Set-theoretic methods in control. Applications to fault tolerant control and motion planning

Habilitation thesis

Florin Stoican

16 mai 2018
Contents

Summary

I. Professional and academic achievements

1. Academic career
   1.1. Information about titles and diplomas
   1.2. Workplace and professional experience
   1.3. Publications, research interests and visibility

2. Capacity of independent research and project management
   2.1. My collaborations
   2.2. Principal investigator
   2.3. Other activities

3. Scientific achievements
   3.1. Main results
   3.2. Full list of publications

II. Scientific results

4. Explicit descriptions of robust invariant sets
   4.1. Preliminaries
   4.2. Prerequisites
   4.3. Explicit representation of the mRPI set
   4.4. RPI constructions using $\Omega_\infty(A, \Delta)$
   4.5. Conclusions
5. **Exact and over-approximated guarantees for corner cutting avoidance**  
   - Preliminaries .......................................................... 37
   - Prerequisites ............................................................ 39
   - Shadow regions descriptions ........................................ 40
     - Exact description of the under-shadow region ................. 41
     - Over-approximation of the shadow region ...................... 43
   - Mixed integer representations ...................................... 46
     - Mixed integer representations for the exact shadow and visible regions .... 47
     - Mixed integer representations for the over-approximated shadow regions .... 49
     - Comparison with the state of the art ............................ 50
   - Study case: an MPC problem with corner cutting avoidance ............... 55
     - Justifications for corner cutting avoidance strategies .......... 56
     - Comparison of various corner cutting avoidance strategies .......... 57
     - Performance and robustness issues .............................. 59
   - Conclusions ............................................................... 60

6. **Active Fault Detection and Isolation in a Zonotopic Framework**  
   - Preliminaries .......................................................... 61
   - Prerequisites ............................................................ 62
   - Problem description .................................................... 63
     - State estimation ...................................................... 64
     - Tracking error ......................................................... 65
     - Set characterizations ............................................... 66
   - Fault tolerant control scheme ........................................ 67
     - Fault tolerant control scheme implementation .................. 68
     - Active fault detection and isolation .............................. 68
   - Illustrative example ................................................... 70
   - Conclusions ............................................................... 71

7. **Hierarchical Control with Guaranteed Fault Diagnosability**  
   - Preliminaries .......................................................... 72
   - Prerequisites ............................................................ 73
     - Multi-sensor plant dynamics ....................................... 74
     - Fault scenario and residual design .............................. 75
     - Set characterizations ............................................... 76
   - Hierarchical reference governor with active FDI .................. 77
     - Hierarchical active FDI validation ............................... 77
     - Guaranteed fault diagnosability for intra-sample behavior ....... 78
   - Illustrative example ................................................... 80
   - Conclusions ............................................................... 82

8. **Dictionary learning strategies for sensor placement and leakage isolation in water networks**  
   - Preliminaries .......................................................... 83
# Chapter 0

8.2. Problem description ................................. 85
8.2.1. Network dynamics ................................. 85
8.2.2. Fault scenarios ................................. 85
8.3. Main idea ........................................ 86
8.3.1. Dictionary learning elements .................. 87
8.3.2. Sensor placement algorithm ................... 88
8.3.3. Fault detection and isolation implementation ... 90
8.4. Additional results ................................. 92
8.4.1. Network partitioning .......................... 92
8.4.2. Staggered sensor deployment .................. 93
8.5. Conclusion ....................................... 93

9. Obstacle avoidance via B-spline parameterizations of flat trajectories 94
9.1. Preliminaries ..................................... 94
9.2. Prerequisites ..................................... 95
9.2.1. Flat trajectories .............................. 95
9.2.2. B-splines .................................... 96
9.3. Flat trajectory generation ......................... 97
9.4. Main idea ....................................... 97
9.4.1. The exact case ................................ 98
9.4.2. The sub-optimal case ......................... 99
9.5. Illustrative example for an UAV system .......... 101
9.6. Conclusions ..................................... 104

10. A combinatorial view of the explicit representation of the MPC problem 105
10.1. Preliminaries ................................... 105
10.2. Prerequisites ................................... 106
10.3. Upper bound for the collection of candidate sets of active constraints .......... 108
10.3.1. Mixed integer representation of the KKT problem .................................. 111
10.4. Partial recursive implementation of the explicit region .......................... 112
10.5. Illustrative example ............................. 115
10.6. Conclusions ..................................... 116

11. On the design of exact penalty functions for MPC using mixed integer programming 117
11.1. Preliminaries ................................... 117
11.2. Bi-level programming ........................... 118
11.3. Problem preliminaries ........................... 120
11.4. Soft constraints in MPC ........................ 121
11.4.1. Equivalence between “hard” and “soft” constraints .......................... 121
11.4.2. Polyhedral norm formulation ................ 123
11.5. Reformulation of the KKT conditions ............ 124
11.5.1. MILP formulation ........................... 124
11.5.2. Explicit imposition of the LICQ condition .................. 125
III. Open issues and future directions 136

12. Open issues in the state of the art 137

12.1. Set theoretic issues for fault tolerant control 137
   12.1.1. Numerical and theoretical difficulties in set constructions related to control problems 137
   12.1.2. Fault tolerant control design for complex systems 138

12.2. Issues for motion planning strategies 138
   12.2.1. Numerical issues of trajectory generation for nonlinear dynamics 139
   12.2.2. Discretization issues for trajectory generation and subsequent tracking 139
   12.2.3. Reliable trajectory generation under model variation and disturbances 140

13. Future directions 141

13.1. Improvements in set descriptions. Numerical issues 141
   13.1.1. Robust estimation and invariance constructions for complex systems 141
   13.1.2. Extensions to chance constrained dynamics 142
   13.1.3. Elements of originality and innovation. Impact 142

13.2. Fault tolerant control with set-theoretic methods 142
   13.2.1. Fault detection and isolation analysis 143
   13.2.2. Control reconfiguration and fault accommodation 143
   13.2.3. Elements of originality and innovation. Impact 144

13.3. Motion planning strategies 144
   13.3.1. Offline trajectory generation via flatness with B-spline parametrizations 145
   13.3.2. Improvements in reliability and stability guarantees under model variation 145
   13.3.3. Elements of originality and innovation 146

13.4. Validation of the results. Experimental setups 147
   13.4.1. Fault tolerant control implementations in complex systems 147
   13.4.2. Motion planning strategies in a multi-agent setting 148

Bibliography 149
List of Tables

4.1. Volume variation for the $R^a_k(A, \Delta)$ sets. .................................................. 36

5.1. Mean and maximum computation times considered for a 30 steps simulation horizon and for 245 feasible initial conditions. .................................................. 59

8.1. Successful FDI percentage for sensor placement cases 1 and 2. ..................... 91
8.2. Successful FDI percentage for sensor placement cases 1 and 2 – with network partitioning. 92

9.1. Evolution of trajectories characteristics for degree $d = 4$ in the non-linear and mixed-integer formulations. ................................................................. 104

10.1. Time necessary to solve (10.16) and (10.17). ................................................. 116
List of Figures

4.1. Illustrations of the mRPI set. ................................................................. 30
4.2. Illustrations of the mRPI set for state matrix with negative eigenvalues. ............ 31
4.3. RPI approximation of the mRPI set. .......................................................... 35
4.4. RPI approximation of the mRPI set using $\Omega_I = S(f \cup L_e)$. ....................... 36

5.1. Illustration of a collection of obstacles and their associated hyperplane arrangement. 40
5.2. Illustration of exact and over-approximated under-shadow regions. ....................... 45
5.3. Illustration of the exact shadow region in a multi-obstacle environment. ................ 52
5.4. Illustration of the over-approximated shadow regions in a multi-obstacle environment. 54
5.5. Inter-sample behavior for double integrator dynamics. .................................... 57
5.6. Illustration of trajectories with and without corner cutting avoidance. ................... 58
5.7. Illustration of mean and maximum computation time for the case ($N_p = 10, T = 1$). 59
5.8. Illustration of path lengths for each feasible initial point. .................................. 60

6.1. Timeline for a fault scenario with set inclusion illustration. .................................. 67
6.2. Illustration of bounding sets for state estimation and tracking errors (for $\tau = 5$). .... 71

7.1. Hierarchical fault tolerant plant control scheme with high-level optimal reference governor. Lined boxes represent observers of the individual sensors. $SW$ is a switch choosing one of the healthy state estimations from the index set $\mathcal{J}_H$ for the state feedback. .... 73
7.2. RPI sets associated with the plant and observer dynamics. .................................. 81
7.3. Illustration of reference governor implementation with active FDI validation. The blue areas depict regions in the state space where FDI is not decidable. .......................... 81
7.4. Illustration of $\Delta$ and $\Delta$-sampled state reference. ..................................... 82

8.1. Hanoi water network. ................................................................................. 84
8.2. Fault detection and isolation via dictionary classification with sensor placement in the Hanoi water network. ................................................................. 86
8.3. Dictionary classification for sensor placement (the case $s = 4$). ......................... 89
8.4. Node placement selection for different number of sensors. ................................. 89
8.5. Active atoms for residual vectors in the Hanoi network (with detail for the fault 17). . 90
8.6. Active atoms for residual vector corresponding to fault $f_{17} = 17$. .................... 91

9.1. Flat trajectories in a multi-obstacle environment. .............................................. 103
10.1. Illustration of feasible selections. .............................................................. 111
10.2. Explicit solution for $N = 12$ and the unions of regions which have the same initial input (indicated by a solid red line). ..................................................... 115
11.1. Feasible region of problem (11.5) and the point where the $\ell_1$-norm of the Lagrangian multipliers is maximized. ............................................................... 132
11.2. Time necessary to solve problem (11.15) – in solid blue, vs. for solving problem (11.23)
– in dashed red. ..................................................................................................... 132
11.3. Depiction of trajectories generated with/out soft constraints (dashed red and solid blue respectively). .............................................................. 133
11.4. Depiction of inputs/outputs generated with soft constraints for a $\ell_1$ and $\ell_\infty$-norm penalty functions when starting from an infeasible point. .................. 134
Summary

Acest manuscris prezintă rezultatele profesionale, academice și științifice obținute în ultimii 7 ani de carieră (ulteriori obținerii titlului de doctor), precum și direcții de interes pe care doresc să le urmăresc pe termen mediu (pe baza problemelor deschise pe care le-am identificat în cadrul domeniului meu de interes).

În particular, și așa cum va deveni evident pe parcurs, sunt interesat de aplicarea metodelor cu mulțimi în cadrul unor probleme de reglare de actualitate (reglare tolerantă la defecte și planificarea mișcării pentru sisteme autonome). Consider că aceste metode sunt încă insuficient folosite datorită unei varietăți de factori (dificultatea teoretică, probleme numerice, etc.). Aplicarea lor pe scară largă va permite o analiză riguroasă a performanței, stabilizării și functionalității unui sistem (e.g., funcționare nominală sau defectuoasă).

Prima parte a manuscrisului face o scurtă trecere în revistă a carierei mele profesionale și academice până în acest moment (post-teză). Principalele etape parcurs au fost urmărirea unui stagiu postdoctoral (NTNU, Norvegia) și angajarea în cadrul departamentului de Automatică și Ingineria Sistemelor al UPB, România unde ocup în prezent poziția de conferențiar. De-a lungul acestei evoluții, am fost implicat în diverse activități de cercetare și didactice ce mi-au permis astfel să îmi dezvolt capacitatea de lucru independent.

A doua parte a manuscrisului enumeră principalele rezultate științifice obținute după finalizarea tezei. În particular, consider următoarele direcții de lucru relevante:

i) folosirea extensivă a mulțimilor invariante (pornind de obicei de la perturbații zonotopice) pentru a caracteriza performanța și funcționarea unui sistem dinamic; aceste abordări mi-au fost de folos atât pentru implementarea a unor strategii de reglare tolerantă la defecte active (detectie și izolație în bucla închisă, parametrizarea unui generator de referință, etc.) cât și pentru a analiza structura problemei rezultate (spre exemplu, pentru a analiza complexitatea reprezentării MPC explicite);

ii) planificare de traiectorii pentru sisteme dinamice neliniare (în particular sisteme de tip UAV aripă fixă și/sau multi-rotor); folosind noțiuni de platitudine, parametrizări cu funcții B-spline am reușit să crez traiectorii de referință ce respectă dinamica unui sistem, constrângeri asupra intrării și/sau stărilor);

iii) relativ recent am devenit interesant în diferența dintre comportamentul în discret și cel în continuu al unui sistem dinamic; în acest sens am studiat problema ocolirii de obstacole în sensul de a propune un număr finit de constrângeri (uzual neliniare) ce garantează ocolirea unui obstacol în timp continuu.
A treia parte a manuscrisului sintetizează problemele pe care le-am identificat de-a lungul ultimilor ani precum și abordările pe care doresc să le aplic. O scurtă enumerare a acestora acoperă:

i) construcții explicite pentru regiuni invariante (robust pozitive sau robust controlabile) pentru a caracteriza funcționalitatea unui sistem dinamic; pentru a îmbunătăți rezultatele existente îmi propun să exploatez forme particulare (perturbații zonotopice) precum și structura intrisecă a problemei;

ii) realizarea unei scheme de reglare tolerantă la defecte realiste: considerarea comportamentului tranzitioriu; comportament în buclă închisă pentru detecție și izolație; problema observabilității în sisteme de mari dimensiuni (e.g., plasarea de senzori în rețele de apă);

iii) analiza traiectoriilor sistemelor neliniare prin prisma platidunii și a diverse parametrizări (e.g., NURBS); garanții îmbunătățite de funcționare în prezență defectelor, a erorilor de urmărire, de incertitudine de model, etc.;

iv) pentru validarea punctelor anterioare voi testa rezultatele obținute pe sisteme complexe (de mari dimensiuni, eterogene din punct de vedere al componentei și/sau localizării geografice, etc.); în particular sunt interesat de problema detectiei în rețele de apă; de controlul sistemelor multi-agent de tip dronă precum și de analiza sistemelor de transmisie electrică.
Part I.

Professional and academic achievements
1. Academic career

My academic career spans the last 10 years (2008 - 2018) and the main highlights are enumerated below: diplomas, professional experience, a brief recapitulation of my publications and visibility at the national and international level.

1.1. Information about titles and diplomas

Details about my BsC and Bachelor diplomas are enumerated below:

2008-2011 PhD Thesis in Automatic Control, Department of Automatic Control, École Supérieure d’Electricité (SUPÉLEC), Gif sur Yvette, France:

*title:* “Fault tolerant control based on set-theoretic methods”;


2003-2008 Bachelor Engineering Diploma , Faculty of Automatic Control and Computer Science, University Politehnica of Bucharest (UPB), România:

*specialization:* Automatics and Industrial Informatics;

*bachelor diploma:* “Multisensor control systems” at SUPÉLEC, Gif sur Yvette, France.

1.2. Workplace and professional experience

My workplace experience consists of:

2013- Associate Professor (Assistant Professor, 2012-2016), Departament of Automatic Control and Systems Engineering (AIS), University Politehnica of Bucharest (UPB), România:

- lecturer for the Bachelor classes “Tehnici de Calcul în Automatică si Informatică” and “Modelare și Simulare” and of the Master class “Autonomous Agents”;
- coordinator for Bachelor (17), Master (3) diplomas and Erasmus placements (2).

2014-2015 Post-doctoral fellow at UPB, România (Innoresearch POSDRU/159/1.5/S/132395);

2011-2012 Post-doctoral fellow at the Norwegian University of Science and Technology (NTNU), Norway; through a grant ERCIM Alain Bensoussan co-financed by FP7 Marie-Curie ABCDE.

My professional experience covers participation as team member or as principal investigator in multiple national and international research projects:
2017 Responsible from UPB’s side to a PN-III “Innovation Check” project: “Implementation and development of algorithms for the dynamic motion planning of robotic systems (DEVROS)”; PN-III-P2-2.1-CI-2017-0403; July 2017 – December 2017; [http://devros.pub.ro/project](http://devros.pub.ro/project);

2015-2017 Principal investigator to a PN-II “Young Team” project: “Set-theoretic approaches for fault tolerant control of complex systems (SETS2FTC)”; PN-II-RU-TE-2014-4-2713; octombrie 2015 - septembrie 2017; [http://sets2ftc.pub.ro/project](http://sets2ftc.pub.ro/project);

2016 Responsible of a “Young Researcher from Diaspora” mobility project, MCT-2016-0037 (Assoc. Professor Ionela Prodan from LCIS, Grenoble INP, France invited for a two week stay);


2013-2018 Team member in multiple national and international projects, financed as follows:

- PN-II and PN-III: “Robust control in nonstandard cases” (2016, PN-II-ID-PCE-2011-3-0235), “Multi-drones system for evaluation of flood effects” (2016-2017, PN-III-BG-2016-0315);

1.3. Publications, research interests and visibility

My publications are:

- book “Set-theoretic fault detection in multi-sensor control” in FOCUS series, edited by Hermes Penton-ISTE Ltd with Wiley (2013);
- 2 book chapters (in Springer collections, 2013 and 2015);
- 13 ISI indexed journal articles (with a cummulated impact factor\(^1\) of 31.93);
- 40 (28 ISI indexed) conference proceedings in international top Control conferences (IEEE Control and Decision Conference, IEEE European Control Conference, IEEE American Control Conference, IFAC World Congress Conference, other IEEE and IFAC affiliated conferences);
- total number of citations (without auto-citations) is: Scopus (230), Web of Science (200), thus, the values of the h-index are 9 and, respectively, 8.

\(^{1}\)In here and the rest of the manuscript, the impact factor and the Q-classification are taken from the JCR 2016 report, the latest available at the manuscript redactation date.
My research interests cover:

- fault tolerant control (via set theoretic methods, with application in multisensor systems);
- positive and controlled invariance (with applications in RPI/mRPI computations; for switched, with dwell-time or with delays dynamics; zonotopic sets);
- mixed integer programming (efficient descriptions for non-convex and non-compact regions through hyperplane arrangements; description of complementarity conditions);
- constrained optimization problems (exact penalty functions, compact explicit representations for the MPC problem, geometrical interpretations).

I am visible through a series of collaborations and visits:

- invited speaker at the 36th International Summer School in Automation (Grenoble, France, 2015);
- invited Professor at the LCIS Laboratory – INP Grenoble (2015, 2016), CentraleSupélec (2015, 2016), TU Chemnitz (2016) for short-length visits (1-4 weeks);
- invited speaker at the SADCO OMPC workshop (young researcher grant, 2013);
- teaching activities at NTNU, Norway (part of a SEE mobility grant, 2014) and at TU Ilmenau, Germania; UPC, Spain (part of Erasmus+ mobility bourses, 2015, 2016);
- associated editor for the Mediterranean Control Conference (2016, 2017, 2018); “Local Arrangements Chair” and IPC for the “4th International Workshop on Advanced Control and Diagnosis” http://www.acd2017.acs.pub.ro/;
- chair and co-chair at the “25th Mediterranean Conference on Control and Automation (MED 2017)” and “21st International Conference on System Theory, Control and Computing (ICSTCC 2017)”;
2. Capacity of independent research and project management

In support of my capacity of independent research as well as management capabilities, I enumerate my collaborations; work done in the projects in which I was principal investigator and various other coordination activities.

2.1. My collaborations

Note that hereafter I mention groups and locations with whom I started collaborating after the end of my PhD (even if in some cases the contacts in these places were known to me from before the end of my thesis period or even if some the resulting papers involved my thesis adviser – as is the case for UPC, Spain).

2.1.1. LCIS of Grenoble INP, France

My collaboration with Ionela Prodan is the longest and most enduring (post-PhD): it started from the period of my thesis and continued after her relocation to the Laboratoire de Conception et d’Intégration des Systèmes of Grenoble INP, France. I have in last few years (starting with 2013) collaborated with students from her group (Thinh Nguyen, Hung Pham) and department (Laurent Lefèvre) on topics ranging from: mixed-integer programming in control with hyperplane arrangement descriptions [C10], [B4]; efficient microgrid control [J4]; B-spline parameterizations for flat descriptions of trajectories [C13], [C1], [C9], [C28]; quadcopter constrained control [C4]; potential field formulations [C7].

I visited the lab on multiple occasions, earliest in ’15 and most recently in September’16 and ’17.

2.1.2. NTNU and SINTEF Digital, Norway

I spent one year (2012) at NTNU as a postdoc fellow under the coordination of Prof. Morten Hovd. The fellowship materialized in a number of papers ranging from mixed-integer formulations for the design of exact penalty function [J7]; explicit mRPI approximations [B1], [C15]; constrained optimization control [C17], [C33]; and MPC formulations [C35], [C36].

Nearer to the present, I kept and developed close connections with people from NTNU and with the closely linked SINTEF Digital lab. These connections manifested through a number of visits (first as part of a SEE mobility grant, 2014, initially and most recently through research visits, 2017). The main topic of interest has been “unmanned vehicles applications”. Specific results cover a mixed-integer formulation to characterize the potential field induced by a hyperplane arrangement in [C7] and an improvement of the “corner cutting” problem (with application in both trajectory generation and set-based fault detection and isolation conditions) [C29].
2.1.3. ACSD Lab of TU Chemnitz, Germany

In the *Automatic Control and System Dynamics* Lab of TU Chemnitz, Germany I have collaborated with Prof. Stefan Streif and members of his group on topics related to active FDI formulations (initial contact has been through an Erasmus+ placement in 2015 and an invited speaker grant received from Prof. Streif in 2016; the most recent visit has been in 2017).

With this group, the topics of interest center of fault tolerant control. The main result is a combination of the geometric constraint imposed by corner cutting guarantees in a hierarchical control scheme where the FDI validation (and subsequent control reconfiguration) is done at the bottom level (with fast sampling time) and the conditions for FDI are ensured at the top level (with slower sampling time). These results have recently been accepted for publication in a top conference in the field.

2.1.4. ODE Dept. of TU Ilmenau, Germany

In the *Ordinary Differential Equations* Dept. of TU Ilmenau, Germany, I am collaborating with Jr. Prof. Karl Worthmann on the topics of FDI for generic piecewise affine systems with non-instantaneous detection and isolation time. Using a combination of zonotopic sets and finite-window observers, faults have been detected and isolated through an active procedure.

The collaboration has been through a student (Daniel-Mihail Ioan) which has visited the department in September’16 and continued his studies there as an Erasmus+ master student for the academic year of 2016-2017. His Master thesis has directly led to the publication of the conference paper [C27].

2.1.5. Automatic Control Dept. of UPC, Spain

In the *Automatic Control Dept.* of UPC I am collaborating with Prof. Vicenc Puig and Carlos-Ocampo Martinez on topics related to water networks and zonotopic descriptions of residual sets. The former has been presented as proceedings in [C3] and the later was materialized through a series of papers where the main author was a PhD student of the aforementioned V. Puig and C. Martinez ([C11], [C12], [C14], [C16], [C32], [C34], [J5], [J9]).

The collaboration has spanned two different periods: 2012-2015 for the series of papers involving the PhD student and 2017 for the ongoing work regarding water networks.

2.1.6. Computer Science Dept. of the University of Bucharest, Romania

In the *Computer Science Dept.* of University of Bucharest, Romania, I am collaborating with Paul Irofti on the topic of sensor placement and subsequent fault detection and isolation in water networks. A first results was published [C3].

2.2. Principal investigator

2.2.1. SETS2FTC project

I was the principal investigator of a national PN-II “Young Team” research project (2015-2017) called “Set-theoretic approaches for fault tolerant control of complex systems (SETS2FTC)”. As part of this project, I coordinated a team of 2 PhDs, 1 postdoc, 2 master students for the study and application of set-theoretic methods in the fault tolerant control of complex systems.
Results have been in: i) efficient invariant set descriptions; ii) sensor placement in large networks in view of accurate fault detection and iii) mixed integer issues for robust control have been presented in 6 conference papers (ISI indexation pending) and 2 journal papers (ISI indexed).

The project entailed a variety of actions: primarily scientific (research activities, dissemination and short visits in other laboratories: LCIS of INP Grenoble; UPC, TU Ilmenau and TU Chemnitz) but also administrative and financial.

2.2.2. DEVROS project

I was the partner responsible from my University’s side in a national PN-III “Innovation Check” project (2017) called “Implementation and development of algorithms for the dynamic motion planning of robotic systems (DEVROS)”. As part of this project, I coordinated a team of 2 post-docs and 1 master student for the application of set-based methods in the generation of trajectories for a mobile robot (TurtleBot3).

The results have been primarily in the implementation of trajectory generation procedures and have been done in collaboration with the beneficiary SME (the call’s aim was to solve a specific, applied research requirement of the SME using academic expertise from the University).

The project entailed applied research, administrative and financial activities.

2.3. Other activities

I was part (in some of them as the main organizer) in multiple project proposals. E.g., I was involved in national PN-II and PN-III calls proposals: PD (postdoc research project) 2012, BG (Bridge Grants) 2017, PED (Project Exploration-Demonstrative) 2017, TE (Young Team) 2017; and international call proposals (COST 2015, 2018 proposals, mobilities Romania-France in 2014 and 2016).

In the last 4 years I coordinated a number of Bachelor (17) and Master (2) licenses. These have covered subjects in my areas of interest: mixed-integer descriptions, model predictive implementations; motion planning for unmanned vehicles (quadcopters in particular).

Two of my students have benefited of Erasmus+ placements: one for his 3rd year (at TU Chemnitz, Germany) and the other during his 1st year of Master (at TU Ilmenau, Germany). These visits have materialized into research articles.
3. Scientific achievements

3.1. Main results

I divide my results and research activities into four main groups as follows:

I) Fault tolerant control with set-theoretic methods. This topic was the main subject of my PhD thesis and dealt with the analysis and design of fault tolerant control (FTC) multisensor schemes with set-theoretic methods [B3]. The main element was the use of invariant / contractive sets to describe the residual signals characterizing the functioning of the system (i.e., “healthy” or “under fault”). A fault detection and isolation (FDI) and the dual recovery mechanisms, based on set-separation properties, have been implemented [J6]. The main interest was in the explicit inclusion of FDI objectives in the control reconfiguration mechanism. It was shown that the FTC design can be completed through reference governor synthesis [J12]; model predictive control (MPC) and static feedback schemes conditioned by FDI set-separation [J10]. These contributions:

- relate to my overall interest in set theoretic methods in control as they provide the knowledge necessary to model faults and to analyze (fault detection and isolation) and accommodate them (control reconfiguration strategies);
- have materialized into 9 journals (4 as main author), 1 book (main author) and over 15 conference papers (more than 10 as main author) with fault tolerant control issues as the principal topic (fault detection, control reconfiguration, benchmark applications, etc.);
- have been my first foray into the academic world and have allowed me to collaborate with groups working in similar areas (France, Australia, Greece); recent results (after the thesis completion) have been with researchers from UPC (Spain), INP Grenoble (France) on: i) nonlinear dynamics (communication delays, switched dynamics with dwell-time [J8]); ii) specific families of sets (interval observers with zonotopic sets) and iii) water networks and microgrid benchmarks.

II) Characterization of non-convex regions via mixed-integer formulations. A multitude of control issues reduces to optimization problems over non-convex (and possibly non-connected) regions. I described their underlying structure through hyperplane arrangements, leading to compact mixed integer formulations [J11] and thus greatly reducing the computation requirements. Various constructions have been studied: cell merging exploiting Boolean algebra logic and alternative mixed integer formulations, [B4]. These results hold over different classes of problems which share the same underlying scaffolding: i) FDI constraints in reference governor synthesis [J12]; ii) conditions to bound the Lagrangian multipliers of “hard-constrained” MPC; iii) persistent excitation constraints in MPC; iv) multi-agent control with anti-collision and obstacle avoidance constraints [B2] and v) explicit parametrization of unobservable regions in a field with obstacles. These contributions:

- provide tools for non-convex constraint descriptions often encountered in motion planning problems;
- have led to 2 journals (co-author), 1 book and 1 book-chapter (co-author) and 8 conference papers.
(6 as main author) discussing various improvements and applications;

- are a mainstay of my current research activities and serve as framework for the non-convex optimization problems I tackle; collaborations on these topics are with groups from France, Norway.

III) **Set-theoretic elements in constrained optimization problems.** I have improved upon the state of the art in several directions by: i) adapting existing results to particular families of sets and ii) exploiting the combinatorial structure of constrained optimization problems. I made use of zonotopic sets (polyhedral sets with a higher degree of symmetry) to improve upon analytic characterizations of robust positive invariant (RPI) sets: i) tighter ultimate bounds; ii) an explicit description of the minimal RPI set and analytic RPI approximations of it [B1]; and iii) star-shaped RPI sets in [J8]. Geometrical interpretations in constrained optimization problems (MPC in particular) allowed me to: i) provide efficient computations under zonotopic constraints, ii) characterize the critical regions of the explicit MPC representation (merge those with the same initial input, recursive relations between explicit representations of successive orders). These contributions:

- are used and studied through my entire list of research interests as a set-theoretic framework is eminently suited to gauge a closed-loop system’s stability and performance;
- are used in almost all of my articles; 3 conference papers (main author) and 1 book-chapter (main author) discuss specific improvements in set invariance construction;
- permeate my entire work and have been the defining element which differentiates it from similar approaches (e.g., invariance analysis is at the forefront of research in the FTC field).

IV) **Flat-derived trajectory generation.** I applied flatness theory and B-spline parametrizations for explicit constraint and cost characterizations. Initial work was done in trajectory generation for an unmanned aerial vehicle (UAV): way-point constraints (passing through or within a predefined neighborhood) and trajectory length costs have been considered over simplified dynamics [C5]. Recently in [C28] I provided guarantees for continuous obstacle avoidance (through the exploitation of the convex hull property of the B-spline functions). These contributions:

- represent a main direction of future interest and are essential for motion planning and exact fault detection;
- appear in most of my recent papers: 1 journal (co-author), 1 book chapter (main author), 5 conference papers (all as main author) discuss trajectory generation and coverage issues;
- are inline with current research directions at the international level as they are at the confluence of motion planning, control and robotics communities.

Further information about my current interests and background can be found at [www.florinstoican.com](http://www.florinstoican.com). My Researcher ID is C-3564-2012 and my ORCID ID is 0000-0002-4550-9113.
Chapter 3. Scientific achievements

3.2. Full list of publications

In what follows I am enumerating my publications.

I am the (co-)author of 4 chapters and books:


Published by: SPRINGER-VERLAG BERLIN (HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY)

DOI: 10.1007/978-3-319-26687-9_17. WOS: 000369162800019. EID: 2-s2.0-84954169584.

Cited: Scopus: 1 Web of Science: 1.


Published by: Kluwer Academic Publishers (P.O. Box 17, 3300 AA Dordrecht, the Netherlands)

DOI: 10.1007/978-94-007-7006-5_17. EID: 2-s2.0-84896528456.


Cited: Scopus: 2 Web of Science: 1.


Published by: JOHN WILEY & SONS (THE ATRIUM, SOUTHERN GATE, CHICHESTER, W SUSSEX PO 19 8SQ, ENGLAND)

WOS: 000327043800009. EID: 2-s2.0-85014342801.

Cited: Web of Science: 11.


Published by: SPRINGER INTERNATIONAL PUBLISHING AG (GEWERBESTRASSE 11, CHAM, CH-6330, SWITZERLAND)

DOI: 10.1007/978-3-319-26995-5. WOS: 000415981400008.

I am the (co-)author of 13 ISI-indexed journal articles:


Published by: MDPI AG (ST ALBAN-ANLAGE 66, CH-4052 BASEL, SWITZERLAND)

Sets in control. Applications to fault tolerant control and motion planning.

IF: 2.67 [from JCR 2016, last at submission date]. Q2 (CHEMISTRY, ANALYTICAL - 25/76); Q2 (ELECTROCHEMISTRY - 12/27); Q1 (INSTRUMENTS & INSTRUMENTATION - 10/58) [from JCR 2016, last at submission date].

Citări: Web of Science: 1.


Published by: WILEY-BLACKWELL (111 RIVER ST, HOBBOKEN 07090-5774, NJ USA)

IF: 1.71 [from JCR 2016, last at submission date]. Q3 (AUTOMATION & CONTROL SYSTEMS - 31/60); Q2 (ENGINEERING, ELECTRICAL & ELECTRONIC - 125/259) [from JCR 2016, last at submission date].

Citări: Scopus: 2 Web of Science: 2.


Published by: UNIV SUCEAVA, FAC ELECTRICAL ENG (UNIV SUCEAVA, FAC ELECTRICAL ENG, STEFAN CEL MARE, UNIVERSITATII 13, SUCEAVA, 720229, ROMANIA)

IF: 0.6 [from JCR 2016, last at submission date]. Q3 (COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE - 120/132); Q3 (ENGINEERING, ELECTRICAL & ELECTRONIC - 225/259) [from JCR 2016, last at submission date].

Citări: Scopus: 1.


Published by: PERGAMON-ELSEVIER SCIENCE LTD (THE BOULEVARD, LANGFORD LANE, KIDLINGTON, OXFORD OX5 1GB, ENGLAND)
DOI: 10.1016/j.energy.2015.08.009. WOS: 000365362700003. EID: 2-s2.0-84946026960.

IF: 4.52 [from JCR 2016, last at submission date]. Q1 (ENERGY & FUELS - 17/89); Q1 (THERMODYNAMICS - 3/58) [from JCR 2016, last at submission date].

Citári: Scopus: 10 Web of Science: 12.


Published by: TAYLOR & FRANCIS LTD (4 PARK SQUARE, MILTON PARK, ABINGDON OX14 4RN, OXON, ENGLAND)

IF: 2.29 [from JCR 2016, last at submission date]. Q2 (AUTOMATION & CONTROL SYSTEMS - 24/60); Q1 (COMPUTER SCIENCE, THEORY & METHODS - 24/104); Q1 (OPERATIONS RESEARCH & MANAGEMENT SCIENCE - 21/83) [from JCR 2016, last at submission date].
Chapter 3. Scientific achievements

Citări: Scopus: 16 Web of Science: 12.


Published by: TAYLOR & FRANCIS LTD (4 PARK SQUARE, MILTON PARK, ABINGDON OX14 4RN, OXON, ENGLAND)

IF: 2.29 [from JCR 2016, last at submission date]. Q2 (AUTOMATION & CONTROL SYSTEMS - 24/60); Q1 (COMPUTER SCIENCE, THEORY & METHODS - 24/104); Q1 (OPERATIONS RESEARCH & MANAGEMENT SCIENCE - 21/83) [from JCR 2016, last at submission date].


Published by: PERGAMON-ELSEVIER SCIENCE LTD (THE BOULEVARD, LANGFORD LANE, KIDLINGTON, OXFORD OX5 1GB, ENGLAND)

IF: 3.02 [from JCR 2016, last at submission date]. Q1 (COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS - 20/105); Q1 (ENGINEERING, CHEMICAL - 28/133) [from JCR 2016, last at submission date].

Citări: Scopus: 6 Web of Science: 6.


Published by: WILEY-BLACKWELL (111 RIVER ST, HOBOKEN 07030-5774, NJ USA)

IF: 3.39 [from JCR 2016, last at submission date]. Q1 (AUTOMATION & CONTROL SYSTEMS - 12/60); Q1 (ENGINEERING, ELECTRICAL & ELECTRONIC - 48/259); Q1 (MATHEMATICS, APPLIED - 4/255) [from JCR 2016, last at submission date].

Citări: Scopus: 5 Web of Science: 4.


Published by: ELSEVIER SCI LTD (THE BOULEVARD, LANGFORD LANE, KIDLINGTON, OXFORD OX5 1GB, OXON, ENGLAND)

IF: 2.7 [from JCR 2016, last at submission date]. Q2 (AUTOMATION & CONTROL SYSTEMS - 16/60); Q2 (ENGINEERING, CHEMICAL - 38/133) [from JCR 2016, last at submission date].

Citări: Scopus: 11 Web of Science: 10.

Sets in control. Applications to fault tolerant control and motion planning.

2013, ISSN: 1751-8644.

Published by: INST ENGINEERING TECHNOLOGY-IET (MICHAEL FARADAY HOUSE SIX HILLS WAY STEVENAGE, HERTFORD SG1 2AY, ENGLAND)

IF: 2.54 [from JCR 2016, last at submission date]. Q2 (AUTOMATION & CONTROL SYSTEMS - 20/60); Q2 (ENGINEERING, ELECTRICAL & ELECTRONIC - 81/259); Q1 (INSTRUMENTS & INSTRUMENTATION - 11/58) [from JCR 2016, last at submission date].

Citări: Scopus: 5 Web of Science: 5.


Published by: SPRINGER/PLENUM PUBLISHERS (233 SPRING ST, NEW YORK, NY 10013 USA)

IF: 1.29 [from JCR 2016, last at submission date]. Q2 (MATHEMATICS, APPLIED - 72/255); Q3 (OPERATIONS RESEARCH & MANAGEMENT SCIENCE - 46/83) [from JCR 2016, last at submission date].

Citări: Scopus: 15 Web of Science: 14.


Published by: ELSEVIER SCI LTD (THE BOULEVARD, LANGFORD LANE, KIDLINGTON, OXFORD OX5 1GB, OXON, ENGLAND)

IF: 2.7 [from JCR 2016, last at submission date]. Q2 (AUTOMATION & CONTROL SYSTEMS - 16/60); Q2 (ENGINEERING, CHEMICAL - 38/133) [from JCR 2016, last at submission date].

Citări: Scopus: 19 Web of Science: 18.


Published by: TAYLOR & FRANCIS LTD (4 PARK SQUARE, MILTON PARK, ABINGDON OX14 4RN, OXON, ENGLAND)

IF: 2.21 [from JCR 2016, last at submission date]. Q2 (AUTOMATION & CONTROL SYSTEMS - 26/60) [from JCR 2016, last at submission date].

Citări: Scopus: 53 Web of Science: 50.

I am the (co-)author of 28 ISI-indexed conference proceedings:


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
Chapter 3. Scientific achievements

DOI: 10.1109/MED.2017.7984185. WOS: 000426926300100. EID: 2-s2.0-85027839276.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/MED.2017.7984263. WOS: 000426926300178. EID: 2-s2.0-85028503305.


Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NEDERLANDS)


Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NEDERLANDS)


Published by: SPRINGER-VERLAG BERLIN (HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY)
DOI: 10.1007/978-3-319-21290-6_38. WOS: 000381804400038. EID: 2-s2.0-84983209088.

Citări: Scopus: 1 Web of Science: 1.


Published by: SPRINGER INTERNATIONAL PUBLISHING AG (GEWERBESTRASSE 11, CHAM, CH-6330, SWITZERLAND)
DOI: 10.1007/978-3-319-48680-2_63. WOS: 000390177400063. EID: 2-s2.0-84994417096.

Citări: Scopus: 2.


Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NEDERLANDS)

Sets in control. Applications to fault tolerant control and motion planning.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/ICSTCC.2015.7321386. WOS: 000382384100127. EID: 2-s2.0-84957827736.

Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/MED.2015.7158827. WOS: 000375056800107. EID: 2-s2.0-84945967558.
Citări: Scopus: 1 Web of Science: 1.

Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
WOS: 000332509703092. EID: 2-s2.0-84893235262.
Citări: Scopus: 3 Web of Science: 3.

Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/SysTol.2013.6693849. WOS: 000331082600008. EID: 2-s2.0-84897677149.
Citări: Scopus: 1 Web of Science: 1.

Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/SysTol.2013.6693860. WOS: 000331082600009. EID: 2-s2.0-84897742081.
Citări: Scopus: 1 Web of Science: 1.

Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/MED.2013.6608834. WOS: 000333245100145. EID: 2-s2.0-84885230747.
Citări: Web of Science: 1.

Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/MED.2013.6608816. WOS: 000333245100127. EID: 2-s2.0-84885207795.

Cățări: Scopus: 1 Web of Science: 1.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/CDC.2013.6760377. WOS: 000352223503108. EID: 2-s2.0-84902322024.

Cățări: Scopus: 1 Web of Science: 1.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/CDC.2013.6760564. WOS: 000352223505003. EID: 2-s2.0-84902322023.

Cățări: Scopus: 4 Web of Science: 4.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/CCA.2012.6402730. WOS: 000320336200062. EID: 2-s2.0-84873161973.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/CDC.2011.6161361. WOS: 000303506204097. EID: 2-s2.0-84860687872.

Cățări: Scopus: 2 Web of Science: 1.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/CDC.2011.6161317. WOS: 000303506202138. EID: 2-s2.0-84860652919.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
WOS: 000295376002086. EID: 2-s2.0-80053161211.

Cățări: Scopus: 4 Web of Science: 1.


Published by: INSTICC-INST SYST TECHNOLOGIES INFORMATION CONTROL & COMMUNICATION (AVENIDA D MANUEL L, 27A 2 ESQUERDO, SETUBAL, 2910-595,
PORTUGAL. Applications to fault tolerant control and motion planning.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/CACSD.2010.5611187. WOS: 000286943900232. EID: 2-s2.0-78649399354.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/SYSTOL.2010.5675962. WOS: 000416069400089. EID: 2-s2.0-78751667497.

