Universitat de Girona

COMPUTATIONALLY RELIABLE APPROACHES OF CONTRACTIVE MPC FOR DISCRETE-TIME SYSTEMS

Jian WAN

ISBN: 978-84-690-8985-9
Dipòsit legal: GI-1380-2007

Computationally Reliable Approaches of Contractive MPC for Discrete-time Systems

Jian Wan

Department of Electronics, Computer Science and Automatic Control

University of Girona

A dissertation submitted in partial fulfilment

of the requirements for the degree of

Doctor of Philosophy

Universitat de Girona
Departament d'Electrònica, Informàtica i Automàtica

Computationally Reliable Approaches of Contractive MPC for Discrete-time Systems

by

Jian Wan

Advisors

Dr. Josep Vehí and Dr. Ningsu Luo

Doctoral Thesis
Girona, Spain, May 2007

To my parents, for their endless love, support and encouragement …

Acknowledgements

It is an exciting moment that finally I could be ready for presenting the thesis. I am deeply grateful for various help and generous support received during this tough process. I would like to take this opportunity to express my sincere gratitude to:

First of all, I would like to thank my supervisors, Dr. Josep Vehí and Dr. Ningsu Luo, for their academic and financial support during the whole period of my study in Girona. They have guided me with great efforts in all research issues. Their confidence on me gave me sufficient freedom and continuous encouragement to explore what I am really interested in.

I also owe deep appreciation to Dr. J. M. Bravo and Dr. T. Alamo from University of Huelva (Spain) and University of Seville (Spain), respectively. The work on studying specific nonlinear discrete-time systems with affine state part and affine control part was initialized during my external stay with them in Huelva and the influence of the stay is profound for broadening my thesis as well as my knowledge.

It is my honor to have the chance to attend the course on interval analysis by Prof. Luc Jaulin from ENSIETA (France) in Terrassa, Spain. The influence of his co-authored book called applied interval analysis on my thesis is also profound. Additional thankfulness is also owed to him as well as Prof. E. F. Camacho from University of Seville and Prof. José Rodellar from Technical University of Catalonia (Spain) for their reviews of my initial report for the Diploma of Advanced Studies (DEA). Special appreciations are also owed to Prof. E. F. Camacho and Dr. E. C. Kerrigan from Imperial College (UK) for their valuable review and expertise on the draft of my thesis.

Daily discussions and communications with the members of Modal Intervals and Control Engineering Laboratory (MICE Lab) have also contributed a lot to my study. Specially, I would like to thank Prof. M. A. Sainz and Dra. Remei Calm Puig for teaching me modal intervals and Mr. Christian G. Quintero for the cooperation on predictive motion control of a MiroSot robot. Sincere gratitude is also owed to other members and friends (Dr. Armengol, Dr. Rodolfo, Dr. Luis, Dr. Pau, Dr. Antonio, Inès, Raül, Gabriela, Oscar, Jorge, Esteban, Mauricio, Sandra, Maira and etc.) along with the secretaries of the school (Anna, Marta, Montse and etc.) for helping me adapt the fresh life of Spain.

Furthermore, I am also grateful for the Ph.D scholarship of the International Graduate School of Catalonia (IGSOC) granted by the government of Catalonia. Additional gratitude is also owed to INTerval LABoratory (INTLAB) by Dr. Siegfried M. Rump, Invariant Set Toolbox (IST) by Dr. E. C. Kerrigan and Multi-Parametric Toolbox (MPT) by M. Kvasnica, P. Grieder and M. Baotic. These uncommercial MATLAB toolboxes have been widely used in the simulations of the thesis and thus many extra efforts have been avoided.

Finally, I would like to express my inherent gratitude to my former supervisors in Northwestern Polytechnical University of China — Prof. Demin Xu and Prof. Yuyao He. They introduced me to the area of control engineering with great patience and constant encouragement.

Jian Wan

University of Girona, Spain
May, 2007

Abstract

Computationally Reliable Approaches of Contractive MPC for Discrete-time Systems[a]

Jian Wan

Ph.D Program of University of Girona in Information Technology

Supervised by

Dr. Josep Vehí and Dr. Ningsu Luo

The essence of **MPC** is to avoid solving the **Hamilton-Jacobi-Bellman** equation by repetitively solving an open-loop optimal control problem instead, which offers the significant ability to treat input and state constraints explicitly at each step. Additional **contractive constraint** is usually needed to be incorporated into the open-loop optimization for guaranteeing the stability of the closed-loop system. The resulting online **constrained optimization** must be fulfilled within the time constraint imposed by the sampling time of an application. Thus computational **reliability** and **efficiency** are two critical issues in applying MPC, especially in applying nonlinear MPC, where normally complex **nonlinear programming** problems are concerned. The thesis aims to explore computationally reliable and efficient approaches of **contractive MPC** for discrete-time systems. Two types of contractive MPC have been studied: MPC with compulsory contractive constraint and MPC with a contractive sequence of **controllable sets**. Techniques based on **convex optimization** and **interval analysis** are applied to deal with linear and nonlinear contractive MPC, respectively. Classical interval analysis is extended to **zonotopes** in **geometry** for designing a terminal **control invariant set** in the dual-mode approach of MPC. It is also extended to **modal intervals** in **modality** for computing robust controllable sets with a clear **semantic** interpretation. The tools of convex optimization and interval analysis have been combined further to improve the efficiency of contractive MPC for various kinds of constrained nonlinear uncertain discrete-time systems. Finally, the addressed two types of contractive MPC have been applied to control a Micro Robot World Cup Soccer Tournament (**MiroSot**) robot and a Continuous Stirred-Tank Reactor (**CSTR**), respectively.

[a]The LaTeX 2_ε style of this dissertation was derived from a standard Cambridge University PhD/MPhil thesis LaTeX template written by Harish Bhanderi.

Vita

Jian Wan was born in Dawu County, Hubei Province, China. He obtained his Bachelor of Engineering on Welding from the College of Materials Science and Engineering in July, 1997, Master of Engineering on Automation from the College of Marine Engineering in April, 2000, both from Northwestern Polytechnical University, Xi'an, China. He was once a research student in Robotics Research Center, Nanyang Technological University, Singapore from July, 2001 to January, 2003. He has been enrolled as a Ph.D student in the Department of Electronics, Computer Science and Automatic Control, University of Girona, Catalonia, Spain ever since February, 2003.

Nomenclature

x	A real scalar
$\mathbf{x}$	A real vector
A	A real matrix
$[a, b]$	A real interval
$[a, b]^*$	A real modal interval
$\mathbf{X}$	A real interval vector
$\mathbf{X}^*$	A real modal interval vector
$f(\mathbf{x})$	A real scalar function
$\mathbf{f}(\mathbf{x})$	A real function vector
$f(\mathbf{X})$	A real interval function
$f^*(\mathbf{X}^*)$	A real modal interval function
$\mathcal{Z}$	A zonotope
$\mathcal{P}$	A polytope
$\mathcal{P}^C$	The complement of the polytope $\mathcal{P}$
$\mathbf{B}^m$	The m-dimensional unitary box
$\emptyset$	The empty set
$\mathbb{R}$	The set of reals
$\mathbb{R}^n$	The set of n-dimensional real vectors
$\mathbb{I}(\mathbb{R})$	The set of real intervals
$\mathbb{I}(\mathbb{R}^n)$	The set of n-dimensional real interval vectors
$\mathbb{I}^*(\mathbb{R})$	The set of real modal intervals
$\mathbb{I}^*(\mathbb{R}^n)$	The set of n-dimensional real modal interval vectors
Ω	The robust control invariant set
$\mathbb{T}$	The terminal set in the dual-mode approach of MPC
$\mathbb{X}$	The admissible state domain
$\mathbb{U}$	The admissible control domain
ε	The bound of error tolerance
$\mathbf{x}^T$	The transpose of the real vector $\mathbf{x}$
$\|\mathbf{x}\|_p$	The p-norm of the vector $\mathbf{x}$, $p = 1, 2, \cdots, \infty$
$\mathcal{K}_i(\mathbb{X}, \mathbb{T})$	The i-step controllable set
$\tilde{\mathcal{K}}_i(\mathbb{X}, \mathbb{T})$	The i-step robust controllable set
$\tilde{\mathcal{K}}_i^q(\mathbb{X}, \mathbb{T})$	The quasi i-step robust controllable sets
$\mathbf{x}(k+i\|k)$	The state value at time instant $k+i$ predicted at time instant k
$\mathbf{u}(k+i\|k)$	The control input at time instant $k+i$ calculated at time instant k

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The essence of MPC is to avoid solving the Hamilton-Jacobi-Bellman equation by repetitively solving an open-loop optimal control problem instead (Rao, 2000), which offers the significant ability to treat input and state constraints explicitly at each step. Additional contractive constraint is usually needed to be incorporated into the open-loop optimization for guaranteeing the stability of the closed-loop system (de Oliveira and Morari, 2000; Mayne et al., 2000). The resulting online constrained optimization must be fulfilled within the time constraint imposed by the sampling time of an application (Cannon, 2004). Thus computational reliability and efficiency are two critical issues in applying MPC, especially in applying nonlinear MPC, where normally complex nonlinear programming problems are concerned. The thesis aims to explore computationally reliable and efficient approaches of contractive MPC for discrete-time systems. The interests of studying contractive MPC lie in its characteristics of reduced control horizons and guaranteed stability, which are specially beneficial for real-time applications. The emphasis of the thesis is on computationally reliable and efficient approaches of contractive MPC for constrained nonlinear discrete-time systems, where still exists many open problems and challenges. Numerical tools for linear systems such as convex optimization and numerical tools for nonlinear systems such as interval analysis are to be used and integrated to some degree for dealing with various issues underlying contractive MPC.

1.2 State of the Art

A standard problem in control is to design a feedback control law that minimizes an objective over an infinite horizon. The optimal solution of this problem can be obtained in principle by solving the corresponding Hamilton-Jacobi-Bellman (HJB) equation (Rao, 2000). This is often a difficult task. One exception is when the system is linear, the objective is quadratic and there are no hard constraints on control inputs or state variables. When either of these conditions is violated, general procedures for solving the HJB equation do not exist (Rao, 2000). The essence of MPC is to avoid solving such tough HJB equations by repetitively solving an open-loop optimal control problem instead, which also offers the ability to treat state and control constraints explicitly in every open-loop optimization. The control horizons of MPC are often selected to be finite for rendering such open-loop optimizations tractable. So the selection of finite control horizons in MPC is a compromise between optimality and tractability of constrained control problems. It is pertinent to say that the selection of finite control horizons in MPC is more to satisfy the requirement of feasibility and stability than to guarantee optimality as all MPC schemes with finite control horizons are identically not optimal in a strict meaning of optimality for closed-loop systems. However, for some control systems such as the examples discussed in (Fernando, 2001; García-Gabín et al., 2002), it is impossible to arrive to the steady state in finite steps and thus even large control horizons cannot qualify terminal equality constraints imposed for guaranteeing the stability of closed-loop systems. The meaning of selecting large control horizons is therefore obscure for such kind of occasions. Moreover, large control horizons render the resulting minimization tasks for MPC or minimax optimization tasks for robust MPC extremely difficult and time-consuming, which is unfavorable for real-time applications.

In order to avoid high-dimensional optimizations associated with MPC configurations of large control horizons, some structurally efficient control schemes were proposed to reduce control horizons as well as computational burdens of corresponding optimizations underlying MPC in the literature (Cannon, 2004). Martin Sanchez and Rodellar used a step-control signal along the control horizon and thus the number of unknown control inputs was reduced to be a single one within the finite control horizon (Sanchez and Rodellar, 1996). However, such an approach is only effective to stable systems.

More efforts were made to enlarge the terminal set of MPC and thus a shorter control horizon can drive the system to such an enlarged terminal set as well (Chen et al., 2001; Limon et al., 2005a). Another research direction was to connect MPC strategies with existing control methods for shortening control horizons of MPC. In (Soroush and Kravaris, 1995), a shortest horizon nonlinear MPC was proposed and it turned out to be an input-output linearizing (geometric) controller. In (Zhang et al., 2004; Zhou et al., 2000, 2001), MPC was combined with Variable Structure Control (VSC) through extending terminal sets in MPC to sliding manifolds in VSC and thus control horizons can be selected to be relatively short. In fact, many dual-mode approaches of MPC can be cast in the framework of VSC just as in (Fernando, 2001; García-Gabín and Camacho, 2003; García-Gabín et al., 2005) because most control processes in MPC can also be divided into two phases: the reaching mode and the terminal mode, which correspond to the reaching mode and the sliding mode in VSC. However, a disadvantage of using the sliding manifold of VSC as the terminal set in MPC is the so-called chattering phenomenon, which is unfavorable for real-time applications.

A practical approach for reducing control horizons is to use contractive MPC, where additional compulsory or existing contractive constraint is imposed on open-loop constrained optimizations to guarantee the stability of closed-loop systems (de Oliveira and Morari, 2000; Limon et al., 2003; Yang and Polak, 1993a,b). Contractive MPC can be further classified as MPC with compulsory contractive constraint and MPC with a contractive sequence of controllable sets. For MPC with compulsory contractive constraint, feasibility tests are needed to find a feasible control horizon for satisfying the imposed compulsory contractive constraint; for MPC with a contractive sequence of controllable sets, one-step controllable sets are needed to be computed for generating the feasible contractive constraint in advance. The computations of controllable sets is quite straightforward for discrete-time linear or piecewise-affine systems with polytopes as their terminal sets, where the set computations can be performed efficiently through polyhedral algebra, linear programming and computational geometry software (Kerrigan, 2000; Rakovic et al., 2003). However, the computation of robust controllable sets for general constrained nonlinear uncertain discrete-time systems is not straightforward and efficient. In (Bravo et al., 2005), a branch-and-bound algorithm based on interval arithmetic was introduced to compute the inner approximations of control invariant sets for constrained nonlinear systems with a given bound of error tolerance. This algorithm was reported to be extended in (Limon et al., 2003) to compute robust controllable sets of constrained nonlinear systems with additive uncertainty, where the computed robust controllable sets were utilized as a contractive sequence of sets to formulate a robust MPC scheme. However, the modality of the interval vector for the admissible state space and the modality of the interval vector for the admissible control space are different: for a subbox of the admissible state space, it belongs to a controllable set if and only if for all states within the subbox there always exists an admissible control that can drive them to the selected terminal set; for a subbox of the admissible control space, it is a feasible control domain if and only if there exists one value within the subbox which can drive the concerned subbox of the state space to the selected terminal set. The bisection and the selection of the admissible state space and the admissible control space were mixed in their approaches, where extra treatments of the subboxes of the admissible domain, which was the combination of the admissible state space and the admissible control space, were needed after each bisection and selection of a subbox of the admissible state space (Bravo et al., 2005). The extra treatments of the subboxes of the admissible domain were time-consuming since the members of the set were numerous and the action of bisection and selection was frequent. Furthermore, there was no semantic interpretation for the discard of the admissible state space: if the action for a subbox of the admissible domain was to discard, this did not mean that the subbox of the admissible state space could not be a part of the one-step controllable set because it might be controllable to the selected terminal set for other parts of the admissible control space. Systematic and efficient approaches for computing robust controllable sets of general constrained nonlinear uncertain discrete-time systems with a clear semantic interpretation are needed to be explored further.

In order to compute one-step controllable sets and implement the dual-mode approach of contractive MPC, a terminal control invariant set along with a local stabilizing feedback control law is needed to be designed in advance. The optimality of the designed terminal control invariant set can be judged according to its volume, where a larger control invariant set is preferred for a shorter control horizon and an easier stabilization (Cannon et al., 2003; Limon et al., 2005a). The analytical design of the maximal control invariant set along with a local stabilizing feedback control law for a constrained nonlinear discrete-time system is quite challenging and usually the obtained control invariant set is restricted to be an ellipsoid or a low-complexity polytope (Cannon et al., 2003; Chen and Allgower, 1998; Kothare et al., 1996; Magni et al., 2001; Michalska and Mayne, 1993). Approaches based on linear dynamic approximation together with Lipschitz bounds on the error of approximation were discussed in (Chen and Allgower, 1998; Michalska and Mayne, 1993) to obtain a terminal control invariant ellipsoid for constrained nonlinear systems. Another approach to obtain a terminal control invariant ellipsoid was based on a linear difference inclusion of the original nonlinear system (Kothare et al., 1996). In (Magni et al., 2001), the local linear feedback control law was designed according to the LQ method for the linearized system of the original nonlinear system and the associated terminal control invariant ellipsoid was obtained by an optimization. In (Cannon et al., 2003), a control invariant polytope was also designed in an optimal way for an input-affine nonlinear system, where the advantage of a control invariant polytope over a control invariant ellipsoid was demonstrated. The design of a control invariant low-complexity polytope with respect to a local feedback linearizing control law for input-affine nonlinear systems was further proposed in (Bacic et al., 2005), where the designed invariant set along with the local feedback linearizing control law was used in the terminal control of the dual-mode approach of MPC. In fact, polytopes are usually the natural expression of physical constraints on state and control variables leading to a more flexible and pertinent description of corresponding control invariant sets (Blanchini, 1999). However, the design of a relatively complex control invariant polytope, which is more likely to have a bigger volume, for a general constrained nonlinear discrete-time system is still an open problem. Novel analytical or numerical methods are needed to be explored further for the design of a terminal control invariant polytope along with a local stabilizing feedback control law for general constrained nonlinear discrete-time systems.

1.3 Outline

The thesis aims to confront some open problems addressed in the above section and explore computationally reliable and efficient approaches of contractive MPC for discrete-time systems. Concretely, interval analysis and its extensions are introduced to compute terminal control invariant sets and controllable sets offline for general constrained nonlinear discrete-time systems in a reliable way; MPC with compulsory contractive constraint or a contractive sequence of controllable sets is adopted for implementing MPC with a shorter control horizon, which renders the resulting open-loop optimizations more tractable. Various numerical tools such as interval analysis, polytope geometry and zontope geometry usually used separately in the literature are combined further in the same framework of convex sets to improve the efficiency of their applications in contractive MPC. The thesis is organized as follows:

▷ **Chapter 1** addresses the motivation of the thesis, the state of the art for the corresponding research issues and the outline of the thesis.

▷ **Chapter 2** introduces the basic principle of MPC, the classification of general MPC based on the system models considered, the classification of contractive MPC based on the imposed contractive constraints.

▷ **Chapter 3** addresses linear contractive MPC via convex optimization, where both linear contractive MPC with compulsory contractive constraint and linear contractive MPC with a contractive sequence of controllable sets are studied. A sliding domain along with the equivalent control deduced from variable structure control is designed in advance as the terminal control invariant set for the dual-mode approach of linear contractive MPC to avoid the chattering phenomenon and convex optimization methods such as linear programming are applied to obtain feasible control horizons and one-step control inputs needed in implementing linear contractive MPC, respectively.

▷ **Chapter 4** addresses nonlinear contractive MPC via classical interval analysis, where zonotope geometry is also introduced as an extension of interval analysis in geometry. The solver of set inversion via interval analysis is extended to set inversion via zonotope geometry for demonstrating geometrically whether a given zonotope is control invariant or no under the related local stabilizing feedback control law. The solver of global optimization for set inversion via zonotope geometry is also proposed to obtain an optimal control invariant zonotope with the maximal volume. Both nonlinear contractive MPC with compulsory contractive constraint and nonlinear contractive MPC with a contractive sequence of controllable sets are studied, where the solver of set inversion via interval analysis is applied to find feasible control horizons and compute controllable sets while the solver of global optimization via interval analysis is applied to find one-step control inputs for nonlinear contractive MPC.

▷ **Chapter 5** addresses nonlinear robust contractive MPC via modal interval analysis, where modal interval analysis is introduced as an extension of classical interval analysis in modality, inclusion, semantics and rational. The solver of 1-dimensional quantified set inversion in modal interval analysis is generalized to multi-dimensional cases and it is applied to compute robust controllable sets of constrained nonlinear uncertain discrete-time systems with a clear semantic interpretation. An interval-based solver of constrained minimax optimization is also proposed to compute one-step control inputs for nonlinear robust contractive MPC with a contractive sequence of robust controllable sets.

▷ **Chapter 6** addresses nonlinear robust contractive MPC via hybrid numerical tools, where classical interval analysis, polytope geometry and zonotope geometry are combined further to improve the efficiency of former proposed approaches for nonlinear robust contractive MPC. Concretely, robust controllable sets obtained as a union of interval vectors or boxes are approximated innerly by polytopes and contractive MPC is implemented by using polytopic robust controllable sets. The structures of nonlinear discrete-time systems are also explored to simplify the corresponding computations of polytopic robust controllable sets and the concept of quasi multi-step robust controllable sets is also proposed for the computation of an approximation of multi-step robust controllable sets directly for a specific kind of constrained nonlinear uncertain discrete-time systems with affine control part.

▷ **Chapter 7** addresses the application of the addressed two types of contractive MPC to the control of a Micro Robot World Cup Soccer Tournament (MiroSot) robot and a Continuous Stirred-Tank Reactor (CSTR), respectively.

▷ **Chapter 8** draws the conclusions of the research work of the thesis and proposes some potential directions for the future work.

Chapter 2

Model Predictive Control

Model Predictive Control (MPC) has a quite unusual history, with separate strands in system theory (where it is generally referred to as Receding Horizon Control), generalized predictive control (where the initial intention was the improvement of adaptive control) and in process control where its almost unique ability to handle hard constraints led to its wide-scale adoption in process industry (Mayne, 1999). More than fifteen years after MPC appeared in process industry as an effective means to deal with multi-variable constrained control problems, a theoretical basis for this technique started to emerge (Camacho and Bordóns, 2005; Morari and Lee, 1999). So it might be justifiably argued that the importance of MPC derives primarily from its industrial success, a fact that delineates it from other design procedures that are, in general, theoretically motivated.

After several years of mutual efforts from both theoretical researchers and industrial practitioners, basic issues in MPC such as feasibility of the on-line optimization, stability and corresponding closed-loop performance are largely understood for systems described by linear models. Many progresses have also been made on these issues for nonlinear models. But many questions still remain in practical applications, including the reliability and the efficiency of on-line optimization configurations or schemes. Moreover, most systems in practice are full of various uncertainties such as modeling errors and exogenous disturbances. Novel MPC analysis and synthesis methods for such nonlinear and uncertain systems are needed so as to guarantee robust stability and tighter performance specifications of nowadays applications.

2.1 The Essence of MPC

A standard problem in control is to design a feedback control law that minimizes an objective over an infinite horizon. The optimal solution of this problem can be obtained in principle by solving the Hamilton-Jacobi-Bellman (HJB) equation (the dynamic program that arises in control). This is often a difficult task. One exception is when the system is linear, the objectives are quadratic and there are no hard constraints on the inputs or states (Rao, 2000). In this case, the optimal cost function can be parameterized as a symmetric matrix and the feedback control law reduces to be a linear quadratic regulator. When either of these conditions is violated, general procedures for solving the HJB equation do not exist (Rao, 2000). The term **Model Predictive Control** refers to a class of computer control algorithms that utilize an explicit model to predict the future response of a plant. At each control interval, an MPC algorithm determines a sequence of manipulated variable adjustments that optimize future plant behavior subject to input and state constraints. The first input in the optimal sequence is then sent into the plant and the entire optimization is repeated at subsequent control intervals. Thus the essence of MPC is to avoid solving the HJB equation by repetitively solving an open-loop optimal control problem instead, which also offers the ability to treat input and state constraints explicitly at every optimization step.

Although there are various kinds of MPC configurations, the basic principle underlying them is quite similar. In the following, the algorithm of linear MPC is adopted as an example for demonstrating the basic principle (Bemporad and Morari, 1999). Let the model of the plant to be controlled be described by the linear discrete-time difference equations in the state space:

$$\begin{cases} \mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k), k = 0, 1, \cdots \\ \mathbf{y}(k) = C\mathbf{x}(k) \end{cases} \tag{2.1}$$

where $\mathbf{x}(k) \in \mathbb{R}^n, \mathbf{u}(k) \in \mathbb{R}^m, \mathbf{y}(k) \in \mathbb{R}^p$ denote the state, control input and output, respectively. Let $\mathbf{x}(k+1|k)$ denote the value of $\mathbf{x}$ at time instant $k+1$ predicted at current time instant k. A receding horizon implementation is typically based on the solution of the following open-loop optimization problem:

$$J(\mathbf{x}(k), \{\mathbf{u}(k+i|k)\}_{i=0}^{M-1}) = \min_{\{\mathbf{u}(k+i|k)\}_{i=0}^{M-1}} \{\mathbf{x}^T(k+N|k)P_0\mathbf{x}(k+N|k) + \sum_{i=1}^{N-1} \mathbf{x}^T(k+i|k)Q\mathbf{x}(k+i|k) + \sum_{i=0}^{M-1} \mathbf{u}^T(k+i|k)R\mathbf{u}(k+i|k)\} \tag{2.2}$$

subject to

$$\begin{cases} F_1\mathbf{u}(k+i|k) \leq G_1 \\ E_2\mathbf{x}(k+i|k) + F_2\mathbf{u}(k+i|k) \leq G_2 \end{cases} \tag{2.3}$$

and corresponding stability constraints, which are to guarantee closed-loop stability through explicitly requiring that the state $\mathbf{x}$ shrinks in certain norm or enters into a terminal region at the end of the prediction horizon (Mayne et al., 2000). P_0, Q, R are weight coefficients of the terminal state of the prediction horizon, intermediate states and control in the cost function, respectively; N denotes the length of the prediction horizon or output horizon, and M denotes the length of the control horizon or input horizon ($M \leq N$); F_1, G_1, E_2, F_2, G_2 are specified coefficients of linear constraints on control and state variables. The basic MPC law can thus be described in the following algorithm:

1). Get the current state $\mathbf{x}(k)$;
2). Solve the optimization problem to get $\mathbf{u}^{Optimal}(k+i|k)\}_{i=0}^{M-1}$;
3). Apply only $\mathbf{u}^{Optimal}(k) = \mathbf{u}^{Optimal}(k|k)$ to the system at time instant k;
4). $k+1 \rightarrow k$, return to 1.

2.2 Nonlinear MPC

Important issues such as feasibility of on-line optimization, closed-loop stability and corresponding closed-loop performance for linear MPC have been studied extensively and they are well addressed in (Mayne et al., 2000; Morari and Lee, 1999; Rawlings, 2000). However, most systems in practice are inherently nonlinear. Furthermore, higher product quality specifications, increasing productivity demands and tighter environmental regulations in process industry often require to operate systems along a wider range of dynamics and (or) closer to the boundary of the admissible operating regions. In such scenarios, linear models are usually inadequate to describe the process dynamics and nonlinear models have to be used (Findeisen and Allgower, 2002).

The basic principle of nonlinear MPC is similar to linear MPC except that the system model considered is nonlinear:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)), \tag{2.4}$$

where $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$ is a nonlinear function vector. Then the optimization problem encountered in nonlinear MPC becomes, in general, a non-convex nonlinear optimization problem. This means that it is much more difficult to find a solution for the constrained optimization and even a feasible solution is found, it can hardly be guaranteed to be globally optimal. Furthermore, it was also recognized early in the development of optimal control theory that, no matter how the control problem is solved, optimality does not necessarily imply closed-loop stability, even when the model represents the true plant perfectly. In the following subsections, current measures for tackling these two tough issues — stability and optimization in nonlinear MPC are reviewed briefly.

2.2.1 Stability of Nonlinear MPC

In theory, the most straightforward way to modify nonlinear MPC algorithms to achieve nominal stability involves setting the prediction horizon and the control horizon to be infinite (Kouvaritakis and Cannon, 2001). With standard technical assumptions, it follows directly from **Bellman Principle of Optimality** that the predicted open-loop input and state trajectories match those in the closed loop. This implies nominal stability because any feasible trajectory terminates at the desired steady state.

From a practical point of view, however, it is simply impossible to solve the nonlinear MPC optimization with infinite horizons for a realistic problem. The focus of recent research efforts has been moved to obtain a computationally tractable approximation of the infinite horizon problem that still retains desirable closed-loop properties. An early solution involves adding a terminal state constraint to the nonlinear MPC:

$$\mathbf{x}(k+N|k) = \mathbf{x}_s, \tag{2.5}$$

where $\mathbf{x}_s$ is the targeted steady state. With such a constraint enforced, the cost function for the controller becomes a Lyapunov function for the closed-loop system, leading to nominal stability (Keerthi and Gilbert, 1988). Unfortunately, such a constraint may be quite difficult to be satisfied in real time: exact satisfaction requires an infinite number of iterations for the numerical solution code. This motivated Michalska and Mayne (Michalska and Mayne, 1993) to seek a less-stringent stability requirement. Their main idea was to define a neighborhood $\mathbb{T}$ around the desired steady state $\mathbf{x}_s$ within which the system can be steered to by a constant linear stabilizing feedback controller. They added to the nonlinear MPC algorithm a constraint of the form:

$$\mathbf{x}(k+N|k) \in \mathbb{T}. \tag{2.6}$$

If the current state $\mathbf{x}(k)$ lies outside the terminal region $\mathbb{T}$, then the nonlinear MPC approach is implemented. Once inside the terminal region $\mathbb{T}$, the control switches to the previously determined constant linear stabilizing feedback controller. Michalska and Mayne described this strategy as a dual-mode approach of MPC. Another idea of guaranteeing stability is to incorporate a contractive constraint into the usual formulation that forces the actual and not only the predicted state to contract at discrete intervals in the future (de Oliveira and Morari, 2000; Yang and Polak, 1993a,b). From this requirement, a Lyapunov function based on the contractive constraint can be constructed easily and therefore stability can also be established.

Most recent research activities are focused on quasi-infinite horizon nonlinear MPC algorithms, first introduced by Chen and Allgower (Chen and Allgower, 1998). The basic idea motivating this method is similar to that of the dual-mode approach. The terminal constraint is imposed so that, at the end of the finite horizon, one can imagine that a linear stabilizing feedback controller takes over. An upper bound for the objective function can then be computed and this term is added as a terminal penalty to the original finite horizon objective. This modified objective is then used regardless of where the current state lies, so that it is not necessary to switch from one controller to another.

2.2.2 Optimization of Nonlinear MPC

Computational complexity of online constrained optimization is another essential issue in nonlinear MPC, especially for fast sampling applications, high-dimensional systems and control problems that demand the use of large prediction horizons. The performance costs and constraints of nonlinear MPC are in general non-convex functions of the predicted control inputs, and their optimization calls for the use of numerical techniques, whose demanding nature may exceed the time available for online computation. A solution to the excessive computation burden is to look for sub-optimal MPC, where global and exact solutions of non-convex and nonlinear optimization problems are not needed and feasibility can imply stability instead (Scokaert et al., 1999). Another sub-optimal approaches for nonlinear MPC include feedback linearization approximations to optimal control Lyapunov functions and interpolations (Bacic et al., 2003; Bloemen et al., 2002). Global optimization via interval analysis can also be applied to solve low-dimensional nonlinear optimizations encountered in nonlinear MPC in a guaranteed numerical way (Bravo et al., 2000; Hansen, 1992).

2.3 Robust MPC

The basic MPC algorithms described in the previous sections assume that the plant to be controlled and the model used for prediction and optimization are the same, and no unmeasured disturbance is acting on the system. The control performance of MPC is thus highly dependent on the accuracy of the open-loop predictions, which in turn depends on the accuracy of the system models. The difference between the plant and the model is known as plant-model mismatch, which can cause the control performance to be sluggish, overly conservative, or in the worst-case scenario, unstable (Wang, 2002). Plant-model mismatches occur frequently due to modeling error, state estimation error and unknown exogenous disturbances. Although the output feedback feature in MPC can reduce the discrepancy between the actual and forecasted behavior of the systems, MPC is not designed to explicitly handle plant-model mismatches. This is due to the lack of explicit functional description of the control law for MPC, which is incurred by the online optimization strategy at every control step. Thus the robust analysis and synthesis have proven to be a fairly complicated and tough issue, even for linear MPC.

2.3.1 Linear Robust MPC

In order to address robustness issues in linear MPC, the following assumptions are made on the basis of usual linear MPC formulation: the true plant $\Sigma_0 = (A, B, H, C, K) \in$

$\mathbb{S}$, where $\mathbb{S}$ is a given family of linear time-invariant systems and (or) an unmeasured noise $\mathbf{w}(k)$ enters the system, i.e.,

$$\begin{cases} \mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) + H\mathbf{w}(k) \\ \mathbf{y}(k+1) = C\mathbf{x}(k) + K\mathbf{w}(k), \end{cases} \tag{2.7}$$

where $\mathbf{w}(k) \in \mathbb{W}$ and $\mathbb{W}$ is a given set (usually a polytope). Stability, constraint fulfillment and control performance of MPC are referred to be robust if the respective property is guaranteed for all possible $\Sigma_0 \in \mathbb{S}, \mathbf{w}(k) \in \mathbb{W}$ (Bemporad and Morari, 1999).

As a part of robust MPC, it is necessary to arrive at an appropriate description of the uncertainty, i.e. the sets $\mathbb{S}$ and $\mathbb{W}$. This is difficult because there is very little experience and no systematic procedures are available. Several uncertainty sets $\mathbb{S}, \mathbb{W}$ have been proposed in the literature in the context of MPC (Bemporad and Morari, 1999). Some of them are uncertainty on the impulse-response or step-response coefficients, structured feedback uncertainty, multi-plant, polytopic uncertainty and bounded input disturbances. It is also possible to adopt a stochastic uncertainty description instead of a set-based description and develop an MPC algorithm that minimizes the expected value of a cost function (Morari and Lee, 1999).

The minimum closed-loop requirement for robust MPC is robust stability, i.e., stability in the presence of uncertainties. In MPC, various design procedures achieve robust stability in two different ways: indirectly by specifying the performance objective and uncertainty description in such a way that the optimal control computations are to minimize the worst performance over the specified uncertainty range and thus the min-max performance optimizations lead to robust stability (Lu and Arkun, 2000; Scokaert and Mayne, 1998); or directly by enforcing a type of robust contraction constraint such as robustly invariant terminal sets, which guarantees that the state shrink for all plants in the uncertainty set (Zheng, 1995). Except for guaranteeing stability when synthesizing robust MPC laws, it is also necessary to enforce state constraints robustly. Due to the curse of dimensionality for some min-max optimizations (Lee and Yu, 1997), more efficient numerical methods such as Linear Matrix Inequalities (LMIs) (Kothare et al., 1996; Wu, 2001) were introduced to solve the heavy minimax optimization computations underlying robust MPC.

2.3.2 Nonlinear Robust MPC

There are a few results on robustness of constrained MPC for nonlinear systems. In (Michalska and Mayne, 1993), a dual-mode approach of receding horizon controller was proposed and robustness under decaying additive uncertainties was achieved by a proper choice of the terminal region. In (Magni et al., 2001), a robust MPC strategy based on H-∞ cost function was presented. In (Limon et al., 2002), a robust MPC for constrained nonlinear discrete-time system with additive uncertainties was presented. It was based on uncertain evolution sets and interval arithmetic was used for the computation of uncertain evolution sets. Interval arithmetic was also applied in the robust analysis of predictive control in (Vehi et al., 2000). Interval arithmetic has gradually turned out to be an appropriate and promising technique for the formulation and solution of robust control problems in MPC, especially in nonlinear MPC. The interval description of both parametric uncertainty and additive disturbances can be explored further and constrained minimax optimization underlying nonlinear robust MPC can also be fulfilled via interval arithmetic (Jaulin et al., 2001).

2.4 Contractive MPC

As addressed in the introduction, one of the main advantages of contractive MPC is the reduced feasible control horizon, which is beneficial for simplifying the resulting optimization. Contractive MPC is categorized into two groups here, which are to be addressed in Subsection 2.4.1 and 2.4.2, respectively.

2.4.1 MPC with Compulsory Contractive Constraint

MPC with compulsory contractive constraint was studied in (de Oliveira and Morari, 2000; Yang and Polak, 1993a,b), where compulsory contractive constraints were usually in the form of contractive norms of state variables, i.e., the following additional contractive constraint is imposed on the constrained optimization underlying MPC:

$$|\mathbf{x}(k+N|k)|_p \leq \alpha |\mathbf{x}(k)|_p, \tag{2.8}$$

where N is the prediction horizon; $|\cdot|_p$ is the p-norm of vectors; and $\alpha \in (0, 1]$ is the contractive parameter for the norm. Since the control target is aimed to contract the distance between the current state and the target state rather than reach to the target state directly, the feasible control horizon used in MPC with compulsory contractive constraint can be relatively short. The control horizon can be usually selected to be equal to the number of state variables. However, additional feasibility tests are still needed to be fulfilled for guaranteeing the feasibility of the resulting constrained optimization with the selected control horizon.

2.4.2 MPC with A Contractive Sequence of Controllable Sets

Consider the general constrained nonlinear uncertain discrete-time system:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{w}(k), \mathbf{u}(k)), k = 0, \cdots, \tag{2.9}$$

where $\mathbf{x}(k) \in \mathbb{X} \subset \mathbb{R}^n$ is the system state; $\mathbf{w}(k) \in \mathbb{W} \subset \mathbb{R}^l$ is the uncertain parameters and (or) additive disturbances; and $\mathbf{u}(k) \in \mathbb{U} \subset \mathbb{R}^m$ is the control input. The set $\mathbb{X}$ is compact, while $\mathbb{U}$ and $\mathbb{W}$ are closed. According to the discussion in (Kerrigan, 2000), the definitions of robust control invariant sets and robust controllable sets are as follows:

Definition 2.1 (Robust Control Invariant Set) The set $\Omega \subset \mathbb{R}^n$ is a robust control invariant set for the system $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{w}(k), \mathbf{u}(k))$ if and only if $\exists \mathbf{u}(k) \in \mathbb{U} : \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{w}(k), \mathbf{u}(k)) \in \Omega, \forall \mathbf{x}(k) \in \Omega, \forall \mathbf{w}(k) \in \mathbb{W}$. Specifically, if the uncertainty of the system is not considered, the corresponding definition is narrowed to be control invariant set.

Definition 2.2 (Robust Controllable Set) The i-step robust controllable set $\tilde{\mathcal{K}}_i(\mathbb{X}, \mathbb{T})$ is the largest set of states in $\mathbb{X}$ for which there exists an admissible control sequence $\{\mathbf{u}(k) \in \mathbb{U}\}_0^{i-1}$ such that the terminal set $\mathbb{T} \subset \mathbb{R}^n$ is reached in i steps, while keeping the evolution of the state inside $\mathbb{X}$ for the first $i-1$ steps, for all allowable disturbance sequences, i.e., $\tilde{\mathcal{K}}_i(\mathbb{X}, \mathbb{T}) \triangleq \{\mathbf{x}(0) \in \mathbb{X} | \exists \{\mathbf{u}(k) \in \mathbb{U}\}_0^{i-1} : \{\mathbf{x}(k) \in \mathbb{X}\}_0^{i-1}, \mathbf{x}(i) \in \mathbb{T}, \forall \{\mathbf{w}(k) \in \mathbb{W}\}_0^{i-1}\}$. Specifically, if the uncertainty of the system is not considered, the corresponding definition is narrowed to be controllable set, which is to be denoted by $\mathcal{K}_i(\mathbb{X}, \mathbb{T})$.

The geometric condition for a set Ω to be robust control invariant is (Mayne and Schroeder, 1997):

$$\Omega \subseteq \tilde{\mathcal{K}}_1(\mathbb{X}, \Omega), \tag{2.10}$$

where $\tilde{\mathcal{K}}_1(\mathbb{X}, \Omega)$ can also be denoted as $\tilde{\mathcal{K}}(\mathbb{X}, \Omega)$. The conservative definition of robust controllable sets is adopted here since $\{\mathbf{u}(k) \in \mathbb{U}\}_0^{i-1}$ is an open-loop admissible control sequence rather than an admissible time-varying state feedback control law $\{\mathbf{u}(k) = \mathbf{h}(k)(\mathbf{x}(k)) \in \mathbb{U}\}_0^{i-1}$, which is more difficult to be designed in advance for constrained nonlinear discrete-time systems.

