linear discrete-times systems in closed-loop with a *saturated* linear feedback controller and state constraints. Ellipsoidal and polyhedral sets will be used for characterizing the domain of attraction. The use of ellipsoidal sets associated with its simple characterization as a solution of an LMI problem, while the use of polyhedral sets offers a better approximation of the domain of attraction.

### 2.4.1 Problem formulation

Consider the following time-varying or uncertain linear discrete-time system

$$x(k+1) = A(k)x(k) + B(k)u(k) \tag{2.37}$$

where

$$\begin{cases} A(k) = \sum_{i=1}^{q} \alpha_i(k)A_i, \quad B(k) = \sum_{i=1}^{q} \alpha_i(k)B_i \\ \sum_{i=1}^{q} \alpha_i(k) = 1, \quad \alpha_i(k) \geq 0 \end{cases} \tag{2.38}$$

with given matrices $A_i \in \mathbb{R}^{n\times n}$ and $B_i \in \mathbb{R}^{n\times m}$, $i = 1,2,\ldots,q$.

Both the state vector $x(k)$ and the control vector $u(k)$ are subject to the constraints

$$\begin{cases} x(k) \in X,\ X = \{x \in \mathbb{R}^n : F_i x \leq g_i\}, \forall i = 1,2,\ldots,n_1 \\ u(k) \in U,\ U = \{u \in \mathbb{R}^m : u_{il} \leq u_i \leq u_{iu}\}, \forall i = 1,2,\ldots,m \end{cases} \tag{2.39}$$

where $F_i \in \mathbb{R}^n$ is the $i-th$ row of the matrix $F_x \in \mathbb{R}^{n_1\times n}$, $g_i$ is the $i-th$ component of the vector $g_x \in \mathbb{R}^{n_1}$, $u_{il}$ and $u_{iu}$ are respectively the $i-th$ component of the vectors $u_l$ and $u_u$, which are the lower and upper bounds of input $u$. It is assumed that the matrix $F_x$ and the vectors $u_l \in \mathbb{R}^m$, $u_u \in \mathbb{R}^m$ are constant with $u_l < 0$ and $u_u > 0$ such that the origin is contained in the interior of $X$ and $U$.

Assume that using established results in control theory, one can find a feedback controller $K \in \mathbb{R}^{m\times n}$ such that

$$u(k) = Kx(k) \tag{2.40}$$

robustly quadratically stabilizes system (2.37). We would like to estimate the domain of attraction of the origin for the closed loop system

$$x(k+1) = A(k)x(k) + B(k)sat(Kx(k)) \tag{2.41}$$

where the state vector and the control vector are subject to the constraints (2.39).

### 2.4.2 Saturation nonlinearity modeling- A linear differential inclusion approach

In this section, a linear differential inclusion approach used for modeling the saturation function is briefly reviewed. This modeling framework was first proposed by

Hu et al. in [57], [59], [60]. Then its generalization was developed by Alamo et al. [6], [7]. The main idea of the differential inclusion approach is to use an auxiliary vector variable $v \in \mathbb{R}^m$, and to compose the output of the saturation function as a convex combination of the actual control signals $u$ and $v$.

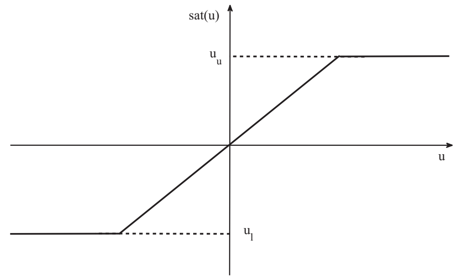



**Fig. 2.7** The saturation function

The saturation function is defined as follows

$$sat(u_i) = \begin{cases} u_{il}, & \text{if } u_i \le u_{il} \\ u, & \text{if } u_{il} \le u_i \le u_{iu} \\ u_{iu}, & \text{if } u_{iu} \le u_i \end{cases} \tag{2.42}$$

for $i = 1, 2, \ldots, m$ and $u_{il}$ and $u_{iu}$ are respectively, the upper bound and the lower bound of $u_i$.

To underline the details of the approach, let us first consider the case when $m = 1$. In this case $u$ and $v$ will be scalars. It is clear that for any arbitrarily $u$, there exist $v$ and $\beta$ such that

$$sat(u) = \beta u + (1 - \beta)v \tag{2.43}$$

where $0 \le \beta \le 1$ and

$$u_l \le v \le u_u \tag{2.44}$$

or equivalently

$$sat(u) \in \text{Conv}\{u, v\} \tag{2.45}$$

Figure 2.8 illustrates this fact.

Analogously, for $m = 2$ and $v$ such that

$$\begin{cases} u_{1l} \le v_1 \le u_{1u} \\ u_{2l} \le v_2 \le u_{2u} \end{cases} \tag{2.46}$$

the saturation function can be expressed as

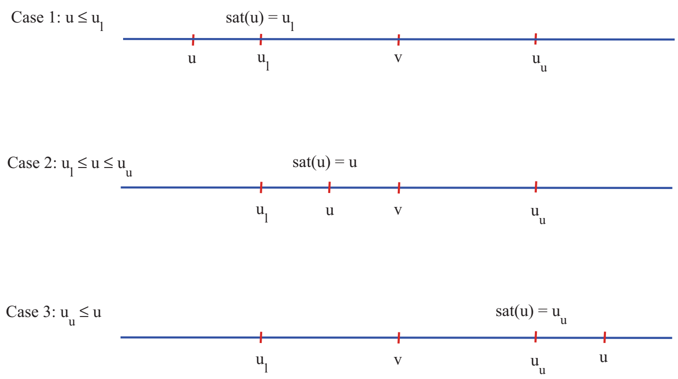



**Fig. 2.8** Linear differential inclusion approach.

$$sat(u) = \beta_1 \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \beta_2 \begin{bmatrix} u_1 \\ v_2 \end{bmatrix} + \beta_3 \begin{bmatrix} v_1 \\ u_2 \end{bmatrix} + \beta_4 \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{2.47}$$

where

$$\sum_{i=1}^{4} \beta_i = 1, \quad \beta_i \geq 0 \tag{2.48}$$

or equivalently

$$sat(u) \in \text{Conv}\left\{ \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \begin{bmatrix} u_1 \\ v_2 \end{bmatrix}, \begin{bmatrix} v_1 \\ u_2 \end{bmatrix}, \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \right\} \tag{2.49}$$

Denote now $D_m$ as the set of $m \times m$ diagonal matrices whose diagonal elements are either 0 or 1. For example, if $m = 2$ then

$$D_2 = \left\{ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\}$$

There are $2^m$ elements in $D_m$. Denote each element of $D_m$ as $E_i$, $i = 1, 2, \ldots, 2^m$ and define $E_i^- = I - E_i$. For example, if

$$E_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

then

$$E_1^- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Clearly if $E_i \in D_m$, then $E_i^-$ is also in $D_m$. The generalization of the results (2.45) (2.49) is reported by the following lemma [57], [59], [60]

**Lemma 2.1.** *[59] Consider two vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^m$ such that $u_{il} \le v_i \le u_{iu}$ for all $i = 1, 2, \ldots, m$, then it holds that*

$$sat(u) \in Conv\{E_i u + E_i^- v\}, i = 1, 2, \ldots, 2^m \tag{2.50}$$

Consequently, there exist $\beta_i$ with $i = 1, 2, \ldots, 2^m$ and

$$\beta_i \ge 0 \text{ and } \sum_{i=1}^{2^m} \beta_i = 1$$

such that

$$sat(u) = \sum_{i=1}^{2^m} \beta_i (E_i u + E_i^- v)$$

### 2.4.3 The ellipsoidal set approach

The aim of this subsection is twofold. First, we provide an invariance condition of ellipsoidal sets for discrete-time linear time-varying or uncertain systems with a saturated input and state constraints [56]. This invariance condition is an extended version of the previously published results in [59] for the robust case. Secondly, we propose a method for computing a nonlinear controller $u(k) = sat(Kx(k))$, which makes a given ellipsoid invariant. For simplicity, consider the case of bounds equal to $u_{max}$, namely

$$-u_l = u_u = u_{max}$$

and let us assume that the polyhedral constraint set $X$ is symmetric with $g_i = 1$, for all $i = 1, 2, \ldots, n_1$. Clearly, this assumption is nonrestrictive as long as

$$F_i x \le g_i \Leftrightarrow \frac{F_i}{g_i} x \le 1$$

for all $g_i > 0$.

For a matrix $H \in \mathbb{R}^{m \times n}$, define $X_c$ as an intersection between the state constraint set $X$ and the polyhedral set $F(H, u_{max}) = \{x : |Hx| \le u_{max}\}$, i.e.

$$X_c = \left\{ x \in \mathbb{R}^n : \begin{bmatrix} F_x \\ H \\ -H \end{bmatrix} x \le \begin{bmatrix} 1 \\ u_{max} \\ u_{max} \end{bmatrix} \right\}$$

We are now ready to state the main result of this subsection

**Theorem 2.2.** *If there exist a symmetric matrix $P \in \mathbb{R}^{n \times n}$ and a matrix $H \in \mathbb{R}^{m \times n}$ such that*

$$\begin{bmatrix} P & \{A_i + B_i(E_jK + E_j^- H)\}P \\ P\{A_i + B_i(E_jK + E_j^- H)\}^T & P \end{bmatrix} \succeq 0, \tag{2.51}$$

*for $\forall i = 1, 2, \ldots, q$, $\forall j = 1, \ldots, 2^m$ and $E(P) \subset X_c$, then the ellipsoid $E(P)$ is a robustly invariant set for the system (2.41) with constraints (2.39).*

*Proof.* Assume that there exist a matrix $P$ and a matrix $H$ such that condition (2.51) is satisfied. Based on Lemma 2.1 and by choosing $v = Hx$, one has

$$sat(Kx) = \sum_{j=1}^{2^m} \beta_j (E_jKx + E_j^- Hx)$$

for all $x$ such that $|Hx| \leq u_{max}$. Subsequently

$$\begin{aligned} x(k+1) &= \sum_{i=1}^{q} \alpha_i(k) \left\{ A_i + B_i \sum_{j=1}^{2^m} \beta_j (E_jK + E_j^- H) \right\} x(k) \\ &= \sum_{i=1}^{q} \alpha_i(k) \left\{ \sum_{j=1}^{2^m} \beta_j A_i + B_i \sum_{j=1}^{2^m} \beta_j (E_jK + E_j^- H) \right\} x(k) \\ &= \sum_{i=1}^{q} \alpha_i(k) \sum_{j=1}^{2^m} \beta_j \left\{ A_i + B_i(E_jK + E_j^- H) \right\} x(k) \\ &= \sum_{i=1}^{q} \sum_{j=1}^{2^m} \alpha_i(k) \beta_j \left\{ A_i + B_i(E_jK + E_j^- H) \right\} x(k) = A_c(k)x(k) \end{aligned}$$

where

$$A_c(k) = \sum_{i=1}^{q} \sum_{j=1}^{2^m} \alpha_i(k) \beta_j \left\{ A_i + B_i(E_jK + E_j^- H) \right\}$$

From the fact that

$$\sum_{i=1}^{q} \sum_{j=1}^{2^m} \alpha_i(k) \beta_j = \sum_{i=1}^{q} \alpha_i(k) \left\{ \sum_{j=1}^{2^m} \beta_j \right\} = 1$$

it is clear that $A_c(k)$ belongs to the polytope $P_c$, the vertices of which are given by taking all possible combinations of $A_i + B_i(E_jK + E_j^- H)$ where $i = 1, 2, \ldots, q$ and $j = 1, 2, \ldots, 2^m$.

The ellipsoid $E(P) = \{x \in \mathbb{R}^n : x^T P^{-1} x \leq 1\}$ is invariant, if and only if for all $x \in \mathbb{R}^n$ such that $x^T P^{-1} x \leq 1$ it holds that

$$x^T A_c(k)^T P^{-1} A_c(k) x \leq 1 \tag{2.52}$$

With the same argument as in Section 2.3.3, it is clear that condition (2.52) can be transformed to

$$\begin{bmatrix} P & A_c(k)P \\ PA_c(k)^T & P \end{bmatrix} \succeq 0 \tag{2.53}$$

The left-hand side of equation (2.52) can be treated as a function of $k$ and reaches the minimum on one of the vertices of $A_c(k)$, so the set of LMI conditions to be satisfied for invariance is the following

$$\begin{bmatrix} P & \{A_i + B_i(E_jK + E_j^- H)\}P \\ P\{A_i + B_i(E_jK + E_j^- H)\}^T & P \end{bmatrix} \succeq 0,$$

for all $i = 1,2,\dots,q$ and for all $j = 1,2,\dots,2^m$. □

Note that conditions (2.51) involve the multiplication between two unknown parameters $H$ and $P$. By denoting $Y = HP$, the LMI condition (2.51) can be rewritten as

$$\begin{bmatrix} P & (A_iP + B_iE_jKP + B_iE_j^- Y) \\ (PA_i^T + PK^TE_jB_i^T + Y^TE_j^- B_i^T) & P \end{bmatrix} \succeq 0, \tag{2.54}$$

for $\forall i = 1,2,\dots,q$, $\forall j = 1,2,\dots,2^m$. Thus the unknown matrices $P$ and $Y$ enter linearly in the conditions (2.54).

Again, as in Section 2.3.3, in general one would like to have the largest invariant ellipsoid for system (2.37) under the feedback $u(k) = sat(Kx(k))$ with respect to constraints (2.39). This can be achieved by solving the following LMI problem

$$J = \max_{P,Y}\{trace(P)\} \tag{2.55}$$

subject to

- Invariance condition

$$\begin{bmatrix} P & (A_iP + B_iE_jKP + B_iE_j^- Y) \\ (PA_i^T + PK^TE_jB_i^T + Y^TE_j^- B_i^T) & P \end{bmatrix} \succeq 0, \tag{2.56}$$

  for all $i = 1,2,\dots,q$ and for all $j = 1,2,\dots,2^m$
- Constraint satisfaction
  - On state
  $$\begin{bmatrix} 1 & F_iP \\ PF_i^T & P \end{bmatrix} \succeq 0, \quad \forall i = 1,2,\dots,n_1$$
  - On input
  $$\begin{bmatrix} u_{imax}^2 & Y_i \\ Y_i^T & P \end{bmatrix} \succeq 0, \quad \forall i = 1,2,\dots,m$$

  where $Y_i$ is the $i-th$ row of the matrix $Y$.

*Example 2.4.* Consider the following linear uncertain discrete-time system

$$x(k+1) = A(k)x(k) + B(k)u(k)$$

with

$$A(k) = \alpha(k)A_1 + (1-\alpha(k))A_2$$
$$B(k) = \alpha(k)B_1 + (1-\alpha(k))B_2$$

and

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}$$

At each sampling time $\alpha(k) \in [0,\ 1]$ is an uniformly distributed pseudo-random number. The constraints are

$$-10 \le x_1 \le 10, -10 \le x_2 \le 10, -1 \le u \le 1$$

The robustly stabilizing feedback matrix gain is chosen as

$$K = [-1.8112 \quad -0.8092]$$

By solving the optimization problem (2.55), the matrices $P$ and $Y$ are obtained

$$P = \begin{bmatrix} 5.0494 & -8.9640 \\ -8.9640 & 28.4285 \end{bmatrix}, \quad Y = [0.4365 \quad -4.2452]$$

Hence

$$H = YP^{-1} = [-0.4058 \quad -0.2773]$$

Based on the LMI problem (2.31), an invariant ellipsoid $E(P_1)$ is obtained under the linear feedback $u(k) = Kx(k)$ with

$$P_1 = \begin{bmatrix} 1.1490 & -3.1747 \\ -3.1747 & 9.9824 \end{bmatrix}$$

Figure 2.9 presents the invariant sets with different control laws. The set $E(P)$ is obtained with the saturated controller $u(k) = sat(Kx(k))$ while the set $E(P_1)$ is obtained with the linear controller $u(k) = Kx(k)$.

Figure 2.10 shows different state trajectories of the closed loop system with the controller $u(k) = sat(Kx(k))$ for different realizations of $\alpha(k)$ and different initial conditions.

In the first part of this subsection, Theorem 2.2 was exploited in the following manner: if the ellipsoid $E(P)$ is robustly invariant for the system

$$x(k+1) = A(k)x(k) + B(k)sat(Kx(k))$$

then there exists a stabilizing linear controller $u(k) = Hx(k)$, such that the ellipsoid $E(P)$ is robustly invariant with respect to the closed-loop system

$$x(k+1) = A(k)x(k) + B(k)Hx(k)$$

The matrix gain $H \in \mathbb{R}^{m\times n}$ is obtained by solving the optimization problem (2.55).

Theorem 2.2 now will be exploited in a different manner. We would like to design a saturated feedback gain $u(k) = sat(Kx(k))$ that makes a *given* invariant ellipsoid

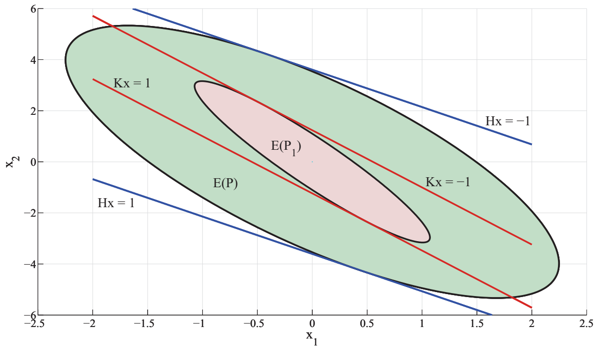



**Fig. 2.9** Invariant sets with different control laws for example 2.4. The set $E(P)$ is obtained with the saturated controller $u(k) = sat(Kx(k))$ while the set $E(P_1)$ is obtained with the linear controller $u(k) = Kx(k)$.

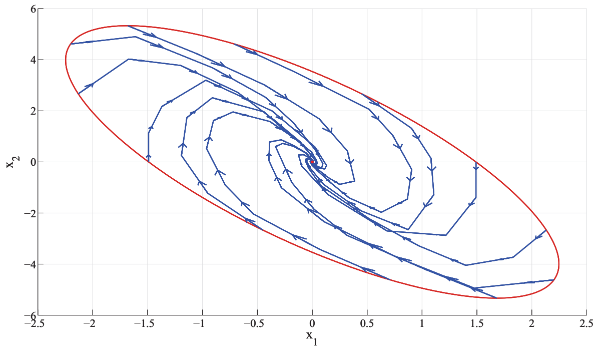



**Fig. 2.10** State trajectories of the closed loop system for example 2.4.

$E(P)$ contractive with a maximal contraction factor. This invariant ellipsoid $E(P)$ can be inherited for example together with a linear feedback gain $u(k) = Hx(k)$ from the optimization of some convex objective function $J(P)$[6], for example $trace(P)$. In the second stage, based on the gain $H$ and the ellipsoid $E(P)$, a saturated controller $u(k) = sat(Kx(k))$ which aims to maximize some contraction factor $1-g$ is computed.

It is worth noticing that the invariance condition (2.30) corresponds to the one in condition (2.54) with $E_j = 0$ and $E_j^- = I - E_j = I$. Following the proof of Theorem 2.2, it is clear that for the following system

[6] Practically, the design of the invariant ellipsoid $E(P)$ and the controller $u(k) = Hx(k)$ can be done by solving the LMI problem (2.30).

$$x(k+1) = A(k)x(k) + B(k)sat(Kx(k))$$

the ellipsoid $E(P)$ is contractive with the contraction factor $1-g$ if

$$\left\{A_i + B_i(E_jK + E_j^- H)\right\}^T P^{-1} \left\{A_i + B_i(E_jK + E_j^- H)\right\} - P^{-1} \preceq -gP^{-1}$$

for all $i = 1,2,\ldots,q$ and for all $j = 1,2,\ldots,2^m$ such that $E_j \neq 0$. By using the Schur complement, this problem can be converted into an LMI optimization as

$$J = \max_{g,K}\{g\} \tag{2.57}$$

subject to

$$\begin{bmatrix} (1-g)P^{-1} & (A_i + B_i(E_jK + E_j^- H))^T \\ (A_i + B_i(E_jK + E_j^- H)) & P \end{bmatrix} \succeq 0$$

for all $i = 1,2,\ldots,p$ and $j = 1,2,\ldots,2^m$ with $E_j \neq 0$. Recall that here the only unknown parameters are the matrix $K \in \mathbb{R}^{m\times n}$ and the scalar $g$, the matrices $P$ and $H$ being given in the first stage.

*Remark 2.4.* The proposed two-stage control design presented here benefits from global uniqueness properties of the solution. This is due to the one-way dependence of the two (prioritized) objectives: the trace maximization precedes the associated contraction factor.

*Example 2.5.* Consider the uncertain system in example 2.4 with the same constraints on the state vector and on the input vector. In the first stage by solving the optimization problem (2.30), one obtains the matrices $P$ and $Y$

$$P = \begin{bmatrix} 100.0000 & -43.1051 \\ -43.1051 & 100.0000 \end{bmatrix}, \quad Y = [-3.5691 \quad -6.5121]$$

Hence $H = YP^{-1} = [-0.0783 \quad -0.0989]$.

In the second stage, by solving the optimization problem (2.57), one obtains the feedback gain $K$

$$K = [-0.3342 - 0.7629]$$

Figure 2.11 shows the invariant ellipsoid $E(P)$. This figure also shows the state trajectories of the closed loop system under the saturated feedback $u(k) = sat(Kx(k))$ for different initial conditions and different realizations of $\alpha(k)$.

For the initial condition $x(0) = [-4 \quad 10]^T$ Figure 2.12(a) presents the state trajectory of the closed loop system with the saturated controller $u(k) = sat(Kx(k))$ and with the linear controller $u(k) = Hx(k)$. It can be observed that the time to regulate the plant to the origin by using the linear controller is longer than the time to regulate the plant to the origin by using the saturated controller. The explanation for this is that when using the controller $u(k) = Hx(k)$, the control action is saturated only at some points of the boundary of the ellipsoid $E(P)$, while using the controller

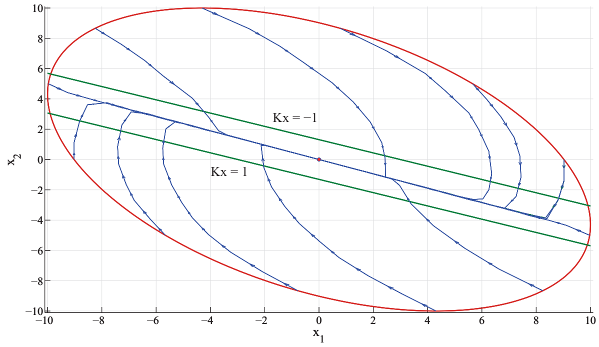
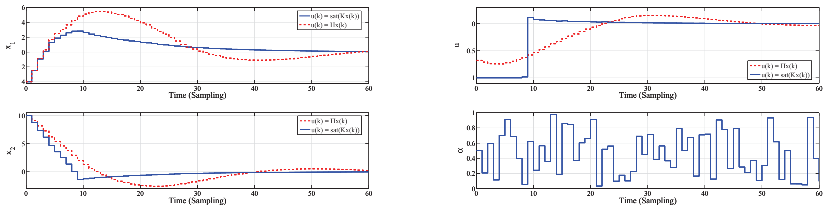



**Fig. 2.11** Invariant ellipsoid and state trajectories of the closed loop system for example 2.5.

$u(k) = sat(Kx(k))$, the control action is saturated not only on the boundary of the set $E(P)$, the saturation being active also inside the set $E(P)$. This effect can be observed in Figure 2.12(b). The same figure presents the realization of $\alpha(k)$.




(a) State trajectory (b) Input trajectory and $\alpha$ realization

**Fig. 2.12** State and input trajectory of the closed loop system as a function of time for example 2.5. The solid blue lines are obtained by using the saturated feedback gain $u(k) = sat(Kx(k))$, and the dashed red lines are obtained by using the linear feedback controller $u(k) = Hx(k)$ in the figures for $x_1, x_2$ and $u$.

### 2.4.4 The polyhedral set approach

In this section, the problem of estimating the domain of attraction is addressed by using polyhedral sets. For a given linear state feedback controller $u(k) = Kx(k)$, it is clear that the largest polyhedral invariant set is the maximal robustly invariant set $\Omega_{max}$. The set $\Omega_{max}$ can be readily found using procedure 2.1 or procedure 2.2. From this point on, it is assumed that the set $\Omega_{max}$ is known.

Our aim in this subsection is to find the the largest polyhedral invariant set characterizing an estimation of the domain of attraction for system (2.37) under the saturated controller $u(k) = sat(Kx(k))$. To this aim, recall that from Lemma (2.1), the saturation function can be expressed as

$$sat(Kx) = \sum_{i=1}^{2^m} \beta_i (E_i Kx + E_i^- v), \quad \sum_{i=1}^{2^m} \beta_i = 1, \quad \beta_i \geq 0 \tag{2.58}$$

with $u_l \leq v \leq u_u$ and $E_i$ is an element of $D_m$[7] and $E_i^- = I - E_i$.

With equation (2.59) the closed loop system can be rewritten as

$$\begin{aligned} x(k+1) &= \sum_{i=1}^{q} \alpha_i(k) \left\{ A_i x(k) + B_i \sum_{j=1}^{2^m} \beta_j (E_j Kx(k) + E_j^- v) \right\} \\ &= \sum_{i=1}^{q} \alpha_i(k) \left\{ \sum_{j=1}^{2^m} \beta_j A_i x(k) + B_i \sum_{j=1}^{2^m} \beta_j (E_j Kx(k) + E_j^- v) \right\} \\ &= \sum_{i=1}^{q} \alpha_i(k) \sum_{j=1}^{2^m} \beta_j \left\{ A_i x(k) + B_i (E_j Kx(k) + E_j^- v) \right\} \end{aligned}$$

or

$$x(k+1) = \sum_{j=1}^{2^m} \beta_j \sum_{i=1}^{q} \alpha_i(k) \left\{ (A_i + B_i E_j K) x(k) + B_i E_j^- v \right\} \tag{2.59}$$

The variables $v \in \mathbb{R}^m$ can be considered as an external controlled input for system (2.59). Hence, the problem of finding the largest polyhedral invariant set $\Omega_s$ for system (2.41) boils down to the problem of computing the largest controlled invariant set for system (2.59).

System (2.59) can be considered as an uncertain system with respect to the parameters $\alpha_i$ and $\beta_j$. Hence the following procedure can be used to obtain the largest polyhedral invariant set $\Omega_s$ for system (2.59) based on the results in Section 2.3.4

---

[7] The set of $m \times m$ diagonal matrices whose diagonal elements are either 0 or 1

**Procedure 2.4: Invariant set computation**

---

- **Input:** The matrices $A_1,\ldots,A_q$, $B_1,\ldots,B_q$, the gain $K$ and the sets $X$, $U$ and the invariant set $\Omega_{max}$
- **Output:** An invariant approximation of the invariant set $\Omega_s$ for the closed loop system (2.41).

1. Set $i=0$ and $C_0=\Omega_{max}$ and let the matrices $F_0$, $g_0$ be the half space representation of the set $C_0$, i.e. $C_0=\{x\in\mathbb{R}^n : F_0x\leq g_0\}$
2. Compute the expanded set $P_{ij}\subset\mathbb{R}^{n+m}$ for all $j=1,2,\ldots,2^m$

$$P_{ij}=\left\{(x,v)\in\mathbb{R}^{n+m}:\begin{bmatrix}F_i\{(A_1+B_1E_jK)x+B_1E_j^-v\}\\ F_i\{(A_2+B_2E_jK)x+B_2E_j^-v\}\\ \vdots\\ F_i\{(A_q+B_qE_jK)x+B_qE_j^-v\}\end{bmatrix}\leq\begin{bmatrix}g_i\\ g_i\\ \vdots\\ g_i\end{bmatrix}\right\}$$

3. Compute the projection of $P_{ij}$ on $\mathbb{R}^n$

$$P_{ij}^n=\{x\in\mathbb{R}^n:\exists v\in U \text{ such that } (x,v)\in P_{ij}\},\forall j=1,2,\ldots,2^m$$

4. Set

$$C_{i+1}=X\bigcap_{j=1}^{2^m}P_{ij}^n$$

and let the matrices $F_{i+1}$, $g_{i+1}$ be the half space representation of the set $C_{i+1}$, i.e.

$$C_{i+1}=\{x\in\mathbb{R}^n : F_{i+1}x\leq g_{i+1}\}$$

5. If $C_{i+1}=C_i$, then stop and set $\Omega_s=C_i$. Else continue.
6. Set $i=i+1$ and go to step 2.

It is clear that $C_{i-1}\subseteq C_i$, since the set $\Omega_{max}$ is robustly invariant. Hence $C_i$ is a robustly invariant set. The set sequence $\{C_0,C_1,\ldots,\}$ converges to $\Omega_s$, which is the largest polyhedral invariant set.

*Remark 2.5.* Each one of the polytopes $C_i$ represents an invariant inner approximation of the domain of attraction for the system (2.37) under the saturated controller $u(k)=sat(Kx(k))$. That means the procedure 2.4 can be stopped at any time before converging to the true largest invariant set $\Omega_s$ and obtain a robustly invariant approximation of the set $\Omega_s$.

It is worth noticing that the matrix $H\in\mathbb{R}^{m\times n}$ resulting from optimization problem (2.55) can also be employed for computing the polyhedral invariant set $\Omega_s^H$ with respect to the saturated controller $u(k)=sat(Kx(k))$. Clearly the set $\Omega_s^H$ is a subset of $\Omega_s$, since the vector $v$ is now in the restricted form $v(k)=Hx(k)$, but this can be an important instrument design tool. In this case, from the equation (2.59) one gets

$$x(k+1) = \sum_{j=1}^{2^m} \beta_j \sum_{i=1}^{q} \alpha_i(k) \left\{ (A_i + B_i E_j K + B_i E_j^- H) x(k) \right\} \tag{2.60}$$

Define the set $X_H$ as follows

$$X_H = \{x \in \mathbb{R}^n : F_H x \leq g_H\} \tag{2.61}$$

where

$$F_H = \begin{bmatrix} F_x \\ H \\ -H \end{bmatrix}, \quad g_H = \begin{bmatrix} g_x \\ u_u \\ u_l \end{bmatrix}$$

With the set $X_H$, the following procedure can be used for computing the polyhedral invariant set $\Omega_s^H$.

**Procedure 2.5: Invariant set computation**

- **Input:** The matrices $A_1, A_2, \ldots, A_q$ and the set $X_H$ and the invariant set $\Omega_{max}$
- **Output:** The invariant set $\Omega_s^H$

1. Set $i = 0$ and $C_0 = \Omega_{max}$ and let the matrices $F_0$, $g_0$ be the half space representation of the set $C_0$, i.e. $C_0 = \{x \in \mathbb{R}^n : F_0 x \leq g_0\}$
2. Compute the set $P_{ij} \subset \mathbb{R}^{n+m}$

$$P_{ij} = \left\{ x \in \mathbb{R}^n : \begin{bmatrix} F_i(A_1 + B_1 E_j K + B_1 E_j^- H)x \\ F_i(A_2 + B_2 E_j K + B_2 E_j^- H)x \\ \vdots \\ F_i(A_q + B_q E_j K + +B_q E_j^- H)x \end{bmatrix} \leq \begin{bmatrix} g_i \\ g_i \\ \vdots \\ g_i \end{bmatrix} \right\}$$

3. Set

$$C_{i+1} = X_H \bigcap_{j=1}^{2^m} P_{ij}^n$$

and let the matrices $F_{i+1}$, $g_{i+1}$ be the half space representation of the set $C_{i+1}$, i.e.

$$C_{i+1} = \{x \in \mathbb{R}^n : F_{i+1} x \leq g_{i+1}\}$$

4. If $C_{i+1} = C_i$, then stop and set $\Omega_s = C_i$. Else continue.
5. Set $i = i + 1$ and go to step 2.

Since the matrix $\sum_{j=1}^{2^m} \beta_j \sum_{i=1}^{q} \alpha_i(k) \left\{ (A_i + B_i E_j K + B_i E_j^- H) \right\}$ has a sub-unitary joint spectral radius, procedure 2.5 terminates in finite time [27]. In other words, there exists a finite index $i = i_{max}$ such that $C_{i_{max}} = C_{i_{max}+1}$.

*Example 2.6.* Consider again the example 2.4. The constraint on the state vector and on the input vector are the same. The controller is $K = [-1.8112 \quad -0.8092]$.

By using procedure 2.4 one obtains the robust polyhedral invariant set $\Omega_s$ as depicted in Figure 2.13. Procedure 2.4 terminated with $i = 121$. Figure 2.13 also shows the robust polyhedral invariant set $\Omega_s^H$ obtained with the auxiliary matrix $H$ where $H = [-0.4058 \quad -0.2773]$ and the robust polyhedral invariant set $\Omega_{max}$ obtained with the controller $u(k) = Kx$.

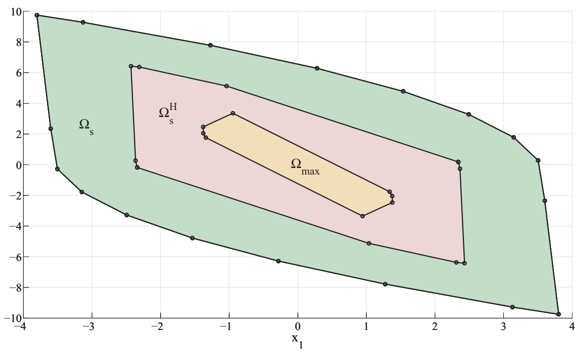



**Fig. 2.13** Robustly invariant sets with different control laws and different methods for example 2.6. The polyhedral set $\Omega_s$ is obtained with respect to the controller $u(k) = sat(Kx(k))$. The polyhedral set $\Omega_s^H$ is obtained with respect to the controller $u(k) = sat(Kx(k))$ using an auxiliary matrix $H$. The polyhedral set $\Omega_{max}$ is obtained with the controller $u(k) = Kx$.

The set $\Omega_s^H$ and $\Omega_s$ are presented in minimal normalized half-space representation as

$$\Omega_s^H = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.8256 & -0.5642 \\ 0.8256 & 0.5642 \\ 0.9999 & 0.0108 \\ -0.9999 & -0.0108 \\ 0.9986 & 0.0532 \\ -0.9986 & -0.0532 \\ -0.6981 & -0.7160 \\ 0.6981 & 0.7160 \\ 0.9791 & 0.2033 \\ -0.9791 & -0.2033 \\ -0.4254 & -0.9050 \\ 0.4254 & 0.9050 \end{bmatrix} x \le \begin{bmatrix} 2.0346 \\ 2.0346 \\ 2.3612 \\ 2.3612 \\ 2.3467 \\ 2.3467 \\ 2.9453 \\ 2.9453 \\ 2.3273 \\ 2.3273 \\ 4.7785 \\ 4.7785 \end{bmatrix} \right\}$$

$$\Omega_s = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.9996 & -0.0273 \\ 0.9996 & 0.0273 \\ -0.9993 & -0.0369 \\ 0.9993 & 0.0369 \\ -0.9731 & -0.2305 \\ 0.9731 & 0.2305 \\ 0.9164 & 0.4004 \\ -0.9164 & -0.4004 \\ 0.8434 & 0.5372 \\ -0.8434 & -0.5372 \\ 0.7669 & 0.6418 \\ -0.7669 & -0.6418 \\ 0.6942 & 0.7198 \\ -0.6942 & -0.7198 \\ 0.6287 & 0.7776 \\ -0.6287 & -0.7776 \\ 0.5712 & 0.8208 \\ -0.5712 & -0.8208 \end{bmatrix} x \leq \begin{bmatrix} 3.5340 \\ 3.5340 \\ 3.5104 \\ 3.5104 \\ 3.4720 \\ 3.4720 \\ 3.5953 \\ 3.5953 \\ 3.8621 \\ 3.8621 \\ 4.2441 \\ 4.2441 \\ 4.7132 \\ 4.7132 \\ 5.2465 \\ 5.2465 \\ 5.8267 \\ 5.8267 \end{bmatrix} \right\}$$

Figure 2.14 presents state trajectories of the closed loop system with the controller $u(k) = sat(Kx(k))$ for different initial conditions and different realizations of $\alpha(k)$.

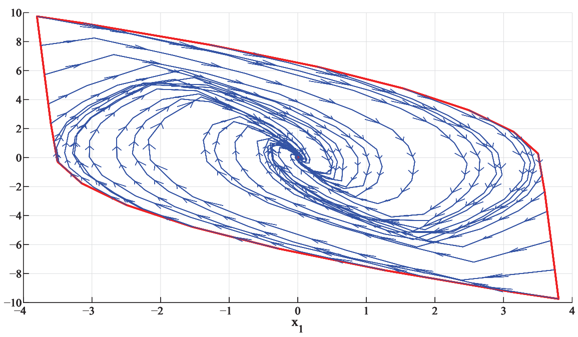



**Fig. 2.14** State trajectories of the closed loop system with the controller $u(k) = sat(Kx(k))$ for example 2.6.

# Chapter 3
# Optimal and Constrained Control - An Overview

In this chapter some of the approaches to constrained and optimal control are briefly reviewed. This review is not intended to be exhaustive but to provide an insight into the existing theoretical background on which, the present manuscript builds on. The chapter includes the following sections

1. Dynamic programming.
2. Pontryagin's maximum principle.
3. Model predictive control: implicit and explicit solutions.
4. Vertex control.

## 3.1 Dynamic programming

The purpose of this section is to present a brief introduction to *dynamic programming*, which provides a sufficient condition for optimality.

Dynamic programming was developed by R.E. Bellman in the early fifties [13], [14], [15], [16]. It provides insight into properties of the control problems for various classes of systems, e.g. linear, time-varying or nonlinear. In general the optimal solution is found in open loop form, without feedback.

Dynamic programming is based on the following principle of optimality [17]:

*An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*

To begin, let us consider the following optimal control problem

$$\min_{x,u} \left\{ \sum_{k=0}^{N-1} L(x(k),u(k)) + E(x(N)) \right\} \tag{3.1}$$

subject to

$$\begin{cases} x(k+1) = f(x(k), u(k)), \ k = 0, 1, \ldots, N-1 \\ u(k) \in U, \ k = 0, 1, \ldots, N-1 \\ x(k) \in X, \ k = 0, 1, \ldots, N \\ x(0) = x_0 \end{cases}$$

where

- $x(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}^m$ are respectively the state and control variables.
- $N > 0$ is called *the time horizon*.
- $L(x(k), u(k))$ is the Lagrange objective function, which represents a cost along the trajectory.
- $E(x(N))$ is the Mayer objective function, which represents the terminal cost.
- $U$ and $X$ are the sets of constraints on the input and state variables, respectively.
- $x(0)$ is the initial condition.

Define the *value function* $V_i(x(i))$ as follows

$$V_i(x(i)) = \min_{x,u} \left\{ E(x(N)) + \sum_{k=i}^{N-1} L(x(k), u(k)) \right\} \tag{3.2}$$

subject to

$$\begin{cases} x(k+1) = f(x(k), u(k)), \ k = 0, 1, \ldots, N-1 \\ u(k) \in U, \ k = i, i+1, \ldots, N-1 \\ x(k) \in X, \ k = i, i+1, \ldots, N \end{cases}$$

for $i = N, N-1, N-2, \ldots, 0$.

Clear $V_i(x(i))$ is the optimal cost on the remaining horizon $[i, N]$, starting from the state $x(i)$. Based on the principle of optimality, one has

$$V_i(x(i)) = \min_{u(i)} \{ L(x(i), u(i)) + V_{i+1}(x(i+1)) \}$$

By substituting

$$x(i+1) = f(x(i), u(i))$$

one gets

$$V_i(z) = \min_{u(i)} \{ L(x(i), u(i)) + V_{i+1}(f(x(i), u(i))) \} \tag{3.3}$$

subject to

$$\begin{cases} u(i) \in U, \\ f(x(i), u(i)) \in X \end{cases}$$

The problem (3.3) is much simpler than the one in (3.1) because it involves only one decision variable $u(i)$. To actually solve this problem, we work backwards in time from $i = N$, starting with

$$V_N(x(N)) = E(x(N))$$

Based on the value function $V_{i+1}(x(i+1))$ with $i = N-1, N-2, \ldots, 0$, the optimal control values $u^*(i)$ can be obtained as

$$u^*(i) = \arg\min_{u(i)} \{L(x(i),u(i)) + V_{i+1}(f(x(i),u(i)))\}$$

subject to

$$\begin{cases} u(i) \in U, \\ f(x(i),u(i)) \in X \end{cases}$$

## 3.2 Pontryagin's maximum principle

The second milestone in the optimal control theory is the Pontryagin's maximum principle [125], [34], offering a basic mathematical technique for calculating the optimal control values in many important problems of mathematics, engineering, economics, e.t.c. This approach, can be seen as a counterpart of the classical calculus of variation approach, allowing us to solve the control problems in which the control input is subject to constraints in a very general way. Here for illustration, we consider the following simple optimal control problem

$$\min_{x,u} \left\{ \sum_{k=0}^{N-1} L(x(k),u(k)) + E(x(N)) \right\} \tag{3.4}$$

subject to

$$\begin{cases} x(k+1) = f(x(k),u(k)),\ k = 0,1,\ldots,N-1 \\ u(k) \in U,\ k = 0,1,\ldots,N-1 \\ x(0) = x_0 \end{cases}$$

For simplicity, the state variables are considered unconstrained. For solving the optimal control problem (3.4) with the Pontryagin's maximum principle, the following Hamiltonian $H_k(\cdot)$ is defined

$$H_k(x(k),u(k),\lambda(k+1)) = L(x(k),u(k)) + \lambda^T(k+1)f(x(k),u(k)) \tag{3.5}$$

where $\lambda(k) \in \mathbb{R}^n$ with $k = 1,2,\ldots,N$ are called the *co-state* or the *adjoint* variables. For problem (3.4), these variables must satisfy the so called co-state equation

$$\lambda^*(k+1) = \frac{\partial H_k}{\partial(x(k))},\ k = 0,1,\ldots,N-2$$

and

$$\lambda^*(N) = \frac{\partial E(x(N))}{\partial(x(N))}$$

For given state and co-state variables, the optimal control value is achieved by choosing control $u^*(k)$ that minimizes the Hamiltonian at each time instant, i.e.

$$H_k(x^*(k),u^*(k),\lambda^*(k+1)) \le H_k(x^*(k),u(k),\lambda^*(k+1)),\ \forall u(k) \in U$$

Note that a convexity assumption on the Hamiltonian is needed, i.e. the function $H_k(x(k),u(k),\lambda(k+1))$ is supposed to be convex with respect to $u(k)$.

## 3.3 Model predictive control

Model predictive control (MPC), or receding horizon control, is one of the most advanced control approaches which, in the last decades, has became a leading industrial control technology for constrained control systems [32], [100], [28], [129], [47], [96], [53]. MPC is an optimization based strategy, where a model of the plant is used to predict the future evolution of the system, see [100], [96]. This prediction uses the current state of the plant as the initial state and, at each time instant, $k$, the controller computes a finite optimal control sequence. Then the first control action in this sequence is applied to the plant at time instant $k$, and at time instant $k+1$ the optimization procedure is repeated with a new plant measurement. This open loop optimal feedback mechanism[1] of the MPC compensates for the prediction error due to structural mismatch between the model and the real system as well as for disturbances and measurement noise. In contrast to the maximal principle or dynamic programming solutions which are open loop optimal, the receding horizon principle behind MPC brings the advantage of the feedback structure.

But again the main advantage which makes MPC industrially desirable is that it can take into account constraints in the control problem. This feature is very important for several reasons

- Often the best performance, which may correspond to the most efficient operation, is obtained when the system is made to operate near the constraints.
- The possibility to explicitly express constraints in the problem formulation offers a natural way to state complex control objectives.
- Stability and other features can be proved, at least in some cases, in contrast to popular ad-hoc methods to handle constraints, like anti-windup control [51], and override control [143].

### *3.3.1 Implicit model predictive control*

Consider the problem of regulating to the origin the following discrete-time linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k) \tag{3.6}$$

where $x(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}^m$ are respectively the state and the input variables, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the system matrices. Both the state vector $x(k)$ and the control vector $u(k)$ are subject to polytopic constraints

[1] It was named OLOF (Open Loop Optimal Feedback) control, by the author of [38]

$$\begin{cases} x(k) \in X, X = \{x : F_x x \le g_x\} \\ u(k) \in U, U = \{u : F_u u \le g_u\} \end{cases} \forall k \ge 0 \tag{3.7}$$

where the matrices $F_x$, $F_u$ and the vectors $g_x$, $g_u$ are assumed to be constant with $g_x > 0$, $g_u > 0$ such that the origin is contained in the interior of $X$ and $U$. Here the inequalities are taken element-wise. It is assumed that the pair $(A, B)$ is stabilizable, i.e. all uncontrollable states have stable dynamics.

Provided that the state $x(k)$ is available from the measurements, the basic finite horizon MPC optimization problem is defined as

$$V(x(k)) = \min_{\mathbf{u}=[u_0,u_1,\ldots,u_{N-1}]} \left\{ \sum_{t=1}^{N} x_t^T Q x_t + \sum_{t=0}^{N-1} u_t^T R u_t \right\} \tag{3.8}$$

subject to

$$\begin{cases} x_{t+1} = Ax_t + Bu_t,\ t = 0,1,\ldots,N-1 \\ x_t \in X,\ t = 1,2,\ldots,N \\ u_t \in U,\ t = 0,1,\ldots,N-1 \\ x_t = x(k) \end{cases}$$

where

- $x_{t+1}$ and $u_t$ are, respectively the predicted states and the predicted inputs, $t = 0,1,\ldots,N-1$.
- $Q \in \mathbb{R}^{n\times n}$ is a real symmetric positive semi-definite matrix.
- $R \in \mathbb{R}^{m\times m}$ is a real symmetric positive definite matrix.
- $N$ is a fixed integer greater than 0. N is called *the time horizon* or *the prediction horizon*.

The conditions on $Q$ and $R$ guarantee that the function $J$ is convex. In term of eigenvalues, the eigenvalues of $Q$ should be non-negative, while those of $R$ should be positive in order to ensure a unique optimal solution.

From the control objective point of view, it is clear that the first term $x_t^T Q x_t$ penalizes the deviation of the state $x$ from the origin, while the second term $u_t^T R u_t$ measures the input control energy. In other words, selecting $Q$ large means that, to keep $V$ small, the state $x_t$ must be as close as possible to the origin in a weighted Euclidean norm. On the other hand, selecting $R$ large means that the control input $u_t$ must be small to keep the cost function $V$ small.

An alternative is a performance measure based on $l_1-$norm

$$\min_{\mathbf{u}=[u_0,u_1,\ldots,u_{N-1}]} \left\{ \sum_{t=1}^{N} |Qx_t|_1 + \sum_{t=0}^{N-1} |Ru_t|_1 \right\} \tag{3.9}$$

or $l_\infty-$norm

$$\min_{\mathbf{u}=[u_0,u_1,\ldots,u_{N-1}]} \left\{ \sum_{t=1}^{N} |Qx_t|_\infty + \sum_{t=0}^{N-1} |Ru_t|_\infty \right\} \tag{3.10}$$

Based on the state space model (3.6), the future state variables are expressed sequentially using the set of future control variable values

$$\begin{cases} x_1 = Ax_0 + Bu_0 \\ x_2 = Ax_1 + Bu_1 = A^2x_0 + ABu_0 + Bu_1 \\ \vdots \\ x_N = A^N x_0 + A^{N-1}Bu_0 + A^{N-2}Bu_1 + \ldots + Bu_{N-1} \end{cases} \tag{3.11}$$

The set of equations (3.11) can be rewritten in a compact matrix form as

$$\mathbf{x} = A_a x_0 + B_a \mathbf{u} = A_a x(k) + B_a \mathbf{u} \tag{3.12}$$

with

$$\mathbf{x} = [x_1^T \; x_2^T \; \ldots \; x_N^T]^T$$
$$\mathbf{u} = [u_0^T \; u_1^T \; \ldots \; u_{N-1}^T]^T$$

and

$$A_a = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad B_a = \begin{bmatrix} B & 0 & \ldots & 0 \\ AB & B & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ A^{N-1}B & A^{N-2}B & \ldots & B \end{bmatrix}$$

The MPC optimization problem (3.8) can be expressed as

$$V(x(k)) = \min_{\mathbf{u}} \{\mathbf{x}^T Q_a \mathbf{x} + \mathbf{u}^T R_a \mathbf{u}\} \tag{3.13}$$

where

$$Q_a = \begin{bmatrix} Q & 0 & \ldots & 0 \\ 0 & Q & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & Q \end{bmatrix}, \quad R_a = \begin{bmatrix} R & 0 & \ldots & 0 \\ 0 & R & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & R \end{bmatrix}$$

and by substituting (3.12) in (3.13), one gets

$$V(x(k)) = \min_{\mathbf{u}} \{\mathbf{u}^T H \mathbf{u} + 2x^T(k) F \mathbf{u} + x^T(k) Y x(k)\} \tag{3.14}$$

where

$$H = B_a^T Q_a B_a + R_a, \quad F = A_a^T Q_a B_a \text{ and } Y = A_a^T Q_a A_a \tag{3.15}$$

Consider now the constraints on state and on input along the horizon. From (3.7) it can be shown that

$$\begin{cases} F_x^a \mathbf{x} \le g_x^a \\ F_u^a \mathbf{u} \le g_u^a \end{cases} \tag{3.16}$$

where

$$F_x^a = \begin{bmatrix} F_x & 0 & \dots & 0 \\ 0 & F_x & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & F_x \end{bmatrix}, \quad g_x^a = \begin{bmatrix} g_x \\ g_x \\ \vdots \\ g_x \end{bmatrix}$$

$$F_u^a = \begin{bmatrix} F_u & 0 & \dots & 0 \\ 0 & F_u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & F_u \end{bmatrix}, \quad g_u^a = \begin{bmatrix} g_u \\ g_u \\ \vdots \\ g_u \end{bmatrix}$$

Using (3.12), the state constraints along the horizon can be expressed as

$$F_x^a\{A_a x(k) + B_a \mathbf{u}\} \leq g_x^a$$

or

$$F_x^a B_a \mathbf{u} \leq -F_x^a A_a x(k) + g_x^a \tag{3.17}$$

Combining (3.16), (3.17), one obtains

$$G\mathbf{u} \leq Ex(k) + S \tag{3.18}$$

where

$$G = \begin{bmatrix} F_u^a \\ F_x^a B_a \end{bmatrix}, \quad E = \begin{bmatrix} 0 \\ -F_x^a A_a \end{bmatrix}, \quad S = \begin{bmatrix} g_u^a \\ g_x^a \end{bmatrix}$$

Based on (3.13) and (3.18), the MPC quadratic program formulation can be defined as

$$V_1(x(k)) = \min_{\mathbf{u}} \left\{ \mathbf{u}^T H \mathbf{u} + 2x^T(k) F \mathbf{u} \right\} \tag{3.19}$$

subject to

$$G\mathbf{u} \leq Ex(k) + W$$

where the term $x^T(k)Yx(k)$ is removed since it does not influence the optimal argument. The value of the cost function at optimum is simply obtained from (3.19) by

$$V(x(k)) = V_1(x(k)) + x^T(k)Yx(k)$$

A simple on-line algorithm for MPC is

**Algorithm 3.1.** Model predictive control - Implicit approach

---

1. Measure the current state of the system $x(k)$.
2. Compute the control signal sequence $\mathbf{u}$ by solving (3.19).
3. Apply first element of the control sequence $\mathbf{u}$ as input to the system (3.6).
4. Wait for the next time instant $k := k + 1$.
5. Go to step 1 and repeat

*Example 3.1.* Consider the following discrete time linear time invariant system

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.7 \end{bmatrix} u(k) \tag{3.20}$$

and the MPC problem with weighting matrices $Q = I$ and $R = 1$ and the prediction horizon $N = 3$.

The constraints are

$$-2 \le x_1 \le 2,\ -5 \le x_2 \le 5,\ -1 \le u \le 1$$

Based on equation (3.15) and (3.18), the MPC problem can be described as a QP problem

$$\min_{\mathbf{u}=\{u_0,u_1,u_2\}} \left\{ \mathbf{u}^T H \mathbf{u} + 2x^T(k) F \mathbf{u} \right\}$$

with

$$H = \begin{bmatrix} 12.1200 & 6.7600 & 2.8900 \\ 6.7600 & 5.8700 & 2.1900 \\ 2.8900 & 2.1900 & 2.4900 \end{bmatrix}, \quad F = \begin{bmatrix} 5.1000 & 2.7000 & 1.0000 \\ 13.7000 & 8.5000 & 3.7000 \end{bmatrix}$$

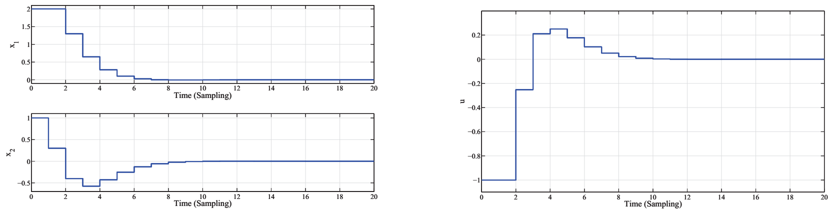
and subject to the following constraints

$$G\mathbf{u} \le S + Ex(k)$$

where

$$G = \begin{bmatrix} 1.0000 & 0 & 0 \\ -1.0000 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & -1.0000 & 0 \\ 0 & 0 & 1.0000 \\ 0 & 0 & -1.0000 \\ 1.0000 & 0 & 0 \\ 0.7000 & 0 & 0 \\ -1.0000 & 0 & 0 \\ -0.7000 & 0 & 0 \\ 1.7000 & 1.0000 & 0 \\ 0.7000 & 0.7000 & 0 \\ -1.7000 & -1.0000 & 0 \\ -0.7000 & -0.7000 & 0 \\ 2.4000 & 1.7000 & 1.0000 \\ 0.7000 & 0.7000 & 0.7000 \\ -2.4000 & -1.7000 & -1.0000 \\ -0.7000 & -0.7000 & -0.7000 \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ 0 & -1 \\ 1 & 1 \\ 0 & 1 \\ -1 & -2 \\ 0 & -1 \\ 1 & 2 \\ 0 & 1 \\ -1 & -3 \\ 0 & -1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 5 \\ 2 \\ 5 \\ 2 \\ 5 \\ 2 \\ 5 \\ 2 \\ 5 \\ 2 \\ 5 \end{bmatrix}$$

For the initial condition $x(0) = [2 \quad 1]^T$ and by using the implicit MPC method, Figure 3.1 shows the state and input trajectory of the closed loop system as a function of time.




(a) State trajectory (b) Input trajectory.

**Fig. 3.1** State and input trajectory of the closed loop system as a function of time for example 3.1.

## *3.3.2 Recursive feasibility and stability*

Recursive feasibility of the optimization problem and stability of the resulting closed-loop system are two important aspects when designing a MPC controller.

Recursive feasibility of the optimization problem (3.19) means that if the problem (3.19) is feasible at time instant $k$, it will be also feasible at time instant $k+1$. In other words there exists an admissible control value that holds the system within the state constraints. The feasibility problem can arise due to model errors, disturbances or the choice of the cost function.

Stability analysis necessitates the use of Lyapunov theory [73], since the presence of the constraints makes the closed-loop system nonlinear. In addition, it is well known that unstable input-constrained system cannot be globally stabilized [144], [97], [138]. Another problem is that the control law is generated by the solution of the optimization problem (3.19) and generally there does not exist any simple[2] closed-form expression for the solution, although it can be shown that the solution is a piecewise affine state feedback law [20].

Recursive feasibility and stability can be assured by adding a terminal cost function in the objective function (3.8) and by including the final state of the planning horizon in a terminal positively invariant set. Let the matrix $P \in \mathbb{R}^{n \times n}$ be the unique solution of the following discrete-time algebraic Riccati equation

$$P = A^T PA - A^T PB(B^T XB + R)^{-1} B^T PA + Q \tag{3.21}$$

[2] Simple here is understood in terms of linear feedback gains as it is the case for the optimal control for unconstrained linear quadratic regulators [68].

and the matrix gain $K \in \mathbb{R}^{m\times n}$ is defined as

$$K = -(B^T PB+R)^{-1}B^T PA \tag{3.22}$$

It is well known [8], [86], [89], [90] that matrix gain $K$ is a solution of the optimization problem (3.8) when the time horizon $N = \infty$ and there are no constraints on the state vector and on the input vector. In this case the cost function is

$$\begin{aligned} V(x(0)) &= \sum_{k=0}^{\infty} \left\{x_k^T Qx_k + u_k^T Ru_k\right\} \\ &= \sum_{k=0}^{\infty} x_k^T \left(Q+K^T RK\right) x_k = x_0^T Px_0 \end{aligned}$$

Once the stabilizing feedback gain $u(k) = Kx(k)$ is defined, the terminal set $\Omega \subseteq X$ can be computed as a maximal invariant set associated with the control law $u(k) = Kx(k)$ for system (3.6) and with respect to the constraints (3.7). Generally, the terminal invariant set $\Omega$ is chosen to be in the ellipsoidal or polyhedral form[3].

Consider now the following MPC optimization problem

$$\min_{\mathbf{u}=[u_0,u_1,\ldots,u_{N-1}]} \left\{ x_N^T Px_N + \sum_{t=0}^{N-1} \left\{x_t^T Qx_t + u_t^T Ru_t\right\} \right\} \tag{3.23}$$

subject to

$$\begin{cases} x_{t+1} = Ax_t + Bu_t,\ t = 0,1,\ldots,N-1 \\ x_t \in X,\ t = 1,2,\ldots,N \\ u_t \in U,\ t = 0,1,\ldots,N-1 \\ x_N \in \Omega \\ x_0 = x(k) \end{cases}$$

then the following theorem holds [100]

**Theorem 3.1.** *[100] Assuming feasibility at the initial state, the MPC controller (3.23) guarantees recursive feasibility and asymptotic stability.*

*Proof.* See [100]. □

The MPC problem considered here uses both a terminal cost function and a terminal set constraint and is called the dual-mode MPC. This MPC scheme is the most attractive version in the MPC literature. In general, it offers better performance compared with other MPC versions and allows a wider range of control problems to be handled. The downside is the dependence of the feasible domain on the prediction horizon. Generally, for a large domain one needs to employ a large prediction horizon.