Citări: Scopus: 1


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/SYSTOL.2010.5675958. WOS: 000416069400077. EID: 2-s2.0-78751669594.

Citări: Scopus: 2 Web of Science: 2


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
DOI: 10.1109/CDC.2010.5718146. WOS: 000295049101001. EID: 2-s2.0-79953130241.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
WOS: 000287187904073. EID: 2-s2.0-77957806550.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
WOS: 000427419900098.


Published by: IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)
WOS: 000391154900167.

I am the (co-)author of 12 BDI-indexed conference proceedings:

1The database used was Scopus.

**Published by:** ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)

DOI: 10.1016/j.ifacol.2015.11.281. EID: 2-s2.0-84964219468.


Citări: Scopus: 1 Web of Science: 1.


**Published by:** ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)


**Published by:** IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)

DOI: 10.1109/ICSTCC.2014.6982514. EID: 2-s2.0-84929448955.


**Published by:** ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)

EID: 2-s2.0-84929815826.


**Published by:** IEEE (345 E 47TH ST, NEW YORK, NY 10017 USA)

DOI: 10.1109/ICSTCC.2014.6982443. EID: 2-s2.0-84929431542.
Sets in control. Applications to fault tolerant control and motion planning.

BDI: Scopus (https://www.scopus.com/inward/record.uri?eid=2-s2.0-84929431542&doi=10.1109%2fICSTCC.2014.6982443&partnerID=40&md5=0df65e98aa8de3b0cfaa9d36e916ad49).


Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)
EID: 2-s2.0-84929773777.
BDI: Scopus (https://www.scopus.com/inward/record.uri?eid=2-s2.0-84929773777&partnerID=40&md5=cb34ebc1f173d8d8570fa730c83e8e1).


Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)
DOI: 10.3182/20120823-5-NL-3013.00079. EID: 2-s2.0-84867056750.


Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)
DOI: 10.3182/20120823-5-NL-3013.00054. EID: 2-s2.0-84867067744.

Citări: Scopus: 5 Web of Science: 2.


Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)

Citări: Scopus: 6 Web of Science: 3.

Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)
DOI: 10.3182/20110828-6-IT-1002.03247. EID: 2-s2.0-84866769502.
Citări: Scopus: 5 Web of Science: 3.


Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)
DOI: 10.3182/20110828-6-IT-1002.01842. EID: 2-s2.0-84864613111.
Citări: Scopus: 22 Web of Science: 16.


Published by: ELSEVIER SCIENCE BV (PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS)
Citări: Scopus: 3 Web of Science: 2.
Part II.

Scientific results
4. Explicit descriptions of robust invariant sets

In this chapter I present a robust positive invariance (RPI) outer-approximation of the minimal RPI (mRPI) set associated to linear dynamics with zonotopic disturbances. I show that the candidate sets considered are either RPI or become so with a scaling factor. The results base on the concomitant computation of extremal points and their extremal hyperplanes. Further, I consider the equivalence with ultimate bounds constructions and show that successive RPI representations become monotonically “tighter” as their complexity increases.

The chapter is based on book chapter [B1]:


DOI: 10.1007/978-3-319-26687-9_17. WOS: 000369162800019. EID: 2-s2.0-84954169584.

4.1. Preliminaries

The notions of *positive invariance (PI)* and its robust counterpart, *robust positive invariance (RPI)* are fundamental in a large number of control topics. I may mention reference governor synthesis [53], predictive controllers with terminal constraints [103], robust time-optimal control [18], safe collision avoidance [132, 133] or fault tolerant control [B3] as areas which make use of these notions.

The common thread in all these applications is that the (robust) invariance permits to ‘push’ the difficult computations (the set construction) into an offline phase and then, at runtime, only simple computations remain to be made.

As the domain is quite large we concentrate in the rest of the chapter on the *minimal RPI (mRPI) set* which represents the smallest set which is still RPI for a given dynamic (its counterpart is the maximal RPI (MRPI) set, which is the largest invariant set respecting given constraints). Further, we will consider linear dynamics with zonotopic disturbances (a symmetric subset of the polyedral sets). These assumptions not only allow easier computations but actually provide a theoretical framework.
for the results shown hereafter.

The mRPI set, barring some particular cases, has no finite description of its boundary. Consequently, a great deal of research was directed towards finding RPI approximations of the mRPI set. The existing results can be classified mainly into: i) iterative and ii) explicit methods. In the former class we consider all the methods which take an initial set and through a recursive iteration improve the approximation by exploiting the contractive properties of the linear mapping, this includes [11, 138] or [J13]. The latter gives an explicit formulation of the boundary of the set, a classic example being the sublevels of the quadratic Lyapunov functions or ultimate bounds formulation [80]. The main parameters that characterize these methods are fidelity of the representation and numerical complexity. With respect to these requirements, the aforementioned procedures have a complementary behavior. The iterative procedures can approximate arbitrarily well ("ε"-outer approximations) but have an exponential increase in the computational effort whereas the explicit formulations are simple to deduce but are conservative.

It is then worthwhile to seek a method which combines the best aspects of both classes: fast computation and accurate representation of the mRPI set. To this end we consider [67, 68] where an explicit representation of the boundary of the null-controllable set was provided (apparently a well-known technique in the state of the art [115]). We have studied these constructions from the point of view of extremal representations in a previous work [C15]. We will point in the present chapter that the notions are readily adapted to the mRPI set and use the techniques described in the respective context for our own ends: the construction of an RPI approximation of the mRPI set with a minimum of computations.

We expand by exploiting additional structure of the boundary. That is, we consider both the extremal points defining the boundary and their associated extremal hyperplanes. This allows to obtain outer-approximations which may be directly RPI or, if not, become so after scaling with a finite scalar. This is in contrast with the work in [C15] where all the constructions start with inner-approximations and therefore require a scaling factor regardless of the set. Further, we show that the construction is strongly related with the notion of ultimate bounds [80] and that the RPI approximations which we obtain are increasingly 'tighter' around the mRPI set proportionally with the complexity of the representation.

4.2. Prerequisites

In order to introduce the main ideas of the chapter we recapitulate some basic invariance results (see, e.g. [22]). Let us consider the following LTI system in $\mathbb{R}^n$:

$$x^+ = Ax + \delta$$ (4.1)

where $A \in \mathbb{R}^{n \times n}$ is a Schur matrix; $x$ and $x^+$ represent the current and successor state of the system, respectively and $\delta$ is a disturbance bounded by $\Delta \subset \mathbb{R}^n$ which is convex, compact, contains the origin in its non-empty interior and is bounded.

System (4.1) is used next for defining basic invariance notions [21].

**Definition 4.1.** Under dynamics (4.1), a set $\Omega \subset \mathbb{R}^n$ is called positive invariant if $A\Omega \subseteq \Omega$ and robust positive invariant set if $A\Omega \oplus \Delta \subseteq \Omega$. 

25 of 160
The mRPI set associated to (4.1), denoted for further use as \( \Omega_{\infty}(A, \Delta) \), is defined as the RPI set contained in any closed RPI set. This is known to be unique, compact and – in the case when \( \Delta \) contains the origin – to contain the origin \([81]\).

An alternative representation is to define it as the limit set to the set recurrence relation\(^1\) \( \Omega_k(A, \Delta) = A\Omega_{k-1}(A, \Delta) \oplus \Delta \) (with \( \Omega_0 = \{0\} \)):

\[
\Omega_\infty(A, \Delta) \triangleq \lim_{k \to \infty} \Omega_k(A, \Delta) = \bigoplus_{i=1}^{\infty} A^i \Delta. \tag{4.2}
\]

The set \( \Omega_k(A, \Delta) \) is the \( k \)-reachable set under dynamics (4.1) starting from \( \{0\} \).

As stated earlier, it is not possible to compute an exact representation of (4.2), except under restrictive assumptions such as when matrix \( A \) is nilpotent \([104]\). Hence, approximations have to be used and various algorithms for the construction of RPI approximations exist in the literature. We point to the necessity that the approximation is itself an RPI set (otherwise the guarantee that a certain signal remains in a bounded area is no longer valid).

The above invariance notions hold for any kind of bounded disturbance set. Hereafter, in order to apply the techniques from \([67]\) (and obtain meaningful results), we restrict our analysis to the case of zonotopic disturbances.

Zonotopes represent a particular class of polytopes characterized by the following definition.

**Definition 4.2.** The subset of \( \mathbb{R}^n \) with center \( c \) and set of generators \( \mathcal{B} \triangleq \{b_1, \ldots, b_m\} \subset \mathbb{R}^n \), such that

\[
Z(c, \mathcal{B}) = \left\{ x \in \mathbb{R}^n : x = c + \sum_{i=1}^{m} \lambda_i b_i, |\lambda_i| \leq 1, \; b_i \in \mathcal{B} \right\}
\]

with \( i = 1, \ldots, m \) is called a zonotope.

A zonotope has the following properties \([50]\):

i) is closed under linear transformation:

\[
LZ(c, \mathcal{B}) = Z(Lc, L\mathcal{B}); \tag{4.3}
\]

ii) is closed under Minkowski sum:

\[
Z(c_1, \mathcal{B}_1) \oplus Z(c_2, \mathcal{B}_2) = Z(c_1 + c_2, \mathcal{B}_1 \cup \mathcal{B}_2). \tag{4.4}
\]

These properties together with the associative property of the Minkowski addition lead to some interesting results.

**Proposition 4.1** (Proposition 1, \([C15]\)). Consider the dynamics (4.1) with a zonotopic disturbance set \( \Delta \). Then, its associated mRPI set, \( \Omega_{\infty}(A, \Delta) \), verifies the following relations:

i) for any \( k \geq 1 \), \( \Omega_{\infty}(A, \Delta) = \Omega_{\infty}(A^k, A^{k-1} \Delta \oplus \cdots \oplus \Delta) \);

ii) given \( \Delta = Z(c, \mathcal{B}) \), the mRPI set can be decomposed as \( \Omega_{\infty}(A, \Delta) = \left\{ (I - A)^{-1} c \right\} \oplus \bigoplus_{i=1}^{m} \Omega_{\infty}(A, \Delta_i) \), where \( \Omega_{\infty}(A, \Delta_i) \) is the mRPI set associated with dynamics \( x^+ = Ax + \delta_i \) and where \( \delta_i \in \Delta_i \triangleq \{\lambda_i b_i, |\lambda_i| \leq 1\} \).

\(^1\)The convergence to a finite limit is guaranteed by the stability of the system and by the boundedness of the perturbation.
Proof. See the proof of Proposition 1, [C15].

Note that the relation between the mRPI set described in case (ii) of Proposition 4.1 also holds for the more general case where the disturbance set $\Delta$ is a Minkowski sum of sets: $\Delta = \Delta_1 \oplus \cdots \oplus \Delta_m$.

In the rest of the chapter we consider a particular case of dynamics of form (4.1).

Assumption 4.1. Consider the dynamics

$$x^+ = Ax + b\lambda$$  \hspace{1cm} (4.5)

with the following particularities:

i) the eigenvalues of matrix $A \in \mathbb{R}^{n \times n}$ are real and positive;

ii) the disturbance varies along a segment defined by the fixed vector $b \in \mathbb{R}^n$. With respect to (4.1) this is equivalent with $\delta \in \Delta = \{\lambda b, |\lambda| \leq 1\}$.

Taking into account Proposition 4.1 we observe that the assumptions affecting (4.5) are manageable, that is, the dynamics can be relaxed up to a system which has: 1) a state matrix with real eigenvalues and 2) zonotopic disturbances. Firstly, notice that since $A^2$ has only positive eigenvalues regardless of the sign of the eigenvalues of $A$ (as long as they are real) it follows that we can compute the mRPI set for dynamics $A^2 \to A$ and $A\Delta \oplus \Delta \to \Delta$ and use it for the original dynamics (4.5) (see case (i)) for $k = 2$ without any loss in the mRPI representation. Secondly, using case (ii) we have that it suffices to compute the mRPI sets for each of the segments composing the zonotopic disturbance and to add them (in the Minkowski sense) at the end of the procedure to recover the mRPI corresponding to dynamics (4.1).

Remark 4.1. The above discussion is important since it makes the link to more general dynamics, in the sense of a state matrix which is Schur, regardless of the sign of the matrix’ eigenvalues. Coupled with the cases covered earlier we cover a reasonably large number of dynamics (keep in mind that usually the perturbations are given as magnitude conditions which can be then modeled as zonotopic sets). Note also that complex eigenvalues and the continuous time case can be discussed [67].

### 4.3. Explicit representation of the mRPI set

The results described here are based on [67] where the shape of the null-controllable set of an anti-stable dynamic is studied. With some minor changes, the same reasoning can be applied to the computation of minimal RPI sets for stable dynamics. Recalling (4.2), the mRPI set for dynamics (4.5) can be seen as the collection of trajectories starting from an initial state $x_0 = \{0\}$ and taking all possible noise realizations:

$$\Omega_\infty(A, \Delta) = \lim_{k \to \infty} \bigcup_{|\lambda_i| \leq 1} \left\{ \sum_{l=1}^{k} A^{k-l} b\lambda_l \right\}$$ \hspace{1cm} (4.6)

Not all these trajectories will result in extremal points (points which lie on the boundary of $\Omega_\infty(A, \Delta)$). A sequence of disturbances $(\lambda_1, \lambda_2, \ldots)$ is called extremal if it steers the trajectory starting from $\{0\}$ into an extremal point (a vertex) of $\Omega_\infty(A, \Delta)$. The collection of these extremal sequences is denoted
as $\Sigma$ and is given by all “bang-bang” disturbance realizations which have at most $n - 1$ switches (see Lemma 3 of [68]):

$$\Sigma = \{ \pm(\lambda_1, \lambda_2, \ldots) : \lambda_l = \begin{cases} 1, & 1 \leq l < i_1, \\ (-1)^j, & i_j \leq l < i_{j+1}, \\ (-1)^{n-1}, & i_{n-1} \leq l \leq \infty \end{cases}, \quad (i_1, \ldots, i_{n-1}) \in I_{n-1}^k \}, \quad (4.7)$$

where

$$I_{n-1}^k = \{ (i_1, \ldots, i_{n-1}) : 1 \leq i_1 \leq \cdots \leq i_{n-1} \leq k \} \quad (4.8)$$

denotes the collection of all switching sequences from $\mathbb{R}^n$ where we have at most ‘n-1’ switches (to which correspond ‘n’ intervals along the time domain) and where the last switch happens no later than the $k$ instant.

**Lemma 4.1.** To an extremal sequence from (4.7) characterized by switching times $i = \{i_1, \ldots i_{n-1}\}$ as in (4.8) corresponds a pair of normal vector and extremal points $(c_i^\top, \pm x_i^*)$ defined as:

i) the normal vector $c_i \in \mathbb{R}^n$ which respects

$$c_i^\top A^j b = 0, \quad \forall i_j \in i; \quad (4.9)$$

ii) extremal point $x_i^* \in \mathbb{R}^n$

$$x_i^* = \left( 2 \sum_{j=1}^{n-1} (-1)^j A^j + (-1)^n I \right) (I - A)^{-1} b. \quad (4.10)$$

**Proof.** These results are derived in [67] for the continuous domain where it is noted that $c_i^\top e A^j b$ has at most $n - 1$ sign changes (‘switches’). This holds for a matrix $A$ with positive eigenvalues. After various computations it is shown that the disturbance sequence $\lambda(l) = \text{sgn}(c_i^\top A^j b)$ is extremal. Based on this, in [68] it is shown that for a given switching sequence correspond both the extremal hyperplane characterized by the normal vector verifying (4.9) and the extremal point (4.10). 

Using Lemma 4.1 we provide a dual representation of the mRPI set (4.6).

**Proposition 4.2.** For dynamics (4.5), the associated mRPI set is given in

i) half-space representation:

$$\Omega_{\infty}(A, \Delta) = \{ x : |c_i^\top x| \leq c_i^\top x_i^*, \quad \forall i \in I_{n-1}^k \} \quad (4.11)$$

ii) generator representation (i.e., as convex hull of its extremal points):

$$\Omega_{\infty}(A, \Delta) = \text{conv}_{i \in I_{n-1}^k} (\pm x_i^*) \quad (4.12)$$

**Proof.** (i) The vector $c_i$ represents the direction along which we find the maximal and minimal values of the mRPI, the extremal points $\pm x_i^*$. Therefore we obtain two extremal hyperplanes which bound

---

Hereinafter, without any loss of generality, we make the convention that $c_i^\top x_i^* \geq 0$. 

28 of 160
Sets in control. Applications to fault tolerant control and motion planning.

the mRPI: $c_i x = c_i x_i^*$ and $c_i x = -c_i x_i^*$. Assuming, as in footnote 2 that the product $c_i x_i^*$ is positive, it follows that the mRPI set lies in $|c_i^T x| \leq c_i^T x_i^*$ which leads to (4.11).

(ii) Since by definition the points $\pm x_i^*$ are extremes of the mRPI, the relation (4.12) follows immediately.

Several remarks are in order.

Remark 4.2. Note that there are three elements which interlock to give the boundary description: the switching sequence $i$, the normal vector $c_i^T$ and the extremal point $x_i^*$. Depending on the desired approach, we may: i) provide a switching sequence $i$ and introduce it in (4.9) and (4.10) to obtain $c_i^T$ and $x_i^*$ respectively; or ii) provide a vector $c_i^T$, obtain the sequence $i$ for which (4.9) is verified and use it in (4.10) to obtain $x_i^*$.

Lastly, we recall the dynamical interpretation given to the extreme points (4.10).

Remark 4.3. Let $x_{e,\pm} = \pm (I - A)^{-1} b$, the equilibrium point of (4.5) for constant disturbance $\lambda(k) = \pm 1$. It can be noted that half of the points in (4.10) are formed by trajectories of (4.5) starting from $x_{e,+}$ under any bang-bang sequence with $n - 2$ or less switches. The other half is symmetric with the first part and consists of the trajectories starting from $x_{e,-}$ under any bang-bang sequence with $n - 2$ or less switches, see [68] for further details. In particular this means that no point (except $x_{e,\pm}$) remains fixed on the boundary, that is, the boundary itself is invariant but it flows in-between the two equilibrium points, see [38] for an analysis of the mRPI boundary behavior for nonlinear dynamics.

Illustrative example

For the purpose of illustration let us consider the two-dimensional case. Keeping the notation of Lemma 4.1, the switching sequence reduces to a single switch: $i = \{i_1\}$, as implied by $n - 1 = 1$. Consequently, the sequence of extremal noise realizations becomes

$$\Sigma = \left\{ \pm (\lambda_1, \lambda_2, \ldots) : \lambda_l = \begin{cases} 1, & 1 \leq l \leq i_1, \\ -1, & i_1 \leq l \leq \infty \end{cases}, i_1 \in \mathbb{N}_\infty \right\}$$

which simplifies the formulations of both (4.9) and (4.10) into:

$$c_i^T A^{i_1} b = 0,$$

$$x_i^* = \pm (2 A^i - I) (I - A)^{-1} b. \tag{4.14a} \tag{4.14b}$$

Further, let us consider the LTI dynamics

$$x^+ = \begin{bmatrix} 0.91 & -0.07 \\ 0.01 & 0.79 \end{bmatrix} x + \begin{bmatrix} 0.03 \\ 0.31 \end{bmatrix} \lambda, \quad |\lambda| \leq 1. \tag{4.15}$$

Over these dynamics we highlight the theoretical results from above. First, we take an arbitrary switching time $i_1 = 5$ and depict in Figure 4.1a the resulting extremal hyperplanes and corresponding extremal points, as resulting from Lemma 4.1:

$$c_i = \begin{bmatrix} 0.7114 \\ 0.2886 \end{bmatrix}, \quad x_i^* = \pm \begin{bmatrix} -0.7176 \\ -0.6062 \end{bmatrix}.$$
Note that (4.14a) does not uniquely define $c_i \in \mathbb{R}^2$ — it only defines the normal vector direction but not its length. To take a unique $c_i$ we add an arbitrary constraint, $c_i^\top \begin{bmatrix} 1 & 1 \end{bmatrix} = 1$. Further, as in Proposition 4.2, we depict the resulting extremal hyperplanes. We illustrate these against the ‘real’ mRPI set (obtained as in (4.12) with a large number of extremal points) and it can be seen that indeed (4.14a) and (4.14b) define extremal hyperplanes for the mRPI set. Moreover, we check in Figure 4.1b that $c_i A^i b$ switches its sign at the desired value $i_1 = 5$ which means that the product $c_i^\top x$ is maximized for the sequence of disturbances $\{1, 1, 1, 1, -1, -1, \ldots \}$.

Further we show in Figure 4.1c the half-space construction (4.11) by enumerating pairs (4.14a)–(4.14b) for a sequence of switching instants $I_{10}^2 = \{0, 1, \ldots, 10\}$ and $\{\infty\}$.

As discussed in Remark 4.3, it is possible to describe the extreme points of the mRPI as two trajectories starting from $x_{e,+}$ and $x_{e,-}$, respectively. E.g., for the two-dimensional case, it means that starting from $x_{e,-}$ and considering a constant $(n - 2 = 0$ switches) disturbance, $\lambda(l) = 1$ for any $l \geq 0$ we pass through half of the extreme points (4.14b). The converse holds when starting from $x_{e,+}$ and considering a constant disturbance $\lambda(l) = -1$ for any $l \geq 0$. We depict this in Figure 4.1d.

As stated earlier these mRPI constructions hold for the restricted dynamics (4.5). In particular,
they are not applicable for state matrices with negative eigenvalues. For illustration we take dynamics

\[
x^+ = \begin{bmatrix}
-1.0559 & 1.1978 \\
-0.1711 & 0.9975
\end{bmatrix} x + \begin{bmatrix} 0.03 \\ 0.31 \end{bmatrix} \lambda, \quad |\lambda| \leq 1.
\]  

(4.16)

The state matrix has a negative eigenvalue and therefore we apply Proposition 4.1 and consider two auxiliary dynamics where the state matrices are positive ‘\(x^+ = A^2 x + Ab\)’ and ‘\(x^+ = A^2 x + b\lambda\)’ (case (i) of the proposition) and then combine the resulting mRPI sets to retrieve the mRPI set for dynamics (4.16), case (ii) of the proposition. These operations are depicted in Figure 4.2a and Figure 4.2b, respectively.

4.4. RPI constructions using \(\Omega_\infty(A, \Delta)\)

The main idea of this chapter is to use the constructions stated in Section 4.3 to provide RPI approximations of the mRPI set. With respect to previous work in [C15] we consider here both the extremal points and their corresponding extremal hyperplanes. This allows to provide half-space outer-approximations of the mRPI set (in contrast with the inner-approximations from [C15]).

Having an explicit description of the mRPI’s boundary does not suffice since in practice we can consider only a finite complexity (finite number of extremal hyperplanes or extremal points\(^3\)). In [C15] finite sequences of extremal points \(\{\pm x_1^\alpha\}\) have been considered together with scaling factors which make the resulting inner approximation into an RPI set.

Here we pursue the dual approach, that is, we consider the half-space representation and inquire about the positive robust invariance of the resulting sets. To this end, let us consider a sequence of \(N \geq n\) pairs of extremal hyperplanes and extremal points \(\{e_i^T, \pm x_1^\alpha\}_{i \in I}\), where \(I \subset I_n\) gathers a collection of switching sequences \(i = \{i_1, \ldots, i_{n-1}\}\).

We can now define the set

\[
S(I) = \{z : |e_i^T x| \leq d_i, \forall i \in I\},
\]

(4.17)

\(^3\)This dual approach comes from the polyhedral sets definition which allows the equivalence between generator representation and half-space representation.
where\(^4\) \(d_i \triangleq c_i^\top x_i^*\) and for compactness we denote \(C_I^\top \triangleq \begin{bmatrix} \cdots & c_1 & \cdots \end{bmatrix}^\top\) and \(d_I = \begin{bmatrix} \cdots & d_i & \cdots \end{bmatrix}^\top\).

**Proposition 4.3.** Assuming the set \(S(I)\) defined as in \((4.17)\) and a scalar \(\mu \in \mathbb{R}^+\), the set \(\mu S(I)\) is RPI under dynamics \((4.5)\) iff

\[
|c_i^\top b| \leq \mu |I - |H||; \forall i \in I. \quad (4.18)
\]

**Proof.** As per \([19]\), it is known that the robust invariance of \(\mu S(I)\) under dynamics \((4.5)\) is validated iff \(\exists H \in \mathbb{R}^{N \times N}\) s.t. \(C_I^\top A = H C_I^\top\) and \(|C_i^\top b| \leq \mu \cdot |I - |H||d_i|\). The later inequality comes from forcing \(|c_i^\top x^*| \leq d_i\) and using the equality given a priori.

Since we assumed that \(N \geq n\) it follows that \(C_I^\top\) is ‘tall’ and full rank which means that it accepts a left pseudo-inverse defined as \(C_I^+ = (C_I C_I^\top)^{-1} C_I\). This guarantees the existence of \(H\). Note that the i-th element of \(d_i\) is given by \(c_i^\top x_i^* = c_i^\top (I - A)^{-1} \left( 2 \sum_{j=1}^{n-1} (-i)^j A^{j} + (-1)^n I \right) b\) where term \((I - A)^{-1} = (I - C_I^+ H C_I^\top)^{-1}\), which, via the ‘Woodbury matrix identity’ transforms into \(C_I^+ (I - (C_I^+ C_I^\top - H))^{-1} C_I^\top\) which means that \(|c_i^\top b| \leq |I - |H||; \forall i \in I\) becomes \((4.18)\). The minimal scaling factor is then found by searching iteratively for the minimal \(\mu\) which validates the constraint for all \(i \in I\). \(\square\)

While \(\mu\) can be easily obtained as the result of an LP optimization problem, it is less obvious what is the link between \(\mu\) and the collection of switching sequences \(I\) or if the LP problem is feasible at all. Still, it has been shown in \([C15]\) that two particular sequences led to monotonically decreasing scaling factors, so we can expect that similar inferences can be drawn as well here. In addition, we note that any formulation of form \((4.17)\) is an outer-approximation (i.e., it contains the mRPI set).

From Lemma 4.1 and Remark 4.2 we recall that we may choose the normals of the extremal hyperplanes as we desire. Taking into account the structure of matrix \(A\) we have the following lemma.

**Lemma 4.2.** Let there be \(V, \Lambda \in \mathbb{R}^{n \times n}\) such that they describe the eigen-decomposition of matrix \(A\) (i.e., \(A = V \Lambda V^{-1}\) where \(\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)\) and \(\lambda_i\) are the eigenvalues of matrix \(A\)). We take the normal vector \(c_i^\top = [V^{-1}]_i\) as the i-th row of the inverse of matrix \(V\) and we have that

i) the switching sequence which define \(c_i\) is

\[
I_e = \{i_1 = \cdots = i_{n-1} = \infty\}. \quad (4.19)
\]

ii) to which corresponds the extremal point

\[
x_i = (I - A)^{-1} b. \quad (4.20)
\]

**Proof.** Note that \(A^k = V \Lambda^k V^{-1}\) which means that \(c_i^\top A^k b\) becomes \(c_i^\top V \Lambda^k V^{-1} b\). Coupled with the choice of \(c_i = [V^{-1}]_i\) we have that \(c_i^\top A^k b = \lambda^k \cdot e_i^\top V^{-1} b\) which means that the mapping never changes sign. From this it follows that the switching happens at ‘infinity’ and thus we reach \((4.19)\) and \((4.20)\). \(\square\)

\(^4\)We assume without loss of generality that \(c_i^\top\) is taken such that \(d_i \geq 0\).
Corollary 4.1. The set $S(I_0)$ is RPI.

Proof. We revisit (4.18) with the normal vectors being the rows of $V^{-1}$ (taken as in Lemma 4.2) which means that $N = n$ and therefore $C^+_I$ is an invertible matrix, i.e., $C^+_I = (C^+_I)^{-1} = V$ and $H = \Lambda$. This simplifies the constraint (4.18) into a more manageable form: the $i$-th element of the right-side term becomes $e_i^T \mu \cdot (I - H) d_i = [\mu \cdot (I - H) d_i]_i$. This reduces to $\mu(1 - \lambda_i)d_i = \mu(1 - \lambda_i)c_i^+ x_i^*$. Noting that $(I - A)^{-1} = V(I - \Lambda)^{-1}V^{-1}$, the fact that $A^{ij}(I - A)^{-1} = (I - A)^{-1}A^{ij}$ and using these in the definition of $x_i^*$ we reach

$$
|c_i^+ b| \leq \mu(1 - \lambda_i)c_i^+ V(I - \Lambda)^{-1}V^{-1} \cdot \left( 2 \sum_{j=1}^{n-1} (-i)^j A^{ij} + (-1)^n I \right) b
$$

(4.21)

which becomes (by noting that $c_i^+ V = e_i^+$)

$$
|c_i^+ b| \leq \mu(1 - \lambda_i)e_i^+(I - \Lambda)^{-1}V^{-1} \cdot \left( 2 \sum_{j=1}^{n-1} (-i)^j A^{ij} + (-1)^n I \right) b
$$

(4.22)

Then, we have that (by noting that $e_i^+(I - \Lambda)^{-1}V^{-1} = \frac{1}{1 - \lambda_i} e_i^+ V^{-1} = \frac{1}{1 - \lambda_i} c_i^+$)

$$
|c_i^+ b| \leq \frac{1}{1 - \lambda_i} \left( 2 \sum_{j=1}^{n-1} (-i)^j A^{ij} + (-1)^n I \right) b
$$

(4.23)

where, recalling that $c_i^+ A^{ij} b = 0$ by construction, we simplify to

$$
|c_i^+ b| \leq \mu \cdot (-1)^n c_i^+ b.
$$

(4.24)

With $c_i^+$ chosen such that the right side of the equation is positive, we have that the right side is in fact equal to the left side for $\mu = 1$. In other words, $S(I_0)$ is RPI.

Remark 4.4. In [80] and related papers the ‘ultimate bounds’ construction is employed. For dynamics (4.5) the set is described as

$$
S_{UB}(A, \Delta) = \{ x : |V^{-1} x| \leq (I - |\Delta|)^{-1} |V^{-1} b| \}.
$$

(4.25)

With the additional assumptions made in this chapter, $\Lambda > 0$ and that $c_i^+ b > 0$ (4.25) reduces to

$$
S_{UB}(A, \Delta) = \{ x : |V^{-1} x| \leq (I - \Lambda)^{-1} V^{-1} b \}
$$

(4.26)

which is in fact equivalent to the construction from Corollary 4.1. Further, this matches with a result from [C38] where it has been shown that under certain assumptions the set (4.25) is tight (i.e., it touches the mRPI set). This is also the case here since (4.20) are extremal points of the mRPI set.

As stated earlier, (4.10) provides an explicit description of the mRPI set boundary. This description involves an infinity of terms and thus cannot be used in practice. We observe that the extremal points given in (4.10) agglomerate towards either of the fixed points $x_{e,-}$ and $x_{e,+}$. Further, the points closer
to the fixed points are generated with *bang-bang* sub-sequences appearing at a latter index in the construction (4.10). Using these two facts we have that:

i) we can keep a finite subset of points by discarding the ones closer to \( x_{e,+} \) and \( x_{e,-} \);

ii) this subset of points is defined by bang-bang sequences happening in a finite time (that is, all the \( n-1 \) switches are done in a finite time).

**Lemma 4.3.** Under dynamics (4.5), for any \( k \in \mathbb{N} \), the set
\[
R_k^c(A, \Delta) \triangleq S(I_k^c) \cup L_c
\]
respects relation
\[
R_{k+1}^c(A, \Delta) = A R_k^c(A, \Delta) \oplus \Delta.
\]

*Proof.* The proof follows from the definition of the mRPI’s boundary and Remark 4.3.

**Proposition 4.4.** Under dynamics (4.5), for any \( k \in \mathbb{N} \),

i) \( R_k^c(A, \Delta) \) is RPI;

ii) for a set
\[
\Omega(I) = S(I \cup L_c)
\]
there exists \( \gamma \geq 1 \) s.t. \( \gamma \Omega(I) \) is RPI.

*Proof.* (i) We have that \( I_k^c \subset I_{k+1}^c \) since all switching sequences where the last switch happens not later than \( k \) are automatically happening not later than \( k + 1 \). This means that \( R_{k+1}^c(A, \Delta) \subset R_k^c(A, \Delta) \) since the former contains all the extremal hyperplanes of the later. Considering Lemma 4.3 as well, it follows that \( A R_k^c(A, \Delta) \oplus \Delta \subset R_k^c(A, \Delta) \) which means that \( R_k^c(A, \Delta) \) is RPI.

(ii) Let there be \( k = \min_l \) s.t. \( (i_1, \ldots, i_{n-1}) \in I, i_{n-1} \leq l \), that is, we identify the latest switch from any of the switching sequences of \( I \). It follows then that \( R_k^c(A, \Delta) \subset \Omega(I) \) since \( R_k^c(A, \Delta) \) contains all the extremal hyperplanes which appear in \( \Omega(I) \). Further, let there be a scalar \( \gamma \geq 1 \) s.t. \( \Omega(I) \subset R_k^c(A, \Delta) \oplus 2^{-1}A^{-1} \Delta \). It follows then that \( A \Omega(I) + \frac{1}{\gamma} \Delta \subset AR_k^c(A, \Delta) \oplus 2^{-1} \Delta + \frac{1}{\gamma} \Delta = AR_k^c(A, \Delta) \oplus \Delta \). Combined with the initial inclusion and case (i) it follows that \( A \Omega(I) + \frac{1}{\gamma} \Delta \subset \Omega(I) \) which means there exists \( \gamma \geq 1 \) s.t. \( \gamma \Omega(I) \) is RPI.

The constructions discussed in Section 4.4 are obtained directly through the enumeration of extremal hyperplanes (and eventually a scaling). Thus, the computationally difficult task of calculating recursive Minkowski sums is avoided (as would be the case in the iterative procedures which assure arbitrarily close outer-approximations). To provide a measure of the storage requirements we give the next result.

**Proposition 4.5.** The number of extremal hyperplanes defining the boundary of \( R_k^c(A, \Delta) \) is given by:
\[
\#R_k^c(A, \Delta) = 2 \cdot \left( 1 + \sum_{i=0}^{n-1} \binom{k}{i} \right),
\]
Proof. The number of normal vectors is actually the number of sequences of at most \( n - 1 \) ordered elements taken from the first \( k \) natural numbers. Thus, we choose first 0, then 1 an so forth until \( n - 1 \) from the first \( k \) integers and obtain the terms in (4.30), with the addition of the two extremal hyperplanes ‘at infinity’. \( \Box \)

Illustrative example

Let us consider again the dynamics (4.15) and apply the RPI approximations presented in Proposition 4.4. We start with the constructions (4.27) and depict in Figure 4.3a the set \( R_3^e(A, \Delta) = S(I^2 \cup I_e) \) – solid blue contour. For comparison we add the set \( R_4^e(A, \Delta) \) – dashed blue contour and the ‘real’ mRPI set – dotted red contour. Note that, as per Lemma 4.3, we have that \( R_4^e(A, \Delta) = A R_3^e(A, \Delta) \odot \Delta \).

Figure 4.3.: RPI approximation of the mRPI set.

A larger number of switching sequences will result in a closer approximation of the mRPI set. In Figure 4.3b we depict the set \( R_{10}^e(A, \Delta) \) – solid blue contour, against the mRPI set – dashed red contour. As it can be seen the difference between the two sets is almost invisible. To illustrate the point we consider the sequence of sets \( R_k^e(A, \Delta) \) with \( k \in \{0, 1, \ldots, 10\} \) and analyze the differences between consecutive elements. For this purpose we measure the set volume (in \( \mathbb{R}^2 \) it is actually its area) and see how it varies. In Table 4.1 the first row denotes the volume of the current set; the second row denotes the variation of the area between two consecutive sets and the third row enumerates the decrease in volume as a percentage. As it can be seen the values decrease (second to third row) monotonically. Moreover the variation is quite fast which means that after a relatively small number of steps the approximation becomes ‘good enough’.

Secondly, we consider the construction (4.29) and test that indeed there exists a scaling factor which makes it invariant. We take \( I = \{5\} \) and compute \( \Omega_I = S(I \cup I_e) \) and illustrate it in Figure 4.4 – solid blue contour, against the mRPI set – dotted red contour. To obtain the RPI set, we solve an LP optimization to find \( \gamma = 1.6921 \). The set \( \gamma \Omega_I \) and its iteration under dynamics (4.15) are depicted as well, dashed blue and densely dashed red contour, respectively.

35 of 160
### 4.5. Conclusions

In this chapter we have provided an explicit description for the boundary of the mRPI set characterizing dynamics with zonotopic disturbances. Further we have constructed RPI outer-approximations from pairs of extremal hyperplanes and extremal points and discussed their properties. We have studied the RPI property of the candidate sets considered, and, where was the case, computed the necessary scaling factors. We have also analyzed the scaling factors associated to each of them.

Further advances are possible. For example, there are results treating the null-controllable set for complex eigenvalues and the continuous time case \[67\]. These should be relatively easy to adapt for the present mRPI approximations. Another direction to be explored is the analytic computation of the scaling factors as a function of the selected switching sequences.
5. Exact and over-approximated guarantees for corner cutting avoidance

The corner cutting avoidance problem is an important but often overlooked part of motion planning strategies. Obstacle and collision avoidance constraints are usually imposed at the sampling time without regards to the intra-sample behavior of the agent. Hence, it is possible for an agent to “cut the corner” of an obstacle while apparently respecting the constraints.

This chapter presents exact and over-approximated descriptions of the under-shadow (and of its complement, the visible) region generated by an agent against obstacles. I employ a hyperplane arrangement construction to handle multiple obstacles simultaneously, provide piecewise descriptions of the regions of interest and parametrizations of the corner cutting conditions (useful, e.g., in finite horizon optimization problems). Mixed-integer representations are used to describe the regions of interest, leading, in the over-approximated case to binary-only constraints.

Illustrative proofs of concept, comparisons with the state of the art and simulations over a standard multi-obstacle avoidance problem showcase the benefits of the proposed approach.

The chapter is based on a submitted journal article (at the “minor review” stage):

Stoican F., I. Prodan and E. Grotli. “Exact and over-approximated guarantees for corner cutting avoidance in a multi-obstacle environment”, in International Journal of Robust and Nonlinear Control, under review (submitted 2017, currently at the minor review stage).

5.1. Preliminaries

Recent advances in computational resources and the proliferation of (semi-)autonomous vehicles has lent new interest to the topic of obstacle and collision avoidance in motion planning strategies. One of the major issues is that avoidance constraints lead to a non-convex feasible domain. It is worth underlining that such formulations are intrinsic to the problem and cannot be avoided [59, 143, 151].

This chapter concentrates on the “corner cutting” issue: avoidance constraints are checked at each sampling time but the control input is ultimately applied (e.g., via a zero-order hold block) to continuous dynamics. Hence, the intra-sample behavior of the agent (the autonomous vehicle to be steered) cannot be overlooked. While alluring, obstacle enlargement techniques [87] or sampling time reduction are not always appropriate. The former increases the conservatism of the formulation (potentially
leading to infeasibility) and the latter reduces the time available for computing the input.

The computational limitations are particularly troublesome since modeling the associated optimization problem is often done via the Mixed-Integer Programming (MIP) framework [75, 0] which scales badly with problem size and complexity (i.e., number of obstacles).

There are many results in the mathematical community related to the *illumination of convex bodies* topic [101] but the emphasis is descriptive rather than constructive. In fact, to the best of our knowledge, there appear to be few results in the control community which describe corner cutting constraints (in either exact or over-approximated form). We are aware of results from [2, 99, 142] which discuss over-approximated corner cutting constraints and provide constructive details: [99] provides the initial construction; [142] and [2] improve it by reducing the number of necessary constraints and by reducing the number of necessary binary variables, respectively.

While interesting, the existing methods, in our opinion, are lacking in several directions:

i) the extension to a multi-obstacle environment is not straightforward;

ii) the underlying structure of the feasible domain is not fully exploited;

iii) constraints involving only binary variables do not describe precisely the position.

The current work is based and expands previous results of the authors. Preliminary results in [C29] discuss the corner cutting topic (expanded here with additional theoretical results, comparisons with the state of the art and extensive illustrative examples). [C13], [C31] discuss the dual problem of coverage in a multi-obstacle environment.

To describe the multi-obstacle environment (issue i)) we propose to use hyperplane arrangements [119]. That is, the domain is partitioned into disjoint cells uniquely characterized by a collection of signs (a “tuple”) which are further labeled as *interdicted* – the obstacles or as *admissible* – the feasible space [B4].

Next, we provide descriptions for the *shadow region* (and its complement, the *visible region*) spanned by the agent in conjunction with the obstacles. The scaffolding provided by the hyperplane arrangement allows to have piecewise characterizations, i.e., to each admissible cell of the arrangement corresponds a specific shape of the shadow region. Calculating these shapes a priori solves issue ii) and allows to parametrize the corner cutting constraints (necessary for example in a finite horizon optimization problem).