The cost to compute the i-step ($i \geq 2$) robust controllable set $\tilde{\mathcal{K}}_i(\mathbb{X}, \mathbb{T})$ directly is somewhat heavy since it is a multi-dimensional optimization problem. In practice, $\tilde{\mathcal{K}}_i(\mathbb{X}, \mathbb{T})$ can be approximated innerly via the following recursive procedure:

$$\tilde{\mathcal{K}}_{i+1}(\mathbb{X}, \mathbb{T}) = \tilde{\mathcal{K}}(\mathbb{X}, \tilde{\mathcal{K}}_i(\mathbb{X}, \mathbb{T})), \tag{2.11}$$

where $\tilde{\mathcal{K}}_0(\mathbb{X}, \mathbb{T}) = \mathbb{T}$. If the terminal set $\mathbb{T}$ is selected to be a robust control invariant set Ω, then $\tilde{\mathcal{K}}_i(\mathbb{X}, \Omega)$ is also a robust control invariant set with

$$\tilde{\mathcal{K}}_i(\mathbb{X}, \Omega) \subseteq \tilde{\mathcal{K}}_{i+1}(\mathbb{X}, \Omega), \tag{2.12}$$

where $\tilde{\mathcal{K}}_i(\mathbb{X}, \Omega)$ is also referred to as the i-step robust stabilisable set (Kerrigan, 2000). Such a geometric property of robust stabilisable sets can be applied to formulate a stable and robust model predictive control scheme since a feasible control sequence, which guarantees to drive the system from any initial state within $\tilde{\mathcal{K}}_i(\mathbb{X}, \Omega)$ to Ω in i steps, can be obtained basing on the feasible contractive sequence of robust stabilisable sets (Limon et al., 2003). The benefit of MPC with a contractive sequence of controllable sets is that the feasible control horizon is equal to one and thus the resulting constrained open-loop optimizations are usually trivial.

2.5 Summary

This chapter has provided a brief introduction and literature review on linear and nonlinear MPC. Two types of contractive MPC, i.e., contractive MPC with compulsory contractive constraint and contractive MPC with a contractive sequence of controllable sets have been introduced as a specific kind of MPC, which is also the main research objective of the thesis. The definitions of (robust) control invariant set and (robust) controllable set have also been introduced and they are to be widely concerned in the following chapters.

Chapter 3

Linear Contractive MPC via Convex Optimization

As addressed in Chapter 2, the essence of MPC is to avoid solving the Hamilton-Jacobi-Bellman equation by repetitively solving an open-loop optimization problem instead. Then the main task underlying MPC is to formulate a corresponding mathematical optimization problem as simply as possible and solve it as efficiently as possible. This chapter is focused on formulating and solving a specific class of mathematical optimization problems for MPC — convex optimization for linear contractive MPC. Although there are few open problems for linear MPC, the study of linear contractive MPC via convex optimization is still meaningful since it provides the theoretical background for the further study of nonlinear contractive MPC. Moreover, many nonlinear programming problems can be transformed or simplified to be a linear programming problem and thus can be solved by convex optimization as well (Boyd and Vandenberghe, 2004). The terminal set of linear contractive MPC considered in this chapter is extended from sliding manifolds in variable structure control to be a sliding domain and thus the chattering phenomenon usually happened in the sliding mode can be avoided.

3.1 Convex Optimization

A convex optimization problem is one of the form:

$$\begin{cases} \text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq b_i, \quad i = 1, \cdots, m, \end{cases} \tag{3.1}$$

where the functions $f_0, \cdots, f_m : \mathbb{R}^n \rightarrow \mathbb{R}$ are convex, i.e., they satisfy

$$f_i(\alpha\mathbf{x} + \beta\mathbf{y}) \leq \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y}) \tag{3.2}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$.

There are great advantages to recognize or formulate a problem as a convex optimization problem. The most basic advantage is that the problem can then be solved, very reliably and efficiently, using interior-point methods or other special methods for convex optimization. These methods are reliable enough to be embedded in a computer-aided design or analysis tool, or even a real-time reactive or automatic control system (Boyd and Vandenberghe, 2004). There are also theoretical or conceptual advantages of formulating a problem as a convex optimization problem. The associated dual problem, for example, often has an interesting interpretation in terms of the original problem, and sometimes leads to an efficient or distributed method for solving it (Boyd and Vandenberghe, 2004).

3.2 Polytope Geometry

Polytope is a general class of convex sets widely used in convex optimization. Concretely, polytope is a bounded polyhedron $\mathcal{P} \subset \mathbb{R}^n$, which can be described as (Kvasnica et al., 2006):

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n | P\mathbf{x} \leq \mathbf{q}\}, \tag{3.3}$$

where P is a matrix of $m \times n$ and $\mathbf{q}$ is a vector of m. Basic polytope manipulations are to compute the complement of a polytope, the intersection of two polytopes, the set difference of two polytopes and the convex hull of a union of polytopes, whose definitions are as follows:

Complement: The complement of a polytope $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n | P\mathbf{x} \leq \mathbf{q}\}$ relative to $\mathbb{X} \subset \mathbb{R}^n$ is a union of polytopes $\mathcal{P}^C := \cup_{i=1}^m \{\mathbf{x} \in \mathbb{X} | P_i\mathbf{x} > \mathbf{q}_i\}$, where P_i and $\mathbf{q}_i$ are the ith row of P and $\mathbf{q}$, respectively.

Intersection: The intersection of two polytopes $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n | P\mathbf{x} \leq \mathbf{q}\}$ and $\mathcal{Q} = \{\mathbf{x} \in \mathbb{R}^n | R\mathbf{x} \leq \mathbf{v}\}$ is a polytope $\mathcal{P} \cap \mathcal{Q} := \{\mathbf{x} \in \mathbb{R}^n | P\mathbf{x} \leq \mathbf{q}, R\mathbf{x} \leq \mathbf{v}\}$.

Set Difference: The set difference of two polytopes $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n | P\mathbf{x} \leq \mathbf{q}\}$ and $\mathcal{Q} = \{\mathbf{x} \in \mathbb{R}^n | R\mathbf{x} \leq \mathbf{v}\}$ is a union of polytopes $\mathcal{P} \setminus \mathcal{Q} := \mathcal{P} \cap \mathcal{Q}^C$.

Convex Hull: The convex hull of a union of polytopes $\mathcal{P}_i \subset \mathbb{R}^n (i = 1, \cdots, p)$ is a polytope $\mathtt{Hull}(\cup_{i=1}^p \mathcal{P}_i) := \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} = \sum_{i=1}^p \alpha_i \mathbf{x}_i, \mathbf{x}_i \in \mathcal{P}_i, 0 \leq \alpha_i \leq 1, \sum_{i=1}^p \alpha_i = 1\}$.

3.3 Linear MPC with Compulsory Contractive Constraint

This section addresses linear contractive MPC with compulsory contractive constraint, where linear programming is applied to find feasible control horizons as well as corresponding control inputs for linear contractive MPC with compulsory contractive constraint.

3.3.1 Problem Statement

Assume that the constrained linear discrete-time system to be controlled is described by the following time-invariant state-space model:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k), k = 0, 1, \cdots, \tag{3.4}$$

where $\mathbf{x}(k) \in \mathbb{X} \subset \mathbb{R}^n$ is a vector of n state variables and $\mathbb{X}$ is a compact set containing the origin; $\mathbf{u}(k) \in \mathbb{U} \subset \mathbb{R}^m$ is a vector of m control inputs and $\mathbb{U}$ is a compact set containing the origin. A dual-mode approach of MPC is adopted and the terminal control invariant set $\mathbb{T}$ is designed in advance as a polytope $\mathcal{P} = \{\mathbf{x} | P\mathbf{x} \leq \mathbf{1}\}$ along with a local stabilizing feedback control law.

The concerned linear contractive MPC with compulsory contractive constraint is the formulation of the following iterative optimization:

$$J(\mathbf{x}(k), \{\mathbf{u}(k+i|k)\}_{i=0}^{M_f-1}) = \min_{\{\mathbf{u}(k+i|k)\}_{i=0}^{M_f-1}} \xi \tag{3.5}$$

subject to

$$\begin{cases} \mathbf{u}(k+i|k) \in \mathbb{U}, i = 0, \cdots, M_f - 1 \\ \mathbf{x}(k+i|k) \in \mathbb{X}, i = 1, \cdots, M_f \\ P\mathbf{x}(k+M_f|k) \leq \xi \\ P\mathbf{x}(k+M_f|k) < \alpha \cdot P\mathbf{x}(k), \alpha \in (0,1], \end{cases} \tag{3.6}$$

where α is the contractive parameter for the value of $P\mathbf{x}$, which can be interpreted physically as the contractiveness of the distance to the target polytope; M_f is the feasible control horizon for satisfying all the imposed constraints.

3.3.2 Feasible Control Horizon via Linear Programming

The main problem for applying linear contractive MPC with compulsory contractive constraint is to find a feasible control horizon M_f for satisfying all the imposed constraints. This problem can be further formulated as a standard linear programming problem since all imposed constraints are linear and linear programming can be applied to test whether a selected control horizon is feasible or no. If not, a larger control horizon should be used for another feasibility test until a feasible control horizon M_f has been found.

3.3.3 Contractive MPC via Convex Optimization

The optimal control inputs $\{\mathbf{u}(k+i|k)\}_{i=0}^{M_f-1}$ can be found simultaneously during the feasibility test for a feasible control horizon M_f. The obtained control inputs can be used to control the system sequentially when the state is outside the designed terminal control invariant set (de Oliveira and Morari, 2000). Once the state enters the designed terminal control invariant set $\mathbb{T}$, the related local stabilizing feedback control law can be applied instead to drive the state to the origin asymptotically.

3.3.4 Example

Consider the following constrained discrete-time linear system described by the state-space model:

$$\begin{cases} x_1(k+1) = 0.9x_1(k) + 0.1x_2(k) \\ x_2(k+1) = -0.5x_1(k) + 0.2x_2(k) + u(k), \end{cases} \tag{3.7}$$

where $x_1(0) = 3, x_2(0) = 3$ and the control target is to drive the state variables of the system to the origin asymptotically with the imposed control constraint $u \in [-2, 2]$

and the state constraint $|\mathbf{x}|_\infty \in [-8, 8]$. The dual-mode approach of MPC is adopted: the system state is to be driven to a terminal control invariant set at first and then a local stabilizing feedback control law is applied instead to drive the system state to the origin asymptotically.

The terminal control invariant set is designed in advance according to the principle of variable structure control, i.e., a sliding domain $\mathbb{S} = \{\mathbf{x}|s(k) = cx_1(k) + x_2(k) = 0, \quad c \in [1, 2], x_1, x_2 \in [-4, 4]\}$ is selected to be the terminal control invariant set and the corresponding equivalent control $u_{eq}(\mathbf{x})$ within the sliding domain is selected to be the related local stabilizing feedback control law for the terminal control invariant set. The designed control invariant set for the system (3.7) is shown in Fig. 3.1. It is easy to prove that sliding motions on $\mathbb{S}$ are asymptotically stable under the equivalent control u_{eq} on $\mathbb{S}$, which also satisfies the control constraint, i.e., $u_{eq}(\mathbf{x}) \in [-2, 2], \mathbf{x} \in \mathbb{S}$.

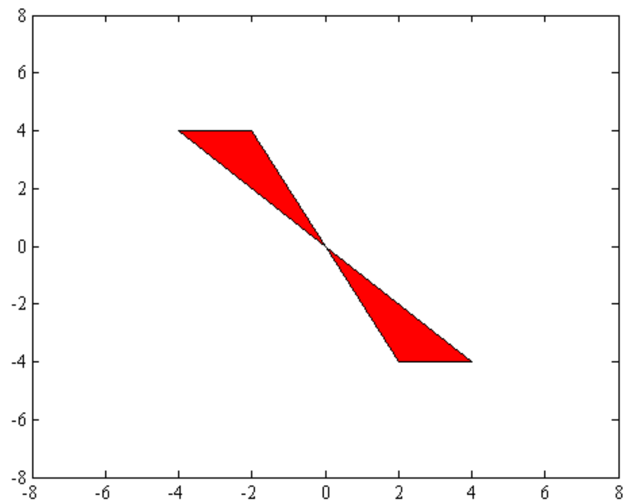


Figure 3.1: A sliding domain is designed as the terminal control invariant set

The feasible control horizon as well as the related feasible control sequence can be obtained efficiently by formulating the linear contractive MPC with compulsory contractive constraint as a linear programming problem, where the convex hull of the designed terminal control invariant set is employed instead to configure the linear programming problem for finding the feasible control horizon as well as the related feasible control sequence. The overall control processes for the dual-mode approach of linear contractive MPC with compulsory contractive constraint starting from the initial state $(6.8, 7.2)$ and $(-7.2, -5.8)$ are shown in Fig. 3.2, respectively. It can be seen that the control strategy is effective for linear contractive MPC with compulsory contractive constraint.

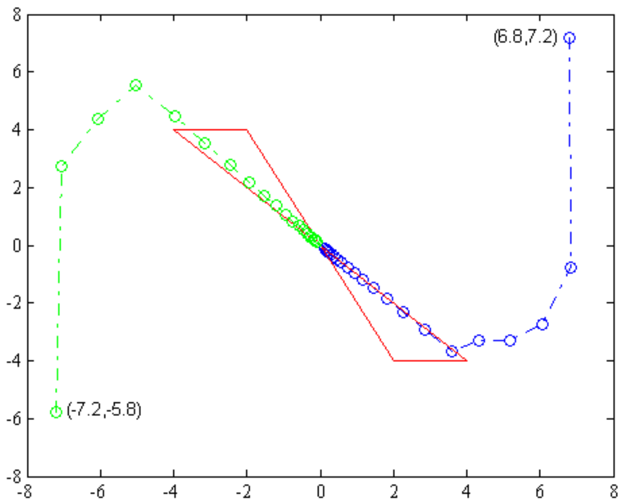


Figure 3.2: Linear contractive MPC with compulsory contractive constraint

3.4 Linear MPC with A Contractive Sequence of Controllable Sets

This section addresses linear contractive MPC with a contractive sequence of controllable sets, where polytope geometry is applied to compute one-step controllable sets offline and convex optimization is applied to compute one-step control inputs for linear contractive MPC with a contractive sequence of controllable sets.

3.4.1 Problem Statement

Assume that the constrained linear discrete-time system to be controlled is described by the following time-invariant state-space model:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k), k = 0, 1, \cdots, \tag{3.8}$$

where $\mathbf{x}(k) \in \mathbb{X} \subset \mathbb{R}^n$ is a vector of n state variables and $\mathbb{X}$ is a compact set containing the origin; $\mathbf{u}(k) \in \mathbb{U} \subset \mathbb{R}^m$ is a vector of m control inputs and $\mathbb{U}$ is a compact set containing the origin. The dual-mode approach of MPC is also adopted and the terminal control invariant set is designed in advance as a polytope $\mathbb{T} = \{\mathbf{x}|T\mathbf{x} \leq \mathbf{1}\}$.

The considered linear contractive MPC with a contractive sequence of controllable sets is the formulation of the following one-step optimization:

$$J(\mathbf{x}(k), \mathbf{u}(k|k)) = \min_{\mathbf{u}(k|k)} \xi \tag{3.9}$$

subject to

$$\begin{cases} \mathbf{u}(k|k) \in \mathbf{U} \\ T\mathbf{x}(k+1|k) \leq \xi \\ \mathbf{x}(k+1|k) \in \mathcal{K}_{i-1}(\mathbb{X}, \mathbb{T}), \end{cases} \tag{3.10}$$

where $\mathbf{x}(k) \in \mathcal{K}_i(\mathbb{X}, \mathbb{T})$; $\mathcal{K}_i(\mathbb{X}, \mathbb{T})$ is the i-step controllable set to the selected terminal set $\mathbb{T}$; and the resulting optimization can be interpreted physically as to minimize the distance to the terminal polytope $\mathbb{T}$.

3.4.2 The Computation of Controllable Sets via Polytope Geometry

The computation of one-step controllable sets for linear or piecewise-affine systems is quite straightforward (Kerrigan, 2000). One-step controllable sets can be computed efficiently via projection or Minkowski summation. **Invariant Set Toolbox** developed by Dr. Eric C. Kerrigan is applied to compute one-step controllable sets needed for implementing linear contractive MPC with a contractive sequence of sets.

3.4.3 One-step Control via Convex Optimization

Once one-step controllable sets have been computed in advance, the feasibility of linear contractive MPC with a contractive sequence of controllable sets can be guaranteed. The one-step control inputs can be obtained via linear programming since the optimization problem in (3.9) is a linear programming problem. Once the system state has entered into the selected terminal control invariant set $\mathbb{T}$, the related local stabilizing feedback control law is applied instead to drive the system state to the origin asymptotically.

3.4.4 Example

Consider the same example to the former section, one-step controllable sets are computed recursively by **Invariant Set Toolbox** (Kerrigan, 2000), which are shown in Fig. 3.3. It is worthy to note that the terminal control invariant set is a union of two polytopes for this specific example and thus the overall one-step controllable set at each recursion is the union of two one-step controllable sets with their respective terminal control invariant polytope. This is different from existing approaches in the literature, where the terminal control invariant set is usually selected to be one polytope.

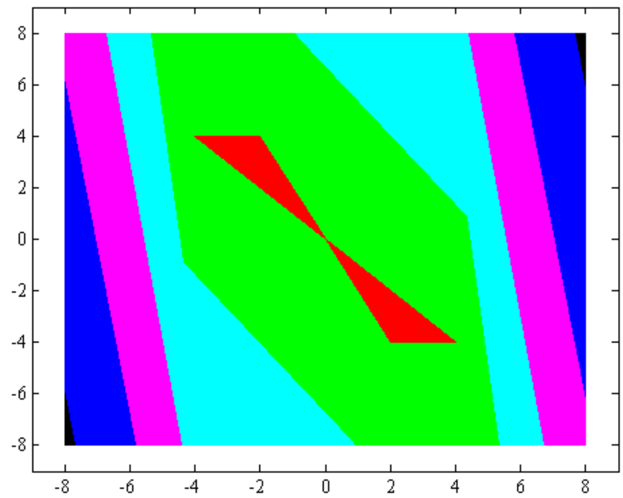


Figure 3.3: The computed one-step controllable sets via polytope geometry

According to the computed one-step controllable sets, linear contractive MPC with a contractive sequence of controllable sets can be formulated and the resulting control processes of the dual-mode approach of MPC for the system (3.7) starting from the initial state $(6.8, 7.2)$ and $(-7.2, -5.8)$ are shown in Fig. 3.4, respectively. It can be seen from Fig. 3.4 that the control strategy of linear contractive MPC with a contractive sequence of controllable sets is effective since the system state has been driven contractively along the computed controllable sets to the terminal control invariant set and then to the origin asymptotically under the related equivalent control within the selected sliding domain.

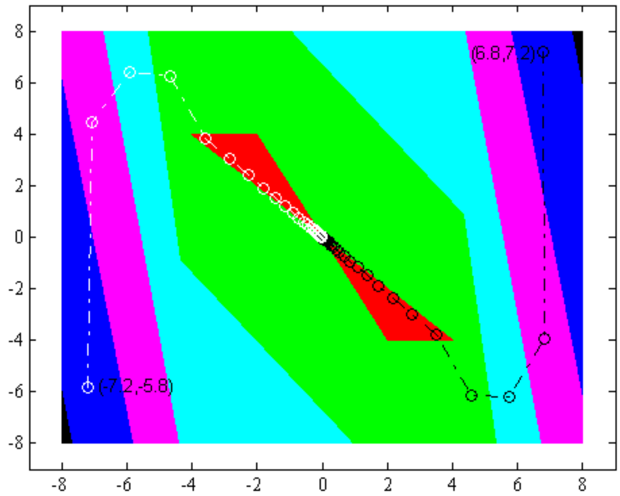


Figure 3.4: Linear contractive MPC with a contractive sequence of controllable sets

3.5 Summary

Convex optimization and polytope geometry have been introduced briefly in this chapter. They have been applied to deal with issues in linear contractive MPC, where linear programming has been applied to find feasible control horizons as well as a feasible control sequence in linear contractive MPC with compulsory contractive constraint and polytope geometry has been applied to compute one-step controllable sets for linear contractive MPC with a contractive sequence of controllable sets. The terminal control invariant set used in the dual-mode approach of linear contractive MPC has been designed in advance as a sliding domain according to the principle of variable structure control. The chattering phenomenon usually happened in the sliding mode of variable structure control has been avoided by the selection of a sliding domain as the terminal set of MPC. The application of convex optimization and polytope geometry to linear contractive MPC in this chapter has provided a theoretical background for the further study of nonlinear contractive MPC in the following chapters.

Chapter 4

Nonlinear Contractive MPC via Classical Interval Analysis

As demonstrated in Chapter 3, convex optimization as well as polytope geometry is a reliable and efficient numerical tool for linear contractive MPC. However, almost all systems encountered in practice are inherently nonlinear, where usually complex nonlinear optimizations are concerned instead. The former tool of convex optimization as well as polytope geometry for linear contractive MPC is not directly suitable for nonlinear contractive MPC and thus new tools are needed further. This chapter introduces a reliable nonlinear numerical tool called classical interval analysis and explores its application in nonlinear contractive MPC with guaranteed feasibility and stability. Classical interval analysis is further generalized to zonotope geometry to test whether a given low-complexity polytope is control invariant for a constrained nonlinear discrete-time system along with a local stabilizing feedback control law. The solver of global optimization for set inversion via zonotope geometry is also proposed to design a terminal control invariant zonotope with a local stabilizing feedback control law for a general constrained nonlinear discrete-time system. One-step controllable sets for nonlinear MPC with a contractive sequence of controllable sets are also obtained via an interval-based algorithm of set computation.

4.1 Classical Interval Analysis

Classical interval analysis is based upon the very simple idea of enclosing real numbers in intervals and real vectors in boxes. Now classical interval analysis has become a fundamental nonlinear numerical tool for representing uncertainties or errors, proving properties of sets, solving sets of equations or inequalities, and optimizing in a global way (Jaulin et al., 2001). Basic concepts and corresponding solvers of classical interval analysis are described briefly in the following subsections.

4.1.1 Basic Concepts

The key concepts of classical interval analysis are interval arithmetic, inclusion function and subpaving, whose definitions are as follows (Jaulin et al., 2001):

Interval Arithmetic: Interval arithmetic is a special case of computation on sets, which include real compact intervals $[a, b] = \{a \leq x \leq b, a \leq b, x, a, b \in \mathbb{R}\}$, real compact interval vectors $\mathbf{X}_{n\times 1}$ and real compact interval matrices $\mathbf{X}_{m\times n}$. The four elementary arithmetic operations $(+, -, \times, \div)$ are extended to intervals. Concretely, for any such binary operator, denoted by $\circ$, performing the operation associated with $\circ$ on the intervals $[a, b]$ and $[c, d]$ means computing $[a, b]\circ[c, d] = [\{x\circ y \in \mathbb{R}|x \in [a, b], y \in [c, d]\}]$, where $[\circ]$ denotes the convex hull of $\{x \circ y \in \mathbb{R}|x \in [a, b], y \in [c, d]\}$. Correspondingly, the set of all interval vectors in the domain of $\mathbb{R}^n$ is denoted to be $\mathbb{I}(\mathbb{R}^n)$.

Inclusion Function: Consider a function $\mathbf{f}$ from $\mathbb{R}^n$ to $\mathbb{R}^m$, the interval function $\mathbf{F}$ from $\mathbb{I}(\mathbb{R}^n)$ to $\mathbb{I}(\mathbb{R}^m)$ is an inclusion function for $\mathbf{f}$ if $\forall \mathbf{X} \in \mathbb{I}(\mathbb{R}^n), \mathbf{f}(\mathbf{X}) \subseteq \mathbf{F}(\mathbf{X})$. The valid semantic statement for $\mathbf{f}(\mathbf{X}) \subseteq \mathbf{F}(\mathbf{X})$ is:

$$\forall x_1 \in [a_1, b_1] \ldots \forall x_n \in [a_n, b_n] \exists \mathbf{z} \in \mathbf{F}(\mathbf{X})\ \mathbf{z} = \mathbf{f}(x_1, \ldots, x_n). \tag{4.1}$$

The natural inclusion function of $\mathbf{f}(\mathbf{X})$ can be obtained by replacing each occurrence of every variable with the corresponding interval variable, by executing all operations according to interval arithmetic, and by computing ranges of the standard functions (Moore, 1966).

Subpaving: A subpaving of a box $\mathbf{X} \in \mathbb{I}(\mathbb{R}^n)$ is a union of non-overlapping sub-boxes with non-zero width, where every subbox is a subset of the box $\mathbf{X}$. A subpaving of $\mathbf{X}$ is regular if each of its subboxes can be obtained from $\mathbf{X}$ by a finite succession of bisections and selections.

The fundamental concepts of classical interval analysis can be integrated to set up various solvers such as set inversion and global optimization. A basic operation within these solvers is to bisect an interval vector into two sub-interval vectors. Taking the interval vector $\mathbf{X} = [a_1, b_1] \times \cdots \times [a_n, b_n]$ as an example, its width is denoted to be (Jaulin et al., 2001):

$$\texttt{Width}(\mathbf{X}) = \max_{i=1,\cdots,n} |a_i - b_i|, \tag{4.2}$$

and the index j is denoted to be:

$$j = \min_{i=1,\cdots,n} \{i|(|a_i - b_i|) = \texttt{Width}(\mathbf{X})\}, \tag{4.3}$$

then the bisection $\texttt{Bisect}(\mathbf{X})$ returns two sub-interval vectors $\texttt{L}\mathbf{X}$ and $\texttt{R}\mathbf{X}$:

$$\begin{cases} \texttt{L}\mathbf{X} := [a_1, b_1] \times \cdots \times [a_j, \frac{(a_j+b_j)}{2}] \times \cdots \times [a_n, b_n] \\ \texttt{R}\mathbf{X} := [a_1, b_1] \times \cdots \times [\frac{(a_j+b_j)}{2}, b_j] \times \cdots \times [a_n, b_n]. \end{cases} \tag{4.4}$$

4.1.2 The Solver of Set Inversion via Interval Analysis

Contrary to the computation of inclusion functions of $\mathbf{f}(\mathbf{X})$ via interval arithmetic, the opposite problem is to find the feasible subpaving of $\mathbf{X}$ with the known image $[a_1, b_1] \times \cdots \times [a_m, b_m]$, i.e., to find $\Sigma_{\mathbf{x}} \subseteq \mathbf{X}$ that satisfies $\mathbf{f}(\Sigma_{\mathbf{x}}) \subseteq [a_1, b_1] \times \cdots \times [a_m, b_m]$. Many physical problems concerning nonlinear inequalities can be formulated as a set inversion problem and the solver of set inversion via interval analysis or the algorithm for set inversion via interval analysis can be applied to find the feasible solutions with a given bound of error tolerance ε. The detail of the solver of set inversion via interval analysis is shown in Algorithm 4.1 (Jaulin et al., 2001), where $\Sigma_{\mathbf{x}}$ is to store all feasible solutions and $\Sigma_{\mathbf{x}}^b$ is to store the neighboring subboxes to $\Sigma_{\mathbf{x}}$ with $\texttt{Width}(\mathbf{X}_k) \leq \varepsilon$.

```
Algorithm 4.1:  Set Inversion Via Interval Analysis  (SIVIA)
```

`In:` $\mathbf{f}, \mathbf{X}, \varepsilon$, `Out:` $\Sigma_{\mathbf{x}}, \Sigma_{\mathbf{x}}^b$

`1. Initialize` $\textsf{Stack} = \mathbf{X}$ `and` $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{x}}^b = \emptyset$;

2. `while` **Stack** $\neq \emptyset$

3.   `Pop out a subbox` $\mathbf{X}_k$ `from` **Stack**;

4.   *if* $\mathbf{F}(\mathbf{X}_k) \subseteq [a_1, b_1] \times \cdots \times [a_m, b_m]$, `then` $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{x}} \cup \mathbf{X}_k$ `and return to 2;`

5.   *elseif* $\mathbf{F}(\mathbf{X}_k) \cap [a_1, b_1] \times \cdots \times [a_m, b_m] = \emptyset$, `then discard` $\mathbf{X}_k$ `and return to 2;`

6.   *elseif* `Width`$(\mathbf{X}_k) \leq \varepsilon$, `then` $\Sigma_{\mathbf{x}}^b = \Sigma_{\mathbf{x}}^b \cup \mathbf{X}_k$ `and return to 2;`

7.   *else*

8.   `Bisect` $\mathbf{X}_k$ `to` L$\mathbf{X}_k$ `and` R$\mathbf{X}_k$, `push them on` **Stack**;

9.   *endif*

10. `endwhile`

For example, for the following multi-affine characteristic polynomial of a closed-loop control system (Jaulin et al., 2001):

$$P(s, \mathbf{p}) = s^3 + (p_1 + p_2 + 2)s^2 + (p_1 + p_2 + 2)s + 2p_1p_2 + 6p_1 + 6p_2 + 3, \tag{4.5}$$

where p_1 and p_2 are uncertain parameters of the system. The corresponding `Routh` vector for the characteristic polynomial is:

$$\mathbf{r}(\mathbf{p}) = \left\{ \begin{array}{l} p_1 + p_2 + 2 \\ (p_1 - 1)^2 + (p_2 - 1)^2 - 1 \\ 2(p_1 + 3)(p_2 + 3) - 15 \end{array} \right\}. \tag{4.6}$$

For $\mathbf{P} = [-3, 9] \times [-3, 9]$ and a given bound of error tolerance $\varepsilon = 0.5$, the subpavings of the parametric space computed via the solver of **SIVIA** are shown in Fig. 4.1, where the parameters p_1 and p_2 covered by the yellow boxes, the blank boxes and the red boxes correspond to stable cases ($\mathbf{r}(\mathbf{p}) > 0$), unstable cases ($\mathbf{r}(\mathbf{p}) \ngtr 0$) and uncertain cases, respectively.

4.1.3 The Solver of Global Optimization via Interval Analysis

Classical interval analysis can also be applied to realize global optimization in a guaranteed numerical way. An algorithm for global minimization is demonstrated here as an illustrative example since it is always possible to transform a global maximization problem into a global minimization problem, for instance, by multiplying the cost

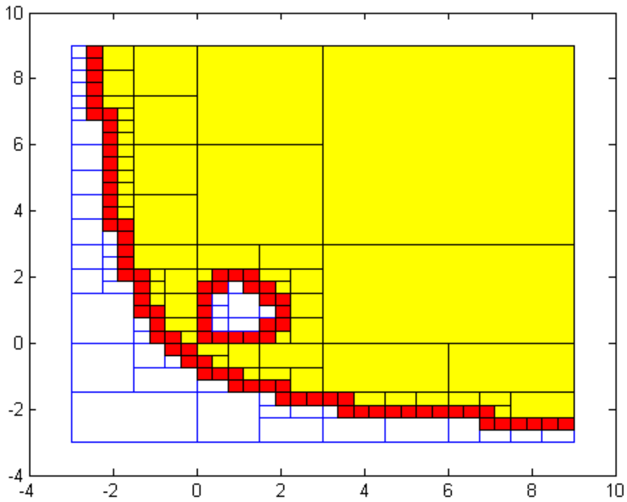


Figure 4.1: An example of set inversion via interval analysis

function by -1 (Jaulin et al., 2001). The problem considered here is the minimization of a scalar cost function $f(\mathbf{x})$ over a compact set $\Sigma_{\mathbf{x}} \subset \mathbb{R}^n$:

$$\min_{\mathbf{x}\in\Sigma_{\mathbf{x}}} f(\mathbf{x}). \tag{4.7}$$

For an unconstrained global minimization problem, $\Sigma_{\mathbf{x}}$ is usually a very large box $\mathbf{X}$ in $\mathbb{I}(\mathbb{R}^n)$; for a constrained global minimization problem, the definition of $\Sigma_{\mathbf{x}}$ involves additional equality and (or) inequality constraints, for instance, $\Sigma_{\mathbf{x}}$ might be defined as:

$$\Sigma_{\mathbf{x}} := \{\mathbf{x} \in \mathbb{R}^n | \mathbf{h}(\mathbf{x}) \leq 0, \mathbf{x} \in \mathbf{X}\}. \tag{4.8}$$

The solver of global minimization via interval analysis is illustrated in Algorithm 4.2 (Jaulin et al., 2001), where $F(\cdot)$ is an inclusion function of $f(\cdot)$, $\bar{f}$ is the upper bound of the global minimum, `Lb` denotes the lower bound of an interval function, and $\hat{F}$ brackets the global minimum by interval evaluations of $f(\cdot)$ over all subboxes in ℓ.

`Algorithm 4.2: Global Minimization Via Interval Analysis (GMVIA)`

`In:` $\Sigma_{\mathbf{x}}, f(\cdot), \varepsilon$, `Out:` $\hat{F}, \ell$

```
Φ := {Σ_x, ∞}; f̄ := ∞; F̂ := ∅; ℓ := ∅;
while Φ ≠ ∅
  Pop out a subbox X_k from Φ;
  if Lb(F(X_k)) ≤ f̄
     if f(Mid(X_k)) < f̄
        f̄ = f(Mid(X_k));
        Remove from Φ any pair (X_i,Lb(F(X_i))) with Lb(F(X_i)) > f̄;
     elseif Width(X_k) ≤ ε
        Push (X_k,Lb(F(X_k))) on ℓ;
     else
        Bisect X_k and push (LX_k,Lb(F(LX_k))) and (RX_k,Lb(F(RX_k))) on Φ;
     endif
  endif
endwhile
Remove from ℓ any pair (X_i,Lb(F(X_i))) with Lb(F(X_i)) > f̄;
For all X_i in ℓ, F̂ = F̂ ∪ F(X_i), and finally F̂ := F̂ ∩ (−∞, f̄].
```

For example, to minimize the following scalar cost function

$$f(x,y) = (x - sin(2x+3y) - cos(3x-5y))^2 + (y - sin(x-2y) + cos(x+3y))^2 \quad (4.9)$$

over $x \in [-2,2]$ and $y \in [-2,2]$ is a typical global optimization problem since the cost function has several local minima, as shown in Fig. 4.2(a). The search process via the interval-based solver of global minimization is shown in Fig. 4.2(b), where the yellow subboxes are intermediate candidates of the global minimizer and the green subboxes are final candidates of the global minimizer. The circled point on the blue box is assumed to be the global minimizer under the given bound of error tolerance $\varepsilon = 0.2$ because the computed value of the cost function on it is smallest among the computed values of the cost function on all vertices of other candidates.

4.2 Zonotope Geometry

This section extends the main concepts in interval analysis to zonotope geometry. Concretely, the implicit definition of a zonotope can be regarded as an extension of an

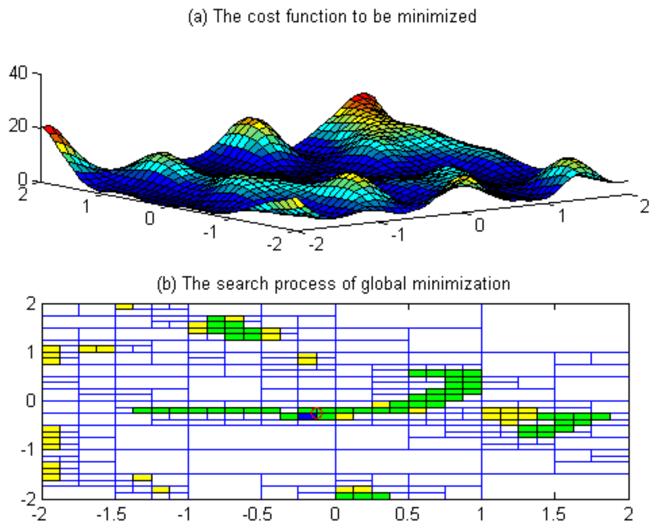


Figure 4.2: An example of global minimization via interval analysis

interval vector or a box in geometry; the explicit construction of a zonotope can be transformed to be the Minkowski sum of polytopes; the bisection of a zonotope is also proposed for the first time according to the idea of bisecting an interval vector; and the zonotope inclusion can be applied for computing the dynamic evolution of nonlinear systems as a kind of centered inclusion functions in contrast to natural inclusion functions usually used in interval analysis.

4.2.1 Zonotope Definition

A zonotope is a centrally symmetric convex polytope and it is closely related to interval analysis. Given a vector $\mathbf{p} \in \mathbb{R}^n$ and a matrix $H \in \mathbb{R}^{n \times m}$, the zonotope $\mathcal{Z}$ of order $n \times m$ is the set:

$$\mathbf{p} \oplus H\mathbf{B}^m = \{\mathbf{p} + H\mathbf{z} | \mathbf{z} \in \mathbf{B}^m\}, \quad (4.10)$$

where $\mathbf{B}^m$ is a box composed of m unitary intervals $\mathbf{B} = [-1, 1]$ and $\oplus$ is the Minkowski sum of sets. Assume that $H = [\mathbf{h}_1 \cdots \mathbf{h}_m]$, then the zonotope can also be regarded as a set spanned by the column vectors of H, which are also called line segment generators:

$$\mathcal{Z} = \{\mathbf{p} + \sum_{i=1}^{m} \alpha_i \mathbf{h}_i | -1 \leq \alpha_i \leq 1\}. \tag{4.11}$$

Geometrically, the zonotope $\mathcal{Z}$ is the transferred Minkowski sum of the line segments defined by the columns of the matrix H to the central point $\mathbf{p}$. Specifically, the zonotope $\mathcal{Z}$ degenerates to be an interval vector as well as a box when H is a diagonal matrix or when $m = 1$. The mathematical concept of zonotopes has not yet been widely used explicitly in the control literature. However, its implicit form, i.e., the Minkowski sum of sets, has already been widely applied to approximate various kinds of invariant sets such as the minimal robust positively invariant set and the minimal disturbance invariant set in a recursive way for discrete-time linear systems (Ong and Gilbert, 2005; Rakovic et al., 2005).

4.2.2 Zonotope Construction

The list of line segment generators is an efficient implicit representation of a zonotope in terms of which set operations such as the Minkowski sum and difference are trivial. However, an explicit representation of a zonotope is needed for some operations such as the judgement of inclusion and exclusion of a polytope to a zonotope. The explicit representation of a zonotope is the zonotope construction problem aiming to list all extreme points of a zonotope defined by its line segment generators. A relatively efficient algorithm was proposed in (Fukuda, 2004) to address the zonotope construction problem, where the addition of line segments was replaced by the addition of convex polytopes. For example, the construction of the zonotope $\mathcal{Z} = \mathbf{p} \oplus H\mathbf{B}^6$, where $\mathbf{p} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ and $H = \begin{bmatrix} 0.4414 & -0.5855 & -0.0484 & 0.2570 & 0.2293 & 0.1498 \\ -0.0016 & -0.3930 & 0.3526 & -0.2396 & 0.4257 & -0.3117 \end{bmatrix}$, can be transformed to be the Minkowski sum of three simpler zonotopes as well as three polytopes, i.e., $\mathcal{Z} = \mathcal{Z}_1 \oplus \mathcal{Z}_2 \oplus \mathcal{Z}_3$, where $\mathcal{Z}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 0.4414 & -0.5855 \\ -0.0016 & -0.3930 \end{bmatrix}\mathbf{B}^2$, $\mathcal{Z}_2 = \begin{bmatrix} -0.0484 & 0.2570 \\ 0.3526 & -0.2396 \end{bmatrix}\mathbf{B}^2$ and $\mathcal{Z}_3 = \begin{bmatrix} 0.2293 & 0.1498 \\ 0.4257 & -0.3117 \end{bmatrix}\mathbf{B}^2$. Then the zonotope $\mathcal{Z}$ can be constructed and plotted as well by using polytope geometry softwares such as **Multi-Parametric Toolbox** (Kvasnica et al., 2006), which is shown in Fig. 4.3.

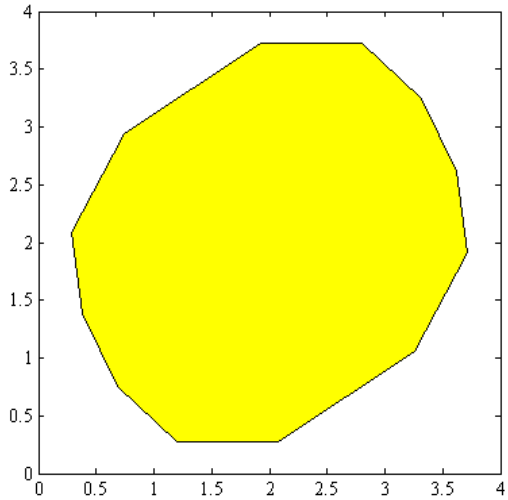


Figure 4.3: An example of zonotope construction

4.2.3 Zonotope Bisection

Similar to an interval vector or a box, a zonotope can be bisected as well. Taking the zonotope $\mathcal{Z} = \mathbf{p} \oplus H\mathbf{B}^m$ as an example, where $\mathbf{p} \in \mathbb{R}^n$, $H \in \mathbb{R}^{n \times m}$ and $m \geq n$, the maximum absolute value among all elements h_{ij} in H is denoted to be:

$$\mathtt{Max}(H) = \max_{i=1,\cdots,n,j=1,\cdots,m} |h_{ij}|, \tag{4.12}$$

and the index k is denoted to be:

$$k = \min_{j=1,\cdots,m} \{j|(|h_{ij}|) = \mathtt{Max}(H), i = 1, \cdots, n\}. \tag{4.13}$$

Then the zonotope $\mathcal{Z}$ can be bisected along the line segment generator $\mathbf{h}_k$, which is addressed in Theorem 4.1.