---

[3] see Section 2.3.3 and Section 2.3.4.

### 3.3.3 Explicit model predictive control - Parameterized vertices

Note that the implicit model predictive control requires running on-line optimization algorithms to solve a quadratic programming (QP) problem associated with the objective function (3.8) or to solve a linear programming (LP) problem with the objective function (3.9), (3.10). Although computational speed and optimization algorithms are continuously improving, solving a QP or LP problem can be computationally costly, specially when the prediction horizon is large, and this has traditionally limited MPC to applications with relatively low complexity/sampling interval ratio.

Indeed the state vector can be interpreted as a vector of parameters in the optimization problem (3.23). The exact optimal solution can be expressed as *a piecewise affine function* of the state over a polyhedral partition of the state space and the MPC control computation can be moved off-line [20], [28], [141], [117]. The control action is then computed on-line by lookup tables and search trees.

Several solutions have been proposed in the literature for constructing a polyhedral partition of the state space [20], [141], [117]. In [20], [18] some iterative techniques use a QP or LP to find feasible points and then split the parameters space by inverting one by one the constraints hyper-planes. As an alternative, in [141] the authors construct the unconstrained polyhedral region and then enumerate the others based on the combinations of active constraints. When the cost function is quadratic, the uniqueness of the optimal solution is guaranteed and the methods proposed in [20], [18], [141] work very well, at least for non-degenerate sets of constraints [153].

It is worth noticing that by using $l_1-$ or $l_\infty-$norms as the performance measure, the cost function is only positive semi-definite and the uniqueness of the optimal solution is not guaranteed and as a consequence, neither the continuity. A control law will have a practical advantage if the control action presents no jumps on the boundaries of the polyhedral partitions. When the optimal solution is not unique, the methods in [20], [18], [141] allow discontinuities as long as during the exploration of the parameters space, the optimal basis is chosen arbitrarily.

Note that based on the cost function (3.9) or (3.10) the MPC problem can be rewritten as follows

$$V(x(k)) = \min_z c^T z \tag{3.24}$$

subject to

$$G_l z \leq E_l x(k) + S_l$$

with

$$z = [\xi_1^T \ \xi_2^T \ \ldots \xi_{N_\xi}^T \ u_0^T \ u_1^T \ \ldots u_{N-1}^T]^T$$

where $\xi_i$, $i = 1,2,\ldots,N_\xi$ are slack variables and $N_\xi$ depends on the norm used and on the time horizon $N$. Details of how to compute vectors $c$, $S_l$ and matrices $G_l$, $E_l$ are well known [18].

The feasible domain for the LP problem (3.24) is defined by a finite number of inequalities with a right hand side linearly dependent on the vector of parameters $x(k)$, describing in fact a *parameterized polytope* [93]

$$P(x(k)) = \{z : G_l z \leq E_l x(k) + S_l\} \tag{3.25}$$

For simplicity, it is assumed that for all $x(k) \in X$, the polyhedral set $P(x(k))$ is bounded. With this assumption, $P(x(k))$ can be expressed in a dual (generator based) form as

$$P(x) = \text{Conv}\{v_i(x(k))\},\ i = 1, 2, \ldots, n_v \tag{3.26}$$

where $v_i$ are the parameterized vertices. Each parameterized vertex in (3.26) is characterized by a set of saturated inequalities. Once this set of active constraints is identified, one can write the linear dependence of the parameterized vertex in the vector of parameters

$$v_i(x(k)) = \overline{G}_{li}^{-1}\overline{E}_{li}x(k) + \overline{G}_{li}^{-1}\overline{S}_{li} \tag{3.27}$$

where $\overline{G}_{li}$, $\overline{E}_{li}$, $\overline{W}_{li}$ correspond to the subset of saturated inequalities for the $i-$th parameterized vertex.

As a first conclusion, the construction of the dual description (3.25), (3.26) requires the determination of the set of parameterized vertices. Efficient algorithms exist in this direction [93], the main idea being the analogy with a non-parameterized polytope in a higher dimension.

When the vector of parameter $x(k)$ varies inside the parameters space, the vertices of the feasible domain (3.25) may split or merge. This means that each parameterized vertex $v_i$ is defined only over a specific region in the parameters space. These regions $VD_i$ are called *validity domains* and can be constructed using simple projection mechanisms [93].

Once the entire family of parameterized vertices and their validity domains are available, the optimal solution can be constructed. It is clear that the space of feasible parameters can be partitioned in non-degenerate polyhedral regions $R_k \in \mathbb{R}^n$ such that the minimum

$$\min\left\{c^T v_i(x(k))| \quad v_i(x(k)) \text{ vertex of } P(x(k)) \text{ valid over } R_k\right\} \tag{3.28}$$

is attained by a constant subset of vertices of $P(x(k))$, denoted $v_i^*(x(k))$. The complete solution over $R_k$ is

$$z_k(x(k)) = \text{Conv}\left\{v_{1k}^*(x(k)), v_{2k}^*(x(k)), \ldots, v_{sk}^*(x(k))\right\} \tag{3.29}$$

The following theorem holds regarding the structure of the polyhedral partitions of the parameters space [118]

**Theorem 3.2.** *[118] Let the multi-parametric program in (3.24) and $v_i(x(k))$, $i = 1, 2, \ldots, n_v$ be the parameterized vertices of the feasible domain (3.25), (3.26) with their corresponding validity domains $VD_i$. If a parameterized vertex takes part in the description of the optimal solution for a region $R_k$, then it will be part of the family of optimal solution over its entire validity domain $VD_i$.*

*Proof.* See [118]. □

Theorem 3.2 states that if a parameterized vertex is selected as an optimal candidate, then it covers all its validity domain.

It is worth noticing that the complete optimal solution (3.29) takes into account the eventual non-uniqueness of the optimum, and it defines the entire family of optimal solutions using the parameterized vertices and their validity domains.

Once the entire family of optimal solutions is available, the continuity of the control law can be guaranteed as follows. Firstly if the optimal solution is unique, then there is no decision to be made, the explicit solution being the collection of the parameterized vertices and their validity domains. The continuity is intrinsic.

Conversely, the family of the optimal solutions can be enriched in the presence of several optimal parameterized vertices

$$\begin{cases} z_k(x(k)) = \alpha_{1k}v_{1k}^* + \alpha_{2k}v_{2k}^* + \ldots + \alpha_{sk}v_{sk}^* \\ \alpha_{ik} \geq 0,\ i = 1,2,\ldots,s \\ \alpha_{1k} + \alpha_{2k} + \ldots + \alpha_{sk} = 1 \end{cases} \tag{3.30}$$

passing to an infinite number of candidates (any function included in the convex combination of vertices being optimal). As mentioned previously, the vertices of the feasible domain split and merge. The changes occur with a preservation of the continuity. Hence the continuity of the control law is guaranteed by the continuity of the parameterized vertices. The interested reader is referred to [118] for further discussions on the related concepts and constructive procedures.

*Example 3.2.* To illustrate the parameterized vertices concept, consider the following feasible domain for the MPC optimization problem

$$P(x(k)) = P_1 \cap P_2(x(k)) \tag{3.31}$$

where $P_1$ is a fixed polytope

$$P_1 = \left\{ z \in \mathbb{R}^2 : \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & -1 \\ -1 & 0 \end{bmatrix} z \leq \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right\} \tag{3.32}$$

and $P_2(x(k))$ is a parameterized polyhedral set

$$P_2(x(k)) = \left\{ z \in \mathbb{R}^2 : \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} z \leq \begin{bmatrix} -1 \\ -1 \end{bmatrix} x(k) + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right\} \tag{3.33}$$

Note that $P_2(x(k))$ is an unbounded set. From equation (3.33), it is clear that

- If $x(k) \leq 0.5$, then $-x(k) + 0.5 \geq 0$. It follows that $P_1 \subset P_2(x(k))$. The polytope $P(x(k)) = P_1$ has the half-space representation as (3.32) and the vertex representation

  $$P(x(k)) = \mathrm{Conv}\{v_1, v_2, v_3, v_4\}$$

  where

  $$v_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ v_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ v_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ v_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

- If $0.5 \leq x(k) \leq 1.5$, then $-1 \leq -x(k)+0.5 \leq 0$. It follows that $P_1 \cap P_2(x(k)) \neq \emptyset$. Note that for the polytope $P(x(k))$ the half-spaces $z_1 = 0$ and $z_2 = 0$ are redundant. The polytope $P(x(k))$ has the half-space representation

$$P(x(k)) = \left\{ z \in \mathbb{R}^2 : \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} z \leq \begin{bmatrix} 0 \\ 0 \\ -1 \\ -1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \\ 0.5 \\ 0.5 \end{bmatrix} \right\}$$

and the vertex representation

$$P(x(k)) = \text{Conv}\{v_4, v_5, v_6, v_7\}$$

with

$$v_5 = \begin{bmatrix} 1 \\ x-0.5 \end{bmatrix}, \; v_6 = \begin{bmatrix} x-0.5 \\ 1 \end{bmatrix}, \; v_7 = \begin{bmatrix} x-0.5 \\ x-0.5 \end{bmatrix},$$

- If $1.5 < x(k)$, then $-x(k)+0.5 < -1$. It follows that $P_1 \cap P_2(x(k)) = \emptyset$. Hence $P(x(k)) = \emptyset$.

In conclusion, the parameterized vertices of $P(x(k))$ are

$$v_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; v_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; v_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; v_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$
$$v_5 = \begin{bmatrix} 1 \\ x(k)-0.5 \end{bmatrix}, \; v_6 = \begin{bmatrix} x(k)-0.5 \\ 1 \end{bmatrix}, \; v_7 = \begin{bmatrix} x(k)-0.5 \\ x(k)-0.5 \end{bmatrix},$$

and the validity domains

$$VD_1 = [-\infty \quad 0.5], \quad VD_2 = [0.5 \quad 1.5], \quad VD_3 = (1.5 \quad +\infty]$$

Table 3.1 presents the validity domains and their corresponding parameterized vertices.

**Table 3.1** Validity domains and their parameterized vertices

| $VD_1$ | $VD_2$ | $VD_3$ |
|---|---|---|
| $v_1, v_2, v_3, v_4$ | $v_4, v_5, v_6, v_7$ | $\emptyset$ |

Figure 3.2 shows the polyhedral sets $P_1$ and $P_2(x(k))$ with $x(k) = 0.3$, $x(k) = 0.9$ and $x(k) = 1.5$.

*Example 3.3.* Consider the discrete time linear time invariant system in example 3.1 with the same constraints on the state and input variables. Here we will use an MPC formulation, which guarantees recursive feasibility and stability.

By solving equations (3.21) and (3.22) with weighting matrices

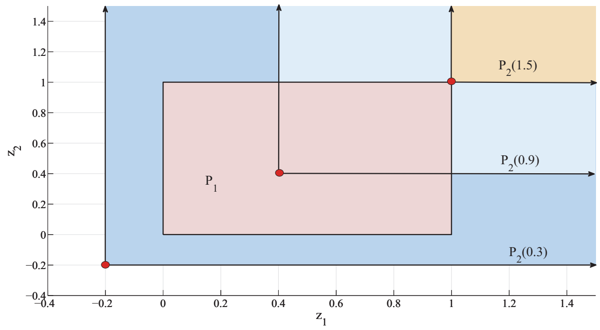



**Fig. 3.2** Polyhedral sets $P_1$ and $P_2(x(k))$ with $x(k) = 0.3$, $x(k) = 0.9$ and $x(k) = 1.5$ for example 3.2. For $x(k) \leq 0.5$, $P_1 \cap P_2(x(k)) = P_1$. For $0.5 \leq x(k) \leq 1.5$, $P_1 \cap P_2(x(k)) \neq \emptyset$. For $x(k) > 1.5$, $P_1 \cap P_2(x(k)) = \emptyset$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 0.1$$

one obtains

$$P = \begin{bmatrix} 1.5076 & -0.1173 \\ -0.1173 & 1.2014 \end{bmatrix}, \quad K = [-0.7015 \quad -1.0576]$$

The terminal set $\Omega$ is computed as a maximal polyhedral invariant set in Section 2.3.4

$$\Omega = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.7979 & -0.6029 \\ -0.7979 & 0.6029 \\ 1.0000 & 0 \\ -1.0000 & 0 \\ -0.5528 & -0.8333 \\ 0.5528 & 0.8333 \end{bmatrix} x \leq \begin{bmatrix} 2.5740 \\ 2.5740 \\ 2.0000 \\ 2.0000 \\ 0.7879 \\ 0.7879 \end{bmatrix} \right\}$$

Figure 3.3 shows the state space partition obtained by using the parameterized vertices framework as a method to construct an explicit solution to the MPC problem (3.23) with prediction horizon $N = 2$.

The control law over the state space partition is

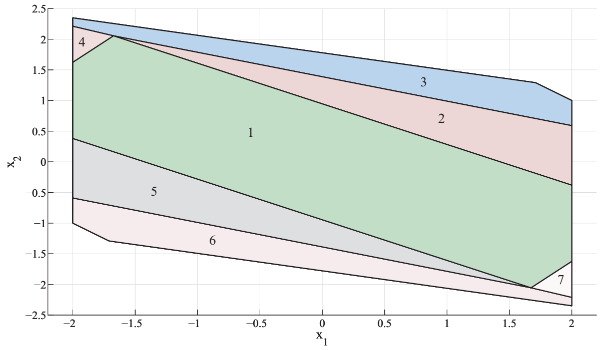
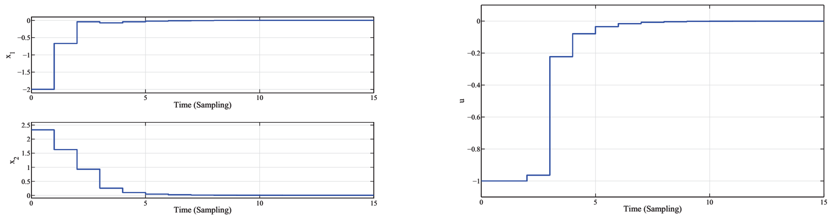



**Fig. 3.3** State space partition for example 3.3. Number of regions $N_r = 7$.

$$u(k) = \begin{cases}
-0.70x_1(k) - 1.06x_2(k) & \text{if } \begin{bmatrix} -0.80 & 0.60 \\ 0.80 & -0.60 \\ 0.55 & 0.83 \\ -0.55 & -0.83 \\ -1.00 & 0.00 \\ 1.00 & 0.00 \end{bmatrix} x(k) \leq \begin{bmatrix} 2.57 \\ 2.57 \\ 0.79 \\ 0.79 \\ 2.00 \\ 2.00 \end{bmatrix} \\
& \text{(Region 1)} \\
-0.56x_1(k) - 1.17x_2(k) + 0.47 & \text{if } \begin{bmatrix} 0.43 & 0.90 \\ -1.00 & 0.00 \\ 0.80 & -0.60 \end{bmatrix} x(k) \leq \begin{bmatrix} 1.14 \\ 2.00 \\ -2.57 \end{bmatrix} \\
& \text{(Region 4)} \\
-0.56x_1(k) - 1.17x_2(k) - 0.47 & \text{if } \begin{bmatrix} -0.43 & -0.90 \\ 1.00 & 0.00 \\ -0.80 & 0.60 \end{bmatrix} x(k) \leq \begin{bmatrix} 1.14 \\ 2.00 \\ -2.57 \end{bmatrix} \\
& \text{(Region 7)} \\
-1 & \text{if } \begin{bmatrix} 0.37 & 0.93 \\ 1.00 & 0.00 \\ -0.55 & -0.83 \end{bmatrix} x(k) \leq \begin{bmatrix} 1.29 \\ 2.00 \\ -0.79 \end{bmatrix} \\
& \text{(Region 2)} \\
1 & \text{if } \begin{bmatrix} -0.37 & -0.93 \\ -1.00 & 0.00 \\ 0.55 & 0.83 \end{bmatrix} x(k) \leq \begin{bmatrix} 1.29 \\ 2.00 \\ -0.79 \end{bmatrix} \\
& \text{(Region 5)} \\
-1 & \text{if } \begin{bmatrix} 0.71 & 0.71 \\ 0.27 & 0.96 \\ -1.00 & 0.00 \\ 1.00 & 0.00 \\ -0.43 & -0.90 \\ -0.37 & -0.93 \end{bmatrix} x(k) \leq \begin{bmatrix} 2.12 \\ 1.71 \\ 2.00 \\ 2.00 \\ -1.14 \\ -1.29 \end{bmatrix} \\
& \text{(Region 3)} \\
1 & \text{if } \begin{bmatrix} -0.71 & -0.71 \\ -0.27 & -0.96 \\ 1.00 & 0.00 \\ -1.00 & 0.00 \\ 0.43 & 0.90 \\ 0.37 & 0.93 \end{bmatrix} x(k) \leq \begin{bmatrix} 2.12 \\ 1.71 \\ 2.00 \\ 2.00 \\ -1.14 \\ -1.29 \end{bmatrix} \\
& \text{(Region 6)}
\end{cases}$$

For the initial condition $x(0) = [-2 \quad 2.33]$, Figure 3.4 shows the state and input trajectory as a function of time.




(a) State trajectory (b) Input trajectory.

**Fig. 3.4** State and input trajectory of the closed loop system as a function of time for example 3.3.

## 3.4 Vertex control

The vertex control framework was first proposed by Gutman and Cwikel in [54]. It gives a necessary and sufficient condition for stabilizing a discrete time linear time invariant system with polyhedral state and control constraints. The condition is that at each vertex of the controlled invariant set[4] $C_N$ there exists an admissible control action that brings the state to the interior of the set $C_N$. Then, this condition was extended to the uncertain plant case by Blanchini in [22]. A stabilizing controller is given by the convex combination of vertex controls in each sector with a Lyapunov function given by shrunken images of the boundary of the set $C_N$ [54], [22].

To begin, let us consider now the system of the form

$$x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \tag{3.34}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $w(k) \in \mathbb{R}^d$ are, respectively the state, input and disturbance vectors.

The matrices $A(k) \in \mathbb{R}^{n\times n}$, $B(k) \in \mathbb{R}^{n\times m}$ and $D(k) \subset \mathbb{R}^{n\times d}$ satisfy

$$\begin{cases} A(k) = \sum\limits_{i=1}^{q} \alpha_i(k)A_i, \quad B(k) = \sum\limits_{i=1}^{q} \alpha_i(k)B_i, \quad D(k) = \sum\limits_{i=1}^{q} \alpha_i(k)D_i \\ \sum\limits_{i=1}^{q} \alpha_i(k) = 1, \quad \alpha_i(k) \geq 0 \end{cases} \tag{3.35}$$

where the matrices $A_i$, $B_i$ and $D_i$ are given. A somewhat more general uncertainty description is given by equation (2.20) in Chapter 2 which can be transformed to the one in (3.35).

The state variables, the control variables and the disturbances are subject to the following polytopic constraints

[4] See Section 2.3.4

$$\begin{cases} x(k) \in X,\ X = \{x \in \mathbb{R}^n : F_x x \le g_x\} \\ u(k) \in U,\ U = \{u \in \mathbb{R}^m : F_u u \le g_u\} \\ w(k) \in W,\ W = \{w \in \mathbb{R}^d : F_w w \le g_w\} \end{cases} \quad (3.36)$$

where the matrices $F_x$, $F_u$, $F_w$ and the vectors $g_x$, $g_u$ and $g_w$ are assumed to be constant with $g_x > 0$, $g_u > 0$, $g_w > 0$ such that the origin is contained in the interior of $X$, $U$ and $W$. Using the results in Section 2.3.4, it is assumed that the robust controlled invariant set $C_N$ with some fixed integer $N > 0$ is determined in the form of a polytope, i.e.

$$C_N = \{x \in \mathbb{R}^n : F_N x \le g_N\} \quad (3.37)$$

Any state $x(k) \in C_N$ can be decomposed as follows

$$x = s x_s + (1-s) x_0 \quad (3.38)$$

where $0 \le s \le 1$, $x_s \in C_N$ and $x_0$ is the origin. In other words, the state $x$ is expressed as a convex combination of the origin and one other point $x_s \in C_N$.

Consider the following optimization problem

$$s^* = \min_{s,x_s} \{s\} \quad (3.39)$$

subject to

$$\begin{cases} s x_s = x \\ F_N x_s \le g_N \\ 0 \le s \le 1 \end{cases}$$

The following theorem holds

**Theorem 3.3.** *For all state $x \in C_N$ and $x$ is not the origin, the optimal solution of the problem (3.39) is reached if and only if $x$ is written as a convex combination of the origin and one point belonging to the boundary of the set $C_N$.*

*Proof.* It the optimal solution candidate $x_s$ is strictly inside the set $C_N$, then by setting

$$x_s^* = \mathrm{Fr}(C_N) \cap \overline{x, x_s}$$

i.e. $x_s^*$ is the intersection between the boundary of $C_N$ and the line connecting $x$ and $x_s$, one obtains

$$x = s^* x_s^* + (1-s^*) x_0 = s^* x_s^*$$

with $s^* < s$. Hence for the optimal solution $x_s^*$, it holds that $x_s^* \in \mathrm{Fr}(C_N)$. □

*Remark 3.1.* The optimal solution $s^*$ of the problem (3.39) can be seen as a measure of the distance from state $x$ to the origin.

*Remark 3.2.* The optimization problem (3.39) can be transformed to an LP problem as

$$s^* = \min_{s} \{s\} \quad (3.40)$$

subject to

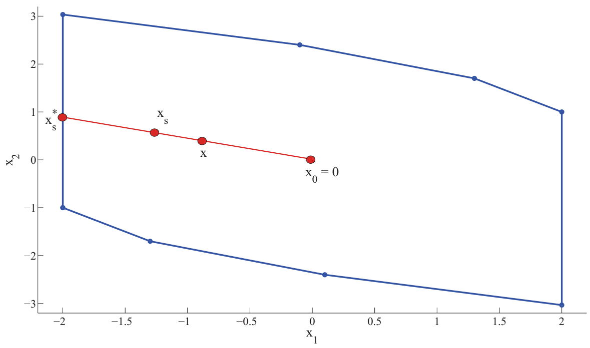



**Fig. 3.5** Graphical illustration of the proof of theorem 3.3.

$$\begin{cases} F_N x \leq s g_N \\ 0 \leq s \leq 1 \end{cases}$$

Clearly, for the given state $x$ the solution to the optimization problem (3.40) is

$$s^* = \max\{\frac{F_N}{g_N}x\} \tag{3.41}$$

where the ratios $\frac{F_N}{g_N}$ are element-wise. It is well known [95], [23] that with $s^*$ given as in (3.41), $s^*$ is the Minkownski functional. The level curves of the function $s^*$ are given by scaling the boundary of the set $C_N$.

Hence the explicit solution of the problem (3.39) is a set of $n-$dimensional pyramids $P_C^{(j)}$, each formed by one facet of $C_N$ as a base and the origin as a common vertex. By decomposing further these pyramids $P_C^{(j)}$ as a sequence of simplicies $C_N^{(j)}$, each formed by $n$ vertices $\{x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)}\}$ of the base of $P_C^{(j)}$ and the origin, having the following properties

- $C_N^{(j)}$ has nonempty interior.
- $\text{Int}(C_N^{(j)}) \cap \text{Int}(C_N^{(l)}) = \emptyset, \forall j \neq l$.
- $\bigcup\limits_j C_N^{(j)} = C_N$.

Let

$$U^{(j)} = [u_1^{(j)} \quad u_2^{(j)} \quad \ldots \quad u_n^{(j)}]$$

be the $m \times n$ matrix defined by chosen admissible control values[5] satisfying (3.34) at the vertices $\{x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)}\}$.

[5] By an admissible control value we understand any control value that is the first of a sequence of control values that bring the state from the vertex to the interior of the feasible set in a *finite number* of steps, see [54].

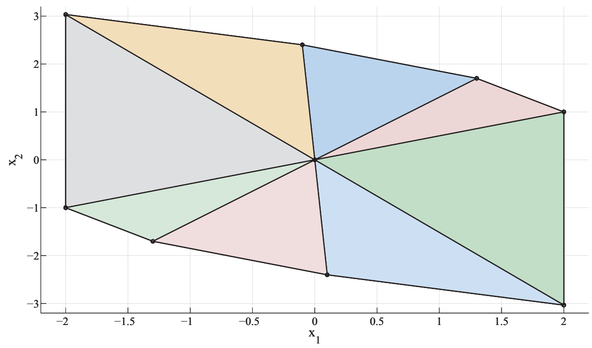



**Fig. 3.6** Graphical illustration of the simplex decomposition.

*Remark 3.3.* Maximizing the control action at the vertices $v \in \mathbb{R}^n$ of the controlled invariant set $C_N$ can be achieved by solving the following optimization problem

$$J = \max_{u} \|u\|_p \tag{3.42}$$

subject to

$$\begin{cases} F_N(A_i v + B_i u) \leq g_N - \max\limits_{w \in W}\{F_N D_i w\}, \forall i = 1, 2 \ldots, q \\ F_u u \leq g_u \end{cases}$$

where $\|u\|_p$ is the $p-$norm of the vector $u$. Since the set $C_N$ is robust controlled invariant, problem (3.42) is always feasible.

For all $x(k) \in C_N$, there exists an index $j$ corresponding to a simplex decomposition of $C_N$ such that $x(k) \in C_N^{(j)}$ and hence $x_s^*(k)$ is on the base of $C_N^{(j)}$. Therefore $x_s^*(k)$ can be written as a convex combination of $\{x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)}\}$, i.e.

$$x_s^*(k) = \beta_1 x_1^{(j)} + \beta_2 x_2^{(j)} + \ldots + \beta_n x_n^{(j)} \tag{3.43}$$

with

$$\begin{cases} \beta_i \geq 0, \forall i = 1, 2, \ldots, n \\ \sum\limits_{i=1}^{n} \beta_i = 1 \end{cases}$$

By substituting $x(k) = s^*(k) x_s^*(k)$ in equation (3.43), one obtains

$$x(k) = s^*(k)\{\beta_1 x_1^{(j)} + \beta_2 x_2^{(j)} + \ldots + \beta_n x_n^{(j)}\}$$

By denoting

$$\gamma_i = s^*(k)\beta_i, \forall i = 1, 2, \ldots, n$$

one gets

$$x(k) = \gamma_1 x_1^{(j)} + \gamma_2 x_2^{(j)} + \ldots + \gamma_n x_n^{(j)} \tag{3.44}$$

with

$$\begin{cases} \gamma_i \geq 0, \forall i = 1, 2, \ldots, n \\ \sum\limits_{i=1}^{n} \gamma_i = s^*(k) \sum\limits_{i=1}^{n} \beta_i = s^*(k) \end{cases}$$

*Remark 3.4.* Let $\{x_1, x_2, \ldots, x_{n_c}\}$ be the vertices of the polytope $C_N$ and $n_c$ be the number of vertices. It is well known [54] that the optimization problem (3.40) is equivalent to the following LP problem

$$\min_{\gamma_i} \{\gamma_1 + \gamma_2 + \ldots + \gamma_{n_c}\} \tag{3.45}$$

subject to

$$\begin{cases} \gamma_1 x_1 + \gamma_2 x_2 + \ldots + \gamma_{n_c} x_{n_c} = x(k) \\ \gamma_i \geq 0, \forall i = 1, 2, \ldots, n \\ \sum\limits_{i=1}^{n_c} \gamma_i \leq 1 \end{cases}$$

Equation (3.44) can be rewritten in a compact form as

$$x(k) = X^{(j)} \gamma$$

where

$$\begin{aligned} X^{(j)} &= [x_1^{(j)} \quad x_2^{(j)} \quad \ldots \quad x_n^{(j)}] \\ \gamma &= [\gamma_1 \quad \gamma_2 \quad \ldots \quad \gamma_n]^T \end{aligned}$$

Since $C_N^{(j)}$ has nonempty interior, matrix $X^{(j)}$ is invertible. It follows that

$$\gamma = \{X^{(j)}\}^{-1} x(k) \tag{3.46}$$

Consider the following control law

$$u(k) = \gamma_1 u_1^{(j)} + \gamma_2 u_2^{(j)} + \ldots + \gamma_n u_n^{(j)} \tag{3.47}$$

or

$$u(k) = U^{(j)} \gamma \tag{3.48}$$

By substituting equation (3.46) in equation (3.48) one gets

$$u(k) = U^{(j)} \{X^{(j)}\}^{-1} x(k) = K^{(j)} x(k) \tag{3.49}$$

with

$$K^{(j)} = U^{(j)} \{X^{(j)}\}^{-1} \tag{3.50}$$

Hence for $x \in C_N^{(j)}$ the controller is an linear feedback state law whose gains are obtained simply by linear interpolation of the control values at the vertices of the simplex.

$$u = K^{(j)}x, \qquad \forall x \in C_N^{(j)} \tag{3.51}$$

The piecewise linear control law (3.51) was first proposed by Gutman and Cwikel in [54] for the discrete-time linear time-invariant system case. In the original work [54], the state feedback control (3.51) was called the *vertex controller*. The extension to the uncertain plant case was proposed by Blanchini in [22].

*Remark 3.5.* Clearly, once the piecewise linear function (3.51) is pre-calculated, the control action can be computed by determining the simplex that contains the current state, which gives an explicit piecewise linear control law. An alternative approach for computing the control action is based on solving *on-line* the LP problem (3.45) and then apply the control

$$u(k) = \gamma_1 u_1 + \gamma_2 u_2 + \ldots + \gamma_{n_c} u_{n_c} \tag{3.52}$$

where $u_1, u_2, \ldots, u_{n_c}$ are the stored control values at the vertices $x_1, x_2, \ldots, x_{n_c}$.

The following theorem holds

**Theorem 3.4.** *For system (3.34) and constraints (3.36), the vertex control law (3.47) or (3.51) guarantees recursive feasibility for all $x \in C_N$.*

*Proof.* A basic explanation is provided in the original work of [54]. Here a new and simpler proof is proposed using convexity of the set $C_N$ and linearity of the system (3.34). For recursive feasibility, one has to prove that for all $x(k) \in C_N$

$$\begin{cases} F_u u(k) \le g_u \\ x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \in C_N \end{cases}$$

For all $x(k) \in C_N$, there exists an index $j$ such that $x(k) \in C_N^{(j)}$. It follows that

$$\begin{aligned} F_u u(k) &= F_u\{\gamma_1 u_1^{(j)} + \gamma_2 u_2^{(j)} + \ldots + \gamma_n u_n^{(j)}\} \\ &= \gamma_1 F_u u_1^{(j)} + \gamma_2 F_u u_2^{(j)} + \ldots + \gamma_n F_u u_n^{(j)} \\ &\le \gamma_1 g_u + \gamma_2 g_u + \ldots + \gamma_n g_u \\ &\le g_u \sum_{i=1}^{n} \gamma_i \le s^*(k) g_u \le g_u \end{aligned}$$

and

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) + D(k)w(k) \\ &= A(k) \sum_{i=1}^{n} \gamma_i x_i^{(j)} + B(k) \sum_{i=1}^{n} \gamma_i u_i^{(j)} + D(k)w(k) \\ &= \sum_{i=1}^{n} \gamma_i \{A(k)x_i^{(j)} + B(k)u_i^{(j)} + D(k)w(k)\} + (1 - s^*(k))D(k)w(k) \end{aligned}$$

One has

$$
\begin{aligned}
F_N x(k+1) &= \sum_{i=1}^{n} \gamma_i F_N \{A(k)x_i^{(j)} + B(k)u_i^{(j)} + D(k)w(k)\} + (1-s^*(k))F_N D(k)w(k) \\
&\leq \sum_{i=1}^{n} \gamma_i g_N + (1-s^*(k))F_N D(k)w(k) \\
&\leq s^*(k)g_N + (1-s^*(k))F_N D(k)w(k)
\end{aligned}
$$

Since the set $C_N$ is robust controlled invariant and containing the origin in its interior, it follows that

$$\max_{w\in W}\{F_N D_i w(k)\} \leq g_N$$

Hence

$$F_N x(k+1) \leq s^*(k)g_N + (1-s^*(k))g_N \leq g_N$$

or in other words, $x(k+1) \in C_N$. □

In the absence of disturbances, i.e. $w(k) = 0$, $\forall k \geq 0$, the following theorem holds

**Theorem 3.5.** *Consider the uncertain system (3.34) with input and state constraints (3.36), then the closed loop system with the piecewise linear control law (3.47) or (3.51) is robustly asymptotically stable.*

*Proof.* A proof is given in [54], [22]. Here we give an alternative proof providing a geometrical insight into the vertex control scheme. Consider the following positive definite function

$$V(x) = s^*(k) \tag{3.53}$$

$V(x)$ is a Lyapunov function candidate. For any $x(k) \in C_N$, there exists an index $j$ such that $x(k) \in C_N^{(j)}$. Hence

$$x(k) = s^*(k)x_s^*(k), \quad \text{and} \quad u(k) = s^*(k)u_s^*(k)$$

where the control action $u_s^*(k)$ is given by 3.43

$$u_s^*(k) = \beta_1 u_1^{(j)} + \beta_2 u_2^{(j)} + \ldots + \beta_n u_n^{(j)}$$

It follows that

$$
\begin{aligned}
x(k+1) &= A(k)x(k) + B(k)u(k) \\
&= A(k)s^*(k)x_s^*(k) + B(k)s^*(k)u_s^*(k) \\
&= s^*(k)\{A(k)x_s^*(k) + B(k)u_s^*(k)\} = s^*(k)x_s(k+1)
\end{aligned}
$$

where $x_s(k+1) = A(k)x_s^*(k) + B(k)u_s^*(k) \in C_N$. Hence $s^*(k)$ gives a possible decomposition (3.38) of $x(k+1)$.

By using the interpolation based on linear programming (3.40), one gets a different and optimal decomposition, namely

$$x(k+1) = s^*(k+1)x_s^*(k+1)$$

with $x_s^*(k+1) \in C_N$. It follows that $s^*(k+1) \leq s^*(k)$ and $V(x)$ is a non-increasing function.

From the fact that the level curves of the function $V(x) = s^*(k)$ are given by scaling the boundary of the feasible set, and the contractiveness property of the control values at the vertices of the feasible set guarantees that there is no initial condition $x(0) \in C_N$ such that $s^*(k) = s^*(0) = 1$ for sufficiently large and finite $k$, one concludes that $V(x) = s^*(k)$ is a Lyapunov function for all $x(k) \in C_N$. Hence the closed loop system with the vertex control law (3.51) is robustly asymptotically stable. □

*Example 3.4.* Consider the discrete time system in example 3.1 and 3.3

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.7 \end{bmatrix} u(k) \tag{3.54}$$

The constraints are

$$-2 \leq x_1 \leq 2, \quad -5 \leq x_2 \leq 5, -1 \leq u \leq 1 \tag{3.55}$$

Based on procedure 2.3 in Section 2.3.4, the controlled invariant set $C_N$ is computed and depicted in Figure 3.7.

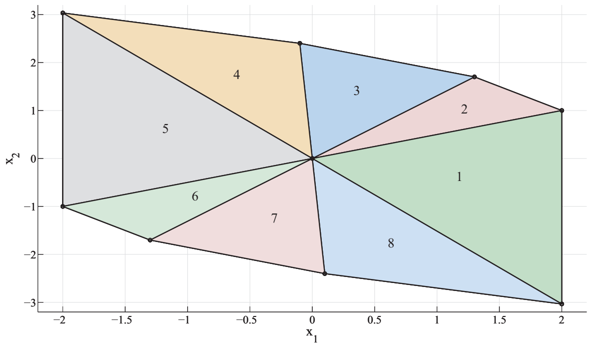



**Fig. 3.7** controlled invariant set $C_N$ and state space partition of vertex control for example 3.4.

The set of vertices of $C_N$ is given by the matrix $V(C_N)$ below, together with the control matrix $U_v$

$$V(C_N) = \begin{bmatrix} 2.00 & 1.30 & -0.10 & -2.00 & -2.00 & -1.30 & 0.10 & 2.00 \\ 1.00 & 1.70 & 2.40 & 3.03 & -1.00 & -1.70 & -2.40 & -3.03 \end{bmatrix} \tag{3.56}$$

and

$$U_v = \begin{bmatrix} -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.57}$$

The vertex control law over the state space partition is

$$u(k) = \begin{cases} -0.25x_1(k) - 0.50x_2(k), & \text{if } x(k) \in C_N^{(1)} \text{ or } x(k) \in C_N^{(5)} \\ -0.33x_1(k) - 0.33x_2(k), & \text{if } x(k) \in C_N^{(2)} \text{ or } x(k) \in C_N^{(6)} \\ -0.21x_1(k) - 0.43x_2(k), & \text{if } x(k) \in C_N^{(3)} \text{ or } x(k) \in C_N^{(7)} \\ -0.14x_1(k) - 0.42x_2(k), & \text{if } x(k) \in C_N^{(4)} \text{ or } x(k) \in C_N^{(8)} \end{cases} \tag{3.58}$$

with

$$\begin{aligned}
C_N^{(1)} &= \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.00 & 0.00 \\ -0.45 & 0.89 \\ -0.83 & -0.55 \end{bmatrix} x \leq \begin{bmatrix} 2.00 \\ 0.00 \\ 0.00 \end{bmatrix} \right\} \\
C_N^{(2)} &= \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.71 & 0.71 \\ 0.45 & -0.89 \\ -0.79 & 0.61 \end{bmatrix} x \leq \begin{bmatrix} 2.12 \\ 0.00 \\ 0.00 \end{bmatrix} \right\} \\
C_N^{(3)} &= \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.79 & -0.61 \\ 0.45 & 0.89 \\ -1.00 & -0.04 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 2.10 \\ 0.00 \end{bmatrix} \right\} \\
C_N^{(4)} &= \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.00 & 0.04 \\ -0.83 & -0.55 \\ 0.32 & 0.95 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 2.25 \end{bmatrix} \right\} \\
C_N^{(5)} &= \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -1.00 & 0.00 \\ 0.45 & -0.89 \\ 0.83 & 0.55 \end{bmatrix} x \leq \begin{bmatrix} 2.00 \\ 0.00 \\ 0.00 \end{bmatrix} \right\} \\
C_N^{(6)} &= \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.71 & -0.71 \\ -0.45 & 0.89 \\ 0.79 & -0.61 \end{bmatrix} x \leq \begin{bmatrix} 2.12 \\ 0.00 \\ 0.00 \end{bmatrix} \right\} \\
C_N^{(7)} &= \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.79 & 0.61 \\ -0.45 & -0.89 \\ 1.00 & 0.04 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 2.10 \\ 0.00 \end{bmatrix} \right\} \\
C_N^{(8)} &= \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -1.00 & -0.04 \\ 0.83 & 0.55 \\ -0.32 & -0.95 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 2.25 \end{bmatrix} \right\}
\end{aligned}$$

Figure 3.8 presents state trajectories of the closed loop system for different initial conditions.

For the initial condition $x(0) = [-2.0000 \quad 3.0333]^T$, Figure 3.9 shows the state trajectory, the input trajectory and the interpolating coefficient $s^*$ as a function of time. As expected $s^*(k)$ is a positive and non-increasing function.

From Figure 3.9(b), it is worth noticing that using the vertex controller, the control values are saturated only on the boundary of the set $C_N$, i.e. when $s^* = 1$. And also the state trajectory at some moments is parallel to the boundary of the set $C_N$, i.e when $s^*$ is constant. At these moments, the control values are also constant due to the choice of the control values at the vertices of the set $C_N$.

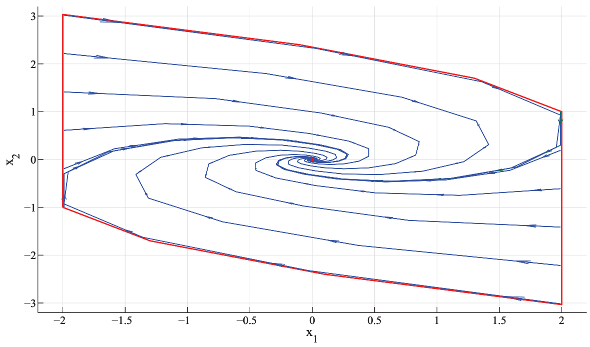
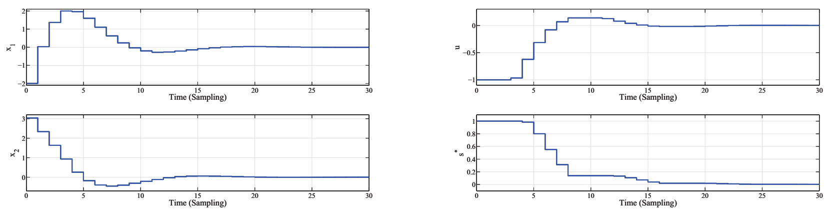



**Fig. 3.8** State trajectories of the closed loop system for example 3.4.




(a) State trajectory (b) Input trajectory and interpolating coefficient

**Fig. 3.9** State trajectory, input trajectory and interpolating coefficient of the closed loop system as a function of time for example 3.4.

# Part II
# Interpolation based control

# Chapter 4
# Interpolation Based Control – Nominal State Feedback Case

This chapter presents several original contributions on constrained control algorithms for discrete-time linear systems. Using a generic design principle, several types of control laws will be proposed for time-invariant models, their robust version being investigated in the next chapters.

The first control law is based on an interpolation technique between a global vertex controller and a local controller through the resolution of a simple linear programming problem. An implicit and explicit solutions of this control law will be presented. The second control law is obtained as a solution of a quadratic programming problem. Then to fully utilize the capacity of actuators and guarantee the input constraints, a saturation function on the input is considered. For the third control law, it is shown that the convex hull of a set of invariant ellipsoids is invariant. A method for constructing a continuous feedback law based on interpolation between the *saturated* controllers of the ellipsoids will also be presented.

For all the types of controllers, recursive feasibility and asymptotic stability will be proved. Several numerical examples are given to support the algorithms with illustrative simulations.

## 4.1 Problem formulation

In this chapter, we consider the problem of regulating to the origin the following discrete-time linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k) \tag{4.1}$$

where $x(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}^m$ are respectively the state and the input, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the system matrices. Both the state vector $x(k)$ and the control vector $u(k)$ are subject to polytopic constraints

$$\begin{cases} x(k) \in X,\, X = \{x \in \mathbb{R}^n : F_x x \leq g_x\} \\ u(k) \in U,\, U = \{u \in \mathbb{R}^m : F_u u \leq g_u\} \end{cases} \forall k \geq 0 \tag{4.2}$$

where the matrices $F_x$, $F_u$ and the vectors $g_x$, $g_u$ are assumed to be constant with $g_x > 0$ and $g_u > 0$ such that the origin is contained in the interior of $X$ and $U$. Recall that the inequalities are taken element-wise. We assume that the states of the system are measurable. We also assume that the pair $(A,B)$ is stabilizable, i.e. all uncontrollable states have stable dynamics.

## 4.2 Interpolation based on linear programming - Implicit solution

Define a linear controller $K \in \mathbb{R}^{m\times n}$, such that

$$u(k) = Kx(k) \tag{4.3}$$

quadratically stabilizes the system (4.1) with some desired performance specifications. The details of such a synthesis procedure are not reproduced here, but we assume that feasibility is guaranteed. Based on procedures 2.1 or 2.2, a maximal invariant set $\Omega_{max}$ can be computed in the form

$$\Omega_{max} = \{x \in \mathbb{R}^n : F_o x \leq g_o\} \tag{4.4}$$

when applying the control law $u(k) = Kx(k)$. Furthermore with some given and fixed integer $N > 0$, based on procedure 2.3 one can find a controlled invariant set $C_N$ in the form

$$C_N = \{x \in \mathbb{R}^n : F_N x \leq g_N\} \tag{4.5}$$

such that all $x \in C_N$ can be steered into $\Omega_{max}$ in no more than $N$ steps when a suitable control is applied. As in Section 3.4, the polytope $C_N$ is decomposed into a set of simplicies $C_N^{(j)}$, each formed by $n$ vertices of $C_N$ and the origin. For all $x(k) \in C_N^{(j)}$, the vertex controller

$$u(k) = K^{(j)}x(k), \quad \forall x(k) \in C_N^{(j)} \tag{4.6}$$

can be applied with $K^{(j)}$ defined as in (3.50). From Section 3.4, it is clear that the closed loop system (4.1) with vertex control is asymptotically stable for all initial states $x \in C_N$.

The main advantage of the vertex control scheme is the size of the domain of attraction, i.e. the set $C_N$. It is clear that the controlled invariant set $C_N$, that is the feasible domain for the vertex control, might be as large as that of any other constrained control scheme. However, a weakness of vertex control is that the full control range is exploited only on the border of the set $C_N$ in the state space, with progressively smaller control action when state approaches the origin. Hence the time to regulate the plant to the origin is often unnecessary long. A way to overcome this shortcoming is to switch to another, more aggressive, local controller, e.g.

a state feedback controller $u(k) = Kx(k)$, when the state reaches the maximal invariant set $\Omega_{max}$ of the local controller. The disadvantage of this solution is that the control action becomes non-smooth [103].

In this section, a method to overcome the non-smooth control action [103] will be proposed. For this purpose, any state $x(k) \in C_N$ can be decomposed as follows

$$x(k) = c(k)x_v(k) + (1 - c(k))x_o(k) \tag{4.7}$$

with $x_v \in C_N$, $x_o \in \Omega_{max}$ and $0 \le c \le 1$. Figure 4.1 illustrates such a decomposition.

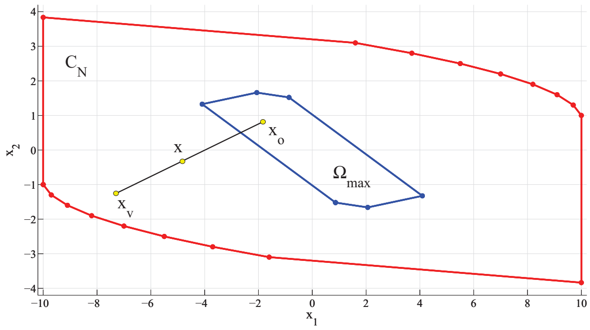



**Fig. 4.1** Interpolation based control. Any state $x(k)$ can be expressed as a convex combination of $x_v(k) \in C_N$ and $x_o(k) \in \Omega_{max}$.

Consider the following control law

$$u(k) = c(k)u_v(k) + (1 - c(k))u_o(k) \tag{4.8}$$

where $u_v(k)$ is obtained by applying the vertex control law (4.6) at $x_v(k)$ and $u_o(k) = Kx_o(k)$ is the control law (4.3) that is feasible in $\Omega_{max}$.

**Theorem 4.1.** *For system (4.1) and constraints (4.2), the control law (4.8) guarantees recursive feasibility for all initial states $x(0) \in C_N$.*

*Proof.* For recursive feasibility, one has to prove that

$$\begin{cases} F_u u(k) \le g_u \\ x(k+1) = Ax(k) + Bu(k) \in C_N \end{cases}$$

for all $x(k) \in C_N$. For the input constraints

$$\begin{aligned} F_u u(k) &= F_u \{c(k)u_v(k) + (1 - c(k))u_o(k)\} \\ &= c(k)F_u u_v(k) + (1 - c(k))F_u u_o(k) \\ &\le c(k)g_u + (1 - c(k))g_u = g_u \end{aligned}$$

and

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= A\{c(k)x_v(k) + (1-c(k))x_o(k)\} + B\{c(k)u_v(k) + (1-c(k))u_o(k)\} \\ &= c(k)\{Ax_v(k) + Bu_v(k)\} + (1-c(k))\{Ax_o(k) + Bu_o(k)\} \end{aligned}$$

Since $Ax_v(k) + Bu_v(k) \in C_N$ and $Ax_o(k) + Bu_o(k) \in \Omega_{max} \subseteq C_N$, it follows that $x(k+1) \in C_N$. □

In order to approach as much as possible to the unconstrained local controller, the minimization of the interpolating coefficient $c(k)$ needs to be considered. This can be done by solving the following nonlinear optimization problem

$$c^* = \min_{x_v, x_o, c} \{c\} \tag{4.9}$$

subject to

$$\begin{cases} F_N x_v \le g_N \\ F_o x_o \le g_o \\ c x_v + (1-c) x_o = x \\ 0 \le c \le 1 \end{cases}$$

Denote $r_v = c x_v \in \mathbb{R}^n$, $r_o = (1-c)x_o \in \mathbb{R}^n$. Since $x_v \in C_N$ and $x_o \in \Omega_{max}$, it follows that $r_v \in cC_N$ and $r_o \in (1-c)\Omega_{max}$ or equivalently

$$\begin{cases} F_N r_v \le c g_N \\ F_o r_o \le (1-c) g_o \end{cases}$$

With this change of variables, the nonlinear optimization problem (4.9) is transformed into a linear programming problem as follows

$$c^* = \min_{r_v, c} \{c\} \tag{4.10}$$

subject to

$$\begin{cases} F_N r_v \le c g_N \\ F_o(x - r_v) \le (1-c) g_o \\ 0 \le c \le 1 \end{cases}$$

*Remark 4.1.* It is interesting to observe that the proposed interpolation scheme, by the minimization of the interpolation coefficient is the antithesis of the maximization of $c$. It is obvious that in the latter case $c = 1$ for all $x \in C_N$ and the interpolating controller (4.8) turns out to be the vertex controller.

**Theorem 4.2.** *The control law using interpolation based on linear programming (4.10) guarantees asymptotic stability for all initial states $x(0) \in C_N$.*

*Proof.* First of all we will prove that all solutions starting in $C_N \setminus \Omega_{max}$ will reach the set $\Omega_{max}$ in *finite time*. For this purpose, consider the following non-negative function

$$V(x(k)) = c^*(k), \quad \forall x(k) \in C_N \setminus \Omega_{max} \tag{4.11}$$

$V(x(k))$ is a candidate Lyapunov function. For any $x(k) \in C_N \setminus \Omega_{max}$, one has

$$x(k) = c^*(k)x_v^*(k) + (1 - c^*(k))x_o^*(k)$$

and consequently

$$u(k) = c^*(k)u_v(k) + (1 - c^*(k))u_o(k)$$

It follows that

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= c^*(k)x_v(k+1) + (1 - c^*(k))x_o(k+1) \end{aligned}$$

where

$$\begin{aligned} x_v(k+1) &= Ax_v^*(k) + Bu_v(k) \in C_N \\ x_o(k+1) &= Ax_o^*(k) + Bu_o(k) \in \Omega_{max} \end{aligned}$$

Hence $c^*(k)$ gives a feasible decomposition (4.7) of $x(k+1)$. By using the interpolation based on linear programming (4.10), a possibly different and optimal decomposition is obtained, namely

$$x(k+1) = c^*(k+1)x_v^*(k+1) + (1 - c^*(k+1))x_o^*(k+1)$$

where $x_v^*(k+1) \in C_N$ and $x_o^*(k+1) \in \Omega_{max}$. It follows that $c^*(k+1) \leq c^*(k)$ and $V(x(k))$ is a non-increasing function and a Lyapunov function in the weak sense as the inequality is not strict.

Using the vertex controller, an interpolation between the vertices of the feasible controlled invariant set $C_N$ and the origin is obtained. Conversely using the controller (4.7), (4.8), (4.10) an interpolation is constructed between the vertices of the feasible controlled invariant set $C_N$ and the vertices of the invariant set $\Omega_{max}$ which contains the origin as an interior point. This last property proves that the vertex controller is a feasible choice for the interpolation based technique. From these facts we conclude that

$$c^*(k) \leq s^*(k)$$

for any $x(k) \in C_N$, with $s^*(k)$ obtained as in (3.40), Section 3.4.

Since the vertex controller is exponentially stable, the state reaches any bounded set around the origin in finite time. In our case this property will imply that using the controller (4.7), (4.8), (4.10) the state of the closed loop system reaches the invariant set $\Omega_{max}$ in *finite time* or equivalently that there exists a finite $k$ such that $c^*(k) = 0$.

The proof is complete by noting that inside $\Omega_{max}$, the LP problem has (4.10) the trivial solution $c^* = 0$. Hence the controller (4.7), (4.8), (4.10) turns out to be the local controller. The feasible stabilizing controller $u(k) = Kx(k)$ is *contractive*, and thus the interpolation-based controller assures asymptotic stability for all $x \in C_N$. □

Since $r_v^*(k) = c^*(k)x_v^*(k)$ and $r_o(k) = (1 - c^*(k))x_o^*(k)$, it follows that

$$u(k) = u_{rv}(k) + u_{ro}(k) \tag{4.12}$$

where $u_{rv}(k)$ is obtained by applying the vertex control law at $r_v^*(k)$ and $u_{ro}(k) = Kr_o(k)$.

A simple on-line algorithm for the interpolation based controller is given here.

**Algorithm 4.1:** Interpolation based control - Implicit solution

1. Measure the current state of the system $x(k)$.
2. Solve the LP problem (4.10).
3. Compute $u_{rv}$ in (4.12) by solving an LP-program or otherwise determine in which $C_N^{(j)}$ simplex $r_v$ belongs and using (3.51), or explicitly solve the LP-program (3.45) and then using (3.52).
4. Implement as input the control value (4.12).
5. Wait for the next time instant $k := k+1$.
6. Go to step 1 and repeat.

*Remark 4.2.* From the computational complexity point of view, we note that at each time instant algorithm 4.1 requires the solution of the LP problem (4.10) of dimension $n+1$ with $n$ being the dimension of state and another LP problem to find $u_{rv}$[1]. Clearly, this extremely simple optimization problem is comparable with a one-step ahead MPC.

## 4.3 Interpolation based on linear programming - Explicit solution

### *4.3.1 Geometrical interpretation*

This subsection is dedicated to the computation and structural implications of the interpolation based on linear programming (4.10), which can be assimilated to a multi-parametric optimization by the fact that the current state plays the role of a vector of parameters. The control law can be pre-computed off-line in an explicit form as a piecewise affine state feedback over a polyhedral partition of the state space, thus avoiding real-time optimization.

*Remark 4.3.* The following properties can be exploited during the construction stage

- For all $x \in \Omega_{max}$, the result of the optimal interpolation problem has the trivial solution $c^* = 0$ and thus $x_o^* = x$ in (4.7).
- Let $x \in C_N \setminus \Omega_{max}$ with a particular convex combination $x = cx_v + (1-c)x_o$ where $x_c \in C_N$ and $x_o \in \Omega_{max}$. If $x_o$ is strictly inside $\Omega_{max}$, define

[1] In the implicit vertex control case.

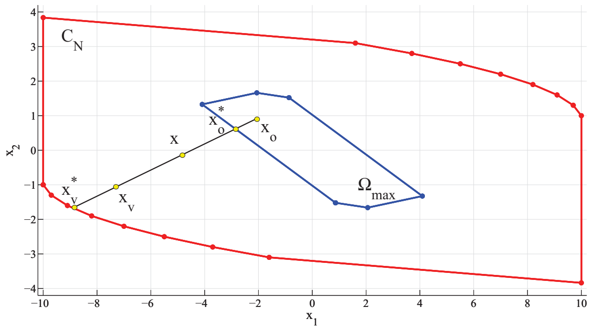



**Fig. 4.2** Graphical illustration for remark 4.3. For all $x \in C_N \setminus \Omega_{max}$, the optimal solution of the problem (4.10) is reached if and only if $x$ is written as a convex combination of $x_v$ and $x_o$ with $x_v \in \mathrm{Fr}(C_N)$ and $x_o \in \mathrm{Fr}(\Omega_{max})$.

$$\widetilde{x_o} = \mathrm{Fr}(\Omega_{max}) \cap \overline{x, x_o}$$

i.e. $\widetilde{x_o}$ is the intersection between the boundary of $\Omega_{max}$ and the segment connecting $x$ and $x_o$. Using convexity arguments, one has

$$x = \widetilde{c}x_v + (1-\widetilde{c})\widetilde{x_o}$$

with $\widetilde{c} < c$. In general terms, for all $x \in C_N \setminus \Omega_{max}$ the optimal interpolation (4.7), (4.8), (4.10) leads to a solution $\{x_v^*, x_o^*\}$ with $x_o^* \in \mathrm{Fr}(\Omega_{max})$.

- On the other hand, if $x_v$ is strictly inside $C_N$, by setting

$$\widehat{x_v} = \mathrm{Fr}(C_N) \cap \overline{x, x_v}$$

i.e. $\widehat{x_v}$ is the intersection between the boundary of $C_N$ and the ray connecting $x$ and $x_v$. One obtains

$$x = \widehat{c}\widehat{x_v} + (1-\widehat{c})x_o$$

with $\widehat{c} < c$, leading to the conclusion that for the optimal solution $\{x_v^*, x_o^*\}$, it holds that $x_v^* \in \mathrm{Fr}(C_N)$.

From the previous remark we conclude that for all $x \in C_N \setminus \Omega_{max}$ the interpolating coefficient $c$ reaches a minimum in (4.10) if and only if $x$ is written as a convex combination of two points, one belonging to the boundary of $\Omega_{max}$ and the other on the boundary of $C_N$.