Existing results deal with binary-only constraints (issue iii)). This makes sense in the collision avoidance context since these binary terms are used to constrain the agent’s position w.r.t. the obstacle(s). In here, for a better insight, we start with exact representations which are parametrized after the actual position of the agent. For the particular case of bounded polyhedral sets we provide here the exact forms for the under-shadow and visible regions and, from them, deduce the over-approximated forms (which in particular circumstances reduce to the constraints presented in the literature).

Finally, by exploiting the structure provided by the hyperplane arrangement, we provide mixed-integer descriptions for the regions of interest. That is, using results from [B4], [C29] together with codification methods found in [173] and the references therein we provide mixed-integer representations which describe both the shadow region and its complement in both the exact and over-approximated forms.
5.2. Prerequisites

Let us consider a finite collection of hyperplanes from $\mathbb{R}^n$, $\mathbb{H} = \{\mathcal{H}_i\}_{i \in \mathcal{I}}$ with

$$\mathcal{H}_i = \left\{ x \in \mathbb{R}^n : h_i^\top x = k_i \right\}, \quad i \in \mathcal{I},$$

(5.1)

where $\mathcal{I} \triangleq \{1 \ldots N\}$ and $(h_i, k_i) \in \mathbb{R}^n \times \mathbb{R}$.

Each of these hyperplanes partitions the space into two disjoint regions (which halve the space and hence are called “half-spaces”):

$$\mathcal{H}_i^+ = \left\{ x \in \mathbb{R}^n : h_i^\top x \leq k_i \right\},$$

(5.2a)

$$\mathcal{H}_i^- = \left\{ x \in \mathbb{R}^n : -h_i^\top x \leq -k_i \right\}.$$  

(5.2b)

Furthermore, hyperplanes (5.1) cut the space $\mathbb{R}^n$ into disjoint cells

$$\mathcal{A}(\sigma) = \bigcap_{i \in \mathcal{I}} \mathcal{H}_i^{\sigma(i)},$$

(5.3)

which are feasible intersections of halfspaces (5.2a)–(5.2b) with the signs appropriately taken from the sign tuple $\sigma = (\sigma(1), \ldots, \sigma(N))$. Such a partitioning of the space is called a hyperplane arrangement and is the union of all cells (5.3), that is, $\mathbb{R}^n = \mathcal{A}(\mathbb{H}) = \bigcup_{\sigma \in \Sigma} \mathcal{A}(\sigma)$ where $\Sigma \subset \{-, +\}^N$ denotes the collection of all tuples describing non-empty regions (5.3).

We can then partition the sign tuples into ‘admissible’ ($\sigma^o \in \Sigma^o$) and ‘interdicted’ ($\sigma^* \in \Sigma^*$) where $\Sigma^o \cap \Sigma^* = \emptyset$ and $\Sigma^* \cup \Sigma^o = \Sigma$. The latter subset describes the obstacles whereas the former describes the complement of the obstacle collection:

$$\mathbb{S} = \bigcup_{\sigma^* \in \Sigma^*} \mathcal{A}(\sigma^*), \quad \mathbb{R}^n \setminus \mathbb{S} = \bigcup_{\sigma^o \in \Sigma^o} \mathcal{A}(\sigma^o).$$

(5.4)

A couple of remarks are in order.

Remark 5.1. In here, we start with the hyperplane arrangement and label accordingly the cells. Just as well, we may have started with a collection of obstacles, gathered their support hyperplanes and subsequently generated the associated arrangement. An intermediary approach is to consider over-approximations which reduce or keep constant the number of hyperplanes (e.g., via homothetic transformations of some seed shape [C19]).

Remark 5.2. The framework presented here represents an additional layer of complexity to the original problem and may, at a first glance, be considered superfluous. In defense of this we may mention two arguments. First, the framework allows to consider an arbitrary multi-obstacle environment with a unitary notation and description. Second, the hyperplane arrangement serves as an underlying scaffolding to the various constructions presented hereinafter. E.g., the shadow region discussed in the next section is piecewise defined over the cells of the arrangement.

For further details about hyperplane arrangements and their use in control problems see the monograph [B4].
**Illustrative example**

For the purpose of illustration let us consider the example depicted in Figure 5.1. We consider a union of three obstacles in $\mathbb{R}^2$, $S = S_1 \cup S_2 \cup S_3$ defined by 7 hyperplanes. These partition the space into 29 cells from which 3 describe the obstacles and the rest characterize the feasible space $\mathbb{R}^2 \setminus S$, as in (5.4). More precisely, we identify $\Sigma^* = \{\sigma^{*1}, \sigma^{*2}, \sigma^{*3}\}$ such that $S_1 = \mathcal{A}(\sigma^{*1})$, $S_2 = \mathcal{A}(\sigma^{*2})$ and $S_3 = \mathcal{A}(\sigma^{*3})$ for $\sigma^{*1} = (+ - + - + + +)$, $\sigma^{*2} = (- - + - + +)$ and $\sigma^{*3} = (+ + + - + + +)$. The remaining 26 feasible tuples are gathered in $\Sigma^0$.

![Figure 5.1: Illustration of a collection of obstacles and their associated hyperplane arrangement.](image)

**5.3. Shadow regions descriptions**

Let us consider a point $x \in \mathbb{R}^n \setminus S$. Then, the shadow region $\mathcal{B}(S, x)$ given as in [C13] is the collection of all the points from $\mathbb{R}^n \setminus S$ which are not “visible” from $x$:

$$\mathcal{B}(S, x) = \{y \in \mathbb{R}^n : [x, y] \cap S \neq \emptyset\}. \quad (5.5)$$

This simply states that if the segment $[x, y]$ intersects $S$ it means that the point $y$ is “hidden” by obstacles $S$ and therefore is not “visible” from the viewpoint of $x$.

Remark 5.3. Definition (5.5) does not exclude the points $y \in S$. In other words, the shadow region contains the obstacles as well. We prefer this formulation because it reduces the complexity of the resulting shapes (e.g., if $S$ is a convex body, then $\mathcal{B}(S, x)$ is also convex whereas $\mathcal{B}(S, x) \setminus S$ may not be).

1Note that there many less feasible tuples than the total number of possible combinations [27].
Considering definition (5.4), region (5.5) is rewritten as
\[ \mathcal{B}(S, x) = \mathcal{B}(\Sigma^*, x) = \bigcup_{\sigma^* \in \Sigma^*} \mathcal{B}(\sigma^*, x). \] (5.6)

Its complement, the visible region \( \mathcal{B}(S, x) = \mathbb{R}^n \setminus \mathcal{B}(S, x) \) is defined as
\[ \overline{\mathcal{B}(S, x)} = \overline{\mathcal{B}(\Sigma^*, x)} = \bigcup_{\sigma^* \in \Sigma^*} \overline{\mathcal{B}(\sigma^*, x)} = \bigcap_{\sigma^* \in \Sigma^*} \overline{\mathcal{B}(\sigma^*, x)}. \] (5.7)

In what follows we will provide the exact form of \( \mathcal{B}(\sigma^*, x) \), its over-approximation (and of their complements, in the sense of (5.7)) in a constructive manner which employs the framework from Section 5.2.

5.3.1. Exact description of the under-shadow region

The definitions given in (5.6) and (5.7) are not amenable to practical implementations. Therefore we need to provide an explicit dependence between the structure of (5.5) and parameter \( x \). As a first step, we define \( \mathbb{V} \), the lattice of points originating from the hyperplane arrangement\(^3\) \( \mathbb{H} \):
\[ \mathbb{V} = \{ v \in \mathbb{R}^n : v = \mathcal{H}_{i_1} \cap \cdots \cap \mathcal{H}_{i_{n-1}}, \text{ with } \{i_1, \ldots, i_{n-1}\} \subset \{1, \ldots, N\} \}. \] (5.8)

\( \mathbb{V}|_{\mathcal{A}(\sigma^*)} \), the restriction of \( \mathbb{V} \) to \( \mathcal{A}(\sigma^*) \) and denotes the extreme points of \( \mathcal{A}(\sigma^*) \). To describe constructively (5.5), we provide first the following lemma.

**Lemma 5.1.** For a forbidden tuple \( \sigma^* \in \Sigma^* \) and a point \( x \notin \mathcal{A}(\sigma^*) \), the extreme points of \( \mathcal{A}(\sigma^*) \) tangent from the point of view of \( x \) are given by:
\[ \mathcal{B}(\sigma^*, x) = \mathbb{V}|_{\mathcal{A}(\sigma^*)} \cap \left( \bigcup_{i : x \notin \mathcal{H}_{i}^{\sigma^*}(i)} \mathcal{H}_{i} \right) \cap \left( \bigcup_{j : x \in \mathcal{H}_{j}^{\sigma^*}(j)} \mathcal{H}_{j} \right). \] (5.9)

**Proof.** By definition, the ray spanned from \( x \) and passing through \( v \in \mathcal{B}(\sigma^*, x) \) should intersect \( \mathcal{A}(\sigma^*) \) only on its boundary point \( v \). That is, \( \beta \beta \geq 0 \text{ s.t. } x + \beta(v - x) \in \text{Int}(\mathcal{A}(\sigma^*)). \)

Let us assume that there exist \( i, j \) such that \( v = \mathcal{H}_{i} \cap \mathcal{H}_{j} \) and \( x \in \mathcal{H}_{i}^{\sigma^*}(i), \ x \in \mathcal{H}_{j}^{\sigma^*}(j). \) Then, relations \( \sigma^*(i)h_i^{\top}x \leq \sigma^*(i)k_i, \ \sigma^*(i)h_i^{\top}v \leq \sigma^*(i)k_i \) and \( \sigma^*(j)h_j^{\top}x \leq \sigma^*(j)k_j, \ \sigma^*(j)h_j^{\top}v \leq \sigma^*(j)k_j \) hold. Taking a scalar \( \beta \) which respects \( 0 \beta \leq 1, \) we multiply the first (and third) and the second (and fourth) inequalities with \( 1 - \beta \) and \( \beta \) respectively. This leads to inequalities \( \sigma^*(i)h_i^{\top}[x + \beta(v - x)] \leq \sigma^*(i)k_i \) and \( \sigma^*(j)h_j^{\top}[x + \beta(v - x)] \leq \sigma^*(j)k_j. \) Therefore, \( x + \beta(v - x) \in \mathcal{H}_{i}^{\sigma^*}(i) \cap \mathcal{H}_{j}^{\sigma^*}(j). \) Since \( \text{Int}(\mathcal{H}_{i}^{\sigma^*}(i) \cap \mathcal{H}_{j}^{\sigma^*}(j)) \supset \text{Int}(\mathcal{A}(\sigma^*)) \) there exists a \( \beta \in (0, 1) \) arbitrarily close to 1 for which an interior point of \( \mathcal{A}(\sigma^*) \) lies on the ray defined by \( x \) and \( v \), thus contradicting the initial assumption. The same reasoning can be applied for the case \( x \notin \mathcal{H}_{i}^{\sigma^*}(i), \ x \notin \mathcal{H}_{j}^{\sigma^*}(j) \) thus concluding the proof. \( \square \)

The next corollary shows that (5.9) is piecewise constant on the arrangement \( \mathbb{A}(\mathbb{H}). \)

\(^2\)To shorten the notation, we write \( \mathcal{B}(\mathcal{A}(\sigma^*), x) \) in the compact form \( \mathcal{B}(\sigma^*, x). \)

\(^3\)Note that we assume the arrangement to be in general position (i.e., no slight perturbation of any of its hyperplanes changes the number of cells, or, in other words: no two hyperplanes coincide or are parallel).
Corollary 5.1. For a \( \sigma^0 \in \Sigma^0 \), the set \( \mathcal{E}(\sigma^*, x) \) remains unchanged\(^4\) for any \( x \in \mathcal{A}(\sigma^0) \):

\[
\mathcal{E}(\sigma^*, \sigma^0) = \mathcal{V}_{|\mathcal{A}(\sigma^*)} \cap \left( \bigcup_{i: \sigma^0(i) \neq \sigma^*(i)} \mathcal{H}_i \right) \cap \left( \bigcup_{j: \sigma^0(j) = \sigma^*(j)} \mathcal{H}_j \right). \tag{5.10}
\]

Proof. Term (5.9) is rewritten in form (5.10) if we note that the indices for which \( x \notin \mathcal{H}_i^{\sigma^0(i)} \) and \( x \in \mathcal{H}_i^{\sigma^0(i)} \) remain the same for any point taken from \( \mathcal{A}(\sigma^0) \) and are in fact given by checking whether the regions \( \mathcal{A}(\sigma^*) \) and \( \mathcal{A}(\sigma^0) \) lie on the same (or opposite) sides of the i-th hyperplane. Then, it is straightforward to replace \( x \notin \mathcal{H}_i^{\sigma^0(i)} \) with \( \sigma^0(i) \neq \sigma^*(i) \) and \( x \in \mathcal{H}_j^{\sigma^0(j)} \) with \( \sigma^0(j) = \sigma^*(j) \).

With the help of Lemma 5.1 and Corollary 5.1 we reach the following proposition.

Proposition 5.1. For \( \sigma^0 \in \Sigma^0 \) and an \( x \in \mathcal{A}(\sigma^0) \), the shadow region \( \mathcal{B}(\sigma^*, x) \) is given by:

\[
\mathcal{B}(\sigma^*, x) = \left( \text{Cone} \left( x, \mathcal{E}(\sigma^*, \sigma^0) \right) \right) \cap \left( \bigcap_{i: \sigma^0(i) \neq \sigma^*(i)} H_i^{\sigma^*(i)} \right), \tag{5.11}
\]

where

\[
\text{Cone} \left( x, \mathcal{E}(\sigma^*, \sigma^0) \right) = x + \sum_{i: v_i \notin \mathcal{V}} \beta_i (v_i - x), \quad \text{with} \quad \begin{cases} \beta_i \geq 0, & v_i \in \mathcal{E}(\sigma^*, \sigma^0) \\ \beta_i = 0, & v_i \notin \mathcal{E}(\sigma^*, \sigma^0) \end{cases}. \tag{5.12}
\]

Proof. Looking at definition (5.5) we notice that any point \( y \in \mathcal{B}(\sigma^*, x) \) has to be part of \( \text{Cone} \left( x, \mathcal{A}(\sigma^*) \right) \). Assuming the opposite would mean that the ray \( x + \beta(y - x), \beta \geq 0 \) never intersects the obstacle \( \mathcal{A}(\sigma^*) \), thus contradicting the starting hypothesis that \( y \) is “in the shadow” of the obstacle. Additionally, we need to discard from the cone all the points \( y \) which sit on the same side of a hyperplane with the initial point \( x \) but are on the opposite side wrt the obstacle (the ray spanned from \( x \) and passing through \( y \) will intersect the obstacle but the segment \( [x, y] \) does not intersect it, i.e., the ray has not yet “reached” the obstacle).

These considerations allows us to define \( \mathcal{B}(\sigma^*, x) = \text{Cone} \left( x, \mathcal{A}(\sigma^*) \right) \setminus \bigcup_{i: x \notin \mathcal{H}_i^{\sigma^*(i)}} \mathcal{H}_i^{\sigma^*(i)} \). Using the equivalences \( \bigcup_i \overline{A}_i = \bigcap_i A_i \) and \( A \setminus B = A \cap \overline{B} \) we have that \( \mathcal{B}(\sigma^*, x) = \left( \text{Cone} \left( x, \mathcal{A}(\sigma^*) \right) \right) \cap \left( \bigcap_{i: x \notin \mathcal{H}_i^{\sigma^*(i)}} \mathcal{H}_i^{\sigma^*(i)} \right) \). Noting that the cone spanned from \( x \) and tangent to \( \mathcal{A}(\sigma^*) \) is completely characterized by \( x \) and \( \mathcal{E}(\sigma^*, x) \) as per Lemma 5.1 and applying Corollary 5.1 we reach formulation (5.11) and thus conclude the proof.

A couple of remarks are in order.

Remark 5.4. In (5.12) we consider terms which have no influence in the cone formulation (those with \( \beta_i = 0 \)) since this will simplify the notation later on, when we will consider multiple obstacles simultaneously.

Remark 5.5. A half-space representation of (5.11) is also possible but raises various numerical issues and is not followed here (both representations are nonlinear but the generator form is easier to write

\(^4\)Hereafter, notation (5.10) will supersede (5.9) in order to underline that (5.9) depends only on \( \sigma^0 \) and not on any particular \( x \in \mathcal{A}(\sigma^*) \).
in the subsequent MI representations of Section 5.4).

\section{5.3.2. Over-approximation of the shadow region}

Proposition 5.1 shows that (5.11) has a piecewise structure and thus, for any \( x \) in a given cell \( \mathcal{A}(\sigma^\circ) \), at runtime we need only to input the current value of \( x \) into (5.11). While this reduces the computation burden, the formulation for the shadow area is still difficult due to \( \text{Cone} \ (x, \mathcal{B}(\sigma^\bullet, \sigma^\circ)) \). The solution pursued here is to consider an over-approximation of the shadow region.

As a first step, we provide the following lemma.

\textbf{Lemma 5.2.} For given \( \sigma^\bullet \in \Sigma^\bullet \) and \( \sigma^\circ \in \Sigma^\circ \) relation

\[
\bigcup_{x \in \mathcal{A}(\sigma^\circ)} \text{Cone} \ (x, \mathcal{B}(\sigma^\bullet, \sigma^\circ)) \supset \bigcap_{i : \sigma^\circ(i) \neq \sigma^\bullet(i)} \mathcal{H}^\sigma(i),
\]

(5.13)

holds.

\textbf{Proof.} Let us assume that for an \( x' \in \bigcup_{i : \sigma^\circ(i) \neq \sigma^\bullet(i)} \mathcal{H}^\sigma(i) \) and check whether there exists an \( x \in \mathcal{A}(\sigma^\circ) \) such that (5.13) holds. In addition consider an extreme point \( v \) of \( \mathcal{A}(\sigma^\bullet) \) such that the ray spanned from \( x' \) and passing through \( v \) is tangent to \( \mathcal{A}(\sigma^\bullet) \). We have the inequalities \( \sigma^\bullet(i) h_i^\top x' \leq \sigma^\circ(i) k_i \) and \( -\sigma^\bullet(i) h_i^\top v \leq -\sigma^\circ(i) k_i \). Taking \( \beta > 1 \) and multiplying the first inequality with \( (1 - \beta) \) and the second with \( \beta \) and adding them leads to \( -\sigma^\bullet(i) h_i^\top [x' + \beta(v - x')] \leq -\sigma^\circ(i) k_i \). Repeating for all indices \( i : \sigma^\circ(i) \neq \sigma^\bullet(i) \) means that there exists a point \( x = x' + \beta(v - x') \) which lies in \( \mathcal{A}(\sigma^\circ) \). Therefore, we conclude that \( x' \in \text{Cone} \ (x, \mathcal{B}(\sigma^\bullet, \sigma^\circ)) \) and thus we complete the proof.

\textbf{Lemma 5.2} helps prove the following corollary.

\textbf{Corollary 5.2.} Let there be \( \mathcal{B}(\sigma^\bullet, \sigma^\circ) = \bigcup_{x \in \mathcal{A}(\sigma^\circ)} \mathcal{B}(\sigma^\bullet, x) \), the shadow region associated to a feasible tuple \( \sigma^\circ \in \Sigma^\circ \). Then, this region depends only on \( \sigma^\circ, \sigma^\bullet \) and is described as follows:

\[
\mathcal{B}(\sigma^\bullet, \sigma^\circ) = \bigcap_{\sigma^\circ(i) \neq \sigma^\bullet(i)} H_i^{\sigma^\bullet(i)},
\]

(5.14)

\textbf{Proof.} From the definition of \( \mathcal{B}(\sigma^\bullet, \sigma^\circ) \), the fact that \( \bigcup_i A_i \cap B = (\bigcup_i A_i) \cap B \) and (5.11) follows that \( \mathcal{B}(\sigma^\bullet, \sigma^\circ) = \bigcup_{x \in \mathcal{A}(\sigma^\circ)} \text{Cone} \ (x, \mathcal{B}(\sigma^\bullet, \sigma^\circ)) \cap \bigcap_{\sigma^\circ(i) \neq \sigma^\bullet(i)} H_i^{\sigma^\bullet(i)} \). Using Lemma 5.2 leads to (5.14), thus concluding the proof.

\textbf{Remark 5.6.} By using the over-approximation (5.14) the shadow region not only retains the same structure for any \( x \in \mathcal{A}(\sigma^\circ) \) but actually remains constant. Hence, at run-time it is necessary only to identify the currently active tuple \( \sigma^\circ \) and use the corresponding region (5.14).

From the implementation point of view, we actually need to characterize the “visible” regions\(^5\) (i.e., the complement of \( \mathcal{B}(\sigma^\bullet, x) \) or of \( \mathcal{B}(\sigma^\bullet, \sigma^\circ) \)) in a multi-obstacle environment.

\textbf{Corollary 5.3.} For given \( \sigma^\bullet \in \Sigma^\bullet \) and \( \sigma^\circ \in \Sigma^\circ \), the visible region is given in its

\(^5\)The under-shadow regions are of interest in the dual problem of guaranteed coverage of a multi-obstacle domain [C31]

- ‘how to position a collection of agents such that, overall, no point of the environment remains unobserved’. 

43 of 160
Chapter 5. Exact and over-approximated guarantees for corner cutting avoidance

\(i)\) exact form:
\[
\mathcal{B}(\sigma^*, x) = \text{Cone} \left( x, \mathcal{B}(\sigma^*, \sigma^o) \right) \cup \bigcup_{\sigma^o(i) \neq \sigma^*(i)} \mathcal{H}^{\sigma^o(i)}
\]  
(5.15)

\(ii)\) over-approximated form:
\[
\mathcal{B}(\sigma^*, \sigma^o) = \bigcup_{\sigma^o(i) \neq \sigma^*(i)} \mathcal{H}^{\sigma^o(i)}
\]  
(5.16)

**Proof:** The proof is straightforward and is based on the de Morgan’s laws: \(\overline{A \cap B} = \overline{A} \cup \overline{B}\) and \(\overline{A \cup B} = \overline{A} \cap \overline{B}\) and uses definitions (5.11), (5.14) and the fact that \(\overline{\mathcal{H}^{\sigma^o(i)}} = \mathcal{H}^{\sigma^o(i)}\) whenever \(\sigma^o(i) \neq \sigma^*(i)\).

\(\square\)

**Remark 5.7.** In general, we may consider the shadow area resulting from a set rather than from a point \((x \in X)\). The only difficulty is to check whether the set \(X\) stays in one or more of the regions (5.3). Defining \(\Sigma_X \triangleq \{ \sigma \in \Sigma^o : X \cap \mathcal{A}(\sigma) \neq \emptyset \} \subseteq \Sigma^o\) allows to characterize the shadow region (along the lines of Proposition 5.1 and Corollary 5.2):

\[
\mathcal{B}(\sigma^*, X) = \bigcup_{x \in X \cap \mathcal{A}(\sigma^o), \sigma^o \in \Sigma_X} \mathcal{B}(\sigma^*, x),
\]  
(5.17a)

\[
\mathcal{B}(\sigma^*, \Sigma_X) = \bigcup_{\sigma^o \in \Sigma_X} \mathcal{B}(\sigma^*, \sigma^o).
\]  
(5.17b)

**Illustrative example**

For the example from Section 5.2 we illustrate the exact and over-approximated under-shadow regions (just for obstacle \(S_1 = \mathcal{A}(\sigma^{*, 1})\) in order to keep the figure uncluttered).

As a first step, we consider the point \(x_1 \in \mathcal{A}(\sigma)\), with \(\sigma = (- - - - + + +)\), characterizing the gray-filled cell in Figure 5.2 (b). Checking the signs for tuples \(\sigma^{*, 1}\) and \(\sigma\) we note that \(x_1\) shares the same half-spaces with \(S_1\) for indices 2, 4, 5, 6 and 7, while for indices 1 and 3 it sits in the opposite half-spaces. Using this information in (5.9), or alternatively in (5.10), we note that the set of tangent points of \(S_1\) from the viewpoint of \(x_1\) is given by \(\mathcal{B}(\sigma^{*, 1}, x_1) = \mathcal{B}(\sigma^{*, 1}, \sigma) = \{ \mathcal{H}_2 \cap \mathcal{H}_3, \mathcal{H}_1 \cap \mathcal{H}_3 \}\).

Having \(\mathcal{B}(\sigma^{*, 1}, \sigma)\) allows to illustrate in Figure 5.2 (a), using (5.11) from Proposition 5.1, the exact under-shadow region \(\mathcal{B}(\sigma^{*, 1}, x_1) = \text{Cone}(x_1, \mathcal{B}(\sigma^{*, 1}, \sigma)) \cap (\mathcal{H}_1^+ \cap \mathcal{H}_3^+)\). Note that the half-spaces which separate between the position \(x_1\) and the obstacle \(S_1\) are those with indices 1 and 3, i.e., the indices for which the sign tuples \(\sigma^{*, 1}\) and \(\sigma\) differ.

To highlight that the structure of the under-shadow region (5.11) remains constant while the agent’s position is taken from the same feasible cell (see Proposition 5.1) we take \(x_2 \in \mathcal{A}(\sigma)\) and note in Figure 5.2 (a) that, indeed, the tangent points to obstacle \(S_1\) remain those of \(\mathcal{B}(\sigma^{*, 1}, \sigma)\) and the separating half-spaces are, again, those with indices 1 and 3: \(\mathcal{B}(\sigma^{*, 1}, x_2) = \text{Cone}(x_2, \mathcal{B}(\sigma^{*, 1}, \sigma)) \cap (\mathcal{H}_1^+ \cap \mathcal{H}_3^+)\).

Lastly, we depict in Figure 5.2 (b) the over-approximated under-shadow region \(\mathcal{B}(\sigma^{*, 1}, \sigma) = \mathcal{H}_1^+ \cap \mathcal{H}_3^+\) as in (5.14) from Corollary 5.2. It can be seen that \(\mathcal{B}(\sigma^{*, 1}, \sigma)\) contains any region \(\mathcal{B}(\sigma^{*, 1}, x)\) for \(x \in \mathcal{A}(\sigma)\), and in particular for \(x \in \{x_1, x_2\}\).
Figure 5.2.: Illustration of exact and over-approximated under-shadow regions.
5.4. Mixed integer representations

In Section 5.3 we gave various formulations for shadow regions observed from the point of view of an agent and with multiple obstacles. Regardless of the particular construction, the issue is that the resulting feasible region is non-convex (and in the case of multiple obstacles, not even connected). Henceforth, we use mixed integer formulations to express the shadow (and their complements) regions in a manageable form. That is, we add binary variables to the original problem such that the resulting relations describe a pseudo-(non)linear formulation which can be solved with specialized solvers (e.g., CPLEX [34], GUROBI [118] or SCIP [98]).

As a first step, let us consider the Hamming distance:

\[ f; g = \sum_{i : \sigma_1(i) \neq \sigma_2(i)} 1. \]  

(5.18)

Abusing the notation and equating ‘-’ with ‘0’ and ‘+’ with ‘1’ allows to rewrite (5.18) as:

\[ \delta(\sigma_1, \sigma_2) = |\sigma_1 - \sigma_2| = \sum_{i} |\sigma_1(i) - \sigma_2(i)| = \sum_{i} \sigma_1(i) + [1 - 2\sigma_1(i)]\sigma_2(i). \]  

(5.19)

Remark 5.8. As long as one of the terms is known (either \( \sigma_1 \) or \( \sigma_2 \)), mapping (5.19) is linear and can be integrated easily in a MI formulation (since the product \( \sigma_1(i)\sigma_2(i) \) appearing in (5.19) has only one variable).

As stated in Remark 5.8 whenever both tuples \( \sigma_1, \sigma_2 \) are unknown, mapping (5.18) is nonlinear, which in turn leads to a mixed integer nonlinear formulation. This issue can be alleviated through the following lemma (the nonlinear (in)equality is replaced by an enumeration of linear inequalities which is equivalent with the former).

Lemma 5.3. For \( \sigma_1, \sigma_2 \in \Sigma^o \) the following equivalencies are true:

\[ \left( |\sigma_1 - \sigma_2| = 0 \right) \iff \left( |\sigma^o - \sigma_2| \leq N \cdot |\sigma^o - \sigma_1|, \forall \sigma^o \in \Sigma^o \right), \]  

(5.20a)

\[ \left( |\sigma_1 - \sigma_2| > 0 \right) \iff \left( |\sigma^o - \sigma_2| > |\sigma^o - \sigma_1|, \forall \sigma^o \in \Sigma^o \right). \]  

(5.20b)

Proof. We consider first the equivalence (5.20a). The ‘\( \rightarrow \)’ implication is straightforward, consequently we concentrate on the ‘\( \leftarrow \)’ implication. The right term of the equivalence (5.20a) consists of an enumeration of inequalities, one per each feasible sign tuple \( \sigma^o \in \Sigma^o \).

Recalling that \( |\sigma^o - \sigma_1| \) takes values from \( \{0, 1, \ldots, N\} \) follows that the right hand side is zero iff \( \sigma^o = \sigma_1 \) which forces \( |\sigma^o - \sigma_2| = 0 \) and leads to \( \sigma_1 = \sigma_2 \), the left-part of the equivalence (5.20a). Whenever \( \sigma^o \neq \sigma_1 \) the right-hand term is large enough to make the inequality redundant (i.e., regardless of the value of \( |\sigma^o - \sigma_2| \), the inequality holds). A similar reasoning is applied for (5.20b), thus concluding the proof.

Remark 5.9. Note that (5.20a)–(5.20b) are not ‘big-M’ formulations (see also Remark 5.10). We are certain that the left-hand sides of the equations cannot be larger than ‘N’ – the number of hyperplanes and hence we do not need to relax the right-hand term to a value larger than that.

\[^6\text{Since } |\cdot| \text{ more intuitively denotes the notion of distance, we will henceforth use it instead of notation } \delta(\cdot, \cdot).\]
Using constructions similar to the ones in [173] and the references therein we provide in what follows MI formulations for the exact and over-approximated shadow regions (and their complements, the visible regions).

### 5.4.1. Mixed integer representations for the exact shadow and visible regions

The following proposition gives a MI representation of the exact shadow region from Proposition 5.1.

**Proposition 5.2.** Let there be a point \( x \in \mathcal{A}(\sigma) \) and the collection of obstacles characterized by \( \Sigma^\bullet \). Then inclusion \( x^+ \in \mathcal{B}(\Sigma^\bullet, x) = \bigcup_{\sigma^\bullet j \in \Sigma^\bullet} \mathcal{B}(\sigma^\bullet j, x) \) holds iff

\[
    x^+ = x + \sum_i \beta(i) (v_i - x), \quad \beta(i) \geq 0,
\]

\[
    \sum_{i : v_i \notin \mathcal{A}(\sigma^\bullet j, \sigma^\circ)} \beta(i) \leq M \left( |\sigma^\circ^\circ - \sigma| + \alpha^j \right), \quad \forall \sigma^\circ^\circ \in \Sigma^\circ, \forall \sigma^\bullet j \in \Sigma^\bullet
\]

\[
    \sigma^\bullet j(\ell) k^\top \ell x^+ \leq \sigma^\bullet j(\ell) k^\top \ell + M \left( 1 - |\sigma^\bullet j(\ell) - \sigma(\ell)| + \alpha^j \right), \forall \ell = 1 \ldots N, \forall \sigma^\bullet j \in \Sigma^\bullet
\]

\[
    \sum_j (1 - \alpha^j) \geq 1,
\]


is feasible. Auxiliary variables \( \beta(i) \in \mathbb{R}, \alpha^j \in \{0,1\} \) are taken appropriately.

**Proof.** Inclusion \( x^+ \in \mathcal{B}(\Sigma^\bullet, x) \) means that there should be at least an active obstacle \( \mathcal{A}(\sigma^\bullet j) \) such that \( x^+ \in \mathcal{B}(\sigma^\bullet j, x) \). Proposition 5.1 shows that this is equivalent with having \( x^+ \in \text{Cone} \left( x, \mathcal{B}(\sigma^\bullet j, \sigma^\circ) \right) \) and \( x^+ \in \bigcap_{\sigma^\bullet j(i) \neq \sigma^\bullet j(i)} H^{\sigma^\bullet j(i)}_i \). We proceed to show that these inclusions are equivalent with having \( (5.21) \) feasible.

Auxiliary binary variable \( \alpha^j \) marks whether \( x^+ \) lies in the shadow region determined by \( \sigma^\bullet j \), or not (\( \alpha^j = 0 \) for the former and \( \alpha^j = 1 \) for the later). Having \( \alpha^j = 1 \) means that the right-hand terms from \( (5.21b) \) and \( (5.21c) \) are always greater than the left-hand terms (see also Remark 5.10). Therefore the inequalities \( (5.21b) \) and \( (5.21c) \) become redundant. \( (5.21d) \) ensures that at least one \( \sigma^\bullet j \) is active (i.e., that there exists at least one index \( j \) s.t. \( (5.21b) \) and \( (5.21c) \) are not redundant).

Assuming, without loss of generality, that we are in the case \( \alpha^j = 0 \) we analyze the rest of the terms. \( (5.21a) \) describes the cone spanned from \( x \) and passing through extreme points \( v_i \). With the help of Lemma 5.3, \( (5.21b) \) ensures that only the extreme points which are active for the pair \( (\sigma, \sigma^\bullet j) \) participate in the cone construction (since the terms \( \beta(i) \) are positive, if their sum is zero then each term is zero). \( (5.21c) \) adds the half-spaces which separate the obstacle \( \mathcal{A}(\sigma^\bullet j) \) from the point \( x \).

A couple of remarks are in order.

**Remark 5.10.** Relations \( (5.21) \) make use of the “big M” representation. That is, we consider (usually in the right hand side of the equation) a combination of binary variables multiplied by a large value (i.e., ‘M’). This means that whenever the binary part is \( \geq 1 \) the right hand is for practical purposes infinite thus making the associated inequality redundant. Conversely, when the binary part is zero it means that the inequality remains active and that it constrains the feasible space.

**Remark 5.11.** Proposition 5.2 assumes that the value of \( x \) and the value of its sign tuple \( \sigma \) are unknown. This is the reason for which terms \( |\sigma^\circ^\circ - \sigma| \) appear in \( (5.21b) \). Similar with the reasoning of Lemma 5.3, enumerating all feasible tuples \( \sigma^\circ^\circ \in \Sigma^\circ \) means that, of all the inequalities, just the ones
for which \( \sigma^o = \sigma \) are non-redundant and contribute to the description. Similarly, in (5.21c), terms \(|\sigma^o \cdot j(\ell) - \sigma(\ell)|\) make sure that only the appropriate half-spaces are taken into consideration.

Conversely, if \( x, \sigma \) are known, the terms mentioned earlier will disappear from formulation (5.21), leading to a simpler form (with both less inequalities and less variables).

Remark 5.11 will prove useful later on when we will consider a model predictive control scheme: the associated optimization problem assumes a sequence of predicted values which are not known a priori but are decision variables in the optimization problem.

In Proposition 5.2 we provided a mixed integer description of the shadow region. Next, we provide its counterpart, the mixed integer description of the visible region.

**Proposition 5.3.** Let there be a point \( x \in \mathcal{A}(\sigma) \) and the collection of obstacles characterized by \( \Sigma^* \). Then exclusion \( x^+ \notin \mathcal{B}(\Sigma^*, x) = \bigcup_{\sigma^* \in \Sigma^*} \mathcal{B}(\sigma^* \cdot j, x) \) holds iff

\[
x^+ = x + \sum_{i : v_{ji} \in \mathcal{V}_{ji}^{(\sigma^* \cdot j)}} \beta^j(i)(v_{ji} - x), \tag{5.22a}
\]

\[
|\beta^j(i)| \leq M \left(|\sigma^o - \sigma| + 1 - \alpha^j\right), i : v_{ji} \notin \mathcal{B}(\sigma^* \cdot j, \sigma^o), \tag{5.22b}
\]

\[
\beta^j(i) \leq M \left(1 - \lambda^j(i) + |\sigma^o - \sigma| + 1 - \alpha^j\right), i : v_{ji} \in \mathcal{B}(\sigma^* \cdot j, \sigma^o), \tag{5.22c}
\]

\[
\sum_{i : v_{ji} \notin \mathcal{B}(\sigma^* \cdot j, \sigma^o)} \lambda^j(i) \geq 1, \forall \sigma^o \in \Sigma^o, \forall \sigma^* \cdot j \in \Sigma^*, \tag{5.22d}
\]

\[
-\sigma^* \cdot j(\ell) h_i^j x^+ \leq -\sigma^* \cdot j(\ell) h_i^j + M \left(|\sigma^* \cdot j(\ell) - \sigma(\ell)| + \rho^j(\ell) + \alpha^j\right), \ell = 1 \ldots N, \tag{5.22e}
\]

\[
\sum_{\ell} \rho^j(\ell) \leq N - 1, \forall \sigma^* \cdot j \in \Sigma^*, \tag{5.22f}
\]

is feasible. Auxiliary variables \( \beta^j(i) \in \mathbb{R}, \alpha^j, \lambda^j(i), \rho^j(\ell) \in \{0, 1\} \) are taken appropriately\(^7\).

**Proof.** Exclusion \( x^+ \notin \mathcal{B}(\Sigma^*, x) \) is equivalent with \( x^+ \in \overline{\mathcal{B}(\Sigma^*, x)} = \bigcap_{\sigma^* \cdot j \in \Sigma^*} \mathcal{B}(\sigma^* \cdot j, x) \). In other words, \( x^+ \) has to stay in the visible region described by each of the obstacles. Looking at the \( j \)-th obstacle, \( \mathcal{A}(\sigma^* \cdot j) \), and recalling Corollary 5.3 i) means that \( x^+ \in \overline{\text{Cone}(x, \mathcal{B}(\sigma^* \cdot j, \sigma^o))} \) or \( x^+ \in \bigcup_{\sigma(i) \neq \sigma^* \cdot j(i)} H^i(\sigma(i)). \) We proceed to show that these relations are equivalent with having (5.22) feasible.

Relations (5.22a)–(5.22d) describe the exterior of \( \text{Cone}(x, \mathcal{B}(\sigma^* \cdot j, \sigma^o)) \): (5.22a) describes the cone spanning from \( x \) and passing through the vertices of the currently active obstacle; the selection of the active coefficients is done in (5.22b) and (5.22c)–(5.22d) force that at least one of these coefficients is negative (thus ensuring that \( x^+ \) lies outside of the cone). The term \(|\sigma^o - \sigma|\) appearing in both (5.22b)–(5.22c) means that whenever \( \sigma \neq \sigma^o \) the coefficients \( \beta^j(i) \) are not constrained (see also Remark 5.11).

Relations (5.22e)–(5.22f) describe inclusion \( x^+ \in \bigcup_{\sigma(i) \neq \sigma^* \cdot j(i)} H^i(\sigma(i)). \) Staying inside a union of half-spaces means that at least one of them should not be redundant. This is done through the addition of terms \( \rho^j(\ell) \) in (5.22e) and by constraining them in (5.22f) such that at least one of them is zero. The terms \(|\sigma^* \cdot j(\ell) - \sigma(\ell)|\) appearing in (5.22e) select the half-spaces which contain \( x \) but not the obstacle (in order to be in the visible region, \( x^+ \) has to lie in at least one of them).

Variables \( \alpha^j \) permit to switch between the cone and half-space inclusions. \( \square \)

---

\(^7\)Notation \( v_{ji} \) denotes the indexing of vertices within the current obstacle (i.e., \( i \) takes values from 1 to the number of vertices in \( \mathcal{V}_{ji}^{(\sigma^* \cdot j)} \)). This underlines that the auxiliary terms \( \beta^j, \lambda^j \) have as many elements as there are extreme vertices in the \( j \)-th obstacle. Notation \( v_{ji} \) denotes indexing in the collection \( \mathcal{V} \).
5.4.2. Mixed integer representations for the over-approximated shadow regions

The exact formulations (5.21a)-(5.21c) and (5.22a)-(5.22f) lead to complex representations. This is due to the presence of term \( \text{Cone}(\sigma^*, x) \). If on the other hand we use the over-approximations proposed in Corollary 5.2 and Corollary 5.3 ii), we greatly simplify the representations.