Theorem 4.1 (Zonotope Bisection) The bisection $\mathtt{Bisect}(\mathcal{Z})$ along the line segment generator $\mathbf{h}_k$ returns two sub-zonotopes $\mathtt{L}\mathcal{Z} = (\mathbf{p} - \frac{\mathbf{h}_k}{2}) \oplus [\mathbf{h}_1 \cdots \frac{\mathbf{h}_k}{2} \cdots \mathbf{h}_m]\mathbf{B}^m$ and $\mathtt{R}\mathcal{Z} = (\mathbf{p} + \frac{\mathbf{h}_k}{2}) \oplus [\mathbf{h}_1 \cdots \frac{\mathbf{h}_k}{2} \cdots \mathbf{h}_m]\mathbf{B}^m$.

Proof. Since $\mathcal{Z} = \mathbf{p} \oplus [\mathbf{h}_1 \cdots \mathbf{h}_k \cdots \mathbf{h}_m]\mathbf{B}^m$, then $\mathcal{Z} = \mathbf{p} \oplus [\mathbf{h}_1 \cdots \mathbf{h}_k \cdots \mathbf{h}_m][[-1, 1]_1 \cdots [-1, 0]_k \cdots [-1, 1]_m]^T \cup \mathbf{p} \oplus [\mathbf{h}_1 \cdots \mathbf{h}_k \cdots \mathbf{h}_m][[-1, 1]_1 \cdots [0, 1]_k \cdots [-1, 1]_m]^T = \mathtt{L}\mathcal{Z} \cup \mathtt{R}\mathcal{Z}$, where $\mathtt{L}\mathcal{Z} = (\mathbf{p} - \frac{\mathbf{h}_k}{2}) \oplus [\mathbf{h}_1 \cdots \frac{\mathbf{h}_k}{2} \cdots \mathbf{h}_m]\mathbf{B}^m$ and $\mathtt{R}\mathcal{Z} = (\mathbf{p} + \frac{\mathbf{h}_k}{2}) \oplus [\mathbf{h}_1 \cdots \frac{\mathbf{h}_k}{2} \cdots \mathbf{h}_m]\mathbf{B}^m$. $\square$

Taking the zonotope shown in Fig. 4.3 as an example, the bisection $\mathtt{Bisect}(\mathcal{Z})$ returns two sub-zonotopes, which are shown in Fig. 4.4. It can be seen that the bisection is not complete for the zonotope of order 2×6. The reason for the overlapping of $\mathtt{L}\mathcal{Z}$ and $\mathtt{R}\mathcal{Z}$ is that the line segment generators $\mathbf{h}_1, \cdots, \mathbf{h}_6$ are not linearly independent or redundant and then the parameters $\alpha_1, \cdots, \alpha_6$ for a same point in $\mathbf{z} = \mathbf{p} + \sum_{i=1}^{6} \alpha_i \mathbf{h}_i$ are sometimes not unique and those points with both positive and negative α_2 in $\mathcal{Z}$ belong to both $\mathtt{L}\mathcal{Z}$ and $\mathtt{R}\mathcal{Z}$. However, for a zonotope $\mathcal{Z}$ with linearly independent line segment generators, the bisection is complete, which is addressed in Theorem 4.2.

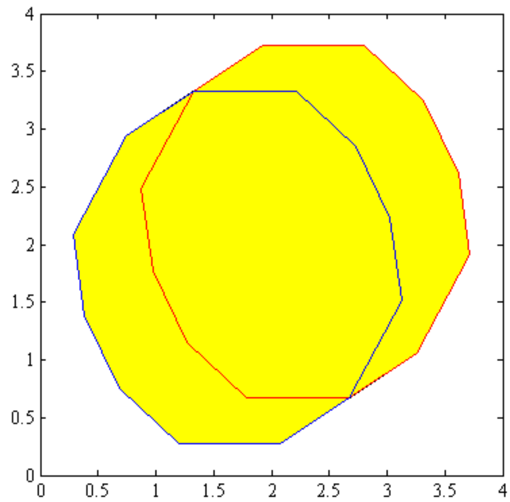


Figure 4.4: Bisection of a zonotope with redundant line segment generators

Theorem 4.2 (Complete Bisection) For a zonotope $\mathcal{Z} = \mathbf{p} \oplus H\mathbf{B}^n$, where $\mathbf{p} \in \mathbb{R}^n$, $H \in \mathbb{R}^{n \times n}$ and $\mathtt{Rank}(H) = n$, the defined bisection is complete, i.e., $\mathtt{L}\mathcal{Z}$ and $\mathtt{R}\mathcal{Z}$ only share a face of dimension $n-1$.

Proof. For $\mathcal{Z} = \{\mathbf{p} + \sum_{i=1}^{n} \alpha_i \mathbf{h}_i | -1 \le \alpha_i \le 1\}$, assume that there exist $-1 \le \alpha_1^L \le 1, \cdots, -1 \le \alpha_k^L \le 0, \cdots, -1 \le \alpha_n^L \le 1$ and $-1 \le \alpha_1^R \le 1, \cdots, 0 \le \alpha_k^R \le 1, \cdots, -1 \le \alpha_n^R \le 1$, s.t. $\mathbf{p} + \sum_{i=1}^{n} \alpha_i^L \mathbf{h}_i = \mathbf{p} + \sum_{i=1}^{n} \alpha_i^R \mathbf{h}_i$, then $\sum_{i=1}^{n} (\alpha_i^L - \alpha_i^R)\mathbf{h}_i = 0$ while

$\texttt{Rank}(H) = n$, so $\alpha_k^L = \alpha_k^R = 0$, i.e., $\texttt{L}\mathcal{Z}$ and $\texttt{R}\mathcal{Z}$ only share a face of dimension $n-1$. $\square$

Taking the bisection of the zonotope $\mathcal{Z} = \mathbf{p} \oplus H\mathbf{B}^2$ as an example, where $\mathbf{p} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ and $H = \begin{bmatrix} 0.6414 & 0.5855 \\ -0.4016 & 0.7930 \end{bmatrix}$, since $\texttt{Rank}(H) = 2$, so the bisection is complete according to Theorem 4.2, just as shown in Fig. 4.5.

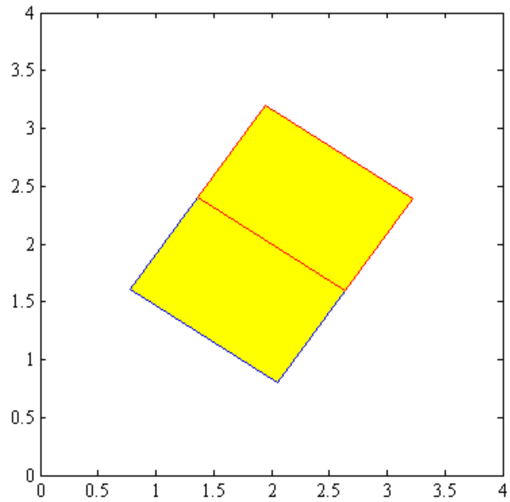


Figure 4.5: Bisection of a zonotope with linearly independent line segment generators

4.2.4 Zonotope Inclusion

Using zonotopes, Kühn developed a procedure to bound the orbits of discrete-time dynamic systems with a guaranteed sub-exponential overestimation. The following theorem introduces the zonotope inclusion operator of Kühn's method (Kühn, 1998).

Theorem 4.3 (Zonotope Inclusion) Consider a family of zonotopes represented by $\mathbb{Z} = \mathbf{p} \oplus \mathbf{M}\mathbf{B}^m$, where $\mathbf{p} \in \mathbb{R}^n$ is a real vector and $\mathbf{M} \in \mathbb{I}^{n\times m}$ is an interval matrix. A zonotope inclusion, denoted by $\diamond(\mathbb{Z})$, is defined by:

$$\diamond(\mathbb{Z}) = \mathbf{p} \oplus [\texttt{Mid}(\mathbf{M}) \;\; G] \begin{bmatrix} \mathbf{B}^m \\ \mathbf{B}^n \end{bmatrix}, \tag{4.14}$$

where $\mathtt{Mid}(\mathbf{M})$ is the centered-point matrix of $\mathbf{M}$ and $G \in \mathbb{R}^{n\times n}$ is a diagonal matrix that satisfies:

$$G_{ii} = \sum_{j=1}^{m} \frac{\mathtt{Diam}(\mathbf{M}_{ij})}{2}, i = 1, \cdots, n, \tag{4.15}$$

where $\mathtt{Diam}(\mathbf{M}_{ij})$ is the length of the interval $\mathbf{M}_{ij}$. Under these definitions, it results that: $\mathbb{Z} \subseteq \diamond(\mathcal{Z})$.

Given a possibly nonlinear one-order differential function $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n, \mathbf{x} \in \mathcal{X} = \mathbf{p} \oplus M\mathbf{B}^m$, its centered inclusion function $\mathbf{F}_c(\mathcal{X}) : \mathbf{f}(\mathcal{X}) \subseteq \mathbf{F}_c(\mathcal{X})$ can be deduced by the mean-value theorem (Jaulin et al., 2001), i.e.,

$$\mathbf{F}_c(\mathcal{X}) \triangleq \mathbf{f}(\mathbf{p}) + \bigtriangledown_{\mathbf{x}}\mathbf{f}(\mathcal{X})(\mathcal{X} - \mathbf{p}), \tag{4.16}$$

where $\mathcal{X} - \mathbf{p} = M\mathbf{B}^m$. Thus the centered inclusion function $\mathbf{F}_c(\mathcal{X})$ of $\mathbf{f}(\mathbf{x})$ turns out to be a family of zonotopes represented by $\mathbb{Z} = \mathbf{p}_{new} \oplus \mathbf{M}_{new}\mathbf{B}^m$, where $\mathbf{p}_{new} = \mathbf{f}(\mathbf{p})$ and $\mathbf{M}_{new} = \bigtriangledown_{\mathbf{x}}\mathbf{f}(\mathcal{X})M$, which can be further bounded by its corresponding zonotope inclusion $\diamond(\mathcal{Z})$. This is the primary principle of Kühn's method to bound the evolution of dynamic systems by zonotopes, where centered inclusion functions are applied instead of natural inclusion functions. Kühn's method can be further extended to bound the evolution of nonlinear uncertain discrete-time systems, which is addressed in the following theorem (Alamo et al., 2003).

Theorem 4.4 (Uncertain Evolution) Given a one-order differential function $\mathbf{f}(\mathbf{x}, \mathbf{w}), \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n, \mathbf{w} \in \mathbf{W} \subset \mathbb{R}^l$, where $\mathcal{X}$ is a zonotope: $\mathbf{p} \oplus H\mathbf{B}^m$ and $\mathbf{W}$ is a box, consider the following:

- A zonotope $\mathbf{q} \oplus S\mathbf{B}^n : \mathbf{f}(\mathbf{p}, \mathbf{W}) \subseteq \mathbf{q} \oplus S\mathbf{B}^n$.
- An interval matrix $\mathbf{M} = \bigtriangledown_{\mathbf{x}}\mathbf{f}(\mathcal{X}, \mathbf{W})H$.
- A zonotope $\mathbf{q} \oplus S\mathbf{B}^n \oplus \diamond(\mathbf{M}\mathbf{B}^m)$.

Under the above assumptions, it results that:
$\mathbf{f}(\mathcal{X}, \mathbf{W}) \subseteq \mathbf{q} \oplus S\mathbf{B}^n \oplus \diamond(\mathbf{M}\mathbf{B}^m)$.

Proof. Given a $\mathbf{w} \in \mathbf{W}$, the application of the mean-value theorem yields:
$\mathbf{f}(\mathcal{X}, \mathbf{w}) \subseteq \mathbf{f}(\mathbf{p}, \mathbf{w}) \oplus \bigtriangledown_{\mathbf{x}}\mathbf{f}(\mathcal{X}, \mathbf{w})H\mathbf{B}^m \Rightarrow \mathbf{f}(\mathcal{X}, \mathbf{W}) \subseteq \mathbf{f}(\mathbf{p}, \mathbf{W}) \oplus \bigtriangledown_{\mathbf{x}}\mathbf{f}(\mathcal{X}, \mathbf{W})H\mathbf{B}^m \subseteq$

$$\mathbf{q} \oplus S\mathbf{B}^n \oplus \bigtriangledown_{\mathbf{x}} \mathbf{f}(\mathcal{X}, \mathbf{W}) H\mathbf{B}^m = \mathbf{q} \oplus S\mathbf{B}^n \oplus \mathbf{M}\mathbf{B}^m \subseteq \mathbf{q} \oplus S\mathbf{B}^n \oplus \diamond(\mathbf{M}\mathbf{B}^m). \ \square$$

Consequently, zonotopes can also be applied to bound the evolution of general nonlinear uncertain discrete-time systems with reduced wrapping effects since zonotope inclusions derived from the mean-value theorem are used instead of natural inclusion functions, where a sort of linearization is performed for the range of functions (Alamo et al., 2003). The reduced wrapping effect using zonotope evolution can be seen in an illustrative example shown in Fig. 4.6, where the interval evolution and the zonotope evolution of four steps for the nonlinear discrete-time system discussed in (Cannon et al., 2003) with the same control sequence and the same initial state domain are compared.

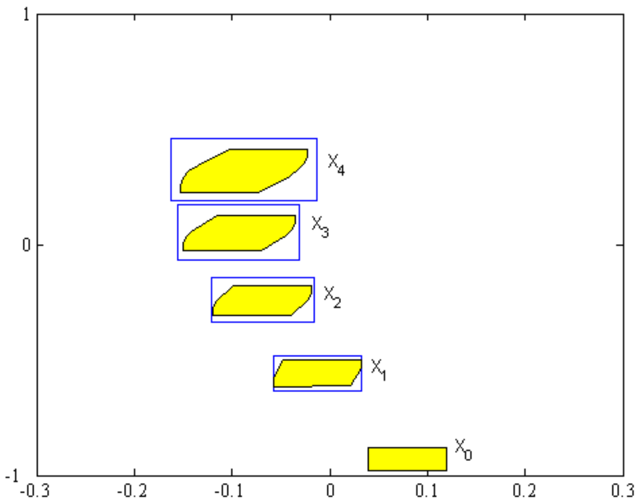


Figure 4.6: The zonotope evolution vs the interval evolution

4.2.5 Set Inversion via Zonotope Geometry

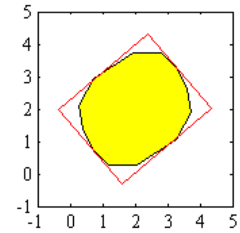
Given a dynamic system $\mathbf{f}(\mathbf{x})$, a zonotope $\mathcal{X}$ as an initial admissible domain and the codomain $\mathbb{T}$, the solver of set inversion via zonotope geometry is listed in Algorithm 4.3, where ε is the bound of error tolerance and $\Sigma_{\mathbf{x}}, \Sigma_{\mathbf{x}}^b$ are to store the feasible sub-zonotopes and the neighboring sub-zonotopes with $\text{Max}(H_i) \le \varepsilon$ to all the feasible sub-zonotopes, respectively.

Algorithm 4.3: Set Inversion Via Zonotope Geometry (SIVZG)

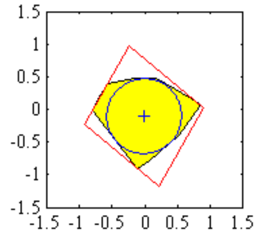
In: $\mathbf{f}, \mathcal{X}, \varepsilon$, Out: $\Sigma_{\mathbf{x}}, \Sigma_{\mathbf{x}}^b$

1. Initialize **Stack** $= \mathcal{X}$ and $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{x}}^b = \emptyset$;
2. while **Stack** $\neq \emptyset$
3.   Pop out a zonotope $\mathcal{X}_i = \mathbf{p}_i \oplus H_i \mathbf{B}^m$ from **Stack**;
4.   *if* $\mathbf{F}_c(\mathcal{X}_i) \subseteq \mathbb{T}$, $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{x}} \cup \mathcal{X}_i$ and return to 2;
5.   *elseif* $\mathbf{F}_c(\mathcal{X}_i) \cap \mathbb{T} = \emptyset$, discard $\mathcal{X}_i$ and return to 2;
6.   *elseif* $\text{Max}(H_i) \le \varepsilon$, $\Sigma_{\mathbf{x}}^b = \Sigma_{\mathbf{x}}^b \cup \mathcal{X}_i$ and return to 2;
7.   *else*
8.   Bisect $\mathcal{X}_i$ to $\text{L}\mathcal{X}_i$ and $\text{R}\mathcal{X}_i$, push them on **Stack**;
9.   *endif*
10. endwhile

The solver of set inversion via zonotope geometry is similar to the solver of set inversion via interval analysis and only the bisections and evolutions of interval vectors are replaced by the bisections and evolutions of zonotopes (Jaulin et al., 2001), where the initial admissible domain is broadened from boxes to zonotopes and the wrapping effects have been reduced by zonotope evolutions instead of interval evolutions. Since the bisection of a zonotope with redundant line segment generators is not complete, an alternative approach is to bound the zonotope with redundant line segment generators by a zonotope with linearly independent line segment generators at first and thus the bisection of the bounding zonotope is complete. The bounding of a zonotope with redundant line segment generators can be realized by using singular value decomposition of the matrix H or a recursive algorithm proposed in (Bravo, 2004). Furthermore, an initial admissible polytope can also be bounded by a zonotope with linearly independent line segment generators using the center of the largest ball inscribed in the polytope as the center of the bounding zonotope (Guibas et al., 2003; Kvasnica et al., 2006). The illustrative examples for the bounding of a zonotope with redundant line segment generators by a parallelogram and the bounding of a polytope by a parallelogram are shown in Fig. 4.7(a) and 4.7(b), respectively.



(a) The bounding of a zonotope of order 2x6 by a parallelogram



(b) The bounding of a 2-D polytope by a parallelogram

Figure 4.7: Bounding by a zonotope with linearly independent line segment generators

Once a polytope has been bounded by a zonotope with linearly independent line segment generators, the solver of set inversion via zonotope geometry can be extended to the set inversion problem with a polytope as the initial admissible domain. A direct application of the solver in Algorithm 4.3 is shown in Fig. 4.8, where a polytope is approximated innerly by a union of zonotopes using a bounding zonotope of the polytope as the initial admissible domain.

4.2.6 Set Inversion via Zonotope Geometry for Set Invariance Test

This subsection gives an illustrative example for the application of the solver of set inversion via zonotope geometry to test set invariance in constrained control. Set invariance plays a significant role in MPC, where a terminal control invariant set along with a local stabilizing feedback control law is usually needed for the terminal control of the dual-mode approach of MPC (Blanchini, 1999; Mayne et al., 2000). However, the design of such a terminal control invariant set for a general constrained nonlinear

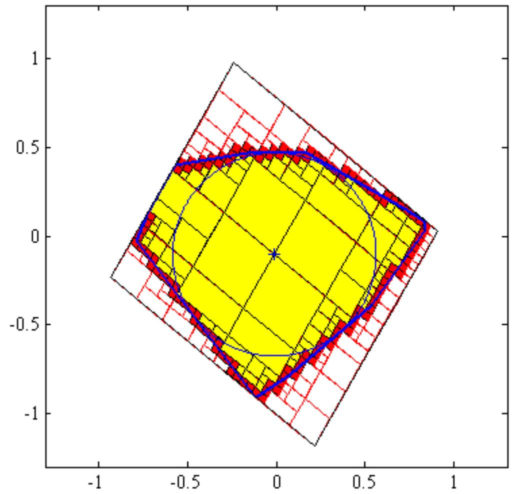


Figure 4.8: Inner approximation of a polytope by a union of zonotopes

discrete-time system is still an open problem while the local stabilizing feedback control law can often be designed in advance as a LQ problem for the corresponding linearized system of the original nonlinear system (Magni et al., 2001). The proposed solver of set inversion via zonotope geometry provides a numerical and geometric method to test whether a given low-complexity polytope as well as a zonotope is control invariant or no under the assigned local stabilizing feedback control law.

Taking the system discussed in (Cannon et al., 2003) as the illustrative example, which is:

$$\begin{cases} x_1(k+1) = x_1(k) + 0.01x_2(k) + 0.01[\mu + (1-\mu)x_1(k)]u(k) \\ x_2(k+1) = 0.01x_1(k) + x_2(k) + 0.01[\mu - 4(1-\mu)x_2(k)]u(k), \end{cases} \tag{4.17}$$

where $\mu = 0.9$, $|u| \leq 2$ and $|\mathbf{x}|_\infty \leq 4$. The local stabilizing feedback control law is selected to be $u = [-1.2131 - 1.2128]\mathbf{x}$ and the terminal set is a low-complexity polytope $\mathcal{P} : |V\mathbf{x}|_\infty \leq 0.9$ [1], where $V = \begin{bmatrix} 0.1638 & -0.3931 \\ 0.6066 & 0.6066 \end{bmatrix}$. The low-complexity polytope $\mathcal{P}$ is

[1]The original polytope in the paper of (Cannon et al., 2003) is $|V\mathbf{x}|_\infty \leq 1$, however, an initial test of $u = [-1.2131-1.2128]\mathbf{x}$ shows that the maximal value of u on it is 2.0002, which is not an admissible control. Such a tiny violation might be induced by the computation errors for the vertices and so a smaller polytope is adopted here for the following test.

also a zonotope and it is to be bisected to test whether $u = [-1.2131 - 1.2128]\mathcal{X}(k) \subseteq [-2, 2]$ and $\mathcal{X}(k+1) = \mathbf{F}_c(\mathcal{X}(k), k\mathcal{X}(k)) \subseteq \mathcal{P}$ for all sub-zonotopes $\mathcal{X}(k)$, as shown in Fig. 4.9. The test result using the solver of set inversion via zonotope geometry in Algorithm 4.3 shows that all sub-zonotopes are control invariant under the related local stabilizing feedback control law $u = [-1.2131 - 1.2128]\mathbf{x}$, i.e., the dynamic evolutions of all sub-zonotopes under the related local stabilizing feedback control law are all within the selected polytope $\mathcal{P}$. Then the provided polytope $\mathcal{P}$ has been demonstrated geometrically to be a valid control invariant set. By testing various zonotopes, the solver of set inversion via zonotope geometry can also be extended to design a terminal robust control invariant zonotope for a general constrained nonlinear uncertain discrete-time system in a numerical way.

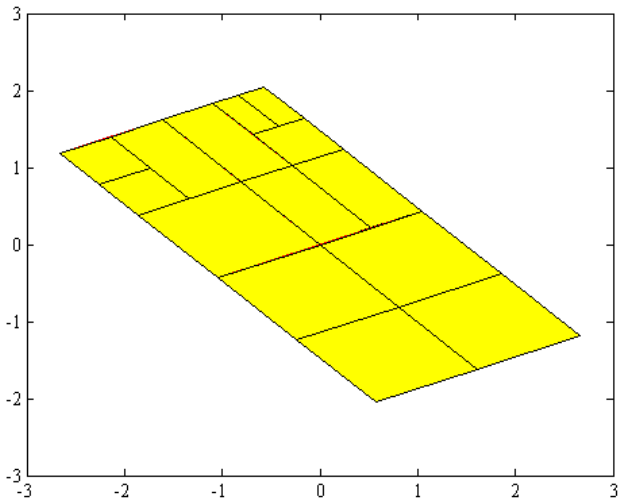


Figure 4.9: Set inversion via zonotope geometry for set invariance test

4.2.7 Global Optimization for Set Inversion via Zonotope Geometry

Given an initial interval matrix $\mathbf{M} \in \mathbb{I}^{n\times m}$, its width is denoted to be:

$$\mathtt{Width}(\mathbf{M}) = \max_{i,j} \mathtt{Width}(\mathbf{M}_{ij}), i = 1, \cdots, n, j = 1, \cdots, m, \tag{4.18}$$

where $\texttt{Width}(\mathbf{M}_{ij})$ is denoted to be the width of the interval $\mathbf{M}_{ij}$. The interval matrix $\mathbf{M}$ can also be bisected into two sub-interval matrix $\texttt{L}\mathbf{M}$ and $\texttt{R}\mathbf{M}$ by bisecting the widest member of its components. Using the bisection and selection of $\mathbf{M}$, the solver of global optimization for set inversion via zonotope geometry can be built to search a biggest (robust) control invariant zonotope for a constrained nonlinear system with a local stabilizing feedback control law $\mathbf{u} = \mathbf{k}\mathbf{x}$, just as shown in Algorithm 4.4.

Algorithm 4.4: Global Optimization For Set Inversion Via Zonotope Geometry (GOFSIVZG)

In: $\mathbf{M}, \varepsilon$, Out: $\mathcal{Z}$

1. Initialize **Stack** $= \mathbf{M}$ and $\mathcal{Z} = \emptyset$;
2. while **Stack** $\neq \emptyset$
3.   Pop out an interval matrix $\mathbf{M}_k$ from **Stack**;
4.   Test if $\diamond(\mathcal{Z}_k) = \mathbf{M}_k\mathbf{B}^m$ is control invariant or no by SIVZG;
5.   *if* $\diamond(\mathcal{Z}_k)$ is control invariant and $\texttt{Vol}(\diamond(\mathcal{Z}_k)) > \texttt{Vol}(\mathcal{Z})$, $\mathcal{Z} = \diamond(\mathcal{Z}_k)$ and return to 2;
6.   *elseif* $\texttt{Width}(\mathbf{M}) \leq \varepsilon$, return to 2;
7.   *else*
8.   Bisect $\mathbf{M}_k$ to $\texttt{L}\mathbf{M}_k$ and $\texttt{R}\mathbf{M}_k$, push them on **Stack**;
9.   *endif*
10. endwhile

As shown in Algorithm 4.4, the solver of global optimization for set inversion via zonotope geometry searches the biggest control invariant zonotope starting from an initial zonotope derived from an initial interval matrix $\mathbf{M}$. The zonotope inclusion of each family of zonotopes represented by $\mathbf{M}_k\mathbf{B}^m$ is passed to the solver of set inversion via zonotope geometry for the test of control invariance with the related local stabilizing feedback control law. It is worthy to note that the complexity and the volume of the obtained optimal control invariant zonotope $\mathcal{Z}$ is closely related to the dimension and the value of the selected initial interval matrix $\mathbf{M}$. Furthermore, other kinds of local stabilizing feedback control laws can be applied as well because the solver of set inversion via zonotope geometry is applicable to a whatever nonlinear autonomous system with first-order differentiability for deriving its centered inclusion function.

Taking the highly nonlinear model of a Continuous Stirred-Tank Reactor (CSTR) (Limon et al., 2003; Magni et al., 2001) as an example, assuming constant liquid volume, the CSTR for an exothermic, irreversible reaction, $\mathbf{A} \rightarrow \mathbf{B}$, is described by the following dynamic model based on a component balance for the reactant $\mathbf{A}$ and an energy balance:

$$\begin{cases} \dot{C_A} = \frac{q}{V}(C_{Af} - C_A) - k_0 \exp(-\frac{E}{RT})C_A, \\ \dot{T} = \frac{q}{V}(T_f - T) + \frac{(-\triangle H)}{\rho C_p} k_0 \exp(-\frac{E}{RT})C_A + \frac{UA}{V\rho C_p}(T_c - T), \end{cases} \tag{4.19}$$

where C_A is the concentration of $\mathbf{A}$ in the reactor, T is the reactor temperature, and T_c is the temperature of the coolant stream. The constraints are $280\mathbf{K} \leq T_c \leq 370\mathbf{K}$, $280\mathbf{K} \leq T \leq 370\mathbf{K}$ and $0 \leq C_A \leq 1mol/l$. The objective is to regulate C_A and T by manipulating T_c. The nominal operating conditions, which correspond to an unstable equilibrium $C_A^{eq} = 0.5mol/l, T^{eq} = 350\mathbf{K}, T_c^{eq} = 300\mathbf{K}$ are: $q = 100l/min, C_{Af} = 1mol/l, T_f = 350\mathbf{K}, V = 100l$, $\rho = 1000g/l, C_p = 0.239\mathbf{J}/g\mathbf{K}$, $\triangle H = -5 \times 10^4\mathbf{J}/mol, E/R = 8750\mathbf{K}$, $k_0 = 7.2 \times 10^{10}min^{-1}, UA = 5 \times 10^4\mathbf{J}/min\mathbf{K}$. The nonlinear discrete-time state-space model is obtained by defining the state vector $\mathbf{x} = [C_A - C_A^{eq} \quad (T - T^{eq})/100]^T$, the manipulated input $u = (T_c - T_c^{eq})/100$ and by discretizing the ODE with a sampling time $\triangle t = 0.03min$ using the Euler method, which is the following discrete-time model:

$$\begin{cases} x_1(k+1) = x_1(k) + 0.03(\frac{q}{V}(C_{Af} - (x_1(k) + C_A^{eq})) - k_0 \exp(-\frac{E}{R(100x_2(k)+T^{eq})})(x_1(k) + C_A^{eq})) \\ x_2(k+1) = x_2(k) + 0.0003(\frac{q}{V}(T_f - (100x_2(k) + T^{eq})) + \frac{(-\triangle H)}{\rho C_p} k_0 \exp(-\frac{E}{R(100x_2(k)+T^{eq})})(x_1(k) + \\ \qquad C_A^{eq}) + \frac{UA}{V\rho C_p}(100u + T_c^{eq} - (100x_2(k) + T^{eq}))), \end{cases} \tag{4.20}$$

A local stabilizing feedback control law $u = [-0.0690 \ -4.3387]\mathbf{x}$ is designed in advance according to the linearized model and the LQ method (Magni et al., 2001). With the designed local stabilizing feedback control law, a control invariant zonotope $\mathcal{Z} = \begin{bmatrix} 0.03 & -0.01 & 0.02 & 0 \\ 0.01 & 0.01 & 0 & 0.02 \end{bmatrix} \mathbf{B}^4$ is obtained through the solver of global optimization via zonotope geometry, where the initial searching interval matrix is selected to be $\mathbf{M} = \begin{bmatrix} [0,0.04] & [-0.04,0] \\ [0,0.04] & [0, 0.04] \end{bmatrix}$; the bound of error tolerance for global optimization is selected to be $\varepsilon = 0.005$ while the bound of error tolerance for set inversion is selected to be $\varepsilon = 0.05$. The optimized zonotope can be demonstrated geometrically to be control invariant, just as shown in Fig. 4.10.

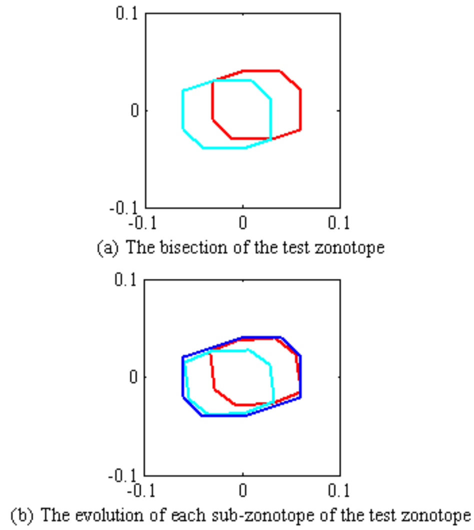


Figure 4.10: The optimized control invariant zonotope

4.3 Nonlinear MPC with Compulsory Contractive Constraint

Similar to linear contractive MPC with compulsory contractive constraint addressed in Chapter 3, the main feature of nonlinear contractive MPC with compulsory contractive constraint is also the imposed contractive constraint, usually in the form of contractive norms of the state vector. However, feasible control horizons can hardly be obtained analytically via local controllability analysis or numerically via linear matrix inequalities and the resulting constrained optimizations are mostly non-convex because of the nonlinearity of the system concerned. In the following subsections, the nonlinear numerical tool of classical interval analysis is applied to confront such tough issues encountered in nonlinear contractive MPC with compulsory contractive constraint.

4.3.1 Problem Statement

Assume that the constrained nonlinear discrete-time system considered is described by the following state-space model:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)), k = 0, 1, \cdots, \tag{4.21}$$

where $\mathbf{x}(k) \in \mathbb{X} \subset \mathbf{R}^n$ is a vector of n state variables and $\mathbb{X}$ is a compact set containing the origin to constrain the state; $\mathbf{u}(k) \in \mathbb{U} \subset \mathbf{R}^m$ is a vector of m process inputs or manipulated variables and $\mathbb{U}$ is a compact set containing the origin to constrain the control. The control target is to drive the system state to the origin asymptotically and the control inputs are to be obtained iteratively via the following constrained open-loop optimization:

$$J(\mathbf{x}(k), \{\mathbf{u}(k+i|k)\}_{i=0}^{N_c-1}) = \min_{\{\mathbf{u}_{k+i|k} \in \mathbb{U}\}_{i=0}^{N_c-1}} [\mathbf{x}(k+N_c|k)^T P_0 \mathbf{x}(k+N_c|k) + \sum_{i=1}^{N_c-1} \mathbf{x}^T(k+i|k) Q \mathbf{x}(k+i|k) + \sum_{i=0}^{N_c-1} (\mathbf{u}(k+i|k))^T R \mathbf{u}(k+i|k)] \quad (4.22)$$

subject to

$$\begin{cases} \mathbf{x}(k+i|k) \in \mathbb{X}, i = 1, \cdots, N_c - 1 \\ |\mathbf{x}(k+N_c|k)|_p < \alpha \cdot |\mathbf{x}(k|k)|_p, \end{cases} \quad (4.23)$$

where N_c is the control horizon and $|\cdot|_p$ is denoted to the p-norm; $\alpha \in (0, 1]$ is the contractive parameter for the p-norm of the state vector (de Oliveira and Morari, 2000).

4.3.2 Feasible Control Horizons via Set Inversion

The key issue of nonlinear contractive MPC with compulsory contractive constraint is also to find a feasible control horizon N_c that renders $|\mathbf{x}(k+N_c|k)|_p \leq |\mathbf{x}(k|k)|_p$ as well as $\mathbf{x}(k+i|k) \in \mathbb{X}, i = 1, \cdots, N_c-1$, which can be transformed to be a set inversion problem in classical interval analysis, i.e., for a given N_c, it is a feasible control horizon if the solution set for the admissible control sequence $\pi_{\mathbf{u}} = \{\mathbf{u}(k|k), \mathbf{u}(k+1|k), \cdots, \mathbf{u}(k+N_c-1|k)\}$ satisfying all the imposed constraints is not empty. Then the solver of set inversion via interval analysis illustrated in Algorithm 4.1 can be applied directly to see whether the solution set for the selected control horizon with a given bound of error tolerance ε is empty or no: if the solution set is empty, the control horizon should be prolonged for another test; if the solution set is not empty, then the selected control horizon is feasible for all the imposed constraints including the imposed compulsory contractive constraint.

4.3.3 Nonlinear Contractive MPC via Global Optimization

Once a feasible control horizon N_c has been obtained by the solver of set inversion via interval analysis, the resulting constrained open-loop optimization can also be solved by the solver of global minimization via interval analysis (Wan et al., 2004b), as addressed in Algorithm 4.2. The overall control scheme of the nonlinear contractive MPC as well as the transferred feasibility test for a feasible control horizon is illustrated in Algorithm 4.5.

```
Algorithm 4.5:  Nonlinear MPC With Compulsory Contractive Constraint
In:  x(0),α);  Out:  x(k)
1.   Get the current state x(k);
2.   if x(k) = 0
3.       Stop;
4.   else
5.       N_c = 1, sign_f = 0;
6.       while (sign_f = 0);
7.               [sign_f, π_u^Optimal] = feasibletest(N_c);
8.               N_c := N_c + 1;
9.       endwhile
10.      Apply π_u^Optimal to the system;
11.  endif
12.  Return to 1 and circulate.
```

As shown in Algorithm 4.5, the feasibility test is an iterative algorithm with the inherent transferred function $\texttt{feasibletest}(N_c)$, which is to be solved via the solver of set inversion via interval analysis. The initial control horizon N_c is set to be 1 and the initial feasible sign $\texttt{sign}_f$ is set to be 0, which stands for unfeasible cases. If the feasible test demonstrates that there exists a feasible control sequence $\pi_{\mathbf{u}}$ satisfying all the imposed constraints, the transferred function renews the feasible sign $\texttt{sign}_f$ to be 1 and returns an optimal feasible control sequence $\pi_{\mathbf{u}}^{Optimal}$ by the solver of global minimization via interval analysis. Otherwise, the function $\texttt{feasibletest}(N_c)$ maintains the former value of the feasible sign $\texttt{sign}_f$ and increases the control horizon N_c for another feasibility test.

4.3.4 Example

Consider the constrained nonlinear discrete-time system described by the state-space model (Grimm et al., 2004):

$$\begin{cases} x_1(k+1) = x_1(k)(1-u(k)) \\ x_2(k+1) = u(k)\sqrt{x_1^2(k) + x_2^2(k)}, \end{cases} \tag{4.24}$$

where $x_1(0) = 2, x_2(0) = 2$ and the control target is to drive the system state to the origin with the imposed control constraint $u \in [0,1]$ and the state constraint $|\mathbf{x}|_\infty \in [-3,3]$. Using the control algorithm in Algorithm 4.5, the control process for the system is shown in Fig. 4.11, where the system states x_1, x_2 are being stabilized asymptotically as time goes on.

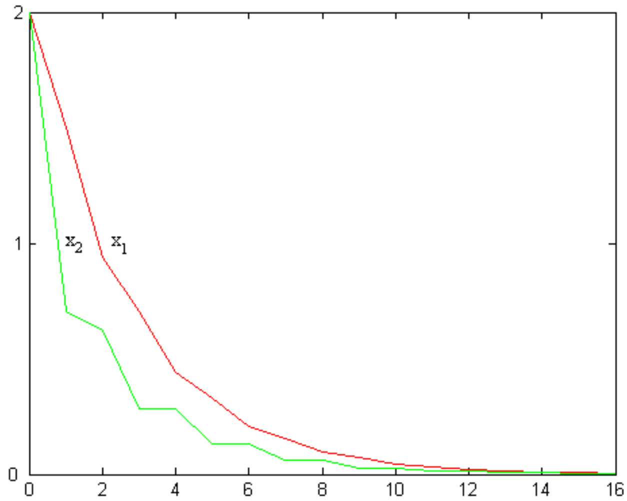


Figure 4.11: Nonlinear MPC with compulsory contractive constraint

4.4 Nonlinear MPC with A Contractive Sequence of Controllable Sets

This section extends linear contractive MPC with a contractive sequence of controllable sets discussed in Chapter 3 to nonlinear contractive MPC with a contractive sequence of controllable sets, where one-step controllable sets of the constrained nonlinear discrete-time system are to be computed in advance on the basis of the solver of set inversion via interval analysis.

4.4.1 Problem Statement

The system to be considered is described by the following constrained nonlinear discrete-time state-space model:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)), k = 0, 1, \cdots, \tag{4.25}$$

where $\mathbf{x}(k) \in \mathbb{X} \subset \mathbb{R}^n$ is a vector of n state variables and $\mathbb{X}$ is a compact set containing the origin; $\mathbf{u}(k) \in \mathbb{U} \subset \mathbb{R}^m$ is a vector of m control inputs and $\mathbb{U}$ is a compact set containing the origin. The domains of $\mathbb{X}$ and $\mathbb{U}$ are assumed to be described by boxes $\mathbf{X}$ and $\mathbf{U}$, i.e., every component of the vectors is an interval. The control target is to drive the system from the initial state $\mathbf{x}(0)$ to the origin asymptotically. The dual-mode approach of MPC is adopted here: at first, the one-step control deriving from contractive MPC drives the system state into a selected terminal control invariant set $\mathbb{T}$; then the related local stabilizing feedback control law is applied instead to drive the system state to the origin asymptotically. Assume that all controllable sets $\mathcal{K}_i(\mathbf{X}, \mathbb{T}), i = 1, \cdots$ within the constrained state space have been obtained by computing one-step controllable sets recursively, then the one-step control inputs of nonlinear contractive MPC with a contractive sequence of controllable sets can be obtained iteratively by solving the following open-loop optimization:

$$\min_{\mathbf{u}(k|k) \in \mathbf{U}} [\mathbf{x}^T(k+1|k) Q \mathbf{x}(k+1|k) + \mathbf{u}^T(k|k) R \mathbf{u}(k|k)] \tag{4.26}$$

subject to

$$\mathbf{x}(k+1|k) \in \mathcal{K}_{i-1}(\mathbf{X}, \mathbb{T}), \tag{4.27}$$

where $\mathbf{x}(k) \in \mathcal{K}_i(\mathbf{X}, \mathbb{T})$, but $\mathbf{x}(k)$ does not belong to $\mathcal{K}_{i-1}(\mathbf{X}, \mathbb{T})$; Q and R are weighted positive definite matrices; and $\mathbf{u}^{Optimal}(k|k)$ is the resulting optimal one-step control input. The terminal control invariant set $\mathbb{T}$ can be designed in advance to be a control invariant polytope along with a local stabilizing feedback control law $u = \mathbf{k}\mathbf{x}$ (Cannon et al., 2003).