**Theorem 4.3.** *For a given $x \in C_N \setminus \Omega_{max}$, the convex combination $x = cx_v + (1-c)x_o$ gives the smallest value of the interpolating coefficient $c$ if and only if the ratio $\frac{\|x_v - x\|}{\|x - x_o\|}$ is maximal, where $\|\cdot\|$ denotes the vector norm.*

*Proof.* One has

$$
\begin{aligned}
& x = c x_v + (1-c) x_o \\
& \Rightarrow x_v - x = x_v - c x_v - (1-c) x_o = (1-c)(x_v - x_o) \\
& \Rightarrow \|x_v - x\| = (1-c)\, \|x_v - x_o\|
\end{aligned}
$$

Analogously, it holds that

$$\|x - x_o\| = c\, \|x_v - x_o\|$$

leading to

$$\frac{\|x_v - x\|}{\|x - x_o\|} = \frac{(1-c)\, \|x_v - x_o\|}{c\, \|x_v - x_o\|} = \frac{1}{c} - 1$$

Apparently, $c$ is minimal if and only if $\frac{1}{c} - 1$ is maximal, or in other words $\frac{\|x_v - x\|}{\|x - x_o\|}$ reaches its maximum. □

### 4.3.2 Analysis in $\mathbb{R}^2$

In this subsection an analysis of the LP problem (4.10) in the $\mathbb{R}^2$ parameter space is presented with reference to Figure 4.3. The discussion is insightful in what concerns the properties of the partition in the explicit solution. The problem considered here is to decompose the polyhedral $X_{1234}$ such that the explicit solution $c^* = \min\{c\}$ is given in the decomposed cells.

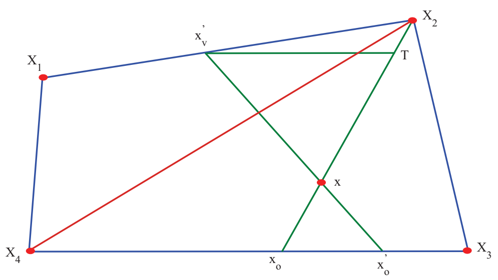



**Fig. 4.3** Graphical illustration in the $\mathbb{R}^2$ case.

For illustration we will consider four vertices $X_i$, $i = 1,2,3,4$ and any point $x \in \text{Conv}(X_1, X_2, X_3, X_4)^2$. Denote $X_{ij}$ as the segment connecting $X_i$ and $X_j$, for $i, j = 1,2,3,4$ and $i \neq j$. The problem is reduced to find the expression of a convex

[2] this schematic view can be generalized to any pair of faces of $C_N$ and $\Omega_{max}$

combination $x = cx_v + (1-c)x_o$, where $x_v \in X_{12} \in \mathrm{Fr}(C_N)$ and $x_o \in X_{34} \in \mathrm{Fr}(\Omega)$, providing the minimal value of $c$.

Without loss of generality, we suppose that the distance from $X_2$ to $X_{34}$ is larger than the distance from $X_1$ to $X_{34}$ or equivalently that the distance from $X_4$ to $X_{12}$ is smaller than the distance from $X_3$ to $X_{12}$.

**Theorem 4.4.** *Under the condition that the distance from $X_2$ to $X_{34}$ is larger than the distance from $X_1$ to $X_{34}$, or the distance from $X_4$ to $X_{12}$ is smaller than the distance from $X_3$ to $X_{12}$, the decomposition of the polytope $X_{1234} = X_{124} \cup X_{234}$, is the result of the minimization of the interpolating coefficient c.*

*Proof.* Without loss of generality, suppose that $x \in X_{234}$. Then $x$ can be decomposed as

$$x = cX_2 + (1-c)x_o$$

where $x_o \in X_{34}$, see Figure 4.3. Another possible decomposition is

$$x = c^{'}x_v^{'} + (1-c^{'})x_o^{'}$$

where $x_o^{'}$ is any point in $X_{34}$ and $x_v^{'}$ is any point in $X_{12}$, see Figure 4.3.

It is clear that if the distance from $X_2$ to $X_{34}$ is larger than the distance from $X_1$ to $X_{34}$ then the distance from $X_2$ to $X_{34}$ is larger than the distance from any point $x_v^{'}$ in $X_{12}$ to $X_{34}$. As a consequence, there exists a point $T \in \overline{X_2, x}$ such that

$$\frac{\left\|x - x_v^{'}\right\|}{\left\|x - x_o^{'}\right\|} = \frac{\|x - T\|}{\|x - x_o\|} \leq \frac{\|x - X_2\|}{\|x - x_o\|}$$

Together with theorem 4.3, one obtains

$$c < c^{'}$$

or $X_{234}$ represent a polyhedral partition of the explicit solution to problem (4.10). Analogously one can prove that $X_{124}$ is polyhedral partition of the explicit solution to problem (4.10). □

Theorem 4.4 states that the minimal value of the interpolating coefficient $c$ is found with the help of the decomposition of the polyhedral $X_{1234}$ as $X_{1234} = X_{124} \cup X_{234}$.

*Remark 4.4.* A singular case where the assumption of the previous case is not fulfilled is represented by the segments $X_{12}$ parallel with $X_{34}$. In this case, any convex combination $x = cx_v + (1-c)x_o$ gives the same value of $c$. Hence the partition may not be unique.

*Remark 4.5.* From theorem 4.4, it is clear that one can subdivide the region $C_N \setminus \Omega_{max}$ into partitions as follows

- For each facet of the maximal admissible set $\Omega_{max}$, one has to find the furthest point on the boundary of the feasible set $C_N$ on the same side of the origin as the facet of $\Omega_{max}$. A polyhedral partition is obtained as the convex hull of that facet of $\Omega_{max}$ and the furthest point in $C_N$. By the bounded polyhedral structure of $C_N$, the existence of such a vertex $C_N$ as the furthest point is guaranteed.
- On the other hand, for each facet of the feasible set $C_N$, one has to find the closest point on the boundary of the set $\Omega_{max}$ on the same side of the origin as the facet of $C_N$ . A polyhedral partition will be in this case the convex hull of that facet of $C_N$ and the closest point in $\Omega_{max}$. In this case again the existence of some vertex $\Omega_{max}$ as the closest point is guaranteed.

*Remark 4.6.* In the $n$ dimensional state space, it may happen that the decomposition of the state space according to the remark 4.5 does not cover the entire set $C_N$, as will be shown in the following example. The feasible outer set $C_N$ and the feasible inner set $\Omega_{max}$ are given by the following vertex representations, displayed in Figure 4.4.

$$C_N = \text{Conv}\left\{ \begin{pmatrix} -4 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix}, \begin{pmatrix} 4 \\ -4 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 4 \\ -4 \end{pmatrix} \right\}$$
$$\Omega_{max} = \text{Conv}\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -0.5 \\ -0.5 \\ -0.5 \end{pmatrix}, \begin{pmatrix} -0.5 \\ 0.5 \\ 0 \end{pmatrix}, \begin{pmatrix} -0.5 \\ -0.5 \\ 0.5 \end{pmatrix} \right\}$$

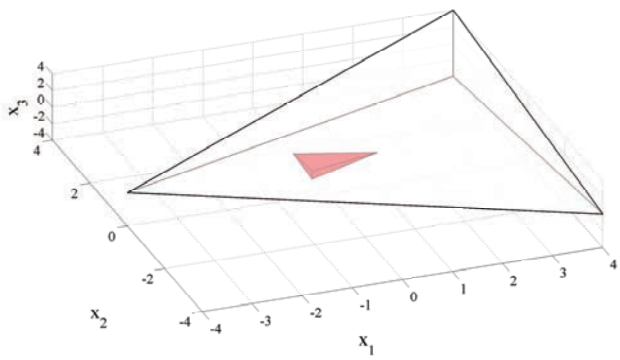



**Fig. 4.4** Graphical illustration for remark 4.6. The white set is the feasible outer set $C_N$. The red set is the feasible inner set $\Omega_{max}$.

By solving the parametric linear program (4.10) in its explicit form, the state space partition is obtained. Figure 4.5 shows two polyhedral partition of the state space partition. The red one is the set $\Omega_{max}$. The blue one is the set, obtained by the convex hull of two points from the inner set $\Omega_{max}$ and two points from the outer set $C_N$.

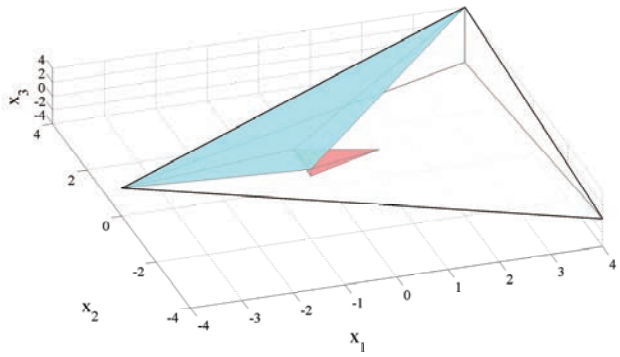



**Fig. 4.5** Graphical illustration for remark 4.6. The partition is obtained by two vertices of the inner set and two vertices of the outer set.

In conclusion, in the $n-$dimensional state space if $x \in C_N \setminus \Omega_{max}$, the smallest value $c$ will be reached when the region $C_N \setminus \Omega_{max}$ is decomposed into polytopes with vertices either on the boundary of $\Omega_{max}$ or on the boundary of $C_N$. These polytopes can be further decomposed into simplices, each formed by $r$ vertices of $C_N$ and $n-r+1$ vertices of $\Omega_{max}$ where $1 \leq r \leq n$. An example of a state space partition is given in Figure 4.6.

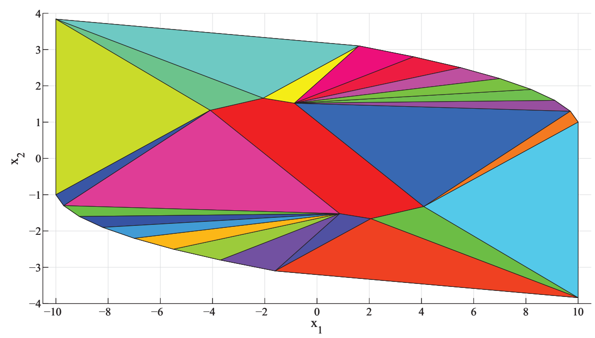



**Fig. 4.6** Simplex based decomposition as an explicit solution of the LP problem (4.10).

### 4.3.3 Explicit solution of the interpolation-based control scheme

Now suppose that $x$ belongs to the simplex formed by $n$ vertices $\{x_1, x_2, \ldots, x_n\}$ of $C_N$ and the vertex $x_o$ of $\Omega_{max}$ (the other cases of $n+1$ vertices distributed in a different manner in between $C_N$ and $\Omega_{max}$ can be treated similarly). In this case, $x$ can be written as a convex combination of $n$ vertices $\{x_1, x_2, \ldots, x_n\}$ and $x_o$, i.e.

$$x = \sum_{i=1}^{n} \alpha_i x_i + \alpha_{i+1} x_o, \tag{4.13}$$

with

$$\sum_{i=1}^{n+1} \alpha_i = 1, \quad \alpha_i \geq 0 \tag{4.14}$$

For a given $x \in C_N \setminus \Omega_{max}$, based on equations (4.13) and (4.14), the interpolating coefficients $\alpha_i$, $\forall i = 1, 2, \ldots, n+1$ are defined uniquely as

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_n & \alpha_{n+1} \end{bmatrix}^T = \begin{bmatrix} x_1 & x_2 & \ldots & x_n & x_o \\ 1 & 1 & \ldots & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ 1 \end{bmatrix} \tag{4.15}$$

with an invertible matrix

$$\begin{bmatrix} x_1 & x_2 & \ldots & x_n & x_o \\ 1 & 1 & \ldots & 1 & 1 \end{bmatrix}$$

since a nonempty simplex is formed by $n+1$ linear independent vertices.

On the other hand, from equation (4.7), the state $x$ can also be expressed as

$$x = cx_v + (1-c)x_o$$

with $0 \leq c \leq 1$.

Due to the uniqueness of the combination, it follows that $\alpha_{i+1} = 1 - c$ and

$$x_v = \sum_{i=1}^{n} \frac{\alpha_i}{c} x_i$$

whenever $c \neq 0$, i.e. for all $x \in C_N \setminus \Omega_{max}$.

By applying the vertex control law, one obtains

$$u_v = \sum_{i=1}^{n} \frac{\alpha_i}{c} u_i$$

and

$$u = cu_v + (1-c)u_o = \sum_{i=1}^{n} \alpha_i u_i + \alpha_{n+1} u_o.$$

or in a compact matrix form

$$u = \begin{bmatrix} u_1 \ u_2 \ \dots \ u_n \ u_o \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \\ \alpha_{n+1} \end{bmatrix}$$

Together with equation (4.15), one can obtain a piecewise affine form

$$\begin{aligned} u &= \begin{bmatrix} u_1 \ u_2 \ \dots \ u_n \ u_o \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \dots & x_n & x_o \\ 1 & 1 & \dots & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ 1 \end{bmatrix} \\ &= Lx + v \end{aligned}$$

where the matrix $L \in \mathbb{R}^{m \times n}$ and the vector $v \in \mathbb{R}^m$ are defined as

$$[L \quad v] = \begin{bmatrix} u_1 \ u_2 \ \dots \ u_n \ u_o \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \dots & x_n & x_o \\ 1 & 1 & \dots & 1 & 1 \end{bmatrix}^{-1}$$

Globally over the entire set $C_N \setminus \Omega_{max}$ the controller is an affine state feedback whose gains are obtained simply by linear interpolation of the control values at the vertices of each simplex. It is worth noticing that the generalization of a simplex-based partition can be highly improved from the complexity point of view by merging of the elementary simplex cells found above [44], [50], [81], [85].

This is not surprising as long as in general terms the interpolation based on linear programming is parameterized in terms of the state vector and leads to a multi-parametric optimization problem. The expected result is a decomposition of the state space corresponding to the distribution of the optimal pairs of extreme points (vertices) used in the interpolation process.

**Algorithm 4.2: Interpolation based control - Explicit solution**

---

**Input:** Given the sets $C_N$, $\Omega_{max}$, the optimal feedback controller $K$ over $\Omega_{max}$ and the control values at the vertices of $C_N$.
**Output:** State space partition, the feedback control laws over the partitions of $C_N$.

1. Solve the LP (4.10) by using explicit multi-parametric programming by exploiting the parameterized vertices formulation, see Section 3.3.3. As a result, one obtains the state space partition of $C_N$.
2. Decompose each polyhedral partition of $C_N \setminus \Omega_{max}$ in a sequence of simplices, each formed by $s$ vertices of $C_N$ and $n-s+1$ vertex of $\Omega_{max}$, where $1 \leq s \leq n$. The result is a the state space partition over $C_N \setminus \Omega_{max}$ in the form of simplices $C^{(k)}$.
3. The control law over $\Omega_{max}$ is $u = Kx$.
4. In each simplex $C^{(k)} \subset C_N \setminus \Omega_{max}$ the control law is defined as:

$$u(x) = L_k x + v_k$$

where $L_k \in \mathbb{R}^{m\times n}$ and $v_k \in \mathbb{R}^{m\times 1}$ are defined as

$$\begin{bmatrix} L_k & v_k \end{bmatrix} = \begin{bmatrix} u_1^{(k)} & u_2^{(k)} & \dots & u_{n+1}^{(k)} \end{bmatrix} \begin{bmatrix} x_1^{(k)} & x_2^{(k)} & \dots & x_{n+1}^{(k)} \\ 1 & 1 & \dots & 1 \end{bmatrix}^{-1}$$

with $\left\{x_1^{(k)}, x_2^{(k)}, \dots x_{n+1}^{(k)}\right\}$ are vertices of $C^{(k)}$ that define a full-dimensional simplex and $\left\{u_1^{(k)}, u_2^{(k)}, \dots u_{n+1}^{(k)}\right\}$ are the corresponding control values at vertices $\left\{x_1^{(k)}, x_2^{(k)}, \dots x_{n+1}^{(k)}\right\}$.

*Remark 4.7.* Based on remark 4.4, it is worth noticing that by using explicit multi-parametric programming, the vertices of the state space partition of $C_N \setminus \Omega_{max}$ might not be the vertices of $C_N$ or $\Omega_{max}$, which might happen for example when some facet of $C_N$ is parallel with a facet of $\Omega_{max}$.

*Remark 4.8.* It can be observed that algorithm 4.2 uses only the information about the state space partition of the explicit solution of the LP problem (4.10). The explicit form of $c$, $r_v$ and $r_o$ as a piecewise affine function of the state is not used.

The sensitive part of algorithm 4.2 is step 2. It is clear that the above simplex-based partition over $C_N \setminus \Omega_{max}$ might be very complex. Also the fact that for all facets of the inner invariant set $\Omega_{max}$, the local controller is in the form $u = Kx$ is not exploited. In addition, as practice usually shows, for each facet of the outer controlled invariant set $C_N$, the vertex controller is usually constant. In these cases, the complexity of the explicit piecewise affine solution of a multi-parametric optimization problem might be reduced as follows.

Consider the case when the state space partition $P^{(k)}$ of $C_N \setminus \Omega_{max}$ is formed by one vertex $x_v$ of $C_N$ and one facet $F_o$ of $\Omega_{max}$. Note that based on remark 4.5 such a partition always exists as an explicit solution to the LP problem (4.10). For all $x \in P^{(k)}$ it follows that

$$x = cx_v + (1-c)x_o = cx_v + r_o$$

with $x_o \in F_o$ and $r_o = (1-c)x_o$.

Let $u_v \in \mathbb{R}^m$ be the control value at the vertex $x_v$ and denote the explicit solution of $c$ and $r_o$ to the LP problem (4.10) for all $x \in P^{(k)}$ as

$$\begin{cases} c = F_k^{(c)} x + g_k^{(c)} \\ r_o = F_k^{(o)} x + g_k^{(o)} \end{cases} \tag{4.16}$$

with $F_k^{(c)} \in \mathbb{R}^n$, $g_k^{(c)} \in \mathbb{R}$ and $F_k^{(o)} \in \mathbb{R}^{n \times n}$, $g_k^{(o)} \in \mathbb{R}^n$. The control value for $x \in P^{(k)}$ is computed as

$$u = cu_v + (1-c)Kx_o = cu_v + Kr_o \tag{4.17}$$

By substituting equation (4.16) into equation (4.17), one obtains

$$u = u_v \left( F_k^{(c)} x + g_k^{(c)} \right) + K \left( F_k^{(o)} x + g_k^{(o)} \right)$$

or equivalently

$$u = \left( u_v F_k^{(c)} + K F_k^{(o)} \right) x + \left( u_v g_k^{(c)} + K g_k^{(o)} \right) \tag{4.18}$$

The fact that the control value is a piecewise affine (PWA) function of state is confirmed. Clearly, the complexity of the explicit solution with the control law (4.18) is lower than the complexity of the explicit solution with the simplex based partition, since one does not have to divide up the facets of $\Omega_{max}$ (and facets of $C_N$, in the case when the vertex control for such facets is constant) into a set of simplices.

### *4.3.4 Interpolation based on linear programming - Qualitative analysis*

Theorem 4.5 below shows the Lipschitz continuity of the control law based on linear programming (4.7), (4.8), (4.10).

**Theorem 4.5.** *Consider the control law resulting from the interpolation based on linear programming (4.7), (4.8), (4.10). This control is Lipschitz with Lipschitz constant* $M = \max_k \|L_k\|$*, where* $k$ *ranges over the set of indices of partitions and* $\|\cdot\|$ *denotes the Euclidean norm.*

*Proof.* For any two points $x_A$ and $x_B$ in $P_N$, there exist $r+1$ points $x_0, x_1, \ldots, x_r$ that lie on the segment, connecting $x_A$ and $x_B$, and such that $x_A = x_0$, $x_B = x_r$ and

$$(x_{k-1}, x_k) = \overline{x_A, x_B} \bigcap \mathrm{Fr}(P^{(i)})$$

(the intersection between the line connecting $x_A, x_B$ and the boundary of some partition $P^{(i)}$, see Figure 4.7).

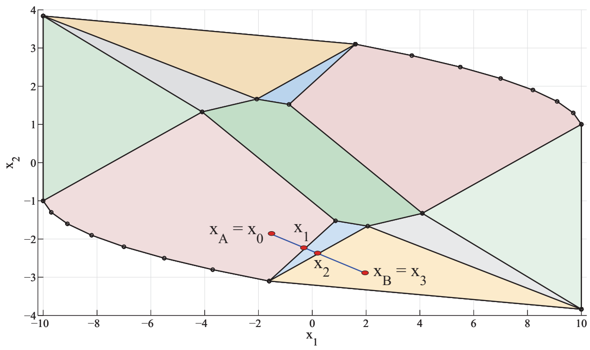



**Fig. 4.7** Graphical illustration of the construction related to Theorem 4.5.

Due to the continuity property of the control law, one has

$$\begin{aligned}
&\|(L_A x_A + v_A) - (L_B x_B + v_B)\| \\
&= \|(L_0 x_0 + v_0) - (L_0 x_1 + v_0) + (L_1 x_1 + v_1) - \ldots - (L_r x_r + v_r)\| \\
&= \|L_0 x_0 - L_0 x_1 + L_1 x_1 - \ldots - L_r x_r\| \\
&\leq \sum_{k=1}^{r} \|L_i(x_k - x_{k-1})\| \leq \sum_{k=1}^{r} \|L_k\| \, \|(x_k - x_{k-1})\| \\
&\leq \max_k\{\|L_k\|\} \sum_{k=1}^{r} \|(x_k - x_{k-1})\| = M\|x_A - x_B\|
\end{aligned}$$

where the last equality holds, since the points $x_i$ with $i = 1, 2, \ldots, r$ are aligned. □

Theorem 4.5 states that the interpolating controller (4.7), (4.8), (4.10) is a Lipschitz continuous function of state, which is a strong form of *uniform continuity* for function.

*Example 4.1.* Consider the following discrete-time linear time-invariant system

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.3 \end{bmatrix} u(k) \tag{4.19}$$

The constraints are

$$-10 \leq x_1(k) \leq 10, \quad -5 \leq x_2(k) \leq 5, \quad -1 \leq u(k) \leq 1 \tag{4.20}$$

Using linear quadratic (LQ) local control with weighting matrices $Q = I$ and $R = 1$ the local feedback gain is obtained

$$K = [-0.5609 \quad -0.9758] \tag{4.21}$$

The invariant set $\Omega_{max}$ and the controlled invariant set $C_N$ with $N = 14$ are shown in Figure 4.1. Note that $C_{14} = C_{15}$. In this case $C_{14}$ is the maximal controlled invariant set. $\Omega_{max}$ is presented in minimal normalized half-space representation as

$$\Omega_{max} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.1627 & -0.9867 \\ -0.1627 & 0.9867 \\ -0.1159 & -0.9933 \\ 0.1159 & 0.9933 \\ -0.4983 & -0.8670 \\ 0.4983 & 0.8670 \end{bmatrix} x \leq \begin{bmatrix} 1.9746 \\ 1.9746 \\ 1.4115 \\ 1.4115 \\ 0.8884 \\ 0.8884 \end{bmatrix} \right\} \tag{4.22}$$

The set $C_N$ can be presented in vertex representation by a set vertices of $C_N$, given by the matrix $V(C_N)$

$$V(C_N) = [V_1 \quad -V_1] \tag{4.23}$$

$$V_1 = \begin{bmatrix} 10.0000 & 9.7000 & 9.1000 & 8.2000 & 7.0000 & 5.5000 & 3.7000 & 1.6027 & -10.0000 \\ 1.0000 & 1.3000 & 1.6000 & 1.9000 & 2.2000 & 2.5000 & 2.8000 & 3.0996 & 3.8368 \end{bmatrix}$$

and the corresponding control values at the vertices of $C_N$

$$U_v = [U_1 \quad -U_1] \tag{4.24}$$

$$U_1 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}$$

Using algorithm 4.2, the state space partition is obtained in Figure 4.6. Merging the regions with identical control laws, the reduced state space partition is obtained in Figure 4.8. This figure also presents different state trajectories of the closed-loop system for different initial conditions.

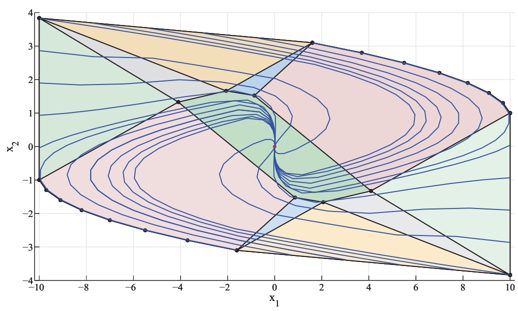
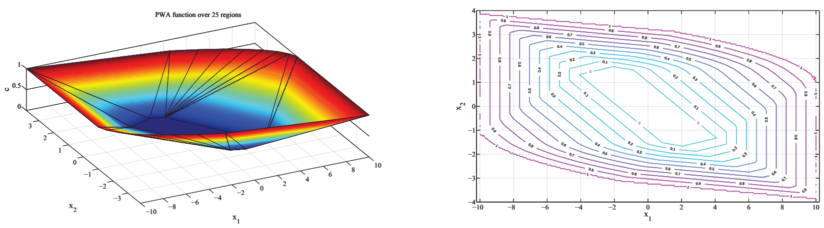



**Fig. 4.8** State space partition and different state trajectories for example 4.1 using algorithm 4.2. Number of regions $N_r = 11$.

Figure 4.9(a) shows the Lyapunov function as a piecewise affine function of state. It is well known[3] that the level sets of the Lyapunov function for vertex control are simply obtained by scaling the boundary of the set $C_N$. For the interpolation based control method (4.7), (4.8), (4.10), the level sets of the Lyapunov function $V(x) = c^*$ depicted in Figure 4.9(b) have a more complicated form and generally are not parallel to the boundary of $C_N$. From Figure 4.9, it can be observed that the Lyapunov level sets $V(x) = c^*$ have the outer set $C_N$ as an external level set (for $c^* = 1$). The inner level sets change the polytopic shape in order to approach the boundary of the inner set $\Omega_{max}$.




(a) Lyapunov function $c^*$ (b) Lyapunov level curves

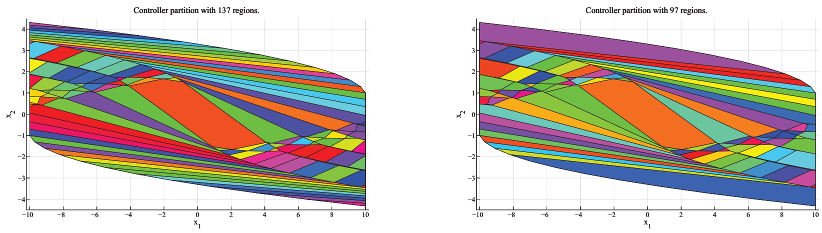
**Fig. 4.9** Lyapunov function and Lyapunov level curves for the interpolation based control method for example 4.1.

The control law over the state space partition is

[3] see Section 3.4

$$u(k) = \begin{cases} -1 & \text{if} \begin{bmatrix} 0.45 & 0.89 \\ 0.24 & 0.97 \\ 0.16 & 0.99 \\ -0.55 & 0.84 \\ 0.14 & 0.99 \\ -0.50 & -0.87 \\ 0.20 & 0.98 \\ 0.32 & 0.95 \\ 0.37 & -0.93 \\ 0.70 & 0.71 \end{bmatrix} x(k) \le \begin{bmatrix} 5.50 \\ 3.83 \\ 3.37 \\ 1.75 \\ 3.30 \\ -0.89 \\ 3.53 \\ 4.40 \\ 2.73 \\ 7.78 \end{bmatrix} \\ -0.38x_1(k)+0.59x_2(k)-2.23 & \text{if} \begin{bmatrix} 0.54 & -0.84 \\ -0.37 & 0.93 \\ -0.12 & -0.99 \end{bmatrix} x(k) \le \begin{bmatrix} -1.75 \\ 2.30 \\ -1.41 \end{bmatrix} \\ -0.02x_1(k)-0.32x_2(k)+0.02 & \text{if} \begin{bmatrix} 0.37 & -0.93 \\ 0.06 & 1.00 \\ -0.26 & -0.96 \end{bmatrix} x(k) \le \begin{bmatrix} -2.30 \\ 3.20 \\ -1.06 \end{bmatrix} \\ -0.43x_1(k)-1.80x_2(k)+1.65 & \text{if} \begin{bmatrix} 0.16 & -0.99 \\ 0.26 & 0.96 \\ -0.39 & -0.92 \end{bmatrix} x(k) \le \begin{bmatrix} -1.97 \\ 1.06 \\ 0.38 \end{bmatrix} \\ 0.16x_1(k)-0.41x_2(k)+2.21 & \text{if} \begin{bmatrix} 0.39 & 0.92 \\ -1.00 & 0 \\ 0.37 & -0.93 \end{bmatrix} x(k) \le \begin{bmatrix} -0.38 \\ 10.00 \\ -2.73 \end{bmatrix} \\ 1 & \text{if} \begin{bmatrix} -0.14 & -0.99 \\ -0.37 & 0.93 \\ -0.24 & -0.97 \\ -0.71 & -0.71 \\ -0.45 & -0.89 \\ -0.32 & -0.95 \\ -0.20 & -0.98 \\ -0.16 & -0.99 \\ 0.50 & 0.87 \\ 0.54 & -0.84 \end{bmatrix} x(k) \le \begin{bmatrix} 3.30 \\ 2.73 \\ 3.83 \\ 7.78 \\ 5.50 \\ 4.40 \\ 3.53 \\ 3.37 \\ -0.89 \\ 1.75 \end{bmatrix} \\ -0.38x_1(k)+0.59x_2(k)+2.23 & \text{if} \begin{bmatrix} 0.12 & 0.99 \\ 0.37 & -0.93 \\ -0.54 & 0.84 \end{bmatrix} x(k) \le \begin{bmatrix} -1.41 \\ 2.30 \\ -1.75 \end{bmatrix} \\ -0.02x_1(k)-0.32x_2(k)-0.02 & \text{if} \begin{bmatrix} 0.26 & 0.96 \\ -0.06 & -1.00 \\ -0.37 & 0.93 \end{bmatrix} x(k) \le \begin{bmatrix} -1.06 \\ 3.20 \\ -2.30 \end{bmatrix} \\ -0.43x_1(k)-1.80x_2(k)-1.65 & \text{if} \begin{bmatrix} 0.39 & 0.92 \\ -0.26 & -0.96 \\ -0.16 & 0.97 \end{bmatrix} x(k) \le \begin{bmatrix} 0.38 \\ 1.06 \\ -1.98 \end{bmatrix} \\ 0.16x_1(k)-0.41x_2(k)-2.21 & \text{if} \begin{bmatrix} 1.00 & 0 \\ -0.37 & 0.93 \\ -0.39 & -0.92 \end{bmatrix} x(k) \le \begin{bmatrix} 10.00 \\ -2.73 \\ -0.38 \end{bmatrix} \\ -0.56x_1(k)-0.98x_2(k) & \text{if} \begin{bmatrix} 0.16 & -0.99 \\ -0.16 & 0.99 \\ -0.12 & -0.99 \\ 0.12 & 0.99 \\ -0.50 & -0.87 \\ 0.50 & 0.87 \end{bmatrix} x(k) \le \begin{bmatrix} 1.97 \\ 1.97 \\ 1.41 \\ 1.41 \\ 0.89 \\ 0.89 \end{bmatrix} \end{cases} \tag{4.25}$$

In view of comparison, consider the explicit model predictive control method with a prediction horizon of 14 steps, Figure 4.10(a) presents the state space partition with a number of regions $N_r = 137$. Merging the polyhedral regions with an identical piecewise affine control function, the reduced state space partition is obtained in Figure 4.10(b).




(a) Before merging (b) After merging

**Fig. 4.10** State space partition before and after merging for example 4.1 using the explicit model predictive control method. Before merging, number of regions $N_r = 137$. After merging $N_r = 97$.

The comparison of the interpolation based control method and the explicit MPC in terms of the number of regions before and after merging is given in Table 4.1.

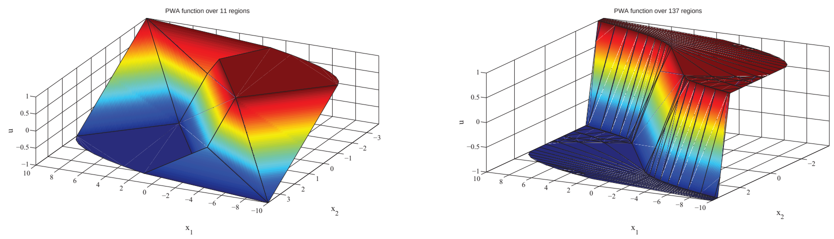
**Table 4.1** Number of regions for the interpolation based control method versus the explicit MPC for example 4.1.

| | Before Merging | After Merging |
|---|---|---|
| Interpolation based control | 25 | 11 |
| Explicit MPC | 137 | 97 |

Figure 4.11(a) and Figure 4.11(b) show the control value as a piecewise affine function of state with the interpolation based control method and the MPC method, respectively.

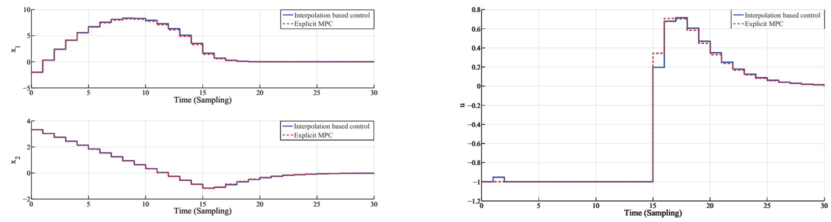
For the initial condition $x(0) = [-2.0000 \quad 3.3284]^T$, Figure 4.12(a) and Figure 4.12(b) show the results of a time-domain simulation for these two control laws.

As a final analysis element, Figure 4.13 presents the interpolating coefficient $c^*(k)$ as a function of time. As expected this function is positive and non-increasing. It is interesting to note that $c^*(k) = 0$, for all $k \geq 15$ implying that from time instant $k = 15$, the state of the closed-loop system is in the invariant set $\Omega_{max}$, and as consequence *optimal* in the MPC cost function terms. The monotonic decrease and the positivity confirms the Lyapunov interpretation given in the present section.




(a) Interpolation based control (b) MPC method

**Fig. 4.11** Control value as a piecewise affine function of state for example 4.1 with the interpolation based control method and the MPC method.




(a) State trajectory (b) Input trajectory

**Fig. 4.12** State and input trajectory for example 4.1. The dashed red curve is obtained by using the explicit MPC method and the solid blue curve is obtained by using the interpolation based control method.

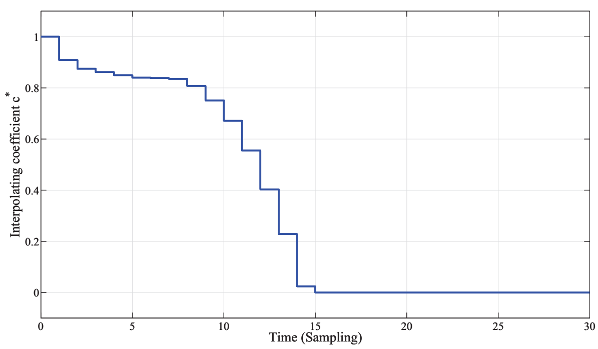



**Fig. 4.13** Interpolating coefficient $c^*$ as a function of time example 4.1.

## 4.4 Performance improvement for the interpolation based control

The interpolation based control method in Sections 4.2 and Section 4.3 can be seen as an approximation of model predictive control, which in the last decade has received significant attention in the control community [62], [83], [140], [131], [19], [66]. From this point of view, it is worthwhile to obtain an interpolation based controller with some given level of accuracy in terms of performance compared with the optimal MPC one. Naturally, the approximation error can be a measure of the level of accuracy. The methods of computing bounds on the approximation error is by now well known in the literature, see for example [62], [19] or [140].

Obviously, the simplest way of improving the contraction factor of the interpolation based control scheme is to use the intermediate s-step controlled invariant sets $C_s$ with $1 \leq s < N$. Then there will be not only one level of interpolation but two or virtually *any* number of interpolation as necessary from the performance point of view. For simplicity of the presentation, we provide in the following a brief study of the case when one intermediate controlled invariant set $C_s$ will be used. Let this set $C_s$ be polyhedral in the form

$$C_s = \{x \in \mathbb{R}^n : F_s x \leq g_s\} \tag{4.26}$$

and satisfying $\Omega_{max} \subset C_s \subset C_N$.

*Remark 4.9.* It has to be noted however that, the expected increase in performance comes at the price of complexity as long as this intermediate set needs to be stored along with its vertex controller.

The vertex controller can be applied for the polyhedral set $C_s$, since $C_s$ is controlled invariant. For further use, the vertex control law applied for the set $C_s$ is denoted as $u_s$.

Using the same philosophy as in Section 4.2, the state $x$ will be decomposed as follows

1. If $x \in C_N$ and $x \notin C_s$, then $x$ is decomposed as

$$x = c_1 x_v + (1 - c_1) x_s \tag{4.27}$$

with $x_v \in C_N$, $x_s \in C_s$ and $0 \leq c_1 \leq 1$. The corresponding control action is then computed as

$$u = c_1 u_v + (1 - c_1) u_s \tag{4.28}$$

2. Else $x \in C_s$ is decomposed as

$$x = c_2 x_s + (1 - c_2) x_o \tag{4.29}$$

with $x_s \in C_s$, $x_o \in \Omega_{max}$ and $0 \leq c_2 \leq 1$. The control action is computed then as

$$x = c_2 u_s + (1 - c_2) u_o \tag{4.30}$$

Depending on the value of $x$, at each time instant, either the interpolating coefficient $c_1$ or $c_2$ is minimized in order to be as close as possible to the optimal controller. This can be done by solving the following (in a parallelized manner) nonlinear optimization problem

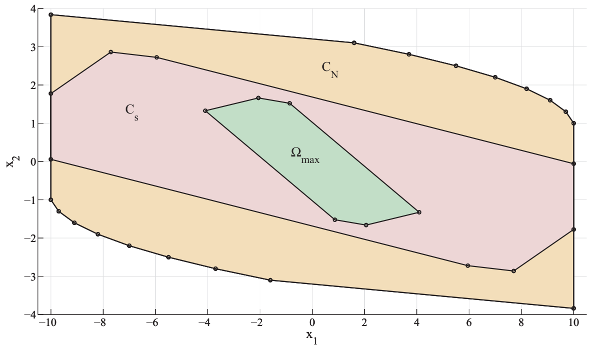



**Fig. 4.14** Two-level interpolation for improving the performance.

1. If $x \in C_N \setminus C_s$

$$c_1^* = \min_{x_v, x_s, c_1} \{c_1\} \tag{4.31}$$

subject to

$$\begin{cases} F_N x_v \le g_N \\ F_s x_s \le g_s \\ c_1 x_v + (1 - c_1) x_s = x \\ 0 \le c_1 \le 1 \end{cases}$$

2. Else $x \in C_s$

$$c_2^* = \min_{x_s, x_o, c_2} \{c_2\} \tag{4.32}$$

subject to

$$\begin{cases} F_s x_s \le g_s \\ F_o x_o \le g_o \\ c_2 x_s + (1 - c_2) x_o = x \\ 0 \le c_2 \le 1 \end{cases}$$

or by changing variables $r_v = c_1 x_v$ and $r_s = c_2 x_s$, the nonlinear optimization problems (4.31) and (4.32) can be transformed in the LP problems as

1. If $x \in C_N \setminus C_s$

$$c_1^* = \min_{r_v, c_1}\{c_1\} \tag{4.33}$$

subject to

$$\begin{cases} F_N r_v \leq c_1 g_N \\ F_s(x - r_v) \leq (1 - c_1) g_s \\ 0 \leq c_1 \leq 1 \end{cases}$$

2. Else $x \in C_s$

$$c_2^* = \min_{r_s, c_2}\{c_2\} \tag{4.34}$$

subject to

$$\begin{cases} F_s r_s \leq c_2 g_s \\ F_o(x - r_s) \leq (1 - c_2) g_o \\ 0 \leq c_2 \leq 1 \end{cases}$$

The following theorem shows recursive feasibility and asymptotic stability of the interpolation based control (4.27), (4.28), (4.29), (4.30), (4.33), (4.34)

**Theorem 4.6.** *The control law using interpolation based on the solution of the LP problems (4.33), (4.34) guarantees recursive feasibility and asymptotic stability of the closed loop system for all initial states* $x(0) \in C_N$.

*Proof.* The proof of the theorem is omitted here, since it follows the same steps as those presented in the feasibility proof 4.1 and the stability proof 4.2 in Section 4.2. □

*Remark 4.10.* Clearly, instead of the second level of interpolation, the MPC approach can be applied for all state inside the set $C_s$. This has very practical consequences in applications, since it is well known that the main issue of the MPC method for the nominal discrete-time linear time-invariant systems is the trade-off between the overall complexity (computational cost) and the size of the domain of attraction. If the prediction horizon of the MPC method is short then the domain of attraction is small. If the prediction horizon is long then the computational cost may be very burdensome for the available hardware. Here the MPC method with the short prediction horizon (equal to one, strictly speaking) is used for the performance and then for enlarging the domain of attraction, the interpolation based on linear programming (4.33) is used. In this way one can achieve the performance and the domain of attraction with a relatively small computational cost.

For the continuity of the control law (4.27), (4.28), (4.29), (4.30), (4.33), (4.34), the following theorem holds

**Theorem 4.7.** *The control law from the interpolation based on linear programming (4.33), (4.34) can be represented as a continuous function of the state.*

*Proof.* Clearly the discontinuity of the control law may arise only on the boundary of the intermediate set $C_s$, since for all $x \in C_N \setminus C_s$ or for all $x \in C_s$, the interpolation based controller (4.27), (4.28), (4.33) or (4.29), (4.30), (4.34) is continuous

It is clear that for all $x \in \text{Fr}(C_s)$, the LP problems (4.33), (4.34) have a trivial solution

$$c_1^* = 0, \quad c_2^* = 1$$

Hence the control action for the interpolation based on 4.27), (4.28), (4.33) is $u = u_s$ and the control action for the interpolation based on (4.29), (4.30), (4.34) is $u = u_s$. These control actions coincide and turn out to be the vertex controller for the set $C_s$. Hence the continuity of the control law is guaranteed. □

*Remark 4.11.* It is interesting to note that by using $N-1$ intermediate sets $C_i$ together with the sets $C_N$ and $\Omega_{max}$, a continuous minimum-time controller is obtained, i.e. a controller that steers all state $x \in C_N$ in $\Omega_{max}$ in no more than $N$ steps.

Concerning the explicit solution of the interpolation based controller using the intermediate set $C_s$, with the same argument as in Section 4.3, it can be concluded that

- If $x \in C_N \setminus C_s$ (or $x \in C_s \setminus \Omega_{max}$), the smallest value $c_1$ (or $c_2$) will be reached when the region $C_N \setminus C_s$ (or $C_S \setminus \Omega_{max}$) is decomposed into partitions in form of simplices with vertices either on the boundary of $C_N$ or on the boundary of $C_s$ (or on the boundary of $C_s$ or on the boundary of $\Omega_{max}$). The control law in each partition is piecewise affine function of state whose gains are obtained by interpolation of control values at the vertices of the simplex.
- If $x \in \Omega_{max}$, then the control law is the optimal unconstrained controller.

*Example 4.2.* Consider again the discrete-time linear time-invariant system in example 4.1 with the same state and control constraints. The local feedback controller is the same as in example 4.1

$$K = [-0.5609 \quad -0.9758] \tag{4.35}$$

With the local controller $K$, the sets $\Omega_{max}$, $C_s$ with $s = 4$ and $C_N$ with $N = 14$ is constructed. The representation of the sets $\Omega_{max}$ and $C_N$ can be found in example 4.1. The set of vertices $V_s$ of the polytope $C_s$ is

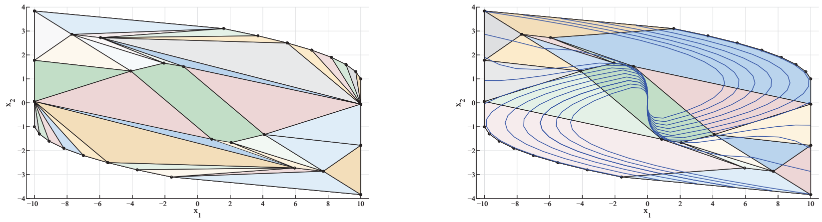
$$V_s = \begin{bmatrix} 10.00 & -5.95 & -7.71 & -10.00 & -10.00 & 5.95 & 7.71 & 10.00 \\ -0.06 & 2.72 & 2.86 & 1.78 & 0.06 & -2.72 & -2.86 & -1.78 \end{bmatrix} \tag{4.36}$$

and the set of the corresponding control actions at the vertices $V_s$ is

$$U_s = \begin{bmatrix} -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{4.37}$$

Using explicit multi-parametric linear programming, the state space partition is obtained in Figure 4.15(a). Merging the regions with identical control laws, the reduced state space partition is obtained in Figure 4.15(b). This figure also shows state trajectories of the closed-loop system for different initial conditions.

Figure 4.16 shows the control value as a piecewise affine function of the state with two-level interpolation.



(a) Before merging (b) After merging

**Fig. 4.15** State space partition before and after merging for example 4.2. Before merging, the number of regions is $N_r = 37$. After merging, $N_r = 19$.

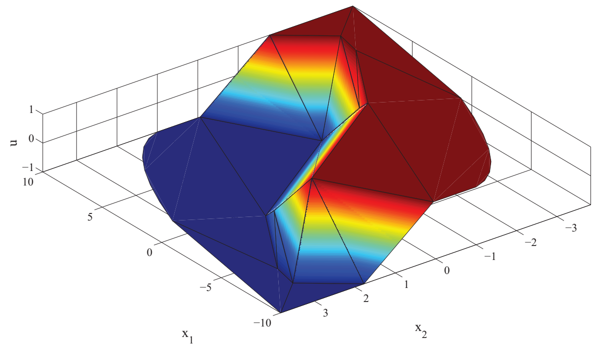


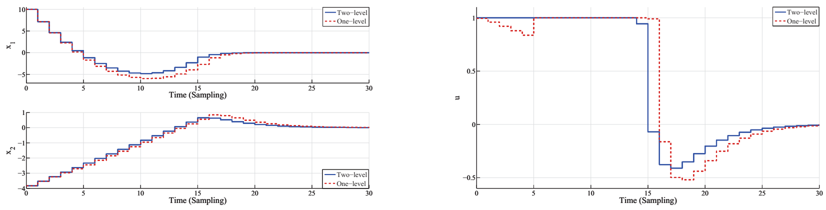

**Fig. 4.16** Control value as a piecewise affine function of state for example 4.2 using two-level interpolation.

For the initial condition $x(0) = [9.9800 \quad -3.8291]^T$, Figure 4.17 shows the results of a time-domain simulation. The two curves correspond to the one-level and two-level interpolation based control, respectively

Figure 4.18 presents the interpolating coefficients as a function of time. As expected $c_1^*$ and $c_2^*$ are positive and non-increasing. It is also interesting to note that $c_1^*(k) = 0$ for all $k \geq 10$, indicating that $x$ is inside $C_s$ and $c^*(k) = 0$ for all $k \geq 14$, indicating that state $x$ is inside $\Omega_{max}$.

## 4.5 Interpolation based on quadratic programming

The interpolation based control method in Section 4.2 and Section 4.3 makes use of linear programming, which is extremely simple. However, the main issue regarding the implementation of the algorithm 4.1 is the non-uniqueness of the solution. Multiple optima are undesirable, as they might lead to a fast switching between the




(a) State trajectory (b) Input trajectory

**Fig. 4.17** State and input trajectories for example 4.2. The dashed red curve is obtained by using the one-level interpolation based control, and the solid blue curve is obtained by using the two-level interpolation based control.

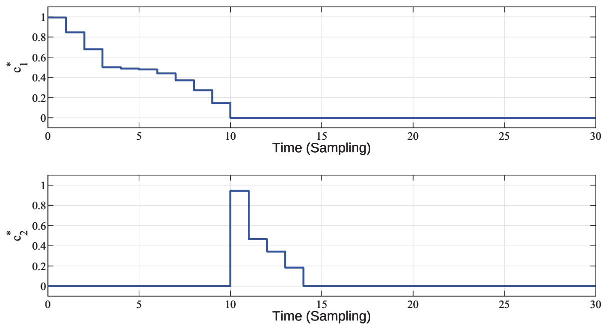



**Fig. 4.18** Interpolating coefficients as a function of time for example 4.2.

different optimal control actions when the LP problem (4.10) is solved on-line. Traditionally model predictive control has been formulated using a quadratic criterion [100]. Hence, also in interpolation based control it is worthwhile to investigate the use of quadratic programming.

Before introducing a QP formulation let us note that the idea of using QP for interpolation control is not new. In [11], [132], Lyapunov theory is used to compute an upper bound of the infinite horizon cost function

$$J = \sum_{k=0}^{\infty} \left\{ x(k)^T Q x(k) + u(k)^T R u(k) \right\} \tag{4.38}$$

where $Q \succeq 0$ and $R \succ 0$ are the state and input weighting matrices. At each time instant, the algorithm in [11] uses an on-line decomposition of the current state, with each component lying in a separate invariant set, after which the corresponding controller is applied to each component separately in order to calculate the control action. Polytopes are employed as candidate invariant sets. Hence, the on-line optimization problem can be formulated as a QP problem. The approach taken in this section follows ideas originally proposed in [11], [132]. In this setting we provide a QP based solution to the constrained control problem.

This section begins with a brief summary of the work of Bacic et al. [11], [132]. For this purpose, it is assumed that using established results in control theory (LQR, LMI based, etc), one obtains a set of unconstrained asymptotically stabilizing feedback controllers $u(k) = K_i x(k)$, $i = 1,2,\ldots,r$ such that the corresponding invariant sets $\Omega_i \subseteq X$

$$\Omega_i = \left\{ x \in \mathbb{R}^n : F_o^{(i)} x \leq g_o^{(i)} \right\} \tag{4.39}$$

are non-empty for $i = 1,2,\ldots,r$.

Denote $\Omega$ as a convex hull of the sets $\Omega_i$, $i = 1,2,\ldots,r$. It follows that $\Omega \subseteq X$, since $\Omega_i \subseteq X$ for all $i = 1,2,\ldots,r$. Any state $x(k) \in \Omega$ can be decomposed as follows

$$x(k) = \lambda_1 \widehat{x}_1 + \lambda_2 \widehat{x}_2 + \ldots + \lambda_r \widehat{x}_r \tag{4.40}$$

where $\widehat{x}_1 \in \Omega_i$ for all $i = 1,2,\ldots,r$ and

$$\sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \geq 0$$

With a slight abuse of notation, denote $x_i = \lambda_i \widehat{x}_i$. Since $\widehat{x}_i \in \Omega_i$, it follows that $x_i \in \lambda_i \Omega_i$ or equivalently

$$F_o^{(i)} x_i \leq \lambda_i g_o^{(i)} \tag{4.41}$$

for all $i = 1,2,\ldots,r$.

Consider the following control law

$$u(k) = \sum_{i=1}^{r} \lambda_i K_i \widehat{x}_i = \sum_{i=1}^{r} K_i x_i \tag{4.42}$$

where $u(k) = K_i x_i(k)$ is the control law, associated to the invariant construction of the set $\Omega_i$. One has

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) = A \sum_{i=1}^{r} x_i(k) + B \sum_{i=1}^{r} K_i x_i(k) \\ &= \sum_{i=1}^{r} (A + BK_i) x_i(k) \end{aligned}$$

or

$$x(k+1) = \sum_{i=1}^{r} x_i(k+1) \tag{4.43}$$

where $x_i(k+1) = A_{ci} x_i(k)$ and $A_{ci} = A + BK_i$.

Denote a vector $z \in \mathbb{R}^{rn}$ as follows

$$z = \begin{bmatrix} x_1^T & x_2^T & \ldots & x_r^T \end{bmatrix}^T$$

From equation (4.43), it follows that

$$z(k+1) = \Phi z(k) \tag{4.44}$$

where

$$\Phi = \begin{bmatrix} A_{c1} & 0 & \dots & 0 \\ 0 & A_{c2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{cr} \end{bmatrix}$$

With the given state and control weighting matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$, consider the following quadratic function

$$V(z) = z^T P z \tag{4.45}$$

where matrix $P \in \mathbb{R}^{rn\times rn}$, $P \succ 0$ is chosen to satisfy

$$V(z(k+1)) - V(z(k)) \leq -x(k)^T Q x(k) - u(k)^T R u(k) \tag{4.46}$$

From equation (4.44), the left hand side of inequality (4.46) can be rewritten as

$$V(z(k+1)) - V(z(k)) = z(k)^T (\Phi^T P \Phi - P) z(k) \tag{4.47}$$

The right hand side of inequality (4.46) can be rewritten as

$$-x(k)^T Q x(k) - u(k)^T R u(k) = z(k)^T (Q_1 + R_1) z(k) \tag{4.48}$$

with

$$Q_1 = -\begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} Q \begin{bmatrix} I \ I \ \dots \ I \end{bmatrix}, \; R_1 = -\begin{bmatrix} K_1^T \\ K_2^T \\ \vdots \\ K_r^T \end{bmatrix} R \begin{bmatrix} K_1 \ K_2 \ \dots \ K_r \end{bmatrix}$$

From equations (4.46), (4.47) and (4.48), one gets

$$\Phi^T P \Phi - P \preceq Q_1 + R_1$$

or by using the Schur complement, one obtains

$$\begin{bmatrix} P + Q_1 + R_1 & A_c^T P \\ P A_c & P \end{bmatrix} \succeq 0 \tag{4.49}$$

Clearly, problem (4.49) is linear with respect to the matrix $P$. This problem is feasible since matrix $\Phi$ has a sub-unitary spectral radius. One way to obtain matrix $P$ is to solve the following LMP problem

$$\min_P \{trace(P)\} \tag{4.50}$$

subject to constraints (4.49).

At each time instant, for a given current state $x$, consider the following optimization problem

$$\min_{x_i,\lambda_i} \begin{bmatrix} x_1^T & x_2^T & \dots & x_r^T \end{bmatrix} P \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \end{bmatrix} \tag{4.51}$$

subject to

$$\begin{cases} F_o^{(i)} x_i \le \lambda_i g_o^{(i)}, \forall i = 1,2,\dots,r \\ \sum\limits_{i=1}^{r} x_i = x \\ \sum\limits_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \ge 0 \end{cases}$$

and implement as input the control action $u = \sum\limits_{i=1}^{r} K_i x_i$.

**Theorem 4.8.** *[11], [132] The control law using interpolation based on the solution of the problem (4.51) guarantees recursive feasibility and the closed loop system is asymptotically stable for all initial states $x(0) \in \Omega$.*

*Proof.* See [11], [132].

Using the approach in [11], [132], it can be observed that, at each time instant we are trying to minimize $x_1, x_2, \dots, x_r$ in the weighted Euclidean norm sense. This is somehow a conflicting task, since

$$x_1 + x_2 + \dots + x_r = x$$

In addition, if the first controller is an optimal controller and play the role of a performance controller, and the remaining controller is used to enlarge the domain of attraction, then one would like to be as close as possible to the first controller, i.e. to the optimal one. This means that in the interpolation scheme (4.40), one would like to have $x_1 = x$ and

$$x_2 = x_3 = \dots = x_r = 0$$

whenever it is possible. And it is not trivial how to do this with the approach in [11], [132].

Below we provide a contribution to this line of research by considering one of the interpolation factors, i.e. control gains to be a performance related one, while the remaining factors play the role of degrees of freedom to enlarge the domain of attraction. This alternative approach can provide the appropriate framework for the constrained control design which builds on the unconstrained controller (generally with high gain) and subsequently need to adjusted them to cope with the constraints and limitations (via interpolation with adequate low gain controllers). From this point of view, in the remaining part of this section we try to build a bridge between the linear interpolation scheme presented in Section 4.2 and the QP based interpolation approaches in [11], [132].

For the given set of state and control weighting matrices $Q_i \in \mathbb{R}^{n\times n}$, $R_i \in \mathbb{R}^{m\times m}$ and $Q_i \succeq 0$, $R_i \succeq 0$, consider the following set of quadratic functions

$$V_i(x_i) = x_i^T P_i x_i, \quad \forall i = 2,3,\ldots,r \tag{4.52}$$

where matrix $P_i \in \mathbb{R}^{n\times n}$ and $P_i \succ 0$ is chosen to satisfy

$$V_i(x_i(k+1)) - V_i(x_i(k)) \leq -x_i(k)^T Q_i x_i(k) - u_i(k)^T R_i u_i(k) \tag{4.53}$$

From inequality (4.53) and since $x_i(k+1) = A_{ci}x_i(k)$, it follows that

$$A_{ci}^T P_i A_{ci} - P_i \preceq -Q_i - K_i^T R_i K_i$$

By using the Schur complement, one obtains

$$\begin{bmatrix} P_i - Q_i - K_i^T R_i K_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} \succeq 0 \tag{4.54}$$

Since matrix $A_{ci}$ has a sub-unitary spectral radius, inequality (4.54) is always feasible. One way to obtain matrix $P_i$ is to solve the following LMI problem

$$\min_{P_i}\{trace(P_i)\} \tag{4.55}$$

subject to constraint (4.54).

Define the vector $z_1 \in \mathbb{R}^{(r-1)(n+1)}$ as follows

$$z_1 = [x_2^T \quad x_3^T \quad \ldots \quad x_r^T \quad \lambda_2 \quad \lambda_3 \quad \ldots \quad \lambda_r]^T$$

With the vector $z_1$, consider the following quadratic function

$$V_1(z_1) = \sum_{i=2}^{r} x_i^T P_i x_i + \sum_{i=2}^{r} \lambda_i^2 \tag{4.56}$$

We underline the fact that the sums are built on indices $\{2,\ldots r\}$, corresponding to the more poorly performing controllers. At each time instant, consider the following optimization problem

$$\min_{z_1} \{V_1(z_1)\} \tag{4.57}$$

subject to the constraints

$$\begin{cases} F_o^{(i)} x_i \leq \lambda_i g_o^{(i)}, \forall i = 1, 2, \dots, r \\ \sum_{i=1}^{r} x_i = x \\ \sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \geq 0 \end{cases}$$

and apply as input the control signal $u = \sum_{i=1}^{r} \{K_i x_i\}$.