**Proposition 5.4.** Let there be a point \( x \in A(\sigma) \) and the collection of obstacles characterized by \( \Sigma^* \). Then the future position \( x^+ \in A(\sigma^+) \) is constrained as follows:

(i) for \( x^+ \in B(\Sigma^*, \sigma) \):

\[
\sum_i |\sigma^*j(i) - \sigma(i)| \cdot |\sigma^*j(i) - \sigma^+(i)| \leq N(1 - \alpha^j), \quad \forall \sigma^*j \in \Sigma^*,
\]

\[
\sum_j \alpha^j \geq 1,
\]

(ii) for \( x^+ \notin B(\Sigma^*, \sigma) \):

\[
\sum_i |\sigma^*j(i) - \sigma(i)| \cdot |\sigma^*j(i) - \sigma^+(i)| > 0, \quad \forall \sigma^*j \in \Sigma^*.
\]

**Proof.** For both cases it is a matter of ignoring the constraints related to the term \( \text{Cone}(\sigma^*, x) \) which means that we remain with (5.21c) and (5.22e)-(5.22f), respectively. Further, we interpret these constraints in terms of three sign tuples: \( \sigma^* \) characterizes the shadow/visible region and is constrained by the current position (described by \( \sigma \)) and the obstacles (described by \( \sigma^*j \in \Sigma^* \)).

For the \( j \)-th obstacle, characterized by \( \sigma^*j \), let us consider the constraint

\[
|\sigma^*j(i) - \sigma(i)| \cdot |\sigma^*j(i) - \sigma^+(i)| = 0.
\]

Whenever \( \sigma^*j(i) \) and \( \sigma(i) \) share the same sign (i.e., \( |\sigma^*j(i) - \sigma(i)| = 0 \)) the value of \( \sigma^+(i) \) is not constrained but whenever \( \sigma^*j(i) \) and \( \sigma(i) \) have opposite signs (i.e., \( |\sigma^*j(i) - \sigma(i)| = 1 \)) we have that \( \sigma^+(i) \) is constrained to have the same sign as \( \sigma^*j(i) \) (thus making \( |\sigma^*j(i) - \sigma^+(i)| = 0 \)). In other words, we have that \( \sigma^*j(i) \neq \sigma(i) \) leads to \( \sigma^*j(i) = \sigma^+(i) \).

For case i), from (5.14) we have that \( \sum_i |\sigma^*j(i) - \sigma(i)| \cdot |\sigma^*j(i) - \sigma^+(i)| = 0 \) is a necessary and sufficient condition for \( x^+ \in B(\sigma^*j, \sigma) \). The addition and constraining of auxiliary binary variables \( \alpha^j \) in (5.23) leads to \( x^+ \in B(\Sigma^*, \sigma) \).

Case ii) is treated similarly: \( x^+ \notin B(\sigma^*j, \sigma) \) means that there should exist at least an index \( i \) such that (5.25) does not hold. Since the sum of positive terms is not zero iff at least a term is not zero, we have that \( \sum_i |\sigma^*j(i) - \sigma(i)| \cdot |\sigma^*j(i) - \sigma^+(i)| > 0 \) is a necessary and sufficient condition for \( x^+ \notin B(\sigma^*j, \sigma) \). Repeating for all \( \sigma^*j \in \Sigma^* \) leads to (5.24), thus concluding the proof.

Recalling Remark 5.8, we note that (5.23)-(5.24) are linear only if \( \sigma \) is known. If \( \sigma \) is itself an unknown parameter we propose the following corollary.

**Corollary 5.4.** Let there be a point \( x \in A(\sigma) \) and the collection of obstacles characterized by \( \Sigma^* \). Then the future position \( x^+ \in A(\sigma^+) \) is constrained as follows:
Chapter 5. Exact and over-approximated guarantees for corner cutting avoidance

(i) for \( x^+ \in \mathcal{B}(\Sigma^*, \sigma) \):
\[
\sum_{i} |\sigma^*(i) - \sigma^0(i)| \cdot |\sigma^*(i) - \sigma^+(i)| \leq N \left( |\sigma^0 - \sigma| + 1 - \alpha^j \right), \quad \forall \sigma^* \in \Sigma^*;
\]
\[
\sum_{j} \alpha^j \geq 1,
\] (5.26)

(ii) for \( x^+ \notin \mathcal{B}(\Sigma^*, \sigma) \):
\[
\sum_{i} |\sigma^*(i) - \sigma^0(i)| \cdot |\sigma^*(i) - \sigma^+(i)| > -|\sigma^0 - \sigma| \quad \forall \sigma^* \in \Sigma^*;
\] (5.27)

for all \( \sigma^0 \in \Sigma^0 \).

Proof. The proof is a direct application of Lemma 5.3 to the relations of Proposition 5.4: out of the enumeration of inequalities from (5.26) and (5.27) the only ones which are non-redundant are those for which \( \sigma^0 = \sigma \) and these are exactly (5.23)–(5.24).

\[ \square \]

Remark 5.12. In Corollary 5.4 and earlier we assume \( x \in \mathcal{A}(\sigma) \). Such an inclusion implies a link between the variable \( x \) and its sign tuple \( \sigma \). Relations

\[ h_i^\top x \leq k_i + M [1 - \sigma(i)], \]
\[ -h_i^\top x \leq -k_i + M \sigma(i), \] (5.28a)

force \( x \) to sit on one side or the other of the \( i \)-th hyperplane, as selected by \( \sigma(i) \). Similar relations can be written for \( x^+ \in \mathcal{A}(\sigma^+) \).

\[ \spadesuit \]

5.4.3. Comparison with the state of the art

In what follows we compare with papers [2, 99, 142] which, to a large extent, have motivated the current work. These papers discuss the corner cutting issue and provide constructive details for the constraints which ensure its validation. Note that they only consider what we call here the ‘over-approximated case’ and therefore the comparison will limit to the results from Section 5.4.2 only.

The principal ideas of [99], with our notation\(^8\), are summarized as follows:

i) there exists at least one half-space containing the obstacle which does not contain the current and successor positions of the agent:
\[
\exists i \text{ s.t. } \sigma(i) = \sigma^+(i) = 0;
\] (5.29)

ii) for each pair of consecutive tuples there exists a collection of constraints such that at least one of them is non-redundant and implies (5.29):
\[
\sum_{i} [1 - \sigma^0(i)] \sigma^+(i) < N + \sum_{i} [1 - 2\sigma^0(i)] \sigma(i), \quad \forall \sigma^0 \in \{0, 1\}^N.
\] (5.30)

\(^8\)As before, we equate ‘-’ with ‘0’ and ‘+’ with ‘1’. In addition, binary variables \( \sigma, \sigma^+, 1 - \sigma^0 \) stand for \( d[i-1], d[i], q \) used throughout [99].
Both (5.29) and (5.30) assume a single obstacle ($\Sigma^* = \{\sigma^{\cdot,1}\}$) where, by convention, $\sigma^{\cdot,1}(i) = 1, \forall i$. This implies that $\sum_i \sigma(i) \leq N - 1$, $\sum_i \sigma^+(i) \leq N - 1$.

For item i), introducing $\sigma^{\cdot,1}$ defined above in (5.24) of Proposition 5.4 ii) leads to $\sum_i |1 - \sigma(i)| \cdot |1 - \sigma^+(i)| > 0$ and, consequently, to the existence of an index $i$ s.t. $|1 - \sigma(i)| \cdot |1 - \sigma^+(i)| = 1$ which directly implies (5.29).

For item ii), introducing $\sigma^{\cdot,1}$ in (5.27) of Corollary 5.4 ii) leads to $\sum_i |1 - \sigma^o(i)| \cdot |1 - \sigma^+(i)| > -\sum_i |\sigma^o(i) - \sigma(i)|, \forall \sigma^o \in \Sigma^o$. Using (5.19), the previous inequality becomes $N - \sum_i \sigma^+(i) + \sum_i \sigma^o(i) \sigma^+(i) > \sum_i [2\sigma^o(i) - 1] \sigma(i)$. A rearranging of the terms leads immediately to (5.30). While (5.30) has many more inequalities than (5.27), since in both cases the only non-redundant one corresponds to $\sigma = \sigma^o$ and $\sigma \in \Sigma^o$ it follows that (5.30) and (5.27) are equivalent for a single obstacle.

[142] improves on [99] by reducing the number of constraints (5.30) from $2^N$ to a more manageable $N$. This is done by forcing two consecutive positions $x, x^+$ to respect the same constraint, with our notation: $-h_i^1 x \leq -k_i + M\sigma^+(i)$ and $-h_i^2 x^+ \leq -k_i + M\sigma^+(i)$ for all $i = 1 \ldots N$. This implies that there exists at least one index $i$ s.t. both $x$ and $x^+$ lie on the same side of the hyperplane (and thus on the opposite side from the obstacle), similar with (5.29).

In both papers the results are discussed over the single obstacle case and it is not obvious how they are expanded to a multi-obstacle environment (even if examples over such cases appear in the papers). We assume that the equations written for the single obstacle case are repeated for each new obstacle. This is both cumbersome and increases the redundancy of the problem (in both the number of constraints and variables).

The issues highlighted above are to a great extent alleviated by the use of the hyperplane arrangement framework proposed in this chapter. Foremost, a multi-obstacle environment can be treated coherently with a single set of constraints and the problem itself is significantly more compact: in number of decision variables if some of the hyperplanes are shared between obstacles and in number of constraints by exploiting the fact that not all possible sign combinations lead to non-empty cells. In fact, according to Buck’s formula [27], a hyperplane arrangement has at most $\binom{0}{N} + \cdots + \binom{4}{N}$ cells, much less than the $2^N$ of possible sign combinations.

Lastly, [2] proposes a logarithmic scheme to reduce the number of binary variables involved in the selection of the active hyperplanes, as a direct improvement to [142]. We have not pursued this approach here but we point out to [B4] for a comparison of encoding methods in the hyperplane arrangement framework.

**Illustrative example**

We consider the example from Section 5.2 and apply it to the results of this section (this time for the multi-obstacle case, with $S_1, S_2$ and $S_3$ considered together). Recall that the same and opposite half-space signs between $\sigma$ and $\sigma^{\cdot,1}$ are $\{2, 4, 5, 6, 7\}$ and $\{1, 3\}$, respectively. A similar reasoning holds for $\sigma^{\cdot,2}$ ($\{1, 2, 4, 6, 7\}$ and $\{3, 5\}$) and $\sigma^{\cdot,3}$ ($\{6, 7\}$ and $\{1, 2, 3, 4, 5\}$). In addition, with the notation from Figure 5.3, we have that $\mathcal{B}(\sigma^{\cdot,1}, \sigma) = \{v_1, v_2\}, \mathcal{B}(\sigma^{\cdot,2}, \sigma) = \{v_3, v_4\}$ and $\mathcal{B}(\sigma^{\cdot,3}, \sigma) = \{v_5, v_6\}$. With these elements we construct both the exact and over-approximated shadow regions (5.11) and (5.14), as depicted in Figure 5.3 and Figure 5.4, respectively (for $x_1 \in \mathcal{A}(\sigma)$).

As illustrated in Figure 5.3 the exact under-shadow region $\mathcal{B}(\Sigma^*, x_1)$ is rather convoluted: in this
particular case, the region is composed from two disjoint pieces, one of them non-convex. Using Proposition 5.2 we provide the mixed-integer description of this region:

\[ x^+ = x + \sum_{i=1}^{10} \beta(i)(v_i - x), \quad \beta(i) \geq 0, \quad (5.31a) \]

\[ \beta(3) + \beta(4) + \beta(5) + \beta(6) + \beta(7) + \beta(8) + \beta(9) + \beta(10) \leq M\alpha^1, \quad (5.31b) \]

\[ \beta(1) + \beta(2) + \beta(5) + \beta(6) + \beta(7) + \beta(8) + \beta(9) + \beta(10) \leq M\alpha^2, \quad (5.31c) \]

\[ \beta(1) + \beta(2) + \beta(3) + \beta(4) + \beta(7) + \beta(8) + \beta(9) + \beta(10) \leq M\alpha^3, \quad (5.31d) \]

\[ h_1^T x^+ \leq k_1 + M\alpha^1, \quad h_3^T x^+ \leq k_3 + M\alpha^1, \quad (5.31e) \]

\[ h_5^T x^+ \leq k_3 + M\alpha^2, \quad -h_5^T x^+ \leq -k_5 + M\alpha^2, \quad (5.31f) \]

\[ h_1^T x^+ \leq k_1 + M\alpha^3, \quad h_2^T x^+ \leq k_2 + M\alpha^3, \quad (5.31g) \]

\[ h_3^T x^+ \leq k_3 + M\alpha^3, \quad h_4^T x^+ \leq k_4 + M\alpha^3, \quad (5.31h) \]

(5.31a) corresponds to (5.21a) and constrains the future position of the agent inside the cone determined by the current position. (5.31b)–(5.31d) correspond to (5.21b) and select which of the extreme points are active (e.g., whenever \( \alpha^1 = 0 \) we have that \( \beta(3) + \beta(4) + \beta(5) + \beta(6) + \beta(7) + \beta(8) + \beta(9) + \beta(10) = 0 \) which implies that only \( \beta(1), \beta(2) \) can be non-zero and thus, (5.31a) reduces to \( x^+ = x + \beta(1)(v_1 - x) + \beta(2)(v_2 - x) \)). (5.31e)–(5.31g) correspond to (5.21c) and force the future state \( x^+ \) to stay in the half-spaces which separate between the active obstacle and the current position (e.g., whenever \( \alpha^1 = 0 \), the only active inequalities remain (5.31e) and these reduce to \( h_1^T x^+ \leq k_1 \),
with (5.32a)–(5.32c) at equals that out of the three obstacles, at least one is active.

Note that for the ease of representation we simplified (5.21b) in the sense that we wrote only the inequalities corresponding to the case \( \sigma^o = \sigma \) (otherwise, we should have repeated the group of inequalities (5.31b)–(5.31d) for each \( \sigma^o \in \Sigma^o \)). Similarly, for (5.21c) we wrote just these inequalities where the term \(|\sigma^o \cdot \ell - \sigma(\ell)|\) reduced to zero (such that, e.g., for \( j = 1 \) we wrote just two inequalities in (5.31e), instead of seven – the number of hyperplanes).

Proposition 5.3 provides the mixed-integer description of the visible region \( \mathcal{B}(\Sigma^*, x_1) \):

\[
x^+ = x + \sum_{i=1}^{4} \beta^1(i)(v_j - x), \quad \beta^1(i) \geq 0, \quad j_i \in \{1, 2, 7, 8\},
\]

\[
x^+ = x + \sum_{i=1}^{3} \beta^2(i)(v_j - x), \quad \beta^2(i) \geq 0, \quad j_i \in \{3, 4, 9\},
\]

\[
x^+ = x + \sum_{i=1}^{3} \beta^3(i)(v_j - x), \quad \beta^3(i) \geq 0, \quad j_i \in \{5, 6, 10\},
\]

\[|\beta^1(3)| \leq M (1 - \alpha^1), \quad |\beta^1(4)| \leq M (1 - \alpha^1), \]

\[|\beta^2(3)| \leq M (1 - \alpha^2), \]

\[|\beta^3(3)| \leq M (1 - \alpha^3), \]

\[\beta^1(i) \leq M (1 - \lambda^1(i) + 1 - \alpha^1), \quad \forall i \in \{1, 2\}, \]

\[\beta^2(i) \leq M (1 - \lambda^2(i) + 1 - \alpha^2), \quad \forall i \in \{1, 2\}, \]

\[\beta^3(i) \leq M (1 - \lambda^3(i) + 1 - \alpha^3), \quad \forall i \in \{1, 2\}, \]

\[\lambda^1(1) + \lambda^1(2) \geq 1, \]

\[\lambda^2(1) + \lambda^2(2) \geq 1, \]

\[\lambda^3(1) + \lambda^3(2) \geq 1, \]

\[-h_1^\top x^+ \leq -k_1 + M \left( \rho^1(1) + \alpha^1 \right), \quad -h_2^\top x^+ \leq -k_3 + M \left( \rho^1(3) + \alpha^1 \right), \]

\[-h_3^\top x^+ \leq -k_3 + M \left( \rho^2(3) + \alpha^2 \right), \quad h_1^\top x^+ \leq k_5 + M \left( \rho^2(5) + \alpha^2 \right), \]

\[-h_1^\top x^+ \leq -k_1 + M \left( \rho^3(1) + \alpha^3 \right), \quad -h_2^\top x^+ \leq -k_2 + M \left( \rho^3(2) + \alpha^3 \right), \]

\[-h_3^\top x^+ \leq -k_3 + M \left( \rho^3(3) + \alpha^3 \right), \quad -h_4^\top x^+ \leq -k_4 + M \left( \rho^3(4) + \alpha^3 \right), \]

\[h_5^\top x^+ \leq k_5 + M \left( \rho^3(5) + \alpha^3 \right), \]

\[1 \geq \rho^1(1) + \rho^1(3), \]

\[1 \geq \rho^2(3) + \rho^2(5), \]

\[4 \geq \rho^3(1) + \rho^3(2) + \rho^3(3) + \rho^3(4) + \rho^3(5). \]

(5.32a)–(5.32c) correspond to (5.22a) and constrain the future position of the agent outside the cones determined by the current position (w.r.t. the obstacles). (5.32d)–(5.32f) correspond to (5.22b) and select which of the extreme points are active in the cone representation (e.g., whenever \( \alpha^1 = 0 \) we have that \( |\beta^1(3)| = 0, |\beta^1(4)| = 0 \) which implies that only \( \beta^1(1), \beta^1(2) \) can be non-zero and thus, (5.32a) reduces to \( x^+ = x + \beta^1(1)(v_1 - x) + \beta^1(2)(v_2 - x) \)). (5.32g)–(5.32i) correspond to (5.22c) and together with (5.32j)–(5.32l) which correspond to (5.22d) ensure that in each of equalities (5.32a)–(5.32c) at
least one coefficient is negative (such that the future position $x^+$ is forced to stay outside of the cone).

(5.32m)–(5.32o) correspond to (5.22e) and together with (5.32p)–(5.32r) which correspond to (5.22f) force the future state $x^+$ to stay in one of the half-spaces which contain the current position but not the active obstacle (e.g., whenever $\alpha^1 = 0$, the inequalities (5.32m) reduce to $-h_1^T x^+ \leq -k_1 + M \rho^1(1)$, $-h_3^T x^+ \leq -k_3 + M \rho^1(3)$ thus forcing the inclusion $x^+ \in \mathcal{H}_1^- \cup \mathcal{H}_3^-$ since (5.32p) imposes that at least one of $\rho^1(1), \rho^1(3)$ is zero).

As before, to reduce the number of similar groups of equations which would result from (5.22), we assumed that we are in the case $\sigma^0 = \sigma$ and discarded the redundant inequalities (e.g., (5.32p)–(5.32r) show only the terms which correspond to non-zero values of the cone coefficients).

![Figure 5.4: Illustration of the over-approximated shadow regions in a multi-obstacle environment.](image)

The over-approximated shadow region $\mathcal{B}^+(\Sigma^*, \sigma^*)$, defined as in (5.14), is shown in Figure 5.4. Using Proposition 5.4 i) we provide the mixed-integer representation of this region:

$$
|1 - \sigma^+(1)| + |1 - \sigma^+(3)| \leq N(1 - \alpha^1),
$$

(5.33a)

$$
|1 - \sigma^+(3)| + |\sigma^+(5)| \leq N(1 - \alpha^2),
$$

(5.33b)

$$
|1 - \sigma^+(1)| + |1 - \sigma^+(2)| + |1 - \sigma^+(3)| + |1 - \sigma^+(4)| + |\sigma^+(5)| \leq N(1 - \alpha^3),
$$

(5.33c)

$$
\alpha^1 + \alpha^2 + \alpha^3 \geq 1.
$$

(5.33d)

(5.33a)–(5.33c) reduce to (5.23a) by introducing the sign tuples of the obstacles in the formulation. (5.33d) which corresponds to (5.23b) forces that at least an obstacle is active. E.g., taking $\alpha^1 = 1, \alpha^2 = \alpha^3 = 0$ means that we remain with (5.33a) which reduces to $|1 - \sigma^+(1)| + |1 - \sigma^+(3)| = 0$. Thus, we have that $x^+$ can lie in any region which respects $\sigma^+(1) = \sigma^+(3) = '++'$, in other words $x^+ \in \mathcal{H}_1^+ \cap \mathcal{H}_3^+$.

The over-approximated visible region $\mathcal{B}^+(\Sigma^*, \sigma^*)$, defined as in (5.16) is written in a mixed-integer
representation using Proposition 5.4 ii): 

\[ |1 - \sigma^+(1)| + |1 - \sigma^+(3)| > 0, \quad (5.34a) \]
\[ |1 - \sigma^+(3)| + |\sigma^+(5)| > 0, \quad (5.34b) \]
\[ |1 - \sigma^+(1)| + |1 - \sigma^+(2)| + |1 - \sigma^+(3)| + |1 - \sigma^+(4)| + |\sigma^+(5)| > 0. \quad (5.34c) \]

As for (5.33a)–(5.33c), we introduce the sign tuples of the obstacles and of the feasible tuple characterizing the current position in (5.24). In each of the resulting inequalities at least a term has to be strictly positive. E.g., in (5.34a) we have that

\[ j_1 + (1) j_3 > 0 \]
and / or 

\[ j_1 + (3) > 0. \]
Thus, we have that

\[ x_k \text{ can lie in any region which does not respect simultaneously } \sigma^+(1) = \sigma^+(3) = '+' \]

in other words \( x_k \in \mathcal{K}_1^- \cup \mathcal{K}_3^- \). Due to space restrictions and to the fact that the extension to it is simple, we do not exemplify here Corollary 5.4.

5.5. Study case: an MPC problem with corner cutting avoidance

To highlight the corner cutting issue and the guaranteed avoidance constraints discussed earlier we consider a Model Predictive Control (MPC) scheme applied to a double integrator dynamics in the multi-obstacle environment presented in the illustrative examples of the previous sections.

To begin we consider first the continuous-time dynamics (often used in path planning scenarios for reference trajectory generation, [58], [56]):

\[ \dot{x}(t) = A_c x(t) + B_c u(t), \quad y(t) = C_c x(t), \quad (5.35) \]

with the state \( x(t) \in \mathbb{R}^4 \) – composed from position and velocity, input \( u(t) \in \mathbb{R}^2 \) – the acceleration and the output \( y(t) \in \mathbb{R}^2 \) – the position component of the state. Matrices \( A_c, B_c, C_c \) are given as follows:

\[ A_c = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad C_c = \begin{bmatrix} I & 0 \end{bmatrix}, \quad (5.36) \]

with \( 0 \in \mathbb{R}^{2 \times 2} \) and \( I \in \mathbb{R}^{2 \times 2} \) the ‘zero’ and ‘identity’ matrices.

The next step is to consider a sampling time \( T \) and give the discrete dynamics associated to (5.35):

\[ x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k, \quad (5.37) \]

with \( x_k, u_k \) and \( y_k \) the discrete counterparts of the continuous variables appearing in (5.35) and matrices \( A, B, C \) (obtained via the zero order hold method):

\[ A = \begin{bmatrix} I & T \cdot I \\ 0 & I \end{bmatrix}, \quad B = \begin{bmatrix} T^2 \cdot I \\ T \cdot I \end{bmatrix}, \quad C = \begin{bmatrix} I & 0 \end{bmatrix}. \quad (5.38) \]

With these prerequisite we have all the necessary ingredients to formulate the corner cutting avoidance MPC problem:

\[ u^* = \arg \min_{u_k, \sigma_{k+1}, \ldots, \sigma_{k+N_p-1}, \sigma_k, u_k} \sum_{i=0}^{N_p-1} \|x_{k+i+1}\|_Q + \|u_k\|_R, \quad (5.39a) \]
Exact and over-approximated guarantees for corner cutting avoidance

\begin{align*}
\text{s.t. } x_{k+i+1} &= A x_{k+i} + B u_{k+i}, y_{k+i} = C x_{k+i},
\quad \text{(5.39b)} \\
x_{k+i} &\in \mathcal{X}, u_{k+i} \in \mathcal{U},
\quad \text{(5.39c)} \\
y_{k+i+1} &\notin \mathcal{B}(\Sigma^*, \sigma_{k+i}), \quad i = 0 \ldots N_p - 1,
\quad \text{(5.39d)}
\end{align*}

with \( N_p \), the prediction horizon length: \( Q = \text{diag}(I, 0) \), \( R = I \), (semi-)positive definite weight matrices for state and input; \( \mathcal{X} = \{ x : |x| \leq \begin{bmatrix} 10 & 10 & 10 & 10 \end{bmatrix}^\top \} \), \( \mathcal{U} = \{ u : |u| \leq \begin{bmatrix} 1 & 1 \end{bmatrix}^\top \} \), bounding sets for state and input.

Standard obstacle avoidance formulations would constrain the output to lie outside the union of obstacles (i.e, \( y_{k+i+1} \notin S \)). In (5.39d) we consider instead the corner cutting avoidance constraint \( y_{k+i+1} \notin \mathcal{B}(\Sigma^*, \sigma_{k+i}) \) which forces the output to lie outside of the shadow region defined by the obstacles and the current sign tuple.

Note that (5.39d) describes the over-approximated case: in the sense of (5.27) from Corollary 5.4 ii) where \( y_{k+i}, y_{k+i+1} \) stand for \( x, x^\top \) and constraints \( y_{k+i} \in \mathcal{A}(\sigma_{k+i}), y_{k+i+1} \in \mathcal{A}(\sigma_{k+i+1}) \), implemented as in Remark 5.12, hold. The exact-case corner cutting avoidance can be implemented by using the exact form of the shadow region: \( y_{k+i+1} \notin \mathcal{B}(\Sigma^*, y_{k+i}) \), to which correspond the constraints given in Proposition 5.3.

Note also that the binary variables \( \sigma_{k+i} \) characterizing the predicted output \( y_{k+i} \) are unknown and the result of the optimization problem (5.39). Therefore, for the implementation of the corner cutting avoidance constraints we employ the piecewise descriptions (either as in Proposition 5.3 – the exact case, or as in Corollary 5.4 ii) – the over-approximated case).

5.5.1. Justifications for corner cutting avoidance strategies

In what follows we highlight the necessity of corner cutting avoidance strategies by showing that simple obstacle avoidance and even classical methods like obstacle enlargement are unsatisfactorily.

For a sampling time \( T = 1 \) and horizon length \( N_p = 5 \) we depict in Figure 5.5 (a) the trajectories obtained for standard obstacle avoidance and with corner cutting avoidance guarantees (the over-approximated case). While both of them provide feasible values (red circle and ‘x’ symbols in the figure), when computing the continuous trajectory (via numerical integration of the continuous dynamics to which the discrete input is applied – blue circle and ‘x’ symbols) we observe that the classic obstacle avoidance trajectory ‘cuts’ one of the obstacles.

As shown in [142], the continuous \( y(t) \) output lies in the region \( y_k + \text{Conv}\{tv_k + \frac{t^2}{2} u_k\}, \) where \( v_k \) denotes the velocity component of the state \( x_k \). In other words, even for a known initial velocity \( v_k \), the next step is uncertain up to the set \( \frac{T^2}{2} \mathcal{U} \). Using this set as a safety region (or, equivalently, enlarging the obstacle(s) with the same amount) guarantees corner cutting avoidance. The obstacles thus enlarged are depicted in Figure 5.5 (b) with dotted contours. As it can be seen, the resulting shapes greatly restrict the feasible domain (in fact, the origin is made infeasible) and increase the complexity of the problem (the enlargement increases the number of facets which define an obstacle and, consequently, the number of associated binary variables).

Reducing the sampling time may prove useful at first glance (since the safety region decreases proportionally with \( T^2 \)) but it also means a reduction in available computation time, and, more importantly an increase in the prediction horizon length (decreasing the sampling time means that the horizon length has to increase in order to cover the same continuous time interval). All these
caveats means that obstacle enlargement cannot be always employed and that alternatives (such as the strategies proposed here) have to be considered.

Henceforth, we will assume that the agent moves along straight lines in-between consecutive discrete points (see the illustration from Figure 5.5 (c)). This is a reasonable assumption for a sampling time $T$ small enough. Note that the deviation from a straight line is given by the trajectory’s curvature which is much less than its reach. In other words, the size of the safety region which covers the curvature of the trajectory is small enough to be manageable. Note also that in [142] additional constraints are introduced to ensure inter-sample avoidance of the real trajectory of an agent with double integrator dynamics.

5.5.2. Comparison of various corner cutting avoidance strategies

In what follows we illustrate in Figure 5.6 various obstacle avoidance strategies:

S1) standard obstacle avoidance, without corner cutting avoidance guarantees (green triangle);
S2) obstacle avoidance with obstacle enlargement for a safety region $4 \cdot \frac{T^2}{\gamma}$ (blue square);
S3) obstacle avoidance with corner cutting avoidance guarantees – the exact case, as in Proposition 5.3 (black diamond);
S4) obstacle avoidance with corner cutting avoidance guarantees – the over-approximated case, as in Corollary 5.4 ii) (red circle);

All these strategies are implemented for the aforementioned MPC problem, with initial point $x_0 = \begin{bmatrix} 2 & 2.5 & 2.5 & 0.5 \end{bmatrix}^T$, sampling time $T = 0.5$, prediction horizon length $N_p = 10$ and use the CPLEX solver [34], except case S3) where the MINLP solver SCIP [98] and a shorter prediction horizon ($N_p = 5$) are used (the latter, due to the excessive size of the problem for the case $N_p = 10$).

Figure 5.6.: Illustration of trajectories with and without corner cutting avoidance.

Assuming a simulation horizon $N_{sim} = 30$, again with the exception of scenario S3) where a shorter $N_{sim} = 10$ was considered, we obtain that the total simulation times are 0.0957, 0.6353, 11.7452 and 0.0232 respectively. With the caveat that these numbers will change for different values of the problem parameters, we note that scenarios S1) and S4) are comparable, that S2) is an order of magnitude slower than them and that S3) is one order of magnitude slower than S2). Broadly, these results are to be expected: S1) and S4) employ binary-only avoidance constraints and lead to a MILP formulation whereas S2), while still using only binary formulations, does so for a more complex feasible domain (due to the obstacle enlargement procedure). Lastly, S3) with its convoluted nonlinear terms leads to a MINLP formulation and is by far the slowest approach.

To assess the performance of each strategy we measure the total trajectory length$^1$ in each case. The results are: 10.0180, 11.5498, 10.1024 and 10.8133, respectively. Again, these values are inline with the expectations: S1) gives the lowest value but does so by cutting the corner of the right-most obstacle; S2) is forced to avoid conservatively the obstacles due to their enlargement; S3) and S4) provide the best results (even if, in the case of S3) the prediction horizon is half the size of the one used in S4)).

$^1$To get the trajectory’s length we add the straight-line segments which link the discrete points of the trajectory rather than the ‘real’ trajectory obtained through numerical integration since the cost appearing in the MPC problem involves only discrete variables.
5.5.3. Performance and robustness issues

To assess the performance and robustness issues of the over-approximated corner cutting avoidance case (strategy S4) we consider multiple prediction horizon lengths \(N_p \in \{5, 10, 15\}\), sampling times \((T \in \{0.1, 0.5, 1\}\) and the feasible points from within the box \(-7.5 \leq y_k \leq 7.5\) spaced with a step of size 1 (there are 245 of them). To each combination of parameters we apply the MPC optimization problem (5.39) with strategy S4) and observe the various elements of interest (computation time, feasibility of the solution, length of the obtained trajectory).

First, we depict in Table 5.1 the mean computation time (averaged over the simulation horizon and over all feasible points) and the maximum time (the largest time spent computing a step in the simulation), respectively. As expected, the computation time (both on average and maximum values) increases with the length of the prediction horizon\(^2\). Interestingly, the same can be told about the sampling time. Having a larger sampling time forces a more coarse behavior on the agent: it has to take larger steps and to discard a larger part of the feasible domain (thus leading to more complex computations).

The maximum computation time shows that there are simulation steps for which the difficulty of the problem is markedly increased w.r.t. the average value (e.g., when the agent is near an obstacle and has to judge the optimal path for corner cutting avoidance). To highlight this variation we illustrate in Figure 5.7 the average and maximum times for each of the 245 initial feasible points in the case \(N_p = 10, T = 1\).

![Figure 5.7: Illustration of mean and maximum computation time for the case \((N_p = 10, T = 1)\)](image)

\(^2\)Note that for \(N_p = 15, T = 0.5\) there exists at least a sample time at which the computation time exceeds the available time.
Another element of interest is the trajectory length from a given starting point. Figure 5.8 shows the trajectories obtained for each combination of prediction horizon and sampling time: circle, diamond and triangle markers denote the sampling time (0.1, 0.5, 1 respectively) and solid blue, dashed red and loosely dashed green denote prediction horizon (5, 10, 15, respectively) starting from $x = \begin{bmatrix} 5.5 & 2.5 & 0 & 0 \end{bmatrix}^T$. E.g., a solid blue line with triangle marker denotes the trajectory obtained for parameters $N_p = 5$ and $T = 1$.

![Figure 5.8](image)

Figure 5.8.: Illustration of path lengths for each feasible initial point.

We observe that the defining element is the sampling time: the trajectories almost coincide for $T = 0.5$ and $T = 1$ and are similar for $T = 0.1$, regardless of the size of the prediction horizon. Furthermore, care should be taken when selecting the sampling time: too small and the trajectory does not reach its destination (the simulation horizon $N_{\text{sim}} = 30$ is not enough for the case $T = 0.1$), too large and the trajectory becomes unwieldy (the case $T = 1$).

5.6. Conclusions

This chapter presented an analysis of the corner cutting avoidance problem for a multi-obstacle environment. Exploiting the underlying structure provided by a hyperplane arrangement, exact and over-approximated forms of the corner cutting avoidance conditions have been provided. We have shown that the mapping of the under-shadow (and of its complement, the visible) region is piecewise defined on the feasible cells of the hyperplane arrangement. Various mixed-integer formulations have been considered and compared with the state of the art. Furthermore, the proposed theoretical results have been implemented in an MPC design in order to show their benefits through simulations and comparison results. Future work will concentrate on using these results in the dual problem: the coverage of a multi-obstacle environment with multiple agents.
6. Active Fault Detection and Isolation in a Zonotopic Framework

In this chapter I show an active fault detection which employs set descriptions and considers transitional behavior. Specifically, I consider a plant affected by multiple faults (modeled through a piecewise affine formalism) and using a bank of finite-window observers and an artificially-induced feedback delay I provide an exact fault detection and isolation (FDI) mechanism, integrated into the overall fault tolerant control scheme. Using zonotopic characterizations for the sets of interest in order provides explicit conditions for FDI exactness (such that a fault occurrence can be signaled unambiguously) and alleviates the numerical issues specific to set operations.

The chapter is based on conference article [C27]:

WOS: 000427419900098.

6.1. Preliminaries

Fault tolerant control (FTC) is one of today’s main topics of interest in the control community [25]. The recent proliferation of large-scale, complex systems has raised the chances of fault occurrences (either at actuator, sensor or plant dynamics level). Thus, a reliable FTC scheme with an exact fault detection and isolation (FDI) together with a robust reconfiguration mechanism is becoming essential.

In what follows, we consider a set-based approach [0, 155]. Having bounding sets for noises, disturbances and model variations it is possible to bound the signals of interest and thus, to characterize explicitly separation conditions which ensure FDI. To this end we use the set-theoretic notions of robust positive invariance and reachability. The former allows offline computations and gives a priori stability guarantees while the latter allows to handle the transitional behavior sparked by fault(s) occurrences [154]. There are several issues in the literature which are not usually tackled:

i) The FDI separation condition may not be verified for the current operation conditions.

ii) The FDI mechanism may need a non-zero observation window to asses a fault occurrence. This means that the closed-loop scheme may use faulty information.
iii) The computation of invariant / reachable sets is cumbersome for large dimensions and/or for long intervals [22].

For issue i) the key is to observe that the sets involved are parametrized after variables influenced by the control scheme [C23]. Therefore, choosing them such that the condition is always respected avoids the issue of false alarms or missed faults.

To simplify the set computations and help with issue ii) we induce a delay in the feedback generation. Assuming fault persistence and given reference dynamics, ‘steady’ and transitional sets which bound the tracking error and estimation error dynamics are computed. Residual sets (coming from the output estimation sets) are used to check the FDI condition. Since these sets are parametrized after the reference state and input, we use these variables to guarantee exact FDI.

Lastly, to handle issue iii), we consider zonotopic sets. A sub-class of the polyhedral sets, they are increasingly used due to their resilience to the “dimensionality curse” [176], e.g., for reachability analysis [7], collision detection [60] or guaranteed state estimation [60]. In contrast to polytopes, the operations with zonotopic sets are significantly less computationally demanding, and, also, do not raise numerical instabilities [9], [8]. Toolboxes like CORA [8] handle zonotopic sets representations and operations efficiently.

Combining all these elements we reach an explicit formulation which ensures FDI and involves reference inputs and states. Due to the nonlinear nature of the problem (a bilevel formulation), we consider the mixed integer formalism proposed in [153] but, instead of directly controlling the inputs applied to the control scheme, we control the reference inputs and assume the feedback law already given.

### 6.2. Prequisites

In the rest of the chapter sets will be used to bound various signals with the end goal of detecting and isolating unambiguously a fault occurrence. This implies the use of (robust) invariance, set projection, set addition and all the other tools used in set-theoretic methods.

While there are multiple choices for representing a system (e.g., via polyhedral or ellipsoidal sets [22]), in this chapter we consider zonotopes as they provide an excellent compromise between numerical complexity and fidelity of representation:

**Definition 6.1 ([84]).** A zonotope is a centrally symmetric polytope, which can be described as a Minkowski sum of line segments. In its generator representation a zonotope $Z$ is described by center $c \in \mathbb{R}^n$ and generators $g_1, \ldots, g_n \in \mathbb{R}^n$:

$$Z = \{c + \sum_{i=1}^{n_g} \xi_i g_i : \|\xi\|_\infty \leq 1\}$$

or, compactly with $G = [g_1, \ldots, g_n]$, as

$$Z = \{G\xi + c : \xi \in \mathbb{R}^{n_g}, \|\xi\|_\infty \leq 1\}$$

In the next sections, we use notation $Z(G, c)$ to denote (6.1).
Let us consider two zonotopes: \( Z_1 = \mathcal{Z}(G_1, c_1) \subset \mathbb{R}^n \), \( Z_1 = \mathcal{Z}(G_2, c_2) \subset \mathbb{R}^n \) and a matrix \( R \in \mathbb{R}^{m \times n} \). Then the following properties hold \([50]\):

i) is closed under linear transformation:

\[
R\mathcal{Z}(G_1, c_1) = \mathcal{Z}(RG_1, Rc_1);
\]

(6.3)

ii) is closed under Minkowski sum:

\[
\mathcal{Z}(G_1, c_1) \oplus \mathcal{Z}(G_2, c_2) = \mathcal{Z}\left[\begin{bmatrix} G_1 & G_2 \end{bmatrix}, c_1 + c_2\right];
\]

(6.4)

iii) is symmetric, up to its center:

\[
-Z_1 = -\mathcal{Z}\{G_1, c_1\} = \mathcal{Z}\{G_1, -c_1\}.
\]

(6.5)

In particular, the Minkowski addition and projection operations greatly simplify for zonotopic sets with respect to their polyhedral counterparts and make large-scale computations feasible and numerically robust \([7]\).

For further use, we define the notion of robust invariance (RPI) \([J13]\).

**Definition 6.2.** For LTI dynamics \( x_{k+1} = Ax_k + \delta_k \) with \( \delta_k \in \Delta \), the set \( \Omega \) is called RPI iff the set inclusion

\[
A\Omega \oplus \Delta \subset \lambda \Omega,
\]

(6.6)

holds for \( 0 < \lambda \leq 1 \).

When \( 0 < \lambda < 1 \) we call \( \Omega \) contractive and convergence inside it in a finite time is guaranteed. ♦

It is well-known \([81]\) that the minimal RPI (mRPI) set \( \Omega_\infty \) can be obtained through the set recurrence:

\[
\Omega_0 = \{0\}, \quad \Omega_{k+1} = A\Omega_k \oplus \Delta.
\]

(6.7)

In general, the limit set of (6.7) cannot be obtained explicitly. Instead, various methods exist for computing arbitrarily close approximations \([136]\). The repeated additions in (6.7) are manageable for zonotopic sets. In fact, due to (6.3) and (6.4), and assuming that \( \Delta = \mathcal{Z}(G, c) \) we have that

\[
\Omega_k = \mathcal{Z}\left[\begin{bmatrix} G & \cdots & A^{k-1}G \end{bmatrix}, c + \cdots + A^{k-1}c\right]
\]

(6.8)

Using (6.8) for \( k \to k + 1 \) and introducing it in (6.6) we have that, for a desired \( \lambda \), the minimal value of \( k \) for which (6.6) holds is

\[
k^* = \arg \min_k A^k \mathcal{Z}(G, c) \subset \lambda \mathcal{Z}(G, c).
\]

(6.9)

### 6.3. Problem description

Let us consider the dynamics

\[
x_{k+1} = A(i_k)x_k + B(i_k)u_k + r(i_k) + B_w(i_k)w_k
\]

(6.10a)
where \( x_k, x_{k+1} \in \mathbb{R}^n \) denote the current and successor states, \( u_k \in \mathbb{R}^p \) the input, \( y_k \in \mathbb{R}^p \) the output, \( w_k \in \mathbb{W} \subset \mathbb{R}^{p\omega} \) the process noise and \( v_k \in \mathbb{V} \subset \mathbb{R}^{m\omega} \) the measurement noise. Matrices \( A, B, B_w, C, D_v \) and bias terms \( r, s \in \mathbb{R}^n \) are of appropriate dimension and take values from pre-defined collections of cardinality \( N \), as indexed by \( i_k \) (e.g., \( A(i_k) \) takes values from \( \{ A_1, A_2, \ldots, A_N \} \)).