4.4.2 The Computation of Controllable Sets via Set Inversion

Assume that the terminal control invariant set $\mathbb{T}$ is designed to be a control invariant polytope, then an inner approximation Σ^- of the one-step controllable set $\mathcal{K}(\mathbf{X}, \mathbb{T})$ can be computed directly via an interval-based branch-and-bound algorithm based on the solver of set inversion via interval analysis, which is listed in Algorithm 4.6, where Σ^- stores an inner approximation of $\mathcal{K}(\mathbf{X}, \mathbb{T})$ as a union of interval vectors.

Algorithm 4.6: Controllable Sets Via Set Inversion

In: $\mathbb{T}, \mathbf{X}, \mathbf{U}, \varepsilon$; Out: Σ^-

1. Initialize **Stack** $1 = \mathbf{X}, \Sigma^- = \emptyset$;
2. while **Stack** $1 \neq \emptyset$
3.   Pop out a $\mathbf{X}_i$ from **Stack** 1;
4.   Compute $\mathbf{f}(\mathbf{X}_i, \mathbf{U})$;
5.   if $\mathbf{f}(\mathbf{X}_i, \mathbf{U}) \cap \mathbb{T} = \emptyset$
6.   Discard $\mathbf{X}_i$ and return to 2;
7.   endif
8.   Initialize **Stack** $2 = \mathbf{U}$;
9.   while **Stack** $2 \neq \emptyset$
10.   Pop out a $\mathbf{U}_j$ from **Stack** 2;
11.   Compute $\mathbf{f}(\mathbf{X}_i, \mathbf{U}_j)$;
12.   if $\mathbf{f}(\mathbf{X}_i, \mathbf{U}_j) \cap \mathbb{T} = \emptyset$
13.   Discard $\mathbf{U}_j$ and return to 9;
14.   elseif $\mathbf{f}(\mathbf{X}_i, \mathtt{Mid}(\mathbf{U}_j)) \subseteq \mathbb{T}$
15. $\Sigma^- = \mathbf{X}_i \cup \Sigma^-$ and return to 2;
16.   elseif $\mathtt{Width}(\mathbf{U}_j) \leq \varepsilon$, then discard $\mathbf{U}_j$ and return to 9;
17.   else
18.   Bisect $\mathbf{U}_j$ to $\mathtt{L}\mathbf{U}_j$,$\mathtt{R}\mathbf{U}_j$, push them on **Stack** 2 and return to 9;
19.   endif
20.   endwhile
21.   if $\mathtt{Width}(\mathbf{X}_i) \leq \varepsilon$, then $\Sigma^+ := \mathbf{X}_i \cup \Sigma^+$ and return to 2;
22.   else
23.   Bisect $\mathbf{X}_i$ to $\mathtt{L}\mathbf{X}_i$ and $\mathtt{R}\mathbf{X}_i$, push them on **Stack** 1 and return to 2;

```
24.    endif
25.  endwhile
```

As shown in Algorithm 4.6, if $\mathbf{f}(\mathbf{X}_i, \mathbf{U}) \cap \mathbb{T} = \emptyset$, then for all $\mathbf{u} \in \mathbf{U}$, it is impossible to drive the state $\mathbf{X}_i$ to the terminal control invariant set $\mathbb{T}$ at the next step, so $\mathbf{X}_i$ does not belong to the one-step controllable set and it is to be discarded in `Step` 6; however, only a part of $\mathbf{U}$ is tested in `Step` 12, i.e., $\mathbf{f}(\mathbf{X}_i, \mathbf{U}_j) \cap \mathbb{T} = \emptyset$, so for all $\mathbf{U}_j$, it is impossible to drive the state $\mathbf{X}_i$ to the terminal set $\mathbb{T}$ at the next step, then $\mathbf{U}_j$ is to be discarded instead in `Step` 13. On the contrary, if there exists a control vector $\texttt{Mid}(\mathbf{U}_j)$ that renders $\mathbf{f}(\mathbf{X}_i, \texttt{Mid}(\mathbf{U}_j)) \subseteq \mathbb{T}$, then the subbox $\mathbf{X}_i$ can be driven to the terminal control invariant set $\mathbb{T}$ via an admissible control vector $\texttt{Mid}(\mathbf{U}_j)$ at the next step, which signifies that $\mathbf{X}_i$ belongs to the one-step controllable set and it is to be stored in Σ^-. If no judgement can be made for $\mathbf{X}_i$ or $\mathbf{U}_j$ and the widths of them are beyond the given bound of error tolerance ε, just as in `Step` 16 and 21, they are to be discarded as well; otherwise, $\mathbf{X}_i$ or $\mathbf{U}_j$ is to be bisected further for a finer judgement, just as shown in `Step` 18 and 23. It is worthy to note that the bisection of the admissible state space and the bisection of the admissible control space are separated by two nested loops in Algorithm 4.6, which is different from the published algorithm in (Bravo et al., 2005) where the admissible state space and the admissible control space were combined to be bisected together and extra treatments of the subboxes of the admissible domain were needed after each bisection and selection of the admissible state space.

4.4.3 One-step Control via Global Optimization

Once all one-step controllable sets $\mathcal{K}_j(\mathbf{X}, \mathbb{T})(j = 1, \cdots, N)$ within the constrained state space have been obtained, the controllability of any initial state can be judged accordingly. Assume that $\mathbf{x}(0) \in \mathcal{K}_N(\mathbf{X}, \mathbb{T})$, i.e., the initial state is controllable to the designed terminal set $\mathbb{T}$ in finite steps, then the nonlinear contractive MPC with a contractive sequence of controllable sets is illustrated in Algorithm 4.7, where the solver of global optimization via interval analysis in Algorithm 4.2 is applied to solve the corresponding constrained nonlinear optimization problem.

`Algorithm 4.7:One-step Control Algorithm Via Controllable Sets`

`In:` $\mathbf{x}(0), \mathcal{K}_i(\mathbf{X}, \mathbb{T})$; `Out:` $\mathbf{u}^{Optimal}(k|k), \mathbf{x}(k)$

1. Get the current state $\mathbf{x}(k)$;
2. *if* $\mathbf{x}(k) \in \mathbb{T}$
3. Switch to the related local stabilizing feedback control law;
4. *else*
5. Find the i: $i = \min_{j=1,\cdots,N}\{\mathbf{x}(k) \in \mathcal{K}_j(\mathbf{X}, \mathbb{T})\}$;
6. Compute $\mathbf{u}^{Optimal}(k|k)$ with the contractive constraint $\mathbf{x}(k+1|k) \in \mathcal{K}_{i-1}(\mathbf{X}, \mathbb{T})$;
7. Apply $\mathbf{u}^{Optimal}(k|k)$ to the system;
8. *endif*
9. Return to 1 and repeat.

4.4.4 Example

The illustrative example considered is described by the following state-space model (Cannon et al., 2003):

$$\begin{cases} x_1(k+1) = x_1(k) + 0.1x_2(k) + 0.1[\mu + (1-\mu)x_1(k)]u(k) \\ x_2(k+1) = 0.1x_1(k) + x_2(k) + 0.1[\mu - 4(1-\mu)x_2(k)]u(k), \end{cases} \tag{4.28}$$

where $\mu = 0.9$, the control is constrained to $|u| \leq 2$ and the state variables are constrained to $|\mathbf{x}|_\infty \leq 2$. The terminal set $\mathbb{T}$ is designed to be a control invariant polytope along with a local stabilizing feedback control law $u = \mathbf{k}\mathbf{x}$, where $\mathbf{k} = [-1.2131 -1.2128]$ (Cannon et al., 2003):

$$\begin{bmatrix} 0.8190 & -1.9655 \\ -0.8199 & 1.9655 \\ 3.033 & 3.033 \\ -3.033 & -3.033 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{4.29}$$

The selected terminal set $\mathbb{T}$ along with the local stabilizing feedback control law $u = \mathbf{k}\mathbf{x}$ can be demonstrated geometrically to be control invariant by using the solver of set inversion via zonotope geometry, where every sub-zonotope is control invariant under the related local stabilizing feedback control law, just as shown in Fig. 4.12.

The inner approximation of the first-step controllable set $\mathcal{K}(\mathbf{X}, \mathbb{T})$ computed via the interval-based algorithm in Algorithm 4.6 is a union of interval vectors, just as shown in Fig. 4.13.

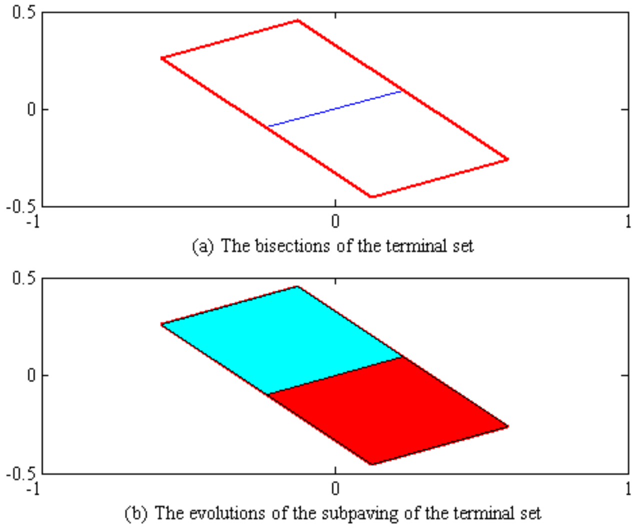


Figure 4.12: The geometrical demonstration of control invariance

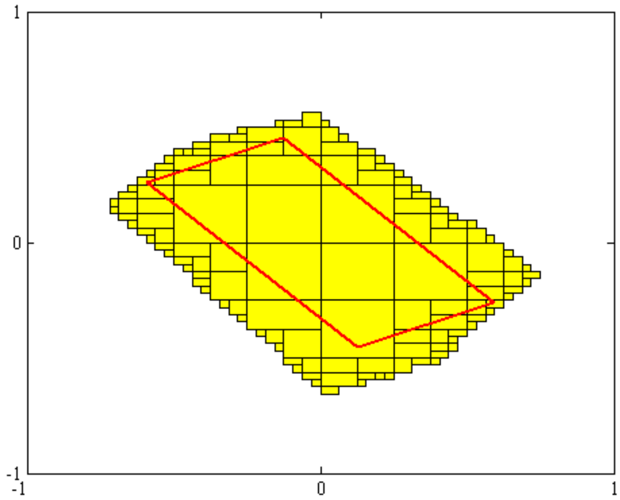


Figure 4.13: The first-step controllable set via classical interval analysis

The following-step controllable sets can be computed iteratively with the renewed terminal set $\mathbb{T}$ according to the interval-based algorithm in Algorithm 4.6 and the inner approximation of the maximal robust controllable set with the bound of error tolerance $\varepsilon = 0.05$ is reached at the 41th step, which is shown in Fig. 4.14.

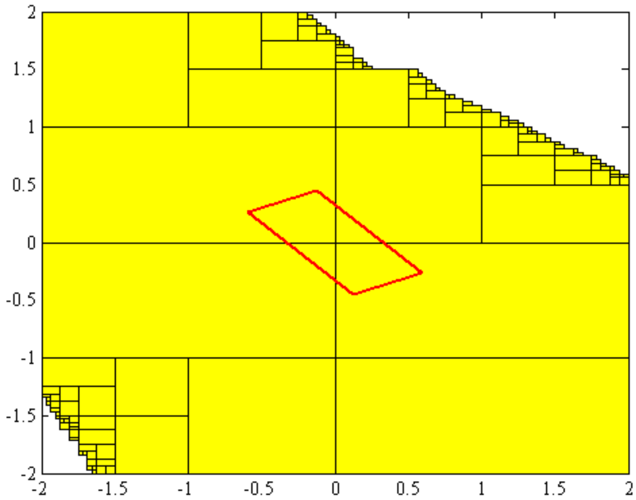


Figure 4.14: The inner approximation of the maximal controllable set

The one-step control is to drive the state contractively along the computed controllable sets to the selected terminal control invariant set $\mathbb{T}$ based on the control algorithm in Algorithm 4.6 and then the related local stabilizing feedback control law is applied instead to drive the state asymptotically to the origin. Assume that the initial state of the nonlinear system is at $\mathbf{x}(0) = (1.9, -1.5)$, the resulting control process of the dual-mode approach of nonlinear contractive MPC with a contractive sequence of controllable sets is shown in Fig. 4.15.

It can be seen from Fig. 4.15 that the one-step control approach based on the computed controllable sets as well as the strategy of contractive MPC with unit control horizon is guaranteed to be feasible and stable, i.e, the system can be driven contractively along the computed controllable sets to the selected terminal control invariant set.

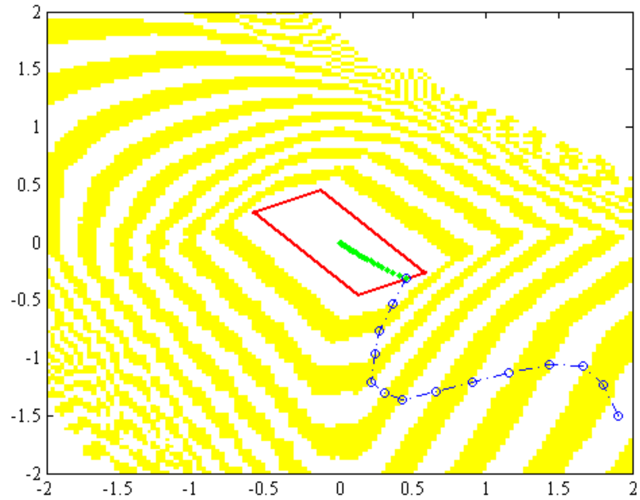


Figure 4.15: Nonlinear MPC with a contractive sequence of controllable sets

4.5 Summary

Classical interval analysis and zonotope geometry have been introduced in this chapter. The solver of set inversion via interval analysis has been extended to set inversion via zonotope geometry, which has been further applied to test set invariance in constrained control. The solver of global optimization for set inversion via zonotope geometry has also been proposed to design a control invariant zonotope of the maximal volume in a geometric way. The solver of set inversion via interval analysis has been applied to find a feasible control horizon and compute controllable sets in nonlinear contractive MPC with compulsory contractive constraint and nonlinear MPC with a contractive sequence of controllable sets, respectively. The solver of global optimization via interval analysis has been applied to compute optimal one-step control inputs in both nonlinear contractive MPC with compulsory contractive constraint and nonlinear contractive MPC with a contractive sequence of controllable sets.

Chapter 5

Nonlinear Robust Contractive MPC via Modal Interval Analysis

As addressed in Chapter 4, classical interval analysis can be applied to compute controllable sets of constrained nonlinear discrete-time systems, where the state space and the control space are described by corresponding classical interval vectors or boxes. However, these two interval vectors have different modalities: for a subbox of the admissible state space, it belongs to a controllable set if and only if for all states within the subbox there always exists an admissible control input that can drive them to the selected terminal set; for a subbox of the admissible control space, it is a feasible control domain if and only if there exists one value within the subbox which can drive the concerned subbox of the admissible state space to the selected terminal set. Classical interval analysis cannot distinguish these two types of physical interval vectors and denotes them uniformly. Additional local searches and skills are needed in the classical interval-based algorithm for computing controllable sets and thus the computation of controllable sets there is not in a directly semantic and strictly mathematical way. Furthermore, it would be more difficult to organize and interpret the computation of robust controllable sets for constrained nonlinear uncertain discrete-time systems semantically since another interval vector with its corresponding modality for characterizing the uncertainty is concerned as well.

This chapter applies an extended interval analysis called modal interval analysis to the computation of robust controllable sets for general constrained nonlinear uncertain discrete-time systems, where the interval vector for the admissible state space, the interval vector for the admissible control space and the interval vector for the uncertainty are all considered as modal interval vectors with corresponding modalities. The chapter is organized as follows: modal interval analysis along with its solvers of quantified set inversion and constrained minimax optimization is introduced in a comparative way relative to classical interval analysis in Section 5.1; the generalized solver of multi-dimensional quantified set inversion is applied to compute robust controllable sets and the proposed solver of constrained minimax optimization is applied to compute one-step control inputs for nonlinear robust contractive MPC of general constrained nonlinear uncertain discrete-time systems in Section 5.2.

5.1 Modal Interval Analysis

Modal interval analysis is an extension of classical interval analysis obtained by differentiating the existential and universal modalities of physical intervals encountered in practical problems. The following subsections give a comprehensive and tutorial introduction of modal interval analysis in a comparative way relative to classical interval analysis and the solvers of quantified set inversion and constrained minimax optimization, where modal interval analysis is treated as an extension of classical interval analysis in modality, inclusion, semantics and rational.

5.1.1 The Initiative of Modal Intervals — Modal Extension

Physical intervals encountered in practical problems have two modalities: there exists a value in $[a, b](a \leq b)$ that possesses a property or some properties concerned and all values in $[a, b](a \leq b)$ possess a property or some properties concerned. Classical interval analysis cannot distinguish these two types of physical intervals and denotes them as $[a, b](a \leq b)$ uniformly (Jaulin et al., 2001). Modal interval analysis does distinguish these two types of physical intervals and denotes them differently: $[a, b]^*(a \leq b)$ for those proper intervals that only require the existence of a value in the domain of $a \leq x \leq b$ to possess a property or some properties concerned; and $[b, a]^*(a \leq b)$ for those improper intervals that require all values in the domain of $a \leq x \leq b$ to possess a property or some properties concerned, where $[a, b]^*$ and $[b, a]^*$ are denoted to modal intervals[1]. The concept of modal intervals is similar to the concept of objects in C++ programming since a modal interval contains not only a purely numerical interval, but also a physical property of the numerical interval. A modal interval $[a,b]^*$ is therefore a pair of a classical interval and a corresponding modality (Gardenes et al., 2001), i.e.,

$$[a,b]^* := (\texttt{Prop}([a,b]^*), \texttt{Q}([a,b]^*)) = \begin{cases} ([a,b],\exists) \; if \; a \le b \\ ([b,a],\forall) \; if \; a \ge b, \end{cases} \tag{5.1}$$

where $\texttt{Prop}([a,b]^*) = [\min\{a,b\}, \max\{a,b\}]$ is the classical interval domain and $\texttt{Q}([a,b]^*) \in \{\forall, \exists\}$ is the modality of $[a,b]^*$. The lower and upper bounds of $[a,b]^*$ are denoted to be $\texttt{Lb}([a,b]^*) = a$ and $\texttt{Ub}([a,b]^*) = b$, respectively. The operator $\texttt{Dual}$ is to change the modality of a modal interval, i.e., $\texttt{Dual}([a,b]^*) = [b,a]^*$. It is interesting to note that classical intervals are often treated implicitly as proper intervals with the modality of $\exists$ in classical interval analysis before the concept of modal intervals was proposed.

Accordingly, the modal interval vector $\mathbf{X}^* \in \mathbb{I}^*(\mathbb{R}^n)$ is denoted to a vector whose components are all modal intervals, where $\mathbb{I}^*(\mathbb{R}^n)$ is denoted to the set of all n-dimensional modal interval vectors. $\mathbf{X}^*$ is usually divided into two sub-vectors according to the modality of every component:

$$\mathbf{X}^* = [\mathbf{X}_p^*, \mathbf{X}_i^*], \tag{5.2}$$

where $\mathbf{X}_p^*$ is composed of proper intervals with the modality $\exists$ and $\mathbf{X}_i^*$ is composed of improper intervals with the modality $\forall$. The basic operation of bisecting a classical interval vector can also be extended to modal interval vectors. Taking the modal interval vector $\mathbf{X}^* = [a_1,b_1]^* \times \cdots \times [a_n,b_n]^*$ as an example, its width is denoted to be:

$$\texttt{Width}(\mathbf{X}^*) = \max_{i=1,\cdots,n} |a_i - b_i|, \tag{5.3}$$

and the index j is denoted to be:

$$j = \min_{i=1,\cdots,n} \{i | (|a_i - b_i|) = \texttt{Width}(\mathbf{X}^*)\}, \tag{5.4}$$

then the bisection $\texttt{Bisect}(\mathbf{X}^*)$ returns two sub-modal interval vectors $\texttt{L}\mathbf{X}^*$ and $\texttt{R}\mathbf{X}^*$:

$$\begin{cases} \texttt{L}\mathbf{X}^* := [a_1,b_1]^* \times \cdots \times [a_j, \frac{(a_j+b_j)}{2}]^* \times \cdots \times [a_n,b_n]^* \\ \texttt{R}\mathbf{X}^* := [a_1,b_1]^* \times \cdots \times [\frac{(a_j+b_j)}{2}, b_j]^* \times \cdots \times [a_n,b_n]^*. \end{cases} \tag{5.5}$$

[1]The notation for modal intervals in this thesis is different from the seminal paper on modal intervals (Gardenes et al., 2001) and thus it only reflects the bias of the author for the introduction of modal intervals and their relationships with classical intervals.

5.1.2 The Quantifier of Modal Intervals — Inclusion Extension

The modality of a modal interval $[a, b]^*$ provides the quantifier for all properties possessed by the modal interval: for a proper interval, the quantifier is $\exists x \in \texttt{Prop}([a, b]^*)$; and for an improper interval, the quantifier is $\forall x \in \texttt{Prop}([a, b]^*)$. The inclusion between classical intervals $[a, b] \subseteq [c, d] (a \leq b, c \leq d)$ can be interpreted as: if there exists a value $x \in [a, b]$ that possesses a property or some properties concerned, then there also exists a value $y \in [c, d]$ that possesses the same property (properties). Accordingly, the inclusion between modal intervals $[a, b]^* \subseteq [c, d]^*$ can be extended to be (Gardenes et al., 2001):

$$[a, b]^* \subseteq [c, d]^* \Leftrightarrow \texttt{Pred}([a, b]^*) \subseteq \texttt{Pred}([c, d]^*), \tag{5.6}$$

where $\texttt{Pred}([a, b]^*)$ and $\texttt{Pred}([c, d]^*)$ are denoted to the sets of all properties possessed by $[a, b]^*$ and $[c, d]^*$ with their corresponding quantifiers, respectively. The extended modal interval inclusion of (5.6) can be further simplified as (Gardenes et al., 2001):

$$[a, b]^* \subseteq [c, d]^* \Leftrightarrow (a \geq c, b \leq d). \tag{5.7}$$

This can be demonstrated by considering the quantifiers of modal intervals and the definition of inclusion in (5.6). For example, for the case of $a \leq b, c \leq d$, i.e., their modalities are both $\exists$, it is natural that $[a, b]^* \subseteq [c, d]^*$ stands for $a \geq c, b \leq d$ since if there exists a value possessing a property or some properties in the domain of $a \leq x \leq b$, there of course exists such a value possessing the same property(properties) in a bigger domain $c \leq y \leq d$, e.g., $[2, 3]^* \subseteq [1, 5]^*$; for the case of $a \geq b, c \geq d$, i.e., their modalities are both $\forall$, it is also natural that $[a, b]^* \subseteq [c, d]^*$ stands for $a \geq c, b \leq d$ since if all values in the domain of $b \leq x \leq a$ possess a property or some properties, then obviously all values in a smaller domain $d \leq x \leq c$ possess the same property(properties), e.g., $[5, 1]^* \subseteq [3, 2]^*$; for the case of $a \leq b, c \geq d$, i.e., $\texttt{Q}([a, b]^*) = \exists$ and $\texttt{Q}([c, d]^*) = \forall$, which means that all values in the domain of $d \leq y \leq c$ possess the property(properties) possessed by an existential value in the domain of $a \leq x \leq b$, then $a = b = c = d$ must be satisfied and thus $a \geq c, b \leq d$ is also satisfied; and finally, for the case of $a \geq b, c \leq d$, i.e., $\texttt{Q}([a, b]^*) = \forall$ and $\texttt{Q}([c, d]^*) = \exists$, which means that there exists a value in the domain of $c \leq y \leq d$ possessing the property(properties) possessed by all values in the domain of $b \leq x \leq a$, then $[b, a] \cap [c, d] \neq \emptyset$ must be satisfied and thus

$a \geq c, b \leq d$ is satisfied as well.

The inclusion conditions between modal interval vectors can also be deduced according to the definition of inclusion in (5.6) for each component of modal interval vectors. Taking the 2-dimensional modal interval vectors $[a,b]^* \times [c,d]^* \subseteq [e,f]^* \times [g,h]^*$ as an example, assume that $[e,f]^*$ and $[g,h]^*$ are both proper modal intervals, then the inclusion relationships between $[a,b]^* \times [c,d]^*$ and $[e,f]^* \times [g,h]^*$ can be classified into four cases according to the modalities of $[a,b]^*$ and $[c,d]^*$, i.e., $\mathtt{Q}([a,b]^*) = \exists$ and $\mathtt{Q}([c,d]^*) = \exists$; $\mathtt{Q}([a,b]^*) = \exists$ and $\mathtt{Q}([c,d]^*) = \forall$; $\mathtt{Q}([a,b]^*) = \forall$ and $\mathtt{Q}([c,d]^*) = \exists$; $\mathtt{Q}([a,b]^*) = \forall$ and $\mathtt{Q}([c,d]^*) = \forall$. These four cases are shown in Fig. 5.1(a), (b), (c) and (d), respectively, where the white boxes denote the classic interval vector $\mathtt{Prop}([a,b]^*) \times \mathtt{Prop}([c,d]^*)$, the yellow boxes denote the classic interval vector $\mathtt{Prop}([e,f]^*) \times \mathtt{Prop}([g,h]^*)$ and the red boxes denote their intersections.

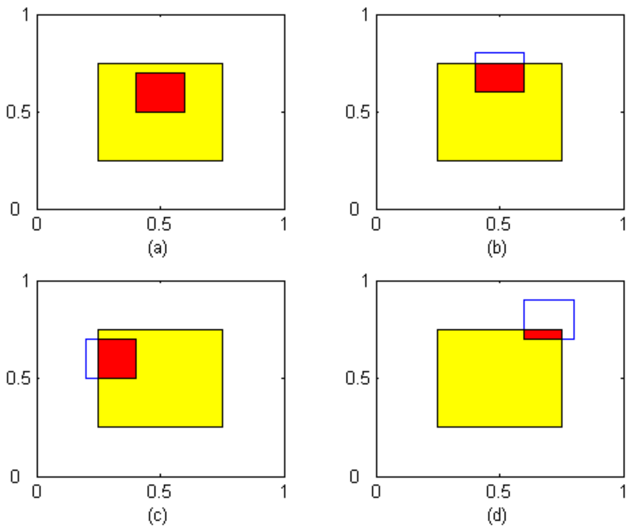


Figure 5.1: The inclusion between two 2-dimensional modal interval vectors

Similar to classical intervals, the inner and outer roundings of $[a,b]^* \in \mathbb{I}^*(\mathbb{R})$ for a given scale are defined to be:

$$\begin{cases} \mathtt{Inn}([a,b]^*) = [\mathtt{Right}(a), \mathtt{Left}(b)]^* \\ \mathtt{Out}([a,b]^*) = [\mathtt{Left}(a), \mathtt{Right}(b)]^*, \end{cases} \tag{5.8}$$

where $\texttt{Left}(x) \leq x, \texttt{Right}(x) \geq x, x \in \mathbb{R}$. It is obvious that

$$\texttt{Inn}([a,b]^*) = \texttt{Dual}(\texttt{Out}(\texttt{Dual}([a,b]^*))), \tag{5.9}$$

and thus the implementation of the inner rounding can be realized by the outer rounding as well. According to the rule of inclusion in (5.7), the following relationship

$$\texttt{Inn}([a,b]^*) \subseteq [a,b]^* \subseteq \texttt{Out}([a,b]^*) \tag{5.10}$$

is always satisfied. However, the same relationship $\texttt{Inn}([a,b]) \subseteq [a,b] \subseteq \texttt{Out}([a,b])$ in classical interval analysis is not always satisfied since $\texttt{Inn}([a,b])$ is not always a valid classical interval, e.g., $\texttt{Inn}([1.346, 1.347])$ for two-decimal digits is $[1.35, 1.34]$ while $[1.35, 1.34]$ is not a valid classical interval (Gardenes et al., 2001).

5.1.3 The Function of Modal Intervals — Semantic Extension

Comparative to the interval function $f(\mathbf{X}) = [\min_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x}), \max_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x})], \mathbf{X} \in \mathbb{I}(\mathbb{R}^n)$ defined in classical interval analysis, the modal interval function $f^*(\mathbf{X}^*) : \mathbb{I}^*(\mathbb{R}^n) \to \mathbb{I}^*(\mathbb{R})$ is defined to be (Gardenes et al., 2001):

$$f^*(\mathbf{X}^*) := [\min_{\mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*)} \max_{\mathbf{x}_i \in \texttt{Prop}(\mathbf{X}_i^*)} f(\mathbf{x}_p, \mathbf{x}_i), \max_{\mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*)} \min_{\mathbf{x}_i \in \texttt{Prop}(\mathbf{X}_i^*)} f(\mathbf{x}_p, \mathbf{x}_i)]^*. \tag{5.11}$$

Its counterpart modal interval function $f^{**}(\mathbf{X}^*)$ is defined to be:

$$f^{**}(\mathbf{X}^*) := [\max_{\mathbf{x}_i \in \texttt{Prop}(\mathbf{X}_i^*)} \min_{\mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*)} f(\mathbf{x}_p, \mathbf{x}_i), \min_{\mathbf{x}_i \in \texttt{Prop}(\mathbf{X}_i^*)} \max_{\mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*)} f(\mathbf{x}_p, \mathbf{x}_i)]^*. \tag{5.12}$$

Naturally, $f^*(\mathbf{X}^*)$ degenerates to $[\min_{\mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*)} f(\mathbf{x}_p), \max_{\mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*)} f(\mathbf{x}_p)]^*$ when all components of $\mathbf{X}^*$ are proper intervals, which is the case of $f(\mathbf{X})$ in classical interval analysis. Furthermore, $f^*(\mathbf{X}^*) = f^{**}(\mathbf{X}^*)$ when all components of $\mathbf{X}^*$ have the same modality $\exists$ or $\forall$.

The semantic statement of (4.1) for $f(\mathbf{X}) \subseteq F(\mathbf{X})$ in classical interval analysis can be extended accordingly for $f^*(\mathbf{X}^*) \subseteq F^*(\mathbf{X}^*)$ in modal interval analysis, i.e., $f^*(\mathbf{X}^*) \subseteq F^*(\mathbf{X}^*)$ is equal to (Gardenes et al., 2001):

$$\forall \mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*) \texttt{Q}(F^*(\mathbf{X}^*)) z \in \texttt{Prop}(F^*(\mathbf{X}^*)) \exists \mathbf{x}_i \in \texttt{Prop}(\mathbf{X}_i^*)\ z = f(\mathbf{x}_p, \mathbf{x}_i). \tag{5.13}$$

This is the fundamental semantic theorem in modal interval analysis since it provides a semantic and physical interpretation for the modal interval function $f^*(\mathbf{X}^*)$ and its inclusion function $F^*(\mathbf{X}^*)$ as well. The semantic theorem can be proved as follows (Gardenes et al., 2001):

$$f^*(\mathbf{X}^*) \subseteq F^*(\mathbf{X}^*) \Leftrightarrow \forall \mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*) f^*(\mathbf{x}_p, \mathbf{X}_i^*) \subseteq F^*(\mathbf{X}^*)$$
$$\Leftrightarrow \forall \mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*) \begin{cases} \text{if } F^*(\mathbf{X}^*) \text{ is proper, then } \texttt{Prop}(f^*(\mathbf{x}_p, \mathbf{X}_i^*)) \cap \texttt{Prop}(F^*(\mathbf{X}^*)) \neq \emptyset \\ \text{if } F^*(\mathbf{X}^*) \text{ is improper, then } \texttt{Prop}(F^*(\mathbf{X}^*)) \subseteq \texttt{Prop}(f^*(\mathbf{x}_p, \mathbf{X}_i^*)) \end{cases}$$
$$\Leftrightarrow \forall \mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*) \texttt{Q}(F^*(\mathbf{X}^*)) z \in \texttt{Prop}(F^*(\mathbf{X}^*)) z \in \texttt{Prop}(f^*(\mathbf{x}_p, \mathbf{X}_i^*))$$
$$\Leftrightarrow \forall \mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*) \texttt{Q}(F^*(\mathbf{X}^*)) z \in \texttt{Prop}(F^*(\mathbf{X}^*)) \exists \mathbf{x}_i \in \texttt{Prop}(\mathbf{X}_i^*)\ z = f(\mathbf{x}_p, \mathbf{x}_i).$$

It can be seen that the semantic statement of (5.13) in modal interval analysis introduces both the quantifier $\forall \mathbf{x}_p \in \texttt{Prop}(\mathbf{X}_p^*)$ and the quantifier $\exists \mathbf{x}_i \in \texttt{Prop}(\mathbf{X}_i^*)$ for function variables while the semantic statement of (4.1) in classical interval analysis only contains the quantifier $\forall \mathbf{x} \in \mathbf{X}$ for all function variables. Such a semantic extension is essential and complete in theory since many physical problems includes both $\forall$ and $\exists$ for function variables.

5.1.4 The Approximation of $f^*(\mathbf{X}^*)$ — Rational Extension

Similar to the interval function $f(\mathbf{X})$ in classical interval analysis, the modal interval function $f^*(\mathbf{X}^*)$ can seldom be computed arithmetically since its bounds concern complex minimax and maximin optimization problems. The inclusion function $F(\mathbf{X})$ computed directly through interval arithmetic is used instead in various solvers of classical interval analysis to provide lower and upper bounds of the original interval function $f(\mathbf{X})$. Such a strategy can be extended to the computation of the modal interval function $f^*(\mathbf{X}^*)$ as well. Modal interval arithmetic is defined to compute an outer approximation $f^*(\mathbf{X}^*) \subseteq \texttt{Outer}(f^*(\mathbf{X}^*))$ as well as an inner approximation $\texttt{Inner}(f^*(\mathbf{X}^*)) \subseteq f^*(\mathbf{X}^*)$ arithmetically and directly while the semantic theorem of (5.13) is applied to interpret such approximations. The outer approximation $\texttt{Outer}(f^*(\mathbf{X}^*))$ of $f^*(\mathbf{X}^*)$ can also be regarded as an inclusion function $F^*(\mathbf{X}^*)$ of $f^*(\mathbf{X}^*)$. In the following paragraphs, the rational extension for computing the inner and outer approximations of the modal interval function $f^*(\mathbf{X}^*)$ arithmetically is to be explained in detail.

Four elementary arithmetic operations $(+, -, \times, \div)$ between two modal intervals $[a, b]^*$ and $[c, d]^*$ can be computed according to the definition of (5.11) or (5.12), i.e., $f^*([a,b]^*,[c,d]^*) = [a,b]^* \diamond [c,d]^* (\diamond = +,-,\times,\div)$ or $f^{**}([a,b]^*,[c,d]^*) = [a,b]^* \diamond [c,d]^* (\diamond = +,-,\times,\div)$. It turns out that these two arithmetic extensions are equal for $(+,-,\times,\div)$ and coincide with Kaucher's complete arithmetic for classical intervals (Kaucher, 1980). Such an arithmetic is adopted as modal interval arithmetic and the semantic theorem of (5.13) is maintained to interpret the computation results of such modal interval arithmetic.

Except for four elementary arithmetic operations, other operators between modal intervals or one-variable operators for a modal interval can be computed according to the definition of (5.11) or (5.12) as well. A general modal rational function can also be computed by decomposing it into basic operations according to the syntactic tree of the expression defining the function once all basic operators have been defined according to the definition of (5.11) or (5.12). This is similar to the computation of natural inclusion functions in classical interval analysis. Concretely, the modal rational *-extension function $fR^*(\mathbf{X}^*)$ is defined to be the function from $\mathbb{I}^*(\mathbb{R}^n)$ to $\mathbb{I}^*(\mathbb{R})$ determined by the syntactic tree of the expression defining the function when all the operators concerned are computed according to the definition of (5.11); the modal rational **-extension function $fR^{**}(\mathbf{X}^*)$ is defined to be the function from $\mathbb{I}^*(\mathbb{R}^n)$ to $\mathbb{I}^*(\mathbb{R})$ determined by the syntactic tree of the expression defining the function when all the operators concerned are computed according to the definition of (5.12); the modal rational function $fR(\mathbf{X}^*)$ is the function alike $fR^*(\mathbf{X}^*)$ or $fR^{**}(\mathbf{X}^*)$ when all its operators concerned are equal according to the definitions of (5.11) and (5.12), i.e., $fR(\mathbf{X}^*) = fR^*(\mathbf{X}^*) = fR^{**}(\mathbf{X}^*)$. Considering the modality, the frequency of occurrence and the monotonicity of every component of $\mathbf{X}^*$ in $f^*(\mathbf{X}^*)$, the further interpretable relationship between $f^*(\mathbf{X}^*)$ and $fR(\chi^*)$ exists, i.e.,

$$f^*(X^*) \subseteq fR(\chi^*), \tag{5.14}$$

where χ^* is transformed from $\mathbf{X}^*$ according to the properties of its components. Then the generally uncomputable $f^*(\mathbf{X}^*)$ can be approximated by the arithmetically computable $fR(\chi^*)$ with the semantic interpretation of (5.13).

Approximations of $f^*(\mathbf{X}^*)$ obtained only by (5.14) are generally over-estimated due to the potential multi-occurrence of some components of $\mathbf{X}^*$. The technique of the branch-and-bound was introduced to compute more precise inner and outer approximations of $f^*(\mathbf{X}^*)$ (Herrero et al., 2005):

$$[\min_{j=\{1,\cdots,m\}} \max_{k=\{1,\cdots,n\}} a_{jk}, \max_{j=\{1,\cdots,m\}} \min_{k=\{1,\cdots,n\}} b_{jk}]^* \subseteq f^*(\mathbf{X}^*), \tag{5.15}$$

$$[\min_{j=\{1,\cdots,m\}} \max_{k=\{1,\cdots,n\}} c_{jk}, \max_{j=\{1,\cdots,m\}} \min_{k=\{1,\cdots,n\}} d_{jk}]^* \supseteq f^*(\mathbf{X}^*), \tag{5.16}$$

where $[a_{jk}, b_{jk}]^* = \mathtt{Inn}(fR(\mathbf{x}^*_{p_j}, \mathbf{X}^*_{i_k}))$; $[c_{jk}, d_{jk}]^* = \mathtt{Out}(fR(\mathbf{X}^*_{p_j}, \mathbf{x}^*_{i_k}))$; $\{\mathbf{X}^*_{p_1}, \cdots, \mathbf{X}^*_{p_m}\}$ is a partition of $\mathbf{X}^*_p$ and $\mathbf{x}^*_{p_j} \in \mathtt{Prop}(\mathbf{X}^*_{p_j})$ for $j = \{1, \cdots, m\}$; $\{\mathbf{X}^*_{i_1}, \cdots, \mathbf{X}^*_{i_n}\}$ is a partition of $\mathbf{X}^*_i$ and $\mathbf{x}^*_{i_k} \in \mathtt{Prop}(\mathbf{X}^*_{i_k})$ for $k = \{1, \cdots, n\}$; $\mathtt{Inn}(fR(\mathbf{x}^*_{p_j}, \mathbf{X}^*_{i_k}))$ is obtained through inner roundings of every modal interval and every operator concerned when computing $fR(\mathbf{x}^*_{p_j}, \mathbf{X}^*_{i_k})$; and $\mathtt{Out}(fR(\mathbf{X}^*_{p_j}, \mathbf{x}^*_{i_k}))$ is obtained through outer roundings of every modal interval and every operator concerned when computing $fR(\mathbf{X}^*_{p_j}, \mathbf{x}^*_{i_k})$ (Herrero et al., 2005). The precision of the inner and outer approximations of $f^*(\mathbf{X}^*)$ is closely related to the partitions of $\mathbf{X}^*_p$ and $\mathbf{X}^*_i$. Obviously, better approximations of $f^*(\mathbf{X}^*)$ can be obtained with finer partitions of $\mathbf{X}^*_p$ and $\mathbf{X}^*_i$.