**Theorem 4.9.** *The control law based on solving on-line the optimization problem (4.57) guarantees recursive feasibility and asymptotic stability for all initial states* $x(0) \in \Omega$.

*Proof.* Theorem 4.9 makes two important claims, namely the recursive feasibility and the asymptotic stability. These can be treated sequentially.

*Recursive feasibility:* It has to be proved that $F_u u(k) \leq g_u$ and $x(k+1) \in \Omega$ for all $x(k) \in \Omega$. It holds that

$$\begin{aligned} F_u u(k) &= F_u \sum_{i=1}^{r} \lambda_i K_i \widehat{x}_i = \sum_{i=1}^{r} \lambda_i F_u K_i \widehat{x}_i \\ &\leq \sum_{i=1}^{r} \lambda_i g_u = g_u \end{aligned}$$

and

$$x(k+1) = Ax(k) + Bu(k) = \sum_{i=1}^{r} \lambda_i A_{ci} \widehat{x}_i(k)$$

Since $A_{ci}\widehat{x}_i(k) \in \Omega_i \subseteq \Omega$, it follows that $x(k+1) \in \Omega$.

*Asymptotic stability:* Consider the positive function $V_1(z_1)$ as a candidate Lyapunov function. It is clear that, if $x_1^o(k), x_2^o(k), \dots, x_r^o(k)$ and $\lambda_1^o(k), \lambda_2^o(k), \dots, \lambda_r^o(k)$ is the solution of the optimization problem (4.57) at time instant $k$, then

$$x_i(k+1) = A_{ci} x_i^o(k)$$

and

$$\lambda_i(k+1) = \lambda_i^o(k)$$

for all $i = 1, 2, \dots, r$ is a feasible solution to the optimization problem (4.57) at time instant $k+1$. Since at each time instant we are trying to minimize $V(z)$, it follows that

$$V_1(z_1^o(k+1)) \leq V_1(z_1(k+1))$$

therefore

$$V_1(z_1^o(k+1)) - V_1(z_1^o(k)) \leq V_1(z_1(k+1)) - V_1(z_1^o(k))$$

together with inequality (4.53), one obtains

$$V_1(z_1^o(k+1)) - V_1(z_1^o(k)) \leq -\sum_{i=2}^{r} \left(x_i^T Q_i x_i + u_i^T R_i u_i\right)$$

Therefore $V(z)$ is a Lyapunov function and the interpolation based controller assures asymptotic stability for all $x \in \Omega$. □

Clearly, the objective function (4.57) can be written as

$$\min_{z_1}\{z_1^T H z_1\} \tag{4.58}$$

where

$$H = \begin{bmatrix} P_2 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & P_3 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P_r & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

And the constraints of the optimization problem (4.57) can be rewritten as

$$\begin{cases} F_o^{(1)}(x - x_2 - \dots - x_r) \leq (1 - \lambda_2 - \dots - \lambda_r) g_o^{(1)} \\ F_o^{(2)} x_2 \leq \lambda_2 g_o^{(2)} \\ \vdots \\ F_o^{(r)} x_r \leq \lambda_r g_o^{(r)} \\ -\lambda_i \leq 0, \quad \forall i = 2, \dots, r \\ \lambda_2 + \lambda_3 + \dots \lambda_r \leq 1 \end{cases}$$

or, equivalently

$$Gz_1 \leq S + Ex \tag{4.59}$$

where

$$G = \begin{bmatrix} -F_o^{(1)} & -F_o^{(1)} & \dots & -F_o^{(1)} & g_o^{(1)} & g_o^{(1)} & \dots & g_o^{(1)} \\ F_o^{(2)} & 0 & \dots & 0 & -g_o^{(2)} & 0 & \dots & 0 \\ 0 & F_o^{(3)} & \dots & 0 & 0 & -g_o^{(3)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & F_o^{(r)} & 0 & 0 & \dots & -g_o^{(r)} \\ 0 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & -1 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 \end{bmatrix}$$

$$S = \left[ (g_o^{(1)})^T \; 0 \; 0 \dots 0 \; 0 \; 0 \dots 0 \; 1 \right]^T$$

$$E = \left[ -(F_o^{(1)})^T \; 0 \; 0 \dots 0 \; 0 \; 0 \dots 0 \; 0 \right]^T$$

Hence, the optimization problem (4.57) is transformed into the quadratic programming problem (4.58) subject to the linear constraints (4.59).

It is worth noticing that for all $x \in \Omega_1$, the QP problem (4.58) subject to the constraints (4.59) has a trivial solution, namely

$$\begin{cases} x_i = 0, \\ \lambda_i = 0 \end{cases} \quad \forall i = 2, 3, \dots, r$$

Hence $x_1 = x$ and $\lambda_1 = 1$. That means, inside the invariant set $\Omega_1$, the interpolating controller turns out to be the optimal unconstrained controller.

An on-line algorithm for the interpolation based controller via quadratic programming is

**Algorithm 4.3: Interpolation based control via quadratic programming**

---

1. Measure the current state of the system $x(k)$.
2. Solve the QP problem (4.58) subject to the constraints (4.59).
3. Apply the control input (4.42).
4. Wait for the next time instant $k := k+1$.
5. Go to step 1 and repeat.

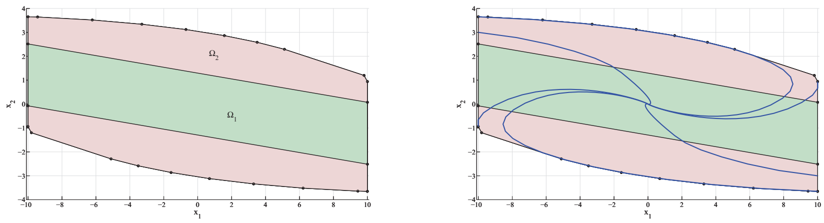
*Remark 4.12.* Note that algorithm 4.3 requires the solution of the QP problem (4.58) of dimension $(r-1)(n+1)$ where $r$ is the number of interpolated controllers and $n$ is the dimension of state. Clearly, solving the QP problem (4.58) can be computationally expensive when the number of interpolated controllers is big. In practice, it is usually enough with $r = 2$ or $r = 3$.

*Example 4.3.* Consider again the discrete-time linear time-invariant system in example (4.1) with the same state and control constraints. Two linear feedback controllers are chosen as

$$\begin{cases} K_1 = [-0.0942 \quad -0.7724] \\ K_2 = [-0.0669 \quad -0.2875] \end{cases} \tag{4.60}$$

The first controller $u(k) = K_1 x(k)$ is a high controller and plays the role of the performance controller, while the second controller $u(k) = K_2 x(k)$ will be used to enlarge the domain of attraction.

Figure 4.19(a) shows the invariant sets $\Omega_1$ and $\Omega_2$ correspond to the controllers $K_1$ and $K_2$ respectively. Figure 4.19(b) shows different state trajectories of the closed loop system for different initial conditions. The state trajectories are obtained by solving on-line quadratic programming problem (4.58) subject to the constraints (4.59).



(a) Feasible invariant sets (b) State trajectories

**Fig. 4.19** Feasible invariant sets and state trajectories of the closed loop system for example 4.3.

The sets $\Omega_1$ and $\Omega_2$ are presented in minimal normalized half-space representation as

$$\Omega_1 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.0000 & 0 \\ -1.0000 & 0 \\ -0.1211 & -0.9926 \\ 0.1211 & 0.9926 \end{bmatrix} x \leq \begin{bmatrix} 10.0000 \\ 10.0000 \\ 1.2851 \\ 1.2851 \end{bmatrix} \right\}$$

and

$$\Omega_2 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.0000 & 0 \\ -1.0000 & 0 \\ -0.2266 & -0.9740 \\ 0.2266 & 0.9740 \\ 0.7948 & 0.6069 \\ -0.7948 & -0.6069 \\ -0.1796 & -0.9837 \\ 0.1796 & 0.9837 \\ -0.1425 & -0.9898 \\ 0.1425 & 0.9898 \\ -0.1117 & -0.9937 \\ 0.1117 & 0.9937 \\ -0.0850 & -0.9964 \\ 0.0850 & 0.9964 \\ -0.0610 & -0.9981 \\ 0.0610 & 0.9981 \\ -0.0386 & -0.9993 \\ 0.0386 & 0.9993 \\ -0.0170 & -0.9999 \\ 0.0170 & 0.9999 \end{bmatrix} x \leq \begin{bmatrix} 10.0000 \\ 10.0000 \\ 3.3878 \\ 3.3878 \\ 8.5177 \\ 8.5177 \\ 3.1696 \\ 3.1696 \\ 3.0552 \\ 3.0552 \\ 3.0182 \\ 3.0182 \\ 3.0449 \\ 3.0449 \\ 3.1299 \\ 3.1299 \\ 3.2732 \\ 3.2732 \\ 3.4795 \\ 3.4795 \end{bmatrix} \right\}$$

With the weighting matrices

$$Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_2 = 1$$

and by solving the LMI problem (4.55), one obtains

$$P_2 = \begin{bmatrix} 5.1917 & 9.9813 \\ 9.9813 & 101.2651 \end{bmatrix} \tag{4.61}$$

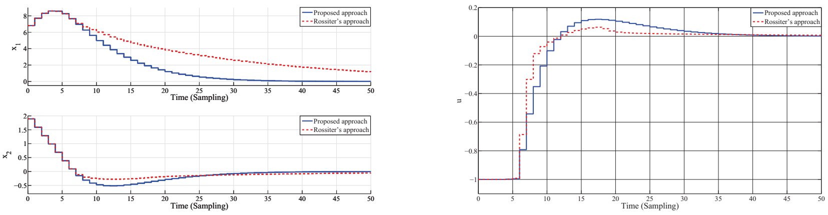
For the initial condition $x(0) = [6.82 \quad 1.889]^T$, Figure 4.20(a) and 4.20(b) present the state and input trajectory of the closed loop system as a function of time. The solid blue line is obtained by solving the QP problem (4.58). The dashed red line is obtained by solving the QP interpolation using algorithm in [132].

For the algorithm in [132], the matrix $P$ in the optimization problem (4.50) is computed as

$$P = \begin{bmatrix} 4.8126 & 2.9389 & 4.5577 & 13.8988 \\ 2.9389 & 7.0130 & 2.2637 & 20.4391 \\ 4.5577 & 2.2637 & 5.1917 & 9.9813 \\ 13.8988 & 20.4391 & 9.9813 & 101.2651 \end{bmatrix}$$

for the following weighting matrices

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1$$




(a) State trajectory (b) Input trajectory

**Fig. 4.20** State and input trajectory of the closed loop system for example 4.3. The solid blue line is obtained by solving the QP problem (4.58). The dashed red line is obtained by using the algorithm in [132].

The interpolating coefficient $\lambda_2$ and the Lyapunov function $V(z)$ as a function of time are depicted in Figure 4.21. As expected $V(z)$ is a positive and non-increasing function.

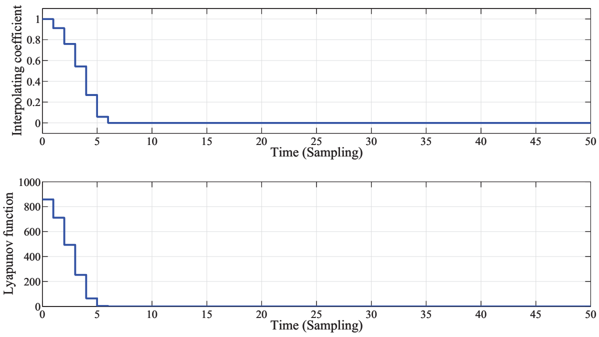



**Fig. 4.21** Interpolating coefficient $\lambda_2(k)$ and Lyapunov function $V(z)$ as a function of time for example 4.3.

## 4.6 An improved interpolation based control method in the presence of actuator saturation

In this section, in order to fully utilize the capability of actuators and guarantee the input constraints satisfaction without handling an unnecessarily complex optimization-based control, a saturation function on the input is considered. Saturation will guarantee that the plant input constraints are satisfied. In our design we exploit the fact that the saturating linear feedback law can be expressed as a convex hull of a group of linear feedback laws according to Hu et al. [59]. Thus, the auxiliary control laws in the convex hull rather than the actual control law will handle the input constraints.

For simplicity, only the single-input single-output system case is considered here, although extensions to the multi-input multi-output system case are straightforward.

Since the saturation function on the input is considered, the system (4.1) becomes

$$x(k+1) = Ax(k) + Bsat(u(k)) \tag{4.62}$$

Clearly, the use of a saturation function is an appropriate choice only for the input constraints (4.2) in a form

$$u(k) \in U, U = \{u \in \mathbb{R} : u_l \le u \le u_u\} \tag{4.63}$$

where $u_l$ and $u_u$ are respectively the lower and upper bounds of input $u$. It is assumed that $u_l$ and $u_u$ are constant with $u_l < 0$ and $u_u > 0$ such that the origin is contained in the interior of $U$. Recall that the state constraints remain the same as in (4.2).

From Lemma 2.1, Section 2.4.1, recall that for a given stabilizing controller $u(k) = Kx(k)$, the saturation function[4] can be expressed as

$$sat(Kx(k)) = \alpha(k)Kx(k) + (1-\alpha(k))Hx(k) \tag{4.64}$$

for all $x$ such that $u_l \le Hx \le u_u$ and with a suitable choice of $0 \le \alpha(k) \le 1$. The vector $H \in \mathbb{R}^n$ can be computed using theorem 2.2. Based on procedure 2.5 in Section 2.4.1, a polyhedral set $\Omega_s^H$ can be computed, with invariance properties with respect to the dynamics

$$x(k+1) = Ax(k) + Bsat(Kx(k)) \tag{4.65}$$

It is assumed that a set of asymptotically stabilizing feedback controllers $K_i \in \mathbb{R}^n$ is available as well as a set of auxiliary vectors $H_i \in \mathbb{R}^n$ for all $i = 1, 2, \dots, r$ such that the corresponding invariant sets $\Omega_s^{H_i} \subseteq X$

$$\Omega_s^{H_i} = \left\{x \in \mathbb{R}^n : F_o^{(i)} x \le g_o^{(i)}\right\} \tag{4.66}$$

are non-empty for $i = 1, 2, \dots, r$.

With a slight abuse of notation, denote $\Omega_s$ as a convex hull of the sets $\Omega_s^{H_i}$. It follows that $\Omega_s \subseteq X$, since $\Omega_s^{H_i} \subseteq X$ for all $i = 1, 2, \dots, r$.

Any state $x(k) \in \Omega_s$ can be decomposed as follows

$$x(k) = \sum_{i=1}^{r} \lambda_i \widehat{x}_i(k) \tag{4.67}$$

where $\widehat{x}_i(k) \in \Omega_s^{H_i}$ for all $i = 1, 2, \dots, r$ and

[4] See Section 2.4.1 for more details.

$$\sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \geq 0$$

Consider the following control law

$$u(k) = \sum_{i=1}^{r} \lambda_i sat(K_i \widehat{x}_i(k)) \tag{4.68}$$

Based on Lemma 2.1, one obtains

$$u(k) = \sum_{i=1}^{r} \lambda_i (\alpha_i(k) K_i + (1-\alpha_i(k)) H_i) \widehat{x}_i(k) \tag{4.69}$$

where $0 \leq \alpha_i \leq 1$ for all $i = 1, 2, \ldots, r$.

Similar with the notation employed in the Section 4.5, we denote $x_i = \lambda_i \widehat{x}_i$. Since $\widehat{x}_i \in \Omega_s^{H_i}$, it follows that $x_i \in \lambda_i \Omega_s^{H_i}$ or

$$F_o^{(i)} x_i \leq \lambda_i g_o^{(i)}, \quad \forall i = 1, 2, \ldots, r \tag{4.70}$$

From equations (4.67) and (4.69), one obtains

$$\begin{cases} x = \sum\limits_{i=1}^{r} x_i \\ u = \sum\limits_{i=1}^{r} (\alpha_i K_i + (1-\alpha_i) H_i) x_i \end{cases} \tag{4.71}$$

As in Section 4.5, the first controller, identified by the high gain $K_1$ will play the role of a performance controller, while the remaining low gain controllers will be used to enlarge the domain of attraction.

With the control input as in the form (4.71), it is clear that $u(k) \in U$, $\forall k \geq 0$. Hence $sat(u(k)) = u(k)$. It follows that

$$\begin{aligned} x(k+1) &= Ax(k) + Bsat(u(k)) = Ax(k) + Bu(k) \\ &= A \sum_{i=1}^{r} x_i(k) + B \sum_{i=1}^{r} (\alpha_i K_i + (1-\alpha_i) H_i) x_i(k) \\ &= \sum_{i=1}^{r} x_i(k+1) \end{aligned}$$

where

$$x_i(k+1) = \{A + B(\alpha_i K_i + (1-\alpha_i) H_i)\} x_i(k)$$

or

$$x_i(k+1) = A_{ci} x_i(k) \tag{4.72}$$

with $A_{ci} = A + B(\alpha_i K_i + (1-\alpha_i) H_i)$.

For the given state and control weighting matrices $Q_i \in \mathbb{R}^{n \times n}$ and $R_i \in \mathbb{R}$, consider the following set of quadratic functions

$$V_i(x_i) = x_i^T P_i x_i, \quad i = 2, 3, \ldots, r \tag{4.73}$$

where matrix $P_i \in \mathbb{R}^{n\times n}$, $P_i \succeq 0$ is chosen to satisfy

$$V_i(x_i(k+1)) - V_i(x_i(k)) \leq -x_i(k)^T Q_i x_i(k) - u_i(k)^T R_i u_i(k) \tag{4.74}$$

By using the same arguments as in the previous section, inequality (4.74) can be rewritten as

$$A_{ci}^T P_i A_{ci} - P_i \preceq -Q_i - (\alpha_i K_i + (1-\alpha_i)H_i)^T R_i (\alpha_i K_i + (1-\alpha_i)H_i)$$

Denote $Y_i = (\alpha_i K_i + (1-\alpha_i)H_i)$, by using the Schur complement, the above condition can be transformed into

$$\begin{bmatrix} P_i - Q_i - Y_i^T R_i Y_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} \succeq 0$$

or, equivalently

$$\begin{bmatrix} P_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} - \begin{bmatrix} Q_i + Y_i^T R_i Y_i & 0 \\ 0 & 0 \end{bmatrix} \succeq 0$$

Denote $Q_i^{\frac{1}{2}}$ and $R_i^{\frac{1}{2}}$ as the Cholesky factor of the matrices $Q_i$ and $R_i$, which satisfy $(Q_i^{\frac{1}{2}})^T Q_i^{\frac{1}{2}} = Q_i$ and $(R_i^{\frac{1}{2}})^T R_i^{\frac{1}{2}} = R_i$. The previous condition can be rewritten as

$$\begin{bmatrix} P_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} - \begin{bmatrix} (Q_i^{\frac{1}{2}})^T & Y_i^T (R_i^{\frac{1}{2}})^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_i^{\frac{1}{2}} & 0 \\ R_i^{\frac{1}{2}} Y_i & 0 \end{bmatrix} \succeq 0$$

or by using the Schur complement, one obtains

$$\begin{bmatrix} P_i & A_{ci}^T P_i & (Q_i^{\frac{1}{2}})^T & Y_i^T (R_i^{\frac{1}{2}})^T \\ P_i A_{ci} & P_i & 0 & 0 \\ Q_i^{\frac{1}{2}} & 0 & I & 0 \\ R_i^{\frac{1}{2}} Y_i & 0 & 0 & I \end{bmatrix} \succeq 0 \tag{4.75}$$

The left hand side of inequality (4.75) is linear in $\alpha_i$, and hence reaches its minimum at either $\alpha_i = 0$ or $\alpha_i = 1$. Consequently, the set of LMI conditions to be checked is following (4.75) and the fact that $Y_i = \alpha_i K_i + (1 - \alpha_i H_i)$

$$\begin{cases} \begin{bmatrix} P_i & (A+BK_i)^T P_i & (Q_i^{\frac{1}{2}})^T & (R_i^{\frac{1}{2}} K_i)^T \\ P_i(A+BK_i) & P_i & 0 & 0 \\ Q_i^{\frac{1}{2}} & 0 & I & 0 \\ R_i^{\frac{1}{2}} K_i & 0 & 0 & I \end{bmatrix} \succeq 0 \\ \begin{bmatrix} P_i & (A+BH_i)^T P_i & (Q_i^{\frac{1}{2}})^T & (R_i^{\frac{1}{2}} H_i)^T \\ P_i(A+BH_i) & P_i & 0 & 0 \\ Q_i^{\frac{1}{2}} & 0 & I & 0 \\ R_i^{\frac{1}{2}} H_i & 0 & 0 & I \end{bmatrix} \succeq 0 \end{cases} \tag{4.76}$$

Condition (4.76) is linear with respect to the matrix $P_i$. One way to calculate $P_i$ is to solve the following LMI problem

$$\min_{P_i}\{trace(P_i)\} \tag{4.77}$$

subject to constraint (4.76).

Once the matrices $P_i$ with $i = 2,3,\ldots,r$ are computed, they can be used in practice for real-time control based on the following algorithm, which can be formulated as an optimization problem that is efficient with respect to structure and complexity.

At each time instant, for a given current state $x$, minimize on-line the quadratic cost function

$$\min_{x_i,\lambda_i}\{\sum_{i=2}^{r} x_i^T P_i x_i + \sum_{i=2}^{r} \lambda_i^2\} \tag{4.78}$$

subject to the linear constraints

$$\begin{cases} F_o^{(i)} x_i \le \lambda_i g_o^{(i)}, \forall i = 1,2,\ldots,r \\ \sum\limits_{i=1}^{r} x_i = x \\ \sum\limits_{i=1}^{r} \lambda_i = 1 \\ \lambda_i \ge 0, \forall i = 1,2,\ldots,r. \end{cases}$$

**Theorem 4.10.** *The control law based on solving the optimization problem (4.78) guarantees recursive feasibility and asymptotic stability of the closed loop system for all initial states* $x(0) \in \Omega_s$.

*Proof.* The proof of this theorem is similar to the one of theorem 4.9. Hence it is omitted here. □

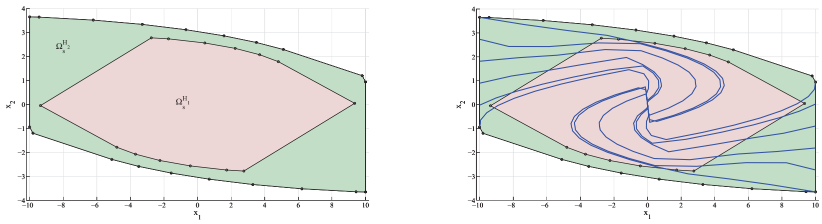
*Example 4.4.* Consider again the discrete-time linear time-invariant system in example (4.1) with the same state and control constraints. Two linear feedback controllers are chosen as

$$\begin{cases} K_1 = [-0.9500 \quad -1.1137] \\ K_2 = [-0.4230 \quad -2.0607] \end{cases} \tag{4.79}$$

Based on theorem 2.2, two auxiliary matrices are computed as

$$\begin{cases} H_1 = [-0.1055 \quad -0.2760] \\ H_2 = [-0.0669 \quad -0.2875] \end{cases} \tag{4.80}$$

With the auxiliary matrices $H_1$ and $H_2$, the invariant sets $\Omega_s^{H_1}$ and $\Omega_s^{H_2}$ are respectively constructed for the saturated controllers $u = sat(K_1 x)$ and $u = sat(K_2 x)$, see Figure 4.22(a). Figure 4.22(b) shows different state trajectories of the closed loop system for different initial conditions.



(a) Feasible invariant sets (b) State trajectories

**Fig. 4.22** Feasible invariant sets and state trajectories of the closed loop system for example 4.4.

The sets $\Omega_s^{H_1}$ and $\Omega_s^{H_2}$ are presented in minimal normalized half-space representation as

$$\Omega_s^{H_1} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.3933 & -0.9194 \\ -0.3933 & 0.9194 \\ -0.0403 & -0.9992 \\ 0.0403 & 0.9992 \\ -0.0791 & -0.9969 \\ 0.0791 & 0.9969 \\ -0.1238 & -0.9923 \\ 0.1238 & 0.9923 \\ -0.1787 & -0.9839 \\ 0.1787 & 0.9839 \\ -0.2515 & -0.9679 \\ 0.2515 & 0.9679 \\ -0.3571 & -0.9341 \\ 0.3571 & 0.9341 \end{bmatrix} x \leq \begin{bmatrix} 3.6367 \\ 3.6367 \\ 2.6637 \\ 2.6637 \\ 2.5901 \\ 2.5901 \\ 2.5977 \\ 2.5977 \\ 2.7012 \\ 2.7012 \\ 2.9371 \\ 2.9371 \\ 3.3844 \\ 3.3844 \end{bmatrix} \right\}$$

and

$$\Omega_s^{H_2} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.0170 & -0.9999 \\ 0.0170 & 0.9999 \\ -0.0386 & -0.9993 \\ 0.0386 & 0.9993 \\ -0.0610 & -0.9981 \\ 0.0610 & 0.9981 \\ -0.0850 & -0.9964 \\ 0.0850 & 0.9964 \\ -0.1117 & -0.9937 \\ 0.1117 & 0.9937 \\ -0.1425 & -0.9898 \\ 0.1425 & 0.9898 \\ 0.7948 & 0.6069 \\ -0.7948 & -0.6069 \\ -0.1796 & -0.9837 \\ 0.1796 & 0.9837 \\ 1.0000 & 0 \\ -1.0000 & 0 \\ -0.2266 & -0.9740 \\ 0.2266 & 0.9740 \end{bmatrix} x \leq \begin{bmatrix} 3.4795 \\ 3.4795 \\ 3.2732 \\ 3.2732 \\ 3.1299 \\ 3.1299 \\ 3.0449 \\ 3.0449 \\ 3.0182 \\ 3.0182 \\ 3.0552 \\ 3.0552 \\ 8.5177 \\ 8.5177 \\ 3.1696 \\ 3.1696 \\ 10.0000 \\ 10.0000 \\ 3.3878 \\ 3.3878 \end{bmatrix} \right\}$$

With the weighting matrices

$$Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R_2 = 0.001$$

and by solving the LMI problem (4.77), one obtains

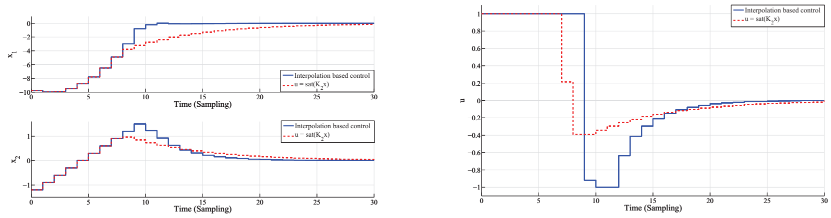
$$P_2 = \begin{bmatrix} 5.4929 & 9.8907 \\ 9.8907 & 104.1516 \end{bmatrix}$$

For the initial condition $x(0) = [-9.79 \quad -1.2]^T$, Figure 4.23 presents the state and input trajectory of the closed loop system as a function of time. The solid blue line is obtained by using the interpolation based control method 4.78, while the dashed red line is obtained by using the saturated controller $u = sat(K_2 x)$, which is the controller corresponding to the largest invariant set.

The interpolating coefficient $\lambda_2$ and the objective function as a Lyapunov function are shown in Figure 4.24.

## 4.7 Convex hull of ellipsoids

In this section, the convex hull of a family of ellipsoids is used for estimating the stability domain for a constrained control system. This is motivated by problems arising from the estimation of the domain of attraction of stable dynamics and the




(a) State trajectory (b) Input trajectory

**Fig. 4.23** State and input trajectory of the closed loop system as a function of time for example 4.4. The solid blue line is obtained by using the interpolation based control method 4.78, while the dashed red line is obtained by using the saturated controller $u = sat(K_2x)$.

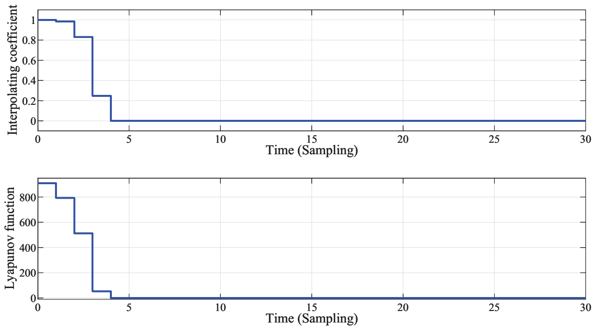



**Fig. 4.24** Interpolating coefficient and Lyapunov function as a function of time for example 4.4.

control design which aims to enlarge such a domain of attraction. In order to briefly describe the class of problems, let us suppose that a set of invariant ellipsoids and an associated set of *saturated* control laws are available. The questions whether the convex hull of this set of ellipsoids is invariant and how to construct a control law for this region are our objectives.

The fact that the convex hull of a set of invariant ellipsoids is also invariant is well known in the literature, for nominal continuous-time linear time-invariant systems, see [58], and for nominal discrete-time linear time-invariant systems, see [11]. In these papers, a method to construct a continuous feedback law based on a set of *linear* feedback laws was proposed to make the convex hull of a set of invariant ellipsoids invariant. The main contribution of this section is to provide a new type of interpolation based controller, that makes invariant the convex hull of invariant ellipsoids.

It is assumed that the polyhedral state constraints $X$ and the polyhedral input constraints $U$ are symmetric. It is also assumed that a set of asymptotically stabilizing feedback controllers $K_i \in \mathbb{R}^{m\times n}$ and using theorem 2.2, a set of auxiliary matrices

$H_i \in \mathbb{R}^{m\times n}$ for $i = 1,2,\ldots,r$ are available such that the corresponding ellipsoidal invariant sets $E(P_i)$

$$E(P_i) = \left\{x \in \mathbb{R}^n : x^T P_i^{-1} x \le 1\right\} \tag{4.81}$$

are non-empty for $i = 1,2,\ldots,r$. Recall that for all $x(k) \in E(P_i)$, it follows that $sat(K_i x) \in U$ and $x(k+1) = Ax(k) + Bsat(K_i x(k)) \in X$. Denote $\Omega_E \subset \mathbb{R}^n$ as a convex hull of $E(P_i)$ for all $i$. It follows that $\Omega_E \subseteq X$, since $E(P_i) \subseteq X$.

Any state $x(k) \in \Omega_E$ can be decomposed as follows

$$x(k) = \sum_{i=1}^{r} \lambda_i \widehat{x}_i(k) \tag{4.82}$$

with $\widehat{x}_i(k) \in E(P_i)$ and $\lambda_i$ are interpolating coefficients, that satisfy

$$\sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \ge 0$$

Consider the following control law

$$u(k) = \sum_{i=1}^{r} \lambda_i sat(K_i \widehat{x}_i(k)) \tag{4.83}$$

where $sat(K_i \widehat{x}_i(k))$ is the saturated control law, that is feasible in $E(P_i)$.

**Theorem 4.11.** *The control law (4.83) is guaranteed to be recursively feasible for any conditions* $x(0) \in \Omega_E$.

*Proof.* Starting with the decomposition (4.82), the control law obtained by the corresponding convex combination of the control actions is leading to the expression in (4.83). One has to prove that $u(k) \in U$ and $x(k+1) = Ax(k) + Bu(k) \in \Omega_E$ for all $x \in \Omega_E$. For the input constraints, from equation (4.83) and since $sat(K_i \widehat{x}_i(k)) \in U$, it follows that $u(k) \in U$.

For the state constraints, it holds that

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= A \sum_{i=1}^{r} \lambda_i \widehat{x}_i(k) + B \sum_{i=1}^{r} \lambda_i sat(K_i \widehat{x}_i(k)) \\ &= \sum_{i=1}^{r} \lambda_i (A\widehat{x}_i(k) + Bsat(K_i \widehat{x}_i(k))) \end{aligned}$$

One has $A\widehat{x}_i(k) + Bsat(K_i \widehat{x}_i(k)) \in E(P_i) \subseteq \Omega_E$ for all $i = 1,2,\ldots,r$, which ultimately assures that $x(k+1) \in \Omega_E$. □

As in the sections 4.5 and 4.6, the first high gain controller will be used for the performance, while the rest of available low gain controllers will be used to enlarge the domain of attraction. For the given current state $x$, consider the following objective function

$$\min_{\widehat{x}_i,\lambda_i}\{\sum_{i=2}^{r}\lambda_i\} \tag{4.84}$$

subject to

$$\begin{cases} \widehat{x}_i^T P_i^{-1}\widehat{x}_i \leq 1, \forall i = 1,2,\ldots,r \\ \sum_{i=1}^{r}\lambda_i\widehat{x}_i = x \\ \sum_{i=1}^{r}\lambda_i = 1 \\ \lambda_i \geq 0, \forall i = 1,2,\ldots,r \end{cases}$$

**Theorem 4.12.** *The control law using interpolation based on the objective function (4.84) guarantees asymptotic stability for all initial states $x(0) \in \Omega_E$.*

*Proof.* Let $\lambda_i^o$ be the solutions of the optimization problem (4.84) and consider the following positive function

$$V(x) = \sum_{i=2}^{r}\lambda_i^o(k) \tag{4.85}$$

for all $x \in \Omega_E \setminus E(P_1)$. $V(x)$ is a Lyapunov function candidate.

For any $x(k) \in \Omega_E \setminus E(P_1)$, one has

$$\begin{cases} x(k) = \sum_{i=1}^{r}\lambda_i^o(k)\widehat{x}_i^o(k) \\ u(k) = \sum_{i=1}^{r}\lambda_i^o(k)sat(K_i\widehat{x}_i^o(k)) \end{cases}$$

It follows that

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= A\sum_{i=1}^{r}\lambda_i^o(k)\widehat{x}_i^o(k) + B\sum_{i=1}^{r}\lambda_i^o(k)sat(K_i\widehat{x}_i^o(k)) \\ &= \sum_{i=1}^{r}\lambda_i^o(k)\widehat{x}_i(k+1) \end{aligned}$$

where $\widehat{x}_i(k+1) = A\widehat{x}_i^o(k) + Bsat(K_i\widehat{x}_i^o(k)) \in E(P_i)$ for all $i = 1,2,\ldots,r$.

By using the interpolation based on the optimization problem (4.84)

$$x(k+1) = \sum_{i=1}^{r}\lambda_i^o(k+1)\widehat{x}_i^o(k+1)$$

where $\widehat{x}_i^o(k+1) \in E(P_i)$. It follows that

$$\sum_{i=2}^{r}\lambda_i^o(k+1) \leq \sum_{i=2}^{r}\lambda_i^o(k)$$

and $V(x)$ is a non-increasing function.

The contractive invariant property of the ellipsoid $E(P_i)$ assures that there is no initial condition $x(0) \in \Omega_E \setminus E(P_1)$ such that $\sum_{i=2}^{r} \lambda_i^o(k+1) \leq \sum_{i=2}^{r} \lambda_i^o(k)$ for all $k \geq 0$. It follows that $V(x) = \sum_{i=2}^{r} \lambda_i^o(k)$ is a Lyapunov function for all $x \in \Omega_E \setminus E(P_1)$.

The proof is completed by noting that inside $E(P_1)$, the feasible stabilizing controller $u = sat(K_1\widehat{x})$ is contractive and thus the interpolation based controller assures asymptotic stability for all $x \in \Omega_E$. □

If we denote $x_i = \lambda_i \widehat{x}_i$, then since $\widehat{x}_i \in E(P_i)$, it follows that $x_i^T P_i^{-1} x_i \leq \lambda_i^2$. The non-linear optimization problem (4.84) can be rewritten as follows

$$\min_{x_i,\lambda_i} \{\sum_{i=2}^{r} \lambda_i\}$$

subject to

$$\begin{cases} x_i^T P_i^{-1} x_i \leq \lambda_i^2, \forall i = 1,2,\ldots,r \\ \sum_{i=1}^{r} x_i = x \\ \sum_{i=1}^{r} \lambda_i = 1 \\ \lambda_i \geq 0, \forall i = 1,2,\ldots,r \end{cases}$$

or by using the Schur complement

$$\min_{x_i,\lambda_i} \sum_{i=2}^{r} \lambda_i \tag{4.86}$$

subject to

$$\begin{cases} \begin{bmatrix} \lambda_i & x_i^T \\ x_i & \lambda_i P_i \end{bmatrix} \succeq 0, \forall i = 1,2,\ldots,r \\ \sum_{i=1}^{r} x_i = x \\ \sum_{i=1}^{r} \lambda_i = 1 \\ \lambda_i \geq 0, \forall i = 1,2,\ldots,r \end{cases}$$

This is an LMI optimization problem.

In summary, at each time instant the interpolation based controller involves the following steps

**Algorithm 4.4** Interpolation based control - Convex hull of ellipsoids

1. Measure the current state of the system $x(k)$.
2. Solve the LMI problem (4.86). In the result, one gets $x_i^o \in E(P_i)$ and $\lambda_i^o$ for all $i = 1,2,\dots,q$.
3. For $x_i^o \in E(P_i)$, one associates the control value $u_i^o = sat(K_i x_i^o)$.
4. The control signal to be applied to the plant $u(k)$ is found as a convex combination of $u_i^o$

$$u(k) = \sum_{i=1}^{r} \lambda_i^o(k) u_i^o$$

*Remark 4.13.* It is worth noticing that for all $x(k) \in E(P_1)$, the LMI problem (4.86) has a trivial solution

$$\lambda_i = 0, \quad \forall i = 2,3,\dots,r$$

Hence $\lambda_1 = 1$ and $x = \widehat{x}_1$. In this case, the interpolation based controller turns out to be the saturated controller $u = sat(K_1 x)$.

*Example 4.5.* Consider again the discrete-time linear time-invariant system in example (4.1) with the same state and control constraints. Three linear feedback controllers are chosen as

$$\begin{cases} K_1 = [-0.9500 \quad -1.1137], \\ K_2 = [-0.4230 \quad -2.0607], \\ K_3 = [-0.5010 \quad -2.1340] \end{cases} \tag{4.87}$$

Based on theorem 2.2 and the optimization problem (2.55), three auxiliary matrices are found as

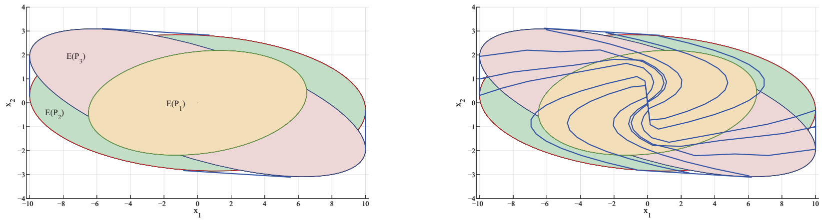
$$\begin{cases} H_1 = [-0.1055 \quad -0.2760], \\ H_2 = [-0.0669 \quad -0.2875], \\ H_3 = [-0.0727 \quad -0.4148] \end{cases} \tag{4.88}$$

With these auxiliary matrices, three invariant ellipsoids $E(P_1)$, $E(P_2)$, $E(P_3)$ are computed corresponding to the saturated controllers $u = sat(K_1 x)$, $u = sat(K_2 x)$ and $u = sat(K_3 x)$. The invariant sets and their convex hull are depicted in Figure 4.25(a). Figure 4.25(b) shows state trajectories of the closed loop system for different initial conditions.

The matrices $P_1$, $P_2$ and $P_3$ are

$$P_1 = \begin{bmatrix} 42.2661 & 2.8153 \\ 2.8153 & 4.7997 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 100.0000 & -3.0957 \\ -3.0957 & 8.1243 \end{bmatrix},$$
$$P_3 = \begin{bmatrix} 100.0000 & -19.4011 \\ -19.4011 & 9.5408 \end{bmatrix}$$

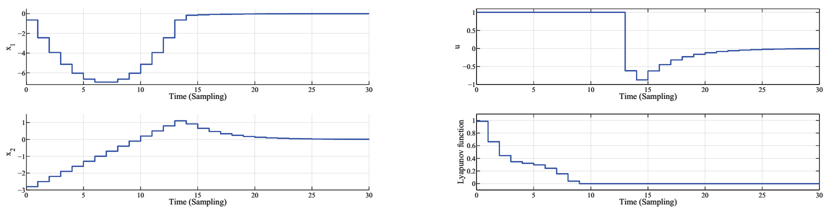
For the initial condition $x(0) = [-0.64 \quad -2.8]^T$, Figure 4.26 presents the state trajectory, the input trajectory and the Lyapunov function of the closed loop system




(a) Invariant ellipsoids (b) State trajectories

**Fig. 4.25** Feasible invariant sets and state trajectories of the closed loop system for example 4.5.

as a function of time. As expected, the Lyapunov function, i.e. the objective function is positive and non-increasing.




(a) State trajectory (b) Input trajectory and Lyapunov function

**Fig. 4.26** State trajectory, input trajectory and Lyapunov function of the closed loop system as a function of time for example 4.5.

# Chapter 5
# Interpolation Based Control – Robust State Feedback Case

In this chapter, the problem of regulating a constrained discrete-time linear time-varying or uncertain system to the origin subject to bounded disturbances is addressed. The robust counterpart of the interpolation technique generalizes the results presented in the previous chapter, recursive feasibility and robust asymptotic stability being preserved. It is shown that in the implicit case, depending on the shape of invariant sets, i.e. polyhedral or ellipsoidal, and depending on the objective functions, i.e. linear or quadratic, two LPs or one QP or one LMI problem is solved at each time instant. In the explicit case, the control law is shown to be a piecewise affine function of state.

## 5.1 Problem formulation

Consider the problem of regulating to the origin the following discrete-time linear *time-varying* or *uncertain* systems subject to additive bounded disturbances

$$x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \tag{5.1}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $w(k) \in \mathbb{R}^d$ are respectively the state, the input and the disturbance vectors. The system matrices $A(k) \in \mathbb{R}^{n\times n}$, $B(k) \in \mathbb{R}^{n\times m}$ and $D(k) \in \mathbb{R}^{n\times d}$ satisfy

$$\begin{cases} A(k) = \sum\limits_{i=1}^{q} \alpha_i(k)A_i, \quad B(k) = \sum\limits_{i=1}^{q} \alpha_i(k)B_i, \quad D(k) = \sum\limits_{i=1}^{q} \alpha_i(k)D_i, \\ \sum\limits_{i=1}^{q} \alpha_i(k) = 1, \quad \alpha_i(k) \geq 0 \end{cases} \tag{5.2}$$

where the matrices $A_i$, $B_i$ and $D_i$ are given. A somewhat more general uncertainty description is given by equation (2.20) in Chapter 2 which can be transformed to the one in (5.2).

The state, the control and the disturbance are subject to the following polytopic constraints

$$\begin{cases} x(k) \in X, & X = \{x \in \mathbb{R}^n : F_x x \leq g_x\} \\ u(k) \in U, & U = \{u \in \mathbb{R}^m : F_u u \leq g_u\} \\ w(k) \in W, & W = \{w \in \mathbb{R}^d : F_w w \leq g_w\} \end{cases} \tag{5.3}$$

where the matrices $F_x$, $F_u$ and $F_w$ and the vectors $g_x$, $g_u$ and $g_w$ are assumed to be constant with $g_x > 0$, $g_u > 0$, $g_w > 0$ such that the origin is contained in the interior of $X$, $U$ and $W$. The inequalities here are component-wise.

In this chapter, it is assumed that the states of the system are measurable.

## 5.2 Interpolation based on linear programming

Starting from the assumption that an unconstrained robust asymptotically stabilizing feedback controller $u(k) = Kx(k)$ is available such that the corresponding maximal robustly invariant set $\Omega_{max} \subseteq X$

$$\Omega_{max} = \{x \in \mathbb{R}^n : F_o x \leq g_o\} \tag{5.4}$$

is non-empty. Furthermore with some given and fixed integer $N > 0$, based on procedure 2.3 presented in section 2.3.4 one can find a robust controlled invariant set $C_N$ in the form

$$C_N = \{x \in \mathbb{R}^n : F_N x \leq g_N\} \tag{5.5}$$

such that all $x \in C_N$ can be steered into $\Omega_{max}$ in no more than $N$ steps when suitable control is applied. The polytope $C_N$ can be decomposed into a set of simplices $C_N^{(j)}$, each formed by $n$ vertices of $C_N$ and the origin. For all $x \in C_N^{(j)}$, the vertex controller

$$u(k) = K^{(j)} x(k) \tag{5.6}$$

can be applied with $K^{(j)} \in \mathbb{R}^{m \times n}$ is defined as in (3.50). In Section 3.4, it was shown that the system (5.1) in closed loop with vertex control is robustly asymptotically stable[1] for all initial states $x \in C_N$.

In the robust case, similar to the nominal case presented in in Chapter 4, Section 4.2, the weakness of vertex control is that the full control range is exploited only on the border of the set $C_N$ in the state space, with progressively smaller control action when state approaches the origin. Hence the time to regulate the plant to the origin is longer than necessary.

Here we provide a method to overcome this shortcoming, where the control action is still smooth. For this purpose, any state $x(k) \in C_N$ is decomposed as

[1] Here by robust asymptotic stability we understand that the state converges asymptotically to a minimal robust positively invariant set [127], [116], [76], which replaces the origin as attractor for the system (5.1) in closed loop with the vertex controller.

$$x(k) = c(k)x_v(k) + (1 - c(k))x_o(k) \tag{5.7}$$

where $x_v \in C_N$, $x_o \in \Omega_{max}$ and $0 \leq c \leq 1$.

Consider the following control law

$$u(k) = c(k)u_v(k) + (1 - c(k))u_o(k) \tag{5.8}$$

where $u_v(k)$ is obtained by applying the vertex control law (5.6) for $x_v(k)$ and $u_o(k) = Kx_o(k)$ is the control law that is feasible in $\Omega_{max}$.

**Theorem 5.1.** *For system (5.1) and constraints (5.3), the control law (5.8) guarantees recursive feasibility for all initial states* $x(0) \in C_N$.

*Proof.* For recursive feasibility, it has to be proved that

$$\begin{cases} F_u u(k) \leq g_u \\ x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \in C_N \end{cases}$$

for all $x(k) \in C_N$. While the feasibility of the input constraints is proved in a similar way to the nominal case[2], the state constraint feasibility deserves an adaptation.

$$\begin{aligned} &x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \\ &= A(k)\{c(k)x_v(k) + (1 - c(k))x_o(k)\} + B(k)\{c(k)u_v(k) + (1 - c(k))u_o(k)\} + D(k)w(k) \\ &= c(k)x_v(k+1) + (1 - c(k))x_o(k+1) \end{aligned}$$

where

$$\begin{aligned} x_v(k+1) &= A(k)x_v(k) + B(k)u_v(k) + D(k)w(k) \in C_N \\ x_o(k+1) &= A(k)x_o(k) + B(k)u_o(k) + D(k)w(k) \in \Omega_{max} \subseteq C_N \end{aligned}$$

It follows that $x(k+1) \in C_N$. □

As in Section 4.2, in order to be as close as possible to the optimal unconstrained local controller, one would like to minimize the interpolating coefficient $c(k)$. This can be done by solving the following nonlinear optimization problem

$$c^* = \min_{x_v, x_o, c} \{c\} \tag{5.9}$$

subject to

$$\begin{cases} F_N x_v \leq g_N \\ F_o x_o \leq g_o \\ c x_v + (1 - c) x_o = x \\ 0 \leq c \leq 1 \end{cases}$$

Define $r_v = c x_v$ and $r_o = (1 - c) x_o$. Since $x_v \in C_N$ and $x_o \in \Omega_{max}$, it follows that $r_v \in c C_N$ and $r_o \in (1 - c)\Omega_{max}$ or equivalently

---

[2] See proof of theorem 4.1 in section 4.2

$$\begin{cases} F_N r_v \leq c g_N \\ F_o r_o \leq (1-c) g_o \end{cases}$$

With this change of variables, the nonlinear optimization problem (5.9) is transformed into a linear programming problem as follows

$$c^* = \min_{r_v, c} \{c\} \tag{5.10}$$

subject to

$$\begin{cases} F_N r_v \leq c g_N \\ F_o (x - r_v) \leq (1-c) g_o \\ 0 \leq c \leq 1 \end{cases}$$

**Theorem 5.2.** *The control law using interpolation based on linear programming (5.10) guarantees robust asymptotic stability*[3] *for all initial states* $x(0) \in C_N$.

*Proof.* The proof of this theorem is omitted here, since it is the same as the proof of theorem 4.2, Section 4.2. □

An on-line algorithm for the interpolation based controller via linear programming is

**Algorithm 5.1: Interpolation based control - Implicit solution**

1. Measure the current state of the system $x(k)$.
2. Solve the LP problem (5.10).
3. Implement as input the control action (5.8).
4. Wait for the next time instant $k := k + 1$.
5. Go to step 1 and repeat.

Although the dimension of the LP problem (5.10) is $n+1$, where $n$ is the dimension of state, the complexity of the control law (5.8) is in direct relationship with the complexity of the vertex controller and can be very high, since in general the complexity of the set $C_N$ is high in terms of vertices. Also it is well known [25] that the number of simplicies of vertex control is typically much greater than the number of vertices. Therefore a question is how to achieve an interpolating controller whose complexity is not correlated with the complexity of the involved sets.

It is obvious that vertex control is only one possible choice for the global outer controller. One can consider any other linear or non-linear controller and the principle of interpolation scheme (5.7), (5.8), (5.10) holds as long as the convexity of the

[3] Here by robust asymptotic stability we understand that the state of the closed loop system with the interpolation based controller converges to the minimal robustly positively invariant set despite the parameter variation and the influence of additive disturbances.

associated controlled invariant set is preserved. A natural candidate for the global controller is the saturated controller $u = sat(K_s x)$ with the associated invariant set $\Omega_s$ computed using procedure 2.4 in Section 2.3.4. The experience usually shows that by properly choosing the saturated gain $K_s \in \mathbb{R}^{m\times n}$, the associated invariant set $\Omega_s$ may approach the invariant sets other constrained controllers might have.

In summary with the global saturated controller $u(k) = sat(K_s x(k))$ the interpolation based control law (5.7), (5.8), (5.10) involves the following steps

1. Design a local gain $K$ and a global gain $K_s$, both stabilizing with some desired performance specifications. Usually $K$ is chosen for the performance, while $K_s$ is designed for the quality of its domain of attraction.
2. Compute the invariant sets $\Omega_{max}$ and $\Omega_s$ associated with the controllers $K$ and $K_s$ respectively. The set $\Omega_{max}$ is computed using procedure 2.2 in Section 2.3.4, while the set $\Omega_s$ is computed using procedure 2.4. in the same section.
3. Implement the control law (5.7), (5.8), (5.10).

Practically, the interpolation scheme using the saturated controller shows to be simpler than the interpolation scheme using vertex control, while the domain of attraction remains typically the same. In order to complete the picture of the available possibilities in the choice of control policies in the interpolation scheme, we provide below an alternative for choosing the global controller. With this aim, some geometrical properties of the solution of the LP problem (5.10) will be recalled.

*Remark 5.1.* In the robust case, similar to the nominal case presented in Section 4.3

1. If $x \in \Omega_{max}$ the result of the optimal interpolation problem has a trivial solution $x_o = x$ and thus $c^* = 0$ in (5.10).
2. If $x \in C_N \setminus \Omega_{max}$, the interpolating coefficient $c$ will reach a minimum in (5.10) if and only if $x$ is written as a convex combination of two points, one belonging to the boundary of $\Omega_{max}$ and the other on the boundary of $C_N$.

As a consequence of remark 5.1, the vertex control law is only one of the candidate of the controller at the boundary of $C_N$. It is clear that any control law that steers the state on the boundary of $C_N$ towards the interior of $C_N$ will make the interpolation based control (5.7), (5.8), (5.10) robustly asymptotically stable.

An intuitive approach is to devise a controller, that pushes the state away from the boundary of the controlled invariant set $C_N$ *as far as possible* in a contractive sense. In order to give a precise definition of *far*, the following definition is introduced [95], [23].

**Definition 5.1. (Minkowski functional)** [95], [23] Given a $C-$set $S$, the Minkowski functional $\Psi_S$ of $S$ is defined as

$$\Psi_S(x) = \min_{\mu}\{\mu \geq 0 : x \in \mu S\} \tag{5.11}$$

It is well known [95] that the function $\Psi_S$ is convex, positively homogeneous of order one, i.e. for any scalar $t \geq 0$, it holds that $\Psi_S(tx) = t\Psi_S(x)$. Furthermore it is a norm if and only if the set $S$ is $0-$symmetric. Its level surfaces are given by scaling the boundary of the set $S$. Thus such boundary defines the shape of the function. Figure 5.1 depicts the set $S$ (the red line) and the level surfaces corresponding to $\Psi = 0.6$ and $\Psi = 0.3$.

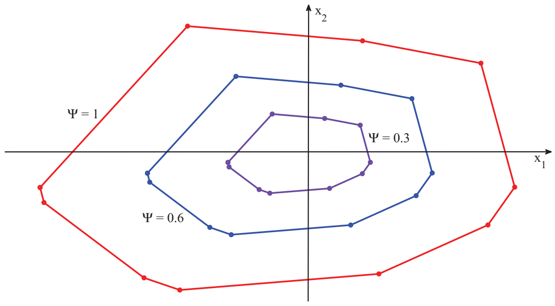



**Fig. 5.1** Minkowski functional.

So at each time instant, we will try to minimize the Minkowski functional for the state $x_v$ at the boundary of the feasible invariant set $C_N$. This can be done by solving the following linear program [27]

$$\mu^* = \min_{u,\mu}\{\mu\} \tag{5.12}$$

subject to

$$\begin{cases} F_N(A_i x_v + B_i u) \leq \mu g_N - \max F_N D_i w \\ F_u u \leq g_u \\ 0 \leq \mu \leq 1 \end{cases}$$

for all $i = 1, 2, \ldots, q$ and for all $w \in W$.

*Remark 5.2.* The minimization of the Minkowski functional can be interpreted in terms of a one-step Model Predictive Control method.

*Remark 5.3.* The non-uniqueness of the solution is a main issue regarding the implementation of the global control based on the LP problem (5.12). This issue might arise from the degeneracy of the LP problem (5.12). Multiple optima are undesirable, as they might lead to a fast switching between the different optimal control actions when the LP problem (5.12) is solved on-line [105], [75]. Note that for vertex control there is no such problem.

In summary, the interpolation based controller involves the following steps

**Algorithm 5.2: Interpolation based control - Implicit solution**

1. Measure the current state of the system $x(k)$.
2. Solve the LP problem (5.10). In the result one gets $x_v, x_o$ and $c^*$ with $x_v \in \mathrm{Fr}(C_N)$, $x_o \in \mathrm{Fr}(\Omega_{max})$ and $x = c^* x_v + (1 - c^*) x_o$.
3. For $x_v \in \mathrm{Fr}(C_N)$, the control value $u_v$ is obtained by solving the LP problem (5.12).
4. Implement as input the control signal $u = c^* u_v + (1 - c^*) u_o$.
5. Wait for the next time instant $k := k + 1$ and the associated state measurements.
6. Go to step 1 and repeat.

It is worth noticing that for algorithm 5.2, at each time instant, two linear programs have to be solved sequentially, one is of dimension $n+1$ and the other is of dimension $m+1$ where $n$ and $m$ are respectively, the dimension of state and control input. Hence algorithm 5.2 is more computationally demanding than algorithm 5.1. However if the number of vertices of the feasible set $C_N$ exceeds the number of facets, algorithm 5.2 is preferable, due to the storage complexity of the global vertex controller used in the evaluation of the control action in equation (5.8) in the algorithm 5.1.

*Remark 5.4.* Concerning the explicit solution of the interpolation based control with the global vertex controller, using the same argument as in Section 4.3, it can be concluded that

- If $x \in C_N \setminus \Omega_{max}$, the smallest value of the interpolating coefficient $c$ will be reached when the region $C_N \setminus \Omega$ is decomposed into partitions in form of simplices with vertices either on the boundary of $C_N$ or on the boundary of $\Omega_{max}$. The control law in each partition is piecewise affine function of state whose gains are obtained by interpolation of control values at the vertices of the simplex.
- If $x \in \Omega_{max}$, then the control law is the optimal unconstrained controller one.

*Example 5.1.* Consider the following uncertain discrete-time system

$$x(k+1) = A(k)x(k) + B(k)u(k) \tag{5.13}$$

where

$$\begin{cases} A(k) = \alpha(k)A_1 + (1 - \alpha(k))A_2 \\ B(k) = \alpha(k)B_1 + (1 - \alpha(k))B_2 \end{cases}$$

and

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}$$

At each time instant $\alpha(k) \in [0,\ 1]$ is an uniformly distributed pseudo-random number. The constraints are

$$\begin{aligned} -10 \leq x_1 \leq 10, \quad -10 \leq x_2 \leq 10, \\ -1 \leq u \leq 1 \end{aligned} \tag{5.14}$$

The stabilizing feedback gain for states near the origin is chosen as

$$K = [-1.8112 \quad -0.8092]$$

Using procedure 2.2 and procedure 2.3 in Chapter 2, one obtains the sets $\Omega_{max}$ and $C_N$ as shown in Figure 5.2. Note that $C_{27} = C_{28}$, in this case $C_{27}$ is a maximal robustly invariant set for system (5.13).