Formulation (6.10) is both general enough to handle realistic dynamics (subject to model variation, noises, faults, etc.) and restrictive enough to provide, coupled with the tools from Section 4.2, numerically manageable constructions. In particular, we use (6.10) to model dynamics which alternate between a healthy mode (by convention, \( i_k = 0 \)) and various pre-defined\(^1\) faulty modes (\( i_k \neq 0 \)). What remains to be assessed is the fault occurrence (via a FDI mechanism).

### 6.3.1. State estimation

Let us assume a bank of observers, one per mode of functioning. That is, we define the \( j \)-th observer as a finite-window Luenberger observer whose internal model is given by the \( j \)-th mode of dynamics

\[
y_k = C(i_k)x_k + s(i_k) + D_v(i_k)v_k
\]

(6.10b)

where \( y_k, x_{k+1} \in \mathbb{R}^m \) denote the current and successor states, \( u_k \in \mathbb{R}^p \) the input, \( y_k \in \mathbb{R}^p \) the output, \( w_k \in \mathbb{W} \subset \mathbb{R}^{p\omega} \) the process noise and \( v_k \in \mathbb{V} \subset \mathbb{R}^{m\omega} \) the measurement noise. Matrices \( A, B, B_w, C, D_v \) and bias terms \( r, s \in \mathbb{R}^n \) are of appropriate dimension and take values from pre-defined collections of cardinality \( N \), as indexed by \( i_k \) (e.g., \( A(i_k) \) takes values from \( \{ A_1, A_2, \ldots, A_N \} \)).

Formulation (6.10) is both general enough to handle realistic dynamics (subject to model variation, noises, faults, etc.) and restrictive enough to provide, coupled with the tools from Section 4.2, numerically manageable constructions. In particular, we use (6.10) to model dynamics which alternate between a healthy mode (by convention, \( i_k = 0 \)) and various pre-defined faulty modes (\( i_k \neq 0 \)). What remains to be assessed is the fault occurrence (via a FDI mechanism).

Dynamics (6.11) can be interpreted as follows: at the current moment ‘\( k \)’, all the available information of the last \( \tau \) instants of time \( (u_{k-\tau+1}, \ldots, u_k, y_{k-\tau+1}, \ldots, y_k) \), together with initialization \( \hat{x}_{k-\tau+1} = x_{k-\tau+1} \) provide a state estimation \( \hat{x}_k \) and output estimation \( \hat{y}_k \). While arguably this construction discards some information (i.e., older information from the estimation), it will prove to be useful for the FDI mechanism.

Combining (6.10) and (6.11) we get the state and output estimation error, \( \Delta_k \triangleq x_k - \hat{x}_k \) and \( \Delta_k \triangleq y_k - \hat{y}_k \), respectively, associated to the \( j \)-th observer:

\[
\Delta_k = \mathbb{J}_k \Delta_{k+1} = z_{k-\tau+1}
\]

(6.12a)

\[
\mathbb{J}_k = [\mathbb{A}(j) - \mathbb{L}^j \mathbb{C}(j)] \mathbb{X}_k + [(\mathbb{A}(i_\ell) - \mathbb{A}(j)) - \mathbb{L}^j (\mathbb{C}(i_\ell) - \mathbb{C}(j))] x_\ell + \mathbb{B}(i_\ell - B(j)] u_\ell + r(i_\ell) - r(j) + B_w(i_\ell) w_\ell - \mathbb{L}^j [s(i_\ell) - s(j)] - \mathbb{L}^j D_v(i_\ell) v_\ell,
\]

(6.12b)

\[
\mathbb{Y}_k = \mathbb{C}(j) \mathbb{X}_k + \mathbb{C}(i_\ell) - \mathbb{C}(j)] x_\ell + s(i_\ell) - s(j) + \mathbb{D}_v v_\ell,
\]

(6.12c)

for \( \ell = k - \tau + 1, \ldots, k \) and \( z_{k-\tau+1} \), the tracking error at time instant \( k - \tau + 1 \), to be defined later.

Matrix \( \mathbb{L}^j \) is taken such that the closed-loop state-matrix \( \mathbb{A}_{L^j} = \mathbb{A}(j) - \mathbb{L}^j \mathbb{C}(j) \) is stable (always possible if the pair \( \{ \mathbb{A}(j), \mathbb{C}(j) \} \) is observable).

Note that when the model used in the observer and the active dynamics coincide (i.e., \( i_k = j \),

---

\(^1\)Note that we assume both the fault models and their magnitudes known.

64 of 160
(6.12) reduces to:

\[
\begin{align*}
x_{k_{\tau+1}}^j &= z_{k_{\tau+1}} \\
x_{\tau+1}^j &= A L x_{\tau}^j + B w_{\ell}^j + L D w_{\ell}^j v_{\ell}, \\
y_{\ell}^j &= C (j) x_{\ell}^j + D v_{\ell}^j v_{\ell}.
\end{align*}
\]

(6.13)

for \( \ell = k - \tau + 1, \ldots, k \) and \( z_{k_{\tau+1}} \), the tracking error at time instant \( k - \tau + 1 \), to be defined later.

### 6.3.2. Tracking error

Let us assume the reference dynamics (with \( u_{k}^j \) persistent references a priori given):

\[
\begin{align*}
x_{k+1}^j &= A(j)x_k^j + B(j)u_k^j + r(j), \\
y_k^j &= C(j)x_k^j + s(j).
\end{align*}
\]

(6.14)

**Remark 6.1.** The reference models (6.14) are in fact the nominal models (with noises \( w_k \) and \( v_k \) discarded) from (6.10).

To close the loop we take

\[
u_k = u_k^j + K^j (\hat{x}_{k_{\tau+1}}^j - x_{k_{\tau+1}}^j)
\]

(6.15)

where the selection of feedback matrix, state estimation and reference values (\( K^j, \hat{x}_{k_{\tau+1}}^j, u_k^j, x_{k_{\tau+1}}^j \)) will be done by the fault tolerant scheme described in Section 6.4.2. Note that the use of an artificially delayed information (the state estimation \( \hat{x}_{k_{\tau+1}}^j \)) is required by the FDI construction.

A typical measure of performance and stability is the analysis and bounding of the state and output tracking errors \( \eta_k^j \hat{=} x_k - x_k^j \) and \( \xi_k^j \hat{=} y_k - y_k^j \) respectively. Gathering (6.10), (6.14) and (6.15) we have:

\[
\begin{align*}
z_{k+1} &= A(i_k)z_k + B(i_k)K^j z_{k_{\tau+1}} + [r(i_k) - r(i_k^j)] + [A(i_k) - A(i_k^j)] x_k^j + [B(i_k) - B(i_k^j)] u_k^j \\
&- B(i_k^j) \hat{x}_{k_{\tau+1}}^j + B(i_k)w_k \\
\xi_k &= C(i_k)z_k + [C(i_k) - C(i_k^j)] x_k^j + D v(i_k) v_k + [s(i_k) - s(i_k^j)].
\end{align*}
\]

(6.16)

Matrices \( K^j \) are taken such that the closed-loop dynamics are stable [106]. Assuming that the active dynamics (6.10) and the selected gain and state estimation coincide (i.e., \( i_k = i_k^j = j \)), the tracking error dynamics (6.16) become

\[
\begin{align*}
z_{k+1} &= A(j)z_k + B(j)K^j z_{k_{\tau+1}} - B(j)K^j x_{k_{\tau+1}} + B w_{\ell}^j w_k \\
\xi_k &= C(j)z_k + D v(j) v_k.
\end{align*}
\]

(6.17)

With (6.15) we revisit (6.12) and highlight the reference state and input:

\[
\begin{align*}
x_{k_{\tau+1}}^j &= z_{k_{\tau+1}} \\
x_{\tau+1}^j &= A L x_{\tau}^j + [A(i_k^j) - A(j)] - L D (C(i_k^j) - C(j)] (x_{\ell}^j + z_{\ell}) \\
&+ [B(i_k^j) - B(j)] (u_{\ell} + K^j (z_{k_{\tau+1}}^j - x_{k_{\tau+1}}^j)) + r(i_k^j) - r(j)
\end{align*}
\]

(6.18)
\begin{equation}
+ B_w(i'_\ell)w_\ell - L^j [s(i'\ell) - s(j)] - L^jD_v(i'_\ell)v_\ell,
\end{equation}
\begin{equation}
y^j_\ell = C(j)x^j_\ell + [C(i'\ell) - C(j)](x^j_\ell + z_\ell) + s(i'_\ell) - s(j) + D_v(i'_\ell)v_\ell,
\end{equation}
for \( \ell = k - \tau + 1, \ldots, k \). We use index \( i'_\ell \) when rewriting \( u_\ell \): for now we make no assumption on the selected index in (6.15).

### 6.3.3. Set characterizations

For further use we require the sets which characterize the state estimation error (6.12) and tracking error (6.16) dynamics.

Recall that by construction, observers (6.11) have a finite window (i.e., the \( k \)-th state and output estimations are derived only from the information available in the interval \( k - \tau + 1, \ldots, k \)). Hence, whatever happens outside of this interval has no influence in the estimations and, implicitly, on the sets which bound them.

First, several assumptions are necessary:

A1) any two consecutive switches are separated by at least \( \tau + \tau_c \) instants of time;

A2) (6.15) uses correct information (the indices of the state estimation and of the dynamics are matched, \( i_k = i'_k = j \));

Assuming that dynamics (6.10) have been under the \( j \)-th mode for at least \( \tau \) consecutive time instants (Assumption A1)), the estimation error of the \( j \)-th observer follows (6.13), thus allowing to compute the corresponding bounding sets:

\begin{align*}
\hat{X}^{j,j}_{k-\tau+1} &= \{z_{k-\tau+1}\}, \\
\hat{X}^{j,j}_{\ell+1} &= A_{L_j}\hat{X}^{j,j}_{\ell} + B_w(j)W \oplus \{-L^jD_v(j)V\}, \\
\hat{Y}^{j,j}_{\ell+1} &= C(j)\hat{X}^{j,j}_{\ell} + D_v(j)V.
\end{align*}

for \( \ell = k - \tau + 1, \ldots, k \). Consequently, we have inclusions \( x^j_{\ell+1} \in \hat{X}^{j,j}_{\ell+1} \) and \( y^j_{\ell} \in \hat{Y}^{j,j}_{\ell} \).

Assuming that the correct estimation is used in (6.15) – Assumption A2) the tracking error follows (6.17). Further assuming that \( x^j_{k-\tau+1} \in \hat{X}^{j,j} \), \( w_k \in W \) hold, an invariant set \( \mathcal{Z}^{j,j} \) is computed as follows: using the extended tracking error state \( \mathbf{x}_k = [z_k^T \ z_{k-1}^T \ \ldots \ z_{k-\tau}^T]^T \), we obtain an invariant set which guarantees \( \mathbf{x}_k \in \mathcal{Z}^j \). Projecting along the subspace associated with \( z_k \) we obtain the set which bounds \( \mathbf{x}_k \) while under the selected index \( j \):

\begin{equation}
\mathbf{z}_k \in \mathcal{Z}^{j,j}.
\end{equation}

Up to now we have shown the sets bounding the ‘steady’ estimation and tracking error (when the observer and plant model coincide). For further use, we have to consider the case of model mismatch as well.

Assuming that dynamics (6.10) have been under the \( j' \)-th mode for at least \( \tau \) consecutive time instants (Assumption A1)), the estimation error of the \( j \)-th observer follows (6.18) with the additions \( i_\ell = i'_\ell = j' \) and \( x^j_{\ell-\tau} \in \hat{X}^{j',j'} \). This leads to bounding sets for the state/output estimation
parametrized after the tracking error and state/input references:

\[
\tilde{x}_{k+1}^{j'} = \{z_{k+1}^{j'}\} \\
\tilde{x}_{k+\ell}^{j'} = A_{k,\ell} \tilde{x}_{k+\ell}^{j'} \\
\oplus \{\{(A'(j') - A(j)) - L'(C'(j') - C(j))\}(x_{\ell}^{j'} + z_{\ell})\} \\
\oplus \{B'(j') - B(j)\}(u_{\ell}^{j'} + \{\{z_{\ell}\} + \{-\tilde{x}_{k+\ell}^{j'}\})\}) \\
\oplus \{r(j') - r(j)\} + Bw(j')W + \{-L'[s(j') - s(j)]\} + \{-L'D_{e}(j')V\},
\]

\[
\tilde{y}_{k+\ell}^{j'} = C(j')\tilde{x}_{k+\ell}^{j'} \oplus \{\{(C'(j') - C(j))(x_{\ell}^{j'} + z_{\ell})\} \oplus \{s(j') - s(j)\} + D_{e}(j')V\}
\]

for \(\ell = k - \tau + 1, \ldots, k\). As expected, (6.21) reduces to (6.19) when \(j = j'\). Lastly, we evaluate the behavior of the tracking error in the interval after a fault and while the correct observer is not yet correctly selected. Using (6.16) and with the notations \(i_{k} \rightarrow j, i_{k}' \rightarrow j'\) and assuming that the switch \(j \rightarrow j'\) happens at \(k\) and that \(x_{k+\ell}^{j'} \in \tilde{x}_{k+\ell}^{j'}\) we have the set recurrence:

\[
\tilde{x}_{k}^{j'} = \{z_{k}\} \\
\tilde{x}_{k+\ell}^{j'} = A(j')\tilde{x}_{k+\ell}^{j'} \oplus \{B(j')K^{j'}z_{k+\ell+1}\} \oplus \{-B(j)K^{j'}\tilde{x}_{k+\ell}^{j'}\} \oplus Bw(j')W \\
\oplus \{[A(j) - A(j')]x_{k+\ell}^{j'} + s(j') - s(j)\} + D_{e}(j')V.
\]

for \(\ell = k + 1, \ldots, k + \tau\).

Remark 6.2. The sets defined here are parametrized after the tracking error and after the reference state and input values\(^2\). The tracking errors appearing in (6.19a), (6.21a)–(6.21c) and (6.22a) will be replaced with bounding sets as resulted from the FTC scheme. The reference state and input appearing in (6.21b)–(6.21c) and (6.22b) will be used as decision variables to ensure FDI.

Figure 6.1.: Timeline for a fault scenario with set inclusion illustration.

6.4. Fault tolerant control scheme

The use of bounding sets allows an exact FDI implementation (without missed faults and false alarms) but it requires that the set inclusions assumed in the construction of (6.19), (6.21), (6.20) and (6.22) hold at all times.

First we discuss the set inclusions and their convergence to ‘steady’ values and second we show how
to explicitly choose the reference states and inputs such that FDI is guaranteed.

\(^2\)To keep the notation simple we ignored this aspect but whenever necessary, we will employ the full form. For example, \(\tilde{y}_{k}^{j'}\) from (6.21) may be written as \(\tilde{y}_{k}^{j'}(z_{k-2\tau+1} \ldots z_{k}, u_{k-\tau+1} \ldots u_{k}, x_{k-\tau+1} \ldots x_{k})\).
6.4.1. Fault tolerant control scheme implementation

As a first step we revisit the sets obtained earlier and show the stability of the closed-loop scheme. To better clarify these issues, let us consider the following fault scenario, also illustrated in Figure 6.1 where, at \( k = k_1 + 1 \) the plant switches from functioning mode \( j \) to mode \( j' \):

I) For \( k \leq k_1 \) the plant dynamics are in the j-th mode for a sufficiently long time and the plant/control models are matched (\( i_k = i'_k = j \)). Therefore, the tracking error and the estimation errors are in their respective sets (\( z_k \in \mathcal{X}^{j,j}, x_k' \in \hat{\mathcal{X}}^{j,j}(\mathcal{X}^{j,j}), x_k'' \in \hat{\mathcal{X}}^{j',j}(\mathcal{X}^{j,j}) \)).

II) From \( k = k_1 + 1 \) onwards the dynamics switch to mode \( i_k = j' \) (i.e., a (different) fault occurs). Hence, the tracking error is no longer guaranteed to use a matched reference model (\( i'_k \neq j' \)) and thus \( z_k \) is given by dynamics (6.16) and its bounding set enlarges (\( z_k \in \mathcal{X}^{j,j'}_k \)). The estimation errors are indefinite since they are based on mixed information (the observation window contains data from both the \( j \) and \( j' \) modes of functioning).

III) From \( k = k_1 + \tau + 1 \) and until \( k = k_1 + \tau + \tau_c \) the \( j' \)-th observer retrieves the correct estimation and the closed-loop is guaranteed to use the correct information (\( i_k = i'_k = j' \)). The tracking error is not yet in its invariant set (\( \mathcal{X}^{j',j'} \)) but converges towards it (in \( \tau_c \) time instants, computed a priori). The estimation error for the mismatched observers (for any \( j \neq j' \)) is computed iteratively via (6.21) but is cumbersome due to the terms \( z_{k_1+\tau+1} \ldots z_{k_1+\tau+\tau_c} \) which are bounded by transitional sets. At \( k = k_1 + \tau + \tau_c \) we have inclusion \( z_k \in \mathcal{X}^{j',j'}_k \) and hence the estimation errors (and associated output errors) are again within their ‘steady’ sets.

Recalling Remark 6.2, note that in step I we use \( \hat{\mathcal{X}}^{j,j}(\mathcal{X}^{j,j}), \hat{\mathcal{X}}^{j',j}(\mathcal{X}^{j,j}) \). By this, we mean that in (6.13) and (6.21) we replace \( z_{k-\tau+1} \) and, \( z_{k-2\tau+1} \ldots z_k \), respectively, with \( \mathcal{X}^{j,j} \).

6.4.2. Active fault detection and isolation

Recall that within our framework, the plant is called “under-fault” whenever the active index in (6.10) is \( i_k \neq 0 \). Thus, we require so-called residual signals which are sensible to specific faults occurrences and whose behavior can unambiguously state whether the system is under fault (detection) and, if so, which is the active fault (isolation). In here we take the observer outputs (6.11) as the residuals.

From (6.19) and (6.21) we have that the output (6.12b) can stay in one of \( N \) possible output sets (in \( \hat{\mathcal{Y}}^{j,j} \) whenever the plant and observer models coincide and in one of the remaining \( N-1 \) sets, \( \hat{\mathcal{Y}}^{j',j'} \) when \( j \neq j' \)). Hence, a sufficient condition to unambiguously decide whether the plant is in the j-th mode of functioning is to have

\[
\hat{\mathcal{Y}}^{j,j}(\mathcal{X}^{j,j}) \cap \hat{\mathcal{Y}}^{j',j}(\mathcal{X}^{j,j}) = \emptyset, \quad \forall j \neq j',
\]  

(6.23)

Then, the set inclusion \( \mathbf{y}_k^j \in \hat{\mathcal{Y}}^{j,j} \) is unambiguous (in the sense that \( \mathbf{y}_k^j \in \hat{\mathcal{Y}}^{j,j} \) implies \( \mathbf{y}_k^j \notin \hat{\mathcal{Y}}^{j',j}, \forall j' \neq j \)). Repeating (6.23) for each of the observers allows to uniquely identify the fault (out of the N observers just the one where the internal model and the dynamics model coincide will respect \( \mathbf{y}_k^j \in \hat{\mathcal{Y}}^{j,j} \)):

\[
i'_k = \arg \min_{j: \mathbf{y}_k^j \in \hat{\mathcal{Y}}^{j,j}} \|\mathbf{y}_k^j\|_2.
\]  

(6.24)
Sets in control. Applications to fault tolerant control and motion planning.

Using the index from (6.24) into (6.15) closes the loop and guarantees that the tracking error dynamic remains in a predefined domain. Several remarks are in order.

**Remark 6.3.** Any fault could be isolated from a single observer as long as \( j' \neq j'' \leftrightarrow \tilde{Y}^{j,j'} \cap \tilde{Y}^{j,j''} = \emptyset \). However, in general this is hard to check and since we already have a bank of observers at our disposal, we prefer to check (6.23).

**Remark 6.4.** Usually the problem of “closing the loop” is ignored when devising the FDI mechanism, i.e., the effects of the fault are ignored when analyzing the residuals. As it can be clearly seen in (6.18) and (6.16) these effects cannot be ignored since \( \tilde{Y}^{j,j'} \).

**Remark 6.5.** Relation (6.23) assumes that \( z_k \in \mathcal{Z}^{j,j} \) holds. This corresponds to step I) from Figure 6.1. Consequently, we avoid checking the FDI condition during steps II) and III). Instead, having detected the jump \( j \rightarrow j' \) at \( k = k_1 + 1 \) we know (via Assumption A1)) that no other switch will happen in this interval. Hence, we let the estimation error and tracking error sets to settle and re-activate the FDI condition only after the lapsing of \( \tau + \tau_c \) instants. Thus, we avoid the cumbersome calculations of the interconnected transitional estimation and tracking error sets resulting from (6.21) and (6.22).

Up to now, we have only stated the constraints which ensure FDI. The more interesting question is whether it is possible (and if so, how) to force the validation of the FDI condition (6.23). The key is to notice that \( \tilde{Y}^{j,j'} \) is parametrized after reference variables\(^3 \) \( x_k^{j-\tau+1}, u_k^{j-\tau+1} \ldots u_k^j \).

What remains is to extract the ‘variable’ part out of \( \tilde{Y}^{j,j'} \). To this end, let us define the following operators:

\[
\begin{align*}
A^N_A &= 
\begin{bmatrix}
A & 0 \\
A^2 & \vdots \\
\vdots & \ddots \\
A^N & 0
\end{bmatrix},
B^N_{A,B} &= 
\begin{bmatrix}
B & 0 \\
AB & 0 \\
\vdots & \ddots \\
A^{N-1}B & B
\end{bmatrix}, \\
C^N_{C,A,B} &= 
\begin{bmatrix}
CA^{N-1}B & \ldots & CB
\end{bmatrix}.
\end{align*}
\]

(6.25a)

Further denoting \( \bar{x}_k^j = \left[ x_k^{j-\tau+1} \ldots x_k^j \right]^\perp, u_k^j = \left[ u_k^{j-\tau+1} \ldots u_k^j \right]^\perp \) and \( X(j',j) = X(j') - X(j) \) we have that the variable part in \( \tilde{Y}^{j,j'} \) is given by

\[
C^T_{C(j),A_{L,j},B(j',j)} \bar{x}_k^j + C^T_{C(j),A_{L,j},A_{L,j'},A_{L,j'}} \bar{x}_k^j + C(j',j) x_k^j
\]

(6.26)

with \( A_{L,j,j'} = A(j',j) - L^j C(j',j) \).

Using (6.14) we have that

\[
\begin{align*}
\bar{F}_k &= A^T_{A(j)} x_k^{j-\tau+1} + B^T_{A(j),B(j)} u_k^j + B^T_{A(j),1} F^j \\
x_k^j &= A^T(j) x_k^{j-\tau+1} + C^T_{1,A(j),B(j)} u_k^j
\end{align*}
\]

(6.27a)

(6.27b)

Introducing (6.27) in (6.26) we have

\[
\left[ C^T_{C(j),A_{L,j},A_{L,j'},A_{L,j'}} + C(j',j) A^T(j) \right] x_k^{j-\tau+1}
\]

\(^3\) Variables \( x_k^{j-\tau+2} \ldots x_k^j \) do not appear since they can be expressed, via (6.14), through the former.
Let us denote with $\mathbf{F}_{j,j'}$ and $\mathbf{G}_{j,j'}$ the matrices which multiply vectors $x_{k-\tau+1}^j$ and $\bar{u}_k^j$ respectively, in (6.28). Further, we denote the fixed part of $\mathcal{Y}^{j,j'}$ by $\mathcal{Y}^{j,j'}$. This includes the fixed offset $\mathbf{E}_{\mathbf{A}(j),\mathbf{I}^j}$ which was extracted from (6.28). All these notations allow to reformulate (6.23) as follows:

$$
\mathbf{F}_{j,j'} x_{k-\tau+1}^j + \mathbf{G}_{j,j'} \bar{u}_k^j \notin \mathcal{Y}^{j,j'}(\mathcal{Z}^{j,j'}) \oplus \{-\mathcal{Y}^{j,j',*}(\mathcal{Z}^{j,j'})\},
$$

for all $j' \neq j$.

The resulting problem is non-convex: the vector of elements $\left[(x_{k-\tau+1}^j)^\top \ (\bar{u}_k^j)^\top \right]^\top$ has to reside outside of a union of convex sets. The solution proposed here is to use the construction from [153] which exploits the zonotopic nature of the sets for a compact representation of the problem. A mixed integer formulation was employed to reformulate the complementarity condition resulting from the rewriting of the bilevel problem.

### 6.5. Illustrative example

We take the example from [153], where a second-order piecewise affine system is considered:

$$
\begin{align*}
\mathbf{A}(1) &= \begin{bmatrix} 0.6 & 0.2 \\ -0.4 & -0.2 \end{bmatrix}, & \mathbf{B}(1) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & \mathbf{C}(1) &= \begin{bmatrix} 1 & 0 \end{bmatrix}, & \mathbf{s}(1) &= \begin{bmatrix} 1 \end{bmatrix} \\
\mathbf{B}_w(1) &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, & \mathbf{D}_v(1) &= \begin{bmatrix} 1 \end{bmatrix}, & \mathbf{r}(1) &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\end{align*}
$$

with the next modifications for the four faulty models ($i = 2, \ldots, 5$):

$$
\begin{align*}
\mathbf{B}(2) &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, & \mathbf{B}(3) &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, & \mathbf{A}(4) &= \begin{bmatrix} 1.2 & 0.2 \\ -0.4 & -0.2 \end{bmatrix}, & \mathbf{A}(5) &= \begin{bmatrix} 2.0 & 0.2 \\ -0.4 & -0.7 \end{bmatrix}.
\end{align*}
$$

The necessary sets, in zonotopic notation, are: $\mathbf{W} = \mathcal{Z}(0.5\mathbf{I}, \mathbf{0})$, $\mathbf{V} = \mathcal{Z}(0.2, \mathbf{0})$, the disturbance noises and $\hat{\mathbf{U}} = \mathcal{Z}(9\mathbf{I}, \mathbf{0})$ and $\hat{\mathbf{X}} = \mathcal{Z}(0.2\mathbf{I}, \mathbf{0})$, the bounds for the references.

We take the observation window $\tau = 5$ and proceed iteratively in order to reach the separation problem (6.29).

First, for each observer (6.11) we choose gain matrices $L^j$ via a discrete Riccati equation (with penalty matrices $Q = \mathbf{I}$ and $R = \mathbf{I}$). Next, using the set-recurrence (6.19), we obtain the state and output estimation error bounds in the matched case. Note that, in order to simplify the construction we over-approximate the initialization value ($\{z_{k-\tau+1}\} \rightarrow \hat{\mathbf{X}}$).

Next, to obtain bounding sets (6.20) we proceed as follows: using the results from [106] we obtain the feedback matrices $K^j$; compute the invariant set associated to the lifted tracking error $\mathbf{z}_k$ as in (6.8) and lastly, we project to the $z_k$ subspace.

We illustrate in Figure 6.2 the resulted sets for each mode of functioning. The first figure shows the $\bar{\mathcal{X}}^{jj'}$ sets obtained in the case $\tau = 5$. The second figure shows the sets $\bar{\mathcal{X}}^{jj'}$ which incorporate the former sets into their description and show that the closed-loop system is stable in closed-loop. We

---

4We fix the value $k^* = 5$ in (6.9) and obtain the corresponding scaling factors $\lambda$. 

have now the tools to compute the transitional sets (6.21). Applying the set-recurrence and taking the tracking error bounds obtained earlier we construct the fixed part of the output sets, $\tilde{Y}^{j,j'}$, as they appear in (6.29).

Computing matrices $F^{j,j'}$, $G^{j,j'}$ and putting (6.29) in the mixed-integer formalism of [153] we obtain a non-convex optimization problem whose solution is a feasible initial reference state $x_{k+1}^{j}$ and sequence $u_{k+1}^{j} \cdots u_{k}^{j}$ which ensures FDI form the point of view of the $j$-th observer.

6.6. Conclusions

We have used here a bank of finite-window observers and an artificially-induced feedback delay to simplify the set construction and guarantee the exactness of an FDI mechanism. Zonotopes have been used extensively due to their theoretical and numerical properties. Further work will be done for the implementation of a FDI-aware reference governor and for the bounding of the transitional sets sparked by a fault occurrence.
7. Hierarchical Control with Guaranteed Fault Diagnosability

This chapter presents a multiple time-scale hierarchical control approach with guarantees for active fault diagnosability (detection and isolation). Robust positive invariance notions are used to provide a set-membership test for FDI validation at the low level and sufficient constraints for guaranteeing FDI at the high level. Due to the computational burden of the involved bilinear optimization problem, the high-level reference governor operates at a slower sampling time than the controller of the low-level system. Therefore, corner cutting avoidance constraints are imposed such that the faster-sampled lower level is still guaranteed to function properly.

The chapter is based on a conference paper accepted for publication:


7.1. Preliminaries

With an increasing degree of automation in production plants of all types, the automated detection and identification of faults (FDI) is crucial for the safe operation of such systems. An important prerequisite for FDI is fault diagnosability. To guarantee fault diagnosability during the operation of the plant, model-based FDI methods may require the states of the system to avoid certain regions in the state space. Note that these sets can be non-convex even for linear systems (see, e.g., [C29]).

The task of avoiding certain sets in the state space is in its core similar to the obstacle avoidance problem for autonomous vehicles, where hierarchical approaches are often used for the generation of collision-free trajectories. In general, the top layer calculates an (optimal) trajectory based on e.g. shortest path methods like A* ([78, 111]), dynamic programming ([127]), or model predictive control (MPC) based approaches ([16, 33, 51, 93]). The major differences are where and how the actual obstacle avoidance is performed: [33] introduces time-varying constraints that represent obstacles, while [16] and [51], e.g., add additional terms (potential-field approaches) that penalize small distances to obstacles to the cost function of the top-layer MPC. [111] also uses potential fields around obstacles but employs them on the lower level. And [78] perform low-level collision avoidance using
a game-theoretic optimization.

A common assumption in obstacle avoidance is that the top layer works on a larger sampling time than the lower one(s). Regarding this assumption, [16] note that too large differences in the sampling times of the individual layers may lead to trajectories that actually intersect with the obstacles. [C29] approaches this problem by using a hyperplane arrangement scaffolding for obstacles, which leads to a mixed integer program (MIP) in the MPC. Note also that in the context of FDI, hierarchical approaches usually refer to different supervisor levels, as e.g. in [180]. [1], e.g., propose a hierarchical fault tolerant model predictive control approach in the sense that a supervisory FDI unit updates the model based on sensor data. Set points obtained in this manner are then used as references in the system’s local controllers.

In this chapter, a multiple time-scales hierarchical control approach that guarantees fault diagnosability by avoiding non-convex sets in the state space is proposed for linear systems with bounded disturbances. The resulting problem setup has similarities to the one in [153] for active FDI. However, instead of reducing the computational burden by approximating the zonotopes, this work proposes a hierarchical approach with a high-level reference governor that operates on a sampling time large enough to perform the optimal trajectory planning. On the lower level, a local controller with a higher sampling rate is then used for reference tracking. The low-level trajectories may, however, deviate from the ones on the high level. Therefore, the predicted low-level system trajectories are explicitly taken into account in the high-level optimization. This results in a bilevel optimization approach with an explicit formulation of the inner problem using KKT conditions in order to still be able to guarantee fault diagnosability.

7.2. Prerequisites

We briefly recapitulate here a standard multi-sensor plant used in a series of papers dealing with set-based fault tolerant control strategies, see, e.g., [B3], [J12]. The overall fault tolerant control (FTC) scheme is depicted in Figure 7.1 with the various components described hereinafter.
7.2.1. Multi-sensor plant dynamics

Let us consider the discrete-time (with the associated sampling time $\Delta$) multi-sensor plant dynamics:

$$x_{k+1} = Ax_k + Bu_k + Ew_k$$  \hspace{1cm} (7.1a)

$$y^i_k = C^i x_k + \eta^i_k$$  \hspace{1cm} (7.1b)

where $x_k, x_{k+1} \in \mathbb{R}^n$ are the current and successor states, $u_k \in \mathbb{R}^m$ is the input, $y^i_k \in \mathbb{R}^p$ are the outputs\(^1\) and $w_k \in W \subset \mathbb{R}^r$, $\eta^i_k \in V^i \subset \mathbb{R}^p$ are bounded process disturbance and measurement noises respectively. Matrices $A$, $B$, $C^i$ are of appropriate dimensions. Index $i$ enumerates through a finite collection of pairs $(C^i, V^i)$.

The goal is for the plant’s state (7.1) to track a reference signal $\bar{x}$ which verifies the nominal plant dynamics

$$\bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k,$$  \hspace{1cm} (7.2)

such that the tracking error $z_k \triangleq x_k - \bar{x}_k$ dynamics

$$z_{k+1} = Az_k + B(u_k - \bar{u}_k) + Ew_k,$$  \hspace{1cm} (7.3)

rest in a neighborhood of the origin (bounded by a set to be characterized later).

To each sensor we attach an observer which provides a state estimation $\hat{x}^i_k$ with dynamics

$$\hat{x}^i_{k+1} = A\hat{x}^i_k + Bu_k + L^i (y^i_k - \hat{y}^i_k),$$  \hspace{1cm} (7.4a)

$$\hat{y}^i_k = C^i \hat{x}^i_k,$$  \hspace{1cm} (7.4b)

whose estimation error $\tilde{x}^i_k \triangleq x_k - \hat{x}^i_k$ dynamics are given by\(^2\):

$$\tilde{x}^i_{k+1} = (A - L^i C^i) \tilde{x}^i_k - L^i \eta^i_k + Ew_k.$$  \hspace{1cm} (7.5)

To close the loop, we define the control law

$$u_k = \bar{u}_k + K \left( \bar{x}_{k-\tau+1} - \tilde{x}^i_{k-\tau+1} \right)$$  \hspace{1cm} (7.6)

composed from the feedforward term $\bar{u}_k$ and the delayed feedback corrective term $K \left( \bar{x}_{k-\tau+1} - \tilde{x}^i_{k-\tau+1} \right)$. Note that the latter uses delayed information (from time instant $k - \tau + 1$) based on a decision taken at the current time instant (index $i_k$). Both of these design choices will be justified by the latter implementation of the fault detection and isolation mechanism\(^3\).

Introducing (7.6) in (7.3) and noting that $\bar{x}_k - \tilde{x}^i_k = \hat{x}^i_k - z_k$ leads to

$$z_{k+1} = Az_k + BK \left( \tilde{x}^i_{k-\tau+1} - z_{k-\tau+1} \right) + Ew_k.$$  \hspace{1cm} (7.7)

\(^1\)Without any loss of generality we assume that the output dimension is the same for all sensors.

\(^2\)We assume that each pair $(A, C^i)$ is observable and hence there exists a static feedback $L^i$ which stabilizes the observer dynamics.

\(^3\)Assuming an exact FDI mechanism, the control reconfiguration reduces to selecting a state estimation (7.4a) from a pool of indices guaranteed to be healthy, as shown, e.g., in [B3].
Remark 7.1. Note that (7.7) has become a dynamic with fixed delay and that the existence of a stabilizing static feedback $K$, even if the pair $(A, B)$ is controllable, is no longer guaranteed [14].

7.2.2. Fault scenario and residual design

Hereinafter we assume total output sensor failures as fault events:

$$y_k^i = C_i x_k + \eta_k^i \text{ FAULT}, \quad y_k^i = 0 \cdot x_k + \eta_k^i$$

with $\eta_k^i F \in \mathbb{V}^{i,F} \subset \mathbb{R}^p$ denoting the under-fault measurement noise.

For the subsequent fault detection and isolation (FDI) we require the construction of a residual signal [26] which is sensitive to fault occurrences and robust to disturbances and noise. While the output estimation error $\hat{y}_k^i - y_k^i$ is a usual choice, we prefer here to construct a residual which considers a finite number of consecutive outputs (thus simultaneously avoiding the filter behavior of a Luenberger observer and the loss of information characteristic to parity equations):

$$r_k^i = y_{[k-\tau+1,k]} - \left( \tau C_i A \hat{x}_{k-\tau} + \tau C_i A B \bar{u}[k-\tau,k-1] \right)$$

with $r_k^i \in \mathbb{R}^p \tau$. The next lemma (based on a result from [J12] and on similar work from the state of the art) shows that the i-th residual (7.9) characterizes a fault occurrence in terms of known quantities.

Lemma 7.1. Assuming a persistent fault (i.e, (7.8) remains unchanged for at least $\tau$ instants of time), it follows that the i-th residual (7.9) lies in one of two possible values:

i) healthy “steady state”:

$$r_k^i H = \tau C_i A \hat{x}_{k-\tau}$$

ii) faulty “steady state”:

$$r_k^i F = \eta_{[k-\tau+1,k]} - \left( \tau C_i A \hat{x}_{k-\tau} + \tau C_i A B \bar{u}[k-\tau,k-1] \right)$$

Proof. Residual (7.9) is in fact a difference between $\tau$ consecutive sensor outputs $(y_{[k-\tau+1,k]}^i)$ and $\tau$ consecutive reference outputs $(\bar{y}_{[k-\tau+1,k]}^i) \triangleq \tau C_i A \hat{x}_{k-\tau} + \tau C_i A B \bar{u}[k-\tau,k-1])$. Expressing the former in the form of the latter we have in the healthy case that $y_{[k-\tau+1,k]}^i = \tau C_i A \hat{x}_{k-\tau} + \tau C_i A B \bar{u}[k-\tau,k-1] + \tau C_i A E \bar{w}[k-\tau,k-1] + \eta_{[k-\tau+1,k]}^i$. Conversely, if the i-th sensor was under fault for the last $\tau$ instants, the right hand side of (7.8) applies and we have that $y_{[k-\tau+1,k]}^i = \eta_{[k-\tau+1,k]}^i$. Further using (7.6) we have that $u[k-\tau,k-1] = \bar{u}[k-\tau,k-1] + \text{diag}(K_{\tau}) \left( \hat{x}_{[k-2\tau+1,k-1]} - \hat{x}_{[k-2\tau+1,k-1]} \right)$. Combining all these elements we reach (7.10) and (7.11) respectively, thus concluding the proof.

A couple of remarks are in order.

Remark 7.2. While at first glance, at time instant $k$, $y_{k+1}^i$ is available for measurement and subsequent analysis, in fact this assumes that $u_k$ has already been ‘decided’ and applied to the plant dynamics.
This negates the scope of the analysis, which is to decide on whether a certain state estimation is of use for the current control action design.

**Remark 7.3.** Whenever \( \tau \), the length of the observation window, is large enough such that \( \tau C_iA \) is full-rank there is no loss of information by multiplying with its inverse (an issue when using parity equations as residuals).

### 7.2.3. Set characterizations

Assuming bounded disturbances and noises leads to bounded estimation error, tracking error and residuals. To alleviate the computational burden we consider the notion of invariant sets associated to a dynamic [23].

**Definition 7.1.** A set \( \Omega \subset \mathbb{R}^n \) is called a robust positively invariant set for dynamics \( x^+ = \Gamma x + \Psi \delta \) with \( \Gamma \subset \mathbb{R}^{n \times n} \), \( \Psi \subset \mathbb{R}^{n \times m} \) and \( \delta \in \Delta \subset \mathbb{R}^m \) iff \( \Gamma \Omega \cup \Psi \Delta \subset \Omega \) holds.

Applying Definition 7.1 to (7.5) for \( 7! A \)

\[
\begin{bmatrix}
A & 0 & \ldots & 0 & -BK \\
I & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & I & 0
\end{bmatrix},
\begin{bmatrix}
BK & E \\
0 & 0 \\
\vdots & \vdots \\
0 & 0
\end{bmatrix},
\]

lead to the invariant set \( \tilde{S}_i \) (i.e, \( \tilde{x}_k^\tau \in \tilde{S}_i \) implies that \( \tilde{x}_{k+\ell}^\tau \in \tilde{S}_i, \forall \ell \geq 0 \)).

Next, the lifted tracking error dynamics of (7.7) with state variable \( z_{[k-\tau+1,k]} \) where

\[
\Gamma \mapsto \begin{bmatrix}
A & 0 & \ldots & 0 & -BK \\
I & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & I & 0
\end{bmatrix}, \quad \Psi \mapsto \begin{bmatrix}
BK & E \\
0 & 0 \\
\vdots & \vdots \\
0 & 0
\end{bmatrix}, \quad (7.12)
\]

lead to the invariant set \( S_z^\tau \) (i.e, \( z_{[k-\tau+1,k]} \in S_z^\tau \) implies that \( z_{[k-\tau+\ell+1,k+\ell]} \in S_z^\tau, \forall \ell \geq 0 \)).