5.1.5 The Solver of Quantified Set Inversion

In contrast to the computation of $f^*(\mathbf{X}^*)$ with the known domain $\mathbf{X}^*$, the opposite problem is to find the feasible solution in $\mathbf{X}^*$ with the known codomain $[a, b]^*$, i.e., to find all $\mathcal{X}^* = [\mathcal{X}^*_p, \mathcal{X}^*_i]$ within an initial admissible domain $\mathbf{X}^*$ that satisfies

$$\forall \mathbf{x}_p \in \mathtt{Prop}(\mathcal{X}^*_p) \mathtt{Q}([a, b]^*) z \in \mathtt{Prop}([a, b]^*) \exists \mathbf{x}_i \in \mathtt{Prop}(\mathcal{X}^*_i)\ z = f(\mathbf{x}_p, \mathbf{x}_i). \tag{5.17}$$

When the modality of all components of $\mathbf{X}^*$ is $\exists$, this problem degenerates to be a traditional set inversion problem discussed in classical interval analysis; when the modality of all components of $\mathbf{X}^*$ is not uniformly $\exists$, this problem generalizes to be a quantified set inversion problem discussed in modal interval analysis (Herrero et al., 2005). Many physical problems can be formulated mathematically as a quantified constraint satisfaction problem and then the solver of quantified set inversion can be applied to solve them. For example, to find the one-step robust controllable set $\tilde{\mathcal{K}}(\mathbf{X}, \mathbb{T})$ is a typical quantified constraint satisfaction problem and then $\tilde{\mathcal{K}}(\mathbf{X}, \mathbb{T})$ can be obtained numerically via the solver of quantified set inversion. Concretely, the problem of finding the subset $\Sigma_{\mathbf{x}} \subseteq \mathtt{Prop}(\mathbf{X}^*) \subseteq \mathbb{I}(\mathbb{R}^n)$ within which there exists a control input

$\mathbf{u} \in \texttt{Prop}(\mathbf{U}^*) \subseteq \mathbb{I}(\mathbb{R}^m)$ to render the scalar function $f(\mathbf{x}, \mathbf{w}, \mathbf{u}) \in \texttt{Prop}([a,b]^*)$ for all uncertain cases $\mathbf{w} \in \texttt{Prop}(\mathbf{W}^*) \subseteq \mathbb{I}(\mathbb{R}^l)$ can be solved using the solver of 1-dimensional quantified set inversion with the function $f^*(\mathbf{X}_p^*, \mathbf{W}_p^*, \mathbf{U}_i^*)$, where $\mathbf{X}_p^*$ is a proper interval with $\mathbf{X}_p^* = \mathbf{X}^*$, $\mathbf{W}_p^*$ is also a proper interval with $\mathbf{W}_p^* = \mathbf{W}^*$, $\mathbf{U}_i^*$ is an improper interval with $\mathbf{U}_i^* = \mathbf{U}^*$ and the scalar function $f(\mathbf{x}, \mathbf{w}, \mathbf{u})$ is usually to describe the selected terminal set $\mathbb{T}$ as an ellipsoid. The solver of 1-dimensional quantified set inversion for finding the inner and outer approximations of $\Sigma_\mathbf{x}$ with a given bound of error tolerance ε is illustrated in Algorithm 5.1.

`Algorithm 5.1:  1-Dimensional Quantified Set Inversion`

`In:` $[a,b]^*, \mathbf{X}_p^*, \varepsilon$; `Out:` Σ^-, Σ^+

1. `Initialize` **Stack** $= \mathbf{X}_p^*$ `and` $\Sigma^- = \Sigma^+ = \emptyset$;
2. `while` **Stack** $\neq \emptyset$
3.   `Pop out a` $\mathcal{X}_p^*$ `from` **Stack**;
4.   `Compute` $\texttt{Inner}(f^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*)) \subseteq f^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*)$;
5.   `if` $\texttt{Inner}(f^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*)) \nsubseteq [a,b]^*$, `then discard` $\mathcal{X}_p^*$ `and return to 2`;
6.   `Compute` $\texttt{Outer}(f^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \mathbf{U}_i^*))$ `where` $f^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \mathbf{U}_i^*) \subseteq \texttt{Outer}(f^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \mathbf{U}_i^*))$;
7.   `if` $\texttt{Outer}(f^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \mathbf{U}_i^*)) \subseteq [a,b]^*$
8. $\Sigma^- := \mathcal{X}_p^* \cup \Sigma^-$;
9.   `elseif` $\texttt{Width}(\mathcal{X}_p^*) \leq \varepsilon$
10. $\Sigma^+ := \mathcal{X}_p^* \cup \Sigma^+$;
11.   `else`
12.   `Bisect` $\mathcal{X}_p^*$ `to` $\texttt{L}\mathcal{X}_p^*, \texttt{R}\mathcal{X}_p^*$ `and push them on` **Stack**;
13.   `endif`
14. `endwhile`

As shown in Algorithm 5.1, $\mathbf{X}_p^*$ is the initial search domain for $\Sigma_\mathbf{x}$; ε is the bound of error tolerance beyond which modal interval vectors are not to be bisected further; Σ^- and Σ^+ are to store the inner approximation and the neighboring set to the solution set $\Sigma_\mathbf{x}$ with $\texttt{Width}(\mathcal{X}_p^*) \leq \varepsilon$, respectively. The inner approximation of $f^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*)$ and the outer approximation of $f^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \mathbf{U}_i^*)$ are computed in `Step` 4 and `Step` 6 according to (5.15) and (5.16), respectively; if $\texttt{Inner}(f^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*)) \nsubseteq [a,b]^*$, then $f^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*) \nsubseteq [a,b]^*$ as well, which signifies $\exists \mathbf{w} \in$

$\mathtt{Prop}(\mathbf{W}_p^*)\mathtt{Q}([b,a]^*)z \in \mathtt{Prop}([a,b]^*)\forall\mathbf{x} \in \mathtt{Prop}(\mathtt{Dual}(\mathcal{X}_p^*))\ \forall\mathbf{u} \in \mathtt{Prop}(\mathbf{U}_i^*)\ z \neq f(\mathbf{x},\mathbf{w},\mathbf{u})$ according to the opposite statement of the semantic theorem in (5.13), thus $\mathcal{X}_p^*$ does not belong to the solution set $\Sigma_\mathbf{x}$ and it is to be discarded in `Step` 5; if $\mathtt{Outer}(f^*(\mathcal{X}_p^*,\mathbf{W}_p^*,\mathbf{U}_i^*)) \subseteq [a,b]^*$, then $f^*(\mathcal{X}_p^*,\mathbf{W}_p^*,\mathbf{U}_i^*) \subseteq [a,b]^*$, which signifies $\forall\mathbf{x} \in \mathtt{Prop}(\mathcal{X}_p^*)\forall\mathbf{w} \in \mathtt{Prop}(\mathbf{W}_p^*)$ $\mathtt{Q}([a,b]^*)z \in \mathtt{Prop}([a,b]^*)\exists\mathbf{u} \in \mathtt{Prop}(\mathbf{U}_i^*)\ z = f(\mathbf{x},\mathbf{w},\mathbf{u})$ according to the semantic theorem in (5.13), thus $\mathcal{X}_p^*$ belongs to the solution set $\Sigma_\mathbf{x}$ and it is to be added into the inner approximation Σ^- of the solution set in `Step` 8; if no judgement can be made in either `Step` 5 or `Step` 7, and the width of the modal interval vector $\mathcal{X}_p^*$ is not bigger than the bound of error tolerance ε, it is to be added into the boundary Σ^+ of the solution set in `Step` 10; and finally, if no judgement can be made in either `Step` 5 or `Step` 7, and $\mathtt{Width}(\mathcal{X}_p^*)$ is bigger than the bound of error tolerance ε, it is to be bisected into two sub-modal interval vectors $\mathtt{L}\mathcal{X}_p^*, \mathtt{R}\mathcal{X}_p^*$ in `Step` 12 and both of them are to be pushed on **Stack** for further judgement.

The solver of 1-dimensional quantified set inversion in Algorithm 5.1 can be extended to $\mathbf{f}^*(\mathbf{X}^*) : \mathbb{I}^*(\mathbb{R}^n) \to \mathbb{I}^*(\mathbb{R}^m)$, where corresponding inclusion and exclusion judgements between modal interval vectors are considered instead. Taking the 2-dimensional quantified set inversion problem as an example, the known codomain of the 2-dimensional function $\mathbf{f}(\mathbf{x},\mathbf{w},\mathbf{u}) = f_1(\mathbf{x},\mathbf{w},\mathbf{u}) \times f_2(\mathbf{x},\mathbf{w},\mathbf{u})$ is assumed to be $[a,b]^* \times [c,d]^*$, where $[a,b]^*$ and $[c,d]^*$ are also assumed to be proper modal intervals. The problem is still to find the subset $\Sigma_\mathbf{x} \subseteq \mathtt{Prop}(\mathbf{X}^*) \subseteq \mathbb{I}(\mathbb{R}^n)$ within which there exists a control input $\mathbf{u} \in \mathtt{Prop}(\mathbf{U}^*) \subseteq \mathbb{I}(\mathbb{R}^m)$ to render the 2-dimensional function $\mathbf{f}(\mathbf{x},\mathbf{w},\mathbf{u}) \in \mathtt{Prop}([a,b]^*) \times \mathtt{Prop}([c,d]^*)$ for all uncertain cases $\mathbf{w} \in \mathtt{Prop}(\mathbf{W}^*) \subseteq \mathbb{I}(\mathbb{R}^l)$. The extended solver of 2-dimensional quantified set inversion for finding the inner and outer approximations of $\Sigma_\mathbf{x}$ with a given bound of error tolerance ε is illustrated in Algorithm 5.2. Since the affirmative semantics for the solution set is the combination of the semantics for $f_1^* \subseteq [a,b]^*$ and $f_2^* \subseteq [c,d]^*$, thus $\mathtt{Dual}(\mathcal{U}_i^*)$ should occur in one of f_1^* and f_2^* to ensure that there exists the same $\mathbf{u} \in \mathtt{Prop}(\mathcal{U}_i^*)$ to render $f_1^* \times f_2^* \subseteq [a,b]^* \times [c,d]^*$ in `Step` 16 and `Step` 20. To require all $\mathbf{u} \in \mathtt{Prop}(\mathcal{U}_i^*)$ to satisfy $f_1^*(\mathcal{X}_p^*,\mathbf{W}_p^*,\mathtt{Dual}(\mathcal{U}_i^*)) \subseteq [a,b]^*$ or $f_2^*(\mathcal{X}_p^*,\mathbf{W}_p^*,\mathtt{Dual}(\mathcal{U}_i^*)) \subseteq [c,d]^*$ is strict and thus $\mathcal{U}_i^*$ needs to be bisected in `Step` 25. Since $f_1^* \times f_2^* \nsubseteq [a,b]^* \times [c,d]^*$ is equal to $f_1^* \nsubseteq [a,b]^*$ or $f_2^* \nsubseteq [c,d]^*$, then the negative semantics for discarding modal interval vectors is the direct combination of $f_1^* \nsubseteq [a,b]^*$ and $f_2^* \nsubseteq [c,d]^*$. However, the entire domain of $\mathbf{U}_i^*$ is tested in `Step` 5 and thus the corresponding $\mathcal{X}_p^*$ is to be discarded in `Step` 6 while only a part of $\mathbf{U}_i^*$ is tested in `Step` 12 and thus only that part of $\mathbf{U}_i^*$ is discarded instead in `Step` 13. It is worthy to note that the principle and the structure of multi-dimensional quantified set inversion algorithm is similar to the 2-dimensional case while only inclusion and exclusion judgements are performed between higher-dimensional modal interval vectors.

`Algorithm 5.2:2-Dimensional Quantified Set Inversion`

`In:` $[a,b]^* \times [c,d]^*, \mathbf{X}_p^*, \mathbf{U}_i^*, \varepsilon$; `Out:` Σ^-, Σ^+

1. `Initialize` **Stack** $1 = \mathbf{X}_p^*,\ \Sigma^- = \Sigma^+ = \emptyset$;
2. `while` **Stack** $1 \neq \emptyset$
3.   `Pop out a` $\mathcal{X}_p^*$ `from` **Stack** `1`;
4.   `Compute` $\texttt{Inner}(f_1^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*)) \times \texttt{Inner}(f_2^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*))$;
5.   `if` $\texttt{Inner}(f_1^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*)) \times \texttt{Inner}(f_2^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathbf{U}_i^*)) \nsubseteq [a,b]^* \times [c,d]^*$
6.   `Discard` $\mathcal{X}_p^*$ `and return to 2;`
7.   `endif`
8.   **Stack** $2 = \mathbf{U}_i^*$;
9.   while **Stack** $2 \neq \emptyset$
10.   `Pop out a` $\mathcal{U}_i^*$ `from` **Stack** `2`;
11.   `Compute` $\texttt{Inner}(f_1^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathcal{U}_i^*)) \times \texttt{Inner}(f_2^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathcal{U}_i^*))$;
12.   if $\texttt{Inner}(f_1^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathcal{U}_i^*)) \times \texttt{Inner}(f_2^*(\texttt{Dual}(\mathcal{X}_p^*), \mathbf{W}_p^*, \mathcal{U}_i^*)) \nsubseteq [a,b]^* \times [c,d]^*$
13.   `Discard` $\mathcal{U}_i^*$ `and return to 9;`
14.   endif
15.   `Compute` $\texttt{Outer}(f_1^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \mathcal{U}_i^*)) \times \texttt{Outer}(f_2^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \texttt{Dual}(\mathcal{U}_i^*)))$;
16.   if $\texttt{Outer}(f_1^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \mathcal{U}_i^*)) \times \texttt{Outer}(f_2^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \texttt{Dual}(\mathcal{U}_i^*))) \subseteq [a,b]^* \times [c,d]^*$
17. $\Sigma^- = \texttt{Prop}(\mathcal{X}_p^*) \cup \Sigma^-$ `and return to 2;`
18.   endif
19.   `Compute` $\texttt{Outer}(f_1^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \texttt{Dual}(\mathcal{U}_i^*))) \times \texttt{Outer}(f_2^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \mathcal{U}_i^*))$;
20.   if $\texttt{Outer}(f_1^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \texttt{Dual}(\mathcal{U}_i^*))) \times \texttt{Outer}(f_2^*(\mathcal{X}_p^*, \mathbf{W}_p^*, \mathcal{U}_i^*)) \subseteq [a,b]^* \times [c,d]^*$
21. $\Sigma^- = \texttt{Prop}(\mathcal{X}_p^*) \cup \Sigma^-$ `and return to 2;`
22.   endif
23.   `if` $\texttt{Width}(\mathcal{U}_i^*) \leq \varepsilon$, `then discard` $\mathcal{U}_i^*$ `and return to 9;`
24.   else

25. `Bisect` $\mathcal{U}_i^*$ `to` L$\mathcal{U}_i^*$,R$\mathcal{U}_i^*$`, push them on` **Stack** `2 and return to 9;`

26. endif

27. endwhile

28. `if Width(`$\mathcal{X}_p^*$`)` $\leq \varepsilon$`, then` $\Sigma^+ :=$ `Prop(`$\mathcal{X}_p^*$`)` $\cup \Sigma^+$ `and return to 2;`

29. `else Bisect` $\mathcal{X}_p^*$ `to` L$\mathcal{X}_p^*$ `and` R$\mathcal{X}_p^*$`, push them on` **Stack** `1 and return to 2;`

30. `endif`

31. `endwhile`

5.1.6 The Solver of Constrained Minimax Optimization

A general constrained minimax optimization problem can be formulated as:

$$\min_{\mathbf{x}\in\mathbf{X}_0} \max_{\mathbf{y}\in\mathbf{Y}_0} f(\mathbf{x},\mathbf{y}), \forall \mathbf{y} \in \mathbf{Y}_0 \exists \mathbf{x} \in \mathbf{X}_0 : \mathbf{g}(\mathbf{x},\mathbf{y}) \in \mathbb{C}, \tag{5.18}$$

where $\mathbf{x} \in \mathbf{X}_0$ is usually referred to as the decision variable(s) and $\mathbf{y} \in \mathbf{Y}_0$ is usually denoted to uncertain parameters or additive disturbances; and then $\forall \mathbf{y} \in \mathbf{Y}_0 \exists \mathbf{x} \in \mathbf{X}_0 : \mathbf{g}(\mathbf{x},\mathbf{y}) \in \mathbb{C}$ is actually a constraint imposed only on $\mathbf{x}$. Various application problems concerning control and decision can be cast in the framework of constrained minimax optimizations. According to the definition of f^* in (5.11), the minimax value of $\min_{\mathbf{x}\in\mathbf{X}_0} \max_{\mathbf{y}\in\mathbf{Y}_0} f(\mathbf{x},\mathbf{y})$ is equal to $\texttt{Lb}(f^*(\mathbf{X}_p^*, \mathbf{X}_i^*))$, where $\mathbf{X}_p^* = \mathbf{X}^*, \mathbf{X}_i^* = \mathbf{Y}^*$ and $\texttt{Prop}(\mathbf{X}^*) = \mathbf{X}_0, \texttt{Prop}(\mathbf{Y}^*) = \mathbf{Y}_0$. The constrained minimax optimization problem (5.18) can thus be transformed to the computation of the lower bound of $f^*(\mathbf{X}_p^*, \mathbf{X}_i^*)$, which is addressed in Section 5.1.4. Furthermore, the constrained minimax optimization should be performed over the feasible set $\Sigma_{\mathbf{x}}$ of the imposed constraint: $\forall \mathbf{y} \in \mathbf{Y}_0 \forall \mathbf{x} \in \Sigma_{\mathbf{x}} : \mathbf{g}(\mathbf{x},\mathbf{y}) \in \mathbb{C}$. So the solver of traditional set inversion should be applied in advance to obtain this feasible set $\Sigma_{\mathbf{x}}$. The detail of the proposed solver of constrained minimax optimization is illustrated in Algorithm 5.3.

`Algorithm 5.3:  Constrained Minimax Optimization Algorithm`

`In:` $\mathbb{C}, \mathbf{X}_0, \mathbf{Y}_0, \varepsilon$`; Out:` Ω

1. `Initialize` **Stack** $1 = \mathbf{X}_0$ `and` $\Sigma^- = \emptyset$;

2. `while` **Stack** $1 \neq \emptyset$

3. `Pop out a` $\mathbf{X}$ `from` **Stack** `1;`

4. `Compute the natural inclusion function` $G(\mathbf{X}, \mathbf{Y}_0)$ `of` $\mathbf{g}(\mathbf{x},\mathbf{y})$;

5. if $G(\mathbf{X}, \mathbf{Y}_0) \cap \mathbb{C} = \emptyset$, then $\mathbf{X}$ is discarded;
6. elseif $G(\mathbf{X}, \mathbf{Y}_0) \subseteq \mathbb{C}$, then $\Sigma^- = \mathbf{X} \cup \Sigma^-$;
7. elseif Width$(\mathbf{X}) \leq \varepsilon$, then $\mathbf{X}$ is discarded;
8. else Bisect $\mathbf{X}$ to L$\mathbf{X}$,R$\mathbf{X}$ and push them on **Stack** 1;
9. endif
10. endwhile
11. Initialize **Stack** $2 = \Sigma^-$ and $a = +\infty, b = -\infty$;
12. **while** **Stack** $2 \neq \emptyset$
13. Pop out a $\mathbf{X}$ from **Stack** 2;
14. *if* Lb(Out($fR(\mathbf{X}_p^*, \mathbf{y})$)) $< a$ where Prop$(\mathbf{X}_p^*) = \mathbf{X}, \mathbf{y} \in \mathbf{Y}_0$, then $a =$ Lb(Out($fR(\mathbf{X}_p^*, \mathbf{y})$));
15. *if* Lb(Inn($fR(\mathbf{x}, \mathbf{Y}_i^*)$)) $> b$ where $\mathbf{x} \in \mathbf{X}$, Prop$(\mathbf{Y}_i^*) = \mathbf{Y}_0$, then $b =$ Lb(Inn($fR(\mathbf{x}, \mathbf{Y}_i^*)$));
16. **endwhile**
17. $\Omega = \{\mathbf{X}_m | F(\mathbf{X}_m, \mathbf{Y}_0) \cap [a, b] \neq \emptyset, \mathbf{X}_m \subseteq \Sigma^-\}$.

As shown in Algorithm 5.3, an inner approximation Σ^- of the feasible set $\Sigma_\mathbf{x}$ of the imposed constraint $\forall \mathbf{y} \in \mathbf{Y}_0 \forall \mathbf{x} \in \Sigma_\mathbf{x} : \mathbf{g}(\mathbf{x}, \mathbf{y}) \in \mathbb{C}$ is obtained via the solver of set inversion via classical interval analysis from Step 1 to Step 10; then an inclusion domain $[a, b]$ for the constrained minimax function, i.e., $\min_{\mathbf{x} \in \Sigma_\mathbf{x}} \max_{\mathbf{y} \in \mathbf{Y}_0} f(\mathbf{x}, \mathbf{y}) \in [a, b]$, can be obtained through the process from Step 11 to Step 16, where local points $\mathbf{x} \in \mathbf{X}$ and $\mathbf{y} \in \mathbf{Y}_0$ are applied to compute the lower and upper bounds of this constrained minimax function. However, local searches for optimal points $\mathbf{x} \in \mathbf{X}, \mathbf{y} \in \mathbf{Y}_0$ and bisections of those boxes with Width$(\mathbf{X}) > \varepsilon$ can be performed further to obtain a more precise inclusion domain. This inclusion domain is used in Step 17 to find any potential and feasible $\mathbf{X}_m$ to be stored in Ω that might result in the overall minimax optimization value.

5.2 Nonlinear Robust MPC with A Contractive Sequence of Robust Controllable Sets

The solvers of quantified set inversion and constrained minimax optimization are applied in nonlinear robust MPC with a contractive sequence of robust controllable sets, where the solver of quantified set inversion is applied to compute corresponding robust controllable sets with a clear semantic interpretation and the solver of constrained minimax optimization is applied to obtain feasible one-step control inputs underlying robust contractive MPC in a guaranteed numerical way.

5.2.1 Problem Statement

The system to be considered is described by the following constrained nonlinear discrete-time state-space model with parametric or additive uncertainty:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{w}(k), \mathbf{u}(k)), k = 0, \cdots, \tag{5.19}$$

where $\mathbf{x}(k) \in \mathbf{X} \subset \mathbb{R}^n$ is a vector of n state variables and $\mathbf{X}$ is a compact set containing the origin; $\mathbf{w}(k) \in \mathbf{W} \subset \mathbb{R}^l$ is a vector of l uncertain parameters or additive disturbances; $\mathbf{u}(k) \in \mathbf{U} \subset \mathbb{R}^m$ is a vector of m control inputs and $\mathbf{U}$ is a compact set containing the origin. The domains of $\mathbf{X}, \mathbf{W}$ and $\mathbf{U}$ are assumed to be described by boxes where every component of the vectors is an interval. Such a model represents a general class of practical systems with constrained state and control as well as uncertain parameters or additive disturbances. The control target is to drive the system from the initial state $\mathbf{x}(0)$ to a sufficiently small region around the origin asymptotically. The dual-mode approach of model predictive control is adopted here: at first, the one-step control deriving from contractive MPC with a contractive sequence of robust controllable sets drives the system state into a selected terminal robust control invariant set $\mathbb{T}$; and then the related local stabilizing feedback control law is applied instead to drive the system state asymptotically to a sufficiently small region around the origin.

Assume that all robust controllable sets $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T}), i = 1, \cdots, N$ within the constrained state space have been obtained by computing one-step robust controllable sets recursively, the controllability of the nonlinear system with uncertainty is obvious: the system is robustly controllable to the terminal set $\mathbb{T}$ if the initial state is within the maximal robust controllable set. The control inputs can be obtained through the strategy of robust model predictive control with feasible unit control horizon when the state is outside the selected terminal set $\mathbb{T}$, i.e., the one-step control inputs are obtained by solving the following one-step minimax optimization iteratively:

$$\mathbf{u}^{Optimal}(k|k) = \arg \min_{\mathbf{u}(k|k)\in\mathbf{U}} \max_{\mathbf{w}(k)\in\mathbf{W}} [\mathbf{x}^T(k+1|k)Q\mathbf{x}(k+1|k) + \mathbf{u}^T(k|k)R\mathbf{u}(k|k)] \tag{5.20}$$

subject to

$$\mathbf{x}(k+1|k) \in \tilde{\mathcal{K}}_{i-1}(\mathbf{X}, \mathbb{T}), \tag{5.21}$$

where $\mathbf{x}(k) \in \tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$, but $\mathbf{x}(k)$ does not belong to $\tilde{\mathcal{K}}_{i-1}(\mathbf{X}, \mathbb{T})$; Q and R are weighted positive definite matrices for the cost function; and $\mathbf{u}^{Optimal}(k|k)$ is the obtained optimal one-step control input. This one-step control scheme differs from traditional minimax configurations of robust model predictive control, in which the control horizon is usually selected to be large enough to satisfy corresponding terminal constraints and the resulting multi-step minimax optimizations usually require extensive computation. The feasibility and stability of the closed-loop system with the unit control horizon can be guaranteed since the state is driven contractively along the computed robust controllable sets to the selected terminal set. Once the state enters the terminal set, a local stabilizing feedback control is applied instead to drive the system state to a sufficiently small region around the origin.

5.2.2 The Computation of Robust Controllable Sets via Quantified Set Inversion

Assume that the terminal set $\mathbb{T}$ is selected to be a robust control invariant set as well as a polytope, the solver of 1-dimensional quantified set inversion cannot be applied to compute one-step robust controllable sets with a polytope as the terminal set since usually a bounded polytope cannot be described analytically by one inequality. Furthermore, the computed first-step robust controllable set via an interval-based algorithm should be a union of interval vectors as well as a union of polytopes and generally a union of interval vectors cannot be described analytically. Then the inclusion and exclusion conditions between a modal interval vector and a polytope or a union of polytopes are concerned instead in the solver of multi-dimensional quantified set inversion. The inclusion relationship between a multi-dimensional modal interval vector and a multi-dimensional polytope or a union of multi-dimensional polytopes can be deduced as well according to the principle of inclusion in modal interval analysis. Taking the inclusion of a 2-dimensional modal interval vector in a 2-dimensional polytope as an example, the inclusion relationships are shown in Fig. 5.2, where the modalities of the 2-dimensional modal interval vector correspond to the four cases in Fig. 5.1. Thus the i-step robust controllable set $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})(i = 1, \cdots, \infty)$ of a 2-dimensional system can be approximated recursively via the solver of 2-dimensional quantified set inversion starting from an initial 2-dimensional polytope. The inner approximation of the maximal robust controllable set $\tilde{\mathcal{K}}_{\infty}^{-}(\mathbf{X}, \mathbb{T})$ within the constrained state space is reached when $\tilde{\mathcal{K}}_{N+1}^{-}(\mathbf{X}, \mathbb{T}) = \tilde{\mathcal{K}}_{N}^{-}(\mathbf{X}, \mathbb{T})$ for some N and thus $\tilde{\mathcal{K}}_{\infty}^{-}(\mathbf{X}, \mathbb{T}) = \tilde{\mathcal{K}}_{N}^{-}(\mathbf{X}, \mathbb{T})$.

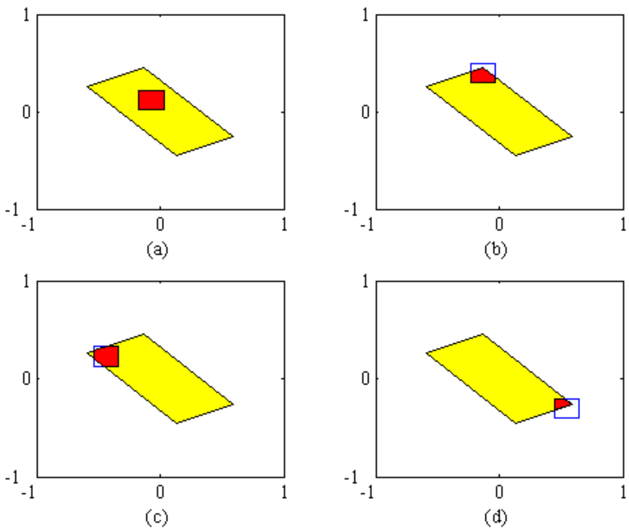


Figure 5.2: An example of the inclusion of a modal interval vector in a polytope

5.2.3 One-step Robust Control via Constrained Minimax Optimization

Once the inner approximations $\tilde{\mathcal{K}}_{i}^{-}(\mathbf{X}, \mathbb{T})(i = 1, \cdots, N)$ of all the i-step robust controllable sets within the constrained state space have been obtained via the solver of quantified set inversion, robust controllability of any initial state can be judged accordingly. Assume that $\mathbf{x}(0) \in \tilde{\mathcal{K}}_{N}^{-}(\mathbf{X}, \mathbb{T})$, i.e., the initial state is robustly controllable to the selected terminal set $\mathbb{T}$ in finite steps, then the one-step control algorithm via robust controllable sets is illustrated in Algorithm 5.4.

`Algorithm 5.4:  One-step Control Algorithm Via Robust Controllable Sets`

`In:` $\mathbf{x}(0), \tilde{\mathcal{K}}_{i}^{-}(\mathbf{X}, \mathbb{T})$`; Out:` $\mathbf{u}^{Optimal}(k|k), \mathbf{x}(k)$

`1.  Get the current state` $\mathbf{x}(k)$`;`

2. `if` $\mathbf{x}(k) \in \mathbb{T}$
3. `Switch to the related local stabilizing feedback control law;`
4. `else`
5. `Find the` i: $i = \min_{i=1,\cdots,\infty}\{\mathbf{x}(k) \in \tilde{\mathcal{K}}_i^-(\mathbf{X}, \mathbb{T})\}$;
6. `Compute` $\mathbf{u}^{Optimal}(k|k)$ `according to (5.20) with the state constraint (5.21);`
7. `Apply` $\mathbf{u}^{Optimal}(k|k)$ `to the system;`
8. `end`
9. `Return to 1 and repeat.`

According to Algorithm 5.4, the one-step control algorithm measures the current state in `Step` 1 and then judges whether the system state has arrived in the selected terminal set $\mathbb{T}$ in `Step` 2. The related local stabilizing feedback control law is applied if the state has arrived in the selected terminal set; otherwise, the algorithm finds the smallest robust controllable set to which the current state belongs in `Step` 5; the one-step control scheme is formulated according to the strategy of robust model predictive control with unit control horizon in `Step` 6 and the optimal one-step control input $\mathbf{u}^{Optimal}(k|k)$ is computed via the proposed solver of constrained minimax optimization illustrated in Algorithm 5.3. It can be seen from the following feasibility and stability analysis that any feasible solution that satisfies the imposed state constraint is an effective control input for the system because such a control input is sufficient to guarantee the feasibility and stability of the closed-loop system. Thus any feasible control input that satisfies the imposed state constraint can be applied instead in the control algorithm to avoid extra efforts for obtaining the exact optimal control input $\mathbf{u}^{Optimal}(k|k)$ in `Step` 6.

5.2.4 Feasibility and Stability Analysis

The feasibility of the addressed one-step control approach means that there always exists a control input $\mathbf{u}$ that can drive the system state from the current robust controllable set $\tilde{\mathcal{K}}_i^-(\mathbf{X}, \mathbb{T})$ to the next robust controllable set $\tilde{\mathcal{K}}_{i-1}^-(\mathbf{X}, \mathbb{T})$. The feasibility of the addressed one-step control approach for a 2-dimensional system is demonstrated in Theorem 5.1.

Theorem 5.1. For any $\mathbf{x}(k) \in \tilde{\mathcal{K}}_i^-(\mathbf{X}, \mathbb{T})$, there always exists an admissible $\mathbf{u}(k) \in \mathbf{U} \subseteq \mathbb{I}(\mathbb{R}^m)$ that renders $\mathbf{x}(k+1) \in \tilde{\mathcal{K}}_{i-1}^-(\mathbf{X}, \mathbb{T}), i = 1, \cdots, \infty$ for all uncertain cases $\mathbf{w}(k) \in \mathbf{W} \subseteq \mathbb{I}(\mathbb{R}^l)$.

Proof. For all robust controllable sets $\tilde{\mathcal{K}}_i^-(\mathbf{X}, \mathbb{T})(i = 1, \cdots, \infty)$ computed by the solver of 2-dimensional quantified set inversion, if $\texttt{Prop}(\mathbf{X}_p^*) \subseteq \tilde{\mathcal{K}}_i^-(\mathbf{X}, \mathbb{T})$, then $\texttt{Outer}(f_1^*(\mathbf{X}_p^*, \mathbf{W}_p^*, \mathbf{U}_i^*)) \times \texttt{Outer}(f_2^*(\mathbf{X}_p^*, \mathbf{W}_p^*, \texttt{Dual}(\mathbf{U}_i^*))) \subseteq \tilde{\mathcal{K}}_{i-1}^-(\mathbf{X}, \mathbb{T})$ or $\texttt{Outer}(f_1^*$ $(\mathbf{X}_p^*, \mathbf{W}_p^*, \texttt{Dual}(\mathbf{U}_i^*))) \times \texttt{Outer}(f_2^*(\mathbf{X}_p^*, \mathbf{W}_p^*, \mathbf{U}_i^*)) \subseteq \tilde{\mathcal{K}}_{i-1}^-(\mathbf{X}, \mathbb{T})$ is satisfied, and according to the semantic statement in (5.13), $\forall \mathbf{x}(k) \in \texttt{Prop}(\mathbf{X}_p^*) \forall \mathbf{w}(k) \in \texttt{Prop}(\mathbf{W}_p^*) \exists \mathbf{x} \in \tilde{\mathcal{K}}_{i-1}^-(\mathbf{X}, \mathbb{T}) \exists \mathbf{u}(k) \in \texttt{Prop}(\mathbf{U}_i^*)\ \mathbf{x}(k+1) = \mathbf{x}$, i.e., the addressed one-step control approach is always feasible. □

The initial control invariant set $\mathbb{T}$ is selected to be a robust control invariant set, then $\tilde{\mathcal{K}}_{i-1}(\mathbf{X}, \mathbb{T}) \subseteq \tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$ for all $i = 1, \cdots, N$ according to the property of robust control invariant sets, i.e., the computed robust controllable sets are the inner approximations of a series of contractive sets. Thus the feasibility of the addressed one-step control approach leads to the stability of the addressed one-step control approach since the control inputs are to drive the system state contractively along the computed robust controllable sets to the selected terminal robust control invariant set. Once the system state enters the selected terminal control invariant set, the related local feedback control law is applied instead to drive the system state asymptotically to a sufficiently small region around the origin.

5.2.5 Example

As an illustrative example for demonstrating the addressed one-step control of constrained nonlinear uncertain discrete-time systems with a contractive sequence of robust controllable sets, the system to be considered is described by the following discrete-time state-space model (Cannon et al., 2003):

$$\begin{cases} x_1(k+1) = x_1(k) + 0.1x_2(k) + 0.1[\mu + (1-\mu)x_1(k)]u(k) \\ x_2(k+1) = 0.1x_1(k) + x_2(k) + 0.1[\mu - 4(1-\mu)x_2(k)]u(k), \end{cases} \tag{5.22}$$

where the control input constraint is $|u| \leq 2$ and the states are constrained to $|\mathbf{x}|_\infty \leq 2$. Different from the discussion in Chapter 4, the system parameter μ is assumed to be uncertain here: $\mu \in [0.85, 0.95]$. The terminal set $\mathbb{T}$ is selected to be a robust control invariant polytope along with a local stabilizing feedback control law $u = \mathbf{kx}$, where $\mathbf{k} = [-1.2131 - 1.2128]$ (Cannon et al., 2003):

$$\begin{bmatrix} 0.8190 & -1.9655 \\ -0.8199 & 1.9655 \\ 3.033 & 3.033 \\ -3.033 & -3.033 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{5.23}$$

The selected terminal set $\mathbb{T}$ along with the local stabilizing feedback control law $u = \mathbf{kx}$ can be demonstrated geometrically to be robust control invariant by using the solver of set inversion via zonotope geometry in Algorithm 4.3, where every sub-zonotope is robust control invariant under the related local stabilizing feedback control law $u = \mathbf{kx}$, just as shown in Fig. 5.3.

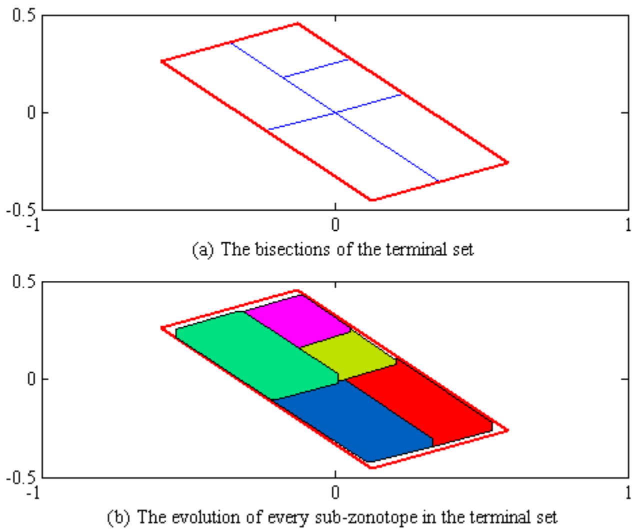


Figure 5.3: The geometrical demonstration of robust control invariance

The inner approximation of the first-step robust controllable set $\tilde{\mathcal{K}}_1(\mathbf{X}, \mathbb{T})$ can be computed via the solver of 2-dimensional quantified set inversion, where the initial terminal set $\mathbb{T}$ is only a polytope. The computed inner approximation of the first-step robust controllable set with the bound of error tolerance $\varepsilon = 0.05$ is shown in Fig. 5.4.

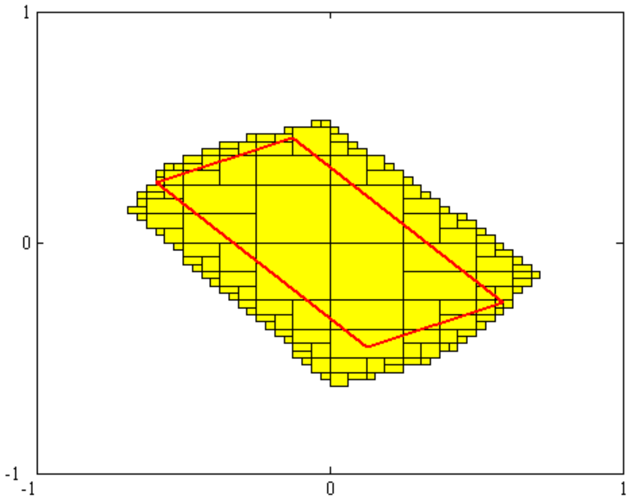


Figure 5.4: The inner approximation of the first-step robust controllable set

The following-step robust controllable sets can be computed recursively starting from the computed inner approximation of the first-step robust controllable set, where the renewed terminal sets for computing the following-step robust controllable sets are a union of interval vectors as well as a union of polytopes. The inner approximation of the maximal robust controllable set with the bound of error tolerance $\varepsilon = 0.05$ is reached when $\tilde{\mathcal{K}}_{48}^{-}(\mathbf{X}, \mathbb{T}) = \tilde{\mathcal{K}}_{49}^{-}(\mathbf{X}, \mathbb{T})$, which is shown in Fig. 5.5.

The dual-mode approach of model predictive control is adopted to control the system, i.e., the addressed one-step control is to drive the system state contractively along the obtained robust controllable sets to the selected terminal set $\mathbb{T}$, and then the local stabilizing feedback control law is applied instead to drive the system state asymptotically to a sufficiently small region around the origin. Assume that the initial state of the considered nonlinear system is $\mathbf{x}(0) = (1.75, -1.6)$, the resulting robust control process of the dual-mode approach of model predictive control is shown in Fig. 5.4, where the uncertain parameter μ is a random number series between 0.85 and 0.95 during the control process and $Q = R = 1$.