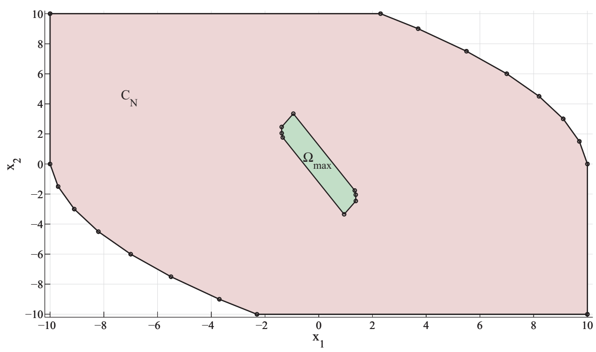



**Fig. 5.2** Feasible invariant sets for example 5.1.

The set $\Omega_{max}$ is presented in minimal normalized half-space representation as

$$\Omega_{max} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.9130 & -0.4079 \\ 0.9130 & 0.4079 \\ 0.8985 & -0.4391 \\ -0.8985 & 0.4391 \\ 1.0000 & 0.0036 \\ -1.0000 & -0.0036 \\ 0.9916 & 0.1297 \\ -0.9916 & -0.1297 \end{bmatrix} \leq \begin{bmatrix} 0.5041 \\ 0.5041 \\ 2.3202 \\ 2.3202 \\ 1.3699 \\ 1.3699 \\ 1.1001 \\ 1.1001 \end{bmatrix} \right\}$$

The set of vertices of $C_N$ is given by the matrix $V(C_N)$ below, together with the control matrix $U_v$ at these vertices

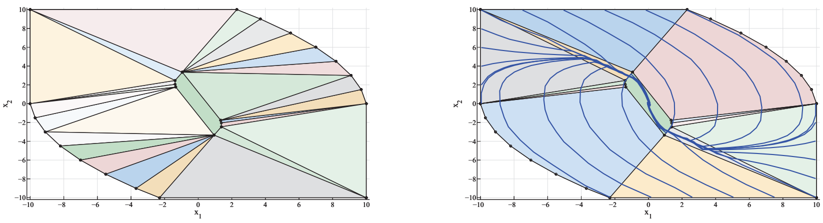
$$\begin{cases} V(P_N) = [V_1 \quad -V_1] \\ U_v \quad = [U_1 \quad -U_1] \end{cases}$$

where

$$V_1 = \begin{bmatrix} 10.0000 & 9.7000 & 9.1000 & 8.2000 & 7.0000 & 5.5000 & 3.7000 & 2.3000 & -10.0000 \\ 0 & 1.5000 & 3.0000 & 4.5000 & 6.0000 & 7.5000 & 9.0000 & 10.0000 & 10.0000 \end{bmatrix}$$

$$U_1 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Solving explicitly the LP problem (5.10) by using multi-parametric linear programming, a state space partition is obtained as depicted in Figure 5.3(a). The number of polyhedral partition is $N_r = 27$. Merging the regions with the identical control law, one obtains the reduced state space partition ($N_r = 13$) in Figure 5.3(b). In the same Figure, different trajectories of the closed loop system are presented for different initial conditions and different realizations of $\alpha(k)$.



(a) Number of regions $N_r = 27$ (b) Number of regions $N_r = 13$

**Fig. 5.3** Explicit solution before and after merging for the interpolation based control method and different trajectories of the closed loop system for example 5.1.

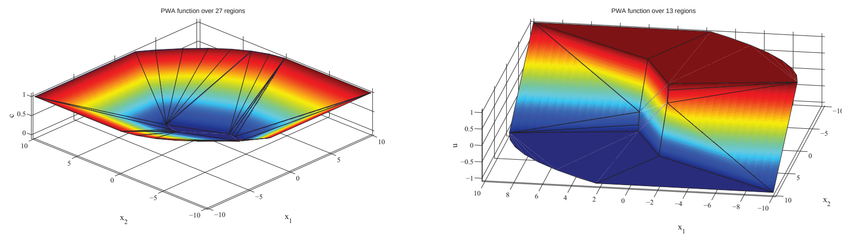
The control law over the state space partition with 13 regions is

$$
u(k) = \begin{cases}
-1 & \text{if} \begin{bmatrix} 0.93 & 0.37 \\ 0.78 & 0.62 \\ 0.64 & 0.77 \\ -0.91 & -0.41 \\ -0.90 & 0.44 \\ 0.58 & 0.81 \\ 0.71 & 0.71 \\ 0.86 & 0.51 \\ 0.20 & -0.98 \\ 0.98 & 0.20 \end{bmatrix} x(k) \le \begin{bmatrix} 9.56 \\ 9.21 \\ 9.28 \\ -0.50 \\ 2.32 \\ 9.47 \\ 9.19 \\ 9.35 \\ 2.00 \\ 9.81 \end{bmatrix} \\
-1 & \text{if} \begin{bmatrix} 0.90 & -0.44 \\ -0.00 & 1.00 \\ -0.59 & -0.81 \end{bmatrix} x(k) \le \begin{bmatrix} -2.32 \\ 10.00 \\ -2.14 \end{bmatrix} \\
-0.92x_1(k) - 1.25x_2(k) + 2.31 & \text{if} \begin{bmatrix} 0.90 & -0.44 \\ 0.59 & 0.81 \\ -0.66 & -0.75 \end{bmatrix} x(k) \le \begin{bmatrix} -2.32 \\ 2.14 \\ -0.95 \end{bmatrix} \\
-0.00x_1(k) - 0.20x_2(k) + 1.00 & \text{if} \begin{bmatrix} 0.66 & 0.75 \\ -1.00 & 0.00 \\ 0.27 & -0.96 \end{bmatrix} x(k) \le \begin{bmatrix} 0.95 \\ 10.00 \\ -2.75 \end{bmatrix} \\
0.17x_1(k) - 0.80x_2(k) + 2.72 & \text{if} \begin{bmatrix} 1.00 & 0.00 \\ 0.23 & -0.97 \\ -0.27 & 0.96 \end{bmatrix} x(k) \le \begin{bmatrix} -1.37 \\ -2.31 \\ 2.75 \end{bmatrix} \\
0.11x_1(k) - 0.56x_2(k) + 2.13 & \text{if} \begin{bmatrix} 0.99 & 0.13 \\ 0.20 & -0.98 \\ -0.23 & 0.97 \end{bmatrix} x(k) \le \begin{bmatrix} -1.10 \\ -2.00 \\ 2.31 \end{bmatrix} \\
-1.81x_1(k) - 0.81x_2(k) & \text{if} \begin{bmatrix} -0.91 & -0.41 \\ 0.91 & 0.41 \\ 0.90 & -0.44 \\ -0.90 & 0.44 \\ 1.00 & 0.00 \\ -1.00 & -0.00 \\ 0.99 & 0.13 \\ -0.99 & -0.13 \end{bmatrix} x(k) \le \begin{bmatrix} 0.50 \\ 0.50 \\ 2.32 \\ 2.32 \\ 1.37 \\ 1.37 \\ 1.10 \\ 1.10 \end{bmatrix}
\end{cases}
$$

(due to symmetry of the explicit solution, only the control law for seven regions are reported here)

The interpolating coefficient and the control input as a piecewise affine function of state is depicted in Figure 5.4. It is worth noticing that $c^* = 0$ inside the set $\Omega_{max}$.

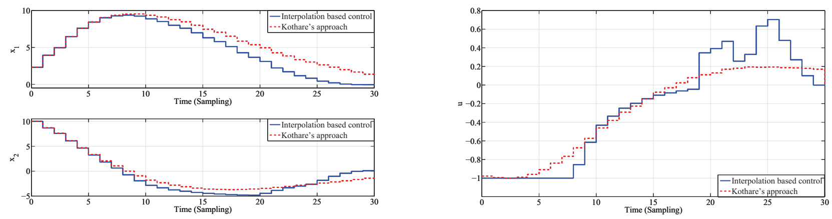
For the initial condition $x_0 = [2.2954 \quad 9.9800]^T$, Figure 5.5(a) and Figure 5.5(b) show the state and input trajectories as a function of time. The solid blue line is obtained by using the interpolation based control method and confirms the stabilizing as well as good performances for regulation. As a comparison, Figure 5.5(a) and Figure 5.5(b) also show the state and input trajectories obtained by using algorithms




(a) Interpolating coefficient (b) Control input

**Fig. 5.4** The interpolating coefficient and the control input as a piecewise affine function of state for example 5.1.

proposed by Kothare et al. in [78]. Note that algorithms in [78] require a solution of a semidefinite problem.




(a) State trajectories (b) Input trajectories

**Fig. 5.5** State and input trajectories as a function of time for example 5.1. The solid blue line is obtained by using the interpolation based control method, and the dashed red line is obtained by using the method in [78].

Figure 5.6 presents the interpolating coefficient and the realization of $\alpha(k)$ as a function of time. As expected $c^*(k)$ is a positive and non-increasing function.

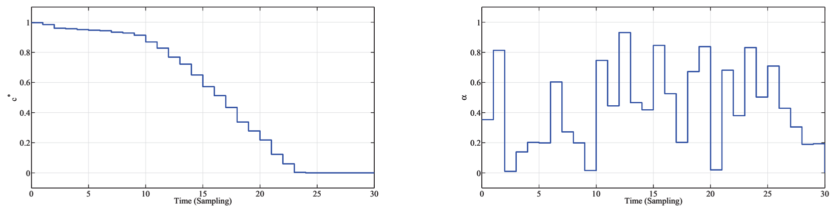
The following state and control weighting matrices were used for the LMI based MPC algorithm in in [78]

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1$$

*Example 5.2.* This example extends the study of the nominal case. The discrete-time linear time-invariant system with disturbances is given as

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k) + w(k) \tag{5.15}$$

The state-input constraints and disturbance bounds are




(a) Interpolating coefficient (b) $\alpha(k)$ realization

**Fig. 5.6** Interpolating coefficient and the realization of $\alpha(k)$ as a function of time for example 5.1.

$$\begin{array}{c} -5 \leq x_1 \leq 5, \quad -5 \leq x_2 \leq 5 \\ -1 \leq u \leq 1 \\ -0.1 \leq w_1 \leq 0.1, \quad -0.1 \leq w_2 \leq 0.1 \end{array} \tag{5.16}$$

An LQ gain with weighting matrices

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = 0.001$$

leads to a local unconstrained feedback gain

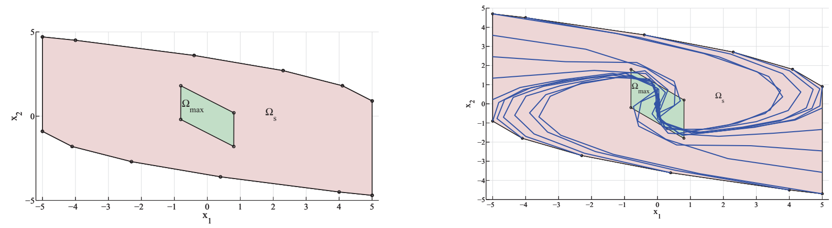
$$K = [-0.9970 \quad -0.9990]$$

The following saturated controller $u(k) = sat(K_s x(k))$ is chosen as a global controller with the linear gain

$$K_s = [-0.1408 \quad -0.3968]$$

Using procedure 2.2 and procedure 2.4 for the control laws $u(k) = Kx(k)$ and $u(k) = sat(K_s x(k))$ respectively, the maximal invariant sets $\Omega_{max}$ and $\Omega_s$ are computed. The result is depicted in Figure 5.7(a). Note that the set $\Omega_s$ is actually the maximal domain of attraction for the system (5.15) with constraints (5.16), which can be verified by comparing the equivalence between the set $\Omega_s$ and its one-step robust controlled invariant set. Figure 5.7(b) presents different state trajectories for different initial conditions and different realizations of $w(k)$. It can be observed that the trajectories do not converge to the origin but to a minimal robust positively invariant set of the system $x(k+1) = (A+BK)x(k) + w(k)$ that contains the origin in the interior.

The sets $\Omega_{max}$ and $\Omega_s$ are presented in minimal normalized half-space representation as



(a) Feasible invariant sets (b) State trajectories

**Fig. 5.7** Feasible invariant sets and different trajectories of the closed loop system for example 5.2.

$$\Omega_{max} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.7064 & -0.7078 \\ 0.7064 & 0.7078 \\ 1.0000 & -0.0020 \\ -1.0000 & 0.0020 \end{bmatrix} x \leq \begin{bmatrix} 0.7085 \\ 0.7085 \\ 0.8060 \\ 0.8060 \end{bmatrix} \right\}$$

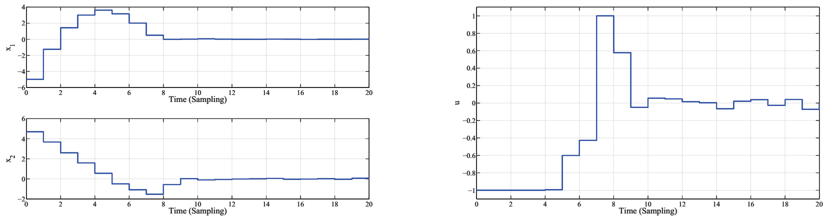
$$\Omega_s = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.0000 & 0 \\ -1.0000 & 0 \\ 0.7071 & 0.7071 \\ -0.7071 & -0.7071 \\ 0.4472 & 0.8944 \\ -0.4472 & -0.8944 \\ 0.3162 & 0.9487 \\ -0.3162 & -0.9487 \\ 0.2425 & 0.9701 \\ -0.2425 & -0.9701 \\ 0.1961 & 0.9806 \\ -0.1961 & -0.9806 \end{bmatrix} x \leq \begin{bmatrix} 5.0000 \\ 5.0000 \\ 4.1719 \\ 4.1719 \\ 3.4435 \\ 3.4435 \\ 3.2888 \\ 3.2888 \\ 3.3955 \\ 3.3955 \\ 3.6281 \\ 3.6281 \end{bmatrix} \right\}$$

For the initial condition $x(0) = [-5.0000 \quad 4.7000]^T$, Figure 5.8 shows the state and input trajectory as a function of time.

Figure 5.9 presents the interpolating coefficient $c^*(k)$ and the realization of $w(k)$ as a function of time.

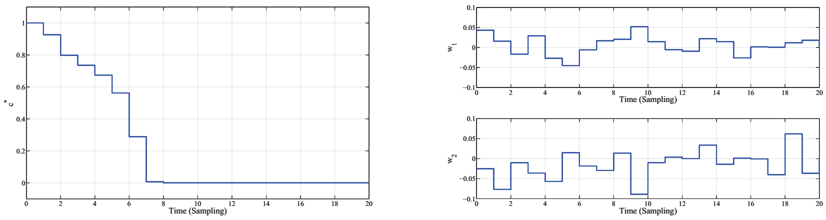
## 5.3 Interpolation based on quadratic programming for uncertain systems

The non-uniqueness of the solution is the main issue regarding the implementation of the interpolation via linear programming in Section 5.2. Hence, as in the nominal case, it is also worthwhile in the robust case to have an interpolation scheme with strictly convex objective function.




(a) State trajectory (b) Input trajectory

**Fig. 5.8** State and input trajectory of the closed loop system as a function of time for example 5.2.




(a) Interpolating coefficient (b) $w(k)$ realization

**Fig. 5.9** Interpolating coefficient and the realization of $w(k)$ as a function of time for example 5.2.

In this section, we consider the problem of regulating to the origin system (5.1) in the absence of disturbances. In other words, the system under consideration is of the form

$$x(k+1) = A(k)x(k) + B(k)u(k) \tag{5.17}$$

where the uncertainty description of $A(k)$ and $B(k)$ is as in (5.2).

For a given set of robust asymptotically stabilizing controllers $u(k) = K_i x(k)$, $i = 1, 2, \ldots, r$ and corresponding maximal robust positively invariant sets $\Omega_i \subseteq X$

$$\Omega_i = \{x \in \mathbb{R}^n : F_o^{(i)} x \le g_o^{(i)}\} \tag{5.18}$$

denote $\Omega$ as a convex hull of $\Omega_i$. It follows from the convexity of $X$ that $\Omega \subseteq X$, since $\Omega_i \subseteq X$ for all $i = 1, 2, \ldots, r$.

By employing the same design scheme in Section 4.5, the first high gain controller in this enumeration will play the role of a performance controller, while the remaining low gain controllers will be used in the interpolation scheme to enlarge the domain of attraction. Any state $x(k) \in \Omega$ can be decomposed as follows

$$x(k) = \lambda_1(k)\widehat{x}_1(k) + \lambda_2(k)\widehat{x}_2(k) + \ldots + \lambda_r(k)\widehat{x}_r(k) \tag{5.19}$$

where $\widehat{x}_i(k) \in \Omega_i$ for all $i = 1, 2, \ldots, r$ and

$$\sum_{i=1}^{r} \lambda_i(k) = 1, \quad \lambda_i(k) \geq 0$$

Consider the following control law

$$u(k) = \lambda_1(k)K_1\widehat{x}_1(k) + \lambda_2(k)K_2\widehat{x}_2(k) + \ldots + \lambda_r(k)K_r\widehat{x}_r(k) \tag{5.20}$$

where $u_i(k) = K_i\widehat{x}_i(k)$ is the control law, associated to the invariant construction of the set $\Omega_i$. With a slight abuse of notation, denote $x_i = \lambda_i\widehat{x}_i$. Since $\widehat{x}_i \in \Omega_i$, it follows that $x_i \in \lambda_i\Omega_i$ or equivalently that the set of inequalities

$$F_o^{(i)}x_i \leq \lambda_i g_o^{(i)} \tag{5.21}$$

is verified for all $i = 1, 2, \ldots, r$.

It holds that

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) = A(k)\sum_{i=1}^{r} x_i(k) + B(k)\sum_{i=1}^{r} K_i x_i(k) \\ &= \sum_{i=1}^{r} (A(k) + B(k)K_i)x_i(k) \end{aligned}$$

or

$$x(k+1) = \sum_{i=1}^{r} x_i(k+1) \tag{5.22}$$

with $x_i(k+1) = A_{ci}x_i(k)$ and $A_{ci} = (A(k) + B(k)K_i)$.

For the given set of state and control weighting matrices $Q_i \in \mathbb{R}^{n\times n}$, $R_i \in \mathbb{R}^{m\times m}$ with $Q_i \succeq 0$, $R_i \succ 0$, consider the following set of quadratic functions

$$V_i(x_i) = x_i^T P_i x_i, \quad \forall i = 2, 3, \ldots, r \tag{5.23}$$

where $P_i \in \mathbb{R}^{n\times n}$ and $P_i \succ 0$ is chosen to satisfy

$$V_i(x_i(k+1)) - V_i(x_i(k)) \leq -x_i(k)^T Q_i x_i(k) - u_i(k)^T R_i u_i(k) \tag{5.24}$$

Since $x_i(k+1) = A_{ci}x_i(k)$, it follows that

$$A_{ci}^T P_i A_{ci} - P_i \preceq -Q_i - K_i^T R_i K_i$$

By using the Schur complement, one obtains

$$\begin{bmatrix} P_i - Q_i - K_i^T R_i K_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} \succ 0 \tag{5.25}$$

Since matrix $A_{ci}$ has a sub-unitary joint spectral radius, problem (5.25) is always feasible [91]. It is clear that this problem reaches the minimum on one of the vertices of $A_{ci}$. Therefore the set of LMI conditions to be satisfied is following

$$\begin{bmatrix} P_i - Q_i - K_i^T R_i K_i & (A_j + B_j K_i)^T P_i \\ P_i(A_j + B_j K_i) & P_i \end{bmatrix} \succ 0, \quad \forall j = 1, 2, \ldots, q \tag{5.26}$$

One way to obtain matrix $P_i$ is to solve the following LMI problem

$$\min_{P_i} \{trace(P_i)\} \tag{5.27}$$

subject to constraints (5.26).

Define the vector $z \in \mathbb{R}^{(r-1)(n+1)}$ as follows

$$z = [x_2^T \quad \ldots \quad x_r^T \quad \lambda_2 \quad \ldots \quad \lambda_r]^T \tag{5.28}$$

With the vector $z$, consider the following quadratic function

$$V(z) = \sum_{i=2}^{r} x_i^T P_i x_i + \sum_{i=2}^{r} \lambda_i^2 \tag{5.29}$$

We underline the fact that the sum is built on the indices $\{2, 3, \ldots, r\}$, which correspond to the more poorly performing controllers. Simultaneously, the cost function is intended to diminish the influence of these controller actions in the interpolation scheme toward the unconstrained optimum with $\lambda_i = 0$. At each time instant, consider the following optimization problem

$$\min_{z} \{V(z)\} \tag{5.30}$$

subject to the constraints

$$\begin{cases} F_o^{(i)} x_i \leq \lambda_i g_o^{(i)} \\ \sum_{i=1}^{r} x_i = x \\ \sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \geq 0 \end{cases}$$

and apply as input the control action $u = \sum_{i=1}^{r} K_i x_i$.

**Theorem 5.3.** *The interpolation based controller obtained by solving on-line the optimization problem (5.30) guarantees recursive feasibility and robust asymptotic stability for all initial states $x(0) \in \Omega$.*

*Proof.* The proof of this theorem follows the same argumentation as the one of theorem 4.9. Hence it is omitted here. □

As in Section 4.5, the objective function in (5.30) can be rewritten in a quadratic form as

$$\min_{z} \{z^T H z\} \tag{5.31}$$

with

$$H = \begin{bmatrix} P_2 & 0 & \dots & 0 & 0\,0\dots0 \\ 0 & P_3 & \dots & 0 & 0\,0\dots0 \\ \vdots & \vdots & \ddots & \vdots & 0\,0\dots0 \\ 0 & 0 & \dots & P_r & 0\,0\dots0 \\ 0 & 0 & \dots & 0 & 1\,0\dots0 \\ 0 & 0 & \dots & 0 & 0\,1\dots0 \\ \vdots & \vdots & \ddots & \vdots & 0\,0\dots0 \\ 0 & 0 & \dots & 0 & 0\,0\dots1 \end{bmatrix}$$

and the constraints of the optimization problem (5.30) can be rewritten as

$$Gz \leq S + Ex(k) \tag{5.32}$$

where

$$G = \begin{bmatrix} -F_o^{(1)} & -F_o^{(1)} & \dots & -F_o^{(1)} & g_o^{(1)} & g_o^{(1)} & \dots & g_o^{(1)} \\ F_o^{(2)} & 0 & \dots & 0 & -g_o^{(2)} & 0 & \dots & 0 \\ 0 & F_o^{(3)} & \dots & 0 & 0 & -g_o^{(3)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & F_o^{(r)} & 0 & 0 & \dots & -g_o^{(r)} \\ 0 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & -1 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 \end{bmatrix}$$

$$S = \left[ (g_o^{(1)})^T \; 0\,0\dots0\,0\,0\dots0\,1 \right]^T$$

$$E = \left[ -(F_o^{(1)})^T \; 0\,0\dots0\,0\,0\dots0\,0 \right]^T$$

Hence the optimization problem (5.30) is transformed into the quadratic programming problem (5.31) subject to the linear constraints (5.32).

It is worth noticing that for all $x \in \Omega_1$, the QP problem (5.31) subject to the constraints (5.32) has a trivial solution, that is

$$\begin{cases} x_i = 0, \\ \lambda_i = 0, \end{cases} \quad \forall i = 2,3,\dots r$$

Hence $x_1 = x$ and $\lambda_1 = 1$ for all $x \in \Omega_1$. This means that, inside the robustly invariant set $\Omega_1$, the interpolating controller turns out to be the optimal one in the sequence $1,2,\dots,r$.

In summary, an on-line algorithm for the interpolation based controller via quadratic programming is

**Algorithm 5.3: Interpolation based control via quadratic programming**

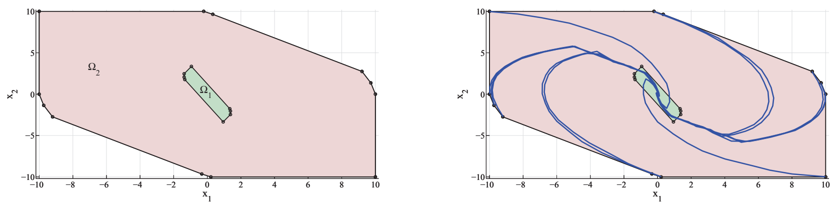
1. Measure the current state of the system $x(k)$.
2. Solve the QP problem (5.31).
3. Implement as input the control action $u = \sum_{i=1}^{r} K_i x_i$.
4. Wait for the next time instant $k := k+1$.
5. Go to step 1 and repeat.

*Example 5.3.* Consider the linear uncertain discrete time system in example (5.1) with the same state and control constraints. Two linear feedback controllers are chosen as

$$\begin{cases} K_1 = [-1.8112 \quad -0.8092] \\ K_2 = [-0.0786 \quad -0.1010] \end{cases} \tag{5.33}$$

The first controller $u(k) = K_1 x(k)$ plays the role of a performance controller, while the second controller $u(k) = K_2 x(k)$ will be used for extending the domain of attraction.

Figure 5.10(a) shows the maximal robustly invariant sets $\Omega_1$ and $\Omega_2$ correspond to the controllers $K_1$ and $K_2$ respectively. Figure 5.10(b) presents different state trajectories of the closed loop system for different initial conditions and different realizations of $\alpha(k)$.



(a) Feasible invariant sets (b) State trajectories

**Fig. 5.10** Feasible invariant sets and different state trajectories of the closed loop system for example 5.3.

The sets $\Omega_1$ and $\Omega_2$ are presented in minimal normalized half-space representation as

$$\Omega_1 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.0000 & 0.0036 \\ -1.0000 & -0.0036 \\ 0.9916 & 0.1297 \\ -0.9916 & -0.1297 \\ 0.8985 & -0.4391 \\ -0.8985 & 0.4391 \\ -0.9130 & -0.4079 \\ 0.9130 & 0.4079 \end{bmatrix} x \le \begin{bmatrix} 1.3699 \\ 1.3699 \\ 1.1001 \\ 1.1001 \\ 2.3202 \\ 2.3202 \\ 0.5041 \\ 0.5041 \end{bmatrix} \right\}$$

$$\Omega_2 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.9352 & 0.3541 \\ -0.9352 & -0.3541 \\ 0.9806 & 0.1961 \\ -0.9806 & -0.1961 \\ -0.5494 & -0.8355 \\ 0.5494 & 0.8355 \\ 1.0000 & 0 \\ 0 & 1.0000 \\ -1.0000 & 0 \\ 0 & -1.0000 \\ -0.6142 & -0.7892 \\ 0.6142 & 0.7892 \end{bmatrix} x \le \begin{bmatrix} 9.5779 \\ 9.5779 \\ 9.8058 \\ 9.8058 \\ 8.2385 \\ 8.2385 \\ 10.0000 \\ 10.0000 \\ 10.0000 \\ 10.0000 \\ 7.8137 \\ 7.8137 \end{bmatrix} \right\}$$

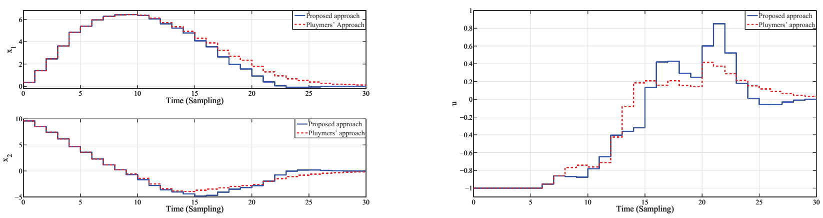
With the weighting matrices

$$Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_2 = 0.001$$

and by solving the LMI problem (5.27), one obtains

$$P_2 = \begin{bmatrix} 17.5066 & 7.9919 \\ 7.9919 & 16.7525 \end{bmatrix}$$

For the initial condition $x(0) = [0.3380 \quad 9.6181]^T$, Figure 5.11(a) and 5.11(b) show the state and input trajectories as a function of time. Figure 5.11(a) and 5.11(b) also show the state and input trajectories, obtained by using algorithm proposed by Pluymers et al. in [123].

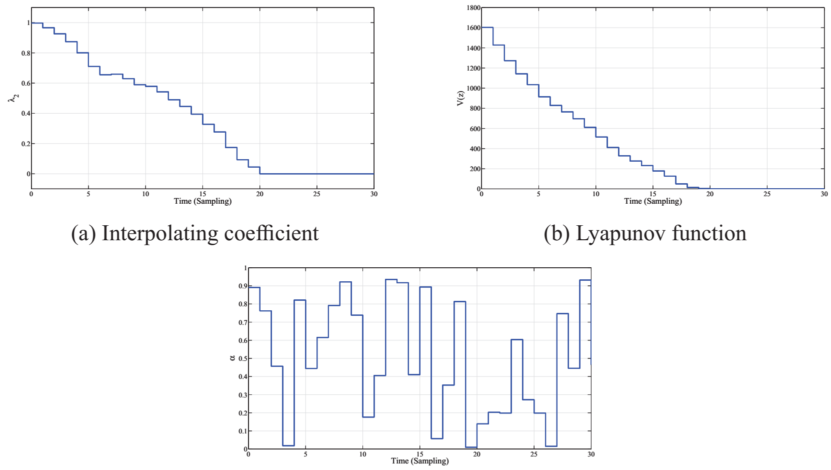



(a) State trajectory (b) Input trajectory

**Fig. 5.11** State and input trajectories of the closed loop system as a function of time for example 5.3. The solid blue line is obtained by using the proposed interpolation based control method, and the dashed red line is obtained by using the method in [123].

The following parameters were used for the approach in [123]. The state and control weighting matrices $Q = 1, R = 0.001$.

Figure 5.12(a), 5.12(b) and 5.12(c) present the interpolating coefficient $\lambda_2(k)$, the objective function i.e. the Lyapunov function and the realization of $\alpha(k)$ as a function of time. It is worth noticing that here $\lambda_2(k)$ is allowed to increase (see for example at time instant $k = 7$).




(a) Interpolating coefficient

(b) Lyapunov function

(c) $\alpha(k)$ realization

**Fig. 5.12** Interpolating coefficient, Lyapunov function and $\alpha(k)$ realization as a function of time for example 5.3.

## 5.4 An improved interpolation based control method in the presence of actuator saturation

In this section, in order to fully utilize the capability of actuators and guarantee the input constraints, a saturation function is applied to the input channel. As in the previous section, we consider the case when $w(k) = 0$ for all $k \geq 0$. For simplicity, only the single input - single output system case is considered here, although extension to the multi-input multi-output system case is straightforward.

Since the saturation function on the input is applied, the dynamical system under consideration is of the form

$$x(k+1) = A(k)x(k) + B(k)sat(u(k)) \tag{5.34}$$

It is assumed that the input constraints are in the form

$$u(k) \in U, U = \{u \in \mathbb{R} : u_l \leq u \leq u_u\} \tag{5.35}$$

where $u_l$ and $u_u$ are respectively the lower and the upper bound of the input $u$. It is also assumed that $u_l$ and $u_u$ are constant with $u_l < 0$ and $u_u > 0$ such that the origin is contained in the interior of $U$. With respect to the the state constraints, their formulation remains the same as in (5.3).

From Lemma 2.1, Section 2.4.1, recall that for a given stabilizing controller $u(k) = Kx(k)$ and for all $x$ such that $u_l \leq Hx \leq u_u$, the saturation function can be expressed as

$$sat(Kx) = \beta(k)Kx(k) + (1-\beta(k))Hx(k) \tag{5.36}$$

with a suitable choice of $0 \leq \beta(k) \leq 1$[4]. The instrumental vector $H \in \mathbb{R}^n$ can be computed based on theorem 2.2. Based on procedure 2.5 in Section 2.4.1, an associated robust polyhedral set $\Omega_s^H$ can be computed, that is invariant for the system

$$x(k+1) = A(k)x(k) + B(k)sat(Kx(k)) \tag{5.37}$$

These design principles can be exploited for a given set of robust asymptotically stabilizing controllers $u(k) = K_i x(k)$ in order to obtain a set of auxiliary vectors $H_i \in \mathbb{R}^n$ with $i = 1,2,\ldots,r$ and a set of robustly invariant sets $\Omega_s^{H_i} \subseteq X$ in the polyhedral form

$$\Omega_s^{H_i} = \{x \in \mathbb{R}^n : F_o^{(i)} x \leq g_o^{(i)}\} \tag{5.38}$$

Let us denote $\Omega_S$ as a convex hull of the sets $\Omega_s^{H_i}$. By the convexity of $X$, it follows that $\Omega_S \subseteq X$, since $\Omega_s^{H_i} \subseteq X$ for all $i = 1,2\ldots,r$.

Any state $x(k) \in \Omega_S$ can be decomposed as follows

$$x(k) = \sum_{i=1}^{r} \lambda_i \widehat{x}_i(k) \tag{5.39}$$

where $\widehat{x}_i(k) \in \Omega_s^{H_i}$ for all $i = 1,2,\ldots,r$ and

$$\sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \geq 0$$

As in the previous section we remark the non-uniqueness of the decomposition. Consider the following control law

$$u(k) = \sum_{i=1}^{r} \lambda_i sat(K_i \widehat{x}_i(k)) \tag{5.40}$$

Based on Lemma 2.1, one obtains

$$u(k) = \sum_{i=1}^{r} \lambda_i (\beta_i K_i + (1-\beta_i)H_i)\widehat{x}_i(k) \tag{5.41}$$

where $0 \leq \beta_i \leq 1$ for all $i = 1,2,\ldots,r$.

---

[4] See Section 2.4.1 for more details.

With the same notation as in the previous sections, let $x_i = \lambda \widehat{x}_i$. Since $\widehat{x}_i \in \Omega_s^{H_i}$, it follows that $x_i \in \lambda \Omega_s^{H_i}$ or

$$F_o^{(i)} x_i \leq \lambda g_o^{(i)}, \quad \forall i = 1, 2, \ldots, r \tag{5.42}$$

From equations (5.39) and (5.41), one gets

$$\begin{cases} x = \sum\limits_{i=1}^{r} x_i \\ u = \sum\limits_{i=1}^{r} (\beta_i K_i + (1-\beta_i) H_i) x_i \end{cases} \tag{5.43}$$

The first high gain controller plays the role of a performance controller, while the remaining low gain controllers will be used to enlarge the domain of attraction. When the control input is of the form (5.43), it is clear that $u(k) \in U$ and $sat(u(k)) = u(k)$ as long as there is no active constraint. It follows that

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)sat(u(k)) = A(k)x(k) + B(k)u(k) \\ &= A(k) \sum_{i=1}^{r} x_i(k) + B(k) \sum_{i=1}^{r} (\beta_i K_i + (1-\beta_i) H_i) x_i(k) \\ &= \sum_{i=1}^{r} x_i(k+1) \end{aligned}$$

with

$$x_i(k+1) = \{A(k) + B(k)(\beta_i K_i + (1-\beta_i) H_i)\} x_i(k)$$

or

$$x_i(k+1) = A_{ci} x_i(k) \tag{5.44}$$

with $A_{ci} = A(k) + B(k)(\beta_i K_i + (1-\beta_i) H_i)$.

For the given set of state and control weighting matrices $Q_i \in \mathbb{R}^{n \times n}$ and $R_i \in \mathbb{R}$, consider the following set of quadratic functions

$$V_i(x_i) = x_i^T P_i x_i, \quad i = 2, 3, \ldots, r \tag{5.45}$$

where matrix $P_i \in \mathbb{R}^{n \times n}$, $P_i \succ 0$ is chosen to satisfy

$$V_i(x_i(k+1)) - V_i(x_i(k)) \leq -x_i(k)^T Q_i x_i(k) - u_i(k)^T R_i u_i(k) \tag{5.46}$$

Denote $Y_i = \beta_i K_i + (1-\beta_i) H_i$. Based on equation (5.44), one can rewrite inequality (5.46) as

$$A_{ci}^T P_i A_{ci} - P_i \preceq -Q_i - Y_i^T R_i Y_i$$

By using the Schur complement, the previous condition can be transformed into

$$\begin{bmatrix} P_i - Q_i - Y_i^T R_i Y_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} \succeq 0$$

or

$$\begin{bmatrix} P_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} - \begin{bmatrix} Q_i + Y_i^T R_i Y_i & 0 \\ 0 & 0 \end{bmatrix} \succeq 0$$

Denote $Q_i^{\frac{1}{2}}$ and $R_i^{\frac{1}{2}}$ as the Cholesky factor of the matrices $Q_i$ and $R_i$, which satisfy $(Q_i^{\frac{1}{2}})^T Q_i^{\frac{1}{2}} = Q_i$ and $(R_i^{\frac{1}{2}})^T R_i^{\frac{1}{2}} = R_i$. The previous condition can be rewritten as

$$\begin{bmatrix} P_i & A_{ci}^T P_i \\ P_i A_{ci} & P_i \end{bmatrix} - \begin{bmatrix} (Q_i^{\frac{1}{2}})^T & Y_i^T (R_i^{\frac{1}{2}})^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_i^{\frac{1}{2}} & 0 \\ R_i^{\frac{1}{2}} Y_i & 0 \end{bmatrix} \succeq 0$$

or by using the Schur complement, one obtains

$$\begin{bmatrix} P_i & A_{ci}^T P_i & (Q_i^{\frac{1}{2}})^T & (R_i^{\frac{1}{2}} Y_i)^T \\ P_i A_{ci} & P_i & 0 & 0 \\ Q_i^{\frac{1}{2}} & 0 & I & 0 \\ R_i^{\frac{1}{2}} Y_i & 0 & 0 & I \end{bmatrix} \succeq 0 \tag{5.47}$$

Clearly, the left-hand side of inequality (5.47) reaches the minimum on one of vertices of $A_{ci}$, $Y_i$, so practically the set of LMI conditions to be checked is the following

$$\begin{cases} \begin{bmatrix} P_i & (A_j + B_j K_i)^T P_i & (Q_i^{\frac{1}{2}})^T & (R_i^{\frac{1}{2}} K_i)^T \\ P_i (A_j + B_j K_i) & P_i & 0 & 0 \\ Q_i^{\frac{1}{2}} & 0 & I & 0 \\ R_i^{\frac{1}{2}} K_i & 0 & 0 & I \end{bmatrix} \succeq 0 \\ \begin{bmatrix} P_i & (A_j + B_j H_i)^T P_i & (Q_i^{\frac{1}{2}})^T & (R_i^{\frac{1}{2}} H_i)^T \\ P_i (A_j + B_j H_i) & P_i & 0 & 0 \\ Q_i^{\frac{1}{2}} & 0 & I & 0 \\ R_i^{\frac{1}{2}} H_i & 0 & 0 & I \end{bmatrix} \succeq 0 \end{cases} \tag{5.48}$$

for all $j = 1, 2, \ldots, q$ and for all $i = 2, 3, \ldots, r$.

Condition (5.48) is linear with respect to the matrix $P_i$. One way to calculate $P_i$ is to solve the following LMI problem

$$\min_{P_i} \{trace(P_i)\} \tag{5.49}$$

subject to constraint (5.48).

Once the matrices $P_i$ with $i = 2, 3, \ldots, r$ are computed, they can be used in practice for real-time control based on the resolution of a low complexity optimization problem.

At each time instant, for a given current state $x$, minimize on-line the following quadratic cost function subject to linear constraints

$$\min_{x_i,\lambda_i}\{\sum_{i=2}^{r} x_i^T P_i x_i + \sum_{i=2}^{r} \lambda_i^2\} \tag{5.50}$$

subject to

$$\begin{cases} F_o^{(i)} x_i \leq \lambda g_o^{(i)}, \forall i = 1,2,\ldots,r \\ \sum\limits_{i=1}^{r} x_i = x \\ \sum\limits_{i=1}^{r} \lambda_i = 1 \\ \lambda_i \geq 0, \forall i = 1,2,\ldots,r. \end{cases}$$

**Theorem 5.4.** *The control law based on solving the optimization problem (5.50) guarantees recursive feasibility and asymptotic stability of the closed loop system for all initial states* $x(0) \in \Omega_S$.

*Proof.* The proof is omitted here, since it is the same as the one of theorem 4.9. □

An on-line algorithm for the interpolation based controller between several saturated controllers via quadratic programming is

**Algorithm 5.4: Interpolation based control via quadratic programming**

---

1. Measure the current state of the system $x(k)$.
2. Solve the QP problem (5.50).
3. Implement as input the control action $u = \sum\limits_{i=1}^{r} \lambda_i sat(K_i x_i)$.
4. Wait for the next time instant $k := k+1$.
5. Go to step 1 and repeat.

*Example 5.4.* We recall the linear uncertain discrete time system

$$x(k+1) = (\alpha(k)A_1 + (1-\alpha(k))A_2)x(k) + (\alpha(k)B_1 + (1-\alpha(k))B_2)u(k) \tag{5.51}$$

in example (5.1) with the same state and control constraints. Two linear feedback controllers in the interpolation scheme are chosen as

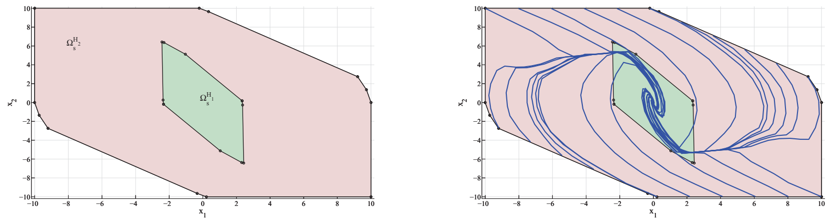
$$\begin{cases} K_1 = [-1.8112 \quad -0.8092], \\ K_2 = [-0.3291 \quad -0.7908] \end{cases} \tag{5.52}$$

Based on theorem 2.2, two auxiliary matrices are computed as

$$\begin{cases} H_1 = [-0.4058 \quad -0.2773] \\ H_2 = [-0.0786 \quad -0.1010] \end{cases} \tag{5.53}$$

With these auxiliary matrices $H_1$ and $H_2$, the robustly invariant sets $\Omega_s^{H_1}$ and $\Omega_s^{H_2}$ are respectively constructed for the saturated controllers $u = sat(K_1 x)$ and $u =$

$sat(K_2x)$, see Figure 5.13(a). Figure 5.13(b) shows different state trajectories of the closed loop system for different initial conditions and different realizations of $\alpha(k)$, obtained by solving the QP problem (5.50).



(a) Feasible invariant sets (b) State trajectories

**Fig. 5.13** Feasible invariant sets and state trajectories of the closed loop system for example 5.4.

The sets $\Omega_s^{H_1}$ and $\Omega_s^{H_2}$ are presented in minimal normalized half-space representation as

$$\Omega_s^{H_1} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.9986 & 0.0532 \\ -0.9986 & -0.0532 \\ 0.9791 & 0.2033 \\ -0.9791 & -0.2033 \\ 0.9999 & 0.0108 \\ -0.9999 & -0.0108 \\ -0.4254 & -0.9050 \\ 0.4254 & 0.9050 \\ -0.6981 & -0.7160 \\ 0.6981 & 0.7160 \\ -0.8256 & -0.5642 \\ 0.8256 & 0.5642 \end{bmatrix} x \leq \begin{bmatrix} 2.3467 \\ 2.3467 \\ 2.3273 \\ 2.3273 \\ 2.3612 \\ 2.3612 \\ 4.7785 \\ 4.7785 \\ 2.9453 \\ 2.9453 \\ 2.0346 \\ 2.0346 \end{bmatrix} \right\}$$

and

$$\Omega_s^{H_2} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 0.9352 & 0.3541 \\ -0.9352 & -0.3541 \\ 0.9806 & 0.1961 \\ -0.9806 & -0.1961 \\ -0.5494 & -0.8355 \\ 0.5494 & 0.8355 \\ 1.0000 & 0 \\ 0 & 1.0000 \\ -1.0000 & 0 \\ 0 & -1.0000 \\ -0.6142 & -0.7892 \\ 0.6142 & 0.7892 \end{bmatrix} x \leq \begin{bmatrix} 9.5779 \\ 9.5779 \\ 9.8058 \\ 9.8058 \\ 8.2385 \\ 8.2385 \\ 10.0000 \\ 10.0000 \\ 10.0000 \\ 10.0000 \\ 7.8137 \\ 7.8137 \end{bmatrix} \right\}$$

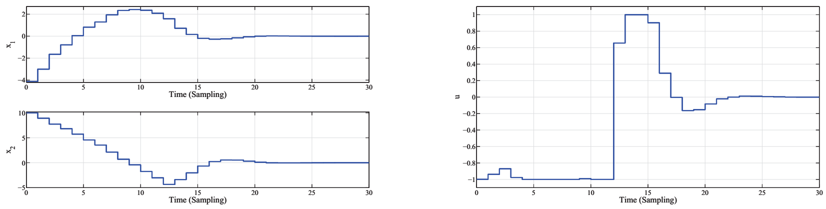
With the weighting matrices

$$Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R_2 = 0.001$$

and by solving the LMI problem (4.77), one obtains

$$P_2 = \begin{bmatrix} 17.7980 & 7.8968 \\ 7.8968 & 16.9858 \end{bmatrix}$$

For the initial condition $x(0) = [-4.1194 \quad 9.9800]^T$, Figure 5.14 shows the state and input trajectory of the closed loop system as a function of time.




(a) State trajectory (b) Input trajectory

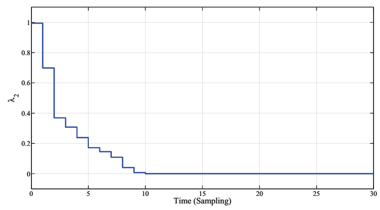
**Fig. 5.14** State and input trajectory of the closed loop system as a function of time for example 5.4.

Figure 5.15 presents the interpolating coefficient $\lambda_2(k)$, the objective function i.e. the Lyapunov function and the realization of $\alpha(k)$ as a function of time.

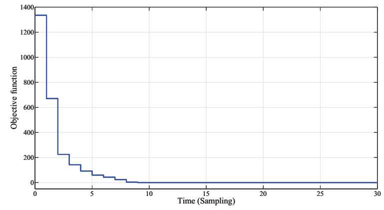
## 5.5 Interpolation via quadratic programming for uncertain systems with disturbances - Algorithm 1

Note that all the development in Sections 5.3 ad 5.4 avoided handling of additive disturbances due to the impossibility of dealing with the robustly asymptotic stability of the origin as an equilibrium point. In this section, an interpolation based control method for system (5.1) with constraints (5.3) using quadratic programming will be proposed to cope with the additive disturbance problem.

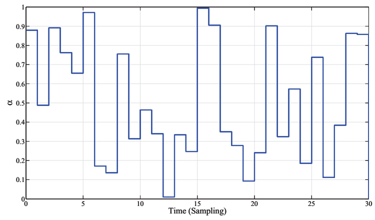
It is clear that when the disturbance is persistent, it is impossible to guarantee the convergence $x(k) \to 0$ as $k \to +\infty$. In other words, it is impossible to achieve asymptotic stability of the closed loop system to the origin. The best that can be hoped for is that the controller steers any initial state to some target set around the origin. Therefore an input-to-state (ISS) stability framework proposed in [61], [99], [88] will be used for characterizing this target region.




(a) Interpolating coefficient




(b) Lyapunov function




(c) $\alpha(k)$ realization

**Fig. 5.15** Interpolating coefficient, Lyapunov function and $\alpha(k)$ realization as a function of time for example 5.4.

### *5.5.1 Input to state stability*

The input to state stability framework provides a natural way to formulate questions of stability with respect to disturbances [145]. This framework attempts to capture the notion of *bounded disturbance input - bounded state*.

Before using the concepts in the specific case of the interpolation schemes, a series of preliminary definitions is introduced.

**Definition 5.2.** **($\mathscr{K}-$function)** A real valued scalar function $\phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class $\mathscr{K}$ if it is continuous, strictly increasing and $\phi(0) = 0$.

**Definition 5.3.** **($\mathscr{K}_\infty-$function)** A function $\phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class $\mathscr{K}_\infty$ if it is a $\mathscr{K}-$function and $\phi(s) \rightarrow +\infty$ as $s \rightarrow +\infty$.

**Definition 5.4.** **($\mathscr{K}\mathscr{L}-$function)** A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class $\mathscr{K}\mathscr{L}$ if for each fixed $k \geq 0$, it follows that $\beta(\cdot, k)$ is a $\mathscr{K}$ function and for each fixed $s \geq 0$, it follows that $\beta(s, \cdot)$ is decreasing and $\beta(s, k) \rightarrow 0$ as $k \rightarrow 0$.

The ISS framework for autonomous discrete-time linear time-varying or uncertain systems, as studied by Jiang and Wang in [61], is briefly reviewed next.

Consider system (5.1) with a feedback controller $u(k) = Kx(k)$ and the corresponding closed loop matrix

$$x(k+1) = A_c(k)x(k) + D(k)w(k) \tag{5.54}$$

where $A_c(k) = A(k) + B(k)K$.

**Definition 5.5. (ISS stability)** The dynamical system (5.54) is ISS with respect to disturbance $w(k)$ if there exist a $\mathscr{K}\mathscr{L}-$function $\beta$ and a $\mathscr{K}-$function $\phi$ such that for all initial states $x(0)$ and for all admissible disturbances $w(k)$, the evolution $x(k)$ of system (5.54) satisfies

$$\|x(k)\| \leq \beta(\|x(0)\|,k) + \phi\left(\sup_{0\leq i\leq k-1} \|w(i)\|\right) \tag{5.55}$$

The function $\phi(\cdot)$ is usually called an ISS gain of system (5.54).

**Definition 5.6. (ISS Lyapunov function)** A function $V:\mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is an ISS Lyapunov function for system (5.54) is there exist $\mathscr{K}_\infty-$functions $\gamma_1$, $\gamma_2$, $\gamma_3$ and a $\mathscr{K}-$function $\theta$ such that

$$\begin{cases} \gamma_1(\|x\|) \leq V(x) \leq \gamma_2(\|x\|) \\ V(x(k+1)) - V(x(k)) \leq -\gamma_3(\|x(k)\|) + \theta(\|w(k)\|) \end{cases} \tag{5.56}$$

**Theorem 5.5.** *System (5.54) is input-to-state stable if it admits an ISS Lyapunov function.*

*Proof.* See [61], [99], [88]. □

*Remark 5.5.* Note that the ISS notion is related to the existence of states $x$ such that

$$\gamma_3(\|x\|) \geq \theta(\|w\|)$$

for all $w \in W$. This implies that there exists a scalar $d \geq 0$ such that

$$\gamma_3(d) = \max_{w\in W} \theta(\|w\|)$$

or $d = \gamma_3^{-1}\left(\max_{w(k)\in W} \theta(\|w(k)\|)\right)$. Here $\gamma_3^{-1}$ denotes the inverse operator of $\gamma_3$.

It follows that for any $\|x(k)\| > d$, one has

$$V(x(k+1)) - V(x(k)) \leq -\gamma_3(\|x(k)\|) + \theta(\|w(k)\|) \leq -\gamma_3(d) + \theta(\|w(k)\|) < 0$$

Thus the trajectory $x(k)$ of the system (5.54) will eventually enter the region $R_x = \{x \in \mathbb{R}^n : \|x(k)\| \leq d\}$. Once inside, the trajectory will never leave this region, due to the monotonicity condition imposed on $V(x(k))$ outside the region $R_x$.

## 5.5.2 Cost function determination

The main contribution presented in the following starts from the assumption that using established results in control theory, one disposes a set of unconstrained robust asymptotically stabilizing feedback controllers $u(k) = K_i x(k)$, $i = 1,2,\ldots,r$, such that for each $i$ the joint spectral radius of the parameter varying matrix $A_{ci}(k)$ is less than one where $A_{ci}(k) = A(k) + B(k)K_i$.

For each controller $u(k) = K_i x(k)$, a maximal robustly positively invariant set $\Omega_i$ can be found in the polyhedral form[5]

$$\Omega_i = \left\{ x \in \mathbb{R}^n : F_o^{(i)} x \leq g_o^{(i)} \right\} \tag{5.57}$$

for all $i = 1, 2, \ldots, r$, such that for all $x(k) \in \Omega_i$, it follows that $x(k+1) \in \Omega_i$ in closed loop with the control law $u(k) = K_i x(k) \in U$ for all $w(k) \in W$. With a slight abuse of notation, denote $\Omega$ as a convex hull of the sets $\Omega_i$, $i = 1, 2, \ldots, r$. It follows that $\Omega \subseteq X$ as a consequence of the fact that $\Omega_i \subseteq X$ for all $i = 1, 2, \ldots, r$.

Any state $x(k) \in \Omega$ can be decomposed as follows

$$x(k) = \sum_{i=1}^{r} \lambda_i(k) \widehat{x}_i(k) \tag{5.58}$$

with $\widehat{x}_i(k) \in \Omega_i$ and

$$\sum_{i=1}^{r} \lambda_i(k) = 1, \quad \lambda_i(k) \geq 0$$

One of the first remark is that according to the cardinal number $r$ and the disposition of the regions $\Omega_i$, the decomposition (5.58) is not unique.

Denote $x_i(k) = \lambda_i(k)\widehat{x}_i(k)$. Equation (5.58) can be rewritten as

$$x(k) = \sum_{i=1}^{r} x_i(k)$$

Hence

$$x_1(k) = x(k) - \sum_{i=2}^{r} x_i(k) \tag{5.59}$$

Since $\widehat{x}_i \in \Omega_i$, it follows that $x_i \in \lambda_i \Omega_i$, or in other words

$$F_o^{(i)} x_i \leq \lambda_i g_o^{(i)} \tag{5.60}$$

Consider the following control law

$$u(k) = \sum_{i=1}^{r} \lambda_i(k) K_i \widehat{x}_i(k) = \sum_{i=1}^{r} K_i x_i(k) \tag{5.61}$$

where $K_i \widehat{x}_i(k)$ is the control law, associated to the construction of the set $\Omega_i$.

From equations (5.59), (5.61), one gets

$$u(k) = K_1 x(k) + \sum_{i=2}^{r} (K_i - K_1) x_i(k) \tag{5.62}$$

[5] See procedure 2.2 in Chapter 2

It holds that

$$
\begin{aligned}
x(k+1) &= A(k)x(k)+B(k)u(k)+D(k)w(k) \\
&= A(k)\sum_{i=1}^{r}\lambda_i(k)\widehat{x}_i(k)+B(k)\sum_{i=1}^{r}\lambda_i(k)K_i\widehat{x}_i(k)+D(k)w(k) \\
&= \sum_{i=1}^{r}\{(A(k)+B(k)K_i)\lambda_i(k)\widehat{x}_i(k)+\lambda_i(k)D(k)w(k)\}
\end{aligned}
$$

or equivalently

$$x(k+1)=\sum_{i=1}^{r}x_i(k+1) \tag{5.63}$$

with

$$x_i(k+1)=A_{ci}(k)x_i(k)+D(k)w_i(k) \tag{5.64}$$

where $A_{ci}(k)=A(k)+B(k)K_i$ and $w_i(k)=\lambda_i w(k)$ for all $i=1,2,\ldots,r$.

From equation (5.63), one obtains

$$x_1(k+1)=x(k+1)-\sum_{i=2}^{r}x_i(k+1)$$

Therefore

$$
\begin{aligned}
x(k+1)-\sum_{i=2}^{r}x_i(k+1) &= A_{c1}(k)x_1(k)+D(k)w_1(k) \\
&= A_{c1}(k)\{x(k)-\sum_{i=2}^{r}x_i(k)\}+D(k)w_1(k)
\end{aligned}
$$

Hence

$$x(k+1)=A_{c1}(k)x(k)+\sum_{i=2}^{r}x_i(k+1)-A_{c1}(k)\sum_{i=2}^{r}x_i(k)+D(k)w_1(k)$$

From equation (5.64), one gets

$$x(k+1)=A_{c1}(k)x(k)+\sum_{i=2}^{r}B(k)(K_i-K_1)x_i(k)+D(k)w(k) \tag{5.65}$$

Equation (5.65) describes the one-step state prediction of system (5.1). Define the vectors $z$ and $\omega$ as

$$
\begin{cases}
z=\begin{bmatrix} x^T & x_2^T & \ldots & x_r^T \end{bmatrix}^T \\
\omega=\begin{bmatrix} w^T & w_2^T & \ldots & w_r^T \end{bmatrix}^T
\end{cases}
$$

Based on equations (5.64), (5.65), one has

$$z(k+1)=\Phi(k)z(k)+\Gamma(k)\omega(k) \tag{5.66}$$

where

$$\Phi(k) = \begin{bmatrix} A_{c1}(k) & B(k)(K_2-K_1) & \dots & B(k)(K_r-K_1) \\ 0 & A_{c2}(k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{cr}(k) \end{bmatrix},$$

$$\Gamma(k) = \begin{bmatrix} D(k) & 0 & \dots & 0 \\ 0 & D(k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D(k) \end{bmatrix}$$

From equation (5.2), it is clear that $\Phi(k)$ and $\Gamma(k)$ can be expressed as a convex combination of $\Phi_j$ and $\Gamma_j$, respectively

$$\begin{cases} \Phi(k) = \sum_{j=1}^{q} \alpha_j(k)\Phi_j \\ \Gamma(k) = \sum_{j=1}^{q} \alpha_j(k)\Phi_j \end{cases} \tag{5.67}$$

where $\sum_{j=1}^{q} \alpha_j(k) = 1$, $\alpha_j(k) \geq 0$ and

$$\Phi_j = \begin{bmatrix} A_j+B_jK_1 & B_j(K_2-K_1) & \dots & B_j(K_r-K_1) \\ 0 & A_j+B_jK_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_j+B_jK_r \end{bmatrix}, \ \Gamma_j = \begin{bmatrix} D_j & 0 & \dots & 0 \\ 0 & D_j & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D_j \end{bmatrix}$$

For the given state and input weighting matrices $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$, $Q \succeq 0$ and $R \succeq 0$, consider the following quadratic function

$$V(z) = z^T P z \tag{5.68}$$

where matrix $\mathscr{P} \succ 0$ is chosen to satisfy

$$V(z(k+1)) - V(z(k)) \leq -x(k)^T Q x(k) - u(k)^T R u(k) + \theta\omega(k)^T\omega(k) \tag{5.69}$$

where $\theta \geq 0$.