Since \( z_{k-\tau} \) appears in (7.10) we also consider the auxiliary bounding set \( S_z^{\tau,*} \), defined as

\[
S_z^{\tau,*} = \text{ConvexHull} \left( \bigcup_{\ell=0, \ldots, \tau-1} \text{Proj} S_z^\tau |_{z_\ell} \right), \quad (7.13)
\]

Construction (7.13) is based on the fact that whenever \( z_{[k-\tau+1,k]} \in S_z^\tau \) holds we have that \( z_{k-\ell} \in S_z^{\tau,*}, \forall \ell = 0 \ldots \tau - 1 \) also holds.

**Corollary 7.1.** Assuming that the state estimations and the tracking error lie inside their bounding sets (\( \tilde{x}_k^\tau \in \tilde{S}_i, z_{[k-\tau+1,k]} \in S_z^\tau \)) allows to characterize the healthy and faulty residual sets:

i) healthy “steady state”:

\[
R_k^{i,H} = V^i,\tau| + \frac{\tau}{C_i,A}\left( S_z^{\tau,*} \right) \oplus \frac{\tau}{C_i,A,B,K}\tilde{S}_i^\tau \oplus \frac{\tau}{C_i,A,B,K}\left( -S_z^\tau \right) \oplus \frac{\tau}{C_i,A,E}\left( W^i,\tau \right), \quad (7.14)
\]

ii) faulty “steady state”:

\[
R_k^{i,F} = V^i,F,\tau| \oplus -\frac{\tau}{C_i,A}\tilde{x}_{k-\tau} - \frac{\tau}{C_i,A,B}\tilde{u}_{[k-\tau,k-1]}, \quad (7.15)
\]

**Proof.** The proof is straightforward in the sense that (7.14)–(7.15) mirror (7.10)–(7.11) and that all variables are either known (the references \( \tilde{x}_{k-\tau} \), \( \tilde{u}_{[k-\tau,k-1]} \)) or are bounded (measurement and process noises and estimation error and tracking error).
7.3. Hierarchical reference governor with active FDI

With the previous section’s notation it follows that separation condition

\[ R_i^i H \cap R_k^i F = \emptyset, \forall i \]  

(7.16)

is a sufficient condition for exact FDI: since the residual \( r_i^k \) has to lie in either \( R_i^i H \) or \( R_k^i F \) (after at least \( \tau \) time instants have passed since the latest switch in (7.8)), (7.16) means that the set membership exclusion \( r_i^k \notin R_i^i H \) is an unambiguous FDI signal.

Further noticing that (7.15) is parametrized after state and input references and that \( R_i^i H \) is time invariant allows to reformulate (7.16) as

\[ \tau \left[C_i, A\bar{x}_{k-\tau} + \tau C_i, A,B\bar{u}_{[k-\tau,k-1]} \right] \notin \left\{ -V^{i,F,[\tau]} \right\} \oplus R^i H. \]  

(7.17)

(7.17) serves as constraint in a constrained optimization problem in order to enforce FDI (thus, an active FDI scheme).

7.3.1. Hierarchical active FDI validation

While (7.17) is a popular formulation, it is often inefficiently implemented. In here, we propose a hierarchical approach where we separate between the computation of the reference values (higher level) and the FDI validation (lower level). This is justified both by the difficulties of solving (7.17), a problem with non-convex constraints, and by the fact that a reference should not be updated more often than it is necessary.

Hence, we propose to consider at the higher level a larger sampling time (to which corresponds a slower update of the reference values) \( \bar{\Delta} = N\Delta \). To this sampling time correspond reference dynamics:

\[ \bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k, \]  

(7.18)

where the “breve” modifier signifies that the variables and associated matrices correspond to the higher-level sampling time \( \bar{\Delta} \). That is, \( \bar{x}_k \) corresponds to \( x_{kN} \), \( \bar{A} = A^N \) and \( \bar{B} = \sum_{s=0}^{N-1} A^sB \).

Note that updating the input at \( \bar{\Delta} = N\Delta \) time instants constrains the \( \Delta \)-sampled references (7.2) as follows:

\[ x_{k+\ell} = A^{\bar{k}}\bar{x}_{\ell} + \sum_{s=0}^{\ell-1} A^sB\bar{u}_{\ell}, \]  

(7.19a)

\[ \bar{u}_{k+\ell} = \bar{u}_{\ell} \]  

(7.19b)

for \( \ell = 0, \ldots N - 1 \) and \( \bar{k} = \lfloor k/N \rfloor \).

While (7.18) and (7.19) allow to express (7.17) in the \( \bar{\Delta} \)-sampling representation regardless of the relation between the FDI-induced delay \( \tau \) and the sampling multiplier \( N \), it is less cumbersome if we take the latter as greater than the former\(^5\): \( N > \tau \). Under this assumption the sequence \( \bar{u}_{[k-\tau,k-1]} \)

---

\(^4\)To emphasize that the right-hand of (7.16) is time-invariant we used \( R^i H \) instead of \( R_i^i H \).

\(^5\)This is a reasonable simplification since both \( \tau \) and \( N \) are parameters selected by the supervisor.
remains constant (see (7.19b)) and (7.17) becomes\(^6\)

\[
\tau_{\tilde{C}_i,A} \tilde{x}_k + \tau_{\tilde{C}_i,A,B} \tilde{u}_k \notin \left\{ -V^{i,F_{\lceil \tau \rceil}} \right\} \oplus R^{i,H}, \quad (7.20)
\]

where

\[
\begin{align*}
\tau_{\tilde{C}_i,A} &= \tau_{\tilde{C}_i,A}, \quad (7.21a) \\
\tau_{\tilde{C}_i,A,B} &= \tau_{\tilde{C}_i,A,B} \cdot \begin{bmatrix} I & \ldots & I \end{bmatrix}^\top. \quad (7.21b)
\end{align*}
\]

We can now write the top-level reference governor:

\[
\hat{u}_{[k,k+N_p-1]} = \arg \min_{\hat{u}_{[k,k+N_p-1]}} \sum_{\ell=1}^{N_p} \mathcal{C} \left( \tilde{x}_{k+\ell+1}, \tilde{u}_{k+\ell} \right) \quad (7.22a)
\]

\[
s.t. \quad \tilde{x}_{k+\ell+1} \in \tilde{X}, \tilde{u}_{k+\ell} \in \tilde{U}, \quad (7.22b)
\]

\[
(7.18),(7.20) \text{ hold for } k \leftarrow k + \ell, \ell = 0 \ldots N_p - 1 \quad (7.22c)
\]

where \(N_p\) is the length of the prediction horizon; \(\mathcal{C}(\cdot, \cdot) : \mathbb{R}^{n \times p} \to \mathbb{R}\) is a, usually quadratic, scalar cost involving the reference states and inputs; \(\tilde{X} \subset \mathbb{R}^n, \tilde{U} \subset \mathbb{R}^p\) are bounding sets for the reference state and input.

### 7.3.2. Guaranteed fault diagnosability for intra-sample behavior

Enforcing (7.20) in (7.22) reduces to (7.17) whenever \(k - \tau\) is a multiple of \(\tilde{k}\). In the rest of the cases, the validation of (7.17) is no longer guaranteed (from the point of view of (7.18)-(7.20) this is an intra-sample behavior and thus completely ignored). In the motion planning parlance, this is called cutting the corner: while apparently a certain constraint is respected at each sampling time, its actual behavior violates the constraints (i.e., “cuts the corner of the obstacle”). Within the chapter’s framework the corner cutting translates to the FDI no longer being guaranteed to be exact. This is a major drawback as FDI decisions are required at each \(\Delta\) sampling time in order to update the control action (7.6).

Therefore, we propose here a reformulation of constraint (7.20) such that (7.17) is guaranteed to hold under restrictions (7.19) at each sampling time \(\Delta\). Using the same reasoning as in [C29], [144] we modify the separation condition (7.22c) into:

\[
\alpha_{k+\ell} \left[ \tau_{\tilde{C}_i,A} \tilde{x}_{k+\ell} + \tau_{\tilde{C}_i,A,B} \tilde{u}_{k+\ell} \right] + \left( 1 - \alpha_{k+\ell} \right) \left[ \tau_{\tilde{C}_i,A} \tilde{x}_{k+\ell+1} + \tau_{\tilde{C}_i,A,B} \tilde{u}_{k+\ell+1} \right] \notin \left\{ -V^{i,F_{\lceil \tau \rceil}} \right\} \oplus R^{i,H}, \quad (7.23)
\]

for \(0 \leq \alpha_{k+\ell} \leq 1\) and \(\ell = 1, \ldots, N_p\).

(7.23) simply states that no point on the segment defined by the state/input combination at times \(\tilde{k} + \ell, \tilde{k} + \ell + 1\) should intersect the region where FDI in undecidable. In other words, no such

\(^6\)Note that we switched to index \(\tilde{k}\) since the delay \(\tau\) is no longer relevant and index \(k\) corresponds to the \(\Delta\)-sampled dynamics.
point should sit in the “shadow” made by the current point with the obstacle. Assuming that the intermediary points (7.19) do not deviate from this segment, this means that (7.17) holds at each sampling time $\Delta$.

To proceed further we consider the half-space representation of \( \{ -V^i, F^i(t) \} \oplus R^i \subset \{ \zeta \in \mathbb{R}^n : H_i^\top \zeta \leq h_i \} \), with \( H^i \in \mathbb{R}^{N_i \times n} \), \( h^i \in \mathbb{R}^{N_i \times 1} \) and denote for compactness with \( \xi_{k+\ell}, \xi_{k+\ell+1} \) the segment’s end-points from (7.23) which becomes, with these notations:

\[
H^i \left[ \alpha_i^{k+\ell} \xi_{k+\ell} + (1 - \alpha_i^{k+\ell}) \xi_{k+\ell+1} \right] \preceq h^i. \tag{7.24}
\]

It is known (see, e.g., [153]) that the exclusion constraint (7.24) is equivalent with the feasibility test of an optimization problem:

\[
\lambda_i^{k+\ell} = \min_{\alpha_i^{k+\ell} \leq \lambda_i^{k+\ell}} \lambda_i^{k+\ell} \tag{7.25a}
\]

s.t.

\[
1 \leq \lambda_i^{k+\ell}, \tag{7.25b}
0 \leq \alpha_i^{k+\ell} \leq 1, \tag{7.25c}
H^i \left[ \alpha_i^{k+\ell} \xi_{k+\ell} + (1 - \alpha_i^{k+\ell}) \xi_{k+\ell+1} \right] \leq \lambda_i^{k+\ell} h^i. \tag{7.25d}
\]

If (7.25) is feasible it means that the line segment has intersected the obstacle for a supraunitary scaling factor, i.e., the obstacle does not actually intersect the segment.

Replacing (7.22c) with (7.25) in (7.22) leads to a bilevel optimization problem: at each step in the optimization procedure the outer problem solves \( N_p \) inner problems. that is, (7.22c) is replaced by

\[
\lambda_i^{k+\ell} \geq 1 \tag{7.26a}
\]

where \( \lambda_i^{k+\ell} \) are solutions to (7.25) \( (7.26b) \)

The complexity induced by the bi-level formulation can be avoided by writing the inner problem in its Karash-Kuhn-Tucker (KKT) form:

\[
\begin{bmatrix}
1 \\
0 \\
-1 \\
-1
\end{bmatrix}
+ \begin{bmatrix}
0 & -1 & 1 & (H^i(\xi_{k+\ell+1} - \xi_{k+\ell}))^\top \\
0 & -1 & 0 & (h^i)^\top
\end{bmatrix}
\mu_{k+\ell}^i = 0, \tag{7.27a}
\]

\[
\mu_{k+\ell}^i \geq 0, \quad g(\xi, \lambda, \alpha) \leq 0, \tag{7.27b}
\]

\[
\mu_{k+\ell}^i \times g(\xi, \lambda, \alpha) = 0. \tag{7.27c}
\]

with \( g(\xi, \lambda, \alpha) \) the shorthand notation for inequalities (7.25b)–(7.25d), \( \mu_{k+\ell}^i \) the associated vector of Lagrangian multipliers and “\( \times \)” the elementwise complementarity operator. The vector in (7.27a) is the gradient of the cost (7.25b), i.e., \( \nabla(\lambda_i^{k+\ell}) \) and the matrix is the gradient of the inequalities (7.25b)–(7.25d), i.e., \( \nabla g(\xi, \lambda, \alpha) \).

Replacing (7.22c) with (7.27) leads to a single-level problem but at the price of introducing nonlinearities (foremost, the complementarity constraint (7.27c) but also the bilinear terms \( \alpha_i^{k+\ell} \xi_{k+\ell}, \alpha_i^{k+\ell} \xi_{k+\ell+1} \) appearing in (7.27b)). While further processing is possible (e.g., relaxing the complementarity constraint into a mixed-integer formulation), we prefer here to leave it as a nonlinear optimization problem which will be further handled by specialized solvers.
Remark 7.4. An important aspect usually neglected in the literature is that the trajectory has a curvature: the intra-sample points (computed at each $\Delta$ time instant) do not actually lie on the segment defined by the consecutive points taken at $\hat{\Delta}$ time instants. In the following (since $N$ is taken comparable with $\tau$ and thus, of small value), we simplify and only consider the line-segments. A more complete approach would be to compute the maximum curvature and enlarge with it the sets appearing in (7.17). Note that in this case the problem comes back to a “straight lines” formulation but for larger sets.

7.4. Illustrative example

For illustration purposes consider the system

$$x_{k+1} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0.5 \\ 1 & 1 \end{bmatrix} u_k + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} w_k$$ (7.28)

with output matrices $C^{(1,2,3)}$:

$$C^1 = \begin{bmatrix} 1.5 & 0 \end{bmatrix}, \quad C^2 = \begin{bmatrix} 1 & -1 \end{bmatrix}, \quad C^3 = \begin{bmatrix} 1.5 & 1 \end{bmatrix}.$$ (7.29)

The process and measurement bounding sets are

$$W = \{ w : |w| \leq 0.1 \},$$ (7.30a)

$$V^i = \{ \eta^i : |\eta^i| \leq 0.1 \}, V^{i,F} = \{ \eta^{i,F} : |\eta^{i,F}| \leq 0.1 \},$$ (7.30b)

for $i \in \{1, 2, 3\}$.

For each sensor we construct an observer as in (7.4) with the gain matrices $L^i$ being the result of a pole placement procedure (with poles assigned between 0.9 and 0.95):

$$L^1 = \begin{bmatrix} 0.100 \\ 0.033 \end{bmatrix}, \quad L^2 = \begin{bmatrix} 0.148 \\ 0.033 \end{bmatrix}, \quad L^3 = \begin{bmatrix} 0.078 \\ 0.033 \end{bmatrix}.$$ (7.31)

Further, we take the feedback delay appearing in (7.6) as $\tau = 2$ (which is enough to guarantee the invertibility of matrices $\Theta^i_{C^i,A}$ appearing in (7.10)–(7.11)). The static feedback is computed as in [14] by representing (7.7) in extended form (in which case the static feedback with delayed information of (7.7) becomes an output static feedback):

$$K = \begin{bmatrix} -0.15 & -0.15 \\ 0.15 & 0.20 \end{bmatrix}.$$ (7.32)

We apply [137] to compute the RPI sets for the state estimation errors ($\tilde{S}^i$) and for the extended tracking error ($S^i_z$). These are further used to construct the bounding set $S^i_z$ as in (7.13), the healthy and faulty residual sets (7.14)–(7.15) and, lastly, the right-hand region appearing in the FDI condition (7.17).

For illustration purposes we depict in Figure 7.2a the sets $\tilde{S}^i$ (solid blue) together with the smallest
box enclosing their union (dashed red), used latter in the construction of \( S^*_z \). Figure 7.2b illustrates the projections of \( S^*_z \) (along each of the \( z \) components of the extended state \( \bar{z}[k-r+1:k] \), solid blue) together with their bounding set, \( S^*_z \) (dashed red).

![Figure 7.2a: State estimation error sets](image)

![Figure 7.2b: Tracking error sets](image)

Figure 7.2.: RPI sets associated with the plant and observer dynamics.

To highlight the analysis carried earlier we consider a sample multiplier \( N = 2 \) and the reference governor (7.22) with prediction horizon \( N_p = 3 \), input and state bounds (\( \bar{U} = \{ \bar{u} : |\bar{u}| \leq 10 \} \), \( \bar{X} = \{ \bar{x} : |\bar{x}| \leq 50 \} \)) and weight matrices \( Q = R = I \).

To illustrate the result we provide an ideal reference to be tracked (points sampled along the circumference of a circle of radius \( r = 25 \)). In Figure 7.3 we depict both the reference (green line with square markers) and the left-hand side of (7.17), the sequence \( \bar{x}_k + (C^t A)^{-1} C^t A B \bar{e}_k \) (blue line with bullet markers). As expected, the constraints (7.27) ensure the corner cutting avoidance constraint:

![Figure 7.3: Illustration of reference governor implementation with active FDI validation. The blue areas depict regions in the state space where FDI is not decidable.](image)

no segment formed by a pair of consecutive points cuts the obstacles.

Using (7.19a) we observe in Figure 7.4 that the the inter-sample reference states lie close to the
straight-lines defined by the $\Delta$-sampled reference states (for the chosen $N = 2$, close enough to validate the assumption that the inter-samples are stringed along the segments).

Figure 7.4: Illustration of $\bar{\Delta}$ and $\Delta$-sampled state reference.

### 7.5. Conclusions

A hierarchical control approach with guarantees for active fault diagnosability (detection and isolation) was presented. Robust positive invariance notions were used to provide a set-membership test for FDI validation at the low level and sufficient constraints for guaranteeing FDI at the high level. The computational burden of the according bilevel optimization problem motivated the use of a larger sampling time for the high-level reference governor. Resulting corner cutting issues have been considered by explicitly taking into account predictions of the low-level system trajectories in the high-level optimization. A preliminary case study demonstrated the viability of the approach.

Future advances will consider explicitly the curvature induced by the fast-sample behavior and different dynamics for the hierarchical control scheme.
8. Dictionary learning strategies for sensor placement and leakage isolation in water networks

This chapter handles pipe leakage in water networks through a fault detection and isolation mechanism coupled with dictionary learning strategies. It considers sparse representation strategies for sensor placement and subsequent dictionary learning and classification for accurate fault detection and isolation. Various sensor placement strategies are proposed and it is shown that faults with varying magnitudes are correctly identified in a detailed emulation benchmark.

The chapter is based on book chapter [C28]:


8.1. Preliminaries

The underlying infrastructure which delivers the basic services (electricity, transport, communication, etc.) of today is becoming increasingly connected and automated. Consequently, issues which are well-understood (and manageable) for small-sized systems have become challenging or are still open for improvement in systems with structural impediments for observation, control and communication [113, 158].

A prime example is the analysis of abnormal behavior in distributed water networks [10, 126, 167]. In particular, we are interested in the fault detection and isolation (FDI) of leakages. While apparently simple (since the leakage directly manifests as a loss of pressure), a typical network may have thousands of junctions but only a limited number of sensors (expensive and hard to install) can be placed. This means that usually a sensor cannot directly measure a fault but rather its influence along the network of junctions and pipes. Compounding the problem, the network dynamics are nonlinear and thus impractical for a model-based analysis.

In light of the previous remarks it is clear that the placement of the sensors is of utmost importance
as it maximizes the chances of detecting and isolating a leakage. The usual approach in the literature is either exhaustive (where all possible combinations are tested) or heuristic. For the latter, genetic algorithms are usually employed [29, 125]. Once the sensors are placed, residual signals (usually the offset between an expected/nominal pressure and the actual measurement) are used to detect and isolate the fault [25].

[166] discuss dictionary learning (DL) for sparse representations (SR) is a relatively new and active research topic in the signal processing community with multiple applications such as segmentation, compression, detection, classification, and denoising which show good results when dealing with audio, image or video processing across varied domains like tomography, magnetic resonance imaging, facial recognition, and astronomy. Typically the DL process trains an overcomplete base (called dictionary) in which signals from a given class can be sparsely represented by using the linear combination of a few dictionary columns (also called atoms).

In this chapter we investigate the application of DL techniques for sensor placement and fault detection in water distribution systems, which to our knowledge has not been tried before.

At the core of our sensor placement strategy stands the orthogonal matching pursuit (OMP) method [122]. OMP is an almost ubiquitous greedy sparse representation algorithm in the DL field, preferred for its speed and theoretical guarantees [36, 41]. This approach allows to select the most “popular” nodes for sensor placement.

Next we approach FDI through DL with LC-KSVD ([71]) which is the classification variant of the popular K-SVD ([4]) algorithm that adds labeling and discriminative properties to the atoms.

More precisely, the faults affecting a given node represent a class and train the dictionary such that its atoms discriminate between these classes. The active atoms for a certain class (fault) are seen as a fault signature against which other residual signals will be tested. Thus the fault signature obtained permits to detect the apparition of a fault and, subsequently, to isolate it (as long as the signature is unique w.r.t. the other possible faults).

In the simulations we consider multiple magnitudes and discuss different sensor deployment strategies.

Figure 8.1.: Hanoi water network.
8.2. Problem description

A water network can be interpreted as a graph \((\mathcal{N}, \mathcal{L})\) where the nodes \(\mathcal{N} = \mathcal{N}^t \cup \mathcal{N}^j\) are either tanks or junctions (the former are flow sources, the latter distribute the existing flow to users and connecting pipes) and edges \(l_{ij} \in \mathcal{L}\) which are the pipes linking nodes (e.g., \(l_{ij}\) denotes a pipe between \(i, j \in \mathcal{N}\)). For further use we note\(^1\) \(|\mathcal{N}^j| = n\).

8.2.1. Network dynamics

From our point of view, the characteristic of a junction node \(i \in \mathcal{N}^j\) is its pressure \(p_i(t)\), proportional to the node’s “head”. This characteristic is influenced by the network structure, user demand and, not in the least, by external perturbations (e.g., junction leakages). The pressure

\[
\dot{p}_i(t) = g_{i1}(p_i(t)) + g_{i2}(p_j(t)), \quad \forall j \text{ s.t. } l_{ij} \in \mathcal{L}
\]

(8.1)
can be described analytically as a nonlinear combination of local and neighbouring influences, the associated nonlinear dynamics are impractical to use \([29]\).

An idea is to measure the pressure values against some known nominal value (usually taken in the middle of the night when user demand can be ignored) in order to detect abnormal behavior. Unfortunately, the sensors used for measuring the pressure are expensive to buy and install. Coupled with the fact that real-life networks may have thousands of nodes it becomes clear that only a limited number of junctions can be observed directly. Consequently, the main question becomes how to place this limited number of sensors such that leakage events are correctly identified as much as possible.

To answer these questions, hereinafter we make use of the EPANET software, a standard tool for water network emulation \([145]\). As a particular benchmark (also used routinely in the community) we consider the Hanoi water network \([29]\). An illustration of this network is given in Fig. 8.1. The main characteristics are as follows: one tank and 31 junction nodes linked through 34 pipes (each of them with a certain length and diameter); each junction node can be affected by a leakage (taking values from a predefined range) and some of the nodes will have sensors mounted on them (see the figure legend for a representation of each of these elements).

8.2.2. Fault scenarios

The junction node pressure can be affected by a node leakage. We interpret such an event as a fault, that is, a junction node \(i \in \mathcal{N}^j\) can be affected by a fault event

\[
f_i(t) \in \{0, 2, 3, \ldots, 33\} \text{ L/min}
\]

(8.2)

where ‘0’ stands for healthy functioning and the remaining non-zero values stand for faulty functioning with various fault magnitudes\(^2\). For further use we partition magnitudes (8.2) into three disjoint sets: \(\mathcal{F}_0\), \(\mathcal{F}_{\text{train}} = \{2, 4, \ldots, 32\}\) and \(\mathcal{F}_{\text{test}} = \{3, 5, \ldots, 33\}\). We also note \(m = |\mathcal{F}_{\text{train}}| = |\mathcal{F}_{\text{test}}|\).

A typical fault scenario consists of one (or multiple) faults affecting junction nodes either directly

---

1Hereafter, notation \(|X|\) denotes the cardinality of set \(X\), i.e., its number of elements.

2We have abused the notation and used \(f_i(t)\) both as a fault signal and as a magnitude measurement.
or through propagation. Hence, (8.1) becomes:

$$\dot{p}_i(t) = g_{i1}(p_i(t), f_i(t)) + g_{i2}(p_j(t), f_j(t)), \forall j \text{ s.t. } l_{ij} \in \mathcal{D},$$

(8.3)

where $f_i(t)$ affects the current node and the terms $p_j(f_j)$ gather the influences of the other nodes (healthy or faulty), as they are transmitted through the network’s pipes.

To detect and isolate a fault we require residual signals ([25]) which emphasize the faults’ effect and minimize other influences. Hence, for a junction node $n_i$ we consider

$$r_i(t) = p_i(t) - \bar{p}_i$$

(8.4)

where $\bar{p}_i$ corresponds to a stationary pressure value, measured when the network is under healthy functioning and the external disturbances are reduced to a minimum. For further use we define $p(t) = \left[ \ldots \ p_i(t) \ \ldots \right]^T \in \mathbb{R}^n$, $\bar{p} = \left[ \ldots \ \bar{p}_i \ \ldots \right]^T \in \mathbb{R}^n$ and $r(t) = \left[ \ldots \ r(t) \ \ldots \right]^T \in \mathbb{R}^n$.

![Figure 8.2: Fault detection and isolation via dictionary classification with sensor placement in the Hanoi water network.](image)

### 8.3. Main idea

Since a model-based approach is impractical (due to the network size and its underlying nonlinear dynamics) we consider here a data-based approach. We use Fig. 8.2 to illustrate the reasoning.

In the three stacked sub-plots of Fig. 8.2 we have illustrated the residuals (8.4) for each of the 31 junction nodes for three fault scenarios: leakage in nodes 5, 12 and 27 (not happening simultaneously), each of them for fault magnitudes $f_i \in \mathcal{F}_{\text{train}}$. As expected, the residual values are largest in the node directly affected by the fault (red markers) and increase proportionally with the fault magnitude.

Clearly, if each node would have a sensor, the leakage localization would be easily decided. Assuming instead a limited number of sensors (e.g., $s = 4$), the question becomes which is the best node placement such that as many as possible faults are detected and isolated\(^3\). The sensor locations illustrated in the figure are $\{9, 12, 15$ and 30\} and are obtained as the result of selecting the most “popular” locations

\(^3\)We are interested in isolating the fault-affected node, not about the magnitude of the fault since, once the leakage is localized, the repair team will in any case assess the magnitude.
Sets in control. Applications to fault tolerant control and motion planning.

with a classification procedure. These represent the measured subset of residuals and will be the only information further used.

The solution, as detailed in the rest of the chapter, is to construct a dictionary of atoms (255 in this example) over which we project the measured residual values with a sparsity constraint. That is, each of the measured residuals is expressed as a combination of a limited number of atoms (i.e., active atoms). These active combinations are in fact a fault signature which uniquely characterizes a certain fault (and its magnitude). This is graphically depicted in the right-most plot of Fig. 8.2: the blue solid markers denote all atoms which are active for all possible fault magnitudes for a given node. As it can be seen, there are clear differences between the subsets of active atoms.

If we assume particular fault magnitudes, e.g, $f_5 = 11$, $f_{12} = 29$ and $f_{27} = 17$ L/min (black lines in the three stacked left-plots) the uniqueness of the active atoms becomes clear (the hollow black circles in the right-most plot).

8.3.1. Dictionary learning elements

In our study we use SR for sensor placement. Let $y$ be an observed dense signal and $A$ an overcomplete feature base, that is known and fixed, such that

$$
\begin{align*}
\text{minimize} & \quad y = Ax \\
\text{subject to} & \quad \|x\|_0 \leq s,
\end{align*}
$$

where $\|\cdot\|_0$ is the $\ell_0$ pseudo-norm counting the non-zero elements of $x$ and $s$ is the sparsity constraint. To solve the sparse representation problem in (8.5) we need to find the support $\mathcal{J}_s$ consisting of at most $s$ columns from $A$ and the corresponding coefficients.

Dictionary learning improves SR by learning the base $A$ from (8.5) for specific signal classes. Starting from a large set of training data $Y \in \mathbb{R}^{p \times m}$, the DL problem tries to find an overcomplete base that is commonly denoted by $D \in \mathbb{R}^{p \times n}$, with $p < n$, called dictionary, that can sparsely represent each data-item by using just a few columns (also called atoms) from the dictionary. Given the sparsity constraint $s$, we formally express this as the optimization problem

$$
\begin{align*}
\text{minimize}_{D,X} & \quad \|Y - DX\|_F^2 \\
\text{subject to} & \quad \|X_i\|_0 \leq s, \ i = 1 : m \\
& \quad \|D_j\|_2 = 1, \ j = 1 : n,
\end{align*}
$$

where $\|\cdot\|_F$ is the Frobenius norm, $X \in \mathbb{R}^{n \times m}$ are the resulting sparse representations of the signals $Y$. We normalize all atoms $D_j$ to avoid the multiplication ambiguity resulting from the $DX$ factorization.

We can adapt the dictionary $D$ to be better suited for classification by specializing subgroups of atoms to act as labels for a particular class (i.e., for a particular fault, regardless of its magnitude). This is useful when performing fault detection for water network systems, but also for analyzing various sensor placement scenarios. To that end, LC-KSVD ([71]) obtains such discriminative dictionaries by
regularizing the objective in eq. (8.6):

\[
\begin{align*}
\text{minimize}_{D,X,A,W} & \quad \|Y - DX\|_F^2 + \alpha \|Q - AX\|_F^2 + \\
& \quad + \beta \|H - WX\|_F^2 \\
\text{subject to} & \quad \|X_i\|_0 \leq s, \quad i = 1 : m \\
& \quad \|D_j\|_2 = 1, \quad j = 1 : n,
\end{align*}
\]  

(8.7)

where the first penalty term adds consistent labeling properties to the sparse entries of each sparse code \(X_i\) and the second reduces classification error by learning the linear operator \(W\). Each column \(i\) of the binary matrix \(Q\) has non-zero entries in the places where the input signal \(Y_i\) and the atom \(D_i\) share the same label. Thus matrix \(A\) is a linear transformation encouraging discrimination in the sparse representations \(X\). For minimizing the classification error, \(H\) is built with its columns as standard basis vectors \(e_i\) where \(i\) is the class label of the corresponding training signal from \(Y\). The resulting matrix \(W\) represents the classifier parameters.

8.3.2. Sensor placement algorithm

OMP iteratively solves (8.5) by selecting at each step the atom that correlates the most with the current residual and then projects the signal to the span of the selected atoms. In the following OMP will be used for sensor placement analysis.

In our water network, given a network of \(n\) nodes, we want to isolate the fault as best as possible with a few given sensors that we conveniently denote with \(s\).

Recalling the definition of (8.4) and the related terms, taking the \(m\) fault magnitudes from \(J_{\text{train}}\), we organize the residuals as columns inside the residue matrix \(R \in \mathbb{R}^{n \times nm}\)

\[
R = \begin{bmatrix}
    r_{1,1} & r_{1,2} & \cdots & r_{1,m} \\
    \vdots  & \vdots  & \ddots & \vdots  \\
    r_{n,1} & r_{n,2} & \cdots & r_{n,m}
\end{bmatrix}.
\]  

(8.8)

The matrix is divided into \(n\) blocks, in which each column represents the residuals for each of the \(m\) fault magnitudes (i.e., \(r_{i,j}\) is the residual associated with a fault of magnitude ‘\(j\)’ affecting the \(i\)-th node).

Recalling Fig.8.2, \(R\) reflects the network topology with most pressure changes taking place in the affected node and its closest neighbours, while distant nodes have small or insignificant fluctuations. Thus the significant entries, or support, of each column from \(R\) are almost the same within one node block and their value varies based on the fault magnitude.

The simulated data in \(R\) was collected as if we had a pressure sensor in every network node. When dealing with less ideal scenarios \(s \neq n\), for each column \(r\) of \(R\) we use OMP to find its vector approximation \(x\) by reformulating (8.5) as

\[
\begin{align*}
\text{minimize}_x & \quad \|r - I_n x\|_2^2 \\
\text{subject to} & \quad \|x\|_0 \leq s.
\end{align*}
\]  

(8.9)

Using the standard basis as the fixed dictionary comes naturally: when \(s = n\) we get pressure data from each node \(x = r\) and, when \(s < n\), OMP selects a subset of unit vectors from \(I_n\) that model \(r\).
best, which is equivalent to performing sensor placement for this specific data-item. The end result is a matrix $X \in \mathbb{R}^{n \times nm}$ whose columns use at most $s$ atoms from $I_n$ to approximate the corresponding vectors from $R$.

Graphically, the result of the iterative application of (8.9) leads to Fig. 8.3. The i-th row corresponds to the i-th atom (or in other words “sensor placed at the i-th position”) and the number of non-zero entries tells how many times it has been used in the description of the residuals from (8.8). Conversely, the j-th column puts in evidence the $s = 4$ most significant atoms which approximate the corresponding residual signal from (8.8).

**Figure 8.3:** Dictionary classification for sensor placement (the case $s = 4$).

As seen in the figure, each residual “proposes” a different combination of sensor placements. The decision on how to select overall the “best” $s$ sensors can be taken in multiple ways. In here we considered:

1. Count how many times each atom appears and select the first ‘$s$’ most common;

2. From each block of $m$ columns (corresponding to the $m$ different magnitudes affecting a given node) take the most ‘$s$’ frequently used atom; of the ‘$n \cdot s$’ atoms, select again the first ‘$s$’.

Applying these selection procedures to the example from Fig. 8.3 leads in case 1 to sensor placement in nodes {15, 12, 30, 9} and in case 2 to {22, 26, 12, 31}. Fig. 8.4 illustrates the selected nodes (filled circle denotes case (1) and hollow circle denotes case (2)). We observe that repeating for various number of sensors ($s = 2 : 10$) there is significant variation in the node selection with respect to the number of sensors (and between the two selection procedures considered).

**Figure 8.4:** Node placement selection for different number of sensors.
8.3.3. Fault detection and isolation implementation

The fault detection and isolation step requires again the tools of Section 8.3.1. This time we obtain simultaneously both the dictionary and the indices of its active atoms for a given measured residual as in (8.7). For training we use the measured residual matrix (the rows from (8.8) which correspond to sensor placements, as obtained in Section 8.3.2). This will lead, as the result of optimization problem (8.7), to a dictionary which discriminates between 31 classes (one per each fault).

For illustration purposes we use parameters $\alpha = 4$, $\beta = 2$, dictionary size $n = 248$ and number of active atoms is 30 per measured residual. In Fig 8.5 we show the active (solid circle) atoms for each residual vector. In the detail plot we can see the active atoms for the faults corresponding to node 17. As it can be seen, the active atoms remain largely unchanged (regardless of the fault magnitude) and the resulting support is unique with respect to the other faults (i.e., the combination of atoms $\{8, 29, 36, 38, 41, 113, 179\}$ uniquely characterizes the fault in node 17).

![Figure 8.5: Active atoms for residual vectors in the Hanoi network (with detail for the fault 17).](image)

With the fault signatures thus obtained it is now possible to test whether different residual signals can be correctly classified (i.e., if the fault affecting them is correctly isolated). To do so, and similarly with (8.8), we construct residuals (8.4) but this time for fault magnitudes taken from $J_{test}$. We depict in Fig. 8.6 the projection of the residual corresponding to fault $f_{17} = 17$ and observe that the signature of active atoms closely resembles the signature obtained for fault $f_{17}$ in Fig. 8.5. That is, the active atoms are $\{29, 38, 113$ and 179\} with weights $\{-0.128, 0.0027, 0.0013, 3.5619\}$ (in the figure we scaled $x \mapsto \log |x|$ to better highlight the weights of the active atoms) which are 4 of the 7 atoms characterizing the class of fault $f_{17}$. Therefore we can conclude that the test residual indeed corresponds to this fault.

90 of 160
Up to now we have illustrated the fault detection and isolation for a specific number of sensors \((s = 4)\). We are interested to test the performance of the training and subsequent classification for a varying number of sensors and placement strategies. To this end, we consider the data from Fig. 8.4 and obtain the results shown in Table 8.1, where the number of sensors varies \((s \in \{2 \ldots 10\})\) and the percentage of successful FDI is shown for each of the two sensor placement strategies described in Section 8.3.2.

<table>
<thead>
<tr>
<th>s</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>73.0</td>
<td>88.3</td>
<td>85.9</td>
<td>88.7</td>
<td>80.4</td>
<td>85.9</td>
<td>89.3</td>
<td>87.7</td>
<td>80.0</td>
</tr>
<tr>
<td>(2)</td>
<td>57.9</td>
<td>86.1</td>
<td>86.3</td>
<td>83.3</td>
<td>86.5</td>
<td>89.5</td>
<td>88.9</td>
<td>90.9</td>
<td>89.5</td>
</tr>
</tbody>
</table>

Table 8.1.: Successful FDI percentage for sensor placement cases 1 and 2.

We proceed with an overview of the steps required for the initial network setup and for performing online fault detection and isolation.

The first part is described by Algorithm 1 which handles sensor placement and dictionary learning for fault detection and isolation. Step 1 uses one of the two strategies for sensor placement, described earlier, to obtain the set of nodes \(\mathcal{I}_s\) where the given \(s\) sensors will be mounted. We then construct the training data \(R_{I_s}\) by eliminating the rows from the residual matrix \(R\) corresponding to the nodes without sensors. Step 2 is required so that \(R_{I_s}\) mimics the data we will receive from the sensors when the water system is running. Step 3 trains the dictionary \(D\) and the linear classifier \(W\) based on the residues \(R_{I_s}\), using parameters \(\alpha\) and \(\beta\) for tweaking the DL regularization as described around eq. (8.7).

**Algorithm 1:** Placement and FDI learning

<table>
<thead>
<tr>
<th>Data:</th>
<th>training residuals (R \in \mathbb{R}^{n \times nm}) parameters (s, \alpha, \beta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result:</td>
<td>dictionary (D), classifier (W), sensor nodes (\mathcal{I}_s)</td>
</tr>
<tr>
<td>1</td>
<td>Select (s) sensor nodes (\mathcal{I}_s) based on matrix (R) as in Section 8.3.2;</td>
</tr>
<tr>
<td>2</td>
<td>Let (R_{I_s}) be the restriction of (R) to the rows in (\mathcal{I}_s);</td>
</tr>
<tr>
<td>3</td>
<td>Use (R_{I_s}), (\alpha) and (\beta) to learn (D) and (W) from (8.7);</td>
</tr>
</tbody>
</table>

Algorithm 2 is used for fault detection and isolation. Given \(mn\) residuals obtained with (8.4) from the \(s\) placed sensors, we start by computing the discriminative sparse representations \(X_k\) using the learned dictionary \(D\) for each residue \(R_k\) (step 2). Step 3 applies the linear classifier \(W\) and produces
the label vector $L_k \in \mathbb{R}^n$. The position $c$ of its largest entry corresponds to the predicted class (or node) where the fault took place (Step 3).

Algorithm 2: Fault detection and isolation

<table>
<thead>
<tr>
<th>Data:</th>
<th>testing residuals $R \in \mathbb{R}^{s \times mn}$</th>
<th>dictionary $D$, classifier $W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result:</td>
<td>prediction $P \in \mathbb{N}^K$</td>
<td></td>
</tr>
</tbody>
</table>

1. for $k = 1$ to $mn$ do
2. Use OMP in (8.5) to obtain $X_k$ using $R_k$ and $D$ ;
3. Label: $L_k = WX_k$ ;
4. Classify: $P_k = \arg\max_c L_k$ ;
5. end

8.4. Additional results

While the previous scheme has given satisfactorily results we are interested in additional use cases. Some of them are discussed below.

8.4.1. Network partitioning

From a practical point of view (i.e., repairing a detected leakage) it makes sense to divide the network into parts combining adjacent nodes. Taking into account the length of the pipes we consider then the partitioning (also depicted in Fig. 8.1, dashed contours):

$$\{1, 2, 3, 4, 16, 17, 18\}, \{5, 6, 7, 8, 13, 14, 15\}, \{9, 10, 11, 12\},$$
$$\{19, 20, 21, 22\}, \{23, 24, 25, 26, 27, 28, 29, 30, 31\}.$$ 

In this case, the fault is associated with the nodes of a partition (hence we have 5 composite faults, one per each partition). Implementation-wise this means that the dictionary from (8.7) is not trained to discern between the nodes of a partition (since they are of the same class). Note that the dictionary training takes into account the partitioning, it is not simply a grouping of nodes to cover observed wrong FDI labeling (i.e., the dictionary atoms are re-computed and the number of classes is changed from 31 to 5). The result of applying the procedure\(^4\) as in Table 8.1 leads to the results of Table 8.2:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>96.2</td>
<td>94.2</td>
<td>97.0</td>
<td>97.2</td>
<td>96.6</td>
<td>85.1</td>
<td>96.2</td>
<td>100.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>94.0</td>
<td>95.0</td>
<td>96.2</td>
<td>96.4</td>
<td>95.6</td>
<td>96.2</td>
<td>96.0</td>
<td>98.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.2.: Successful FDI percentage for sensor placement cases 1 and 2 – with network partitioning.