It can be seen from Fig. 5.6 that the addressed nonlinear robust contractive MPC

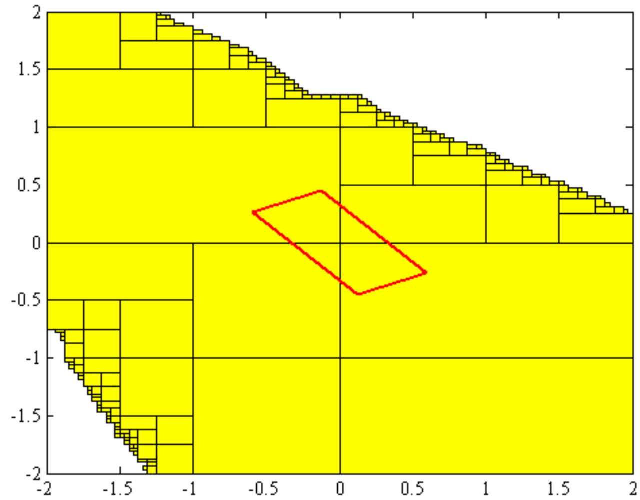


Figure 5.5: The inner approximation of the maximal robust controllable set

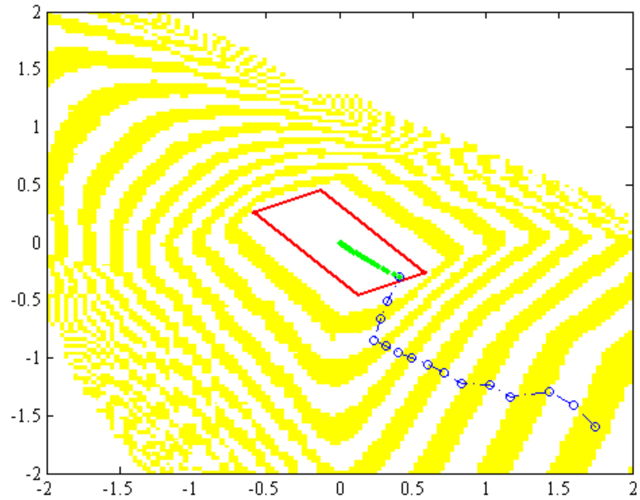


Figure 5.6: The robust control process with $\mathbf{x}(0) = (1.75, -1.6)$

with a contractive sequence of robust controllable sets is guaranteed to be feasible and stable, i.e, the system can be driven contractively along the computed robust controllable sets to the selected robust control invariant set despite of an uncertain parameter during the control process.

5.3 Summary

Modal interval analysis has been introduced in a comparative way relative to classical interval analysis, where every concept of modal interval analysis has been derived by extending a counterpart concept of classical interval analysis and modal interval analysis has been treated as an extension of classical interval analysis in modality, inclusion, semantics and rational. The solver of 1-dimensional quantified set inversion via modal interval analysis has been generalized to multi-dimensional cases for computing robust controllable sets of constrained nonlinear uncertain discrete-time systems with a clear semantic interpretation. An interval-based solver of constrained minimax optimization has also been proposed to compute one-step control inputs for the addressed nonlinear robust contractive MPC with a contractive sequence of robust controllable sets in a reliable way.

Chapter 6

Nonlinear Robust Contractive MPC via Hybrid Tools

Using modal interval analysis, robust controllable sets of constrained nonlinear uncertain discrete-time systems can be computed with a clear semantic interpretation, as demonstrated in Chapter 5. Based on the computed robust controllable sets as a contractive sequence of sets, nonlinear robust contractive MPC with a contractive sequence of robust controllable sets can be formulated and it turns out to be a robust MPC scheme with feasible unit control horizon and additional contractive constraint. However, the contractive sequence of robust controllable sets are represented by unions of interval vectors therein and thus extra efforts are required to judge the inclusion of the measured state to a union of interval vectors during the real-time control process, which increases the memory and computational burdens of the control task. Furthermore, the burden of off-line computations for robust controllable sets grows exponentially with the total dimension of the state space and the control space, which makes the computation of robust controllable sets of high-dimensional systems prohibitively heavy.

In order to improve the efficiency of nonlinear robust contractive MPC with a contractive sequence of robust controllable sets addressed in Chapter 5, some additional measures are to be taken in this chapter. First, the computed robust controllable sets are to be approximated innerly by polytopes in Section 6.1. The simplified nonlinear robust contractive MPC with a contractive sequence of polytopic robust controllable sets is also discussed in Section 6.1. The structures of various constrained nonlinear uncertain discrete-time systems are to be explored further in Section 6.2 and 6.3 to reduce the total dimension of the state space and the control space needed to be bisected during the computation of polytopic robust controllable sets.

6.1 Nonlinear Robust MPC with A Contractive Sequence of Polytopic Robust Controllable Sets

This section addresses the computation of polytopic robust controllable sets for a general constrained nonlinear uncertain discrete-time system and provides a general framework for nonlinear robust contractive MPC with a contractive sequence of polytopic robust controllable sets. It is an extension and simplification of Chapter 5 since robust controllable sets are represented by polytopes rather than unions of interval vectors. The idea of distinguishing the modalities of intervals in modal interval analysis is also used implicitly in the proposed algorithm for computing polytopic robust controllable sets although the algorithm is in the form of classical interval analysis. The proposed approach in this section can be treated as a most general framework of nonlinear robust contractive MPC with a contractive sequence of polytopic robust controllable sets for constrained nonlinear uncertain discrete-time systems.

6.1.1 Problem Statement

The system to be considered is described by the following constrained nonlinear uncertain discrete-time state-space model:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{w}(k), \mathbf{u}(k)), k = 0, \cdots, \tag{6.1}$$

where $\mathbf{x}(k) \in \mathbb{X} \subset \mathbb{R}^n$ is a vector of n state variables and $\mathbb{X}$ is a compact set containing the origin; $\mathbf{w}(k) \in \mathbb{W} \subset \mathbb{R}^l$ is a vector of l uncertain parameters and (or) additive disturbances; $\mathbf{u}(k) \in \mathbb{U} \subset \mathbb{R}^m$ is a vector of m control inputs and $\mathbb{U}$ is a compact set containing the origin. The domains of $\mathbb{X}, \mathbb{W}$ and $\mathbb{U}$ are assumed to be described by boxes $\mathbf{X}, \mathbf{W}, \mathbf{U}$, i.e., every component of the vectors is an interval. The control target is to drive the system state asymptotically from the initial state $\mathbf{x}(0)$ to a sufficiently small region around the origin. The dual-mode approach of MPC is adopted here: at first, the one-step control deriving from nonlinear robust contractive MPC with a contractive sequence of robust controllable sets drives the system state into a selected terminal set $\mathbb{T}$; and then a local stabilizing feedback control law is applied instead to drive the system to a sufficiently small region around the origin.

Assume that all robust controllable sets $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T}), i = 1, \cdots$ within the constrained state space have been obtained by computing one-step robust controllable sets recursively, then the one-step control inputs underlying nonlinear robust contractive MPC with a contractive sequence of robust controllable sets can be obtained as well by solving the following constrained minimax optimization iteratively:

$$\min_{\mathbf{u}(k|k)\in\mathbf{U}} \max_{\mathbf{w}(k|k)\in\mathbf{W}} [\mathbf{x}^T(k+1|k)Q\mathbf{x}(k+1|k) + \mathbf{u}^T(k|k)R\mathbf{u}(k|k)] \tag{6.2}$$

subject to

$$\mathbf{x}(k+1|k) \in \tilde{\mathcal{K}}_{i-1}(\mathbf{X}, \mathbb{T}), \tag{6.3}$$

where $\mathbf{x}(k) \in \tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$, but $\mathbf{x}(k)$ does not belong to $\tilde{\mathcal{K}}_{i-1}(\mathbf{X}, \mathbb{T})$; Q and R are weighted positive definite matrices; and $\mathbf{u}^{Optimal}(k|k)$ is the resulting optimal one-step control input. The terminal set $\mathbb{T}$ can be designed in advance to be a robust control invariant polytope along with a local stabilizing feedback control law $u = \mathbf{kx}$ (Cannon et al., 2003).

6.1.2 The First-step Robust Controllable Set Approximation Algorithm

Assume that the terminal set $\mathbb{T}$ is designed to be a robust control invariant polytope for the system (6.1), then an inner approximation of the first-step robust controllable set can be computed directly by an interval-based branch-and-bound algorithm, which is listed in Algorithm 6.1.

Algorithm 6.1: The First-step Robust Controllable Set Approximation Algorithm

In: $\mathbf{X}, \mathbf{W}, \mathbf{U}, \mathbb{T}, \varepsilon$; Out: $\Sigma_{\mathbf{x}}$

1. Initialize **Stack** $1 = \mathbf{X}, \Sigma_{\mathbf{x}} = \emptyset$;
2. while **Stack** $1 \neq \emptyset$
3. Pop out a $\mathbf{X}_i$ from **Stack** 1;
4. Compute $\mathbf{f}(\mathbf{X}_i, \mathbf{w}_{Local}, \mathbf{U})$;
5. if $\mathbf{f}(\mathbf{X}_i, \mathbf{w}_{Local}, \mathbf{U}) \cap \mathbb{T} = \emptyset$

6. Discard $\mathbf{X}_i$ and return to 2;
7. endif
8. Initialize **Stack** $2 = \mathbf{U}$;
9. while **Stack** $2 \neq \emptyset$
10. Pop out a $\mathbf{U}_j$ from **Stack** 2;
11. Compute $\mathbf{f}(\mathbf{X}_i, \mathbf{w}_{Local}, \mathbf{U}_j)$;
12. if $\mathbf{f}(\mathbf{X}_i, \mathbf{w}_{Local}, \mathbf{U}_j) \cap \mathbb{T} = \emptyset$
13. Discard $\mathbf{U}_j$ and return to 9;
14. elseif $\mathbf{f}(\mathbf{X}_i, \mathbf{W}, \mathtt{Mid}(\mathbf{U}_j)) \subseteq \mathbb{T}$
15. $\Sigma_{\mathbf{x}} = \mathbf{X}_i \cup \Sigma_{\mathbf{x}}$ and return to 2;
16. elseif $\mathtt{Width}(\mathbf{U}_j) \leq \varepsilon$, then discard $\mathbf{U}_j$ and return to 9;
17. else
18. Bisect $\mathbf{U}_j$ to $\mathtt{L}\mathbf{U}_j$,$\mathtt{R}\mathbf{U}_j$, push them on **Stack** 2 and return to 9;
19. endif
20. endwhile
21. if $\mathtt{Width}(\mathbf{X}_i) \leq \varepsilon$, then discard $\mathbf{X}_i$ and return to 2;
22. else
23. Bisect $\mathbf{X}_i$ to $\mathtt{L}\mathbf{X}_i$ and $\mathtt{R}\mathbf{X}_i$, push them on **Stack** 1 and return to 2;
24. endif
25. endwhile

As shown in Algorithm 6.1, $\mathbf{w}_{Local}$ in Step 4 and 11 relates to a local search of the concrete value $\mathbf{w}(k) \in \mathbf{W}$: if there exists such a value $\mathbf{w}_{Local}$ that renders $\mathbf{f}(\mathbf{X}_i, \mathbf{w}_{Local}, \mathbf{U}) \cap \mathbb{T} = \emptyset$, then for all $\mathbf{u} \in \mathbf{U}$, it is impossible to drive the state $\mathbf{X}_i$ to the terminal set $\mathbb{T}$ in the case of $\mathbf{w}_{Local}$ at the next step, so $\mathbf{X}_i$ does not belong to the first-step robust controllable set and it is discarded in Step 6; however, only a part of $\mathbf{U}$ is tested in Step 12, i.e., $\mathbf{f}(\mathbf{X}_i, \mathbf{w}_{Local}, \mathbf{U}_j) \cap \mathbb{T} = \emptyset$, so for all $\mathbf{U}_j$, it is impossible to drive the state $\mathbf{X}_i$ to the terminal set $\mathbb{T}$ in the case of $\mathbf{w}_{Local}$ at the next step, then $\mathbf{U}_j$ is discarded instead in Step 13. On the contrary, if there exists a control input such as the middle value $\mathtt{Mid}(\mathbf{U}_j)$ that renders $\mathbf{f}(\mathbf{X}_i, \mathbf{W}, \mathtt{Mid}(\mathbf{U}_j)) \subseteq \mathbb{T}$, then for all uncertain cases $\mathbf{w}(k) \in \mathbf{W}$, the state $\mathbf{X}_i$ can be driven to the terminal set $\mathbb{T}$ via an admissible control input $\mathtt{Mid}(\mathbf{U}_j)$ at the next step, which signifies that $\mathbf{X}_i$ belongs to the first-step robust controllable set and it is to be stored in $\Sigma_{\mathbf{x}}$. If no judgement can be made for

$\mathbf{X}_i$ or $\mathbf{U}_j$ and the widths of them are beyond the error tolerance ε, just as in `Step` 16 and 21, they are to be discarded as well; otherwise, $\mathbf{X}_i$ or $\mathbf{U}_j$ is to be bisected further for a finer judgement, just as shown in `Step` 18 and 23.

6.1.3 The Revised Polytopic Approximation Algorithm

According to the first-step robust controllable set approximation algorithm in Algorithm 6.1, the obtained inner approximation of the first-step robust controllable set $\Sigma_{\mathbf{x}}$ is a union of boxes as well as a union of polytopes. The union of polytopes can be further approximated innerly by one polytope according to the one-step set polytopic approximation algorithm proposed in (Bravo et al., 2005). The benefits of representing a robust controllable set by one polytope rather than by a union of boxes range from reducing memory resources, facilitating the synthesis of real-time constrained control and so on. The convex hull of the union of polytopes is used in the following revised polytopic approximation algorithm for improving the efficiency of the published polytopic approximation algorithm through decreasing the number of complementary sets as well as separating complementary sets and the contracted convex hull instead of separating complementary sets and the contracted union of boxes during the process of computing α-support hyperplane for each complementary set.

Assume that the convex hull of the union of polytopes $\mathcal{H} = \texttt{Hull}(\Sigma_{\mathbf{x}})$ as well as its vertices has been obtained via vertex enumeration (Kvasnica et al., 2006), i.e., $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^n | H^x\mathbf{x} \leq K^b\}$ and its vertices are $\{\mathbf{v}_k^{\mathcal{H}}\}_{k=1}^{n_h}$, where n_h is the number of vertices on $\mathcal{H}$. Then the complementary set $\mathcal{C}$ of $\Sigma_{\mathbf{x}}$ relative to its convex hull $\mathcal{H}$ is a union of polytopes and it can be obtained by the set difference $\mathcal{C} = \mathcal{H} \setminus \Sigma_{\mathbf{x}} = \cup_{m=1}^{n_c} \mathcal{C}_m$, where n_c is the number of polytopes in $\mathcal{C}$. The vertices of each polytope $\mathcal{C}_m$ in $\mathcal{C}$ can be obtained as well and they are assumed to be $\{\mathbf{v}_j^{\mathcal{C}_m}\}_{j=1}^{n_{c_m}}$, where n_{c_m} is the number of vertices on $\mathcal{C}_m$. The $\alpha-$support hyperplane for $\mathcal{C}_m$ is a hyperplane $\mathbf{c}_m^T\mathbf{x} = 1$ such that (Bravo et al., 2005): $\mathbf{c}_m^T\mathbf{x} > 1$ for every $\mathbf{x} \in \mathcal{C}_m$ and $\alpha \cdot \mathbf{c}_m^T\mathbf{x} \leq 1$ for every $\mathbf{x} \in \mathcal{H}$, where $\alpha \in [0, 1]$. The computation of the $\alpha-$support hyperplane for each $\mathcal{C}_m$ can be transformed to be a linear programming problem, i.e.,

$$\min_{\{\mathbf{c}_m, \gamma\}} \gamma \tag{6.4}$$

subject to

$$\begin{cases} \mathbf{c}_m^T \mathbf{v}_j^{\mathcal{C}_m} > 1, j = 1, \cdots, n_{c_m} \\ \mathbf{c}_m^T \mathbf{v}_k^{\mathcal{H}} \leq \gamma, k = 1, \cdots, n_h \\ \gamma \geq 1 \end{cases} \tag{6.5}$$

Once the α-support hyperplane for $\mathcal{C}_m$ is obtained, those $\{\mathcal{C}_r | \mathbf{c}_m^T \mathbf{v}_j^{\mathcal{C}_r} > 1, j = \{1, \cdots, n_{c_r}\}$ $, r \in \{m+1, \cdots, n_c\}$ should be discarded to avoid redundant separation from its corresponding contracted convex hull. Then the resulting polytopic approximation for $\Sigma_\mathbf{x}$ is to be:

$$\mathcal{P}_a = \cap_{m=1}^{n_{c_f}} \{\mathbf{x} \in \mathbb{R}^n | \mathbf{c}_m^T \mathbf{x} \leq 1\} \tag{6.6}$$

where n_{c_f} is the number of all processed polytopes in $\mathcal{C}$. The detail of the revised polytopic approximation algorithm is shown in Algorithm 6.2.

Algorithm 6.2: The Revised Polytopic Approximation Algorithm

In: $\Sigma_\mathbf{x}$; Out: $\mathcal{P}^a$

1. $\mathcal{H} = \texttt{Hull}(\Sigma_\mathbf{x})$;
2. $\mathcal{P}^a = \mathcal{H}$;
3. $\mathcal{C} = \mathcal{H} \setminus \Sigma_\mathbf{x} = \cup_{m=1}^{n_c} \mathcal{C}_m$;
4. for $m = 1 : 1 : n_c$
5.   *if* $\mathcal{C}_m \cap \mathcal{P}^a \neq \emptyset$
6. $\mathbf{c}_m = \arg\min_{\{\mathbf{c}_m, \gamma\}} \gamma$;
7. $\mathcal{P}^a = \mathcal{P}^a \cap \{\mathbf{x} \in \mathbb{R}^n | \mathbf{c}_m^T \mathbf{x} \leq 1\}$;
8.   *end*
9. end

6.1.4 The Following-step Robust Controllable Set Approximation Algorithm

Once the inner approximation of the first-step robust controllable set $\Sigma_\mathbf{x}$ has been approximated by one polytope $\mathcal{P}_a$ through the revised polytopic approximation algorithm in Algorithm 6.2, the following-step robust controllable sets can be computed iteratively by renewing the terminal set $\mathbb{T}$ with $\mathcal{P}_i^a$ in Algorithm 6.1, where i is the polytopic approximation of i-step robust controllable set and $\mathcal{P}_1^a = \mathcal{P}_a$. If the terminal set $\mathbb{T}$ is designed to be robust control invariant in advance, then theoretically the computed first-step robust controllable set should contain it. However, the limitation of the bound of error tolerance ε of the interval-based algorithm in Algorithm 6.1 and the conservativeness of the polytopic approximation algorithm in Algorithm 6.2 might lead to:

$$\mathbb{T} \nsubseteq \mathcal{P}_1^a. \tag{6.7}$$

Then the obtained first-step polytopic robust controllable set $\mathcal{P}_1^a$ is not robust control invariant. A remedy for this problem is to replace $\mathcal{P}_1^a$ by a union $\mathcal{P}_1^a \cup \mathbb{T}$ in the computation of the second-step robust controllable set and obviously $\mathcal{P}_1^a \cup \mathbb{T}$ is a robust control invariant set for the system (6.1). To obtain a union of polytopes as a robust controllable set also happens in piecewise-affine and hybrid systems (Rakovic et al., 2003). Generally, the terminal set $\mathbb{T}$ for computing $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$ is renewed to be:

$$\cup_{j=0}^{i-1} \mathcal{P}_j^a, \tag{6.8}$$

where $\mathcal{P}_0^a = \mathbb{T}$. Corresponding exclusion test and inclusion test between a box and a union of polytopes are concerned instead in `Step` 5, 12 and 14 of Algorithm 6.1, respectively. An illustrative example demonstrating the inclusion and the exclusion between a box and a union of two polytopes is shown in Fig. 6.1.

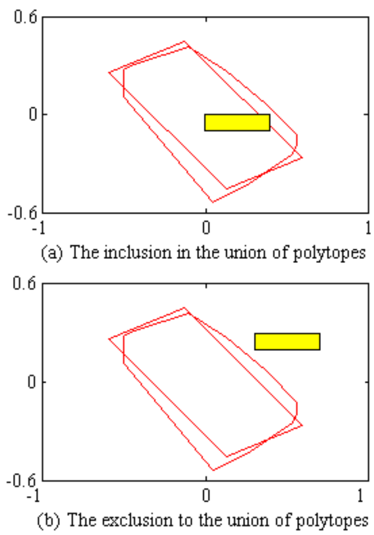


Figure 6.1: The inclusion and the exclusion between a box and a union of two polytopes

To judge whether a box is included in a union of polytopes can be transformed to judge whether their set difference is empty, which can be further transformed to be a linear programming problem as well (Kerrigan, 2000). The polytopic approximation of the maximal robust controllable set $\tilde{\mathcal{K}}_\infty(\mathbf{X}, \mathbb{T})$ within the constrained state space is reached when $\mathcal{P}_N^a = \mathcal{P}_{N+1}^a$ for some N. Nevertheless, the polytopic robust controllable set approximation algorithm can be simplified if $\mathcal{P}_{i-1}^a \subseteqq \mathcal{P}_i^a$ for all $i = 1, \cdots, N$, where corresponding exclusion test and inclusion test are fulfilled just between two polytopes.

6.1.5 Nonlinear Robust Contractive MPC with A Contractive Sequence of Polytopic Robust Controllable Sets

Once all the polytopic robust controllable sets $\mathcal{P}_j^a (j = 1, \cdots, N)$ have been obtained, the robust controllability of any initial state can be judged accordingly. Assume that $\mathbf{x}(0) \in \cup_{j=0}^N \mathcal{P}_j^a$, i.e., the initial state is robustly controllable to the designed terminal set $\mathbb{T}$ in finite steps, then the nonlinear robust contractive MPC with a contractive sequence of polytopic robust controllable sets is illustrated in Algorithm 6.3.

`Algorithm 6.3:  Nonlinear MPC With Polytopic Robust Controllable Sets`

`In:` $\mathbf{x}(0), \mathcal{P}_j^a (j = 0, \cdots, N)$; `Out:` $\mathbf{u}^{Optimal}(k|k), \mathbf{x}$

1. `Get the current state` $\mathbf{x}(k)$;
2. if $\mathbf{x}(k) \in \mathbb{T}$
3. `Switch to a designed local stabilizing feedback control law;`
4. else
5. `Find the` i: $i = \min_{j=1,\cdots,N}\{\mathbf{x}(k) \in \mathcal{P}_j^a\}$;
6. `Compute` $\mathbf{u}^{Optimal}(k|k)$ `with the contractive constraint (6.3);`
7. `Apply` $\mathbf{u}^{Optimal}(k|k)$ `to the system;`
8. end
9. `Return to 1 and repeat.`

According to Algorithm 6.3, the control algorithm measures the current state in `Step` 1 and then judges whether the system state has arrived in the terminal set in `Step` 2. A local stabilizing feedback control is applied if the state has arrived in the terminal set $\mathbb{T}$; otherwise, the algorithm finds the smallest polytopic robust controllable set to which the current state belongs in `Step` 5; the one-step control scheme is formulated according to the strategy of robust MPC with unit control horizon and additional contractive constraint in `Step` 6, where $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$ is denoted to $\cup_{j=0}^{i} \mathcal{P}_j^a$ according to the proposed algorithm of computing polytopic robust controllable sets. It can be seen that any feasible solution that satisfies the contractive constraint is an effective control input for the system since such a control input is sufficient to guarantee the feasibility and stability of the closed-loop system. A feasible control input can be obtained as well via an interval-based minimax optimization algorithm addressed in Chapter 5 in a guaranteed numerical way.

6.1.6 Example

The illustrative example to be considered is (Cannon et al., 2003):

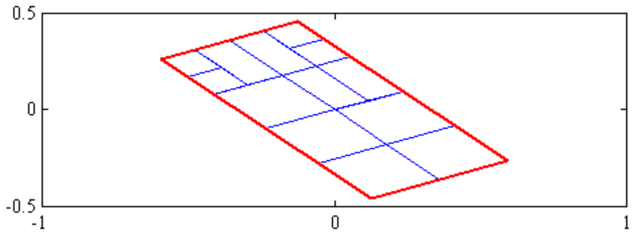
$$\begin{cases} x_1(k+1) = x_1(k) + 0.1x_2(k) + 0.1[\mu + (1-\mu)x_1(k)]u(k) \\ x_2(k+1) = 0.1x_1(k) + x_2(k) + 0.1[\mu - 4(1-\mu)x_2(k)]u(k), \end{cases} \tag{6.9}$$

where the control is constrained to $|u| \leq 2$; the state variables are constrained to $|\mathbf{x}|_\infty \leq 4$; and the parameter μ is assumed to be uncertain: $\mu \in [0.75, 0.95]$. The terminal set $\mathbb{T}$ is selected to be a polytope along with a local stabilizing feedback control law $u = \mathbf{kx}$, where $\mathbf{k} = [-1.2131 - 1.2128]$ (Cannon et al., 2003):

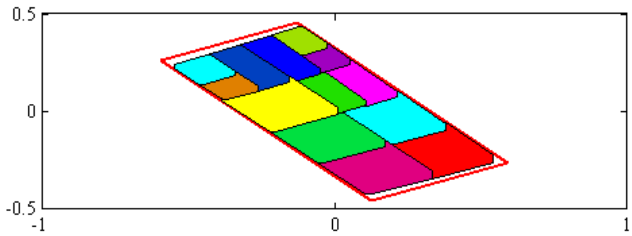
$$\begin{bmatrix} 0.8190 & -1.9655 \\ -0.8199 & 1.9655 \\ 3.033 & 3.033 \\ -3.033 & -3.033 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{6.10}$$

The selected terminal set $\mathbb{T}$ along with the related local stabilizing feedback control law $u = \mathbf{kx}$ can be demonstrated geometrically to be robust control invariant by using the solver of set inversion via zonotope geometry in Algorithm 4.3, where every sub-zonotope is robust control invariant under the related local feedback control law $u = \mathbf{kx}$, just as shown in Fig. 6.2.

The first-step robust controllable sets with the bound of error tolerance $\varepsilon = 0.025$ and $\varepsilon = 0.05$ as well as their polytopic approximations are shown in Fig. 6.3(a) and 6.3(b), respectively. It can be seen from Fig. 6.3(a) that the terminal set is robust control invariant since $\mathbb{T} \subset \mathcal{P}_1^a \subset \tilde{\mathcal{K}}(\mathbf{X}, \mathbb{T})$. Obviously, the obtained polytopic robust



(a) The bisections of the terminal set



(b) The evolution of every sub-zonotope of the terminal set

Figure 6.2: The geometrical demonstration of robust control invariance

controllable set with $\varepsilon = 0.025$ is bigger than the obtained polytopic robust controllable set with $\varepsilon = 0.05$. Furthermore, the bigger bound of error tolerance $\varepsilon = 0.05$ renders $\mathbb{T} \nsubseteq \mathcal{P}_1^a$ in Fig. 6.3(b). This case is more general since the polytopic approximation is also conservative and might lead to $\mathbb{T} \nsubseteq \mathcal{P}_1^a$ as well. In order to demonstrate the general principle of the proposed polytopic robust controllable set approximation algorithm completely, the bound of error tolerance is selected to be $\varepsilon = 0.05$ for computing all polytopic robust controllable sets.

According to the polytopic robust controllable set approximation algorithm, the terminal set for computing the second-step robust controllable set is renewed to be a union of polytopes: $\mathbb{T} \cup \mathcal{P}_1^a$. All polytopic robust controllable sets within the constrained state space can be computed iteratively with the bound of error tolerance $\varepsilon = 0.05$. The computed polytopic robust controllable sets for the system are shown in Fig. 6.4.

The robust control processes of the dual-mode approach of MPC for the system with the initial state $\mathbf{x}(0) = (-1.5, 1.1)$ and $\mathbf{x}(0) = (0.3, -2.1)$ are shown in Fig. 6.5, where the system state has been driven contractively along the computed polytopic robust controllable sets to the terminal robust control invariant polytope $\mathbb{T}$ and then

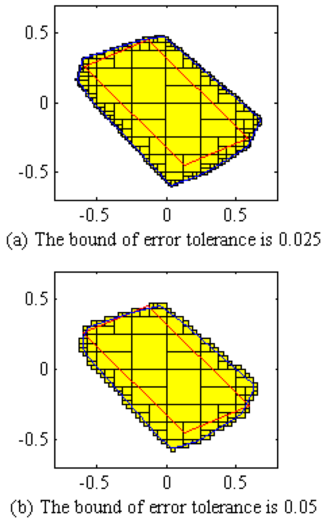


Figure 6.3: The first-step robust controllable sets with $\varepsilon = 0.025$ and $\varepsilon = 0.05$

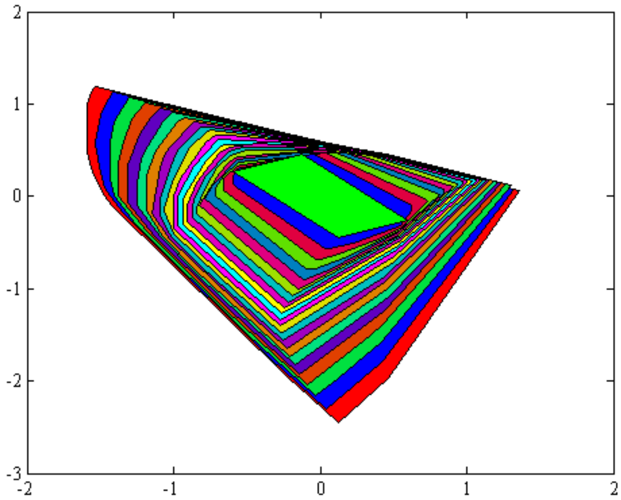


Figure 6.4: The computed polytopic robust controllable sets with $\varepsilon = 0.05$

the related local stabilizing feedback control law is applied instead to drive the state to a sufficiently small region around the origin.

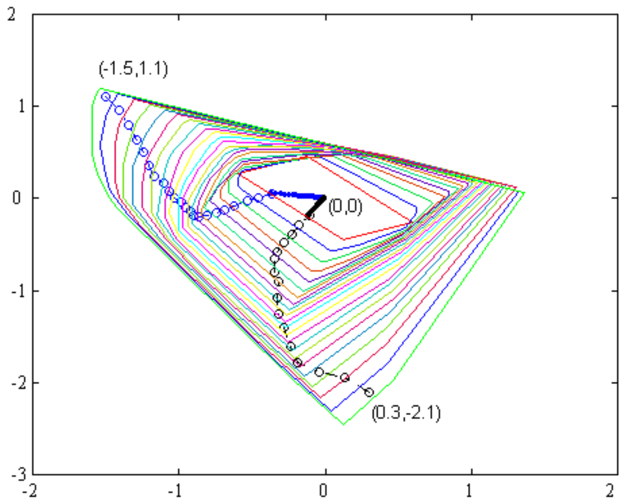


Figure 6.5: The robust control processes of the dual-mode approach

6.2 Robust Contractive MPC of Nonlinear Systems with Affine State Part

[1] A general framework for computing polytopic robust controllable sets of constrained nonlinear uncertain discrete-time systems has been addressed in Section 6.1. However, the computation burden of the proposed interval-based approach for computing polytopic robust controllable sets grows exponentially with the total dimension of the state space and the control space. In this section, the structure of a specific kind of constrained nonlinear uncertain discrete-time systems with affine state part is to be explored further to improve the efficiency of the interval-based approach for computing polytopic robust controllable sets.

[1] The initial idea of studying nonlinear discrete-time systems with affine state part was proposed by Dr. J. M. Bravo during the external stay in Huelva and the author just embodied the idea by designing all algorithms needed, selecting a suitable example and presenting the simulation results.

6.2.1 Problem Statement

The constrained system considered is described by a nonlinear discrete-time state-space model with the following structure:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + \mathbf{f}(\mathbf{x}_{NL}(k), \mathbf{w}(k), \mathbf{u}(k)), k = 0, \cdots, \tag{6.11}$$

where $\mathbf{x}(k) \in \mathbb{X} \subset \mathbb{R}^n$ is a vector of n state variables; $\mathbf{f}(\mathbf{x}_{NL}(k), \mathbf{u}(k), \mathbf{w}(k))$ is the nonlinear part of the system containing only part of the state variables $\mathbf{x}_{NL}(k) \in \mathbb{R}^p (p < n)$; $\mathbf{w}(k) \in \mathbb{W} \subset \mathbb{R}^l$ is a vector of l uncertain parameters and (or) additive disturbances; and $\mathbf{u}(k) \in \mathbb{U} \subset \mathbb{R}^m$ is a vector of m control inputs. The domains of $\mathbb{X}$,$\mathbb{W}$ and $\mathbb{U}$ are assumed to be compact sets described by boxes $\mathbf{X}, \mathbf{W}, \mathbf{U}$, where every component of the vectors is an interval. The terminal set $\mathbb{T}$ for the system is designed in advance to be a robust control invariant set, i.e., for all $\mathbf{x}(k) \in \mathbb{T}$, there exists an admissible control input $\mathbf{u}(k) \in \mathbf{U}$ such that $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) \in \mathbb{T}$ for all uncertain cases $\mathbf{w}(k) \in \mathbf{W}$.

The nonlinear robust MPC with a contractive sequence of polytopic robust controllable sets is also adopted as the control strategy for the system (6.11). However, the computation of polytopic robust controllable sets can be simplified here since only the nonlinear part of the state space is needed to be bisected during the process of computing robust controllable sets. Assume that all robust controllable sets $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T}), i = 1, \cdots$ within the constrained state space have been approximated innerly by computing one-step robust controllable sets recursively, then the one-step control inputs underlying robust MPC can be obtained as well by solving the following constrained minimax optimization iteratively:

$$\min_{\mathbf{u}(k|k)\in\mathbf{U}} \max_{\mathbf{w}(k|k)\in\mathbf{W}} [\mathbf{x}^T(k+1|k)Q\mathbf{x}(k+1|k) + \mathbf{u}^T(k|k)R\mathbf{u}(k|k)] \tag{6.12}$$

subject to

$$\mathbf{x}(k+1|k) \in \tilde{\mathcal{K}}_{i-1}(\mathbf{X}, \mathbb{T}), \tag{6.13}$$

where $\mathbf{x}(k) \in \tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$, but $\mathbf{x}(k)$ does not belong to $\tilde{\mathcal{K}}_{i-1}(\mathbf{X}, \mathbb{T})$; Q and R are weighted positive definite matrices; and $\mathbf{u}^{Optimal}(k|k)$ is the resulting optimal one-step control input. The terminal set $\mathbb{T}$ can be designed in advance to be a robust control invariant polytope.

6.2.2 The Computation of Robust Controllable Sets via Interval Analysis and Polytope Geometry

The one-step robust controllable set approximation algorithm for the specific system (6.11) is based on interval analysis and polytope geometry. Concretely, interval analysis is applied to branch and bound the nonlinear part $\mathbf{f}(\mathbf{x}_{NL}(k), \mathbf{u}(k), \mathbf{w}(k))$ of the system as well as the control space and polytope geometry is applied to compute sub-controllable sets for each $\mathbf{X}_{NL}^{i}(k)$ and $\mathbf{U}^{j}(k)$ of their subpavings. Assume that $\Sigma_{\mathbf{X}_{NL}}$ is the subpaving of $\mathbf{X}_{NL}$, $\Sigma_{\mathbf{U}}$ is the subpaving of $\mathbf{U}$ and $\mathbb{T} = \{\mathbf{x} \in \mathbb{R}^n | T_{t\times n}^{x}\mathbf{x} \leq T_{t\times 1}^{b}\}$ is the terminal set, then the one-step robust controllable set approximation algorithm is shown in Algorithm 6.4.

Algorithm 6.4: One-step Robust Controllable Set Approximation Algorithm

In: $\Sigma_{\mathbf{X}_{NL}}, \Sigma_{\mathbf{U}}, \mathbb{T}$; Out: $\Sigma_{\mathbf{x}}$

Stack $1 = \Sigma_{\mathbf{X}_{NL}}, \Sigma_{\mathbf{x}} = \emptyset$;
while Stack $1 \neq \emptyset$
  Pop out a $\mathbf{X}_{NL}$ from **Stack** 1;
  Stack $2 = \Sigma_{\mathbf{U}}$;
  while **Stack** $2 \neq \emptyset$
    Pop out a $\mathbf{U}$ from **Stack** 2;
    Compute $\Sigma = \{\mathbf{x} \subseteq \mathbb{R}^n | T_{t\times n}^{x} \cdot A\mathbf{x} \leq T_{t\times 1}^{b} - T_{t\times n}^{x}\mathbf{f}(\mathbf{X}_{NL}, \mathbf{U}, \mathbf{W})\}$;
    if $\Sigma \neq \emptyset$
      Push Σ on $\Sigma_{\mathbf{x}}$;
    end
  end
end

6.2.3 Polytopic Approximation of Robust Controllable Sets

The computed one-step robust controllable set $\Sigma_{\mathbf{x}}$ is a union of polytopes. The revised polytopic approximation algorithm addressed in Section 6.1 can also be applied to approximate the union of polytopes innerly by one polytope. However, the number of the union's complementary sets relative to its convex hull grows exponentially with the number of polytopes. It becomes difficult to obtain a polytopic approximation for a high-dimensional systems. An alternative polytopic approximation algorithm is proposed in this subsection.

Assume that the convex hull of the union of polytopes $\mathcal{H} = \texttt{Hull}(\Sigma_{\mathbf{x}})$ has been obtained via vertex enumeration, i.e., $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^n | H^x_{m\times n}\mathbf{x} \leq K^b_{m\times 1}\}$. The principle of the proposed polytopic approximation algorithm is to contract the convex hull $\mathcal{H}$ as small as possible until the contracted convex hull $\mathcal{H}^c = \alpha \cdot \mathcal{H}$ is a subset of $\Sigma_{\mathbf{x}}$, i.e., $\mathcal{H}^c \subseteq \Sigma_{\mathbf{x}}$ where $\alpha \in [0, 1]$. This is equal to the optimization problem of $\max_\alpha \alpha$ subject to $\mathcal{H}^c = \mathcal{H}^c \cap \Sigma_{\mathbf{x}}$, where $\mathcal{H}^c \cap \Sigma_{\mathbf{x}}$ is a union of polytopes. To judge whether a polytope is equal to a union of polytopes can be transformed to judge whether the set difference $\mathcal{H}^c \setminus (\mathcal{H}^c \cap \Sigma_{\mathbf{x}})$ or $\mathcal{H}^c \setminus \Sigma_{\mathbf{x}}$ is empty, which can be further transformed to be a linear programming problem, i.e., the polytope $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^n | H^x_{m\times n}\mathbf{x} \leq H^b_{m\times 1}\}$ is empty if and only if $\zeta > 0$ where $\zeta = \arg\min \zeta$ subject to $H^x_{m\times n}\mathbf{x} \leq H^b_{m\times 1} + \zeta \cdot 1_{m\times 1}$. The proposed polytopic algorithm is shown in Algorithm 6.5.

```
Algorithm 6.5:  The Proposed Polytopic Approximation Algorithm
In:  Σx, ε; Out:  P_a
1.  Initialize:  [a, b] = [0, 1] and m = (a + b)/2;
2.  H = Hull(Σx);
3.  H^c = m · H;
4.   while (b − a) < ε
5.        if H^c \ Σx = ∅
6.             a = m, b = b;
7.        else
8.             a = a, b = m;
9.        end
10.  end
11.  P_a = a ∗ H.
```

6.2.4 Nonlinear Robust MPC with A Contractive Sequence of Polytopic Robust Controllable Sets

Once all the polytopic robust controllable sets $\mathcal{P}_j^a (j = 1, \cdots, N)$ have been obtained, the robust controllability of any initial state can be judged accordingly. Assume that $\mathbf{x}(0) \in \cup_{j=0}^{N} \mathcal{P}_j^a$, i.e., the initial state is robustly controllable to the selected terminal set $\mathbb{T}$ in finite steps, then the one-step robust control via polytopic robust controllable sets is illustrated in Algorithm 6.6.

`Algorithm 6.6: MPC Via Polytopic Robust Controllable Sets`

`In:` $\mathbf{x}(0), \mathcal{P}_j^a (j = 0, \cdots, N)$; `Out:` $\mathbf{u}^{Optimal}(k|k), \mathbf{x}$

1. `Get the current state` $\mathbf{x}(k)$;
2. `if` $\mathbf{x}(k) \in \mathbb{T}$
3. **Switch to the related local stabilizing feedback control;**
4. `else`
5. **Find the** i**:** $i = \min_{j=1,\cdots,N}\{\mathbf{x}(k) \in \mathcal{P}_j^a\}$;
6. **Compute** $\mathbf{u}^{Optimal}(k|k)$ **with the contractive constraint (6.13);**
7. **Apply** $\mathbf{u}^{Optimal}(k|k)$ **to the system;**
8. `end`
9. **Return to 1 and repeat.**

According to Algorithm 6.6, the control algorithm measures the current state in `Step` 1 and then judges whether the system state has arrived in the terminal set in `Step` 2. A local stabilizing feedback control law is applied if the state has arrived in the terminal set; otherwise, the algorithm finds the smallest polytopic robust controllable set to which the current state belongs in `Step` 5; the one-step control scheme is formulated according to the strategy of robust MPC with unit control horizon and additional contractive constraint in `Step` 6, where $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$ is denoted to $\mathcal{P}_i^a$ according to the proposed algorithm of computing polytopic robust controllable sets. It can also be seen that any feasible solution that satisfies the contractive constraint is an effective control input for the system since such a control input is sufficient to guarantee the feasibility and stability of the closed-loop system. A feasible control input can be obtained as well via an interval-based minimax optimization algorithm addressed in Chapter 5 in a guaranteed numerical way.