Based on equation (5.66), the left hand side of inequality (5.69) can be written as

$$\begin{aligned} V(z(k+1)) - V(z(k)) &= (\Phi z + \Gamma\omega)^T P(\Phi z + \Gamma\omega) - z^T P z \\ &= \begin{bmatrix} z^T & \omega^T \end{bmatrix} \begin{bmatrix} \Phi^T \\ \Gamma^T \end{bmatrix} P \begin{bmatrix} \Phi & \Gamma \end{bmatrix} \begin{bmatrix} z \\ \omega \end{bmatrix} - \begin{bmatrix} z^T & \omega^T \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \omega \end{bmatrix} \end{aligned} \tag{5.70}$$

And the right hand side

$$
\begin{aligned}
&-x(k)^T Qx(k) - u(k)^T Ru(k) + \theta \omega(k)^T \omega(k) \\
&= z(k)^T(-Q_1 - R_1)z(k) + \theta \omega(k)^T \omega(k) \\
&= \begin{bmatrix} z^T & \omega^T \end{bmatrix} \begin{bmatrix} -Q_1 - R_1 & 0 \\ 0 & \theta I \end{bmatrix} \begin{bmatrix} z \\ \omega \end{bmatrix}
\end{aligned} \tag{5.71}
$$

where

$$
Q_1 = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} Q \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix},
$$

$$
R_1 = \begin{bmatrix} K_1^T \\ (K_2 - K_1)^T \\ \vdots \\ (K_r - K_1)^T \end{bmatrix} R \begin{bmatrix} K_1 & (K_2 - K_1) & \dots & (K_r - K_1) \end{bmatrix}
$$

From equations (5.69), (5.70), (5.71) one gets

$$
\begin{bmatrix} \Phi^T \\ \Gamma^T \end{bmatrix} P \begin{bmatrix} \Phi & \Gamma \end{bmatrix} - \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} \preceq \begin{bmatrix} -Q_1 - R_1 & 0 \\ 0 & \theta I \end{bmatrix}
$$

or equivalently

$$
\begin{bmatrix} P - Q_1 - R_1 & 0 \\ 0 & \theta I \end{bmatrix} - \begin{bmatrix} \Phi^T \\ \Gamma^T \end{bmatrix} P \begin{bmatrix} \Phi & \Gamma \end{bmatrix} \succeq 0 \tag{5.72}
$$

Using the Schur complement, equation (5.72) can be brought to

$$
\begin{bmatrix} P - Q_1 - R_1 & 0 & \Phi^T P \\ 0 & \theta I & \Gamma^T P \\ P\Phi & P\Gamma & P \end{bmatrix} \succeq 0 \tag{5.73}
$$

From equation (5.72) it is clear that the problem (5.73) is feasible if and only if the joint spectral radius of matrix $\Phi(k)$ is less than one, or in other words, all matrices $A_{ci}(k)$ are asymptotically stable.

The left hand side of the inequality (5.73) is linear with respect to $\alpha_i(k)$ . Hence it reaches the minimum if and only if $\alpha_i(k) = 0$ or $\alpha_i(k) = 1$. Therefore the set of LMI conditions to be checked is as follows

$$
\begin{bmatrix} P - Q_1 - R_1 & 0 & \Phi_j^T P \\ 0 & \theta I & \Gamma_j^T P \\ P\Phi_j & P\Gamma_j & P \end{bmatrix} \succeq 0 \tag{5.74}
$$

for all $j = 1, 2, \dots, q$.

*Remark 5.6.* The results presented here are based on the common Lyapunov function (5.68) but they can be relaxed by using a parameter dependent Lyapunov function concept, see [36].

Structurally, problem (5.74) is linear with respect to the matrix $P$ and to the scalar $\theta$. It is well known [88] that in the sense of the ISS gain having a smaller $\theta$ is a desirable property. The smallest value of $\theta$ can be found by solving the following LMI optimization problem

$$\min_{P,\theta}\{\theta\} \tag{5.75}$$

subject to constraint (5.74).

### 5.5.3 Interpolation via quadratic programming

Once the matrix $P$ is computed as a solution of the problem (5.75), it can be used in practice for real time control based on the resolution of a low complexity optimization problem with respect to structure and complexity. The resulting control law can be seen as a predictive control type of construction if the function (5.68) is interpreted as an upper bound for a receding horizon cost function.

Define the vector $z_1$ and the matrix $P_1$ as follows

$$z_1 = \begin{bmatrix} x^T \; x_2^T \; \dots \; x_r^T \; \lambda_2 \; \lambda_3 \; \dots \; \lambda_r \end{bmatrix}^T$$
$$P_1 = \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix}$$

With the vectors $z_1$ and matrix $P_1$, at each time instant, for a given current state $x$, minimize on-line the following quadratic cost function

$$V_1(z_1) = \min_{z_1}\{z_1^T P_1 z_1\} \tag{5.76}$$

subject to linear constraints

$$\begin{cases} F_o^{(i)} x_i \le \lambda_i g_o^{(i)}, \forall i = 1,2,\dots,r \\ \sum_{i=1}^{r} x_i = x, \\ \lambda_i \ge 0, \forall i = 1,2,\dots,r \\ \sum_{i=1}^{r} \lambda_i = 1 \end{cases}$$

and implement as input the control action $u = K_1 x + \sum_{i=2}^{r}(K_i - K_1)x_i$.

**Theorem 5.6.** *The control law using interpolation based on the solution of the problem (5.76) guarantees recursive feasibility and the closed loop system is ISS for all initial states* $x(0) \in \Omega$.

*Proof.* Theorem 5.6 stands on two important claims, namely the recursive feasibility and the input-to-state stability. These can be treated sequentially.

*Recursive feasibility:* It has to be proved that $F_u u(k) \leq g_u$ and $x(k+1) \in \Omega$ for all $x(k) \in \Omega$. It holds that

$$\begin{aligned} F_u u(k) &= F_u \sum_{i=1}^{r} \lambda_i(k) K_i \widehat{x}_i(k) = \sum_{i=1}^{r} \lambda_i(k) F_u K_i \widehat{x}_i(k) \\ &\leq \sum_{i=1}^{r} \lambda_i(k) g_u = g_u \end{aligned}$$

and

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) + D(k)w(k) \\ &= \sum_{i=1}^{r} \lambda_i(k)\{(A(k)+B(k)K_i)\widehat{x}_i(k) + D(k)w(k)\} \end{aligned}$$

Since $(A(k)+B(k)K_i)\widehat{x}_i(k) + D(k)w(k) \in \Omega_i \subseteq \Omega$, it follows that $x(k+1) \in \Omega$.

*ISS stability:* From the feasibility proof, it is clear that if $x_i^o(k)$ and $\lambda_i^o(k)$, $i = 1,2,\ldots,r$ are a solution of the optimization problem (5.76) at time instant $k$, then

$$x_i(k+1) = A_{ci}(k)x_i^o(k) + D(k)w_i(k)$$

and $\lambda_i(k+1) = \lambda_i^o(k)$ is a feasible solution at time instant $k+1$. By solving the quadratic programming problem (5.76), one gets

$$V_1(z_1^o(k+1)) \leq V_1(z_1(k+1))$$

and by using inequality (5.69), it follows that

$$\begin{aligned} V_1(z_1^o(k+1)) - V_1(z_1^o(k)) &\leq V_1(z_1(k+1)) - V_1(z_1^o(k)) \\ &\leq -x(k)^T Q x(k) - u(k)^T R u(k) + \theta \omega(k)^T \omega(k) \end{aligned}$$

Hence $V_1(z_1)$ is an ISS Lyapunov function of the system (5.66). It follows that the closed loop system with the interpolation based controller is ISS. □

*Remark 5.7.* Matrix $P$ can be chosen as follows

$$P = \begin{bmatrix} P_{11} & 0 & \ldots & 0 \\ 0 & P_{22} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & P_{rr} \end{bmatrix} \tag{5.77}$$

In this case, the cost function (5.76) can be written by

$$V_1(z_1) = x^T P_{11} x + \sum_{i=2}^{r} x_i^T P_{ii} x_i + \sum_{i=2}^{r} \lambda_i^2$$

Hence, when the current state $x$ is in the set $\Omega_1$, the optimization problem (5.76) has the trivial solution as

$$\begin{cases} x_i = 0, \\ \lambda_i = 0 \end{cases} \forall i = 2,3,\ldots,r$$

and thus $x_1 = x$ and $\lambda_1 = 0$. Therefore, the interpolation based controller turns out to be the optimal unconstrained controller $u = K_1 x$. It follows that the minimal robust positively invariant set $R_\infty$ of the system

$$x(k+1) = (A(k) + B(k)K_1)x(k) + D(k)w(k)$$

is an attractor of the closed loop system with the interpolating controller. In the other words, all trajectories will converge to the set $R_\infty$.

In summary, the interpolation based controller via quadratic programming involves the following steps

**Algorithm 5.5: Interpolation based control via quadratic programming - Algorithm 1**

---

1. Measure the current state of the system $x(k)$.
2. Solve the QP problem (5.76).
3. Implement as input the control action $u = K_1 x + \sum_{i=2}^{r} (K_i - K_1)x_i$.
4. Wait for the next time instant $k := k+1$.
5. Go to step 1 and repeat.

*Example 5.5.* This example is based on a nominal case. Consider the following discrete time system

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) + w(k) \tag{5.78}$$

The constraints on the state variables, the control variable and the disturbances are

$$\begin{cases} -5 \le x_1 \le 5, \quad -5 \le x_2 \le 5, \quad -1 \le u \le 1 \\ -0.1 \le w_1 \le 0.1, \quad -0.1 \le w_2 \le 0.1 \end{cases}$$

Two linear feedback controllers are chosen as

$$\begin{cases} K_1 = [-0.6136 \quad -1.6099] \\ K_2 = [-0.1686 \quad -0.4427] \end{cases} \tag{5.79}$$

The first controller $u = K_1 x$ is an LQ controller with the weighting matrices $Q = I$, $R = 0.01$. The second controller $u = K_2 x$ is used to enlarge the domain of attraction. The sets $\Omega_1$ and $\Omega_2$ are presented in minimal normalized half-space representation as

$$\Omega_1 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.3561 & -0.9344 \\ 0.3561 & 0.9344 \\ 0.7128 & 0.7014 \\ -0.7128 & -0.7014 \end{bmatrix} x \le \begin{bmatrix} 0.5804 \\ 0.5804 \\ 1.4811 \\ 1.4811 \end{bmatrix} \right\}$$

$$\Omega_2 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.0000 & 0 \\ -1.0000 & 0 \\ -0.3559 & -0.9345 \\ 0.3559 & 0.9345 \\ 0.7071 & 0.7071 \\ -0.7071 & -0.7071 \\ -0.2207 & -0.9754 \\ 0.2207 & 0.9754 \\ -0.0734 & -0.9973 \\ 0.0734 & 0.9973 \end{bmatrix} x \leq \begin{bmatrix} 5.0000 \\ 5.0000 \\ 2.1110 \\ 2.1110 \\ 3.4648 \\ 3.4648 \\ 2.2049 \\ 2.2049 \\ 2.7213 \\ 2.7213 \end{bmatrix} \right\}$$

With the weighting matrices $Q = I$ and $R = 0.01$, by solving the optimization problem (5.75) with a block diagonal matrix $P$, one obtains

$$P_1 = \begin{bmatrix} 2.7055 & 1.7145 \\ 1.7145 & 2.8068 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 2.6609 & 3.7471 \\ 3.7471 & 21.9710 \end{bmatrix}$$

and $\theta = 88.0236$.

Figure 5.16 shows the maximal robust positively invariant sets $\Omega_1$, $\Omega_2$, associated with the feedback gains $K_1$ and $K_2$, respectively. This figure also presents state trajectories of the closed loop system for different initial conditions and different realizations of $w(k)$.

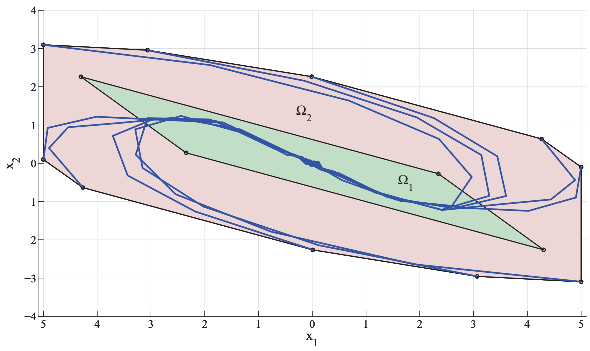



**Fig. 5.16** Feasible invariant sets and state trajectories of the closed loop system for example 5.5.

For the initial condition $x_0 = [-3.0654 \quad 2.9541]^T$, Figure 5.17 and Figure 5.18 present the state and input trajectories of the closed loop system as a function of time. The solid blue line is obtained by using the interpolation based control method and confirms the stabilizing as well as good performances for regulation.

Needless to say the literature on robust MPC for linear systems is very rich nowadays, and one needs to confront the solutions in terms of complexity and performance. In order to compare the proposed technique and the simulation results, we choose one of the most attractive solutions, which is the tube MPC in [87]. The dashed red lines in Figure 5.17 and Figure 5.18 are obtained by using this technique.

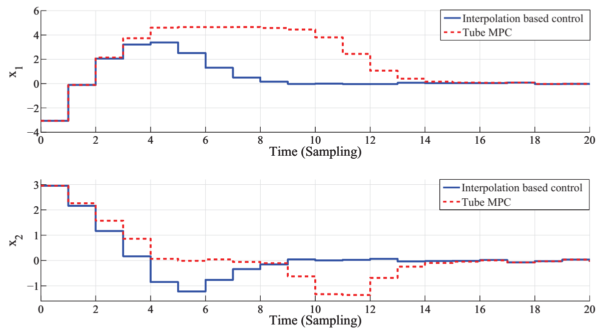



**Fig. 5.17** State trajectories as functions of time for example 5.5. The solid blue line is obtained by using the interpolation based control method, and the dashed red line is obtained by using the tube model predictive control method in [87].

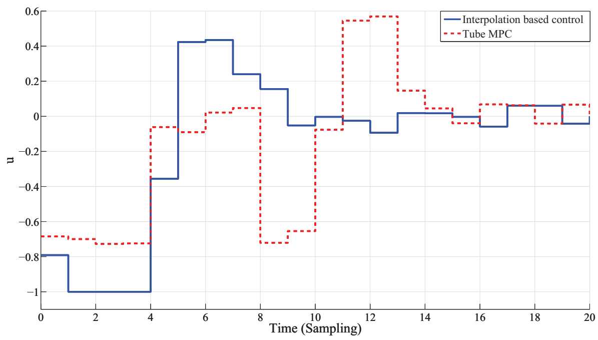



**Fig. 5.18** Input trajectories as functions of time for example 5.5. The solid blue line is obtained by using the interpolation based control method, and the dashed red line is obtained by using the tube model predictive control method in [87].

The following parameters were used for the tube MPC. The minimal robust positively invariant set $R_\infty$ was constructed for system

$$x(k+1) = (A + BK_1)x(k) + w(k)$$

using a method in [127]. The set $R_\infty$ is depicted in Figure 5.19. The setup of the MPC

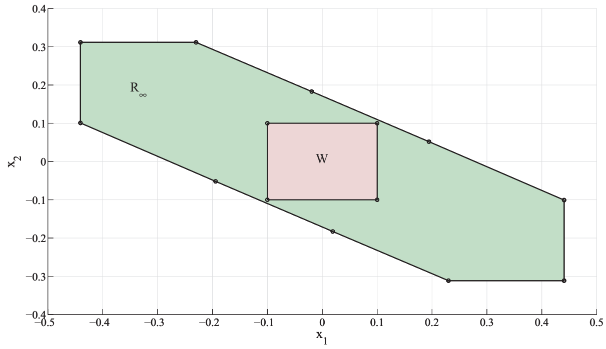
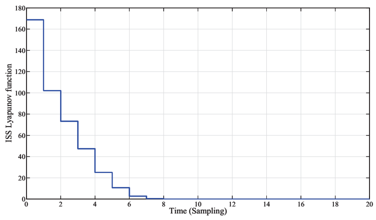


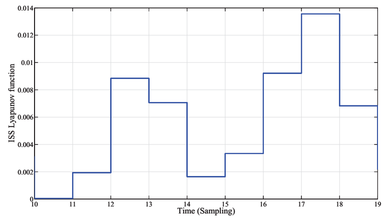

**Fig. 5.19** Minimal invariant set for example 5.5.

approach for the nominal system of the tube MPC framework is $Q = I$, $R = 0.01$. The prediction horizon $N = 10$.

The objective function $V_1(z_1)$ is depicted in Figure 5.20(a). It is worth noticing that $V_1(z_1)$ is only an ISS Lyapunov function. This means, when the state is near to the origin, the function $V_1(z_1)$ might be increasing at some time instants as shown in Figure 5.20(b).



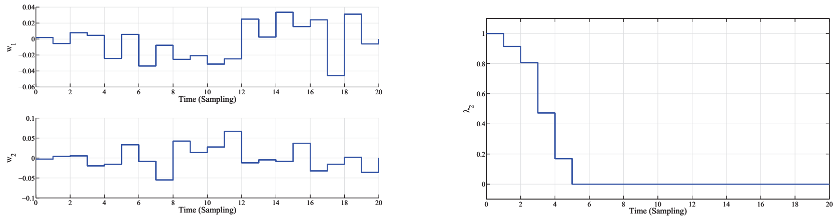

(a) ISS Lyapunov function




(b) Non-decreasing effect of the ISS Lyapunov function

**Fig. 5.20** ISS Lyapunov function and non-decreasing effect of the ISS Lyapunov function as a function of time for example 5.5.

The realization of disturbances $w(k)$ and the interpolating coefficient $\lambda_2$ are respectively, depicted in Figure 5.21(a) and Figure 5.21(b) as a function of time.



(a) $w(k)$ realization (b) Interpolating coefficient $\lambda_2(k)$

**Fig. 5.21** $w(k)$ realization and interpolating coefficient $\lambda_2(k)$ as a function of time for example 5.5.

## 5.6 Interpolation based on quadratic programming for uncertain systems with bounded disturbances - Algorithm 2

In this section, an alternative approach to constrained control of uncertain systems with bounded disturbances will be proposed. Following [130], any state $x$ is decomposed as

$$x = \sum_{i=1}^{r} x_i \tag{5.80}$$

where $x_i \in \mathbb{R}^n$ with $i = 1,2,\dots,r$ are slack variables. The corresponding control value is of the form

$$u = \sum_{i=1}^{r} K_i x_i \tag{5.81}$$

with $K_i \in \mathbb{R}^{m\times n}$ are given such that the joint spectral radius of matrices $A_{ci}(k)$ is sub-unitary where

$$A_{ci}(k) = A(k) + B(k)K_i, \forall i = 1,2,\dots,r$$

One has

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) + D(k)w(k) \\ &= A(k)\sum_{i=1}^{r} x_i(k) + B(k)\sum_{i=1}^{r} K_i x_i(k) + D(k)w(k) \end{aligned}$$

or equivalently

$$x(k+1) = \sum_{i=1}^{r} x_i(k+1)$$

where

$$\begin{cases} x_1(k+1) = (A(k) + B(k)K_1)x_1(k) + D(k)w(k) \\ x_i(k+1) = (A(k) + B(k)K_i)x_i(k), \forall i = 2,3,\dots,r \end{cases} \tag{5.82}$$

Therefore $x_1(k+1) = x(k+1) - \sum\limits_{i=2}^{r} x_i(k+1)$. From the first equation of (5.82), one gets

$$x(k+1) - \sum_{i=2}^{r} x_i(k+1) = A_{c1}(k)\left(x(k) - \sum_{i=2}^{r} x_i(k)\right) + D(k)w(k)$$

or

$$x(k+1) = A_{c1}(k)x(k) + B(k)\sum_{i=2}^{r}(K_i - K_1)x_i(k) + D(k)w(k)$$

Together with the second equation of (5.82), one obtains an augmented system

$$\begin{cases} x(k+1) = A_{c1}(k)x(k) + B(k)\sum\limits_{i=2}^{r}(K_i - K_1)x_i(k) + D(k)w(k) \\ x_i(k+1) = A_{ci}(k)x_i(k), \forall i = 2,3,\ldots,r \end{cases}$$

or in a matrix form

$$\begin{bmatrix} x(k+1) \\ x_2(k+1) \\ \vdots \\ x_r(k+1) \end{bmatrix} = \Lambda(k) \begin{bmatrix} x(k) \\ x_2(k) \\ \vdots \\ x_r(k) \end{bmatrix} + \Xi(k)w(k) \tag{5.83}$$

with

$$\Lambda(k) = \begin{bmatrix} A_{c1}(k) & B(k)(K_2 - K_1) & \ldots & B(k)(K_r - K_1) \\ 0 & A_{c2}(k) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & A_{cr}(k) \end{bmatrix}, \Xi(k) = \begin{bmatrix} D(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Clearly $\Lambda(k)$ and $\Xi(k)$ can be respectively, expressed as a convex combination of $\Lambda_j$ and $\Xi_j$, i.e.

$$\Lambda(k) = \sum_{j=1}^{q} \alpha_j(k)\Lambda_j,\ \Xi(k) = \sum_{j=1}^{q} \alpha_j(k)\Xi_j \tag{5.84}$$

where $\sum\limits_{j=1}^{q} \alpha_j(k) = 1$, $\alpha_j(k) \geq 0$ and

$$\Lambda_j = \begin{bmatrix} A_j + B_jK_1 & B_j(K_2 - K_1) & \ldots & B_j(K_r - K_1) \\ 0 & A_j + B_jK_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & A_j + B_jK_r \end{bmatrix}, \Xi(k) = \begin{bmatrix} D_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The constraints on the augmented state

$$x_s = \begin{bmatrix} x \\ x_2 \\ \vdots \\ x_r \end{bmatrix}$$

are

$$\begin{bmatrix} F_x & 0 & \dots & 0 \\ F_u K_1 & F_u(K_2 - K_1) & \dots & F_u(K_r - K_1) \end{bmatrix} x_s \leq \begin{bmatrix} g_x \\ g_u \end{bmatrix} \tag{5.85}$$

For system (5.83) with constraints (5.85), using procedure 2.2, Chapter 2, one can compute the maximal robust positively invariant set $\Psi_a \subset \mathbb{R}^{rn}$ in the form

$$\Psi_a = \{x_s \in \mathbb{R}^{rn} : F_a x_s \leq g_a\} \tag{5.86}$$

such that for all $x_s(k) \in \Psi_a$, it follows that $x_s(k+1) \in \Psi_a$ and $u(k) = K_1 x(k) + \sum_{i=2}^{r} (K_i - K_1) x_i(k) \in U$. Define $\Psi \in \mathbb{R}^n$ as a set obtained by projecting the polyhedral set $\Psi_a$ onto the state space $x$.

**Theorem 5.7.** *For the given system (5.1), the polyhedral set $\Psi$ is robust controlled positively invariant and admissible with respect to the constraints (5.3).*

*Proof.* Clearly, for all $x(k) \in \Psi$, there exist $x_i(k) \in \mathbb{R}^n$ with $i = 2, 3, \dots, r$ such that

- The augmented state $x_s(k)$ is in $\Psi_a$.
- The control action $u(k) = K_1 x(k) + \sum_{i=2}^{r} (K_i - K_1) x_i(k)$ is in $U$.
- The successor augmented state $x_s(k+1)$ is in $\Psi_a$.

Since $x_s(k+1) \in \Psi_a$, it follows that $x(k+1) \in \Psi$. Hence $\Psi$ is a robust positively invariant set. □

Following the principle of the construction introduced in the Section 5.5, for the given state and input weighting matrices $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$, consider the following quadratic function

$$V(x_s) = x_s^T P x_s \tag{5.87}$$

where the matrix $P \in \mathbb{R}^{rn \times rn}$ and $P \succ 0$ is chosen to satisfy

$$V(x_s(k+1)) - V(x_s(k)) \leq -x(k)^T Q x(k) - u(k)^T R u(k) + \tau w(k)^T w(k) \tag{5.88}$$

The left hand side of inequality (5.88) can be rewritten as

$$\begin{aligned} & V(x_s(k+1)) - V(x_s(k)) \\ & = \begin{bmatrix} x_s^T & w^T \end{bmatrix} \begin{bmatrix} \Lambda^T \\ \Xi^T \end{bmatrix} P \begin{bmatrix} \Lambda & \Xi \end{bmatrix} \begin{bmatrix} x_s \\ w \end{bmatrix} - \begin{bmatrix} x_s^T & w^T \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ w \end{bmatrix} \end{aligned} \tag{5.89}$$

and the right hand side

$$-x(k)^T Qx(k) - u(k)^T Ru(k) + \tau w(k)^T w(k) = \begin{bmatrix} x_s^T & w^T \end{bmatrix} \begin{bmatrix} -Q_1 - R_1 & 0 \\ 0 & \tau I \end{bmatrix} \begin{bmatrix} x_s \\ w \end{bmatrix} \tag{5.90}$$

where

$$Q_1 = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} Q \begin{bmatrix} I & 0 & \ldots & 0 \end{bmatrix},$$

$$R_1 = \begin{bmatrix} K_1^T \\ (K_2^T - K_1^T) \\ \vdots \\ (K_r^T - K_1^T) \end{bmatrix} R \begin{bmatrix} K_1 & (K_2 - K_1) & \ldots & (K_r - K_1) \end{bmatrix}$$

Substituting equations (5.89) and (5.90) into equation (5.88), one gets

$$\begin{bmatrix} \Lambda^T \\ \Xi^T \end{bmatrix} P \begin{bmatrix} \Lambda & \Xi \end{bmatrix} - \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} \preceq \begin{bmatrix} -Q_1 - R_1 & 0 \\ 0 & \tau I \end{bmatrix}$$

or equivalently

$$\begin{bmatrix} P - Q_1 - R_1 & 0 \\ 0 & \tau I \end{bmatrix} - \begin{bmatrix} \Lambda^T \\ \Xi^T \end{bmatrix} P \begin{bmatrix} \Lambda & \Xi \end{bmatrix} \succeq 0$$

or by using the Schur complement, one obtains

$$\begin{bmatrix} P - Q_1 - R_1 & 0 & \Lambda^T P \\ 0 & \tau I & \Xi^T P \\ P\Lambda & P\Xi & P \end{bmatrix} \succeq 0 \tag{5.91}$$

The left hand side of inequality (5.91) reaches the minimum on one of the vertices of $\Lambda(k)$, $\Xi(k)$ so the set of LMI conditions to be satisfied is the following

$$\begin{bmatrix} P - Q_1 - R_1 & 0 & \Lambda_j^T P \\ 0 & \tau I & \Xi_j^T P \\ P\Lambda_j & P\Xi_j & P \end{bmatrix} \succeq 0, \forall j = 1, 2, \ldots, q \tag{5.92}$$

Again, one would like to have the smallest value of $\tau$. This can be done by solving the following LMI optimization problem

$$\min_{P,\tau} \{\tau\} \tag{5.93}$$

subject to constraint (5.92)

Let $P$ be the solution of the problem (5.93). At each time instant for a given current state $x$, minimize on-line the following quadratic cost function subject to linear constraints

$$\min_{x_s}\{x_s^T P x_s\} \tag{5.94}$$

subject to

$$F_a x_s \leq g_a$$

The control input is in the form

$$u = K_1 x + \sum_{i=2}^{r}(K_i - K_1)x_i \tag{5.95}$$

**Theorem 5.8.** *The control law (5.95) where $x_s$ is a solution of the quadratic programming problem (5.94) guarantees recursive feasibility and the closed loop system is ISS for all initial states $x(0) \in \Psi$.*

*Proof.*

*Recursive feasibility:* One has to prove that $u(k) \in U$ and $x(k+1) \in \Psi$ for all $x(k) \in \Psi$.

Since for all $x(k) \in \Psi$, there exist $x_i(k)$ with $i = 2,3,\ldots,r$ such that $x_s(k) \in \Psi_a$. Hence the optimization problem (5.94) is always feasible. From equation (5.95), it follows that

$$u(k) = K_1 x(k) + \sum_{i=2}^{r}(K_i - K_1)x_i(k) \in U$$

With this control input, it holds that $x_s(k+1) \in \Psi_a$. Hence $x(k+1) \in \Psi$.

*ISS stability:* Since matrix $P$ is a solution of the LMI problem (5.93), it is clear that the objective function is an ISS Lyapunov function, which then subsequently guarantees the ISS stability. □

An on-line interpolation based control via quadratic programming is

**Algorithm 5.6: Interpolation based control via quadratic programming**

---

- Measure the current state of the system $x(k)$.
- Solve the QP problem (5.94).
- Implement as input the control action $u = K_1 x + \sum_{i=2}^{r}(K_i - K_1)x_i$.
- Wait for the next time instant $k := k+1$.
- Go to step 1 and repeat.

*Example 5.6.* Consider the following uncertain linear discrete-time system

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k) \tag{5.96}$$

where

$$\begin{aligned} A(k) &= \alpha(k)A_1 + (1-\alpha(k))A_2, \\ B(k) &= \alpha(k)B_1 + (1-\alpha(k))B_2 \end{aligned}$$

and

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$
$$A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

At each sampling time $\alpha(k) \in [0, \quad 1]$ is an uniformly distributed pseudo-random number.

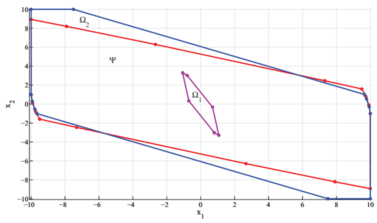
he constraints are

$$-10 \leq x_1 \leq 10, \quad -10 \leq x_2 \leq 10, \quad -1 \leq u \leq 1,$$
$$-0.1 \leq w_1 \leq 0.1, \quad -0.1 \leq w_2 \leq 0.1$$

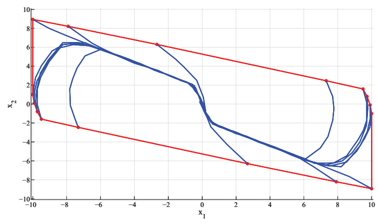
Two feedback controllers are chosen as

$$\begin{cases} K_1 = [-1.8112 \quad -0.8092], \\ K_2 = [-0.0863 \quad -0.1647] \end{cases} \tag{5.97}$$

Figure 5.22(a) presents the robust controlled invariant set $\Psi$, obtained by projecting the augmented invariant set $\Psi_a$ onto the $x$ parameter space. This figure also shows the maximal robustly invariant sets $\Omega_1$ and $\Omega_2$ obtained by using the single controllers $K_1$ and $K_2$. It can be observed that the set $\Psi$ is different from the convex hull of the sets $\Omega_1$ and $\Omega_2$. Figure 5.22(b) presents different state trajectories of the closed loop system for different initial conditions and different realizations of $w(k)$.



(a) Feasible invariant sets



(b) State trajectories

**Fig. 5.22** Feasible invariant sets and state trajectories of the closed loop system for example 5.6.

The set $\Psi$ is presented in minimal normalized half-space representation as

$$\Psi = \left\{ x \in \mathbb{R}^n : \begin{bmatrix} -0.9806 & -0.1961 \\ -0.9578 & -0.2873 \\ -0.3711 & -0.9286 \\ -0.3583 & -0.9336 \\ -0.3428 & -0.9394 \\ -0.3307 & -0.9437 \\ 0.9578 & 0.2873 \\ 0.3711 & 0.9286 \\ 0.3583 & 0.9336 \\ 0.3307 & 0.9437 \\ 0.3428 & 0.9394 \\ 0.9806 & 0.1961 \\ 1.0000 & 0 \\ -1.0000 & 0 \\ 0.9950 & 0.0995 \\ -0.9950 & -0.0995 \end{bmatrix} x \leq \begin{bmatrix} 9.6979 \\ 9.5495 \\ 5.0013 \\ 4.9198 \\ 4.9974 \\ 5.1290 \\ 9.5495 \\ 5.0013 \\ 4.9198 \\ 5.1290 \\ 4.9974 \\ 9.6979 \\ 10.0000 \\ 10.0000 \\ 9.8509 \\ 9.8509 \end{bmatrix} \right\}$$

With the weighting matrices

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1$$

and by solving the optimization problem (5.93) with a block diagonal matrix $P$, one obtains

$$P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}$$

with

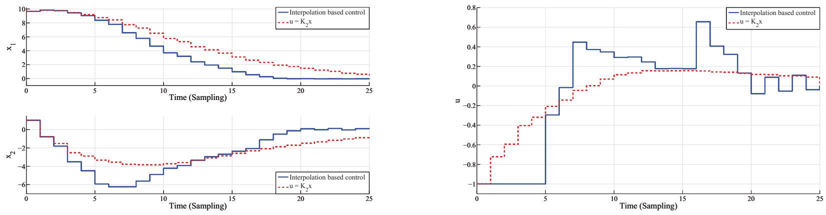
$$P_1 = \begin{bmatrix} 65.9170 & 17.5264 \\ 17.5264 & 7.6016 \end{bmatrix}, \quad P_2 = 10^5 \begin{bmatrix} 1.2839 & 0.7525 \\ 0.7525 & 0.7869 \end{bmatrix},$$

and $\tau = 120.0136$.

For the initial condition $x(0) = [9.6528 \quad 1.0137]^T$, Figure 5.23 shows the state and input trajectories of the closed loop system as a function of time. The solid blue lines are obtained by using interpolation based control method. The dashed red lines are obtained by using the controller $u(k) = K_2x(k)$. From the figures, it is clear that the performance of the closed loop system with the interpolation based controller is better than the closed loop system with the single feedback $u(k) = K_2x(k)$

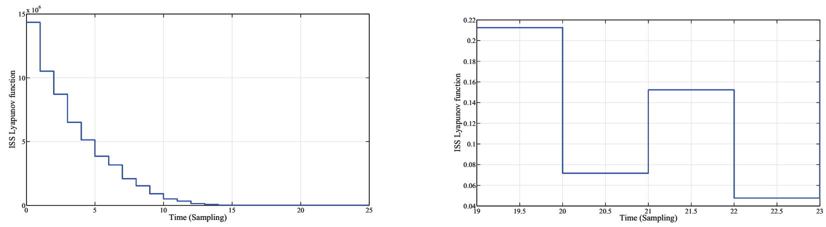
The ISS Lyapunov function and its non-decreasing effect where states near the origin are depicted in Figure 5.24(a) and Figure 5.24(b), respectively.

Figure 5.25 shows the $\alpha(k)$ and $w(k)$ realizations as a function of time.



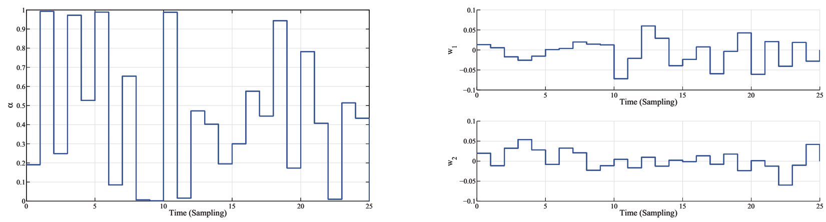

(a) State trajectory (b) Input trajectory

**Fig. 5.23** State and input trajectory of the closed loop system as a function of time for example 5.6. The solid blue lines are obtained by using interpolation based control method. The dashed red lines are obtained by using the controller $u(k) = K_2 x(k)$.




(a) ISS Lyapunov function (b) Non-decreasing effect

**Fig. 5.24** Lyapunov function and its non-decreasing effect as a function of time for example 5.6.




(a) $\alpha(k)$ realization (b) $w(k)$ realization

**Fig. 5.25** $\alpha(k)$ and $w(k)$ realizations as a function of time for example 5.6.

## 5.7 Convex hull of invariant ellipsoids for uncertain systems

### *5.7.1 Interpolation based on LMI*

In this subsection, a set of quadratic functions will be used for estimating the domain of attraction for a constrained discrete-time linear time-varying or uncertain systems. It will be shown that the convex hull of a set of invariant ellipsoids is invariant. The ultimate goal is to design a method for constructing a constrained feedback law based on an interpolation technique for a given set of *saturated* feedback laws.

In the absence of disturbances, the system considered is of the form

$$x(k+1) = A(k)x(k) + B(k)u(k) \tag{5.98}$$

It is assumed that the polyhedral state constraints $X$ and the polyhedral input constraints $U$ are symmetric. Using established result in control theory and theorem 2.2, one obtains a set of asymptotically stabilizing feedback controllers $K_i \in \mathbb{R}^{m\times n}$ and a set of auxiliary matrices $H_i \in \mathbb{R}^{m\times n}$ for $i = 1,2,\ldots,r$ such that the corresponding ellipsoidal invariant sets $E(P_i)$

$$E(P_i) = \left\{x \in \mathbb{R}^n : x^T P_i^{-1} x \le 1\right\} \tag{5.99}$$

is non-empty for $i = 1,2,\ldots,r$. Recall that for all $x(k) \in E(P_i)$, it follows that $sat(K_i x) \in U$ and $x(k+1) = A(k)x(k) + B(k)sat(K_i x(k)) \in X$. Denote $\Omega_E \subset \mathbb{R}^n$ as a convex hull of $E(P_i)$. It follows that $\Omega_E \subseteq X$, since $E(P_i) \subseteq X$.

Any state $x(k) \in \Omega_E$ can be decomposed as follows

$$x(k) = \sum_{i=1}^{r} \lambda_i \widehat{x}_i(k) \tag{5.100}$$

with $\widehat{x}_i(k) \in E(P_i)$ and $\lambda_i$ are interpolating coefficients, that satisfy

$$\sum_{i=1}^{r} \lambda_i = 1, \quad \lambda_i \ge 0$$

Consider the following control law

$$u(k) = \sum_{i=1}^{r} \lambda_i sat(K_i \widehat{x}_i(k)) \tag{5.101}$$

where $sat(K_i \widehat{x}_i(k))$ is the saturated control law, that is feasible in $E(P_i)$.

**Theorem 5.9.** *The control law (5.101) guarantees recursive feasibility for all* $x(0) \in \Omega_E$.

*Proof.* The proof of this theorem is the same as the proof of theorem 4.11 and is omitted here. □

As in the previous Sections, the first feedback gain in the sequence will be used for satisfying performance specifications near the origin, while the remaining gains will be used to enlarge the domain of attraction. For the given current state $x$, consider the following optimization problem

$$\min_{\widehat{x}_i,\lambda_i}\{\sum_{i=2}^{r}\lambda_i\} \tag{5.102}$$

subject to

$$\begin{cases} \widehat{x}_i^T P_i^{-1}\widehat{x}_i \le 1, \forall i = 1,2,\ldots,r \\ \sum\limits_{i=1}^{r}\lambda_i\widehat{x}_i = x \\ \sum\limits_{i=1}^{r}\lambda_i = 1 \\ \lambda_i \ge 0, \forall i = 1,2,\ldots,r \end{cases}$$

**Theorem 5.10.** *The control law using interpolation based on the objective function in (5.102) guarantees robust asymptotic stability for all initial states $x(0) \in \Omega_E$.*

*Proof.* Let $\lambda_i^o$ be the solutions of the optimization problem (5.102) and consider the following positive function

$$V(x) = \sum_{i=2}^{r}\lambda_i^o(k) \tag{5.103}$$

for all $x \in \Omega_E \setminus E(P_1)$. $V(x)$ is a Lyapunov function candidate.

For any $x(k) \in \Omega_E$, one has

$$\begin{cases} x(k) = \sum\limits_{i=1}^{r}\lambda_i^o(k)\widehat{x}_i^o(k) \\ u(k) = \sum\limits_{i=1}^{r}\lambda_i^o(k)sat(K_i\widehat{x}_i^o(k)) \end{cases}$$

It follows that

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) \\ &= A(k)\sum_{i=1}^{r}\lambda_i^o(k)\widehat{x}_i^o(k) + B(k)\sum_{i=1}^{r}\lambda_i^o(k)sat(K_i\widehat{x}_i^o(k)) \\ &= \sum_{i=1}^{r}\lambda_i^o(k)\widehat{x}_i(k+1) \end{aligned}$$

where $\widehat{x}_i(k+1) = A(k)\widehat{x}_i^o(k) + B(k)sat(K_i\widehat{x}_i^o(k)) \in E(P_i)$ for all $i = 1,2,\ldots,r$.

By using the interpolation based on the optimization problem (5.102)

$$x(k+1) = \sum_{i=1}^{r}\lambda_i^o(k+1)\widehat{x}_i^o(k+1)$$

where $\widehat{x}_i^o(k+1) \in E(P_i)$. It follows that

$$\sum_{i=2}^{r}\lambda_i^o(k+1) \le \sum_{i=2}^{r}\lambda_i^o(k)$$

and $V(x)$ is a non-increasing function.

The contractive invariant property of the ellipsoid $E(P_i)$ assures that there is no initial condition $x(0) \in \Omega_E \setminus E(P_1)$ such that $\sum\limits_{i=2}^{r} \lambda_i^o(k+1) = \sum\limits_{i=2}^{r} \lambda_i^o(k)$ for all $k \geq 0$. It follows that $V(x) = \sum\limits_{i=2}^{r} \lambda_i^o(k)$ is a Lyapunov function for all $x \in \Omega_E \setminus E(P_1)$.

The proof is complete by noting that inside $E(P_1)$, the robust stabilizing controller $u = sat(K_1 \widehat{x})$ is contractive and thus the interpolation based controller assures robust asymptotic stability for all $x \in \Omega_E$. □

With a slight abuse of notation, denote $x_i = \lambda_i \widehat{x}_i$. Since $\widehat{x}_i \in E(P_i)$, it follows that $x_i^T P_i^{-1} x_i \leq \lambda_i^2$. The non-linear optimization problem (5.102) can be rewritten as follows

$$\min_{x_i, \lambda_i} \{\sum_{i=2}^{r} \lambda_i\}$$

subject to

$$\begin{cases} x_i^T P_i^{-1} x_i \leq \lambda_i^2, \forall i = 1, 2, \ldots, r \\ \sum\limits_{i=1}^{r} x_i = x \\ \sum\limits_{i=1}^{r} \lambda_i = 1 \\ \lambda_i \geq 0, \forall i = 1, 2, \ldots, r \end{cases}$$

or by using the Schur complement

$$\min_{x_i, \lambda_i} \sum_{i=2}^{r} \lambda_i \tag{5.104}$$

subject to

$$\begin{cases} \begin{bmatrix} \lambda_i & x_i^T \\ x_i & \lambda_i P_i \end{bmatrix} \succeq 0, \forall i = 1, 2, \ldots, r \\ \sum\limits_{i=1}^{r} x_i = x \\ \sum\limits_{i=1}^{r} \lambda_i = 1 \\ \lambda_i \geq 0, \forall i = 1, 2, \ldots, r \end{cases}$$

This can be cast in terms of an LMI optimization. In summary, at each time instant the interpolation based controller involves the following steps

**Algorithm 5.7 Interpolation based control - Convex hull of ellipsoids**

1. Measure the current state of the system $x(k)$.
2. Solve the LMI problem (5.104). In the result, one gets $x_i^o \in E(P_i)$ and $\lambda_i^o$ for all $i = 1, 2, \ldots, q$.
3. For $x_i^o \in E(P_i)$, one associates the control value $u_i^o = sat(K_i x_i^o)$.
4. The control value $u(k)$ is found as a convex combination of $u_i^o$

$$u(k) = \sum_{i=1}^{r} \lambda_i^o(k) u_i^o$$

## 5.7.2 *Geometrical properties of the solution*

The aim of this section is to highlight the properties of the solution of the optimization problem (5.104).

Define a vector $x_{2r} \in \mathbb{R}^n$ and a set $\Omega_{2r} \subset \mathbb{R}^n$ as follows

$$\begin{cases} x_{2r} = \sum_{i=2}^{r} \lambda_i x_i \\ \Omega_{2r} = \text{Convex hull}(E(P_i)),\ i = 2, 3, \ldots, r \end{cases}$$

The following theorem holds

**Theorem 5.11.** *For all $x(k) \notin E(P_1)$, the solution of the optimization problem (5.104) is reached if $x(k)$ is written as a convex combination of two points $x_1^o$ and $x_{2r}^o$, where $x_1^o \in Fr(E(P_1))$ and $x_{2r}^o \in Fr(\Omega_{2r})$.*

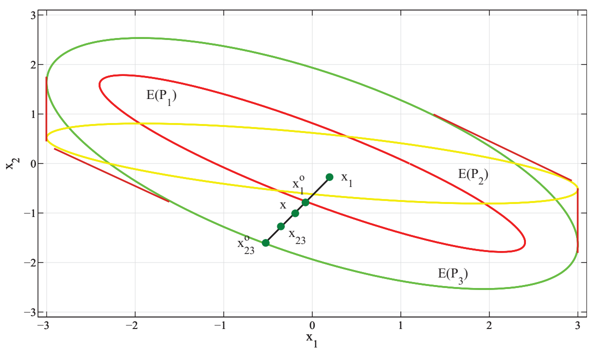



**Fig. 5.26** Graphical illustration of the construction related to theorem 5.11.

*Proof.* Suppose that $x$ is decomposed as $x = \lambda_1 x_1 + \lambda_{2r} x_{2r}$, where $x_1 \in E(P1)$ and $x_{2r} \in \Omega_{2r}$. If $x_{2r}$ is strictly inside $\Omega_{2r}$, by setting $x_{2r}^o = \text{Fr}(\Omega_{2r}) \cap \overline{x, x_{2r}}$ (the intersection between the boundary of $\Omega_{2r}$ and the line connecting $x$ and $x_{2r}$), one has $x = \lambda_1^o x_1^o + \lambda_{2r}^o x_{2r}^o$ with $\lambda_{2r}^o < \lambda_{2r}$ which leads to a contradiction from the optimization point of view. Thus the first conclusion is that in general terms, for the optimal solution one has $(x_1^o, x_{2r}^o)$, where $x_{2r}^o \in \text{Fr}(\Omega_{2r})$.

Analogously, if $x_1$ is strictly inside $E(P_1)$, by setting $x_1^o = \text{Fr}(E(P_1)) \cap \overline{x, x_1}$ (the intersection between the boundary of $E(P_1)$ and the line connecting $x$ and $x_1$) one obtains $x = \lambda_1^o x_1^o + \lambda_{2r}^o x_{2r}^o$ where $\lambda_1^o \geq \lambda_1$ and $\lambda_{2r}^o \leq \lambda_{2r}$. This is again a contradiction leading to the conclusion that for the optimal solution $(x_1^o, x_{2r}^o)$, one has $x_1^o \in E(P_1)$. □

*Remark 5.8.* For all $x(k) \in E(P_1)$ the result of the optimization problem (5.104) has a trivial solution $x_1(k) = x(k)$ and thus $\lambda_1 = 1$ and $\lambda_{2r} = 0$.

For all $x(k) \notin E(P_1)$, the fact that $x_{2r}^o$ belongs to the boundary of $\Omega_{2r}$ implies that either $x_i^o \in \text{Fr}(E(P_i))$ or $x_i^o = 0$. Or by denoting $\widehat{x}_i = \lambda_i x_i$, one concludes that the optimal solutions of the problem (5.104) satisfy

$$\widehat{x}_i^T P_i^{-1} \widehat{x}_i = \lambda_i^2, \forall i = 2, 3, \ldots, r$$

and for all $x(k) \notin E(P_1)$

$$\widehat{x}_1^T P_1^{-1} \widehat{x}_1 = \lambda_1^2$$

Hence for all $x(k) \notin E(P_1)$, the optimal solution of the problem (5.104) satisfy

$$\widehat{x}_i^T P_i^{-1} \widehat{x}_i = \lambda_i^2, \forall i = 1, 2, \ldots, r$$

*Example 5.7.* Consider the uncertain linear discrete-time system in example (5.1) with the same state and control constraints. Two linear feedback controllers is chosen as

$$\begin{cases} K_1 = [-0.6451 \quad -0.7740], \\ K_2 = [-0.2416 \quad -0.7824] \end{cases} \tag{5.105}$$

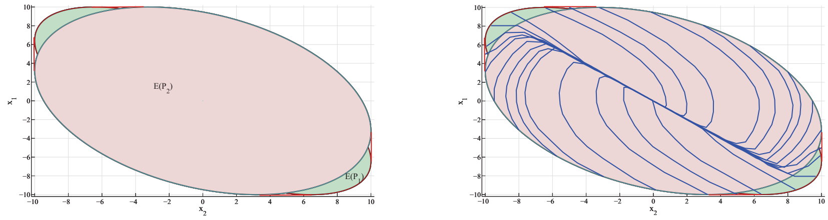
Based on theorem 2.2, two auxiliary matrices are defined as

$$\begin{cases} H_1 = [-0.0816 \quad -0.1186] \\ H_2 = [-0.0884 \quad -0.0683] \end{cases} \tag{5.106}$$

With the auxiliary matrices $H_1$ and $H_2$, two invariant ellipsoids $E(P_1)$ and $E(P_2)$ are respectively constructed for the saturated controllers $u = sat(K_1 x)$ and $u = sat(K_2 x)$, see Figure 5.27(a). Figure 5.27(b) shows the state trajectories of the closed loop system for different initial conditions and different realization of $\alpha(k)$.

The matrices $P_1$ and $P_2$ are

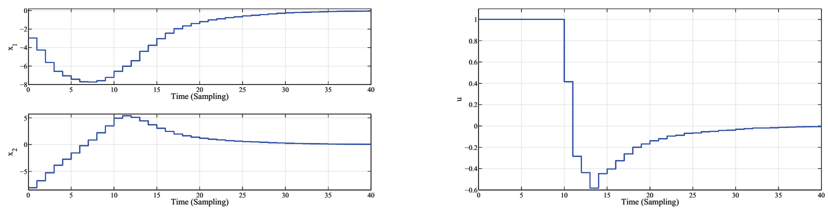
$$P_1 = \begin{bmatrix} 100.0000 & -64.4190 \\ -64.4190 & 100.00000 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 100.0000 & -32.2659 \\ -32.2659 & 100.0000 \end{bmatrix}$$




(a) Feasible invariant sets (b) State trajectories

**Fig. 5.27** Feasible invariant sets and state trajectories of the closed loop system for example 5.7.

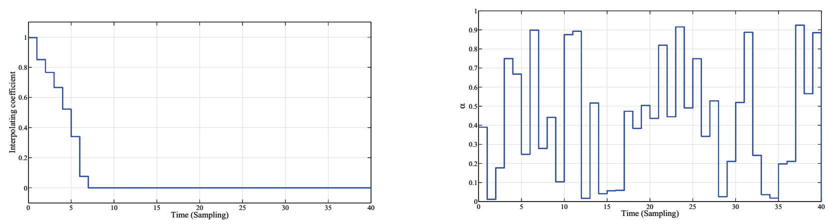
For the initial condition $x(0) = [-2.96 \quad -8.08]^T$, Figure 5.28 shows the state and input trajectory of the closed loop system as a function of time.




(a) State trajectory (b) Input trajectory

**Fig. 5.28** State and input trajectory of the closed loop system as a function of time for example 5.7.

Figure 5.29 presents the interpolating coefficient $\lambda_2(k)$ as a Lyapunov function and the realization of $\alpha(k)$ as a function of time. As expected, the function $\lambda_2(k)$ is positive and non-increasing.




(a) Interpolating coefficient (b) $\alpha(k)$ realization

**Fig. 5.29** Interpolating coefficient and $\alpha(k)$ realization as a function of time for example 5.7.

# Chapter 6
# Interpolation Based Control – Output Feedback Case

So far, in this manuscript, state feedback control problems have been considered. However, in practice, direct information (measurement) of the complete state of dynamic systems may not be available. In this case, an observer could possibly be used for the state estimation. A serious drawback of the observer-based approaches is the observer error, which one has to include in the uncertainty. In addition, whenever the constraints become active, the nonlinearity dominates the properties of the state feedback control system and one cannot expect the separation principle to hold. Moreover there is no guarantee that the constraints will be satisfied along the closed-loop trajectories.

In the chapter, we revisit the problem of state reconstruction through measurement and storage of appropriate previous measurements. Even if this model might be *non-minimal*, it is directly measurable and will provide an appropriate model for the control design with constraint handling guarantees. Finally it will be shown how the interpolation-based control principles can lead to an output-feedback control design procedure.

## 6.1 Problem formulation

Consider the problem of regulating to the origin the following discrete-time linear time-varying or uncertain system, described by the input-output relationship

$$\begin{aligned} &y(k+1)+E_1y(k)+E_2y(k-1)+\ldots+E_sy(k-s+1) \\ &\quad = N_1u(k)+N_2u(k-1)+\ldots+N_ru(k-r+1)+w(k) \end{aligned} \tag{6.1}$$

where $y(k) \in \mathbb{R}^p$, $u(k) \in \mathbb{R}^m$ and $w(k) \in \mathbb{R}^p$ are respectively the output, the input and the disturbance vector. The matrices $E_i$ for $i = 1,\ldots,s$ and $N_i$ for $i = 1,\ldots,r$ have suitable dimensions.

For simplicity, it is assumed that $s = r$. The matrices $E_i$ and $N_i$ for $i = 1,2,\ldots,s$ satisfy

$$\Gamma = \begin{bmatrix} E_1 & E_2 & \dots & E_s \\ N_1 & N_2 & \dots & N_s \end{bmatrix} = \sum_{i=1}^{q} \alpha_i(k)\Gamma_i \tag{6.2}$$

where $\alpha_i(k) \geq 0$ and $\sum_{i=1}^{q} \alpha_i(k) = 1$ and

$$\Gamma_i = \begin{bmatrix} E_1^i & E_2^i & \dots & E_s^i \\ N_1^i & N_2^i & \dots & N_s^i \end{bmatrix}$$

are the extreme realizations of a polytopic model.

The output and control vectors are subject to the following hard constraints

$$\begin{cases} y(k) \in Y, & Y = \{y \in \mathbb{R}^p : F_y y \leq g_y\} \\ u(k) \in U, & U = \{u \in \mathbb{R}^m : F_u u \leq g_u\} \end{cases} \tag{6.3}$$

where $Y$ and $U$ are C-sets. It is assumed that the disturbance $w(k)$ is unknown, additive and lie in the polytope $W$, i.e. $w(k) \in W$, where $W = \{w \in \mathbb{R}^p : F_w w \leq g_w\}$ is a C-set.

## 6.2 Output feedback - Nominal case

In this section, we consider the case when the matrices $E_j$ and $N_j$ for $j = 1, 2, \dots, s$ are known and fixed. The case when $E_j$ and $N_j$ for $j = 1, 2, \dots, s$ are unknown or time-varying will be treated in the next section.

A state space representation will be constructed along the lines of [152]. All the steps of the construction are detailed such that the presentation of the results are self contained. The state of the system is chosen as a vector of dimension $p \times s$ with the following components

$$x(k) = \left[ x_1(k)^T \; x_2(k)^T \; \dots \; x_s(k)^T \right]^T \tag{6.4}$$

where

$$\begin{cases} x_1(k) = y(k) \\ x_2(k) = -E_s x_1(k-1) + N_s u(k-1) \\ x_3(k) = -E_{s-1} x_1(k-1) + x_2(k-1) + N_{s-1} u(k-1) \\ x_4(k) = -E_{s-2} x_1(k-1) + x_3(k-1) + N_{s-2} u(k-1) \\ \vdots \\ x_s(k) = -E_2 x_1(k-1) + x_{s-1}(k-1) + N_2 u(k-1) \end{cases} \tag{6.5}$$

The subcomponents of the state vector can be interpreted exclusively in terms of the input and output contributions as

$$\begin{aligned}
x_2(k) &= -E_s y(k-1) + N_s u(k-1) \\
x_3(k) &= -E_{s-1} y(k-1) - E_s y(k-2) + N_{s-1} u(k-1) + N_s u(k-2) \\
&\vdots \\
x_s(k) &= -E_2 y(k-1) - E_3 y(k-2) - \ldots - E_s y(k-s+1) + \\
&\quad + N_2 u(k-1) + N_3 u(k-2) + \ldots + N_s u(k-s+1)
\end{aligned}$$

One has

$$\begin{aligned}
y(k+1) = &-E_1 y(k) - E_2 y(k-1) - \ldots - E_s y(k-s+1) \\
&+ N_1 u(k) + N_2 u(k-1) + \ldots + N_s u(k-s+1) + w(k)
\end{aligned}$$

or, equivalently

$$x_1(k+1) = -E_1 x_1(k) + x_s(k) + N_1 u(k) + w(k)$$

The state space model is then defined in a compact linear difference equation form as follows

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + Dw(k) \\ y(k) = Cx(k) \end{cases} \tag{6.6}$$

where

$$A = \begin{bmatrix} -E_1 & 0 & 0 & \ldots & 0 & I \\ -E_s & 0 & 0 & \ldots & 0 & 0 \\ -E_{s-1} & I & 0 & \ldots & 0 & 0 \\ -E_{s-2} & 0 & I & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -E_2 & 0 & 0 & \ldots & I & 0 \end{bmatrix}, \quad B = \begin{bmatrix} N_1 \\ N_s \\ N_{s-1} \\ N_{s-2} \\ \vdots \\ N_2 \end{bmatrix}, \quad D = \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$C = \begin{bmatrix} I & 0 & 0 & 0 & \ldots & 0 \end{bmatrix}$$

The model (6.6) has been elaborated such that the state to be available by simple storage of input values and output signal measurements. One natural question is if this important advantage will be paid in terms of dimensions. In comparison with classical state space representations, the model (6.6) is *minimal* in the single-input single-output case. However, in the multi-input multi-output cases, this realization might not be minimal, as shown in the following example.