As it can be seen, the success coefficient is significantly larger. The drawback is of course that the fault cannot be localized to a single node.

\(^4\)The sensor selections are the same as in the first case since this part of the algorithm does not change.
8.4.2. Staggered sensor deployment

Another aspect of practical interest is the sensor deployment. Assuming that some of the network nodes already have sensors, the question arises of how to place the additional sensors such that overall the FDI accuracy increases. This is tested by considering two scenarios: i) place \( s = s' + s'' \) in one step and ii) place \( s' \) and after, the remaining \( s'' \). We take \( s' = 4 \) and \( s'' = 3 \) and observe that in case (i) the selected nodes are \( \{9, 28, 12, 31, 24, 7, 25\} \) with an FDI accuracy of 89.7\% whereas in case (ii) we obtain \( \{9, 30, 12, 15\} \) and then \( \{29, 8, 10\} \) with an overall FDI accuracy of 89.3\%. We note that the selections differ but still the accuracy FDI is nearly the same.

8.5. Conclusion

We have shown that classification and dictionary learning can be adapted to the the twin problems of sensor placement and leakage isolation in a water network. The results have shown good accuracy and pointed towards some promising directions of study like: network partitioning, tweaking of the dictionary learning procedure, and alternative sensor placement strategies.
9. Obstacle avoidance via B-spline parameterizations of flat trajectories

This chapter considers the collision avoidance problem in a multi-agent multi-obstacle framework. The originality in solving this intensively studied problem resides in the proposed geometrical view combined with differential flatness for trajectory generation and B-splines for the flat output parametrization. Using some important properties of these theoretical tools we show that the constraints can be validated at all times. Exact and sub-optimal constructions of the collision avoidance optimization problem are provided. The results are validated through extensive simulations over standard autonomous aerial vehicle dynamics.

The chapter is based on the conference article [C28]:

WOS: 000391154900167.

9.1. Preliminaries

One of the main issues in multi-obstacle, multi-agent environments is the collision avoidance assessment. Usually the avoidance constraints have to be considered both between an agent and fixed obstacles and between any two agents. The problem is intensively studied, yet actual, in the literature but is usually tilted towards heuristic approaches or online validations. The first usually lacks stability and performances guarantees and the later is easily boggled into numerical issues [3, 28].

An alternative approach is to solve offline the difficult trajectory generation part of the overall problem, online only a straightforward trajectory tracking is employed. This reduces significantly the online computations and allows stability and performance analysis. The caveat is that the trajectory to be computed has to respect the dynamics of the agent and to validate the collision avoidance constraints at all times [105, 131].

An interesting implementation is represented by flat trajectory design which guarantees that the corresponding system dynamics are respected (with the caveat that state and input constraints are not easily accounted for [37, 163]). This construction shifts the state and input constraints into
constraints over the flat output. To handle this, usually, the flat output is projected over some basis functions which means that only the coefficients of the projections need to be found. In this sense, B-spline functions represent an ideal choice since they have enough flexibility [162] and nice theoretical properties (of which we will make extensive use throughout the chapter).

The present work builds on results sketched in [161] and further advances the topic in several directions. Foremost, we provide exact and sub-optimal formulations of the collision avoidance problems between an agent and the obstacles and between any two agents. In both cases we make use of the geometrical properties of the B-spline functions which allow to bound locally the trajectories obtained through them. Hence, the collision problems become separation problems between sets of consecutive points. In the exact case, these constraints lead to nonlinear formulation where both the control points and the separation hyperplanes are variables. A simplified (and hence sub-optimal approach) is to select the separation hyperplanes from the support hyperplanes of the obstacles therefore reducing the problem to a mixed integer formulation.

9.2. Prerequisites

Designing reference trajectories in a multi-agent multi-obstacle environment is difficult. A popular way is to characterize them through flatness constructions [37, 47, 91]. Here we describe some of the basics of flat trajectory and their parametrization via B-spline basis functions.

9.2.1. Flat trajectories

A nonlinear time invariant system:

\[ \dot{x}(t) = f(x(t), u(t)), \]

(9.1)

where \( x(t) \in \mathbb{R}^n \) and \( u(t) \in \mathbb{R}^m \) are the state/input vectors is called differentially flat if there exists the flat output \( z(t) \in \mathbb{R}^m \):

\[ z(t) = \gamma(x(t), u(t), \dot{u}(t), \cdots, u^{(q)}(t)) \]

(9.2)

such that the states and inputs can be algebraically expressed in terms of \( z(t) \) and a finite number of its higher-order derivatives:

\[ x(t) = \Theta(z(t), \dot{z}(t), \cdots, z^{(q)}(t)), \]
\[ u(t) = \Phi(z(t), \dot{z}(t), \cdots, z^{(q)}(t)). \]

Within the multi-agent framework, the most important aspect of construction (9.2)–(9.3) is that it reduces the problem of trajectory generation to finding an adequate flat output (9.2). This means choosing \( z(t) \) such that, via mappings \( \Theta(\cdot), \Phi(\cdot) \), various constraints on state and inputs (9.3) are verified. Since the flat output may be difficult to compute under these restrictions, we parametrize \( z(t) \) using a set of smooth basis functions \( \Lambda_i(t) \):

\[ z(t) = \sum_{i=1}^{N} \alpha_i \Lambda_i(t), \quad \alpha_i \in \mathbb{R}. \]

(9.3)

Parameter \( N \) depends on the number of constraints imposed onto the dynamics [175].
There are multiple choices for the basis functions $\Lambda^i(t)$. Among these, $B$-spline basis functions are well-suited to flatness parametrization due to their ease of enforcing continuity and because their degree depends only up to which derivative is needed to ensure continuity [37, 162].

9.2.2. B-splines

A B-spline of order $d$ is characterized by a knot-vector \[ T = \{\tau_0, \tau_1 \ldots \tau_m\}, \tag{9.4}\]
of non-decreasing time instants ($\tau_0 \leq \tau_1 \leq \cdots \leq \tau_m$) which parametrizes the associated basis functions $B_{i,d}(t)$:

$$B_{i,1}(t) = \begin{cases} 
1, & \text{for } \tau_i \leq t < \tau_{i+1} \\
0 & \text{otherwise}
\end{cases}, \tag{9.5a}$$

$$B_{i,d}(t) = \frac{t - \tau_i}{\tau_{i+d-1} - \tau_i} B_{i,d-1}(t) + \frac{\tau_{i+d} - t}{\tau_{i+d} - \tau_{i+1}} B_{i+1,d-1}(t) \tag{9.5b}$$

for $d > 1$ and $i = 0, 1 \ldots n = m - d$.

Considering a collection of control points $P = \{p_0, p_1 \ldots p_n\}$, we define a $B$-spline curve as a linear combination of the control points (9.6) and the B-spline functions (9.5a)–(9.5b)

$$z(t) = \sum_{i=0}^{n} B_{i,d}(t)p_i = PB_d(t) \tag{9.7}$$

where $P = [p_0 \ldots p_n]$ and $B_d(t) = [B_{0,d}(t) \ldots B_{n,d}(t)]^T$. This construction yields several properties [128]:

P1) $z(t)$ is $C^\infty$ in any $t \notin T$ and $C^{d-1}$ in any $t \in T$;

P2) at $\tau_i < t < \tau_{i+1}$, $z(t)$ depends only on the B-splines $B_{i-d+1,d}(t) \ldots B_{i,d}(t)$; consequently, the B-spline curve $z(t)$ lies within the union of all convex hulls formed by all $d$ successive control points;

P3) the ‘r’ order derivatives of B-spline basis functions can be expressed as linear combinations of B-splines of lower order ($B_d^{(r)}(t) = M_r B_{d-r}(t)$ with matrices $M_r$ of appropriate dimensions and content);

P4) taking the first and last $d$ knot elements equal ($\tau_0 = \cdots = \tau_{d-1}$ and $\tau_{n+1} = \cdots = \tau_{n+d}$) leads to a clamped B-spline curve where the first and last control points coincide with the curve’s end points.
9.3. Flat trajectory generation

Let us consider a collection of $N + 1$ way-points and the time stamps associated to them:

$$\mathcal{W} = \{w_s\} \text{ and } T_W = \{t_s\},$$  \hspace{1cm} (9.8)

for any $s = 0 \ldots N$. The goal is to construct a flat trajectory which passes through each way-point $w_s$ at the time instant $t_s$ (or through a predefined neighborhood of it \[161\]), i.e., to find a flat output $z(t)$ such that

$$x(t_s) = \Theta(z(t_{s}), \ldots z^{(r)}(t_{s})) = w_s, \; \forall s = 0 \ldots N.$$  \hspace{1cm} (9.9)

Remark 9.1. Note that here we assume that the way-points are defined over the entire state. Arguably there might be situations where only a subspace of the state is of interest (e.g., only the position components of the state).

Making use of the B-spline framework we provide a vector of control points (9.6) and its associated knot-vector (9.4) such that (9.9) is verified (parameter $d$ is chosen such that continuity constraints are respected):

$$\tilde{\Theta}(B_d(t_s), P) = w_s, \; \forall s = 0 \ldots N,$$  \hspace{1cm} (9.10)

where $\tilde{\Theta}(B_d(t), P) = \Theta(PB_d(t) \ldots PM_dB_{d-r}(t))$ is constructed along Property P3).

Let us assume that the knot-vector is fixed ($\tau_0 = t_0, \tau_{n+d} = t_N$). Then, we can write an optimization problem with control points $p_i$ as decision variables whose goal is to minimize a cost $\Xi(x(t), u(t))$ along the interval $[t_0, t_N]$:

$$P = \arg\min_P \int_{t_0}^{t_N} ||\tilde{\Xi}(B_d(t), P)||_Q dt \text{ s.t. constraints (9.10) are verified}$$  \hspace{1cm} (9.11)

with $Q$ a positive symmetric matrix. The cost

$$\tilde{\Xi}(B_d(t), P) = \Xi(\tilde{\Theta}(B_d(t), P), \tilde{\Phi}(B_d(t), P))$$

can impose any penalization we deem necessary (length of the trajectory, input variation, input magnitude, etc). In general, such a problem is nonlinear (due to mappings $\tilde{\Theta}(\cdot)$ and $\tilde{\Phi}(\cdot)$) and hence difficult to solve. A nonlinear MPC iterative approach has been extensively studied \[37\].

9.4. Main idea

Let us consider a collection of polyhedral obstacles

$$\mathcal{O} = \{O_1 \ldots O_{N_o}\}$$  \hspace{1cm} (9.12)

and assume that the k-th agent follows a trajectory $r_k(t)$ during the interval $[t_0, t_N]$, generated as in (9.9) through a collection of control points$^{1}$ $\mathbb{P}^k = \{p^k_j\}$ and the associated knot vector $\mathbb{T}^k = \{\tau^k_j\}$.

$^1$For convenience we keep a common degree ‘d’ and number of control points ‘n’. This is a reasonable assumption as long as the agents have a common dynamic and follow similar restrictions.
Consequently, the collision avoidance conditions can be formulated as follows:

i) collision avoidance between the k-th agent and l-th obstacle:

\[ r_{k}(t) \notin O_l, \forall t \in [t_0, t_N], \quad (9.13) \]

ii) collision avoidance between the \( k_1 \)-th and \( k_2 \)-th agents (for any \( k_1 \neq k_2 \)):

\[ r_{k_1}(t) \neq r_{k_2}(t), \forall t \in [t_0, t_N]. \quad (9.14) \]

### 9.4.1. The exact case

The distinctive feature of conditions (9.13)–(9.14) is that they require a continuous time interval \([t_0, t_N]\) validation (i.e., imposing constraints at discrete time instants \( t_k \) along the interval is not deemed sufficient). We make use of Property P2) which allows to bound the continuous B-spline parametrized curve by its control points. Coupling this with the separating hyperplane theorem (a well-known construction [159] which states that for any two disjoint convex objects there exists a separating hyperplane) several results are attainable. First, we provide a slight reformulation of Proposition 1 from [161].

**Proposition 9.1.** The k-th agent is guaranteed to avoid obstacles (9.12), i.e., to verify (9.13), if there exists \( c_{il}^k \in \mathbb{R}^n \) s.t.

\[
\max_{j \in \{ i-d+1 \ldots i \}} \left( c_{il}^k \right)^\top p_j^k \leq \min_{x \in \Theta^{-1}(O_l)} \left( c_{il}^k \right)^\top x, \quad (9.15)
\]

for \( i = d-1 \ldots n \) and \( \forall O_l \in \mathbb{O} \).

**Proof.** Condition (9.15) states that there exists a hyperplane defined by its normal \( c_{il}^k \) which separates the points \( \{ p_{i-d+1}^k \ldots p_i^k \} \) from the obstacle \( \Theta^{-1}(O_l) \). Since, according to Property P2), the curve (9.9) is contained in \( \cup_{i=d \ldots n} \text{ConvexHull} \{ p_{i-d+1}^k \ldots p_i^k \} \) it follows that (9.15) is a sufficient condition to verify (9.13). \( \square \)

**Remark 9.2.** In Proposition 9.1 note the use of mapping \( \Theta^{-1}(\cdot) \). This appears because the obstacle avoidance constraint is the state-space whereas (9.15) is in the control point space. \( \diamond \)

A similar reasoning is employed for the inter-agent collision condition (9.14).

**Proposition 9.2.** The pair \((k_1, k_2)\) of agents, with \( k_1 \neq k_2 \), is guaranteed to avoid collision, i.e., to validate (9.14), if there exists \( c_{i_1i_2}^{k_1k_2} \in \mathbb{R}^n \) s.t.

\[
\max_{j \in \{ i_1-d+1 \ldots i_1 \}} \left( c_{i_1i_2}^{k_1k_2} \right)^\top p_{j}^{k_1} \leq \min_{j \in \{ i_2-d+1 \ldots i_2 \}} \left( c_{i_1i_2}^{k_1k_2} \right)^\top p_{j}^{k_2}, \quad (9.16)
\]

for all possible pairs \((i_1, i_2)\) which validate

\[
\{(i_1, i_2) : \tau_{i_1}^{k_1}, \tau_{i_1+1}^{k_1} \cap \tau_{i_2}^{k_2}, \tau_{i_2+1}^{k_2} \neq \emptyset\}. \quad (9.17)
\]
Proof. Recall that (as per Property P2)) a region ConvexHull\{p_{t-d+1} \ldots p_t\} contains the B-spline curve in the time interval \([\tau_i, \tau_{i+1}]\). Applying this to the agents \(k_1\) and \(k_2\) means that all regions corresponding to indices (9.17) should not intersect as they contain overlapping time instants. The separation is enforced by (9.16) which is a sufficient condition for (9.14).

Remark 9.3. Eq. (9.17) can be avoided altogether if the B-spline parametrizations share the same knot vector (i.e., \(T^{k_1} = T^{k_2}\)). In such a case, variable \(c_{i_1i_2}^{k_1k_2}\) becomes \(c_{i_1}^{k_1}k_2\) and condition (9.16) is simplified to
\[
\max_{j \in (i-d+1 \ldots i)} \left( c_{i_1}^{k_1}k_2 \right)^\top p_{j_1}^{k_1} \leq \min_{j \in (i-d+1 \ldots i)} \left( c_{i_1}^{k_1}k_2 \right)^\top p_{j_1}^{k_2},
\]
for all \(i \in \{d-1 \ldots n\}\).

\[\]

**9.4.2. The sub-optimal case**

Verifying (9.15) (or (9.16)) is difficult in practice due to the presence of bi-linear terms (e.g., in Proposition 9.1 both \(c_{il}\) and \(p_j^k\) are variables). Hereafter we propose a simpler (and hence sub-optimal) implementation.

The main idea is that instead of letting the separating hyperplane from Proposition 9.1 or 9.2 be itself a variable, we choose from within a predefined pool of hyperplanes. A natural choice is to select from the support hyperplanes of the obstacles. By definition, such a hyperplane contains on one side the obstacle and hence, it remains only to check whether the control points lie on the opposite side. The selection of the active hyperplane is done through decision variables (i.e., binary variables) which leads to a mixed-integer pseudo-linear formulation.

To generate the collection of hyperplanes, we consider the polyhedral sets bounding\(^2\) \(\hat{\Theta}^{-1}(O_l)\) and take the support hyperplanes which characterize them:
\[
\mathcal{H}_m = \{ x : h_m^\top x = k_m \}, \forall m = 1 \ldots M. \tag{9.19}
\]
Each of these hyperplanes partitions the space in two “half-spaces”:
\[
\mathcal{H}_m^+ = \{ x : h_m^\top x \leq k_m \}, \tag{9.20}
\]
\[
\mathcal{H}_m^- = \{ x : -h_m^\top x \leq -k_m \}. \tag{9.21}
\]

Taking into account all possible combinations of half-spaces leads to a hyperplane arrangement which divides the space into a collection of disjoint cells which are completely characterized by sign tuples [44]:
\[
\mathcal{H} = \bigcup_{\sigma \in \Sigma} \mathcal{A}(\sigma) = \bigcup_{\sigma \in \Sigma} \left( \bigcup_{m=1}^M \mathcal{H}_m^{\sigma(m)} \right). \tag{9.22}
\]
where \(\Sigma \subset \{-, +\}^M\) denotes the collection of all feasible (corresponding to non-empty regions \(\mathcal{A}(\sigma)\)) sign tuples. Each of these tuples can be allocated to either

\(^2\)We assume that \(\hat{\Theta}^{-1}(O_l)\) is a polyhedral set as well. In general this might not hold, but in that case a polyhedral approximation can be obtained.

99 of 160
Chapter 9. Obstacle avoidance via B-spline parameterizations of flat trajectories

1. the admissible domain $\mathbb{R}^n \setminus \mathcal{O}$:

$$\Sigma^0 = \{ \sigma : \mathcal{A}(\sigma) \cap \mathcal{O} = \emptyset \}, \quad (9.23)$$

2. or the interdicted domain $\mathcal{O}$:

$$\Sigma^* = \{ \sigma : \mathcal{A}(\sigma) \cap \mathcal{O} \neq \emptyset \}, \quad (9.24)$$

where $\Sigma^* \cap \Sigma^0 = \emptyset$ and $\Sigma^* \cup \Sigma^0 = \Sigma$. With these elements we can provide the following corollaries.

**Corollary 9.1.** For an obstacle $\mathcal{A}(\sigma^*)$ with $\sigma^* \in \Sigma^*$, a sufficient condition to guarantee $3^{(9.13)}$ is:

$$-\sigma^*(m) h^\top_m p_j^k \leq -\sigma^*(m) k_m + T \alpha^k_{im},$$

$$\forall m = 1 \ldots M, j = i - d + 1 \ldots i \quad (9.25a)$$

$$\sum_{m=1}^{M} \alpha^k_{im} \leq M - 1 \quad (9.25b)$$

for $i \in \{d - 1 \ldots n\}$. □

**Proof.** Taking in $(9.25a)$ the binary variable $\alpha^k_{im} = 0$ means that the i-th region Convex Hull $\{p^k_{i-d+1} \ldots p^k_i\}$ of the k-th agent sits on the opposite side of the obstacle $\mathcal{A}(\sigma^*)$ with respect to the m-th hyperplane. The converse, taking $\alpha^k_{im} = 1$ means that inequality $(9.25a)$ is discarded since the right hand term is sufficiently large to ignore the values on the left side (assuming that $T$ was taken as a sufficiently large positive constant).

Condition $(9.25b)$ forces that for any consecutive $d + 1$ points at least one of the inequalities $(9.25a)$ is enforced since at least one of the variables $\alpha^k_{im}$ has to be zero. □

Furthermore, we relax Proposition 9.2 (in addition to the simplification proposed in Remark 9.3) into the following corollary.

**Corollary 9.2.** The pair $(k_1, k_2)$ of agents, with $k_1 \neq k_2$, is guaranteed to avoid collision, i.e., to validate $(9.14)$, if:

$$\max_{j \in \{i-d \ldots i\}} h^\top_m p_{j}^{k_1} \leq \min_{j \in \{i-d+1 \ldots i\}} h^\top_m p_{j}^{k_2} + T \beta^{k_1 k_2}_{im},$$

$$\sum_{m=1}^{M} \beta^{k_1 k_2}_{im} \leq M - 1 \quad (9.26)$$

for $i \in \{d - 1 \ldots n\}$. □

**Proof.** The binary variables $\beta^{k_1 k_2}_{im}$ denote whether the i-th regions Convex Hull $\{p^k_{i-d+1} \ldots p^k_i\}$ and Convex Hull $\{p^k_{i-d+1} \ldots p^k_i\}$ are separated through the m-th hyperplane (whenever $\beta^{k_1 k_2}_{im} = 0$ the inequality $(9.26)$ is enforced and otherwise is discarded). Eq. $(9.27)$ assures that at least one of the hyperplanes is active. □

**Remark 9.4.** Corollary 9.2 considers a simultaneous computation of trajectories. An alternative is to compute them iteratively such that from the point of view of the current agent the obstacles to be

---

3 We make the simplifying assumption that to each obstacle $O_l$ corresponds a single sign tuple.
avoided at \( t \in [\tau_i, \tau_{i+1}] \) become:

\[
\mathcal{O} \leftarrow \mathcal{O} \cup \left( \bigcup_{l<k} \text{ConvexHull}\{p_{l-d+1}^l \ldots p_k^l\} \right)
\] (9.28)

where besides the obstacles (9.12), the i-th regions \( \text{ConvexHull}\{p_{l-d+1}^l \ldots p_k^l\} \) of the previous agents (with index \( l < k \) and whose trajectories are hence already computed) are also considered as obstacles.

\[ \star \]

**Remark 9.5.** An agent may have a safety region around it (i.e., because the agent cannot be reduced to a point or due to the presence of disturbances in the dynamics). Whatever the reason, and the modality to obtain it, a safety region \( S_k \) can be attached to the k-th agent. Consequently, the collision avoidance constraints (9.13)–(9.14) become:

\[
\{r_k(t)\} \oplus S_k \notin O_l, \forall t \in [t_0, t_N],
\] (9.29)

and

\[
\{r_{k_1}(t)\} \oplus S_{k_1} \neq \{r_{k_2}(t)\} \oplus S_{k_2}, \forall t \in [t_0, t_N].
\] (9.30)

The previous results can be easily adapted to constraints (9.29)–(9.30) by enlarging the obstacles \( (r_k(t) \notin O_l \oplus \{-S_k\}) \) and by requiring a larger inter-distance between agents respectively \( (r_{k_1}(t) - r_{k_2}(t) \notin \{-S_{k_1}\} \oplus S_{k_2}) \).

\[ \star \]

**Remark 9.6.** Lastly, it is worth mentioning that in all previous propositions and corollaries there is no guarantee of feasibility for the optimization problems. The solution is to incrementally increase the number of variables (i.e., the control points) until a feasible solution is reached.

\[ \star \]

### 9.5. Illustrative example for an UAV system

We revisit the test case from [161]. A 2D 3-DOF model (9.31) of an airplane in which the autopilot forces coordinated turns (zero side-slip) at a fixed altitude:

\[
\begin{align*}
\dot{x}(t) &= V_a(t) \cos \Psi(t), \\
\dot{y}(t) &= V_a(t) \sin \Psi(t), \\
\dot{\Psi}(t) &= \frac{g \tan \Phi(t)}{V_a(t)}
\end{align*}
\] (9.31)

The state variables are represented by the position \((x(t), y(t))\) and the heading (yaw) angle \(\Psi(t) \in [0, 2\pi]\) rad. The input signals are the airspeed velocity \(V_a(t)\) and the bank (roll) angle \(\Phi(t)\), respectively.

We take as flat output the position components of the state, \(z(t) = \begin{bmatrix} z_1(t) & z_2(t) \end{bmatrix}^T = \begin{bmatrix} x(t) & y(t) \end{bmatrix}^T\) which permits to compute the remaining variables:

\[
\begin{align*}
\Psi(t) &= \arctan \left( \frac{\dot{z}_2(t)}{\dot{z}_1(t)} \right), \\
V_a(t) &= \sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}
\end{align*}
\] (9.32a) (9.32b)
\( \Phi(t) = \arctan \left( \frac{1}{g} \frac{\ddot{z}_2(t) \dot{z}_1(t) - \dot{z}_2(t) \ddot{z}_1(t)}{\sqrt{\dot{z}_2^2(t) + \ddot{z}_2^2(t)}} \right). \) 

(9.32c)

Note that in the heading component of the state appear 1st order and in the roll angle input appear 2nd order derivatives of the flat outputs. Hence, if we wish to have smooth state and input (their derivatives to be continuous) it follows that the B-spline parametrization has to have at least degree \( d = 4 \).

Further, we consider way-points which fix only the position components of the state and time-stamps at which the trajectory has to pass through them. Thus we manage to skirt some of the thornier numerical aspects: the dependence between the B-spline basis functions and the position components is linear (\( \Theta(B_d(t), P) = PB_d(t) \)), and hence the cost and constraints will be easily written.

We take as cost to be minimized the length of the curve, i.e., \( \Xi(B_d(t), P) = ||\dot{z}(t)|| \). This translates into the integral cost:

\[
\int_{t_0}^{t_N} ||\dot{z}(t)|| \, dt = \int_{t_0}^{t_N} ||PM_1B_{d-1}(t)|| \, dt
\]

\[
= \sum_{i,j} ([PM_1]_i)^T \left( \int_{t_0}^{t_N} B_{i,d-1}(t)B_{j,d-1}(t) \, dt \right) [PM_1]_j
\]

(9.33)

where matrix \( M_1 \) links \( B_{d-1}(t) \) and \( \dot{B}_d(t) \) as in property Property P3) and \([\cdot]_i \) extracts the i-th column from the argument. Since the inner integrals can be computed numerically, we have a quadratic cost which can be used with the various constructions from Section 9.3, see also \([37]\) for a similar treatment of cost computations.

For illustration purposes we consider 9 hyperplanes:

\[
H = \begin{bmatrix}
-0.5931 & 0.8051 \\
0.1814 & 0.9834 \\
-0.0044 & 1.0000 \\
-0.1323 & 0.9912 \\
-0.7011 & -0.7131 \\
0.8152 & -0.5792 \\
0.4352 & 0.9003 \\
1.0000 & -0.0075 \\
-0.5961 & -0.8029
\end{bmatrix}^T, \quad h = \begin{bmatrix}
4.2239 \\
0.1719 \\
0.9975 \\
0.2728 \\
3.6785 \\
0.0317 \\
1.6598 \\
4.5790 \\
1.0280
\end{bmatrix}
\]

which lead to a hyperplane arrangement (9.22) where the interdicted tuples (9.24)

\( \Sigma^* = \{(+++---+++-), (++-+-+-+++), (+-+++---++)\} \)

correspond to three obstacles. Further, we take three way-points (initial, intermediary and final)
though which the trajectory has to pass at predefined times:

\[ \mathcal{W} = \left\{ \begin{bmatrix} -9 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 6 \\ 0 \end{bmatrix} \right\}, \quad T_{\mathcal{W}} = \{0, 5, 10\}. \]

We compute a flat trajectory (parametrized by \( n = 12 \) and \( d = 6 \)) which passes through the given way-points, minimizes the total path-length and respects one of the following scenarios: i) without any collision avoidance restriction; ii) with collision avoidance as in Proposition 9.1; iii) with collision avoidance as in Corollary 9.1. The obstacles, their support hyperplanes and the resulting trajectories are depicted in Figure 9.1a.

Scenarios ii) and iii) both accomplish the task of avoiding the obstacles with comparable computation times and path lengths. Figures 9.1b and 9.1c show details of the collision avoidance. In both cases the 1st and 3rd convex regions are considered (Convex Hull \( \{p_0 \ldots p_5\} \) and Convex Hull \( \{p_2 \ldots p_7\} \)) together with their separating hyperplanes. In Figure 9.1b these hyperplanes are the 1st and respectively 3rd support hyperplane whereas in Figure 9.1c the separating hyperplanes are the result of the optimization problem and are \( \mathcal{H}'_1 = \left\{ \begin{bmatrix} -0.1273 \\ 0.2913 \end{bmatrix}^\top x = 11 \right\}, \mathcal{H}'_3 = \left\{ \begin{bmatrix} -0.05490.8461 \end{bmatrix}^\top x = 11 \right\}. \)

As mentioned earlier, the result of the optimization problem (computation time, total length of the trajectory) depends heavily on the number of control points \( n + 1 \) and degree \( d \). We illustrate these evolutions in Table 9.1. Several remarks are in order. First, it seems that after an initial decrease in the path length the future reductions are negligible and at a significant computation time for the
mixed-integer method. Next, and somewhat surprising, is that the non-linear method is extremely sensitive to parameter variations (number of control points, degree, positioning and number of the way-points, etc) such that the results obtained are not trustworthy. The one advantage of the latter over the former is that it may provide a feasible solution for small values of \( n \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{MI} )</td>
<td>0.0985</td>
<td>6.231</td>
<td>17.106</td>
<td>15.034</td>
<td>21.522</td>
</tr>
<tr>
<td>( \ell_{MI} )</td>
<td>*</td>
<td>16.877</td>
<td>16.307</td>
<td>16.202</td>
<td>16.536</td>
</tr>
<tr>
<td>( t_{NL} )</td>
<td>37.426</td>
<td>0.6412</td>
<td>1.055</td>
<td>0.6321</td>
<td>*</td>
</tr>
<tr>
<td>( \ell_{NL} )</td>
<td>203.51</td>
<td>33.469</td>
<td>202.076</td>
<td>31.159</td>
<td>*</td>
</tr>
</tbody>
</table>

Table 9.1.: Evolution of trajectories characteristics for degree \( d = 4 \) in the non-linear and mixed-integer formulations.

The collision avoidance between two agents is similar and not depicted here. The numerical simulations have been done using Yalmip [94] and MPT [88] in Matlab 2013a. The nonlinear solver used was IPOPT [174].

### 9.6. Conclusions

This chapter considers collision avoidance in a multi-agent, multi-obstacle framework. Using differential flatness for trajectory generation and B-splines for the flat output parametrization we show that the restrictions can be validated at all times. Exact and sub-optimal constructions are provided. Future work may consist in analysis of the feasibility of the problem, relaxation of way-point restrictions (similar with work done in [161]).
10. A combinatorial view of the explicit representation of the MPC problem

This chapter revisits the explicit MPC representation and related notions. I point to the special structure of the constraint matrices and exploit it in order to provide novel results. I give an upper bound for the collection of admissible active sets with use in the mixed integer representation of the KKT problem and a partial recursive description of the explicit partitioning of the MPC problem. The results are tested over illustrative examples.

The chapter is based on book chapter [C28]:


DOI: 10.3182/20120823-5-NL-3013.00079. EID: 2-s2.0-84867056750.

10.1. Preliminaries

A typical, and well-studied example of constrained optimization problem, is the Model Predictive Control (MPC) formulation (see, e.g., [97]) which minimizes a cost function over a finite prediction horizon and considers constraints at every step. MPC has been a remarkable industrial success, with thousands of installations worldwide [134] due to the relative ease with which constraints in both states/outputs and inputs are handled.

When the current state is considered as a parameter, the MPC can be formulated as a multi-parametric problem and the optimal control is given explicitly as a piecewise affine function of the state [15]. This approach has the benefit of delegating the intensive computations into the off-line part of the procedure. There is extensive work detailing the construction of the explicit solution, see, e.g., the excellent survey [6]. Broadly, the algorithms for constructing the explicit solution can be classified into two categories: state space exploration and combinatorial methods. The state space exploration methods [15, 165] explore the state space and partition it into regions until the entire feasible domain is covered. The combinatorial methods [74, 109, 156] on the other hand, consider a collection of
candidate sets of active constraints and keep the ones which correspond to non-empty regions in the state space. Both methods are affected by complementary shortcomings. The state space exploration methods generally avoid unnecessary complexity but are computationally expensive. Conversely, the combinatorial methods are afflicted with an exponential increase in the set of admissible combinations of active constraints.

In this chapter we give special attention to the structure of the MPC problem in order to provide novel results regarding the explicit representation. The key aspect is that up to the \( k \)th instant of time only the first \( k \) inputs will appear in the constraint descriptions (a property which is to be expected, otherwise the underlying dynamics would not be causal). Consequently, the matrix grouping these constraints will have a lower block-triangular structure. We exploit this structure to provide: i) a stricter bound for the number of admissible sets of active constraints and the means to enumerate them and ii) a partial recursive description of the explicit partitioning of the MPC problem. Each of these improvements can be used in the methodologies mentioned above: by giving an upper bound and an enumeration of the admissible combinations of active constraints we improve upon the existing combinatorial methods and by providing a partial description of the explicit partitioning we reduce the portion of state-space which has to be analyzed by the state-space exploration methods. For the former we use the newly found bounds to improve upon the mixed integer representation of the associated KKT problem. For the latter we point to a more compact formulation of the explicit representation whenever only the first input is considered relevant.

10.2. Prerequisites

Let \( \mathbb{R} \), \( \mathbb{Z} \) and \( \mathbb{Z}_+ \) denote the field of real numbers, the set of integers and the set of non-negative integers, respectively. Let \( x_{[c_1, c_2]} = \begin{bmatrix} x(k + c_1) & \ldots & x(k + c_2) \end{bmatrix}^T \), with \( c_1, c_2 \in \mathbb{Z} \) denote a column vector of elements whose index increases monotonically and where \( k \in \mathbb{Z}_+ \) denotes the current instant of time. A finite intersection of inequalities which describes a non-empty region is called a polyhedral set. A polytope is a bounded polyhedral set. The cardinality of a set \( A \) is denoted as \( \#A \).

The collection of all possible \( N \) combinations of binary variables will be noted

\[
\{0, 1\}^N = \{(b_1, \ldots, b_N) : b_i \in \{0, 1\}, \forall i = 1, \ldots, N\}.
\]

Let us consider a fairly typical (i.e., LTI dynamics, quadratic cost function and linear constraints) MPC formulation:

\[
\begin{align*}
\min_{u_0, \ldots, u_{N-1}} & \sum_{k=0}^{N-1} (u_k^T R u_k + x_k^T Q x_k) + x_N^T Q x_N \\
\text{s.t.} \quad & G_k x_k + H_k u_k \leq b_k, \quad k = 0, \ldots, N - 1 \\
& x_{k+1} = A x_k + B u_k, \quad k = 0, \ldots, N - 1
\end{align*}
\]

(10.1a)

\( (10.1b) \)

\( (10.1c) \)

where \( Q \succeq 0, Q_f \succeq 0, R \succ 0 \) are the cost matrices with \( Q, Q_f \in \mathbb{R}^{n \times n} \) and \( R \in \mathbb{R}^{m \times m} \). \( x_0 \in \mathbb{R}^n \) is the current state and \( N \) is the length of the prediction horizon.
For compactness of notation, we recast (10.1) as
\[
\begin{align*}
\min_u & \frac{1}{2} u^T H u + x_0^T F u \\
\text{s.t.} & \quad Gu \leq W + Ex_0. 
\end{align*}
\tag{10.2a}
\]
where
\[
u = \begin{bmatrix} u_0^T & \ldots & u_{N-1}^T \end{bmatrix}^T
\]
and matrices \(H, F, W\) and \(E\) are constructed accordingly. We note for further use that the optimization problem resides in the \(\mathbb{R}^{N+m}\) space and assume that the set of constraints has \(q\) inequalities.

We consider that the optimization problem (10.2) is convex and regular, admitting a unique optimal solution everywhere in the feasible domain. Then it can be replaced by its Karush-Kuhn-Tucker (KKT) conditions, resulting in:
\[
\begin{align*}
Hu + F^T x_0 + G^T \lambda &= 0 \quad (10.3a) \\
\lambda &\geq 0 \quad (10.3b) \\
Gu - W - Ex_0 &\leq 0 \quad (10.3c) \\
\lambda \times (Gu - W - Ex_0) &= 0 \quad (10.3d)
\end{align*}
\]
where the “\(\times\)” symbol indicates that the \(k^{th}\) element of the vector \(\lambda\) of Lagrangian multipliers multiplies the \(k^{th}\) constraint from (10.2b).

Let us partition the set of constraints (10.3c) into active and inactive constraints. We consider that the (in)active constraints are (not) saturated by the optimal solution \(u\) and the parameter \(x_0\). Let us recall the following notion characterizing an active set of constraints.

**Definition 10.1** ([165]). For an active set, we say that the linear independence constraint qualification (LICQ) holds if the set of active constraint gradients are linearly independent.

We denote the matrices associated to the active set by \(\tilde{G}, \tilde{E}\) and \(\tilde{W}\) and note that they verify
\[
\tilde{Gu} = \tilde{W} + \tilde{E}x_0. \tag{10.4}
\]
As long as the set of active constraint gradients (i.e., \(\tilde{G}\)) respects LICQ, the KKT problem (10.3) is well-posed in the sense that both the primal solution \(u\) and the dual solution \(\lambda\) are uniquely defined.

Since \(\tilde{G}\) is full-rank there exists \(M = (\tilde{G}^T H^{-1} \tilde{G}^T)^{-1}\). Further, (see [15]), we introduce (10.4) into (10.3) and obtain both the control as a piecewise affine function of \(x_0\),
\[
U_N(x_0) = H^{-1} \tilde{G}^T M (\tilde{W} + \tilde{E}x_0) = F_N x_0 + g_N, \ \forall x_0 \in D_N^i, \tag{10.5}
\]
and the critical region
\[
D_N^i = \left\{ x : \begin{array}{ll}
-M(\tilde{W} + \tilde{E}x_0) \geq 0 \\
GH^{-1} \tilde{G}^T M(\tilde{W} + \tilde{E}x_0) \leq W + Ex_0
\end{array} \right\} \tag{10.6}
\]
over which it is optimal.

\(^1\)Whenever there is no confusion regarding the length of the prediction horizon we abuse the notation and denote \(u_{[0,N-1]}\) as \(u\).
The feasible domain

$$\mathbb{D}_N = \bigcup_{i \in \mathcal{I}_N} \mathcal{D}_i^N$$

(10.7)

describes the polyhedral region in state-space for which an optimal control exists. We denote $\mathcal{I}_N$ as the collection of LICQ\(^2\) active sets of constraints which lead to a nonempty region (10.6).

For completeness, we define also the reverse operator $\mathcal{R}_N$ which, provided with an active set of constraints $\mathcal{A}_i$, gives the associated feasible region: $\mathcal{D}_i^N = \mathcal{R}_N(\mathcal{A}_i)$.

Roughly speaking, there are two complementary directions: exploration of the state space for domain identification such that new regions (10.6) are added until they are exhausted or selection of candidate sets of active constraints which are added to $\mathcal{I}_N$ if they correspond to a non-empty region (10.6). Both approaches have to deal with the exponential increase in complexity of the explicit partitioning as a function of the length of the prediction horizon. Therefore, any “shortcut” which helps reduce the computational load is welcomed.

In the rest of the chapter we propose to exploit the structure of the constraint matrix $G$ and to derive novel (to the best of our knowledge) results regarding the explicit representation and the associated sets of active constraints.

As seen in (10.1b), up to the time instant $k$ only the initial state $x_0$ and the sequence of inputs $u_0, \ldots, u_k$ appear in the constraints. Algebraically, this means that matrix $G$ has a lower block-triangular structure:

$$G = \begin{bmatrix}
\begin{array}{c}
R_1 \\
R_2 \\
\vdots \\
R_N
\end{array}
\end{bmatrix}
\begin{array}{c}
\cdots \\
q_1 \times m \\
q_2 \times 2m \\
q_N \times Nm
\end{array}
$$

(10.8)

where each block $R_k$ describing the $k^{th}$ order constraints has $q_k$ rows and $k \cdot m$ columns (note that $q_1 + q_2 + \ldots q_N = q$).

In the rest of the chapter we use this structure to provide: i) a stricter bound for the number of sets of active constraints/critical regions in the explicit solution than the one commonly found in the literature and ii) a partial recursive description of the explicit partitioning of the MPC problem.

10.3. Upper bound for the collection of candidate sets of active constraints

A number of papers (see, e.g., [109]) try to enumerate a collection of candidate sets of active constraints. Each of these candidates is then used to compute both the optimal control and its feasible region. If the resulting region is nonempty then the control law is added to the global solution.