6.2.5 Example

The illustrative example to be considered is (Khalil, 2002):

$$\begin{cases} x_1(k+1) = x_1(k) + 0.1x_1^2(k) - 0.1u(k) \cdot x_1^3(k) + 0.1x_2(k) \\ x_2(k+1) = x_2(k) + 0.1u(k) + w, \end{cases} \tag{6.14}$$

where the control is constrained to $|u| \leq 3$, the state variables are constrained to $|\mathbf{x}|_\infty \leq 1$ and $w \in [-0.01, 0.01]$ is the additive disturbance.

The terminal set $\mathbb{T}$ is selected to be a polytope, as shown in Fig. 6.6. The inner approximation of the first-step robust controllable set $\tilde{\mathcal{K}}(\mathbf{X}, \mathbb{T})$ computed via the addressed algorithm is also shown in Fig. 6.6 and $\mathbb{T} \subseteq \tilde{\mathcal{K}}(\mathbf{X}, \mathbb{T})$, which also demonstrates geometrically that the selected polytope is robust control invariant for the system.

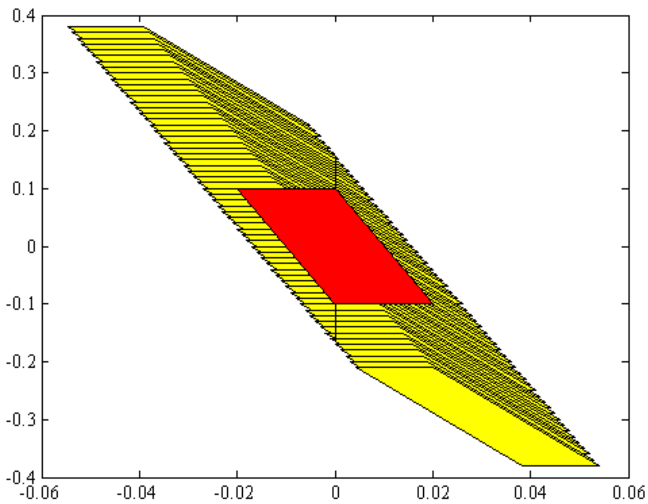


Figure 6.6: The inner approximation of the first-step robust controllable set

The obtained inner approximation of the first-step robust controllable set is a union of polytopes and it can be approximated innerly by one polytope via the proposed polytopic approximation algorithm addressed in this section or the revised polytopic approximation algorithm addressed in the former section, just as shown in Fig. 6.7.

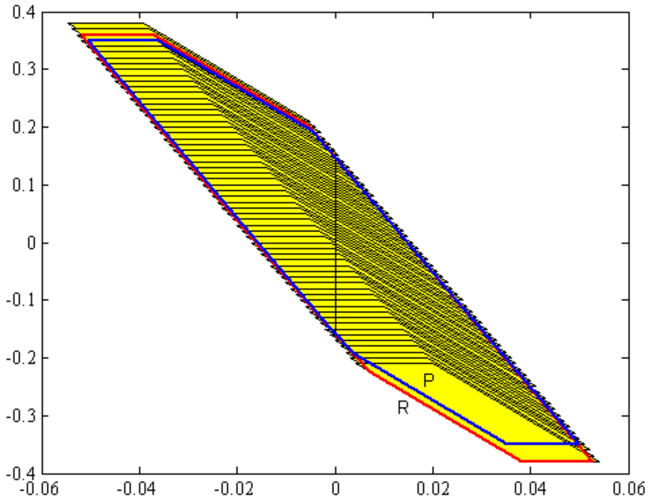


Figure 6.7: The comparison of two polytopic approximation algorithms

All polytopic robust controllable sets within the constrained state space can be computed iteratively. The maximal polytopic robust controllable set is reached when $\mathcal{P}^a_{39} = \mathcal{P}^a_{40}$ and all the computed polytopic robust controllable sets are shown in Fig. 6.8.

The proposed one-step control process for the system with the initial state $\mathbf{x}(0) = (-0.5, -0.8)$ is shown in Fig. 6.9, where the system state has been driven contractively along the computed polytopic robust controllable sets to the terminal robust control invariant polytope $\mathbb{T}$.

6.3 Nonlinear Robust MPC with A Contractive Sequence of Quasi Multi-step Robust Controllable Sets

[1] The recursive computation of one-step robust controllable sets for general constrained nonlinear uncertain discrete-time systems via interval arithmetic is not so

[1]The initial idea of computing multi-step robust controllable sets was proposed by Dr. T. Alamo during the external stay in Huelva and the author just embodied the idea by designing all algorithms needed, selecting a suitable example and presenting the simulation results.

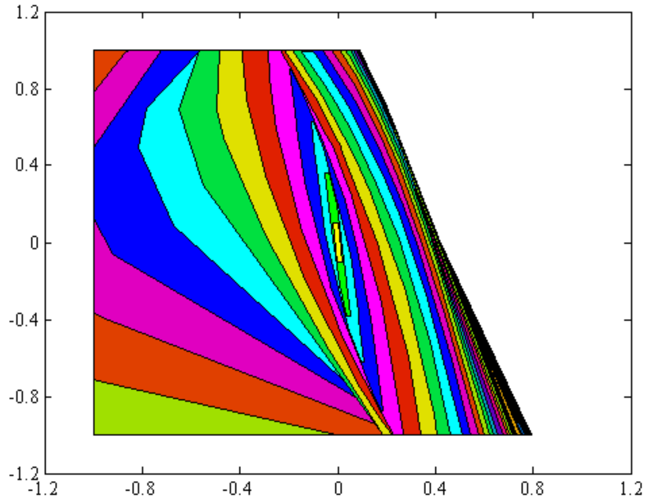


Figure 6.8: All the computed polytopic robust controllable sets

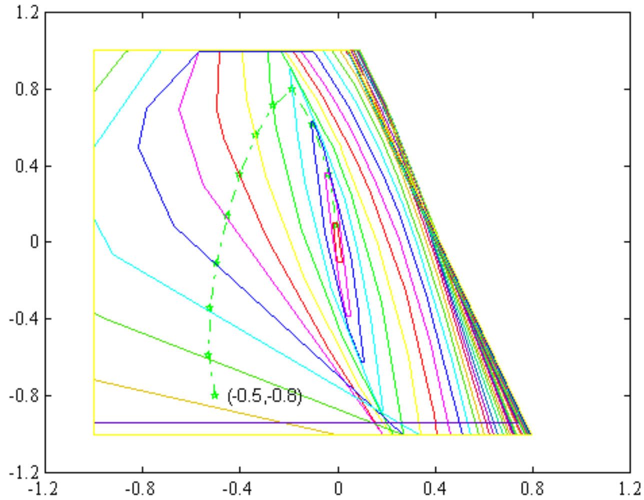


Figure 6.9: The one-step robust control process

efficient as the recursive computation of one-step robust controllable sets for general constrained linear systems via polytope geometry since the state space as well as the control space is needed to be bisected at each step. Furthermore, the one-step set polytopic approximation is also needed to be fulfilled at each step, which also increases the computation burden and contracts the obtained robust controllable sets frequently. One-step robust controllable sets are to be generalized to quasi multi-step robust controllable sets in this section. The structure of a specific kind of constrained nonlinear uncertain discrete-time systems is to be explored and it turns out that quasi multi-step robust controllable sets of such kind of systems can be computed directly and efficiently via an interval-based algorithm as well. The section is organized as follows: corresponding definitions and problem statement are introduced in Section 6.3.1; the computation of the outer bounds of reachable sets via zonotope geometry is addressed in Section 6.3.2; the detail of the proposed quasi multi-step robust controllable set approximation algorithm is illustrated in Section 6.3.3; the obtained polytopic representations of quasi i-step robust controllable sets are applied in robust MPC as a contractive sequence of sets in Section 6.3.4; finally, an illustrative example is given in Section 6.3.5.

6.3.1 The Concept of Quasi Multi-step Robust Controllable Sets

Consider the general constrained nonlinear uncertain discrete-time system:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)), k = 0, \cdots, \tag{6.15}$$

where $\mathbf{x}(k) in \mathbb{X} \subset \mathbb{R}^n$ is the system state; $\mathbf{u}(k) \in \mathbb{U} \subset \mathbb{R}^m$ is the control input; and $\mathbf{w}(k) \in \mathbb{W} \subset \mathbb{R}^l$ is the unknown disturbance. The set $\mathbb{X}$ is compact while $\mathbb{U}$ and $\mathbb{W}$ are closed.

The cost to compute i-step ($i \geq 2$) robust controllable set $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$ directly is somewhat heavy since it is a multi-dimensional optimization problem. In practice, $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$ can be approximated innerly via the following recursive procedure:

$$\tilde{\mathcal{K}}_{i+1}(\mathbf{X}, \mathbb{T}) = \tilde{\mathcal{K}}_1(\mathbf{X}, \tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})), \tag{6.16}$$

where $\tilde{\mathcal{K}}_0(\mathbf{X}, \mathbb{T}) = \mathbb{T}$. If the terminal set $\mathbb{T}$ is selected to be a robust control invariant set Ω, then $\tilde{\mathcal{K}}_i(\mathbf{X}, \Omega)$ is also a robust control invariant set with

$$\tilde{\mathcal{K}}_i(\mathbf{X}, \Omega) \subseteq \tilde{\mathcal{K}}_{i+1}(\mathbf{X}, \Omega), \tag{6.17}$$

where $\tilde{\mathcal{K}}_i(\mathbf{X}, \Omega)$ is also referred to as the i-step robust stabilisable set (Kerrigan, 2000). Such a geometric property of robust stabilisable sets can be applied in MPC since a feasible control sequence, which guarantees to drive the system from any initial state within $\tilde{\mathcal{K}}_i(\mathbf{X}, \Omega)$ to Ω in i steps, can be obtained.

The interval-based one-step recursive procedure for computing controllable sets addressed in Chapter 4 can be extended to compute $\tilde{\mathcal{K}}_i(\mathbf{X}, \Omega)$ as well. However, a direct extension of such a branch-and-bound algorithm is not efficient since the subboxes of the state space which do not belong to $\tilde{\mathcal{K}}_1(\mathbf{X}, \tilde{\mathcal{K}}_{i-1}(\mathbf{X}, \mathbb{T}))$ are to be discarded at each step although they possibly belong to $\tilde{\mathcal{K}}_n(\mathbf{X}, \tilde{\mathcal{K}}_{i-1}(\mathbf{X}, \mathbb{T}))$, where $n > 1$. Furthermore, the computed one-step robust controllable set is to be approximated innerly by one polytope at each step, which contracts the computed one-step robust controllable sets seriously and frequently. In order to improve the efficiency of the former interval-based recursive procedure and decrease the frequency of contracting the computed robust controllable sets, an alternative approach is to compute inner approximations of multi-step robust controllable sets directly. The following concept of quasi multi-step robust controllable sets is proposed to demonstrate the new approach:

Definition 6.1 (Quasi Robust Controllable Set) The quasi i-step robust controllable set $\tilde{\mathcal{K}}_i^q(\mathbf{X}, \mathbb{T})$ is the set of states in $\mathbf{X}$ within which certain admissible control sequence $\{\mathbf{u}(k) \in \mathbf{U}\}_0^{i-1}$ for each state guarantees to drive the system to the terminal set $\mathbb{T} \subset \mathbb{R}^n$ in i steps while keeping the evolution of the state inside $\mathbf{X}$, for all allowable disturbance sequences, i.e., $\tilde{\mathcal{K}}_i^q(\mathbf{X}, \mathbb{T}) \triangleq \{\mathbf{x}(0) \in \mathbf{X} | \{\mathbf{u}(k) \in \mathbf{U}\}_0^{i-1} : \{\mathbf{x}_k \in \mathbf{X}\}_0^{i-1}, \mathbf{x}_i \in \mathbb{T}, \forall \{\mathbf{w}_k \in \mathbf{W}\}_0^{i-1}\}$.

Obviously, $\tilde{\mathcal{K}}_i^q(\mathbf{X}, \mathbb{T})$ is an inner approximation of $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$ since the control sequences used in computing $\tilde{\mathcal{K}}_i^q(\mathbf{X}, \mathbb{T})$ might not be as globally optimal as the control sequences used in computing $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$. However, $\tilde{\mathcal{K}}_i^q(\mathbf{X}, \mathbb{T})$ can approach to $\tilde{\mathcal{K}}_i(\mathbf{X}, \mathbb{T})$ closely when the control sequences used in computing it are well selected.

Certain control sequences used in computing quasi multi-step robust controllable sets can be well selected sequentially for a specific kind of constrained nonlinear uncertain discrete-time systems via linear programming, i.e., the systems with the following structure:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{w}(k)) + B\mathbf{u}(k), \tag{6.18}$$

where $\mathbf{x}(k) \in \mathbb{X} \subset \mathbb{R}^n$, $\mathbf{u}(k) \in \mathbb{U} \subset \mathbb{R}^m$, $\mathbf{w}(k) \in \mathbb{W} \subset \mathbb{R}^l$ and $\mathbb{X}, \mathbb{U}, \mathbb{W}$ are all compact sets described by boxes of proper dimensions. The terminal set Ω is selected to be a robust control invariant set represented by a polytope containing the origin:

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n | T_{p\times n}\mathbf{x} \leq 1_{p\times 1}\}. \tag{6.19}$$

The control sequence $\{\mathbf{u}(k) \in \mathbf{U}\}_0^{i-1}$ used in testing whether it could drive a subbox $\mathbf{X}_0$ of the constrained state space $\mathbf{X}$ to the terminal set Ω in i steps for all uncertain cases $\forall \mathbf{w} \in \mathbf{W}$ while computing $\tilde{\mathcal{K}}_i^q(\mathbf{X}, \Omega)$ can be obtained iteratively via linear programming of the following problem:

$$J(\mathbf{X}(k), \mathbf{u}(k)) = \min_{\{\mathbf{u}(k)\in\mathbf{U},\xi\}} \xi \tag{6.20}$$

subject to

$$T\mathbf{f}(\mathbf{X}(k), \mathbf{W}) + TB\mathbf{u}(k) \leq \xi, \tag{6.21}$$

where $\mathbf{f}(\mathbf{X}(k), \mathbf{W})$ is to be represented by a box bounding the zonotope inclusion of its centered inclusion function and $\mathbf{X}(k)(k > 1)$ is to be represented by the zonotope inclusion of the centered inclusion function of $\mathbf{f}(\mathbf{X}(k-1), \mathbf{W}) + B\mathbf{u}(k-1)$. Physically, the linear programming is aimed to find an optimal control input $\mathbf{u}(k)$ which is most likely to drive the system state into the target polytope Ω at each step. The evolution of the system is to be bounded by zonotope inclusions with reduced wrapping effects.

6.3.2 The Computation of the Outer Bounds of Reachable Sets via Zonotope Geometry

According to the problem statement, in order to compute quasi i-step robust controllable set $\tilde{\mathcal{K}}_i^q(\mathbf{X}, \Omega)$ for the system (6.18), the admissible state space $\mathbf{X}$ needs to be bisected into subboxes and then the control inputs $\{\mathbf{u}(k) \in \mathbf{U}\}_0^{i-1}$ are to be obtained sequentially for each subbox $\mathbf{X}_0$ of $\mathbf{X}$ via linear programming. $\mathbf{X}_0$ belongs to $\tilde{\mathcal{K}}_i^q(\mathbf{X}, \Omega)$ if the computed control sequence $\{\mathbf{u}(k) \in \mathbf{U}\}_0^{i-1}$ can drive all the states in $\mathbf{X}_0$ to the terminal set Ω in i steps. The following definition of reachable sets is concerned while computing the evolution of the system.

Definition 6.2 (Reachable Set) Consider the system $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k))$ with the initial set of states $\mathbb{X}_0$ and certain admissible sequence of control inputs $\{\mathbf{u}(k) \in \mathbb{U}\}_0^{i-1}$, then the reachable sets $\{\mathbf{X}(k)\}_1^i$ of the system are obtained from the recursion: $\mathbf{X}(k+1) = \mathbf{f}(\mathbf{X}(k), \mathbf{u}(k), \mathbb{W})$.

Physically, $\mathbb{X}(i)$ is the set of all states that can be reached by the evolution of the uncertain system at step i applying the sequence of control inputs $\{\mathbf{u}(k) \in \mathbb{U}\}_0^{i-1}$. If $\mathbb{X}(i) \subseteq \Omega$, then the initial set of states $\mathbf{X}_0$ belongs to the quasi i-step robust controllable set $\tilde{\mathcal{K}}_i^q(\mathbb{X}, \Omega)$.

The exact computation of $\{\mathbf{X}(k)\}_1^i$ is a difficult task. The natural inclusion function of $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k))$ was used to calculate the outer bounds of reachable sets. Although it is an efficient solution, the direct application of natural inclusion functions for computing $\{\mathbf{X}(k)\}_1^i$ might produce large over-estimations of extract reachable sets. An alternative way of obtaining the outer bounds of reachable sets with reduced wrapping effects is to use zonotopes to bound the evolution of general nonlinear uncertain discrete-time systems with reduced wrapping effects, just as discussed in Chapter 4. An illustrative example for demonstrating the reduced wrapping effects is shown in Fig. 6.10, where an initial subbox of the state space along with a control sequence is proved to be driven to a selected terminal set in four steps using zonotope evolutions while the same initial subbox of the state space along with the same control sequence cannot be proven to be driven to the selected terminal set in four steps.

6.3.3 The Computation of Quasi Multi-step Robust Controllable Sets via Interval Analysis and Zonotope Geometry

Applying the iterative linear programming of control inputs in Section 6.3.1 and zonotope evolutions of the system state in Section 6.3.2, the detailed algorithm for computing quasi i-step robust controllable set is shown in Algorithm 6.7, where $\mathbf{X}, \mathbf{U}, \mathbf{W}$ are the domains of state, control and disturbance described by boxes of proper dimensions, Ω is the terminal robust control invariant set, i is the number of steps to be computed, and ε is the bound of error tolerance. With these inputs, the algorithm returns an inner approximation of $\tilde{\mathcal{K}}_n^q(\mathbf{X}, \mathbb{T})$ represented by a union of subboxes of $\mathbf{X}$,

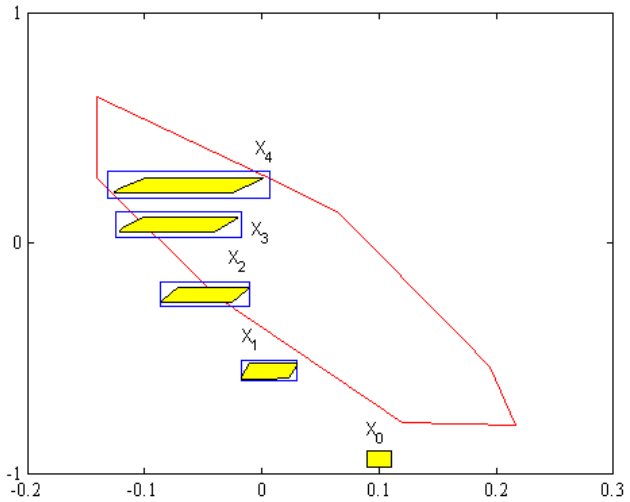


Figure 6.10: The comparison of zonotope evolution and interval evolution

which is to be denoted as Σ_i. The principle of the algorithm is as follows: for a subbox $\mathbf{X}_0$ of $\mathbf{X}$, the algorithm first tests whether the i-step reachable set with $\forall \mathbf{u} \in \mathbf{U}$ intersects with the terminal set Ω in `Step` 5; if the intersection with Ω is empty, the subbox $\mathbf{X}_0$ is to be discarded since it is impossible to drive this subbox to the target set in i steps for all admissible control input $\mathbf{u} \in \mathbf{U}$; if the intersection computed is not empty, then it is possible to drive this subbox $\mathbf{X}_0$ to the terminal set Ω in i steps and the algorithm computes the evolution of the system state with a control sequence $\{\mathbf{u}(k) \in \mathbf{U}\}_0^{i-1}$ obtained iteratively via linear programming from `Step` 6 to `Step` 16; if all system states starting from the initial subbox $\mathbf{X}_0$ have already arrived to the target set Ω in less than i steps or in exactly i steps, as stated in `Step` 10, the subbox $\mathbf{X}_0$ belongs to $\tilde{\mathcal{K}}_i^q(\mathbf{X}, \mathbb{T})$ and it is to be added to Σ_i in `Step` 11; Otherwise, the subbox $\mathbf{X}_0$ is to be bisected in `Step` 20 for further judgements if its width is not beyond the bound of error tolerance ε, which is on the contrary of `Step` 17. It is worthy to note that the evolutions of the system state in both `Step` 4 and `Step` 9 are to be computed via zonotope inclusions, where less wrapping effects are anticipated than corresponding evolutions computed via interval arithmetic.

Algorithm 6.7: Quasi i-step Robust Controllable Set Approximation Algorithm

In: $\mathbf{X}, \mathbf{U}, \mathbf{W}, \Omega, i, \varepsilon$; Out: Σ_i

1. Stack $= \mathbf{X}, \Sigma_i = \emptyset$;
2. **while** *Stack* $\neq \emptyset$
3. Pop out a $\mathbf{X}_0$ from *Stack*;
4. Compute $\mathbb{X}(k)$ recursively by
$\mathbb{X}(k+1)\mathbf{f}(\mathbb{X}(k), \mathbf{W}) + B\mathbf{U}, k = 0, \cdots, i-1$;
5. if $\mathbb{X}(i) \cap \Omega = \emptyset$, return to Step 3;
6. for $k = 0 : 1 : i-1$
7. if $\mathbf{X}(k) \subseteq \mathbf{X}$
8. $\mathbf{u}(k) = \min_{\{\mathbf{u}(k), \xi\}} \xi$;
9. $\mathbf{X}(k+1) = \mathbf{f}(\mathbf{X}(k), \mathbf{W}) + B\mathbf{u}(k)$;
10. *if* $\mathbf{X}(k+1) \subseteq \Omega$
11. Add $\mathbf{X}_0$ to Σ_i and return to Step 3;
12. *end*
13. else
14. Break and transfer to Step 17;
15. end
16. end
17. if width$(\mathbf{X}_0) \leq \varepsilon$
18. Return to Step 3;
19. else
20. Bisect $\mathbf{X}_0$ into L$\mathbf{X}_0$ and R$\mathbf{X}_0$, push them on Stack and return to Step 3;
21. end
22. **end**

Taking the computation of quasi 3-step robust controllable sets as an example, the four cases to be encountered according to the algorithm are shown in Fig. 6.11, where the subbox $\mathbf{A}_0$ is to be discarded according to its evolution with $\forall \mathbf{u} \in \mathbf{U}$, just as stated in Step 5; the subbox $\mathbf{B}_0$ is to be bisected for further judgement if its width is bigger than ε, just as in Step 20, or it is also to be discarded if its width is not bigger than ε, just as in Step 17; the subbox $\mathbf{C}_0$ and $\mathbf{D}_0$ belong to $\tilde{\mathcal{K}}_3^q(\mathbf{X}, \Omega)$ since the control sequence applied can drive the system from any initial state in them to the target set

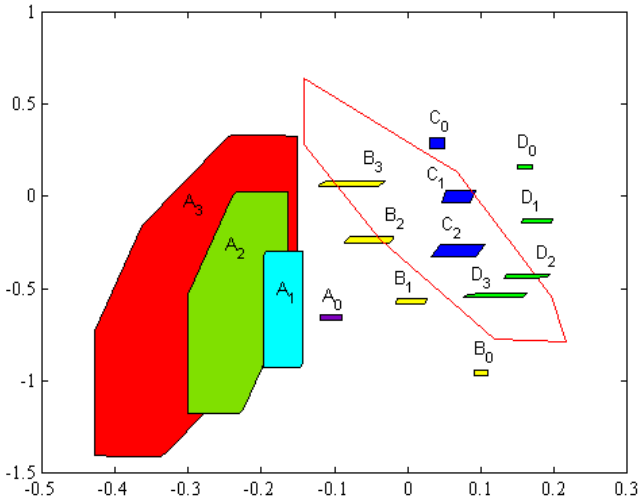


Figure 6.11: An illustrative example of computing quasi 3-step robust controllable set

Ω in less than 3 steps and in exactly 3 steps, respectively. Zonotopes are applied to bound the evolution of the system state for each case, which can also be seen in the figure.

6.3.4 Multi-step Robust Control via Constrained Minimax Optimization

The second quasi i-step robust controllable set, which is to be denoted as $\tilde{\mathcal{K}}_i^{q_2}(\mathbf{X},\Omega)$, can be computed similarly once the polytopic approximation $\mathcal{P}_a$ of the first quasi i-step robust controllable set $\tilde{\mathcal{K}}_i^{q_1}(\mathbf{X},\Omega)$ has been obtained via the polytopic approximation algorithm illustrated in Section 6.1. Then the inner approximation of the maximal quasi i-step robust controllable set $\tilde{\mathcal{K}}_i^{q_\infty}(\mathbf{X},\Omega)$ within the constrained state space $\mathbf{X}$ can be obtained as well through computing quasi i-step robust controllable sets recursively. The system is robustly controllable to the target set Ω in finite steps if the initial state is within the maximal quasi i-step robust controllable set $\tilde{\mathcal{K}}_i^{q_\infty}(\mathbf{X},\Omega)$, which is assumed to be reached when $\tilde{\mathcal{K}}_i^{q_M}(\mathbf{X},\Omega) = \tilde{\mathcal{K}}_i^{q_{M+1}}(\mathbf{X},\Omega)$. The computed quasi i-step robust controllable sets can be applied in robust MPC as a contractive sequence of sets, i.e., the control inputs for the constrained nonlinear uncertain discrete-time system can be obtained through the strategy of MPC with the known feasible control horizon i and additional contractive constraint, which is to solve the following optimization problem iteratively:

$$J(\mathbf{x}(k), \{\mathbf{u}(k+j|k)\}_{j=0}^{i-1}) = \min_{\{\mathbf{u}(k+j|k)\}_{j=0}^{i-1}} \{\mathbf{x}^T(k+i|k)P_0\mathbf{x}(k+i|k) + \sum_{j=1}^{i-1}\mathbf{x}^T(k+j|k)Q\mathbf{x}(k+j|k) + \sum_{j=0}^{i-1}\mathbf{u}^T(k+j|k)R\mathbf{u}(k+j|k)\} \tag{6.22}$$

subject to

$$\begin{cases} \mathbf{x}(k+j|k) \in \mathbf{X}, j = 1, \cdots, i-1 \\ \mathbf{x}(k+i|k) \in \mathcal{K}_i^{q_{m-1}}(\mathbf{X}, \Omega), \end{cases} \tag{6.23}$$

where $\mathbf{x}(k) \in \tilde{\mathcal{K}}_i^{q_m}(\mathbf{X}, \Omega)$, but $\mathbf{x}(k)$ does not belong to $\tilde{\mathcal{K}}_i^{q_{m-1}}(\mathbf{X}, \Omega)$; P_0, Q and R are weighted positive definite matrices; and $\{(\mathbf{u}(k+j|k))^{Optimal}\}_{j=0}^{i-1}$ is the optimal control sequence of i steps, which can be obtained in a reliable and global way by the interval-based global optimization algorithm. The resulting control algorithm of MPC using quasi i-step robust controllable sets as a contractive sequence of sets is shown in Algorithm 6.8, where all the control inputs of i steps obtained at each iteration are to be applied to the system sequentially. Such an approach is guaranteed to be feasible and stable according to the property of $\tilde{\mathcal{K}}_i^{q_n}(\mathbf{X}, \Omega)$.

Algorithm 6.8: MPC Via Quasi i-step Robust Controllable Sets

In: $\mathbf{x}(k), \{\tilde{\mathcal{K}}_i^{q_m}\}_{m=1}^{M}, \Omega$; Out: $\{(\mathbf{u}(k+j|k))^{Optimal}\}_{j=0}^{i-1}$

1. Get the current state $\mathbf{x}(k)$;
2. if $\mathbf{x}(k) \in \Omega$
3. Switch to the local stabilizing feedback control;
4. else
5. Find the j: $j = \min_{m=1,\cdots,M}\{\mathbf{x}(k) \in \tilde{\mathcal{K}}_i^{q_m}\}$;
6. Compute $\{(\mathbf{u}(k+j|k))^{Optimal}\}_{j=0}^{i-1}$;
7. Apply $\{(\mathbf{u}(k+j|k))^{Optimal}\}_{j=0}^{i-1}$ to the system;
8. end
9. Return to Step 1 and circulate.

6.3.5 Example

As an illustrative example, the system to be considered is (Khalil, 2002):

$$\begin{cases} x_1(k+1) = x_1(k) + 0.1x_2(k) + w(k) \\ x_2(k+1) = x_2(k) + 0.1[x_1^2(k) + p(k)x_2^2(k) + u(k)] \end{cases} \tag{6.24}$$

where the control is constrained to $|u| \leq 3$; the state variables are constrained to $|\mathbf{x}|_\infty \leq 1$; the additive disturbance is constrained to $|\mathbf{w}|_\infty \leq 0.01$ and the uncertain parameter is $p \in [0.9, 1.0]$.

The initial robust control invariant set is designed to be a polytope Ω, which is shown in Fig. 6.12. It is enlarged from a low-complexity polytope as well as a control invariant sliding domain, which is also shown in Fig. 6.12. The selected set Ω can be demonstrated geometrically to be robust control invariant through computing corresponding quasi one-step robust controllable set $\tilde{\mathcal{K}}_1^q(\mathbf{X}, \Omega)$, which satisfies $\Omega \subseteq \tilde{\mathcal{K}}_1^q(\mathbf{X}, \Omega)$. The computed first quasi one-step robust controllable set is a union of boxes as well as a union of polytopes. One polytope to approximate the union of boxes innerly is obtained via the revised polytopic approximation algorithm and it is shown in Fig. 6.12 as well.

The first quasi 2-step robust controllable set can be computed accordingly according to the algorithm with the selected initial robust control invariant set Ω shown in Fig. 6.13, where interval arithmetic is applied to branch and bound the constrained state space and linear programming is applied to compute optimal control inputs at each step. The computed first quasi 2-step robust controllable set and its polytopic approximation are shown in Fig. 6.13. The polytopic approximation of the first quasi 2-step robust controllable set is also compared with two polytopic quasi one-step robust controllable sets computed recursively from the target set Ω, as shown in Fig. 6.13 as well. It can be seen that an approximation of the 2-step robust controllable set is obtained directly by applying the quasi 2-step robust controllable set approximation algorithm and thus an intermediate polytopic approximation is avoided comparative to the recursive one-step approach, which is specially beneficial to high-dimensional cases where corresponding computations for vertex numerations and complementary sets are heavy.

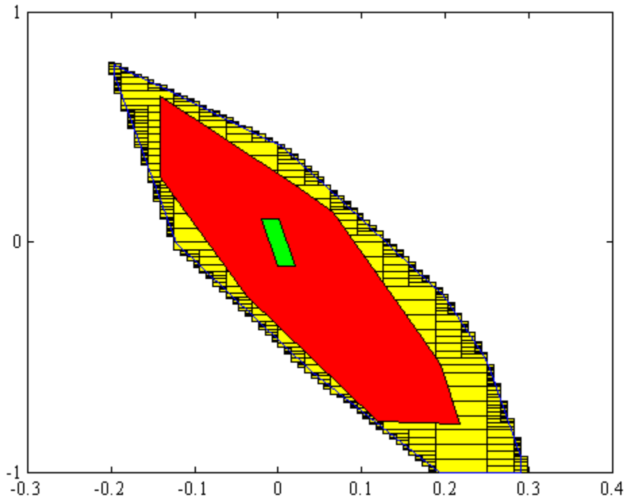


Figure 6.12: The terminal set, the quasi one-step set and its polytopic approximation

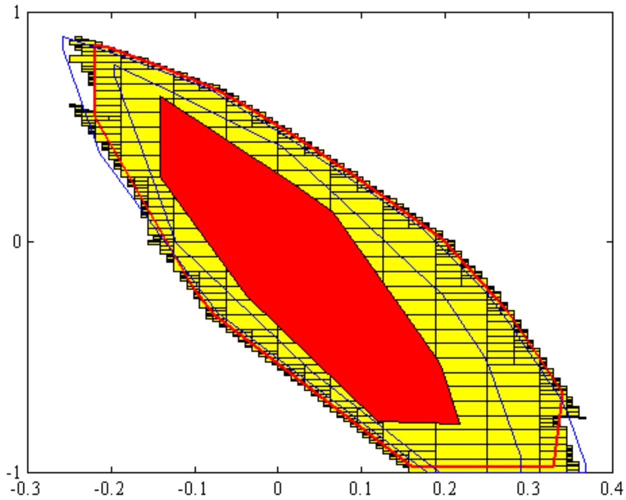


Figure 6.13: The comparison of the one-step and two-step approach

The second quasi 2-step robust controllable set starting from the polytopic representation of the computed first quasi 2-step robust controllable set can be computed similarly and thus the inner approximation of the maximal quasi 2-step robust controllable set within the constrained state space can be obtained as well by computing quasi 2-step robust controllable sets recursively, which is reached at $\mathcal{P}^a_{20} = \mathcal{P}^a_{21}$ in this case. The computed polytopic quasi 2-step robust controllable set sequence up to the maximal polytopic quasi 2-step robust controllable set is shown in Fig. 6.14.

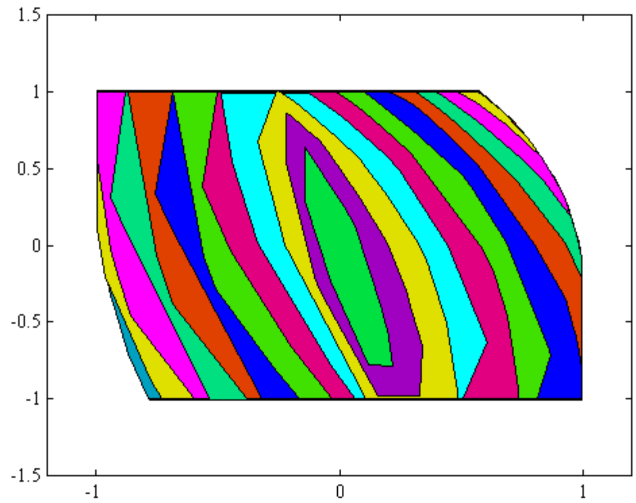


Figure 6.14: The polytopic quasi 2-step robust controllable set sequence

The computed polytopic quasi 2-step robust controllable sets can be applied in robust MPC as a contractive sequence of sets and the feasible control horizon is equal to 2. The control inputs are to drive the system sequentially along the computed polytopic quasi 2-step robust controllable sets to the initial robust control invariant set. The two-step robust control processes for the system with the initial state $\mathbf{x}(0) = (-0.7, -0.9)$ and $\mathbf{x}(0) = (0.77, 0.6)$ are shown in Fig. 6.15, where $P_0 = 1, R = 0.1, Q = 0$, the additive disturbance w and the uncertain parameter p are random number series within their domains during the control processes. It can be seen that the control scheme guarantees to drive the system state contractively to the designed initial robust control invariant set despite of the additive disturbance and the uncertain parameter.

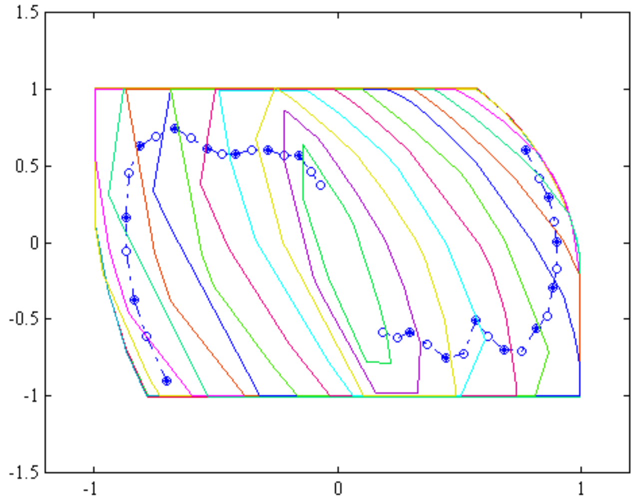


Figure 6.15: The two-step robust control processes

6.4 Summary

Interval analysis, polytope geometry and zonotope geometry have been combined in this chapter to improve the efficiency of nonlinear robust contractive MPC with a contractive sequence of polytopic robust controllable sets. Concretely, the computed robust controllable sets via an interval-based branch-and-bound algorithm have been approximated innerly by polytopes. The structures of two types of nonlinear systems have been explored further to reduce the dimension needed to be bisected and the concept of quasi multi-step robust controllable sets has been proposed to reduce the number of polytopic approximations during the computation of polytopic robust controllable sets.

Chapter 7

Applications of Contractive MPC

The addressed two types of contractive MPC in the former chapters are to be applied to control a Micro Robot World Cup Soccer Tournament (MiroSot) robot and a Continuous Stirred-Tank Reactor (CSTR), respectively. Concretely, the control of a MiroSot robot via contractive MPC with compulsory contractive constraint is to be addressed in Section 7.1 and the control of a CSTR via contractive MPC with a contractive sequence of polytopic controllable sets is to be addressed in Section 7.2. The two practical examples used in this chapter are to demonstrate the applicability of the addressed computationally reliable approaches of contractive MPC.

7.1 Example A — MiroSot Robot Control

Robot soccer has attracted more and more interests as an intriguing test bed for intelligent control of dynamic systems in a multi-agent collaborative environment (Messom, 1998). It is also a typical multi-disciplinary project, which involves in-depth knowledge in the fields of motion control, radio communication, image processing and strategy programming. The use of global vision has been increasing in robot soccer because of the emphasis on the coordination and cooperation of multiple robots (Pereira et al., 2000a). In such a scenario, playing robots are controlled by a centralized computing system through the visual information received from a camera mounted above the playground. The motion control of such a configuration is usually difficult due to large time delays in the image processing stage and the lack of local sensors. Various methods have been applied to control mobile robots (Watanabe, 1996). Model predictive control has also been applied to robotic control because of their inherent capability of prediction for future states of time-delay systems in a straightforward way (Messom et al., 2003; Pereira et al., 2000b). However, predictors were only used for predicting the state of the target or the robot and obstacle avoidance as well as path planning was not considered in their control algorithms.

In this section, an integrated predictive control algorithm is proposed to control a MiroSot robot using global vision, where the stability of the time-delay system is to be guaranteed by incorporating additional contractive constraint and obstacle avoidance as well as path planning is realized automatically by incorporating additional distant constraints into the open-loop optimization of control inputs. This section is organized as follows: first, in Section 7.1.1, the dynamic model of the robot is deduced by taking into account the whole process, which includes the vision system, the dynamic system and the transmission system; then in Section 7.1.2, a predictive control algorithm with automatic obstacle avoidance and inherent path planning is proposed for the control of the resulting nonlinear time-delay dynamic system; the simulation result of the proposed algorithm is provided in Section 7.1.3.

7.1.1 Modeling

The variables measured by the global vision system are the position (x, y) of the geometric center of the robot and the angle θ between the main axis of the robot and the axis X of the playing field, as shown in Fig. 7.1.

Applying the Newton's second law, a dynamic model for the robot can be derived as:

$$\begin{cases} x(k)=x(k-1)+[v(k-1)T+(a_1u_1(k-d)+a_2u_2(k-d)+F_{a1}(k-1)+F_{a2}(k-1))T^2/m]\cos(\theta(k-1)) \\ y(k)=y(k-1)+[v(k-1)T+(a_1u_1(k-d)+a_2u_2(k-d)+F_{a1}(k-1)+F_{a2}(k-1))T^2/m]\sin(\theta(k-1)) \\ \theta(k)=\theta(k-1)+\omega(k-1)+(a_1u_1(k-d)-a_2u_2(k-d))T^2/(2I), \end{cases} \tag{7.1}$$

where v and ω are the linear and angular velocity of the robot; F_{a1} and F_{a2} are the friction forces at the contact line between the bearing and the floor; m represents the robot mass and I is the inertia moment around the robot's center of the mass G; the robot is commanded by two signals, u_1 and u_2, which represent the magnitudes of the voltage at the right and left motors, respectively; and the time delay between

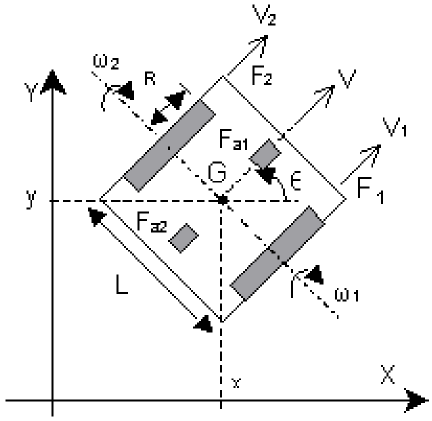


Figure 7.1: The configuration of the MiroSot robot

the time of the action of these signals and the visualization of its effects is denoted by d.