Consider the following single-input multi-output discrete-time system

$$y(k+1) + \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} y(k) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} y(k-1) = \begin{bmatrix} 0.5 \\ 2 \end{bmatrix} u(k) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u(k-1) + w(k) \tag{6.7}$$

Using the above construction, the state space model is given as follows

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) + Dw(k) \\ y(k) &= Cx(k) \end{cases}$$

where

$$A = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ -1.5 \end{bmatrix}, \quad E = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix},$$
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

This realization is not minimal, since it unnecessarily replicates the common poles of the denominators in the input-output description. There exists an alternative lower dimensional construction like

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 2 \end{bmatrix}, B = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Denote

$$z(k) = [y(k)^T \ \dots \ y(k-s+1)^T \ u(k-1)^T \ \dots \ u(k-s+1)^T \,]^T \tag{6.8}$$

Based on equation (6.4) the state vector $x(k)$ is related to the vector $z(k)$ as follows

$$x(k) = Tz(k) \tag{6.9}$$

where

$$T = [T_1 \quad T_2]$$
$$T_1 = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ 0 & -E_s & 0 & \dots & 0 \\ 0 & -E_{s-1} & -E_s & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -E_2 & -E_3 & \dots & -E_s \end{bmatrix}, \quad T_2 = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ N_s & 0 & 0 & \dots & 0 \\ N_{s-1} & N_s & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ N_2 & N_3 & N_4 & \dots & N_s \end{bmatrix}$$

From equation (6.9), it becomes obvious that at any time instant $k$, the state variable vector is available exclusively though measurement and storage of appropriate previous measurements.

Our main objective remains the treatment of the constraints (6.3). After simple set manipulations, these can be translated in state constraints of the type $x_i(k) \in X_i$ where $X_i$ is given by

$$\begin{cases} X_1 = Y \\ X_2 = E_s(-X_1) \oplus N_s U \\ X_i = E_{s+2-i}(-X_1) \oplus X_{i-1} \oplus N_{s+2-i} U, \quad \forall i = 3, \dots, s \end{cases} \tag{6.10}$$

In summary, the constraints on the state are $x \in X$, where $X = \{x : \ F_x x \le g_x\}$.

*Example 6.1.* Consider the following discrete-time system

$$y(k+1) - 2y(k) + y(k-1) = 0.5u(k) + 0.5u(k-1) + w(k) \tag{6.11}$$

The constraints on output and input and on disturbance are

$$\begin{cases} -5 \leq y(k) \leq 5 \\ -5 \leq u(k) \leq 5 \end{cases}$$

and

$$-0.1 \leq w(k) \leq 0.1$$

The state space model is given by

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + Dw(k) \\ y(k) = Cx(k) \end{cases}$$

where

$$A = \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad E = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

and

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The state $x(k)$ is available though the measured input, output and their past measured values as follows

$$x(k) = Tz(k)$$

where

$$z(k) = \begin{bmatrix} y(k) & y(k-1) & u(k-1) \end{bmatrix}^T,$$
$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0.5 \end{bmatrix}$$

The constraints on the state according to (6.10) are

$$\begin{cases} -5 \leq x_1 \leq 5 \\ -7.5 \leq x_2 \leq 7.5 \end{cases}$$

Using the linear quadratic regulator with the weighting matrices

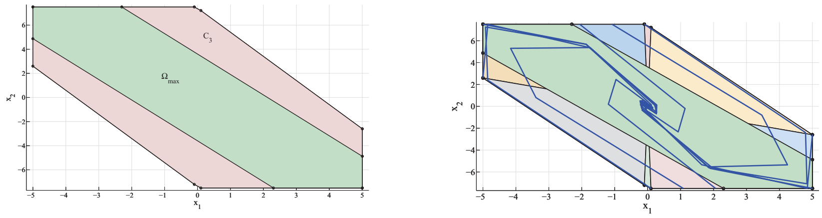
$$Q = C^T C = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = 0.1$$

as the local controller, the feedback gain is obtained

$$K = \begin{bmatrix} -2.3548 & -1.3895 \end{bmatrix}$$

Algorithm 5.1 in Section 5.2 will be employed with the global vertex controller in this example. Using procedures 2.2 and 2.3 Chapter 2, one obtains the set $\Omega_{max}$ and $C_N$ as shown in Figure 6.1(a). Note that $C_3 = C_4$, in this case $C_3$ is a maximal invariant set for system (6.11). Figure 6.1(b) presents different state trajectories for different initial conditions and different realizations of $w(k)$.

The set of vertices of $C_N$ is given by the matrix $V(C_N)$ below, together with the control matrix $U_v$



(a) Feasible invariant sets (b) State trajectories

**Fig. 6.1** Feasible invariant sets and state trajectories for example 6.1.

$$V(C_N) = \begin{bmatrix} -5 & -0.1 & 5 & 0.1 & -0.1 & -5 & 0.1 & 5 \\ 7.5 & 7.5 & -2.6 & 7.2 & -7.2 & 2.6 & -7.5 & -7.5 \end{bmatrix}$$

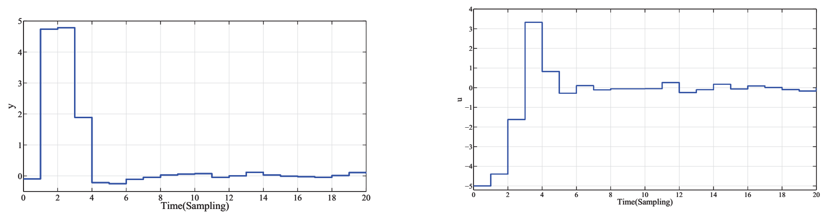
and

$$U_v = \begin{bmatrix} -5 & -5 & -5 & -4.9 & 5 & 5 & 5 & 4.9 \end{bmatrix}$$

The set $\Omega_{max}$ is presented in minimal normalized half-space representation as

$$\Omega_{max} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} 1.0000 & 0 \\ 0 & 1.0000 \\ -1.0000 & 0 \\ 0 & -1.0000 \\ -0.8612 & -0.5082 \\ 0.8612 & 0.5082 \end{bmatrix} x \leq \begin{bmatrix} 5.0000 \\ 7.5000 \\ 5.0000 \\ 7.5000 \\ 1.8287 \\ 1.8287 \end{bmatrix} \right\}$$

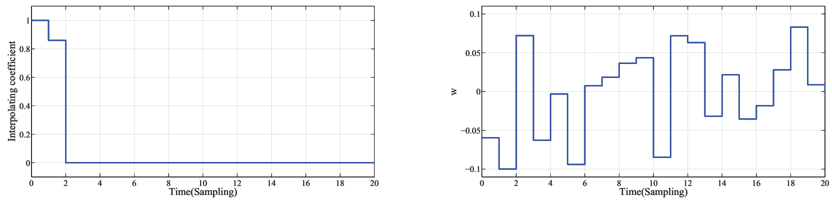
For the initial condition $x(0) = [-0.1000 \quad 7.5000]^T$, Figure 6.2 shows the output and input trajectory of the closed loop system as a function of time.



(a) Output trajectory (b) Input trajectory

**Fig. 6.2** Output and input trajectory of the closed loop system for example 6.1.

The interpolating coefficient and the realization of $w(k)$ as a function of time are depicted in Figure 6.3. As expected the interpolating coefficient, i.e. the Lyapunov function is positive and non-increasing.




(a) Interpolating coefficient (b) $w(k)$ realization

**Fig. 6.3** Interpolating coefficient and realization of $w(k)$ for example 6.1.

In order to provide a term of comparison for the present approach, we present a solution based on the well-known steady state Kalman filter. Figure 6.4(a) shows the output trajectories using the constrained output feedback approach and the Kalman filter + constrained state feedback approach. It is obvious that, the minimal robust positively invariant set for the Kalman filter based approach is larger than the minimal robust positively invariant set of the approach, presented in this section.

For the sake of completeness of the comparison, we mention that, the Matlab routine with the command *'kalman'* was used for designing the Kalman filter. The process noise is a white noise with an uniform distribution and no measurement noise was considered.

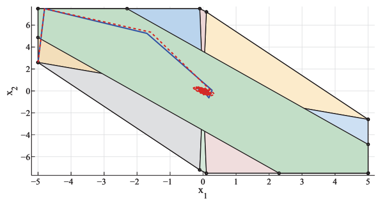
The disturbance $w$ is a random number with an uniform distribution, $w_l \leq w \leq w_u$ where $w_l = -0.1$ and $w_u = 0.1$. The variance of $w$ is given as

$$C_w = \frac{(w_u - w_l + 1)^2 - 1}{12} = 0.0367$$

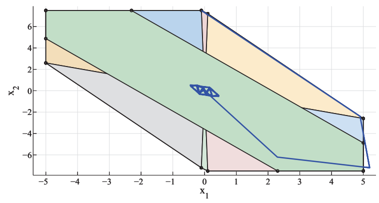
The estimator gain of the Kalman filter is obtained as

$$L = [2 \quad -1]^T$$

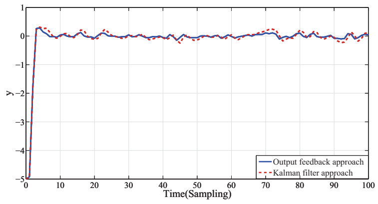
The Kalman filter is used to estimate the state of the system and then this estimation is used to close the loop with the interpolated control law. In contrast to the output feedback approach, where the state is exact with respect to the measurement, in the Kalman filter approach, an extra level of uncertainty is introduced around the state trajectory by mixing the additive disturbances in the estimation process. Thus there is no guarantee that the constraints are satisfied in the transitory stage. This constraint violation effect is shown in Figure 6.4(b). Figure 6.4(c) presents the output trajectories of our approach and the Kalman filter based approach.




(a) State trajectories




(b) Constraints violation




(c) Output trajectories

**Fig. 6.4** Comparison between the output feedback approach and the Kalman filter based approach for example 6.1.

## 6.3 Output feedback - Robust case

A weakness of the approach in Section 6.2 is that the state measurement is available if and only if the parameters of the system are known. For uncertain or time-varying system, that is not the case. In this section, we provide another method for constructing the state variables, that do not use the information of the system parameter. Based on the measured plant input, output and their past measured values, the state of the system (6.1) is chosen as

$$x(k) = [y(k)^T \ \dots \ y(k-s+1)^T \ u(k-1)^T \ \dots \ u(k-s+1)^T]^T \tag{6.12}$$

The state space model is then defined as follows

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) + Dw(k) \\ y(k) &= Cx(k) \end{cases} \tag{6.13}$$

where

$$A = \begin{bmatrix} -E_1 & -E_2 & \dots & -E_s & N_2 & \dots & N_{s-1} & N_s \\ I & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & O & I & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & O & 0 & \dots & I & 0 \end{bmatrix}, B = \begin{bmatrix} N_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix}, D = \begin{bmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} I \; 0 \; 0 \dots 0 \; 0 \; 0 \dots 0 \end{bmatrix}$$

From equation (6.2), it is clear that matrices $A$ and $B$ belong to a polytopic set

$$(A,B) \in \Delta \tag{6.14}$$

where

$$\Delta = \text{Convex hull}\{(A_1,B_1),(A_2,B_2)\dots,(A_q,B_q)\}$$

The vertices $(A_i,B_i)$ are obtained from the vertices of (6.2).

Although the obtained representation is non-minimal, it has the merit that the original output-feedback problem for the uncertain plant has been transformed into a state-feedback problem where the matrices $A$ and $B$ lie in the polytope defined by (6.14) without any additional uncertainty and any state-feedback control which is designed for this representation in the form $u = Kx$ can be translated into a dynamic output feedback controller.

Based on equation (6.3), it is clear that $x(k) \in X \subset \mathbb{R}^{n_x}$, with $n_x = n(q+p)$. Explicitly, $X$ is given by

$$X = \underbrace{Y \times Y \times \dots \times Y}_{\text{s times}} \times \underbrace{U \times U \times \dots \times U}_{\text{s times}} = \{x \in \mathbb{R}^{n_x} : F_x x \le g_x\}$$

*Example 6.2.* Consider the following transfer function

$$P(s) = \frac{k_1 s + 1}{s(s+k_2)} \tag{6.15}$$

where $k_1 = 0.787$, $0.1 \le k_2 \le 3$. Using a sampling time of 0.1 and Euler's first order approximation for the derivative, the following input-output relationship is obtained

$$\begin{aligned} y(k+1) - (2-0.1k_2)y(k) + (1-0.1k_2)y(k-1) = \\ = 0.1k_1 u(k) + (0.01-0.1k_2)u(k-1) + w(k) \end{aligned} \tag{6.16}$$

The signal $w(k)$ represents the process noise with $-0.01 \le w \le 0.01$. The following constraints are considered on the measured variables

$$\begin{cases} -10 \le y(k) \le 10 \\ -5 \le u(k) \le 5 \end{cases}$$

The state $x(k)$ is constructed as follows

$$x(k) = [y(k) \;\; y(k-1) \;\; u(k-1)]^T$$

Hence, the state space model is given by

$$\begin{cases} x(k+1) & = Ax(k) + Bu(k) + Dw(k) \\ y(k) & = Cx(k) \end{cases}$$

where

$$A = \begin{bmatrix} (2-0.1k_2) & -(1-0.1k_2) & (0.01-0.1k_1) \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.1k_1 \\ 0 \\ 1 \end{bmatrix}, D = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Using the polytopic uncertainty description, one obtains

$$A = \alpha A_1 + (1-\alpha)A_2$$

where

$$A_1 = \begin{bmatrix} 1.99 & -0.99 & -0.0687 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 1.7 & -0.7 & -0.0687 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

At each time instant $0 \le \alpha \le 1$ and $-0.01 \le w \le 0.01$ are uniformly distributed pseudo-random numbers. Algorithm 5.1 in Section 5.2 will be employed with a global saturated controller in this example. For this purpose, two controllers have been designed

- The local linear controller $u(k) = Kx(k)$ for the performance. In this example, the peak to peak controller[1] is chosen

$$K = [-22.7252 \quad 10.7369 \quad 0.8729]$$

- The global saturated controller $u(k) = sat(K_s x(k))$ for the domain of attraction

$$K_s = [-4.8069 \quad 4.5625 \quad 0.3365]$$

It is worth noticing that, this controller can be described in the output-feedback form as

$$K(z) = \frac{-22.7894 + 10.7369z^{-1}}{1 - 0.8729z^{-1}}$$

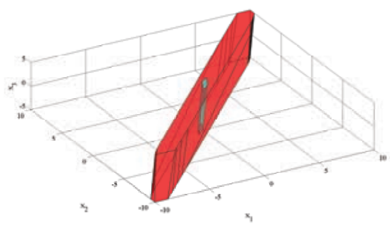
and respectively

$$K_s(z) = \frac{-4.8069 + 4.5625z^{-1}}{1 - 0.3365z^{-1}}$$

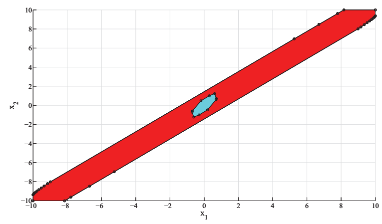
[1] The control law *peak to peak* is developed in the next Section 6.4.

Overall the control scheme is described by a second order plant and two first order controllers, which provide a reduced order solution for the stabilization problem.

Using procedure 2.2 and procedure 2.4 and corresponding to the control laws $u(k) = Kx(k)$ and $u(k) = sat(K_s x(k))$, the maximal robustly invariant sets $\Omega_{max}$ and $\Omega_s$ are computed and depicted in Figure 6.5(a). The blue set is $\Omega_{max}$ and the red set is $\Omega_s$ . Figure 6.5(b) presents the projection of the sets $\Omega_{max}$ and $\Omega_s$ onto the $(x_1, x_2)$ state space.



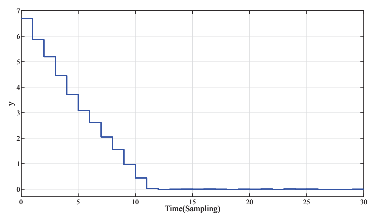
(a)Feasible sets



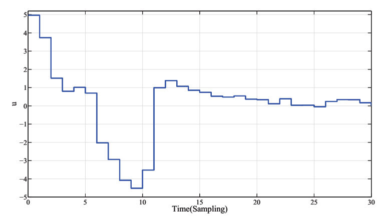
(b) Projection onto $(x_1, x_2)$ space

**Fig. 6.5** Feasible invariant sets for example 6.2.

For the initial condition $x(0) = [6.6970 \quad 7.7760 \quad 5.0000]^T$, Figure 6.6 presents the output and input trajectory of the closed loop system as a function of time.



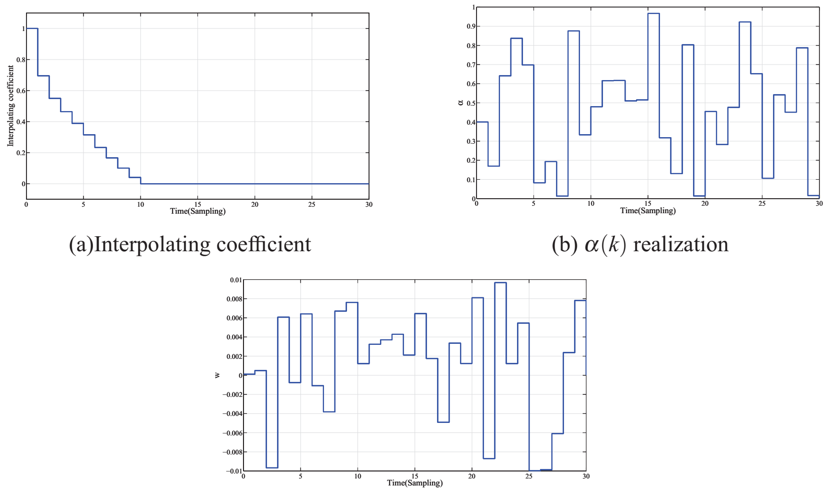
(a)Output trajectory



(b) Input trajectory

**Fig. 6.6** Output and input trajectory of the closed loop system for example 6.2.

Finally, Figure 6.7 shows the interpolating coefficient, the realization of $\alpha(k)$ and $w(k)$ as a function of time.



(a)Interpolating coefficient

(b) $\alpha(k)$ realization

(c) $w(k)$ realization

**Fig. 6.7** Interpolating coefficient and realization of $\alpha(k)$ and $w(k)$ for example 6.2.

## 6.4 Some remark on local controllers

In this section, we will revisit and provide a novel method for the local control design problem. It is clear that the local controller can be any feasible and stabilizing controller. Usually, one would like to ensure a certain level of optimality for the local controller, since when the state of the system reaches the local feasible invariant set, the interpolating controller turns out to be the local controller. Note that the local controller will not encounter constraints. Therefore it can be designed as e.g. an optimal controller, or as a controller satisfying some performance specifications, e.g. a QFT controller.

In the presence of persistent bounded disturbances, a controller with a good disturbance rejection ability might be desirable. The measure of disturbance rejection can be defined as the peak value of the state variable over the peak value of the disturbance. This basic idea will be exploited in the design procedure presented in the remainder of this chapter.

### *6.4.1 Problem formulation*

Consider the problem of regulating to the origin the following discrete-time linear time-varying and uncertain system

$$x(k+1) = A(k)x(k) + B(k)u(k) + D(k)w(k) \tag{6.17}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ are respectively the measurable state variable and the control variable. The matrices $A(k) \in \mathbb{R}^{n \times n}$, $B(k) \in \mathbb{R}^{n \times m}$ and $D(k) \in \mathbb{R}^{n \times d}$ satisfy

$$\begin{cases} A(k) = \sum_{i=1}^{q} \alpha_i(k) A_i,\, B(k) = \sum_{i=1}^{s} \alpha_i(k) B_i,\, D(k) = \sum_{i=1}^{q} \alpha_i(k) D_i, \\ \sum_{i=1}^{q} \alpha_i = 1, \quad \alpha_i \geq 0, \forall i = 1, \ldots, q \end{cases} \tag{6.18}$$

where the matrices $A_i$, $B_i$ and $D_i$ are given.

Both the state and control are subject to the following constraints:

$$\begin{cases} x(k) \in X,\, X = \{x \in \mathbb{R}^n : |Fx| \leq 1\} \\ u(k) \in U,\, U = \{u \in \mathbb{R}^m : |u_i| \leq u_{imax}\} \end{cases} \quad \forall k \geq 0 \tag{6.19}$$

where $u_{imax}$ is the $i-th$ component of the vector $u_{max} \in \mathbb{R}^m$. The matrix $F$ and the vector $u_{max}$ are assumed to be constant with $u_{max} > 0$ such that the origin is contained in the interior of $X$ and $U$.

The signal $w(k) \in \mathbb{R}^d$ represents the additive disturbance input. Using a change of variables

$$D_1(k) = D(k)P^{\frac{1}{2}} \text{ and } w_1(k) = P^{-\frac{1}{2}} w(k)$$

for an appropriate matrix $P$, one can always assume that $w(k)^T w(k) \leq 1$.

### 6.4.2 Robustness analysis

Before going to the synthesis problem, let us consider the analysis problem of the following discrete-time system[2]

$$x(k+1) = H(k)x(k) + D(k)w(k) \tag{6.20}$$

where matrix $H(k) \in \mathbb{R}^{n \times n}$ satisfies

$$H(k) = \sum_{i=1}^{q} \alpha_i(k) H_i$$

where

$$\sum_{i=1}^{q} \alpha_i(k) = 1, \quad \alpha_i(k) \geq 0,$$

and the matrices $H_i$ are extreme realizations of $H(k)$. It is assumed that $w(k)^T w(k) \leq 1$ and $\rho(H(k)) < 1$, where $\rho(H(k))$ is the joint spectral radius of matrix $H(k)$. This

[2] The autonomous version of equation (6.17)

condition implies that when $w(k) = 0$, system (6.20) is robustly asymptotically stable.

Recall that the ellipsoid set $E(P)$ is robust positively invariant for system (6.20), if the condition $x(0) \in E(P)$ implies $x(k) \in E(P)$, $\forall k \geq 1$. In other words, starting from any point in $E(P)$, the state of the system will never leave this set under any admissible uncertainty and disturbance. The following theorem provides a necessary and sufficient condition for invariance of ellipsoid $E(P)$ for the system (6.20).

**Theorem 6.1.** *The ellipsoid $E(P)$ is invariant for system (6.20) if and only if there exists a positive definite matrix $P \in \mathbb{R}^{n\times n}$ satisfying the following LMI conditions*

$$\begin{bmatrix} (1-\tau)P & 0 & PH_i^T \\ 0 & \tau I & D_i^T \\ H_iP & D_i & P \end{bmatrix} \succeq 0$$

*for all $i = 1,2,\ldots,q$ and for some number* $0 < \tau < 1$.

*Proof.* Define the following quadratic function

$$V(x(k)) = x(k)^T P^{-1} x(k)$$

For the invariant property of the set $E(P) = \{x(k) \in \mathbb{R}^n : V(x(k)) \leq 1\}$, it is required that $V(x(k+1)) \leq 1$ for all possible state trajectories and disturbance realizations. That is

$$(Hx + Dw)^T P^{-1} (Hx + Dw) \leq 1$$

for all $x$ and $w$ such that $x^T P^{-1} x \leq 1$ and $w^T w \leq 1$ or

$$\begin{bmatrix} x \\ w \end{bmatrix}^T \begin{bmatrix} H^T P^{-1} H & H^T P^{-1} D \\ D^T P^{-1} H & D^T P^{-1} D \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \leq 1 \tag{6.21}$$

for all $x$ and $w$ such that

$$\begin{bmatrix} x \\ w \end{bmatrix}^T \begin{bmatrix} P^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \leq 1 \tag{6.22}$$

and

$$\begin{bmatrix} x \\ w \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \leq 1 \tag{6.23}$$

By using the $S-$procedure [124], [67] with two quadratic constraints, the conditions (6.21), (6.22), (6.23) can be equivalently rewritten as

$$\begin{bmatrix} H^T P^{-1} H & H^T P^{-1} D \\ D^T P^{-1} H & D^T P^{-1} D \end{bmatrix} \preceq \begin{bmatrix} \tau_1 P^{-1} & 0 \\ 0 & \tau_2 I \end{bmatrix} \tag{6.24}$$

for some values of $\tau_1 \geq 0$, $\tau_2 \geq 0$, such that $\tau_1 + \tau_2 \leq 1$.

As a consequence of the fact that $P^{-1} \succ 0$, it follows that $H^T P^{-1} H \succ 0$ and $D^T P^{-1} D \succ 0$. Hence $\tau_1$ and $\tau_2$ must be strictly positive. It is clear that if the inequality (6.24) holds for some $\tau_1^0 < 1-\tau_2$, then it also holds for $\tau_1 = 1-\tau_2$, since for all $\tau_1^0 < 1-\tau_2$

$$\begin{bmatrix} \tau_1^0 P^{-1} & 0 \\ 0 & \tau_2 I \end{bmatrix} \preceq \begin{bmatrix} (1-\tau_2)P^{-1} & 0 \\ 0 & \tau_2 I \end{bmatrix}$$

Hence, it is nonrestrictive to use $\tau_1 = 1-\tau_2$. Condition (6.24) is thus equivalent to the LMI

$$\begin{bmatrix} (1-\tau)P^{-1} & 0 \\ 0 & \tau I \end{bmatrix} - \begin{bmatrix} H^T \\ D^T \end{bmatrix} P^{-1} \begin{bmatrix} H & D \end{bmatrix} \succeq 0$$

where $\tau = \tau_2$, $0 < \tau < 1$. By using the Schur complement one has

$$\begin{bmatrix} (1-\tau)P^{-1} & 0 & H^T \\ 0 & \tau I & D^T \\ H & D & P \end{bmatrix} \succeq 0$$

or

$$\begin{bmatrix} \tau I & D^T \\ D & P \end{bmatrix} - \frac{1}{1-\tau} \begin{bmatrix} 0 \\ H \end{bmatrix} P \begin{bmatrix} 0 & H^T \end{bmatrix} \succeq 0$$

or, equivalently

$$\begin{bmatrix} (1-\tau)P & 0 & PH^T \\ 0 & \tau I & D^T \\ HP & D & P \end{bmatrix} \succeq 0 \tag{6.25}$$

Clearly, the left hand side of condition (6.25) reaches the minimum on one of the vertices of $H(k)$, $D(k)$, so the set of LMI conditions to be satisfied is the following

$$\begin{bmatrix} (1-\tau)P & 0 & PH_i^T \\ 0 & \tau I & D_i^T \\ H_iP & D_i & P \end{bmatrix} \succeq 0 \tag{6.26}$$

for all $i = 1, 2, \ldots, q$ and for some number $0 < \tau < 1$. □

Theorem 6.1 states that for all admissible uncertainties and disturbances, the set $E(P) = \{x : x^T P^{-1} x \leq 1\}$ is invariant, where $P$ is a solution of (6.26).

*Remark 6.1.* It has to be mentioned that the LMI conditions for ellipsoidal sets to be minimal invariant for continuous-time linear time-invariant systems have been presented in [1] and for discrete time linear time invariant systems in [104], [74]. By the previous theorem we extended the results in [1], [104], [74] to discrete-time linear time-varying and uncertain systems. In addition, the LMI conditions (6.26) are applicable for different types of invariant ellipsoids, e.g. minimal invariant ellipsoids, maximal invariant ellipsoids, etc.

*Remark 6.2.* It is clear that ellipsoid $E(P)$, resulting from problem (6.26) might not be contractive although being invariant. In order to ensure such additional properties it is required that for all $x(k) \in E(P) = \{x(k) : V(x(k)) \leq 1\}$ to have

$$V(x(k+1)) - V(x(k)) < 0 \tag{6.27}$$

or in other words, the Lyapunov function $V(x(k))$ is strictly decreasing. By using the same argument as the proof of theorem 6.1, condition (6.27) can be transformed into

$$\begin{bmatrix} (1-\tau)P & 0 & PH_i^T \\ 0 & \tau I & D_i^T \\ H_iP & D_i & P \end{bmatrix} \succ 0 \tag{6.28}$$

for all $i = 1,2,\ldots,q$ and for some number $0 < \tau < 1$.

### 6.4.3 Robust optimal design

The peak value of the state variable over the peak value of the disturbance is defined as follows.

$$J = \frac{\|x\|_\infty}{\|d\|_\infty} \tag{6.29}$$

The existence of an invariant ellipsoid $E(P_p)$ can be used as an upper bound of the peak to peak value in (6.29). Based on Theorem 6.1, a linear feedback controller $u = K_p x$, which minimizes the size of the invariant ellipsoid $E(P_p)$, can be designed for system (6.17) with constraints (6.19) by solving the following optimization problem

$$\min_{P_p,Y_p} \{trace(P_p(\tau))\} \tag{6.30}$$

subject to

- Invariance condition

$$\begin{bmatrix} (1-\tau)P_p & 0 & P_pA_i^T + Y_p^TB_i^T \\ 0 & \tau I & D_i^T \\ A_iP_p + B_iY_p & D_i & P_p \end{bmatrix} \succeq 0 \tag{6.31}$$

- Constraint satisfaction [3]
  +) On state:

$$\begin{bmatrix} 1 & f_iP_p \\ P_pf_i^T & P_p \end{bmatrix} \succeq 0 \tag{6.32}$$

  where $f_i$ is the $i-th$ row of the matrix $F$.
  +) On input:

$$\begin{bmatrix} u_{imax}^2 & K_{ip}P_p \\ P_pK_{ip}^T & P_p \end{bmatrix} \succeq 0, \quad \forall i = 1,2,\ldots,m$$

  or

$$\begin{bmatrix} u_{imax}^2 & Y_{ip} \\ Y_{ip}^T & P_p \end{bmatrix} \succeq 0, \quad \forall i = 1,2,\ldots,m \tag{6.33}$$

[3] see Section 2.3.3

with $Y_p = K_p P_p \in \mathbb{R}^{m\times n}$, $K_{ip}$ is the $i-th$ row of the matrix $K$ and $Y_{ip} = K_{ip}P_p$ is the $i-th$ row of the matrix $Y$. The trace of a square matrix is defined to be the sum of the elements on the main diagonal of the matrix. Minimization of the trace of matrices corresponds to the search for the minimal sum of eigenvalues of matrices.

It is important to note that when $\tau$ is fixed, the optimization problem (6.30), (6.31), (6.32) is an LMI problem, for which nowadays, there exist several effective solvers, e.g. [94], [49].

*Remark 6.3.* At the same time, optimizing the peak to peak feedback gain $K_p$ can lead to a maximal invariant ellipsoid $E(P^m) \subset X$, such that for all $x(k) \in E(P^m)$, it follows that $x(k+1) \in E(P^m)$ and $u(k) = K_p x(k) \in U$. This can be done by solving the following LMI problem

$$\max_{P_p}\{trace(P^m(\tau))\} \tag{6.34}$$

subject to the constraints (6.32) and (6.33).

*Remark 6.4.* Recall that the ellipsoid $E(P_p)$ is the limit set of all trajectories of the system (6.17) with the feedback gain $u = K_p x$, i.e. all trajectories starting from the origin, are bounded by $E(P_p)$ and all trajectories, starting outside $E(P_p)$ converge to $E(P_p)$. On the other hand, the set $E(P^m)$ is the admissible ellipsoid, which maximizes the cost function (6.34) for system (6.17) with the feedback gain $u = K_p x$.

*Remark 6.5.* Aside performance (here in the sense of disturbance rejection), another desideratum in the control design is the approximation by invariant ellipsoids of the maximal domain of attraction. It is well known that by using the LMI technique, one can determine the largest invariant ellipsoid $E(P)$ with respect to the inclusion of some reference direction defined by $x_0$, meaning that the set $E(P)$ will include the point $\theta x_0$, where $\theta$ is a scaling factor on the direction pointed by the vector $x_0$. Indeed, $\theta x_0 \in E(P)$ implies that $\theta^2 x_0^T P^{-1} x_0 \le 1$ or by using the Schur complements

$$\begin{bmatrix} 1 & \theta x_0^T \\ \theta x_0 & P \end{bmatrix} \succeq 0 \tag{6.35}$$

Therefore the following LMI optimization problem can be used to obtain an invariant ellipsoid $E(P_i)$, that contains the most important extension on a certain direction defined by the reference point $x_i$

$$\max_{P_i, Y_i, \theta} \theta(\tau) \tag{6.36}$$

subject to constraints (6.28), (6.32), (6.33), (6.35).

*Example 6.3.* Consider the following discrete-time linear time varying system

$$x(k+1) = A(k)x(k) + B(k)u(k) + Dw(k) \tag{6.37}$$

where

$$A(k) = \alpha(k)A_1 + (1-\alpha(k))A_2, \quad B(k) = \alpha(k)B_1 + (1-\alpha(k))B_2$$

with

$$A_1 = \begin{bmatrix} 2.0200 & 1.0200 & -0.9800 \\ -0.9600 & 2.0400 & 2.0400 \\ 1.0600 & 3.0600 & 1.0600 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.9800 & 0.9800 & -1.0200 \\ -1.0400 & 1.9600 & 1.9600 \\ 0.9400 & 2.9400 & 0.9400 \end{bmatrix}$$
$$B_1 = \begin{bmatrix} 1.0100 & 0.0100 \\ 0.0200 & 1.0200 \\ 1.0300 & 1.0300 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0.9900 & -0.0100 \\ -0.0200 & 0.9800 \\ 0.9700 & 0.9700 \end{bmatrix}, \quad D = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

The constraint on the disturbance is $w(k)^T w(k) \leq 1$. For simplicity, we do not consider any constraints on state or on input. By solving the LMI problem (6.30) a robust peak to peak controller $u(k) = K_p x(k)$ is obtained with

$$K_p = \begin{bmatrix} -1.9751 & -1.0497 & 1.0249 \\ 0.9953 & -1.9906 & -2.0047 \end{bmatrix}$$

together with the invariant ellipsoid $E(P_p)$ where

$$P_p = \begin{bmatrix} 1.0206 & 1.0311 & 1.0417 \\ 1.0311 & 1.0522 & 1.0732 \\ 1.0417 & 1.0732 & 1.1048 \end{bmatrix}$$

Figure 6.8 shows the projection of ellipsoid $E(P_p)$ on the $(x_1, x_2)$ state space. This figure also shows the state trajectory $(x_1, x_2)$ of the closed loop system as a function of time corresponding to a certain initial point inside $E(P_p)$. From Figure 6.8, it can be observed that the state trajectory travels closed to the boundary of the projection of the set $E(P_p)$.

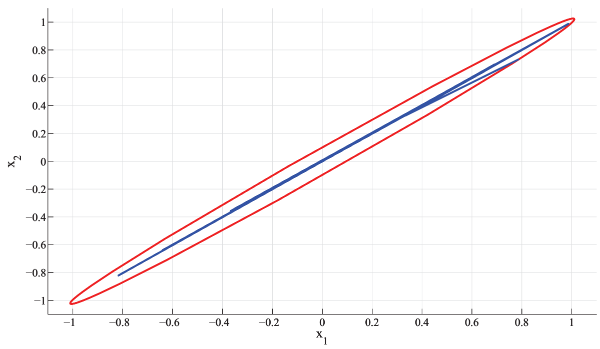



**Fig. 6.8** Optimal robustly invariant ellipsoid and state trajectory for example 6.3.

# Part III
# Applications

# Chapter 7
# Ball and plate system

The main purpose of this chapter is to apply the algorithms discussed in the previous chapters for the constrained control of the ball and plate experiment with an actuation on the angles of the plate. The presence of constraints on the positions of the ball and the angle of the plate makes the experiment an adequate benchmark test for the proposed theories and have been used in previous constrained control experiments, see for example [28].

## 7.1 System description

The ball and plate benchmark is depicted in Figure 7.1. The system consists of a mechanical plate, two actuation mechanisms for tilting the plate around two orthogonal axes and a ball position sensor. The entire system is mounted on a mechanical (steel) base plate and is supported by four vertical springs and a central joint. The motors are operated in *angular position mode* for the simplicity of the modeling and control. A pulse-width modulated signal is employed for this purpose. The servos are powered by a 6V DC power supply.

A resistive touch sensitive glass screen that is actually meant to be a computer touchscreen was used for sensing the ball position. It provides an extremely reliable, accurate, and economical solution to the ball position sensing problem. The screen consists of three layers: a glass sheet, a conductive coating on the glass sheet, and a hard-coated conductive top-sheet.

## 7.2 System identification

For the identification purpose, the following notations will be used

- $x^r(m)$ – position of the ball along the $x-$axis,
- $y^r(m)$ – position of the ball along the $y-$axis,

**Fig. 7.1** Ball and plate system

- $x_{max} = 0.1056m$ – maximum position of the ball on the $x-$axis,
- $y_{max} = 0.0792m$ – maximum position of the ball on the $y-$axis,
- $u_x^r(rad)$ – angle of the plate around the $x-$axis,
- $u_y^r(rad)$ – angle of the plate around the $y-$axis,
- $u_{xmax} = \pm\frac{\pi}{6}rad$ – maximum angle of the plate w.r.t. the $x-$axis,
- $u_{ymax} = \pm\frac{\pi}{6}rad$ – maximum angle of the plate w.r.t. the $y-$axis,
- $g = 9.8m/s^2$ – acceleration due to gravity,
- $x = \frac{x^r}{x_{max}}$ – scaled position of the ball along the $x-$axis, $-1 \le x \le 1$,
- $y = \frac{y^r}{y_{max}}$ – scaled position of the ball along the $y-$axis, $-1 \le y \le 1$,
- $u_x = \frac{u_x^r}{u_{xmax}}$ – scaled angle of the plate around the $x-$axis, $-1 \le u_x \le 1$,
- $u_y = \frac{u_y^r}{u_{ymax}}$ – scaled angle of the plate around the $y-$axis, $-1 \le u_y \le 1$.

### *7.2.1 The identification procedure*

The dynamical model of the ball and plate system can be derived by Newton's second law. It is well known [46] that under the assumptions of negligible friction between the ball and the plate and the motor friction and for small $u_x$ and $u_y$, the dynamics of the ball and plate system can be modeled as

$$\begin{cases} \ddot{x} = \frac{5x_{xmax}}{7u_{xmax}} g\sin(u_x) \approx \frac{5x_{xmax}}{7u_{xmax}} gu_x \\ \ddot{y} = \frac{5x_{ymax}}{7u_{ymax}} g\sin(u_y) \approx \frac{5y_{ymax}}{7u_{ymax}} gu_y \end{cases} \tag{7.1}$$

However it was experienced experimentally that model (7.1) is not accurate enough for capturing the dynamics of the ball and plate system. The experimental time response is *four times* slower than what is given by the simulation model (7.1).

It is assumed that the ball and plate system is a collection of decoupled dynamics operating simultaneously, one is along the $x-$axis and the other one is along the $y-$axis. Hence, similar but independent systems can be used for modeling the ball motion in each coordinate. The problem of identifying only the system along the $x-$axis, based on data from experiments, is discussed here. The identification problem of the system along the $y-$axis can be developed along the same lines.

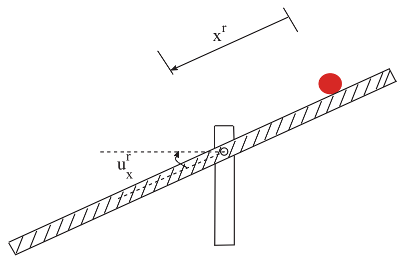



**Fig. 7.2** Ball and plate system along the $x-$axis.

With a slight abuse of notation, $x$ is also denoted as a scaled position of the ball. The scaled difference equation of the ball and plate along the $x-$axis is described as follows

$$\begin{aligned} &x(k+1)+a_1x(k)+\ldots+a_nx(k-n+1)= \\ &=b_1u_x(k)+b_2u_x(k-1)+\ldots+b_mu_x(k-m+1) \end{aligned} \tag{7.2}$$

where $x(k)$ is the position of the ball at time instant $k$, $u_x(k)$ is the angle of the plate at time instant $k$, $n$ is the unknown number of poles for the model dynamic, $m-1$ is the unknown number of zeros, $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_m$ are the unknown coefficients of the model.

For simplicity, it is assumed that $m=n$. With this assumption, equation (7.2) can be rewritten as

$$\begin{aligned} x(k+1)=&-a_1x(k)-\ldots-a_nx(k-n+1)+ \\ &+b_1u_x(k)+b_2u_x(k-1)+\ldots+b_nu_x(k-n+1) \end{aligned}$$

or in a compact matrix form

$$x(k+1)=\Phi(k)\theta \tag{7.3}$$

where

$$\Phi(k) = [-x(k) \ldots -x(k-n+1)\; u_x(k) \ldots u_x(k-n+1)]$$
$$\theta = [a_1 \quad a_2 \quad \ldots \quad a_n \quad b_1 \quad b_2 \quad \ldots \quad b_n]^T$$

Using equation (7.3), one has

$$\begin{cases} x(k+2) = \Phi(k+1)\theta \\ x(k+3) = \Phi(k+2)\theta \\ \vdots \\ x(k+N+1) = \Phi(k+N)\theta \end{cases}$$

or in matrix form, the regression can be expressed as

$$X = \Phi\theta \tag{7.4}$$

where

$$X = \left[x(k+1)\; x(k+2) \;\ldots\; x(k+N+1)\right]^T,$$
$$\Phi = \begin{bmatrix} x(k) & \ldots & x(k-n+1) & u_x(k) & \ldots & u_x(k-n+1) \\ x(k+1) & \ldots & x(k-n+2) & u_x(k+1) & \ldots & u_x(k-n+2) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x(k+N) & \ldots & x(k+N-n+1) & u_x(k+N) & \ldots & u_x(k+N-n+1) \end{bmatrix}$$

and $N+n+1$ is the length of the collected data. In general, the number of equations $N$ in (7.4) is much bigger than the number of unknown parameters $a_i$, $b_i$, $i = 1,2,\ldots,n$. Hence, the parameters $a_i$, $b_i$, $i = 1,2,\ldots,n$, as well as the order of system $n$ can be obtained using the least square method minimizing the following loss function

$$\min_{\theta,n} E(\theta,n) \tag{7.5}$$

where

$$E(\theta,n) = (X - \Phi\theta)^T (X - \Phi\theta) \tag{7.6}$$

When $n$ is fixed, it is well known that the optimization problem (7.5) admits a unique solution [92]

$$\theta = (\Phi^T\Phi)^{-1}\Phi^T X \tag{7.7}$$

Once the matrices $\Phi$ and $X$ are defined and the order of the system $n$ is fixed, the parameters of the system $\theta$ is easily found by using equation (7.7).

### *7.2.2 Identification of the ball and plate system*

For the ball and plate system, the identification experiment was carried out by applying a random binary input signal, independently on each axes. This input signal was generated by using Microchip dsPIC33F micro-controller hosted on FLEX boards. The touchscreen was used for sensing the ball position. For recording the scaled position of the ball $x$ and the scaled angle of the plate $u_x$, the program packet Scicos/Scilab was used.

It is clear that if the full control range $u_x$ is exploited, the ball will quickly hit the border of the plate. On the other hand if $u_x$ is small, the dynamics of the ball and plate become highly nonlinear due to friction. Here we chose the scaled input level in between $-0.45$ and $0.45$. For the sampling period $T_s = 0.1sec$, Figure 7.3 presents the stored input $u_x$ and measured output $x$. The data has been obtained during $86sec$.

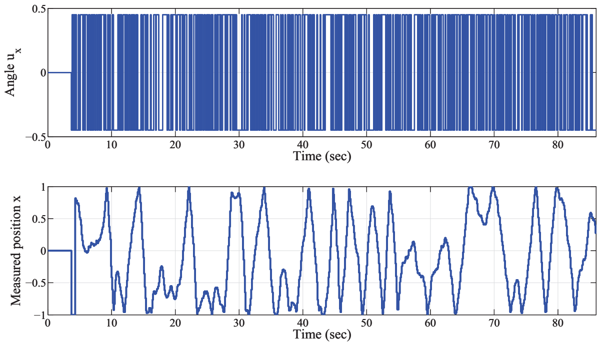



**Fig. 7.3** Actual angle of the plate and actual position of the ball.

Figure 7.4 shows the scaled actual angle of the plate and scaled actual position of the ball, selected from the data in Figure 7.3, when the ball did not hit the border of the plate.

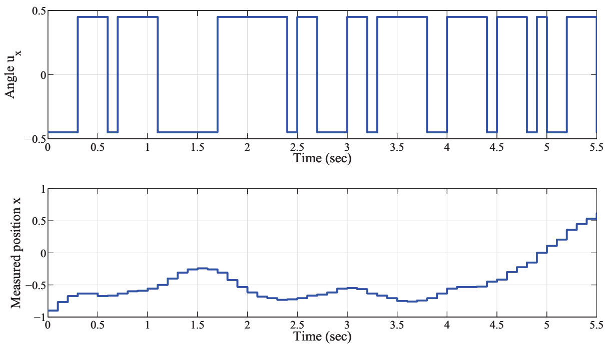



**Fig. 7.4** Actual angle of the plate and actual position of the ball when the ball did not hit the border of the plate.

Based on the data in Figure 7.4 and depending on $n$, the coefficients $a_i$, $b_i$, $i = 1,2,\ldots,n$ are obtained. For choosing the reasonable order for the system, the loss function $E(\theta,n)$ is plotted in Figure 7.5 as a function of order.

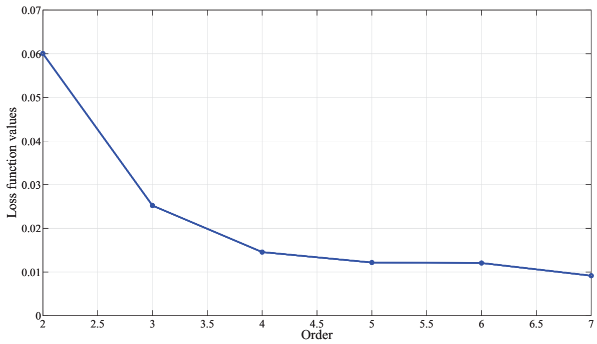



**Fig. 7.5** Loss function value as a function of order

From Figure 7.5, it can be observed that the loss function presents a steeper decrease from order 2 to order 3 than from order 3 to order 4. Another observation is that from the order 4, the loss function value decreases very slowly and is almost constant. It follows that the order of the system can be taken as either 3 or 4 with an appropriate degree of precision.

The Nyquist plot of the ball and plate can also be used to choose the right order of the model and to find the unknown coefficients $a_i$, $b_i$, $i = 1,2,\ldots,n$. However due to frictions and the small size of the plate and by consequence to the constraint activation, the frequency experiments were very difficult to realize. We therefore did not integrate any frequency experiment in the present identification process.

For validation of the ball and plate system model, we tested the third and the fourth order models. The results are satisfactory for both models without visible dynamical difference. Here we will present the numerical result for the third order model, as this will be chosen for the design purposes in the second part of the paper.

Based on equation (7.7) and data shown in Figure 7.4, the dynamic of the ball and plate system along the $x-$axis is obtained

$$\begin{aligned} &x(k+1) - 1.6736x(k) + 0.3632x(k-1) + 0.2959x(k-2) = \\ &= 0.0157u_x(k-1) + 0.0701u_x(k-2) \end{aligned} \tag{7.8}$$

It is worth noticing that the ball and plate model (7.8) is an input-delay system with one step time delay. According to Figure 7.6, the predicted positions using model (7.8) are close to the actual position of the plate.

For validating model (7.8), another experiment was carried out. The random binary sequence is applied as an input signal and the open-loop trajectories of the experiment and of the simulation model compared for a validity check.

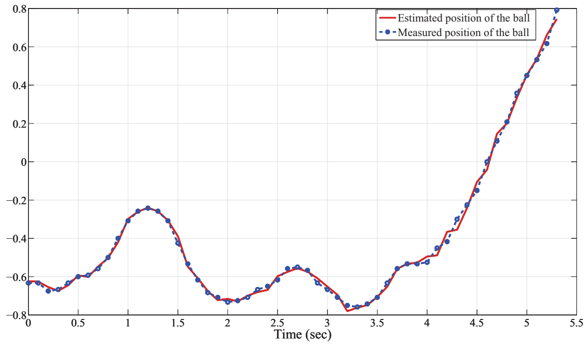



**Fig. 7.6** Comparison between the actual position of the plate and predicted position using model (7.8)

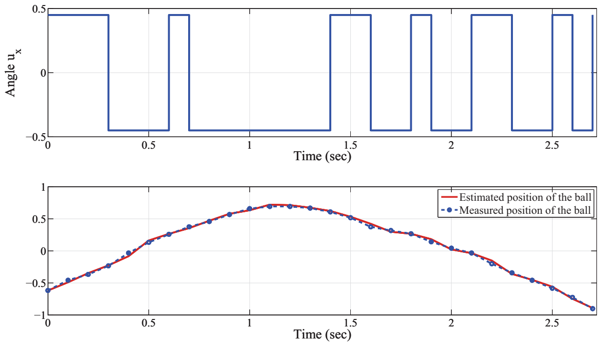



**Fig. 7.7** Validation of model (7.8).

Figure 7.7 supports the claim that that model (7.8) is accurate enough for capturing the dynamics of the ball and plate system.

Analogously, the scaled difference equation of the ball and plate along the $y-$axis is described as follows

$$\begin{aligned} y(k+1) - 1.6736y(k) + 0.3632y(k-1) + 0.2959y(k-2) = \\ = 0.0209u_y(k-1) + 0.0935u_y(k-2) \end{aligned} \tag{7.9}$$

## 7.3 Controller design

In this section, we concentrate only on the controller design for the ball and plate system along the $x-$axis. A controller along the $y-$axis can be easily obtained using similar arguments.

### 7.3.1 State space realization

Equation (7.8) can be rewritten as

$$x(k+1) = 1.6736x(k) - 0.3632x(k-1) - 0.2959x(k-2) + \\ +0.0157u_x(k-1) + 0.0701u_x(k-2)$$

The state variables are defined as

$$\begin{cases} x_1(k) = x(k) \\ x_2(k) = x(k-1) \\ x_3(k) = x(k-2) \\ x_4(k) = u(k-1) \\ x_5(k) = u(k-2) \end{cases}$$

With these state variables, the state space model is

$$x_n(k+1) = Ax_n(k) + Bu_x(k) \tag{7.10}$$

where

$$x_n(k) = \begin{bmatrix} x_1(k) \; x_2(k) \; x_3(k) \; x_4(k) \; x_5(k) \end{bmatrix}^T$$
$$A = \begin{bmatrix} 1.6736 & -0.3632 & -0.2959 & 0.0157 & 0.0701 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$
$$B = \begin{bmatrix} 0 \; 0 \; 0 \; 1 \; 0 \end{bmatrix}^T$$

The state space model (7.10) is non-minimal. One can easily find a third order minimal state space realization for system (7.8). However using model (7.10) all the state variables are available through output and input measurement and storage of appropriate previous measurements. Referring to the discussion in Section 6.3, this is an important aspect in the constrained control design, as the construction of an estimation for a state constrained dynamics is avoided.

The constraints on the state variables and on the input variables are

$$\begin{array}{lll} -1 \le x_1(k) \le 1, & -1 \le x_2(k) \le 1, & -1 \le x_3(k) \le 1 \\ -1 \le x_4(k) \le 1, & -1 \le x_5(k) \le 1, & -1 \le u_x(k) \le 1 \end{array} \tag{7.11}$$

## 7.3.2 Interpolation based control

The explicit interpolation based control described in Section 4.3 will be used for the ball and plate system. With this aim, we firstly choose the local feedback gain $u_x(k) = Kx_n(k)$ as

$$K = [-9.0103 \; 4.5692 \; 2.2384 \; -0.5503 \; -0.5303] \tag{7.12}$$

It is worth noticing that this controller is equivalent to the output controller

$$K(z) = \frac{-9.0103 + 4.5692z^{-1} + 2.2384z^{-2}}{1 + 0.5503z^{-1} + 0.5303z^{-2}} \tag{7.13}$$

The plant is third order and the local controller is second order. A reduced order controller is obtained, which proves to be an advantage from the point of view of real-time implementation (sampling time of $0.1sec$).

For the controller $u_x(k) = Kx_n(k)$ and by using procedure 2.2, the maximal invariant set $\Omega_{max}$ is obtained for the system (7.10) with constraints (7.11). Based on the set $\Omega_{max}$, the controlled invariant set $C_N$ with $N = 8$ is computed. Note that $C_8 = C_9$. In this case $C_8$ is a maximal controlled invariant set. The sets $\Omega_{max}$ and $C_N$ are illustrated in Figure 7.8, cut through $x_4 = 0$ and $x_5 = 0$.

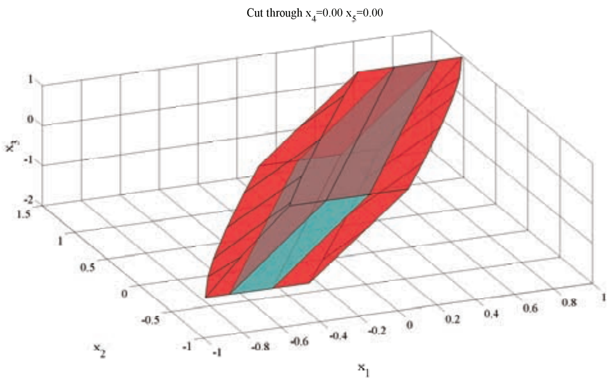



**Fig. 7.8** Feasible sets for the ball and plate system. $\Omega_{max}$ is the blue set, when applying the control law $u_x = Kx_n$. The controlled invariant set $C_N$ is the red set.

Figure 7.9 shows the feasible sets obtained by projecting the feasible sets $C_N$ and $\Omega_{max}$ onto the $x_1 - x_2$ state space.

The number of vertices of the set $C_N$ is 148 and these are not reported here. The control values at vertices are found by maximizing the control action at the vertices of the feasible invariant set $C_N$, while keeping the state inside the set.

Since we do not dispose of an embedded LP solver within the existing platform, the implicit version of the interpolation based control can not be deployed for the

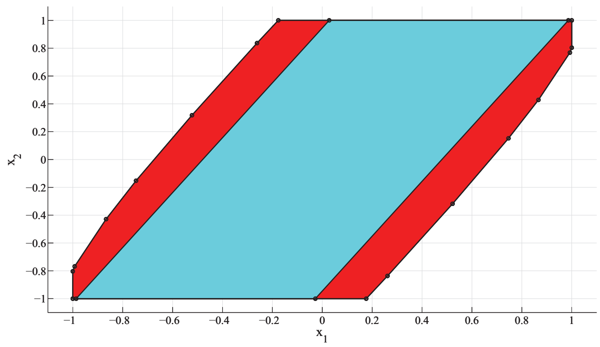



**Fig. 7.9** Feasible sets for the ball and plate system in the $x_1 - x_2$ state space.

ball and plate system and an explicit form needs to be constructed. For the explicit version, after merging the regions with the identical control it was found that the explicit interpolation based controller computed here coincides with a saturated controller

$$u_x(k) = \begin{cases} -1, \text{ if } & u_1(k) \leq -1 \\ u_1(k), \text{ if } & -1 \leq u_1(k) \leq 1 \\ 1, \text{ if } & u_1(k) \geq 1 \end{cases} \tag{7.14}$$

where

$$u_1(k) = -9.01x(k) + 4.57x(k-1) + 2.24x(k-2) - 0.55u_x(k-1) - 0.53u_x(k-2)$$

Analogously, along the $y-$coordinate, the explicit interpolation based controller is described by

$$u_y(k) = \begin{cases} -1, \text{ if } & u_2(k) \leq -1 \\ u_2(k), \text{ if } & -1 \leq u_2(k) \leq 1 \\ 1, \text{ if } & u_2(k) \geq 1 \end{cases} \tag{7.15}$$

where

$$u_2(k) = -6.54y(k) + 3.33y(k-1) + 1.63y(k-2) - 0.54u_y(k-1) - 0.52u_y(k-2)$$

## 7.4 Experimental results

The result of the implementation of the interpolation based control law is reported in Figure 7.10, Figure 7.11 and Figure 7.12.

Finally we point the reader to the video recording of the experiment publicly available at: `http://www.youtube.com/watch?v=P91cStkCKu4`.

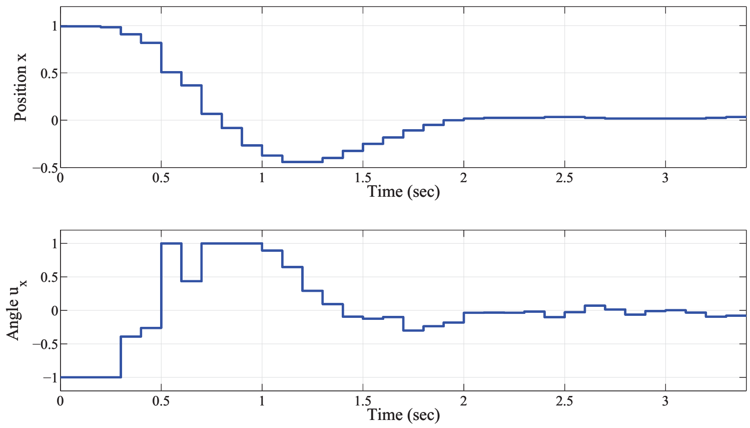



**Fig. 7.10** Interpolation based control for the ball and plate system along the $x-$coordinate. Experimental result.

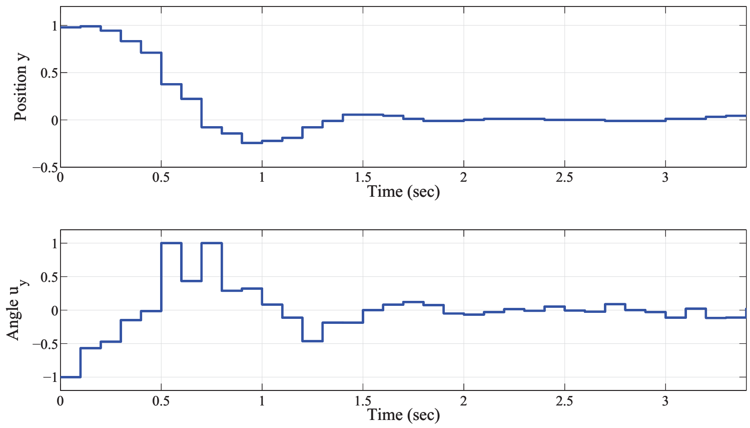



**Fig. 7.11** Interpolation based control for the ball and plate system along the $y-$coordinate. Experimental result.

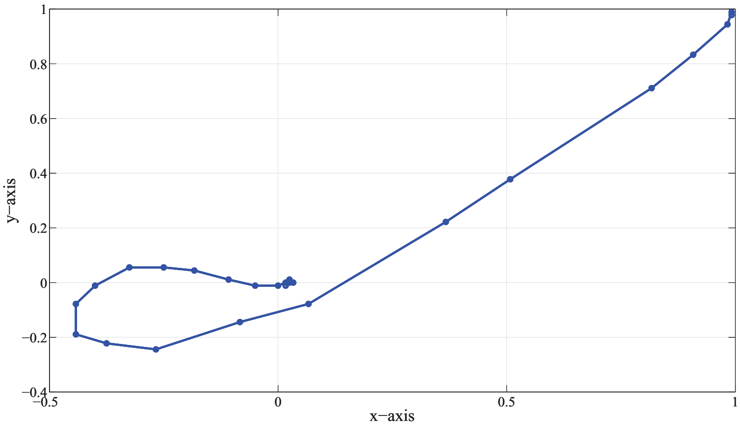



**Fig. 7.12** Interpolation based control for the ball and plate system in the space coordinates $x - y$. Experimental result.