The model in (7.1) is a physically-motivated approximate description of the system. One of the problems in the model is that some terms such as friction forces, is difficult to obtain; another relevant problem is that the velocity terms are not directly measured by the vision system. These problems can be circumvented by an adequate parametrization of the model followed by a consistent parameter estimation. The physical model can therefore be re-written as:

$$\begin{cases} x(k)=x(k-1)+c_1 v_x(k-1)T+(c_2u_1(k-d)+c_3u_2(k-d))\cos(\theta(k-1)) \\ y(k)=y(k-1)+c_4 v_y(k-1)T+(c_5u_1(k-d)+c_6u_2(k-d))\sin(\theta(k-1)) \\ \theta(k)=\theta(k-1)+c_7\omega(k-1)+c_8u_1(k-d)+c_9u_2(k-d), \end{cases} \tag{7.2}$$

where v_x and v_y are the projections of the linear velocity v on the axis X and Y, respectively. It is important to note that the mass m, the sampling time T and the friction forces F_{a1}, F_{a2} are grouped together in parameters $c_i (i = 1, \cdots, 9)$. The velocities v_x, v_y and θ can be roughly approximated by:

$$v_x(k-1)=\frac{x_{k-1}-x_{k-12}}{T}, v_y(k-1)=\frac{y_{k-1}-y_{k-12}}{T}, \theta(k-1)=\frac{\omega(k-1)-\omega(k-2)}{T}. \tag{7.3}$$

Then the model (7.2) can be represented as an auto-regressive model with exogenous inputs of the form:

$$\begin{cases} x(k)=a_{1x}x(k-1)+a_{2x}x(k-2)+(b_{1x}u_1(k-d)+b_{2x}u_2(k-d))\cos(\theta(k-1)) \\ y(k)=a_{1y}y(k-1)+a_{2y}y(k-2)+(b_{1y}u_1(k-d)+b_{2y}u_2(k-d))\sin(\theta(k-1)) \\ \theta(k)=a_{1\theta}\theta(k-1)+a_{2\theta}\theta(k-2)+b_{1\theta}u_1(k-d)+b_{2\theta}u_2(k-d). \end{cases} \tag{7.4}$$

Using the extended least-square method for estimation, the following model is obtained:

$$\begin{cases} x(k)=1.3724x(k-1)-0.3724x(k-2)+(0.0096u_1(k-d)+0.0119u_2(k-d))\cos(\theta(k-1)) \\ y(k)=1.221y(k-1)-0.221y(k-2)+(0.0117u_1(k-d)+0.0121u_2(k-d))\sin(\theta(k-1)) \\ \theta(k)=1.121\theta(k-1)-0.121\theta(k-2)-0.00396u_1(k-d)+0.00422u_2(k-d). \end{cases} \tag{7.5}$$

7.1.2 Contractive MPC with Compulsory Contractive Constraint

Based on the obtained model in (7.5), the control task of robotic interception is to catch the target with a proper orientation; meanwhile, the robot does not collide with obstacles during its movement to the target. The position of an obstacle is denoted by (x_{ob}, y_{ob}). Thus the corresponding control problem to be solved is to compute a sequence of control inputs $\mathbf{u}(k+i|k) = (u_1(k+i|k), u_2(k+i|k))$ that drives the robot from its current state (x_k, y_k, θ_k) to the desired state (x_d, y_d, θ_d) with additional constraints for keeping a distance from all obstacles and guaranteeing the stability of the closed-loop system. The desired state of the robot is to be determined by the position of the target and the angle of interception.

According to the principle of contractive MPC with compulsory contractive constraint, the sequence of control inputs $\{u_1(k+i|k), u_2(k+i|k)\}$ is to be obtained through minimizing the following cost function on the basis of satisfying all imposed constraints:

$$J(\mathbf{x}(k), \{\mathbf{u}(k+i|k)\}_{i=0}^{n-1}) = \min_{\{\mathbf{u}_{k+i|k} \in \mathbf{U}\}_{i=0}^{n-1}} [\mathbf{x}(k+n|k)^T P_0 \mathbf{x}(k+n|k) + \\ \sum_{i=1}^{n-1} \mathbf{x}^T(k+i|k) Q \mathbf{x}(k+i|k) + \sum_{i=0}^{n-1} (\mathbf{u}(k+i|k))^T R \mathbf{u}(k+i|k)] \tag{7.6}$$

subject to

$$\begin{cases} |\mathbf{u}(k+i|k)|_\infty \leq U \\ |[x(k+i|k)\ y(k+i|k)] - [x_{ob}\ y_{ob}]|_2 \leq D \\ |[x(k+n|k)\ y(k+n|k)] - [x_d\ y_d]|_2 < \alpha |[x(k)\ y(k)] - [x_d\ y_d]|_2, \alpha \in (0,1], \end{cases} \tag{7.7}$$

where n denotes to the length of the control horizon; U denotes the maximum absolute value of control signals; D denotes the minimum distance between the robot and the obstacle; and α determines the degree of state contraction for each open-loop optimization (Wan et al., 2004c).

7.1.3 Simulation

The simulation result for the proposed algorithm is shown in Fig. 7.2, where $n = 12$ and $\alpha = 0.95$. It can be seen from the simulation result that the MiroSot robot has been driven to the target place with the desired orientation and automatic obstacle avoidance. It is worthy to note that the control strategy has also the capacity of inherent path planning as it figures out a feasible path automatically with the consideration of all imposed constraints (Wan et al., 2004a).

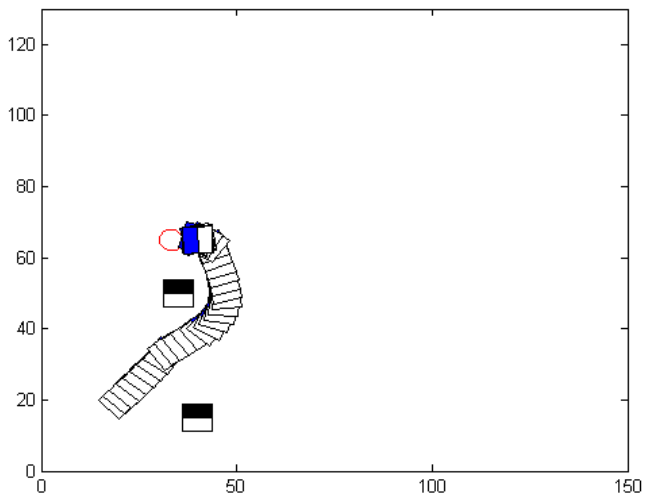


Figure 7.2: The control of a MiroSot robot with automatic obstacle avoidance

7.2 Example B — CSTR Control

Contractive MPC with a contractive sequence of polytopic controllable sets is applied to control a highly nonlinear model of a Continuous Stirred-Tank Reactor (CSTR) (Limon et al., 2003; Magni et al., 2001), which is shown in Fig. 7.3.

Assuming constant liquid volume, the CSTR for an exothermic, irreversible reaction, $\mathbf{A} \rightarrow \mathbf{B}$, is described by the following dynamic model based on a component

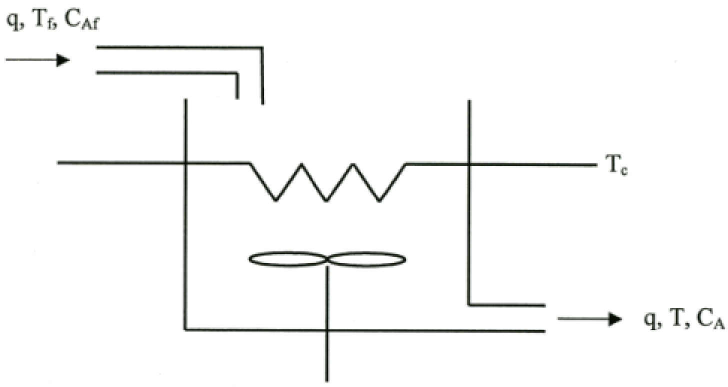


Figure 7.3: The configuration of the CSTR

balance for the reactant **A** and an energy balance:

$$\begin{cases} \dot{C}_A = \frac{q}{V}(C_{Af} - C_A) - k_0 \exp(-\frac{E}{RT})C_A, \\ \dot{T} = \frac{q}{V}(T_f - T) + \frac{(-\triangle H)}{\rho C_p} k_0 \exp(-\frac{E}{RT})C_A + \frac{UA}{V\rho C_p}(T_c - T), \end{cases} \tag{7.8}$$

where C_A is the concentration of **A** in the reactor, T is the reactor temperature, and T_c is the temperature of the coolant stream. The constraints are $280\mathbf{K} \leq T_c \leq 370\mathbf{K}$, $280\mathbf{K} \leq T \leq 370\mathbf{K}$ and $0 \leq C_A \leq 1mol/l$. The objective is to regulate C_A and T by manipulating T_c. The nominal operating conditions, which correspond to an unstable equilibrium $C_A^{eq} = 0.5mol/l, T^{eq} = 350\mathbf{K}, T_c^{eq} = 300\mathbf{K}$ are: $q = 100l/min, C_{Af} = 1mol/l, T_f = 350\mathbf{K}, V = 100l$, $\rho = 1000g/l, C_p = 0.239\mathbf{J}/g\mathbf{K}$, $\triangle H = -5 \times 10^4\mathbf{J}/mol, E/R = 8750\mathbf{K}$, $k_0 = 7.2 \times 10^{10}min^{-1}, UA = 5 \times 10^4\mathbf{J}/min\mathbf{K}$. The nonlinear discrete-time state-space model is obtained by defining the state vector $\mathbf{x} = [C_A - C_A^{eq} \quad (T - T^{eq})/100]^T$, the manipulated input $u = (T_c - T_c^{eq})/100$ and by discretizing the ODE with a sampling time $\triangle t = 0.03min$ using the Euler method, which is the following discrete-time model:

$$\begin{cases} x_1(k+1) = x_1(k) + 0.03(\frac{q}{V}(C_{Af} - (x_1(k) + C_A^{eq})) - k_0 \exp(-\frac{E}{R(100x_2(k)+T^{eq})})(x_1(k) + C_A^{eq})) \\ x_2(k+1) = x_2(k) + 0.0003(\frac{q}{V}(T_f - (100x_2(k) + T^{eq})) + \frac{(-\triangle H)}{\rho C_p} k_0 \exp(-\frac{E}{R(100x_2(k)+T^{eq})})(x_1(k) + \\ \qquad C_A^{eq}) + \frac{UA}{V\rho C_p}(100u + T_c^{eq} - (100x_2(k) + T^{eq}))), \end{cases} \tag{7.9}$$

A local stabilizing feedback control law $u = [-0.0690 \quad -4.3387]\mathbf{x}$ is designed in advance according to the linearized model and the LQ method (Magni et al., 2001).

With the designed local stabilizing feedback control law, a terminal control invariant zonotope is obtained via the solver of global optimization for set inversion via zonotope geometry in Section 4.2.7, which is the following polytope:

$$\begin{bmatrix} 0.31623 & -0.94868 \\ -0.31623 & 0.94868 \\ -0.70711 & -0.70711 \\ 0.70711 & 0.70711 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} 0.037947 \\ 0.037947 \\ 0.056569 \\ 0.056569 \\ 0.06 \\ 0.06 \\ 0.04 \\ 0.04 \end{bmatrix}. \tag{7.10}$$

The first-step controllable set to the selected terminal polytope can be computed via Algorithm 4.6 and the obtained first-step controllable set can be approximated innerly by one polytope via Algorithm 6.2. The computed first-step controllable set and its polytopic approximation are shown in Fig. 7.4, where the bound of error tolerance is $\varepsilon = 0.001$.

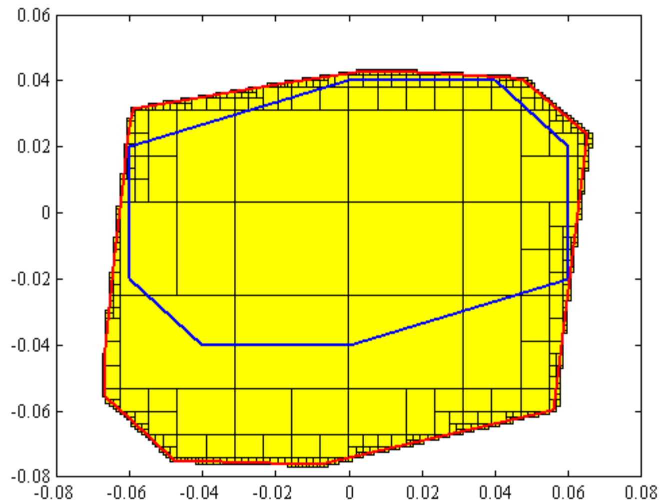


Figure 7.4: The first-step controllable set and its polytopic approximation

The following-step controllable sets can be computed accordingly by renewing the terminal set. The computed polytopic controllable sets of the discretized system are shown in Fig. 7.5. The resulting control process of the dual-mode approach of nonlinear contractive MPC with a contractive sequence of polytopic controllable sets for the discretized system with the initial state $(0.4mol/l, 326\mathbf{K})$ is shown in Fig. 7.6, where the coordinates are transformed to be the original values of the controlled system.

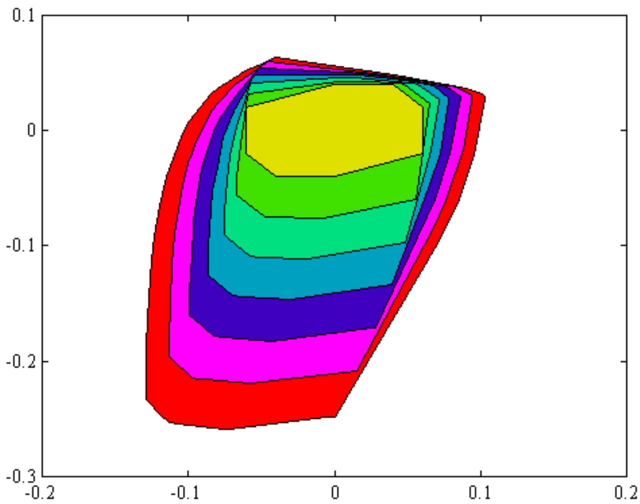


Figure 7.5: The computed polytopic controllable sets of the CSTR

7.3 Summary

Nonlinear contractive MPC with compulsory contractive constraint and nonlinear contractive MPC with a contractive sequence of polytopic controllable sets have been applied to control a MiroSot robot and a CSTR, respectively. Simulation results have demonstrated the effectiveness of contractive MPC for these applications.

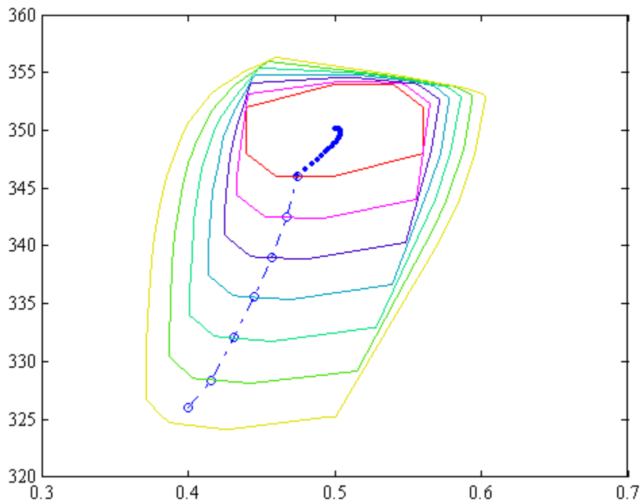


Figure 7.6: The dual-mode control process of the CSTR

Chapter 8

Conclusions and Future Work

8.1 Contributions

Computationally reliable approaches of contractive MPC for discrete-time systems have been studied profoundly in this thesis. Numerical tools for linear systems such as polytope geometry and numerical tools for nonlinear systems such as classical interval analysis, zonotope geometry and modal interval analysis have been introduced and unified to some degree in the same framework of convex sets. Using the developed integrated numerical tools, the terminal control invariant set along with a local stabilizing feedback control law and robust controllable sets needed in the dual-mode approach of contractive MPC can be computed geometrically for general constrained nonlinear uncertain discrete-time systems. Concretely, the contributions or the main work of the thesis in the main chapters are as follows:

Chapter 3:

• A sliding domain along with the related equivalent control deriving from variable structure control is proposed as a terminal control invariant set for the dual-mode approach of linear contractive MPC.

• Linear contractive MPC with compulsory contractive constraint is formulated as a linear programming problem and a contractive sequence of controllable sets with a union of two polytopes as the terminal control invariant set is computed for linear contractive MPC with a contractive sequence of controllable sets.

Chapter 4:

• The bisection of a zonotope is proposed for the first time according to the idea of bisecting an interval and thus the solver of set inversion via interval analysis can be generalized to be the solver of set inversion via zonotope geometry. The generalized solver of set inversion via zonotope geometry is applied to test geometrically whether a given low-complexity polytope along with a local stabilizing feedback control law is (robust) control invariant or no. The solver of global optimization for set inversion via zonotope geometry is also proposed to obtain a control invariant zonotope of the maximal volume for a general constrained nonlinear discrete-time system.

• The solver of set inversion via interval analysis is applied to test the feasibility of nonlinear contractive MPC with compulsory contractive constraint and compute controllable sets of constrained nonlinear discrete-time systems to formulate nonlinear contractive MPC with a contractive sequence of controllable sets.

• The solver of global optimization via interval analysis is applied to obtain one-step control inputs for nonlinear contractive MPC with a contractive sequence of controllable sets.

Chapter 5:

• The theory of modal interval analysis is introduced in a comparative way relative to classical interval analysis, where every concept of modal interval analysis is derived by extending a counterpart concept of classical interval analysis and thus modal interval analysis is treated as an extension of classical interval analysis in modality, inclusion, semantics and rational.

• The solver of 1-dimensional quantified set inversion via modal interval analysis is generalized to multi-dimensional cases and an approximate solver of constrained minimax optimization via interval analysis is also proposed.

• An algorithm based on the generalized solver of multi-dimensional quantified set inversion via modal interval analysis is proposed to compute robust controllable sets of constrained nonlinear uncertain discrete-time systems with a clear semantic interpretation. The proposed solver of constrained minimax optimization is applied to obtain one-step control inputs for nonlinear contractive MPC with a contractive sequence of robust controllable sets.

Chapter 6:

• Two algorithms based on the concept of convex hull are proposed to approximate a union of interval vectors or boxes innerly by one polytope.

• An algorithm based on classical interval analysis and polytope geometry is proposed to compute polytopic robust controllable sets systematically for general constrained nonlinear uncertain discrete-time systems to formulate nonlinear robust contractive MPC with a contractive sequence of polytopic robust controllable sets.

• An algorithm based on interval analysis and polytope geometry is proposed to compute polytopic robust controllable sets for a specific kind of constrained nonlinear uncertain discrete-time systems with affine state part.

• The concept of quasi multi-step robust controllable sets is proposed and an algorithm based on interval analysis, zonotope geometry and polytope geometry is proposed to compute polytopic quasi multi-step robust controllable sets for a specific kind of constrained nonlinear uncertain discrete-time systems with affine control part.

Chapter 7:

• An algorithm based on contractive MPC with compulsory contractive constraint is proposed to control a MiroSot robot with the capacity of automatic obstacle avoidance and inherent path planning.

• Nonlinear contractive MPC with a contractive sequence of polytopic controllable set is applied to control a CSTR.

8.2 Future Work

Numerical tools such as polytope geometry, zonotope geometry and interval analysis are unified to some degree in the same framework of convex sets. Using the developed integrated numerical tools, current research results on explicit MPC and hybrid control of linear and piecewise-affine systems can be extended to nonlinear systems as well. Based on the research progress of the thesis, the following research topics can be explored further:

• For contractive MPC with compulsory contractive constraint, the feasible control horizon is usually equal to the number of the state variables. There might exist some implicit relationships between local controllability and the feasible control horizon of contractive MPC with compulsory contractive constraint. However, such analytical relationships are needed to be explored further to provide the guideline of selecting feasible control horizons for contractive MPC with compulsory contractive constraint easily.

• Interval analysis is extended to zonotope geometry in this thesis. Then current research results and applications of interval analysis can be extended to use zonotope geometry as well to broad the initial search domain from boxes to zonotopes and reduce the wrapping effect of dynamic evolutions.

• The numerical tools of zonotope geometry and interval analysis are powerful for dealing with nonlinear systems. Thus current research results on hybrid control of linear and piecewise-affine systems in the literature can be extended to nonlinear systems as well.

- The relationships between interval vectors or boxes, polytopes and zonotopes have been explored in the thesis, where polytopes are the most general convex sets. Current algorithms developed in the thesis are based on several uncommercial MATLAB toolboxes such as INTerval LABoratory (INTLAB) by Dr. Siegfried M. Rump, Invariant Set Toolbox (IST) by Dr. Eric C. Kerrigan and Multi-Parametric Toolbox (MPT) by M. Kvasnica, P. Grieder and M. Baotic. A user-friendly and independent hybrid polytope interface is needed to be provided in the near future for an easier and wider application of them to MPC.

Appendix A

Hybrid Polytope Interface

The most general convex sets are polytopes and all the convex sets considered in the thesis such as interval vectors and zonotopes can be transformed to the format of polytopes in Multi-Parametric Toolbox (MPT) (Kvasnica et al., 2006) or Invariant Set Toolbox (IST) (Kerrigan, 2000). So MPT and IST play a central role in the unified framework of convex sets, which is to be called Hybrid Polytope Interface (HPI) as the name of all MATLAB routines developed in this thesis. On the other hand, the introduction of nonlinear numerical tools such as interval analysis and zonotopes to MPT and IST extends their capacities from treating not only linear and piecewise-affine systems, but also nonlinear systems. The following sections introduce some primary and conceptual MATLAB routines in HPI to fulfill the transformations from interval vectors and zonotopes to polytopes in 2-dimensional cases, respectively. With the help of these seminal routines in HPT and the existing toolboxes of INTerval LABoratory (INTLAB) (Rump, 2006), MPT and IST, other researchers are anticipated to be able to repeat the simulation results in the thesis or apply them in other research topics.

A.1 Transform from Interval Vectors to Polytopes

A.1.1 Transform An Interval Vector or A Box to A Polytope

The transform of a 2-D interval vector or a 2-D box in INTLAB to a polytope in MPT is realized in the MATLAB file of `box2poly.m`:

```
% To transform a 2-D box to a 2-D polytope
function P=box2poly(X,Y)
% X --- the interval for x
% Y --- the interval for y
% P --- the polytope in MPT: P=polytope(H,K)
P=polytope([1 0;0 1;-1 0;0 -1],[sup(X);sup(Y);-inf(X);-inf(Y)]);
```

Accordingly, a union of 2-D interval vectors or boxes can be transformed straightforwardly to be a union of 2-D polytopes, which is realized in the MATLAB file of `intu2polyu.m`.

```
% To transform a union of 2-D boxes to a union of 2-D polytopes
function polyu=intu2polyu(intu)
% intu --- a union of 2-D boxes stored in an array
% polyu --- a union of 2-D polytopes stored in an array
[m,n]=size(intu);
boxone=intu(1,:);
polyu=box2poly(boxone(1,1),boxone(1,2));
for i=2:1:m
    boxi=intu(i,:);
    in1=boxi(1,1);
    in2=boxi(1,2);
    polyi=box2poly(in1,in2);
    polyu=[polyu polyi];
end
```

A.1.2 Approximate A Union of Boxes Innerly By One Polytope

The inner approximation of a union of 2-D boxes in INTLAB by one 2-D polytope in MPT is realized in the MATLAB file of `polytopica.m` according to Algorithm 6.2:

The cost function used in `polytopica.m` is realized in the MATLAB file of `costfunminp.m`:

A.2 Transform from Zonotopes to Polytopes

A.2.1 Zonotope Definition

The zonotope definition is realized in the MATLAB file of `zonotope.m` as an object under the directory `@zonotope`:

```
% To approximate a union of 2-D boxes innerly by one 2-D polytope
function polyone=polytopica(intu)
% intu --- a union of 2-D boxes stored in an array
% polyone --- the polytope in MPT: P=polytope(H,K)
polyu=intu2polyu(intu);
hu=hull(polyu);
polyone=hu;
vh=extreme(hu);
compliset=hu\polyu;
[m,n]=size(compliset);
for i=1:1:n
    complione=compliset(i);
    [Hc,Kc]=double(complione);
    [Ho,Ko]=double(polyone);
    polych=std2aug([Hc;Ho],[Kc;Ko]);
    if isemptyset(polych)==0
       vc=extreme(complione);
       [mc,nc]=size(vc);
       [mh,nh]=size(vh);
       Ah=[vh(1,:) -1];
       bh=0;
       for k=2:1:mh
           Ah=[Ah;vh(k,:) -1];
           bh=[bh;0];
       end
       for j=1:1:mc
           Ah=[Ah;(-1)*vc(j,:) 0];
           bh=[bh;-1];
       end
       X=fmincon(@costfunminp,[1;1;1],Ah,bh,[],[],[-Inf;-Inf;1],[Inf;Inf;Inf]);
       polyone=polytope([Ho;X(1) X(2)],[Ko;1]);
   end
end
```

```
% The cost function of linear programming for a separating hyperplane
function costv=costfunminp(x)
costv=x(3);
```

```
% The class of a zonotope: Definition
function z=zonotope(p,M)
% p --- the center of the zonotope
% M --- z=p+M*B_(n x m)
z.p=p;
z.M=M;
z=class(z,'zonotope');
```

A.2.2 Zonotope Bisection

The bisection of a zonotope is realized in the MATLAB file of `bisectzono.m` under the directory `@zonotope`:

```
% Bisect a zonotope Z=p+MB_(n x m)
function [Za,Zb]=bisectzono(Z)
% Z --- the zonotope to be bisected
% Za and Zb --- the obtained two zonotopes: Z=Za U Zb
p=Z.p;
M=Z.M;
[rvector,rn]=max(abs(M));
[mvalue,cn]=max(abs(rvector));
pone=p-0.5*M(:,cn);
Mone=M;
Mone(:,cn)=0.5*M(:,cn);
Za=zonotope(pone,Mone);
ptwo=p+0.5*M(:,cn);
Mtwo=M;
Mtwo(:,cn)=0.5*M(:,cn);
Zb=zonotope(ptwo,Mtwo);
```

A.2.3 Transform A Zonotope to A Polytope

A zonotope is a centrally-symmetric polytope and the construction of a 2-D zonotope is realized in the MATLAB file of `zono2poly.m` [1]:

[1] The listed `zono2poly.m` is only for demonstrating the principle of the 2-D zonotope construction via polytope additions while the actual algorithm used for the simulations of the thesis is more complex because of extra treatments of potentially-degenerated polytopes.

```
% Transform a 2-D zonotope to a 2-D polytope
function P=zono2poly(Z)
% Z --- an object of zonotope: Z=zonotope(p,M)
% P --- an object of polytope in MPT: P=polytope(H,K) or P=polytope(V)
p=Z.p;
M=Z.M;
[m,n]=size(M);
if  mod(n,2) == 0
    if n == 2
       vt1=M(:,1)'+M(:,2)';
       vt2=M(:,1)'-M(:,2)';
       vt3=-M(:,1)'+M(:,2)';
       vt4=-M(:,1)'-M(:,2)';
       vt=[vt1;vt2;vt3;vt4];
       Po=polytope(vt);
   else
       vt1=M(:,1)'+M(:,2)';
       vt2=M(:,1)'-M(:,2)';
       vt3=-M(:,1)'+M(:,2)';
       vt4=-M(:,1)'-M(:,2)';
       vt=[vt1;vt2;vt3;vt4];
       Po=polytope(vt);
       for i=3:2:n-1
           vt1=M(:,i)'+M(:,i+1)';
           vt2=M(:,i)'-M(:,i+1)';
           vt3=-M(:,i)'+M(:,i+1)';
           vt4=-M(:,i)'-M(:,i+1)';
           vi=[vt1;vt2;vt3;vt4];
           pi=polytope(vi);
           Po=Po+pi;
       end
    end
else
```

```
vt1=M(:,1)'+M(:,2)'; vt2=M(:,1)'-M(:,2)'; vt3=-M(:,1)'+M(:,2)'; vt4=-M(:,1)'-M(:,2)';
vt=[vt1;vt2;vt3;vt4];
p1=polytope(vt);
if  n == 3
    va=zeros(1,m);
    for j=1:1:4
        vt1a=vt(j,:)+M(:,3)';
        vt2a=vt(j,:)-M(:,3)';
        va=[va;vt1a;vt2a];
    end
    Po=polytope(va);
else
    for k=3:2:n-2
        vt1=M(:,k)'+M(:,k+1)';
        vt2=M(:,k)'-M(:,k+1)';
        vt3=-M(:,k)'+M(:,k+1)';
        vt4=-M(:,k)'-M(:,k+1)';
        vt=[vt1;vt2;vt3;vt4];
        pk=polytope(vt);
        p1=p1+pk;
    end
    vf=extreme(p1);
    [mf,nf]=size(vf);
    vz=zeros(1,m);
    for z=1:1:mf
        vt1b=vf(z,:)+M(:,n)';
        vt2b=vf(z,:)-M(:,n)';
        vz=[vz;vt1b;vt2b];
    end
    Po=polytope(vz);
  end
end
[Ho,Ko]=double(Po); PKo=std2aug(Ho,Ko); PK=translate(p,PKo); [H,K]=aug2std(PK); P=polytope(H,K);
```

Bibliography

T. Alamo, J. M. Bravo, and E. F. Camacho. Guaranteed state estimation by zonotopes. *Proceedings of 42nd CDC*, pages 5831–5836, 2003. 39, 40

M. Bacic, M. Cannon, Y. I. Lee, and B. Kouvaritakis. General interpolation in mpc and its advantages. *IEEE Transactions on Automatic Control*, 48(6):1092–1096, 2003. 13

M. Bacic, M. Cannon, and B. Kouvaritakis. Invariant sets for feedback linearisation based nonlinear predictive control. *IEE Control Theory and Applications*, 152(3): 259–265, 2005. 5

A. Bemporad and M. Morari. Robust model predictive control: a survey. *No. 245 in Lecture Notes in Control and Information Sciences*, pages 207–226, 1999. 9, 14

F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999. 5, 42

H. H. J. Bloemen, M. Cannon, and B. Kouvaritakis. An interpolation strategy for discrete-time bilinear mpc problems. *IEEE Transactions on Automatic Control*, 47 (5):775–778, 2002. 13

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 18, 19

J. M. Bravo. *Control predictivo no lineal robusto basado en técnicas intervalares (Spanish)*. PhD thesis, University of Seville, 2004. 41

J. M. Bravo, C. G. Varet, and E. F. Camacho. Interval model predictive control. *IFAC Workshop 6th Algorithms and Architectures for Real-Time Control*, 2000. 13

J. M. Bravo, D. Limon, T. Alamo, and E. F. Camacho. On the computation of invariant sets for constrained nonlinear systems: an interval arithmetic approach. *Automatica*, 41(9):1583–1589, 2005. 3, 4, 53, 85

E. F. Camacho and C. Bordóns. *Model Predictive Control*. Springer Verlag, second edition, 2005. 8

M. Cannon. Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control*, 28(2):229–237, 2004. 1, 2

M. Cannon, V. Deshmukh, and B. Kouvaritakis. Nonlinear model predictive control with polytopic invariant sets. *Automatica*, 39(8):1487–1494, 2003. 4, 5, 40, 43, 51, 54, 76, 77, 83, 89

H. Chen and F. Allgower. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998. 5, 12

W. Chen, D. Ballance, and J. O'Reilly. Optimization of attraction domains of nonlinear mpc via lmi methods. *Proceedings of the 2001 American Control Conference*, pages 3067–3072, 2001. 3

S. L. de Oliveira and M. Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 45(6):1053–1071, 2000. 1, 3, 12, 15, 21, 48

A. C. C. F. Fernando. A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42(2):127–143, 2001. 2, 3

R. Findeisen and F. Allgower. An introduction to nonlinear model predictive control. *21st Benelux Meeting on Systems and Control*, 2002. 11

K. Fukuda. From the zonotope construction to the minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261–1272, 2004. 35

Winston García-Gabín and E. F. Camacho. Sliding mode model based predictive control for non minimum phase systems. *Proceedings of the 2003 European Control Conference*, 2003. 3

Winston García-Gabín, Darine Zambrano, and E. F. Camacho. Multivariable model predictive control of process with instable transmision zeros. *Proceedings of the 2002 American Control Conference*, pages 4189–4190, 2002. 2

Winston García-Gabín, Darine Zambrano, and E. F. Camacho. Sliding mode predictive control for chemical process with time delay. *16th IFAC World Congress*, 2005. 3

E. Gardenes, M. A. Sainz, L. Jorba, R. Calm, R. Estela, H. Mielgo, and A. Trepat. Modal intervals. *Reliable Computing*, 7(2):77–111, 2001. 60, 61, 63, 64

G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel. Examples when nonlinear model predictive control is nonrobust. *Automatica*, 40(10):1729–1738, 2004. 50

L. J. Guibas, A. Nguyen, and L. Zhang. Zonotopes as bounding volumes. *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete Algorithms*, pages 803–812, 2003. 41

E. Hansen. *Global Optimization Using Interval Analysis*. Marcel Dekker, New York, 1992. 13

P. Herrero, M. A. Sainz, J. Vehi, and L. Jaulin. Quantified set inversion algorithm with applications to control. *Reliable Computing*, 11(5):369–382, 2005. 66

L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, London, 2001. 15, 29, 30, 31, 32, 39, 41, 59

E. Kaucher. Interval analysis in the extended interval space ir. *Computing Supplementum 2*, pages 33–49, 1980. 65

S. S. Keerthi and E. G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving horizon approximations. *Journal of Optimization Theory and Applications*, 57(2):265–293, 1988. 12

E. C. Kerrigan. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*. PhD thesis, University of Cambridge, 2000. 3, 16, 17, 24, 25, 88, 101, 126

H. K. Khalil. *Nonlinear Systems*. Prentice Hall, third edition, 2002. 97, 108

M. V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996. 5, 14

B. Kouvaritakis and M. Cannon. *Nonlinear Predictive Control - Theory and Practice*. Institute of Electrical Engineers, London, UK, 2001. 11

W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61(1):47–67, 1998. 38

M. Kvasnica, P. Grieder, and M. Baotić. Multi-parametric toolbox (mpt), 2006. URL http://control.ee.ethz.ch/~mpt/. 19, 35, 41, 85, 126

J. H. Lee and Z. Yu. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33(5):763–781, 1997. 14

D. Limon, J. M. Bravo, T. Alamo, and E. F. Camacho. Robust mpc of constrained discrete-time nonlinear systems based on uncertain evolution sets: application to a cstr model. *Proceedings of IEEE International Symposium on Computer Aided Control System Design*, pages 657–662, 2002. 15

D. Limon, T. Alamo, and E. F. Camacho. Robust mpc control based on a contractive sequence of sets. *Proceedings of 42nd CDC*, pages 3706–3711, 2003. 3, 4, 17, 46, 116

D. Limon, T. Alamo, and E. F. Camacho. Enlarging the domain of attraction of mpc controllers. *Automatica*, 41(4):629–635, 2005a. 3, 4

Y. Lu and Y. Arkun. Quasi-min-max mpc algorithms for lpv systems. *Automatica*, 36 (4):527–540, 2000. 14

L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37(9):1351–1362, 2001. 5, 15, 43, 46, 116, 117

D. Q. Mayne. Model predictive control: the challenge of uncertainty. *IEE Two-Day Workshop on Model Predictive Control: Techniques and Applications*, pages 6/1–6/5, 1999. 8

D. Q. Mayne and W. R. Schroeder. Robust time-optimal control of constrained linear systems. *Automatica*, 33(12):2103–2118, 1997. 16

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. B. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36(6):789–814, 2000. 1, 10, 42

C. H. Messom. Robot soccer — sensing, planning, strategy and control, a distributed real time intelligent system approach. *The Third International Symposium on Artificial Life and Robotics*, pages 422–426, 1998. 112

C. H. Messom, G. Sen Gupta, S. Demidenko, and Lim Yuen Siong. Improving predictive control of a mobile robot: application of image processing and kalman filtering. *IEEE Instrumentation and Measurement Technology Conference*, pages 1492–1496, 2003. 113

H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993. 5, 12, 15

R. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966. 29

M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23(4):667–682, 1999. 8, 10, 14

C. J. Ong and E. G. Gilbert. Outer approximations of the minimal disturbance invariant set. *Proceedings of 44th IEEE Conference on Decision and Control and the European Control Conference 2005*, pages 1678–1682, 2005. 35

G. A. S. Pereira, M. F. M. Campos, and L. A. Aguirre. Data based dynamical model of vision observed small robots. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 3312–3317, 2000a. 112

G. A. S. Pereira, M. F. M. Campos, and L. A. Aguirre. Improved control of visually observed robotic agents based on autoregressive model prediction. *Proceedings of IEEE/RJS International Conference on Intelligent Robots and Systems*, pages 608–614, 2000b. 113

S. V. Rakovic, E. C. Kerrigan, and D. Q. Mayne. Reachability computations for constrained discrete-time systems with state- and input-dependent disturbances. *Proceedings of 42nd CDC*, pages 3905–3910, 2003. 3, 87

S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005. 35

C. V. Rao. *Moving horizon strategies for the constrained monitoring and control of nonlinear discrete-time systems*. PhD thesis, University of Wisconsin-Madison, 2000. 1, 2, 9

J. B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Magazine*, 20(3):38–52, 2000. 10

Siegfried M. Rump. Interval laboratory (intlab), 2006. URL `http://www.ti3.tu-harburg.de/~rump/intlab/`. 126

J. M. Martin Sanchez and J. Rodellar. *Adaptive Predictive Control: From the Concepts to Plant Optimization*. Prentice Hall, 1996. 2

P. O. M. Scokaert and D. Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998. 14

P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability. *IEEE Transactions on Automatic Control*, 44 (3):648–654, 1999. 13

M. Soroush and C. Kravaris. Short horizon nonlinear model predictive control. *Proceedings of the 4th IEEE Conference on Control Applications*, pages 943–948, 1995. 3

J. Vehi, J. Rodellar, M. A. Sainz, and J. Armengol. Analysis of the robustness of predictive controllers via modal intervals. *Reliable Computing*, 6(3):281–301, 2000. 15

J. Wan, C. G. Quintero, N. Luo, and J. Vehi. Predictive motion control of a mirosot mobile robot. *Proceedings of World Automation Congress 2004*, 15:325–330, 2004a. 116

J. Wan, J. Vehi, and N. Luo. Nonlinear model predictive control via interval analysis. *Proceedings of NOLCOS 2004*, pages 949–952, 2004b. 49

J. Wan, J. Vehi, and N. Luo. Local receding horizon control with contractive constraints. *Proceedings of World Automation Congress 2004*, 16:243–247, 2004c. 115

Y. Wang. *Robust model predictive control*. PhD thesis, University of Wisconsin-Madison, 2002. 13

K. Watanabe. Intelligent control for robotic and mechatronic systems - a review. *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pages 322–327, 1996. 112

F. Wu. Lmi-based robust model predictive control and its application to an industrial cstr problem. *Journal of Process Control*, 11(6):649–659, 2001. 14

T. H. Yang and E. Polak. Moving horizon control of linear systems with input saturation and plant uncertainty. *International Journal of Control*, 58(3):613–638, 1993a. 3, 12, 15

T. H. Yang and E. Polak. Moving horizon control of linear systems with input saturation and plant uncertainty. *International Journal of Control*, 58(3):639–663, 1993b. 3, 12, 15

J. Zhang, S. Cheng, and B. Wang. Model predictive control for time-delay systems with terminal sliding mode constraints. *Proceedings of the 5th World Congress on Intelligent Control and Automation*, pages 705–708, 2004. 3

Z. Q. Zheng. *Robust control of systems subject to constraints*. PhD thesis, California Institute of Technology, 1995. 14

J. Zhou, Z. Liu, and R. Pei. Sliding mode model predictive control with terminal constraints. *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, pages 2791–2795, 2000. 3

J. Zhou, Z. Liu, and R. Pei. A new nonlinear model predictive control scheme for discrete-time system based on sliding mode control. *Proceedings of the 2001 American Control Conference*, pages 3079–3084, 2001. 3