# Chapter 8
# Non-isothermal continuous stirred tank reactor

This chapter deals with the control of a non-isothermal continuous stirred tank reactor (CSTR). The aim of this chapter is not to repeat the theoretical results for the controller design discussed in the previous chapters, but is to show how the interpolation based control approach works for uncertain and multi-input multi-ouput systems.

## 8.1 Continuous stirred tank reactor model

The case of a single non-isothermal continuous stirred tank reactor [136], [72], [98] is studied in this chapter. The reactor is the one presented in various works by Perez et al. [119] and [120] in which the exothermic reaction $\mathscr{A} \to \mathscr{B}$ is assumed to take place. The heat of reaction is removed via the cooling jacket that surrounds the reactor. The jacket cooling water is assumed to be perfectly mixed and the mass of the metal walls is considered negligible, so that the thermal inertia of the metal is not considered. The reactor is also assumed to be perfectly mixed and heat losses are regarded as negligible.

The continuous linearized reactor model is the following [98]

$$\dot{x} = A_c x + B_c u \tag{8.1}$$

where $x = [x_1 \quad x_2]^T$ with $x_1$ is the reactor concentration and $x_2$ is the reactor temperature, $u = [u_1 \quad u_2]^T$ with $u_1$ is the feed concentration and $u_2$ is the coolant flow. Depending on the operating points, matrices $A_c$ and $B_c$ are defined as follows

$$\begin{aligned} A_c &= \begin{bmatrix} -\frac{F}{V} - k_0(t)e^{-\frac{E}{RT_s}} & -\frac{E}{RT_s^2}k_0(t)e^{-\frac{E}{RT_s}}C_{As} \\ -\frac{\Delta H_{rxn}(t)k_0(t)e^{-\frac{E}{RT_s}}}{\rho C_p} & -\frac{F}{V} - \frac{UA}{V\rho C_p} - \Delta H_{rxn}(t)\frac{E}{\rho C_p R T_s^2}k_0(t)e^{-\frac{E}{RT_s}}C_{As} \end{bmatrix}, \\ B_c &= \begin{bmatrix} \frac{F}{V} & 0 \\ 0 & -2.098 \times 10^5 \frac{T_s - 365}{V\rho C_p} \end{bmatrix} \end{aligned} \tag{8.2}$$

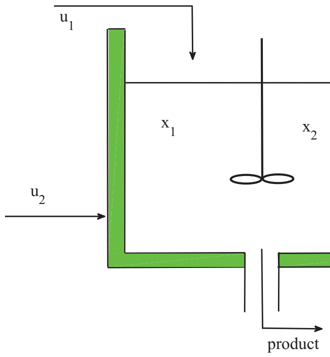



**Fig. 8.1** Continuous stirred tank reactor.

The operating parameters are shown in table 8.1

**Table 8.1** The operating parameters of non-isothermal CSTR

| Parameter | Value | Unit |
|---|---|---|
| $F$ | 1 | $m^3/min$ |
| $V$ | 1 | $m^3$ |
| $\rho$ | $10^6$ | $g/m^3$ |
| $C_p$ | 1 | $cal/g.K$ |
| $\Delta H_{rxn}$ | $10^7 - 10^8$ | $cal/kmol$ |
| $E/R$ | 8330.1 | $K$ |
| $k_o$ | $10^9 - 10^10$ | $min^{-1}$ |
| $UA$ | $5.34 \times 10^6$ | $cal/K.min$ |

The linearized model at steady state $x_1 = 0.265 kmol/m^3$ and $x_2 = 394K$ and under the uncertain parameters $k_0$ and $-\Delta H_{rxn}$ will be considered. The system (8.1) is discretized with a sampling time of 0.15min in order to obtain the following uncertain system [155]

$$\begin{cases} x(k+1) & = A(k)x(k) + Bu(k) \\ y(k) & = Cx(k) \end{cases} \tag{8.3}$$

where

$$A(k) = \begin{bmatrix} 0.85 - 0.0986\beta_1(k) & -0.0014\beta_1(k) \\ 0.9864\beta_1(k)\beta_2(k) & 0.0487 + 0.01403\beta_1(k)\beta_2(k) \end{bmatrix},$$
$$B = \begin{bmatrix} 0.15 & 0 \\ 0 & -0.912 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and the parameter variation bounded by

$$\begin{cases} 1 \le \beta_1(k) = \frac{k_0}{10^9} \le 10 \\ 1 \le \beta_2(k) = -\frac{\Delta H_{rxn}}{10^7} \le 10 \end{cases}$$

Matrix $A(k)$ can be written as

$$A(k) = \alpha_1(k)A_1 + \alpha_2(k)A_2 + \alpha_3(k)A_3 + \alpha_4(k)A_4$$

where

$$A_1 = \begin{bmatrix} 0.751 & -0.0014 \\ 0.986 & 0.063 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.751 & -0.0014 \\ 9.864 & 0.189 \end{bmatrix}$$
$$A_3 = \begin{bmatrix} -0.136 & -0.014 \\ 9.864 & 0.189 \end{bmatrix}, \quad A_4 = \begin{bmatrix} -0.136 & -0.014 \\ 98.644 & 1.451 \end{bmatrix}$$

and $\sum_{i=1}^{4} \alpha_i(k) = 1$, $\alpha_i(k) \ge 0$.

The constraints on input and state are

$$\begin{cases} -0.5 \le x_1 \le 0.5, \quad -20 \le x_2 \le 20 \\ -0.5 \le u_1 \le 0.5, \quad -1 \le u_2 \le 1 \end{cases} \tag{8.4}$$

## 8.2 Controller design

Algorithm 5.1 in Section 5.2 will be employed with the global vertex controller in this example. The local feedback controller $u(k) = Kx(k)$ is chosen as

$$K = \begin{bmatrix} -2.7187 & 0.0259 \\ 17.2865 & 0.1166 \end{bmatrix}$$

For this controller and based on procedure 2.2, the robust maximal invariant set $\Omega_{max}$ is computed. Based on the set $\Omega_{max}$ and by using procedure 2.3, the $N-$step controlled invariant set $C_N$ with $N = 7$ is defined. Note that $C_7 = C_8$. In this case $C_7$ is a maximal controlled invariant set for system (8.3) with constraints (8.4). The sets $\Omega_{max}$ and $C_N$ are depicted in Figure 8.2.

The set $\Omega_{max}$ is presented in minimal normalized half-space representation as

$$\Omega_{max} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -1.0000 & 0.0095 \\ 1.0000 & 0.0067 \\ 1.0000 & -0.0095 \\ -1.0000 & -0.0067 \end{bmatrix} x \le \begin{bmatrix} 0.1839 \\ 0.0578 \\ 0.1839 \\ 0.0578 \end{bmatrix} \right\}$$

The set of vertices of $C_N$ together with the vertex control matrix $U_v$ are

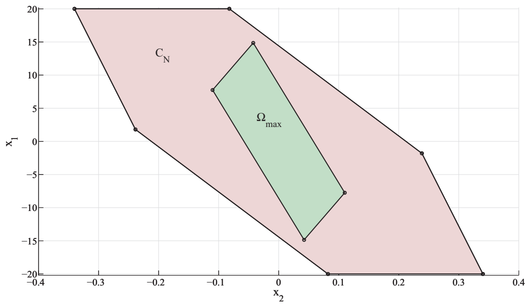



**Fig. 8.2** Feasible invariant sets for the interpolation based control method.

$$V(C_N) = \begin{bmatrix} 0.3401 & 0.2385 & -0.0822 & -0.3401 & -0.2385 & 0.0822 \\ -20.0000 & -1.8031 & 20.0000 & 20.0000 & 1.8031 & -20.0000 \end{bmatrix}$$

$$U_v = \begin{bmatrix} -0.5000 & -0.5000 & 0.3534 & 0.5000 & 0.5000 & -0.3540 \\ 1.0000 & 1.0000 & 1.0000 & -1.0000 & -1.0000 & -1.0000 \end{bmatrix}$$

Solving the LP problem (5.10), Section 5.2 explicitly by using multi-parametric linear programming, one obtains the state space partition in Figure 8.3.

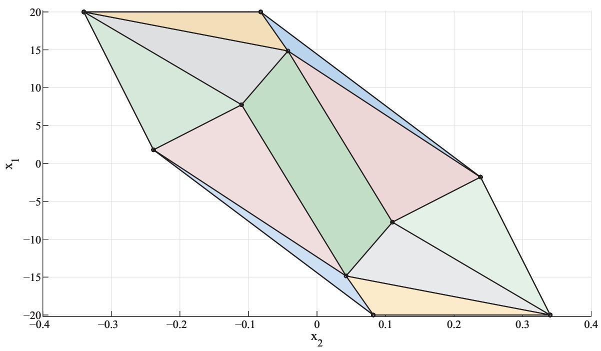
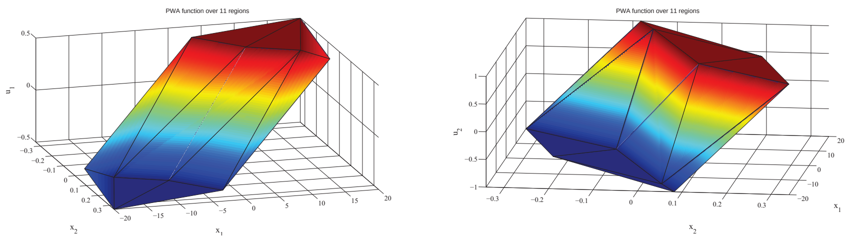



**Fig. 8.3** State space partition of the CSTR system.

Figure 8.4 show the control inputs as a piecewise affine function of state.
The control law over state space partition is




(a) Input $u_1$ (b) Input $u_2$

**Fig. 8.4** Control inputs as a piecewise affine function of state of the CSTR system.

$$
u(k) = \begin{cases}
\begin{bmatrix} -0.50 \\ 1.00 \end{bmatrix} & \text{if } \begin{bmatrix} 1.00 & 0.01 \\ -1.00 & -0.02 \\ -1.00 & 0.02 \end{bmatrix} x(k) \leq \begin{bmatrix} 0.23 \\ 0.04 \\ -0.28 \end{bmatrix} \\
\begin{bmatrix} -1.56 & 0.03 \\ 0.00 & 0.00 \end{bmatrix} x(k) + \begin{bmatrix} -0.07 \\ 1.00 \end{bmatrix} & \text{if } \begin{bmatrix} 1.00 & -0.02 \\ 1.00 & 0.02 \\ -1.00 & -0.01 \end{bmatrix} x(k) \leq \begin{bmatrix} 0.28 \\ 0.21 \\ -0.06 \end{bmatrix} \\
\begin{bmatrix} -9.71 & -0.10 \\ 0.00 & 0.00 \end{bmatrix} x(k) + \begin{bmatrix} 1.63 \\ 1 \end{bmatrix} & \text{if } \begin{bmatrix} -1.00 & -0.02 \\ 1.00 & 0.01 \\ -1.00 & -0.01 \end{bmatrix} x(k) \leq \begin{bmatrix} -0.21 \\ 0.21 \\ -0.07 \end{bmatrix} \\
\begin{bmatrix} -0.57 & -0.03 \\ 7.75 & 0.06 \end{bmatrix} x(k) + \begin{bmatrix} 0.96 \\ 0.44 \end{bmatrix} & \text{if } \begin{bmatrix} 1.00 & 0.01 \\ -1.00 & -0.06 \\ 0 & 1.00 \end{bmatrix} x(k) \leq \begin{bmatrix} 0.07 \\ -0.82 \\ 20.00 \end{bmatrix} \\
\begin{bmatrix} 0 & 0 \\ 9.94 & 0.17 \end{bmatrix} x(k) + \begin{bmatrix} 0.5 \\ -1.35 \end{bmatrix} & \text{if } \begin{bmatrix} 1.00 & -0.01 \\ 1.00 & 0.06 \\ -1.00 & -0.02 \end{bmatrix} x(k) \leq \begin{bmatrix} -0.18 \\ 0.82 \\ -0.04 \end{bmatrix} \\
\begin{bmatrix} 0.50 \\ -1.00 \end{bmatrix} & \text{if } \begin{bmatrix} -1.00 & -0.01 \\ 1.00 & 0.02 \\ 1.00 & -0.02 \end{bmatrix} x(k) \leq \begin{bmatrix} 0.23 \\ 0.04 \\ -0.28 \end{bmatrix} \\
\begin{bmatrix} -1.56 & 0.03 \\ 0.00 & 0.00 \end{bmatrix} x(k) + \begin{bmatrix} 0.07 \\ -1.00 \end{bmatrix} & \text{if } \begin{bmatrix} -1.00 & 0.02 \\ -1.00 & -0.02 \\ 1.00 & 0.01 \end{bmatrix} x(k) \leq \begin{bmatrix} 0.28 \\ 0.21 \\ -0.06 \end{bmatrix} \\
\begin{bmatrix} -9.71 & -0.10 \\ 0.00 & 0.00 \end{bmatrix} x(k) + \begin{bmatrix} -1.63 \\ -1 \end{bmatrix} & \text{if } \begin{bmatrix} 1.00 & 0.02 \\ -1.00 & -0.01 \\ 1.00 & 0.01 \end{bmatrix} x(k) \leq \begin{bmatrix} -0.21 \\ 0.21 \\ -0.07 \end{bmatrix} \\
\begin{bmatrix} -0.57 & -0.03 \\ 7.75 & 0.06 \end{bmatrix} x(k) + \begin{bmatrix} -0.96 \\ -0.44 \end{bmatrix} & \text{if } \begin{bmatrix} -1.00 & -0.01 \\ -1.00 & 0.06 \\ 0 & -1.00 \end{bmatrix} x(k) \leq \begin{bmatrix} 0.07 \\ -0.82 \\ 20.00 \end{bmatrix} \\
\begin{bmatrix} 0 & 0 \\ 9.94 & 0.17 \end{bmatrix} x(k) + \begin{bmatrix} -0.5 \\ 1.35 \end{bmatrix} & \text{if } \begin{bmatrix} -1.00 & 0.01 \\ -1.00 & -0.06 \\ 1.00 & 0.02 \end{bmatrix} x(k) \leq \begin{bmatrix} -0.18 \\ 0.82 \\ -0.04 \end{bmatrix} \\
\begin{bmatrix} -2.72 & 0.03 \\ 17.29 & 0.12 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{if } \begin{bmatrix} -1.00 & 0.01 \\ 1.00 & 0.01 \\ 1.00 & -0.01 \\ -1.00 & -0.01 \end{bmatrix} x(k) \leq \begin{bmatrix} 0.18 \\ 0.06 \\ 0.18 \\ 0.06 \end{bmatrix}
\end{cases}
$$

Figure 8.5 presents state trajectories of the closed loop system for different initial conditions and different realizations of $\alpha(k)$.

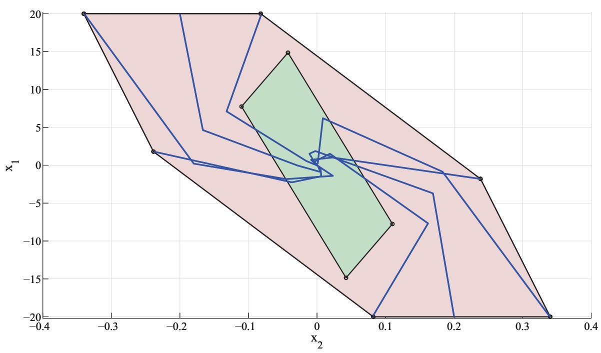



**Fig. 8.5** State trajectories of the closed loop system.

For the initial condition $x(0) = [0.2000 \quad -15.0000]^T$, Figure 8.6 and Figure 8.7 show the state and input trajectories of the closed loop system as a function of time. The solid blue lines are obtained by using the interpolation based control method, while the dashed red lines are obtained by using an algorithm, proposed by Kothare et al. in [78]. From Figure 8.7 and Figure 8.8, it can be observed that there is no much difference between the performance of the interpolation controller and the performance of the Kothare et al. controller. But the point here is that, the algorithm in [78] requires a solution of a semidefinite problem. In our algorithm, the control action is computed on-line by lookup tables and search trees. In addition, the feasible set of Kothare's algorithm is smaller than the feasible set of our algorithm, as can be seen in Figure 8.8.

Figure 8.9 presents the interpolating coefficient, i.e. the Lyapunov function, while the realizations of $\alpha_i(k)$, $i = 1,2,3,4$ as a function of time are depicted in Figure 8.10.

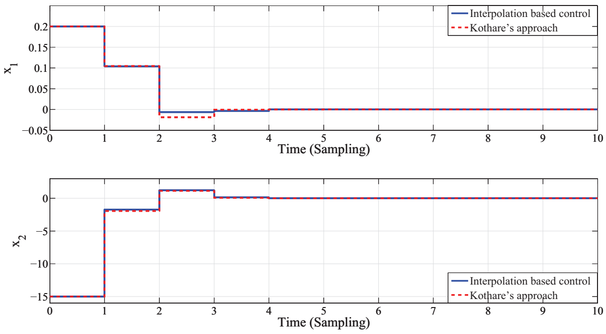



**Fig. 8.6** State trajectories as a function of time. The solid blue lines are obtained by using the interpolation based control method, while the dashed red lines are obtained by using an algorithm, proposed by Kothare et al. in [78].

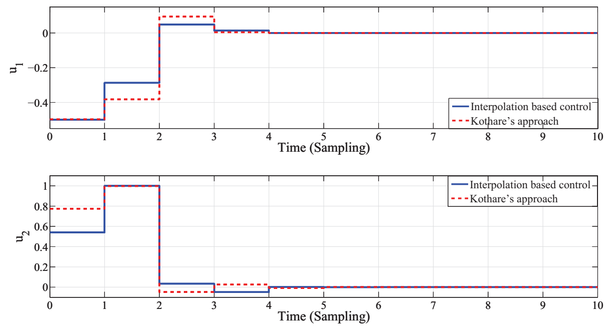



**Fig. 8.7** Input trajectories as a function of time. The solid blue lines are obtained by using the interpolation based control method, while the dashed red lines are obtained by using an algorithm, proposed by Kothare et al. in [78].

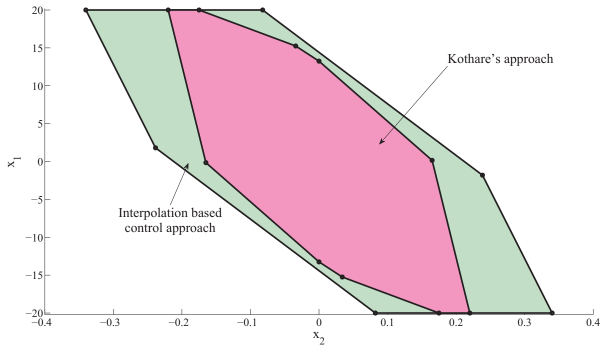



**Fig. 8.8** Feasible sets of interpolation based control algorithm and algorithm in [78].

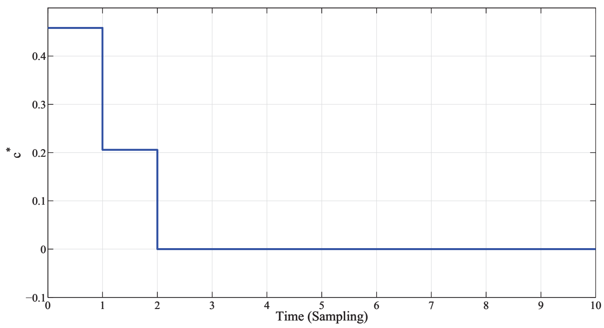



**Fig. 8.9** Interpolating coefficient as a function of time.

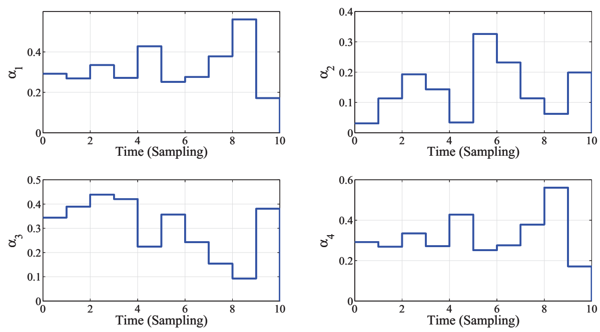



**Fig. 8.10** The realization of $\alpha(k)$ as a function of time.

# Part IV
# Conclusions and Future directions

# Chapter 9
# Conclusions and Future directions

## 9.1 Conclusions

The central idea behind the present manuscript was to develop a framework for the synthesis of robust constrained (output) feedback controllers. The thesis presents the results for linear discrete-time systems. The main contributions of the thesis are summarized below.

### *9.1.1 Domain of attraction*

Contributions to the analysis of constrained control dynamics:

- Improve the existing methods for computing an invariant ellipsoid for uncertain systems, by using the linear differential inclusion framework for modeling the saturated function proposed by Hu et al. [59].
- Propose a new method for designing a saturated control law based on LMI synthesis based on the same modeling framework.
- Present a new method for computing an polyhedral invariant set for a system with a saturated controller. This polyhedral invariant set may be suitable for interpolation based control via linear programming, when the vertex control scheme might exhibit high complexity.

### *9.1.2 Interpolation based control*

To overcome the performance weakness of the classical vertex controller on the one hand and the complexity of predictive control laws on the other hand, we propose interpolation, based on linear programming between the global vertex controller and a local more aggressive controller. This interpolation scheme shares a few basic principles which qualifies it as a generic one. It can be applied efficiently with discrete-time linear uncertain or time-varying systems subject to bounded disturbances. A few fundamental contributions were presented:

- A simple Lyapunov function is used for the proof of closed loop stability using the interpolation index as main ingredient.
- In comparison with the MPC method, the interpolation based control has extremely simple and fast LP-computations in the implicit case, and hence requires less complex on-line computations. For the explicit interpolation based control method, the control value is a piecewise affine function of state defined over a polyhedral subdivision of the feasible set with in general fewer subdivisions than explicit MPC.
- The solution of the interpolation via linear programming might not be unique due to the semi-positive definiteness of the objective function, i.e. the interpolating coefficient. Hence, the interpolation via quadratic programming was proposed in order to guarantee the uniqueness of the optimal solutions by adapting the QP results of Rossiter et al. [132], [123].
- For the interpolation based on quadratic programming, in order to fully utilize the capability of actuators and guarantee the satisfaction of input constraints, a saturation function on the input is considered for the control law deployment. This saturation function will guarantee the constraint satisfaction of the input of the controlled plant. Hence, an interpolation scheme between several saturated controllers is obtained.
- In the presence of bounded disturbances, an input to state stability (ISS) framework is employed. Two control algorithms were proposed. The first one is based on the interpolation approach and the optimization problem can be placed in the class of a semi-quadratic optimization problem. The second one is based on the linear decomposition principle. In this case, the on-line optimization problem can be formulated as a quadratic programming problem.

### 9.1.3 LMI synthesis condition

By preserving an optimization-based control framework for constrained dynamical systems the LMI routines have been investigated, often with an *interpolation* background in the synthesis procedure

- A necessary and sufficient condition for invariance of ellipsoid for a discrete-time linear uncertain or time-varying system with bounded disturbances was provided.
- A control synthesis procedure was presented, which aims to minimize the peak value of the state variable over the peak value of the disturbance.
- For high dimensional systems, the polyhedral based control methods might be impractical, since the number of vertices or half-spaces may lead to an exponential complexity. For these cases, ellipsoids seem to be the suitable class of sets in the interpolation. It was shown that the convex hull of a set of invariant

ellipsoids for discrete time linear uncertain or parameter-varying systems with saturated controllers is controlled invariant. A continuous feedback control law is constructed based on solving an LMI problem at each time instant. Some geometrical properties of this algorithm were highlighted.

## 9.2 Future directions

By using interpolation based control method, several potentially interesting future research topics can be identified. In this section a few of the most interesting research directions are highlighted.

### *9.2.1 Interpolation based control for non-linear system*

When applying the interpolation based control technique, we limited ourselves to linear systems. There are two main reasons for this. Firstly, it is easy to compute an invariant set for linear systems. Secondly, for the recursive feasibility and asymptotic stability proof of the interpolation based control scheme, the linearity of the systems and the convexity of the feasible regions bring important structural advantages.

The analysis and control design become significantly more difficult, as soon as we renounce at linearity and enlarge the class of dynamics to include nonlinear models. To the best of the author's knowledge, there is no constructive procedure for computing the feasible invariant sets for non-linear systems. In addition, these sets are generally non-convex. As such, even if the interpolation principle does hold, the practical computation of the level sets and the associated vertex control will not be straightforward.

However, if the linear assumptions are relaxed but a certain structure is preserved, the situation is completely different for example by restricting the attention to the class of homogeneous non-linear systems.

**Definition 9.1. (Positively homogeneous system)** The system

$$x(k+1) = f(x(k), u(k)) \tag{9.1}$$

is positively homogeneous of degree $p$ if for any real scalar $\lambda$ it holds that

$$f(\lambda x, \lambda u) = \lambda^p f(x, u)$$

For system (9.1) it is assumed that $f(0,0) = 0$. Below we provide some of our preliminary results on the analysis and control design for the class of homogeneous systems. With this aim, the following definition is introduced.

**Definition 9.2. (Star-shape set)** [134], [135], [146] A star-shape set $S$ is a connected and generally non-convex set for which exists a non-empty kernel

$$kern(S) = \{s \in S : s + \lambda(x - s) \in S, \forall x \in S, \lambda \in [0,\ 1]\}$$

A set is radiant or star-shaped at the origin if $0 \in kern(S)$. This point represents a center point and the star shapeness property assures that any segment of line starting from the center to an arbitrary point of the set is included in the set.

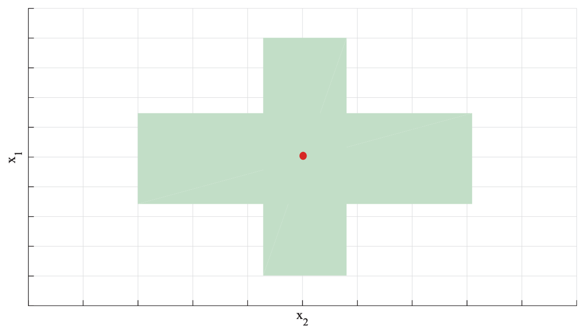



**Fig. 9.1** Example of star-shaped set.

The following theorem holds true

**Theorem 9.1.** *For system 9.1, if the set S is controlled invariant, then it is radiant (i.e. star-shaped set at the origin).*

*Proof.* It has to be proved that any segment of line starting from the origin to an arbitrary point of $S$ is included in the set $S$. For this purpose, it is sufficient to prove that any segment connecting the origin and any point on the boundary of $S$ is contained in $S$.

Consider any point $x \in S$ and let $\tilde{x}$ be the unique intersection of the ray starting from the origin and passing though $x$ with the boundary of the set $S$. Clearly, from the convexity argument, it is possible to write

$$x = \lambda \tilde{x} \tag{9.2}$$

with $0 \le \lambda \le 1$.

Consider the following control law

$$u = \lambda \tilde{u} \tag{9.3}$$

where $\tilde{u}$ is the control action that pushes $\tilde{x}$ inside $S$. It follows that

$$x(k+1) = f(x(k), u(k)) = f(\lambda \tilde{x}(k), \lambda \tilde{u}(k)) = \lambda^p f(\tilde{x}(k), \tilde{u}(k)) \in S$$

Hence $S$ is a radiant set. □

*Remark 9.1.* It is clear that for the system 9.1 if the order $p \geq 1$, then the control law (9.2), (9.3) guarantees asymptotic stability for all states $x(0) \in S$, since $\lambda$ will be a Lyapunov function. Hence we obtain a classical vertex control law for the class of homogeneous systems.

There still remains two open questions for the constrained homogeneous system

1. Although the star-shaped topology of the controlled invariant set $S$ being known, tractable procedures for calculating such sets with arbitrary precision are not known.
2. Clearly, as for the linear system case, with the vertex control law, the full control range is exploited only on the border of the set $S$, with progressively smaller control action when state approaches the origin. Therefore the time to regulate the plant to the origin is longer than necessary. An interesting and open question is how to interpolate between the global vertex control with some local more aggressive controller to get a better performance.

### 9.2.2 Obstacle avoidance

So far, the *convex* state and input constraints have been considered. Non-convex polyhedral constraints, defined as the non-convex union of a finite number of polyhedral sets, arise naturally in problems such as obstacle avoidance, which is inherently non-convex. The importance of this problem is stressed with applications in several engineering fields (see, for example, managing multiple agents [102], pedestrian behavior in the crowd [40], telescope manipulation [139] and so on) and has been addressed using different approaches: dynamic programming [148], optimal control, Lyapunov methods, viability theory [10]. Translated in the predictive control framework, the obstacle avoidance it is known to be a difficult problem, since it leads to non-convex constraints handling [39].

Clearly, for linear systems, by using the interpolation based control approach it can be shown that the optimal decomposition at time instant $k$ is a feasible decomposition at time instant $k+1$. Hence the objective function is a Lyapunov function or in the other words, asymptotic stability is guaranteed. However due to the non-convex state and control constraints, the feasible sets are generally non-convex. That may lead to an infeasibility problem. This issue should be further investigated.

## References


1. Abedor, J., Nagpal, K., Poolla, K.: A linear matrix inequality approach to peak-to-peak gain minimization. International Journal of Robust and Nonlinear Control **6**(9-10), 899–927 (1996)
2. Ackermann, J., Bartlett, A., Kaesbauer, D., Sienel, W., Steinhauser, R.: Robust control: Systems with uncertain physical parameters. Springer-Verlag New York, Inc. (2001)
3. Alamir, M.: Stabilization of nonlinear system by receding horizon control scheme: A parametrized approach for fast systems (2006)
4. Alamir, M.: A framework for real-time implementation of low-dimensional parameterized nmpc. Automatica (2011)
5. Alamir, M., Bornard, G.: Stability of a truncated infinite constrained receding horizon scheme: The general discrete nonlinear case. Automatica **31**(9), 1353–1356 (1995)
6. Alamo, T., Cepeda, A., Limon, D.: Improved computation of ellipsoidal invariant sets for saturated control systems. In: Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on, pp. 6216–6221. IEEE (2005)
7. Alamo, T., Cepeda, A., Limon, D., Camacho, E.: Estimation of the domain of attraction for saturated discrete-time systems. International journal of systems science **37**(8), 575–583 (2006)
8. Anderson, B., Moore, J.: Optimal control: linear quadratic methods, vol. 1. Prentice Hall Englewood Cliffs, NJ (1990)
9. Åström, K., Murray, R.: Feedback systems: an introduction for scientists and engineers. Princeton Univ Pr (2008)
10. Aubin, J., Bayen, A., Saint-Pierre, P.: Viability Theory: New Directions. Springer Verlag (2011)
11. Bacic, M., Cannon, M., Lee, Y., Kouvaritakis, B.: General interpolation in mpc and its advantages. Automatic Control, IEEE Transactions on **48**(6), 1092–1096 (2003)
12. Balas, E.: Projection with a minimal system of inequalities. Computational Optimization and Applications **10**(2), 189–193 (1998)
13. Bellman, R.: On the theory of dynamic programming. Proceedings of the National Academy of Sciences of the United States of America **38**(8), 716 (1952)
14. Bellman, R.: The theory of dynamic programming. Bull. Amer. Math. Soc **60**(6), 503–515 (1954)
15. Bellman, R.: Dynamic programming and lagrange multipliers. Proceedings of the National Academy of Sciences of the United States of America **42**(10), 767 (1956)
16. Bellman, R.: Dynamic programming. Science **153**(3731), 34 (1966)
17. Bellman, R., Dreyfus, S.: Applied dynamic programming (1962)
18. Bemporad, A., Borrelli, F., Morari, M.: Model predictive control based on linear programming˜ the explicit solution. Automatic Control, IEEE Transactions on **47**(12), 1974–1985 (2002)
19. Bemporad, A., Filippi, C.: Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. Journal of optimization theory and applications **117**(1), 9–38 (2003)
20. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.: The explicit linear quadratic regulator for constrained systems. Automatica **38**(1), 3–20 (2002)
21. Bitsoris, G.: Positively invariant polyhedral sets of discrete-time linear systems. International Journal of Control **47**(6), 1713–1726 (1988)
22. Blanchini, F.: Nonquadratic lyapunov functions for robust control. Automatica **31**(3), 451–461 (1995)
23. Blanchini, F.: Set invariance in control. Automatica **35**(11), 1747–1767 (1999)
24. Blanchini, F., Miani, S.: On the transient estimate for linear systems with time-varying uncertain parameters. Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on **43**(7), 592–596 (1996)

25. Blanchini, F., Miani, S.: Constrained stabilization via smooth lyapunov functions. Systems & control letters **35**(3), 155–163 (1998)
26. Blanchini, F., Miani, S.: Stabilization of lpv systems: state feedback, state estimation and duality. In: Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, vol. 2, pp. 1492–1497. IEEE (2003)
27. Blanchini, F., Miani, S.: Set-theoretic methods in control. Springer (2008)
28. Borrelli, F.: Constrained optimal control of linear and hybrid systems, vol. 290. Springer Verlag (2003)
29. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear matrix inequalities in system and control theory, vol. 15. Society for Industrial Mathematics (1994)
30. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ Pr (2004)
31. Bronstein, E.: Approximation of convex sets by polytopes. Journal of Mathematical Sciences **153**(6), 727–762 (2008)
32. Camacho, E., Bordons, C.: Model predictive control. Springer Verlag (2004)
33. Clarke, D., Mohtadi, C., Tuffs, P.: Generalized predictive control–part i. the basic algorithm. Automatica **23**(2), 137–148 (1987)
34. Clarke, F.: Optimization and nonsmooth analysis, vol. 5. Society for Industrial Mathematics (1990)
35. Cwikel, M., Gutman, P.: Convergence of an algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states. Automatic Control, IEEE Transactions on **31**(5), 457–459 (1986)
36. Daafouz, J., Bernussou, J.: Parameter dependent lyapunov functions for discrete time systems with time varying parametric uncertainties. Systems & control letters **43**(5), 355–359 (2001)
37. Dantzig, G.: Fourier-motzkin elimination and its dual. Tech. rep., DTIC Document (1972)
38. Dreyfus, S.: Some types of optimal control of stochastic systems. Tech. rep., DTIC Document (1963)
39. Earl, M., D'Andrea, R.: Modeling and control of a multi-agent system using mixed integer linear programming. In: Decision and Control, 2002, Proceedings of the 41st IEEE Conference on, vol. 1, pp. 107–111. IEEE (2002)
40. Fang, Z., Song, W., Zhang, J., Wu, H.: Experiment and modeling of exit-selecting behaviors during a building evacuation. Physica A: Statistical Mechanics and its Applications **389**(4), 815–824 (2010)
41. Fukuda, K.: cdd/cdd+ reference manual. Institute for operations Research ETH-Zentrum, ETH-Zentrum (1997)
42. Fukuda, K.: Frequently asked questions in polyhedral computation. Report http://www. ifor. math. ethz. ch/fukuda/polyfaq/polyfaq. html, ETH, Zürich (2000)
43. Gahinet, P., Nemirovskii, A., Laub, A., Chilali, M.: The lmi control toolbox. In: Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on, vol. 3, pp. 2038–2041. IEEE (1994)
44. Geyer, T., Torrisi, F., Morari, M.: Optimal complexity reduction of piecewise affine models based on hyperplane arrangements. In: American Control Conference, 2004. Proceedings of the 2004, vol. 2, pp. 1190–1195. IEEE (2004)
45. Gilbert, E., Tan, K.: Linear systems with state and control constraints: The theory and application of maximal output admissible sets. Automatic Control, IEEE Transactions on **36**(9), 1008–1020 (1991)
46. Goodwin, G., Graebe, S., Salgado, M.: Control system design, vol. 1. Prentice Hall New Jersey (2001)
47. Goodwin, G., Seron, M., De Dona, J.: Constrained control and estimation: an optimisation approach. Springer Verlag (2005)
48. Grancharova, A., Johansen, T.: Explicit Nonlinear Model Predictive Control: Theory and Applications, vol. 429. Springer Verlag (2012)
49. Grant, M., Boyd, S.: Cvx: Matlab software for disciplined convex programming. Available httpstanford edu boydcvx (2008)
50. Grieder, P., Morari, M.: Complexity reduction of receding horizon control. In: Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, vol. 3, pp. 3179–3190. IEEE (2003)

51. Grimm, G., Hatfield, J., Postlethwaite, I., Teel, A., Turner, M., Zaccarian, L.: Antiwindup for stable linear systems with input saturation: an lmi-based synthesis. Automatic Control, IEEE Transactions on **48**(9), 1509–1525 (2003)
52. Grünbaum, B.: Convex polytopes, vol. 221. Springer Verlag (2003)
53. Gutman, P.: A linear programming regulator applied to hydroelectric reservoir level control. Automatica **22**(5), 533–541 (1986)
54. Gutman, P., Cwikel, M.: Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states. Automatic Control, IEEE Transactions on **31**(4), 373–376 (1986)
55. Gutman, P., Cwikel, M.: An algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states. IEEE TRANS. AUTOM. CONTROL. **32**(3), 251–254 (1987)
56. Gutman, P., Hagander, P.: A new design of constrained controllers for linear systems. Automatic Control, IEEE Transactions on **30**(1), 22–33 (1985)
57. Hu, T., Lin, Z.: Control systems with actuator saturation: analysis and design. Birkhauser (2001)
58. Hu, T., Lin, Z.: Composite quadratic lyapunov functions for constrained control systems. Automatic Control, IEEE Transactions on **48**(3), 440–450 (2003)
59. Hu, T., Lin, Z., Chen, B.: Analysis and design for discrete-time linear systems subject to actuator saturation. Systems & Control Letters **45**(2), 97–112 (2002)
60. Hu, T., Lin, Z., Chen, B.: An analysis and design method for linear systems subject to actuator saturation and disturbance. Automatica **38**(2), 351–359 (2002)
61. Jiang, Z., Wang, Y.: Input-to-state stability for discrete-time nonlinear systems. Automatica **37**(6), 857–869 (2001)
62. Johansen, T., Grancharova, A.: Approximate explicit constrained linear model predictive control via orthogonal search tree. Automatic Control, IEEE Transactions on **48**(5), 810–815 (2003)
63. Jones, C., Kerrigan, E., Maciejowski, J., of Cambridge. Engineering Dept, U.: Equality set projection: A new algorithm for the projection of polytopes in halfspace representation. University of Cambridge, Dept. of Engineering (2004)
64. Jones, C., Morari, M.: The double description method for the approximation of explicit mpc control laws. In: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, pp. 4724–4730. IEEE (2008)
65. Jones, C., Morari, M.: Approximate explicit mpc using bilevel optimization. In: European Control Conference (2009)
66. Jones, C., Morari, M.: Polytopic approximation of explicit model predictive controllers. Automatic Control, IEEE Transactions on **55**(11), 2542–2553 (2010)
67. Jönsson, U.: A lecture on the s-procedure. Lecture Note at the Royal Institute of technology, Sweden (2001)
68. Kalman, R.: Contributions to the theory of optimal control. Bol. Soc. Mat. Mexicana **5**(2), 102–119 (1960)
69. Keerthi, S., Sridharan, K.: Solution of parametrized linear inequalities by fourier elimination and its applications. Journal of Optimization Theory and Applications **65**(1), 161–169 (1990)
70. Kerrigan, E.: Robust constraint satisfaction: Invariant sets and predictive control. Department of Engineering, University of Cambridge, UK (2000)
71. Kerrigan, E.: Invariant set toolbox for matlab (2003)
72. Kevrekidis, I., Aris, R., Schmidt, L.: The stirred tank forced. Chemical engineering science **41**(6), 1549–1560 (1986)
73. Khalil, H., Grizzle, J.: Nonlinear systems, vol. 3. Prentice hall (1992)
74. Khlebnikov, M., Polyak, B., Kuntsevich, V.: Optimization of linear systems subject to bounded exogenous disturbances: The invariant ellipsoid technique. Automation and Remote Control **72**(11), 2227–2275 (2011)
75. Kojima, M., Mizuno, S., Yoshise, A.: A primal-dual interior point algorithm for linear programming. Springer-Verlag New York, Inc. (1988)

76. Kolmanovsky, I., Gilbert, E.: Theory and computation of disturbance invariant sets for discrete-time linear systems. Mathematical Problems in Engineering **4**(4), 317–367 (1998)
77. Kolmogorov, A., Fomin, S.: Elements of the theory of functions and functional analysis, vol. 1. Dover publications (1999)
78. Kothare, M., Balakrishnan, V., Morari, M.: Robust constrained model predictive control using linear matrix inequalities. Automatica **32**(10), 1361–1379 (1996)
79. Kothare, M., Campo, P., Morari, M., Nett, C.: A unified framework for the study of anti-windup designs. Automatica **30**(12), 1869–1883 (1994)
80. Kurzhanskiy, A., Varaiya, P.: Ellipsoidal toolbox (et). In: Decision and Control, 2006 45th IEEE Conference on, pp. 1498–1503. IEEE (2006)
81. Kvasnica, M., Fikar, M.: Clipping-based complexity reduction in explicit mpc. Automatic Control, IEEE Transactions on (99), 1–1 (2011)
82. Kvasnica, M., Grieder, P., Baotić, M., Morari, M.: Multi-parametric toolbox (mpt). Hybrid Systems: Computation and Control pp. 121–124 (2004)
83. Kvasnica, M., Löfberg, J., Fikar, M.: Stabilizing polynomial approximation of explicit mpc. Automatica (2011)
84. Kvasnica, M., Lo?fberg, J., Herceg, M., Cirka, L., Fikar, M.: Low-complexity polynomial approximation of explicit mpc via linear programming. In: American Control Conference (ACC), 2010, pp. 4713–4718. IEEE (2010)
85. Kvasnica, M., Rauová, I., Fikar, M.: Simplification of explicit mpc feedback laws via separation functions. In: IFAC World Congress. Milano, Italy (2011)
86. Kwakernaak, H., Sivan, R.: Linear optimal control systems, vol. 172. Wiley-Interscience New York (1972)
87. Langson, W., Chryssochoos, I., Raković, S., Mayne, D.: Robust model predictive control using tubes. Automatica **40**(1), 125–133 (2004)
88. Lazar, M., Munoz de la Pena, D., Heemels, W., Alamo, T.: On input-to-state stability of min-max nonlinear model predictive control. Systems & Control Letters **57**(1), 39–48 (2008)
89. Lee, E.: Foundations of optimal control theory. Tech. rep., DTIC Document (1967)
90. Lewis, F., Syrmos, V.: Optimal control. Wiley-Interscience (1995)
91. Liberzon, D.: Switching in systems and control. Springer (2003)
92. Ljung, L.: System identification. Wiley Online Library (1999)
93. Loechner, V., Wilde, D.: Parameterized polyhedra and their vertices. International Journal of Parallel Programming **25**(6), 525–549 (1997)
94. Lofberg, J.: Yalmip: A toolbox for modeling and optimization in matlab. In: Computer Aided Control Systems Design, 2004 IEEE International Symposium on, pp. 284–289. IEEE (2004)
95. Luenberger, D.: Optimization by vector space methods. Wiley-Interscience (1997)
96. Maciejowski, J.: Predictive control: with constraints. Pearson education (2002)
97. Macki, J., Strauss, A.: Introduction to optimal control theory. Springer (1982)
98. Marlin, T.: Process control: designing processes and control systems for dynamic performance. McGraw-Hill New York (1995)
99. Marruedo, D., Alamo, T., Camacho, E.: Input-to-state stable mpc for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: Decision and Control, 2002, Proceedings of the 41st IEEE Conference on, vol. 4, pp. 4619–4624. IEEE (2002)
100. Mayne, D., Rawlings, J., Rao, C., Scokaert, P.: Constrained model predictive control: Stability and optimality. Automatica **36**(6), 789–814 (2000)
101. Motzkin, T., Raiffa, H., Thompson, G., Thrall, R.: The double description method (1953)
102. Murphey, R., Pardalos, P.: Cooperative control and optimization, vol. 66. Kluwer Academic Pub (2002)
103. Nam, N., Olaru, S., Hovd, M.: Patchy approximate explicit model predictive control. In: Control Automation and Systems (ICCAS), 2010 International Conference on, pp. 1287–1292. IEEE (2010)
104. Nazin, S., Polyak, B., Topunov, M.: Rejection of bounded exogenous disturbances by the method of invariant ellipsoids. Automation and Remote Control **68**(3), 467–486 (2007)
105. Nesterov, Y., Nemirovsky, A.: Interior point polynomial methods in convex programming (1994)

106. Nguyen, H., Gutman, P., Olaru, S., Hovd, M.: Explicit constraint control based on interpolation techniques for time-varying and uncertain linear discrete-time systems. In: World Congress, vol. 18, pp. 5741–5746 (2011)
107. Nguyen, H., Gutman, P., Olaru, S., Hovd, M.: An interpolation approach for robust constrained output feedback (2011)
108. Nguyen, H., Gutman, P., Olaru, S., Hovd, M., Colledani, F.: Improved vertex control for time-varying and uncertain linear discrete-time systems with control and state constraints. In: American Control Conference (ACC), 2011, pp. 4386–4391. IEEE (2011)
109. Nguyen, H., Olaru, S., Gutman, P., Hovd, M.: Robust output feedback interpolation based control for constrained linear systems. Informatics in Control, Automation and Robotics pp. 21–36
110. Nguyen, H., Olaru, S., Gutman, P., Hovd, M.: Constrained interpolation-based control for polytopic uncertain systems (2011)
111. Nguyen, H., Olaru, S., Gutman, P., Hovd, M.: Improved vertex control for a ball and plate system (2012)
112. Nguyen, H., Olaru, S., Gutman, P., Hovd, M.: Interpolation based control for constrained linear time-varying or uncertain systems in the presence of disturbances (2012)
113. Nguyen, H., Olaru, S., Hovd, M.: A patchy approximation of explicit model predictive control (2012)
114. Nguyen, N., Olaru, S.: Hybrid modeling and optimal control of juggling systems. International Journal of systems Science (2011). DOI 10.1080/00207721.2011.600514. URL http://www.tandfonline.com/doi/abs/10.1080/00207721.2011.600514
115. Nguyen, N., Olaru, S., Stoican, F.: On maximal robustly positively invariant sets. In: Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO), p. 6 pages. Noordwijkerhout, Pays-Bas (2011). URL http://hal-supelec.archives-ouvertes.fr/hal-00593078
116. Olaru, S., De Doná, J., Seron, M., Stoican, F.: Positive invariant sets for fault tolerant multisensor control schemes. International Journal of Control **83**(12), 2622–2640 (2010)
117. Olaru, S., Dumur, D.: A parameterized polyhedra approach for explicit constrained predictive control. In: Decision and Control, 2004. CDC. 43rd IEEE Conference on, vol. 2, pp. 1580–1585. IEEE (2004)
118. Olaru, S., Dumur, D.: On the continuity and complexity of control laws based on multiparametric linear programs. In: Decision and Control, 2006 45th IEEE Conference on, pp. 5465–5470. IEEE (2006)
119. Pérez, M., Albertos, P.: Self-oscillating and chaotic behaviour of a pi-controlled cstr with control valve saturation. Journal of Process Control **14**(1), 51–59 (2004)
120. Pérez, M., Font, R., Montava, M.: Regular self-oscillating and chaotic dynamics of a continuous stirred tank reactor. Computers & chemical engineering **26**(6), 889–901 (2002)
121. Pistikopoulos, E., Georgiadis, M., Dua, V.: Multi-parametric model-based control: theory and applications, vol. 2. Vch Verlagsgesellschaft Mbh (2007)
122. Pistikopoulos, E., Georgiadis, M., Dua, V.: Multi-parametric programming: theory, algorithms, and applications, vol. 1. Wiley-VCH Verlag GmbH (2007)
123. Pluymers, B., Rossiter, J., Suykens, J., De Moor, B.: Interpolation based mpc for lpv systems using polyhedral invariant sets. In: American Control Conference, 2005. Proceedings of the 2005, pp. 810–815. IEEE (2005)
124. Polyak, B.: Convexity of quadratic transformations and its use in control and optimization. Journal of Optimization Theory and Applications **99**(3), 553–583 (1998)
125. Pontryagin, L., Gamkrelidze, R.: The mathematical theory of optimal processes, vol. 4. CRC (1986)
126. Propoi, A.: Use of linear programming methods for synthesizing sampled-data automatic systems. Automation and Remote Control **24**(7), 837–844 (1963)
127. Rakovic, S., Kerrigan, E., Kouramas, K., Mayne, D.: Invariant approximations of the minimal robust positively invariant set. Automatic Control, IEEE Transactions on **50**(3), 406–410 (2005)

128. Rockafellar, R.: Convex analysis, vol. 28. Princeton Univ Pr (1997)
129. Rossiter, J.: Model-based predictive control: a practical approach. CRC (2003)
130. Rossiter, J., Ding, Y.: Interpolation methods in model predictive control: an overview. International Journal of Control **83**(2), 297–312 (2010)
131. Rossiter, J., Grieder, P.: Using interpolation to improve efficiency of multiparametric predictive control. Automatica **41**(4), 637–643 (2005)
132. Rossiter, J., Kouvaritakis, B., Bacic, M.: Interpolation based computationally efficient predictive control. International Journal of Control **77**(3), 290–301 (2004)
133. Rossiter, J., Kouvaritakis, B., Cannon, M.: Computationally efficient algorithms for constraint handling with guaranteed stability and near optimality. International Journal of Control **74**(17), 1678–1689 (2001)
134. Rubinov, A., Sharikov, E.: Star-shaped separability with applications. Journal of Convex Analysis **13**(3/4), 849 (2006)
135. Rubinov, A., Yagubov, A.: The space of star-shaped sets and its applications in nonsmooth optimization. Quasidifferential Calculus pp. 176–202 (1986)
136. Russo, L., Bequette, B.: Effect of process design on the open-loop behavior of a jacketed exothermic cstr. Computers & chemical engineering **20**(4), 417–426 (1996)
137. Scherer, C., Weiland, S.: Linear matrix inequalities in control. Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands (2000)
138. Schmitendorf, W., Barmish, B.: Null controllability of linear systems with constrained controls. SIAM Journal on control and optimization **18**, 327 (1980)
139. Schneider, J.: Pathway toward a mid-infrared interferometer for the direct characterization of exoplanets. Arxiv preprint arXiv:0906.0068 (2009)
140. Scibilia, F., Olaru, S., Hovd, M., et al.: Approximate explicit linear mpc via delaunay tessellation (2009)
141. Seron, M., Goodwin, G., Doná, J.: Characterisation of receding horizon control for constrained linear systems. Asian Journal of Control **5**(2), 271–286 (2003)
142. Skogestad, S., Postlethwaite, I.: Multivariable feedback control: analysis and design, vol. 2. Wiley (2007)
143. Smith, C., Corripio, A.: Principles and practice of automatic process control, vol. 2. Wiley New York (1985)
144. Sontag, E.: Algebraic approach to bounded controllability of linear systems. International Journal of Control **39**(1), 181–188 (1984)
145. Sontag, E., Wang, Y.: On characterizations of the input-to-state stability property. Systems & Control Letters **24**(5), 351–359 (1995)
146. Stoican, F.: Fault tolerant control based on set-theoretic methods. Ph.D. thesis, ÉCOLE SUPÉRIEURE DÉLECTRICITÉ (2011)
147. Sturm, J.: Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optimization methods and software **11**(1-4), 625–653 (1999)
148. Sundar, S., Shiller, Z.: Optimal obstacle avoidance based on the hamilton-jacobi-bellman equation. Robotics and Automation, IEEE Transactions on **13**(2), 305–310 (1997)
149. Tarbouriech, S., Garcia, G., Glattfelder, A.: Advanced strategies in control systems with input and output constraints, vol. 346. Springer Verlag (2007)
150. Tarbouriech, S., Garcia, G., da Silva, J., Queinnec, I.: Stability and stabilization of linear systems with saturating actuators. Springer (2011)
151. Tarbouriech, S., Turner, M.: Anti-windup design: an overview of some recent advances and open problems. Control Theory & Applications, IET **3**(1), 1–19 (2009)
152. Taylor, C., Chotai, A., Young, P.: State space control system design based on non-minimal state-variable feedback: further generalization and unification results. International Journal of Control **73**(14), 1329–1345 (2000)
153. Tøndel, P., Johansen, T., Bemporad, A.: An algorithm for multi-parametric quadratic programming and explicit mpc solutions. Automatica **39**(3), 489–497 (2003)
154. Veres, S.: Geometric bounding toolbox (gbt) for matlab. Official website: http://www. sysbrain. com (2003)

155. Wan, Z., Kothare, M.: An efficient off-line formulation of robust model predictive control using linear matrix inequalities. Automatica **39**(5), 837–846 (2003)
156. Wang, Y., Boyd, S.: Fast model predictive control using online optimization. Control Systems Technology, IEEE Transactions on **18**(2), 267–278 (2010)
157. Zaccarian, L., Teel, A.: A common framework for anti-windup, bumpless transfer and reliable designs. Automatica **38**(10), 1735–1744 (2002)
158. Ziegler, G.: Lectures on polytopes, vol. 152. Springer (1995)

## Abstract :

A fundamental problem in automatic control is the control of uncertain plants in the presence of input and state or output constraints. An elegant and theoretically most satisfying framework is represented by optimal control policies which, however, rarely gives an analytical feedback solution, and oftentimes builds on numerical solutions (approximations).

Therefore, in practice, the problem has seen many ad-hoc solutions, such as *override control*, *anti-windup*, as well as modern techniques developed during the last decades usually based on state space models. One of the popular example is *Model Predictive Control* (MPC) where an optimal control problem is solved at each sampling instant, and the element of the control vector meant for the nearest sampling interval is applied. In spite of the increased computational power of control computers, MPC is at present mainly suitable for low-order, nominally linear systems. The robust version of MPC is conservative and computationally complicated, while the *explicit* version of MPC that gives an affine state feedback solution involves a very complicated division of the state space into polyhedral cells.

In this thesis a novel and computationally cheap solution is presented for linear, time-varying or uncertain, discrete-time systems with polytopic bounded control and state (or output) vectors, with bounded disturbances. The approach is based on the interpolation between a stabilizing, outer controller that respects the control and state constraints, and an inner, more aggressive controller, designed by any method that has a robustly positively invariant set within the constraints. A simple Lyapunov function is used for the proof of closed loop stability.

In contrast to MPC, the new interpolation based controller is not necessarily employing an optimization criterion inspired by performance. In its explicit form, the cell partitioning is simpler that the MPC counterpart. For the implicit version, the on-line computational demand can be restricted to the solution of one linear program or quadratic program. Several simulation examples are given, including uncertain linear systems with output feedback and disturbances. Some examples are compared with MPC. The control of a laboratory ball-and-plate system is also demonstrated. It is believed that the new controller might see wide-spread use in industry, including the automotive industry, also for the control of fast, high-order systems with constraints.

## Résumé :

Un problème fondamental à résoudre en Automatique réside dans la commande des systèmes incertains soumis à des contraintes portant sur ses entrées, ses états ou ses sorties. Ce problème peut être théoriquement résolu au moyen d'une commande optimale. Cependant la commande optimale par principe n'offre que très rarement une solution sous forme de commande par retour d'état ou retour de sortie et ne fournit souvent une trajectoire optimale que par l'intermédiaire d'une résolution numérique approchée.

Par conséquent, dans la pratique, le problème peut être approché par de nombreuses méthodes, telles que la commande dite « override » ou utilisant des dispositifs « d'anti-windup ». Une autre solution, qui a gagné en popularité au cours des dernières décennies, est la commande prédictive. Cette méthode propose de résoudre un problème de commande optimale à chaque instant d'échantillonnage, appliquant uniquement le premier échantillon de la séquence de commande obtenue selon le principe de l'horizon glissant. En dépit de l'augmentation de la puissance de calcul temps-réel, la commande prédictive est à l'heure actuelle principalement indiquée lorsque l'ordre du système est faible, bien connu, et souvent dans un cadre linéaire. La version robuste de la commande prédictive s'avère souvent conservatrice et complexe à mettre en œuvre, alors que la version explicite de la commande prédictive donnant une solution affine par morceaux implique un fractionnement de l'espace d'état en régions polyédrales, fractionnement souvent complexe.

Dans cette thèse, une solution élégante et peu coûteuse en temps de calcul est présentée dans un contexte de systèmes linéaires, variant dans le temps ou incertains. Les développements se concentrent sur les systèmes à temps discret avec contraintes polyédrales sur l'entrée et l'état (ou la sortie), sujets à des perturbations bornées. Cette solution est fondée sur l'interpolation entre un correcteur stabilisant dédié à la région extérieure qui respecte les contraintes sur l'entrée et l'état, et un autre relatif à la région intérieure, ce dernier ayant un gain plus élevé, conçu par n'importe quelle méthode classique, ayant un ensemble robuste positif invariant associé à l'intérieur des contraintes. Une simple fonction de Lyapunov est utilisée afin d'apporter la preuve de la stabilité en boucle fermée.