\(^2\)Since only these constraints determine a full-dimensional region in the state-space.
Let us define the collection of candidate sets of active constraints as $\mathcal{F}_N^\circ$. Since we are interested only in candidate sets which respect LICQ it is clear that we can select at most $N \cdot m$ constraints from the existing $q$ without violating the linear independence requirement in the $\mathbb{R}^{N \cdot m}$ space. This leads to an upper bound of

$$\#\mathcal{F}_N^\circ = \sum_{j=0}^{N-m} \binom{q}{j}$$

(10.9)

candidates. The bound is obtained by selecting from the available $q$ constraints $0, 1, \ldots, N \cdot m$ active constraints successively.

However, the resulting bound is conservative, and a better bound can be computed by making use of the special structure seen in (10.8). In the following result we give conditions which restrict the number of rows selections from the constraint matrix.

**Proposition 10.1.** Let $i_k$ denote the number of constraints selected from each block $R_k$. Then the conditions

$$0 \leq i_k \leq \min(q_k, k \cdot m), \quad \sum_{j=0}^k i_j \leq k \cdot m, \quad \forall k = 1, \ldots, N$$

(10.10)

define all the selections of constraints which can be LICQ and the collection $\mathcal{F}_N^\circ$ of candidate active sets has at most

$$\#\mathcal{F}_N^\circ = \sum_{(i_1, \ldots, i_N) \text{ verifies (10.10)}} \left( \prod_{j=1}^{N} \binom{q_j}{i_j} \right)$$

(10.11)

elements.

**Proof:** The reasoning for the conditions in (10.10) is based upon the fact that in a collection of `$y$' rows where only the first `$x$' elements in each row are non-zero, at most `$\min(x, y)$' rows can be selected and still be linearly independent. This property leads directly to the first inequality in (10.10) whereas for the second we observe that the following selection rule applies: from the first $q_1$ rows we can select at most $m$, from the first $q_1 + q_2$ rows we can select at most $2 \cdot m$ and so forth until, from the first $q_1 + \cdots + q_N = q$ rows we can select at most $N \cdot m$ rows.

The bound can be computed by simply counting the candidate sets of active constraints. For a selection $(i_1, \ldots, i_N)$ we take each block $R_j$ and extract all the combinations of $i_j$ rows from it, which means $\binom{q_j}{i_j}$ possibilities. Each of these combinations is then considered with each of the other combinations from the other blocks.

The construction of $\mathcal{F}_N^\circ$ was made assuming that any $i_j$ subset of rows from the block $R_j$ is linearly independent, assumption which may not hold in practical cases (e.g., when the bounds upon states/inputs are symmetrical). In this sense, consider the next definition.

**Definition 10.2.** We refer to an mpQP problem of the form (10.1) as input-symmetric if the involved constraints can be rewritten as

$$\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} u - \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} - \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} x_0 \leq 0$$

(10.12)

where $G_2 = -G_1$. Consequently, each block-matrix $R_j$ with $q_j$ rows given as in (10.8) can be written
as \( R_j = \begin{bmatrix} \bar{R}_j \\ -\bar{R}_j \end{bmatrix} \) where \( \bar{R}_j \) has \( q_j^2 \) rows.

**Corollary 10.1.** Consider that (10.1) is input-symmetrical, then with the notation of Proposition 10.1, the LICQ constraint selections have to respect

\[
0 \leq i_k \leq \min\left(\frac{q_k}{2}, k \cdot m\right), \quad \sum_{j=0}^{k} i_j \leq (k \cdot m), \quad \forall k = 1, \ldots, N
\]

and the collection \( \mathcal{J}_N^o \) of candidate active sets has at most

\[
\#\mathcal{J}_N^o = \sum_{(i_1, \ldots, i_N)} \text{verifies (10.13)} \left( \prod_{j=1}^{N} \left( \frac{q_j}{2} \right) \cdot 2^{i_j} \right)
\]

elements.

**Proof:** Relations (10.13) describe the same conditions as in Proposition 10.1 with the difference that at most \( \frac{q_j^2}{2} \) rows can be selected from a block \( R_j \).

For the bound (10.14) the reasoning is as follows: we select \( i_j \) rows from the \( q_j^2 \) rows which define the sub-block \( \bar{R}_j \); subsequently, we consider all the possible sign permutations for the selected constraints (i.e., \( 2^{i_j} \)); the rest of the procedure follows the proof of Proposition 10.1.

**Remark 10.1.** Evidently, for either Proposition 10.1 or Corollary 10.1, \( \mathcal{J}_N \subseteq \mathcal{J}_N^o \) is verified but usually the inclusion is strict since not all the candidate sets will have a non-empty feasible region in the state-space.

**Remark 10.2.** The conditions (10.10) and (10.13) describe an integral polytope (i.e., a polytope whose extreme points are integers) and the selections \((i_1, \ldots, i_N)\) are points contained in this polytope. These notions can be described formally using the Ehrhart polynomial and the associated theory [39]. Using these elements it is possible to determine the number of integer points which respect (10.10) or (10.13) and to enumerate them. The actual computation of the Ehrhart polynomial coefficients and the enumeration of the selections \((i_1, \ldots, i_N)\) can be done with the Ehrhart routines of the PolyLib library (see, [31] for details).

**Illustrative example**

For illustration purposes we take a very simple case. Consider a prediction horizon \( N = 2, m = 1 \) and \( q_1 = q_2 = 4 \). For these numerical values, conditions (10.10) become:

\[
0 \leq i_1 \leq 1, \ 0 \leq i_2 \leq 2 \text{ and } 0 \leq i_1 + i_2 \leq 2.
\]

As seen in Figure 10.1 there are 5 feasible selections \((i_1, i_2)\) which respect the constraints. For each of these we consider the number of row combinations in order to enumerate the candidate sets of \( \mathcal{J}_N^o \).

For example, for \((i_1, i_2) = (1, 1)\) we have \( \binom{4}{1} \cdot \binom{4}{1} = 16 \) possibilities: from \( R_1 \) we select the first row and pair it successively with each of the other 4 from \( R_2 \); we repeat the procedure with the 3 remaining rows from \( R_1 \). Taking into account all the possible combinations we obtain, as in Proposition 10.1,
that \( \#\mathcal{F}_N^c = 31 \) which is significantly less than the “naive” bound \( 2^{q_1+q_2} = 2^8 = 256 \) or even the LICQ-induced bound \( \sum_{j=0}^{N-m} \binom{N}{j} = \binom{8}{2} + \binom{8}{1} + \binom{8}{0} = 37. \)

Lastly, consider that the constrains define an input-symmetrical problem. Then conditions (10.13) apply and result, for these particular numerical values, in the same set of feasible selections as for (10.10). On the other hand, applying the bound (10.14) we obtain that \( \#\mathcal{F}_N^c = 29. \)

Of course, for larger values of the prediction horizon and for the number of constraints, the differences between the discussed bounds become significantly larger.

A tighter bound (as given in Proposition 10.1 or Corollary 10.1) has practical consequences. Most evidently, it offers an upper limit for the memory requirements when storing the critical regions of an explicit solution. Also, the relations which assure LICQ can be added into a mixed integer optimization problem (as detailed in the next section) and reduce the computation time.

### 10.3.1. Mixed integer representation of the KKT problem

In certain situations in control theory, multi-level programming problems arise naturally. For example, constructing a “soft constraint” MPC based upon the “hard constraint” problem (10.1) requires the computation of the maximal Lagrange multiplier which verifies the lower optimization problem (10.3), see, e.g. [62].

The difficulty resides in the lower-level problem where the complementarity condition (10.3d) induces nonlinearity. We recall here the technique proposed in [48] to reformulate (10.3) using the auxiliary binary variables \( s \in \{0,1\}^N \):

\[
\begin{align*}
Hu + F^T x_0 + G^T \lambda &= 0 \\
\lambda &\geq 0 \\
\lambda &\leq M^s \\
Gu - W - Ex_0 &\leq 0 \\
Gu - W - Ex_0 &\geq -M^u(1 - s)
\end{align*}
\]  

(10.15a) 
(10.15b) 
(10.15c) 
(10.15d) 
(10.15e)
Chapter 10. A combinatorial view of the explicit representation of the MPC problem

where $M^\lambda$ and $M^u$ are diagonal matrices with sufficiently large values for the diagonal elements.

The nonlinear condition (10.3d) is replaced by conditions (10.15c) and (10.15e) in the sense that they provide the same result as the original equality through suitable combinations of binary variables $s \in \{0, 1\}^N$: whenever $s_i = 1$, the set of inequalities (10.15d)−(10.15e) degenerates to an equality (i.e., $(Gu - W - Ex_0)_i = 0$) and, correspondingly, when $s_i = 0$, the inequality (10.15e) becomes redundant with respect to the set described by (10.15d) for parameter $M^u$ sufficiently large.

The direct inclusion of $\lambda$ as free variables in (10.15) allows for unnecessarily large solutions – bounded only by $M^\lambda$ from (10.15c). In order to deal with this issue, in [62], an additional minimization level was introduced in (10.15):

$$
\begin{align*}
Gu - W - Ex_0 &\leq 0 & (10.16a) \\
Gu - W - Ex_0 &\geq -M^u s & (10.16b) \\
\min_\lambda &\lambda^T \lambda & (10.16c) \\
\text{s.t.} & \lambda \geq 0 & (10.16d) \\
& \lambda \leq M^\lambda (1 - s) & (10.16e) \\
& Hu + F^T x_0 + G^T \lambda = 0. & (10.16f)
\end{align*}
$$

This approach has the drawback that it increases the number of auxiliary variables and thus, adds to the computational difficulty. Here, using Proposition 10.1 we can reformulate (10.15) more efficiently by adding constraints which force the selection of only LICQ sets of constraints:

$$
\begin{align*}
Hu + F^T x_0 + G^T \lambda & = 0 & (10.17a) \\
\lambda & \geq 0 & (10.17b) \\
\lambda & \leq M^\lambda s & (10.17c) \\
Gu - W - Ex_0 & \leq 0 & (10.17d) \\
Gu - W - Ex_0 & \geq -M^u (1 - s) & (10.17e) \\
0 & \leq \sum_{i=q_{k-1}+1}^{q_k} s_i \leq \min(q_k, k \cdot m), \sum_{i=1}^{q_1+\cdots+q_k} s_i \leq k \cdot m & (10.17f)
\end{align*}
$$

for any $k = 1, \ldots, N$.

Thus, the computation time is significantly reduced for (10.17) with respect to the original formulation (10.15).

Remark 10.3. The construction (10.17) can be adapted for the case of input-symmetric constraints. Using the first part of Corollary 10.1 the constraints (10.17f) can be further refined.

10.4. Partial recursive implementation of the explicit region

In this section we exploit the structure seen in (10.8) to reduce the computation time for the explicit description of an MPC problem when explicit descriptions of lower order are available. Assume that the explicit representation $\mathcal{U}_{N-\tau}(\cdot)$ is known. Then we can give the next result which offers a partial description of $\mathcal{U}_N(\cdot)$ while avoiding to solve any optimization problem.
Proposition 10.2. Consider a control sequence $u_{[0,\tau-1]}$ which saturates $\tau \cdot m$ linear independent constraints from the blocks $R_1, \ldots, R_\tau$. Then $u_{[0,\tau-1]}$ can be written as $u_{[0,\tau-1]} = \Psi x_0 + \phi$ and the explicit control of degree $N$ is partially given by

$$
\mathcal{U}_N(x_0) = \begin{bmatrix}
\Psi \\
F_{N-\tau}^j (A_\tau + B_\tau \Psi)
\end{bmatrix} x_0 + \begin{bmatrix}
\phi \\
F_{N-\tau}^j B_\tau \phi + g_{N-\tau}^j
\end{bmatrix}, x_0 \in \mathcal{D}_N^i. \quad (10.18)
$$

The feasible domain $\mathcal{D}_N^i$ is computed by introducing (10.18) into (10.3) and the subset $\{j_i\}$ represents the indices taken from $\{1, \ldots, \#\mathcal{I}_{N-\tau}\}$ which correspond to non-empty domains $\mathcal{D}_N^i$. Notations $A_\tau$ and $B_\tau$ stand for $A$ and $[A_1 \ldots B]$, respectively.

**Proof:** Recalling (10.4) and the lower block-triangular structure seen in (10.8) (which is inherited by any ordered selection of active rows $\tilde{G}$), we can partition the matrices associated to active constraints as

$$
\tilde{G} = \begin{bmatrix}
G^\tau & 0 \\
\bullet & \bullet
\end{bmatrix}, \quad \tilde{W} = \begin{bmatrix}
\tilde{W}^\tau \\
\bullet
\end{bmatrix}, \quad \tilde{E} = \begin{bmatrix}
\tilde{E}^\tau \\
\bullet
\end{bmatrix}
$$

where the sub-matrices with index $\tau$ correspond to the first $\tau$ controls (i.e., the sequence $u_{[0,\tau-1]}$). Introducing them into (10.4) we observe that the next relation holds:

$$
\tilde{G}^\tau u_{[0,\tau-1]} = \tilde{W}^\tau + \tilde{E}^\tau x_0.
$$

Since we take $\tau \cdot m$ linear independent rows from the first $\tau$ blocks, $\tilde{G}^\tau$ is square and invertible which allows to write

$$
u_{[0,\tau-1]} = (\tilde{G}^\tau)^{-1} \tilde{E}^\tau x_0 + (\tilde{G}^\tau)^{-1} \tilde{W}^\tau. \quad (10.19)
$$

We observe that under these circumstances, the value of $u_{[0,\tau-1]}$ cannot be the subject of change, regardless of the subsequent values of $u_{[\tau,N-1]}$.

Subsequently, inputs $u_{[\tau,N-1]}$ are to be found by applying the $N - \tau$ order MPC problem. We enumerate all the affine controls (10.18) provided by the explicit representation $\mathcal{U}_{N-\tau}(\cdot)$ with the initial state $A_\tau x_0 + B_\tau u_{[0,\tau-1]}$ and keep the ones which result in non-empty feasible regions $\mathcal{D}_N^i$. \quad \square

**Remark 10.4.** Keeping with the notation of Proposition 10.1 we consider the next constraints:

$$
0 \leq k \leq \min(q_k, k \cdot m), \sum_{j=0}^{k} i_j \leq k \cdot m, 1 \leq k < \tau \quad (10.19a)
$$

$$
0 \leq i_\tau \leq \min(q_\tau, \tau \cdot m), \sum_{j=0}^{\tau} i = \tau \cdot m, \quad (10.19b)
$$

$$
0 \leq i_k \leq \min(q_k, k \cdot m), \sum_{j=0}^{k} i_j \leq k \cdot m, \tau < k \leq N. \quad (10.19c)
$$

All the possible selections which provide candidate sets of active constraints as required in Proposition 10.2 are described by conditions (10.19a)–(10.19b). Further, for a given combination of active
rows from blocks $R_1, \ldots, R_\tau$, condition (10.19c) describes all the subsequent feasible selections of active rows. We note with $\mathcal{F}_N^{\circ, \tau}$ the collection of candidate sets of active constraints which verify (10.19).

One may wonder why, once $u_{[0, \tau-1]}$ is fixed, we don’t simply consider all combinations respecting (10.19c) and test them to see if they correspond to non-empty feasible regions. This approach is wasteful since the number of candidates can be quite large. It is much more “economical” to use the information provided by existing explicit representation of order $N - \tau$.

We use the above results to give a complete working scheme which, starting with a set of known explicit representations $\mathcal{U}_{N-\tau_k}()$ where $1 \leq \tau_1 < \tau_2 \cdots < N$, permits to compute part of the $N$-degree explicit representation $\mathcal{U}_N()$.

Algorithm 3: Computation of the $N$-degree explicit representation of the MPC problem

Input: $\tau_k$, $\mathcal{U}_{N-\tau_k}()$, $\mathcal{F}_{N-\tau_k}$
Output: $\mathcal{U}_N()$
1 $\mathcal{D}_N = \emptyset$, $\mathcal{F}_N = \emptyset$, $i = 0$
2 provide a candidate collection $\mathcal{F}_N^c$
3 foreach $k$ do
4 foreach combination of active rows which verifies (10.19a)-(10.19b) do
5 construct $\Psi, \phi$ as in Proposition 10.2 such that $u_{[0, \tau_k-1]} = \Psi x_0 + \phi$
6 foreach $j = 1 \ldots \#\mathcal{F}_{N-\tau_k}$ do
7 compute $F_N^c, g_N$ and $\mathcal{R}_N$ as in (10.18) using $\Psi, \phi$
8 if $\mathcal{R}_N \neq \emptyset$ then
9 $\mathcal{D}_N = \mathcal{D}_N \cup \mathcal{R}_N$, $\mathcal{F}_N = \mathcal{F}_N \cup \mathcal{R}_N^{-1}(\mathcal{D}_N)$
10 $i = i + 1$
11 end
12 end
13 end
14 update the candidate collection of active constraints: $\mathcal{F}_N^c = \mathcal{F}_N^c \setminus \mathcal{F}_N^{\circ, \tau_k}$
15 solve the rest of the problem using either combinatorial ($\mathcal{F}_N^c$) or state-exploration ($\mathcal{D}_N$) methods;

Algorithm 3 takes as input a set of known explicit representations of lower order. We consider a collection of candidate active sets in step 2 given, e.g., as in Section 10.3. For each $\tau_k$, all the combinations which saturate $u_{[0, \tau_k-1]}$ are considered. Further, to each of these combinations we apply Proposition 10.2 and enumerate all the resulting affine laws and regions. As seen in step 9 we keep the ones which correspond to non-empty regions and update the feasible domain in the state-space (step 10). We update also the collection of candidate active sets by removing the subset treated for a given $\tau_k$ in step 15. At the end of the procedure we reach a partially defined explicit solution of degree $N$. In step 17 we give the partially explored feasible domain/remaining candidate sets of active constraints to an existing algorithm in order to complete the solution. The advantage is of course that we provide a problem which is already partially solved and thus, we reduce the computational load.

Lastly, we remark upon a question which apparently was neglected in the literature: Why should we be interested in all the predicted input values ? Why not consider only the first input (the one that is actually used) and describe the explicit partitioning with respect to it ?

Remark 10.5. All the critical regions whose first $\tau$ inputs are saturating $\tau \cdot m$ constraints from the first $\tau$ blocks (see Proposition 10.2) can be grouped into a merged region. In particular, this means that for $\tau = 1$ we can extract from $\mathcal{D}_N$ regions which have the same initial input $u_0$. This means that

\[\text{114 of 160}\]
we can group the union of regions with the same initial input and thus reduce the complexity of the explicit representation. Such a union of regions is not necessarily convex. However, applying merging techniques (e.g., [52]) it is possible to reduce the total number of regions describing the union.

10.5. Illustrative example

We consider as illustrative example a double integrator [66], described by

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0.3 \end{bmatrix}$$

with constraints $-5 \leq u_k \leq 5$, $\begin{bmatrix} -25 \\ -25 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 25 \\ 25 \end{bmatrix}$. The weight matrices used are $Q = I_2$ and $R = 1$, whereas the prediction horizon $N = 12$ is used, resulting in 46 constraints in the MPC formulation (10.2). We have used the MPT toolbox [89] to obtain the explicit representation as depicted in Figure 10.2.

![Figure 10.2](image-url)

Figure 10.2.: Explicit solution for $N = 12$ and the unions of regions which have the same initial input (indicated by a solid red line).

Figure 10.2 with 51 nonempty domains, by applying Proposition 10.2 we obtain that 32 of them can be obtained recursively from the $N = 11$ degree explicit representation. These domains can be grouped into two distinct unions of regions (as seen in the figure) such that the upper union contains all the regions which have the first optimal control equal with $u_0 = 5$ and the lower union has all the regions with $u_0 = -5$. Further, by merging the unions of regions determined a priori (see Remark 10.5) we can reduce the total number of regions from 51 to 33: each of the unions obtained from Proposition 10.2 had 16 regions which subsequently were merged into 7.

In Table 10.1 we depict the cost difference between solving the mixed integer optimization problem (10.16) and (10.17), respectively, for a prediction horizon ranging from $N = 5$ to $N = 20$ (in both cases we employ the CPLEX solver). We observe that the presence of the constraints (10.17f) reduces the computation time significantly.
Table 10.1.: Time necessary to solve (10.16) and (10.17).

<table>
<thead>
<tr>
<th>N</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard</td>
<td>0.125</td>
<td>0.234</td>
<td>2.07</td>
<td>9</td>
<td>31.4</td>
</tr>
<tr>
<td>LICQ</td>
<td>0.015</td>
<td>0.14</td>
<td>1.47</td>
<td>4.17</td>
<td>14.4</td>
</tr>
</tbody>
</table>

10.6. Conclusions

In this chapter we have analyzed the influences of the structure of the constraints upon the explicit MPC formulation. We observed that we can provide a better upper bound for the number of candidate sets of active constraints and that we can describe a partial recursive relation between successive explicit MPC representation. In both cases we aim to reduce the computation time, either by reducing the number of combinations to be checked or by avoiding lengthy computations for part of the critical regions of the state space. We have also studied the partitioning of the explicit region when the only element of interest is the first element in the sequence of optimal inputs.
11. On the design of exact penalty functions for MPC using mixed integer programming

This chapter develops procedures for the minimizing (according to some norm) of the Lagrange multipliers associated with a given mp-QP problem, assumed to originate from an MPC problem formulation. To this end I exploit the LICQ condition in order to efficiently formulate the optimization problem, and thereby improve upon existing mixed integer formulations and enhance the tractability of the problem. The results are used to design penalty functions such that corresponding soft constraints are made exact, that is, the original (hard) constraints are violated only if there exists no solution where all constraints are satisfied.

The chapter is based on the journal article [C28]:


IF: 3.02 [from JCR 2016, last at submission date]. Q1 (COMPUTER SCIENCE, INTER-DISCIPLINARY APPLICATIONS - 20/105); Q1 (ENGINEERING, CHEMICAL - 28/133) [from JCR 2016, last at submission date].

11.1. Preliminaries

Model Predictive Control (MPC) has been a remarkable industrial success, with thousands of installations worldwide [134]. A distinguishing feature of MPC controllers is the relative ease with which constraints in both states/outputs and inputs are handled. Nevertheless, such constraints may introduce many complexities that an industrial MPC controller needs to address. There has been particular focus on the effect of hard output constraints on stability [177], [116] as well as the use of soft constraint formulations to ensure a feasible optimization problem, see [152], [168], [63] and references therein.

This chapter advances the ideas shown in [62]. Using the KKT conditions derived from the MPC problem we give the conditions which make the penalty function exact for a “soft” set of constraints.
We formulate this condition using polyhedral norms over the Lagrange multipliers of the original problem. This is a step further with respect to the usual approach where only the $\ell_1$ and $\ell_\infty$-norm cases are discussed, see, e.g. [77].

Explicit formulations like the one in [15] provide a piecewise description of the Lagrange multipliers but the calculations quickly become intractable for high dimension and/or a large set of constraints defining the feasible region. Here, we keep the implicit description of the problem and recast it in a mixed integer formulation, that is, we use auxiliary binary variables to linearize the complementarity condition.

This relaxation means that the Lagrange multipliers might not be uniquely defined anymore for a given combination of states/inputs. In [62] this was solved by adding additional minimization subproblems such that the minimum multiplier (after the $\ell_2$ norm) is selected from the space of available solutions. Here, we restrict the selection of active constraints such that only linearly independent combinations are considered (the LICQ condition) which in turn means that the Lagrange multipliers are uniquely determined and there is no need to add another level of minimization. The numerical aspects are also analyzed and we note how and under which conditions numerical errors can be minimized and the computation time reduced.

11.2. Bi-level programming

In bi-level programming, the constraints of the main optimization problem involve the solution of another (lower level) optimization problem:

\[
\begin{align*}
\min_{y, z} & \quad V_U(y, z) \\
\text{subject to} & \quad G_{UI} \leq 0 \\
& \quad G_{UE}(y, z) = 0 \\
& \quad z = \arg \min_z V_L(y, z) \\
& \quad G_{LI}(y, z) \leq 0 \\
& \quad G_{LE}(y, z) = 0
\end{align*}
\]

Bi-level programming has been addressed since the 1970’s, and the survey [32] lists several contributions in the control area going back to the early 1980’s, but due to the inherent difficulty of these problem formulations, they have been used rather sporadically since. However, with increasing availability of computing power, interest in these problems is returning (e.g., [65, 72, 82, 100]).

Assume that the lower level optimization problem admits a unique optimal solution everywhere in its feasible region. Then, it can be replaced by its Karush-Kuhn-Tucker conditions (KKT), resulting in

\[
\begin{align*}
\min_{y, z, \lambda, \nu} & \quad V_U(y, z) \\
\text{subject to} & \quad G_{UI} \leq 0 \\
& \quad G_{UE}(y, z) = 0
\end{align*}
\]
Sets in control. Applications to fault tolerant control and motion planning.

\[ \lambda \geq 0 \]
\[ G_{LI}(y, z) \leq 0 \]  \hspace{1cm} (11.2d)
\[ G_{LE}(y, z) = 0 \]  \hspace{1cm} (11.2e)
\[ \lambda \times G_{LI}(y, z) = 0 \]  \hspace{1cm} (11.2f)
\[ \nabla_z \mathcal{L}(y, z, \lambda, \nu) = 0 \]  \hspace{1cm} (11.2g)

where the “\( \times \)” symbol indicates that the \( k^{th} \) element of the vector \( \lambda \) of Lagrangian multipliers multiplies the \( k^{th} \) constraint in the original lower-level constraints. \( \mathcal{L}(y, z, \lambda, \nu) = V_L(y, z) + \lambda^T G_{LI}(y, z) + \nu^T G_{LE}(y, z) \) is the Lagrangian function of the lower-level problem. Notice that there are no nonnegativity constraints for the Lagrangian multipliers \( \nu \) for the equality constraints.

The difficulty resides in the lower-level problem where the complementarity condition (11.2g) induces nonlinearity. We recall here the technique proposed in [48] to reformulate (11.2) using auxiliary binary variables \( s \in \{0, 1\}^N \) (where \( N \) denotes the number of inequalities in (11.2e)):

\[
\begin{align*}
\min_{y, z, \lambda, \nu} & \quad V_U(y, z) \\
\text{subject to} & \quad G_{UI} \leq 0 \\
& \quad G_{UE}(y, z) = 0 \\
& \quad \lambda \geq 0 \\
& \quad \lambda \leq M^\lambda s \\
& \quad G_{LI}(y, z) \leq 0 \\
& \quad G_{LE}(y, z) = 0 \\
& \quad G_{LI}(y, z) \geq -M^\nu(1 - s) \\
& \quad \nabla_z \mathcal{L}(y, z, \lambda, \nu) = 0
\end{align*}
\]

(11.3a) \hspace{1cm} (11.3b) \hspace{1cm} (11.3c) \hspace{1cm} (11.3d) \hspace{1cm} (11.3e) \hspace{1cm} (11.3f) \hspace{1cm} (11.3g) \hspace{1cm} (11.3h) \hspace{1cm} (11.3i)

where \( M^\lambda \) and \( M^\nu \) are diagonal matrices of appropriate dimensions and with sufficiently large values for the diagonal elements. The nonlinear condition (11.2g) is replaced by conditions (11.3e) and (11.3h) in the sense that they provide the same result as the original complementarity condition through suitable combinations of binary variables \( s \in \{0, 1\}^N \): whenever \( s_i = 1 \), the set of inequalities (11.3f)–(11.3h) degenerates to an equality (i.e., \( G_{LI}(y, z) \)) and, correspondingly, when \( s_i = 0 \), the inequality (11.3h) becomes redundant.

In the following sections we will consider this approach for the particular case of an MPC problem as lower level optimization and the Lagrangian multipliers maximization as the upper level optimization. We will show how these results can be used for the design of an exact penalty function and what improvements in the bilevel optimization problem can be added by considering the special structure of the constraints.
11.3. Problem preliminaries

Let us consider a fairly typical (i.e., LTI dynamics, quadratic cost function and linear constraints) MPC formulation:

\[
\min_{u_0, \ldots, u_{N-1}} \sum_{k=0}^{N-1} (u_k^T R u_k + x_k^T Q x_k) + x_N^T Q_f x_N
\]  
(11.4a)

s.t. \(G_k x_k + H_k u_k \leq b_k, \ k = 0, \ldots, N\)  
(11.4b)

\(x_{k+1} = A x_k + B u_k\)  
(11.4c)

where \(Q \succeq 0, \ Q_f \succeq 0, \ R \succ 0\) are the cost matrices, \(x_0\) is the current state and \(N\) is the length of the prediction horizon.

For compactness of notation, we recast (11.4) as

\[
\min_u \frac{1}{2} u^T H u + x_0^T F u
\]  
(11.5a)

s.t. \(G u \leq W + E x_0\)  
(11.5b)

where \(u = \begin{bmatrix} u_0^T & \ldots & u_{N-1}^T \end{bmatrix}^T\) and matrices \(H, F, W\) and \(E\) are constructed accordingly. We assume for further use that the optimization problem resides in the \(\mathbb{R}^{N \times m}\) space (where \(m\) denotes the dimension of the input) and that the set of constraints has \(q\) inequalities.

We note that the optimization problem (11.5) is convex and regular, admitting a unique optimal solution everywhere in the feasible domain. Then it can be replaced by its Karush-Kuhn-Tucker (KKT) conditions, resulting in:

\[
H u + F^T x_0 + G^T \lambda = 0
\]  
(11.6a)

\(\lambda \geq 0\)  
(11.6b)

\(G u - W - E x_0 \leq 0\)  
(11.6c)

\(\lambda \times (G u - W - E x_0) = 0\)  
(11.6d)

where the “\(\times\)” symbol indicates that the \(k^{th}\) element of the vector \(\lambda\) of Lagrangian multipliers multiplies the \(k^{th}\) constraint from (11.5b).

The MPC formulation shown above is a so called hard constrained problem. There may be initial states \(x_0\) for which there exists no input sequence \(u\) for which the constraints are fulfilled. In such a situation the optimization solver will find no solution and consequently will not provide any input for the plant. This is in general considered unacceptable in industrial practice. Practical MPC implementations therefore include some way of relaxing the constraints to ensure that the optimization problem is always feasible and that the input to the plant is always well defined. There are several ways of doing this \([152]\), one of the simplest and most common is to use soft constraints, as detailed in the next section.
11.4. Soft constraints in MPC

When using soft constraints, the MPC formulation adds variables in the constraint equations which allow relaxing (some of) the constraints, while the optimization cost function includes terms which penalize the constraint violation. Thus, with a soft constraint formulation, (11.4) is replaced with:

\[
\begin{align*}
\min_{u, \epsilon} & \quad \frac{1}{2} u^T H u + x_0^T F u + \phi(\epsilon) \\
\text{s.t.} & \quad G u \leq W + E x_0 + \theta(\epsilon) \\
& \quad \epsilon \geq 0
\end{align*}
\]  

(11.7a)

\[
\epsilon \geq 0
\]  

(11.7b)

where \(\phi(\epsilon) : \mathbb{R}^{d_\epsilon} \to \mathbb{R}\) and \(\theta(\epsilon) : \mathbb{R}^{d_\epsilon} \to \mathbb{R}^N\) represent the penalty function and the constraint function, respectively.

**Remark 11.1.** Naturally, we will soften constraints only if this is physically meaningful and safe to do so. Input constraints are typically hard constraints given by the physics of the process, and it would be absurd to soften such constraints. However, many state/output constraints represent operational desirables (product quality specifications, comfort of operators, etc.), and violating such constraints for some period may be acceptable.

11.4.1. Equivalence between “hard” and “soft” constraints

The conditions of exact correspondence between “hard” and “soft” constraints follow classic optimization results [46] and are detailed in the next proposition.

**Proposition 11.1.** Functions \(\phi(\epsilon)\) and \(\theta(\epsilon)\) assure an exact correspondence between (11.5) and (11.7) whenever \(\epsilon = 0\) if the next conditions are fulfilled:

i) \(\frac{\partial \phi(\epsilon)}{\partial u} |_{\epsilon=0} = 0\),

ii) \(\theta(0) = 0\),

iii) \(\frac{\partial \phi(\epsilon)}{\partial \epsilon} |_{\epsilon=0} \geq \frac{\partial \theta(\epsilon)}{\partial \epsilon} |_{\epsilon=0} \cdot \lambda\).

for any \(\lambda\) verifying (11.6).

\[\square\]

**Proof.** We write\(^1\) the KKT conditions for (11.7)

\[
\begin{align*}
Hu + F^T x_0 + G^T \lambda + \frac{\partial \phi(\epsilon)}{\partial u} &= 0 \quad \text{(11.8a)} \\
\frac{\partial \phi(\epsilon)}{\partial \epsilon} - \frac{\partial \theta(\epsilon)}{\partial \epsilon} \lambda - \mu &= 0 \quad \text{(11.8b)} \\
\lambda &\geq 0 \quad \text{(11.8c)} \\
\mu &\geq 0 \quad \text{(11.8d)} \\
Gu - W - Ex_0 - \theta(\epsilon) &\leq 0 \quad \text{(11.8e)} \\
\epsilon &\geq 0 \quad \text{(11.8f)} \\
\lambda \times (Gu - W - Ex_0 - \theta(\epsilon)) &= 0 \quad \text{(11.8g)}
\end{align*}
\]

\(^1\)As for (11.5), we assume that (11.7) admits a KKT formulation.
\[
\mu \times \epsilon = 0 \tag{11.8h}
\]

and observe that conditions (11.8a), (11.8c), (11.8e) and (11.8g) are equal with (11.6a)–(11.6d) at \(\epsilon = 0\) if and only if items (i)) and (ii)) are verified. In (11.8b) an additional condition on \(\lambda\) with respect to (11.6) appears. In order to allow the same values as in (11.6) for the Lagrange multipliers of (11.7b) it suffices to verify\(^2\) item (iii)), thus concluding the proof.

Obviously, we would like to keep the extended problem (11.7) in a tractable formulation. Consequently, we take

\[
\phi(\epsilon) = F^T \epsilon + \epsilon^T H \epsilon \tag{11.9a}
\]

\[
\theta(\epsilon) = G \epsilon \tag{11.9b}
\]

in order to keep the cost function quadratic and the set of constraints linear. This choice of functions leads to the following corollary.

**Corollary 11.1.** Functions \(\phi(\epsilon)\) and \(\theta(\epsilon)\) written as in (11.9) assure an exact correspondence between (11.5) and (11.7) whenever \(\epsilon = 0\) if the condition

\[
F_{\epsilon} \geq G^T_{\epsilon} \lambda. \tag{11.10}
\]

is fulfilled for any \(\lambda\) verifying (11.6).

**Proof.** The proof is immediate and follows from Proposition 11.1. By replacing (11.10) into Proposition 11.1 we observe that items (i)) and (ii)) are always verified and that item (iii)) becomes (11.10) and remains thus the only condition which needs to be checked.

**Remark 11.2.** Needless to say, for a fixed \(G^T_{\epsilon}\) there exists a matrix \(F_{\epsilon}\) verifying (11.10) only if the feasible region in the space of \(x_0\) is bounded (which implies that the Lagrange multipliers are also bounded).

**Remark 11.3.** Note that in condition (11.10) only the linear term \(F_{\epsilon}\) in the penalty function (11.9) determines whether the soft constraints are exact. The quadratic term \(H_{\epsilon}\) should ensure that the modified problem is a standard QP, but otherwise is held to be of less importance. Typically, the elements of \(H_{\epsilon}\) are therefore small, although some careful choices of the matrix components may influence the trade-off between constraint violations in different variables. This issue will not be pursued any further here. Ensuring that the soft constraints are exact is considered to be of primary importance, and we will therefore focus on the linear term.

In the following we will keep the general definition (11.9) both for its flexibility (a compromise between the extremes illustrated by the \(\ell_1\) and \(\ell_{\infty}\) norms – as shown in the next subsection) and for treating with only one optimization problem (instead of having separate cases for each of the usual norms).

\(^2\)Note that we took into account that all the Lagrange multipliers attached to (11.7c) are greater than or equal with zero since their constraints are active \((\epsilon_i = 0 \rightarrow \mu_i \geq 0)\).
11.4.2. Polyhedral norm formulation

In literature (e.g., [77]), the \( \ell_1 \) and \( \ell_\infty \) norms are frequently used as penalty functions. The condition (11.10) can be written easily for both of these by using the concept of a polyhedral norm.

**Definition 11.1** ([20]). Having a polyhedral set given in half-space representation \( \{ x : (Gx)_i \leq 1, i = 1 \ldots n \} \), its associated polyhedral norm is defined as \( \Psi(G, x) = \max_{i=1\ldots n} ((Gx)_i) \) where \( n \) denotes the number of rows in matrix \( G \).

Additionally, we note that \( F_\epsilon \) is elementwise positive (since there exists a solution of (11.6) for which no constraints are saturated we have that \( \lambda = 0 \) has to verify (11.10)) which, without any loss of generality, allows to write

\[
F_\epsilon = k \cdot 1_{d_\epsilon}. \tag{11.11}
\]

This, together with Definition 11.1 permits to reformulate Corollary 11.1 as follows.

**Corollary 11.2.** Functions \( \phi(\epsilon) \) and \( \theta(\epsilon) \) written as in (11.9) assure an exact correspondence between (11.5) and (11.7) whenever \( \epsilon = 0 \) if the weight \( k \) appearing in (11.11) respects

\[
k \geq \max_\lambda \Psi(G^T_\epsilon, \lambda) \tag{11.12}
\]

for any \( \lambda \) verifying (11.6).

**Proof.** Introducing (11.11) into (11.10) we obtain \( G^T_\epsilon \lambda \leq k \cdot 1_{d_\epsilon} \) which, using Definition 11.1, gives condition (11.12).

**Remark 11.4.** Condition (11.12) encompasses the classical \( \ell_1 \) and \( \ell_\infty \) conditions for exact hard constraints:

- by taking \( \theta(\epsilon) = I_N \cdot \epsilon \) we have that \( \phi(\epsilon) = k \cdot ||\epsilon||_1 \) where \( k \geq \max_\lambda ||\lambda||_\infty \);
- by taking \( \theta(\epsilon) = 1_N \cdot \epsilon \) we have that \( \phi(\epsilon) = k \cdot ||\epsilon||_\infty \) where \( k \geq \max_\lambda ||\lambda||_1 \).

**Remark 11.5.** As it can be seen from Remark 11.4 and as stated in [62], the weight \( k \) on the linear term of the penalty function has to be larger than the maximal value of the dual norm of the Lagrangian multipliers of the corresponding hard-constrained optimization problem. This holds for the more general case of the polyhedral norm, since the norm \( \Psi(G^T_\epsilon, x) \) is dual to the norm associated with the constraint function \( \theta(\epsilon) = \Psi(G_\epsilon, \epsilon) - [181] \).

The \( \ell_1 \) norm penalty function increases the number of decision variables in the optimization problem by the number of constraints that are relaxed. In contrast, the \( \ell_\infty \) norm penalty norm only increases the number of decision variables in the optimization problem by 1 — since the same slack variable can be used for all relaxed constraints. For this reason, \( \ell_\infty \) norm penalty functions are often preferred, although it is shown in [141] that the addition of the \( \ell_1 \) norm optimization variables can be handled at virtually no additional computational cost if problem structure is utilized in the QP solver.

---

\( ^3 \)We make use of the fact that the Lagrange multipliers are positive and we can ignore the \( | \cdot | \) operator which appears in the norm definition:

\[
||a||_p = \left( \sum_i |a_i|^p \right)^{\frac{1}{p}}
\]
On the other hand, the $\ell_1$ norm can result in unexpected behavior and poor performance if it is used to soften an output constraint for which there is an inverse response. In [63] it was shown how to minimize this problem by using time-dependent weights in the optimization criterion.

A sufficiently high value of the linear term in the penalty function will ensure that the soft constraints are exact (see Corollary 11.2). However, a too large term is generally not desirable, since it may lead to unnecessarily violent control should the plant for some reason be outside of the (hard constrained) feasible region. As such, we wish to find the minimal values for which (11.12) is still satisfied with verifying (11.6). This is a non-convex optimization problem which in general has been considered intractable. In the next section we will reformulate conditions (11.6) in order to recast the optimization problem into a tractable MI(L)P problem.

### 11.5. Reformulation of the KKT conditions

From Corollary 11.2, we want to solve the problem

$$\max_{x_0} \Psi(G^T \epsilon, \lambda)$$

(11.13a)

$$\min_u \frac{1}{2} u^T H u + x_0^T F u$$

(11.13b)

s.t. $G u \leq W + E x_0$.  

(11.13c)

Problem (11.13) is a bi-level programming problem where the constraints of the main optimization problem involve the solution of another (lower level) optimization problem. However, it differs from standard bi-level optimization problems in that the upper level objective function is not determined before the lower-level problem is replaced by its KKT conditions (i.e., finding the maximum of the polyhedral norm $\Psi(G^T \epsilon, \lambda)$ in (11.13a) requires finding first the multipliers which satisfy (11.13b-c).

#### 11.5.1. MILP formulation

We proceed by replacing the lower-level problem in (11.13) by its KKT condition, and from (11.6d) realize that the complementarity condition induces nonlinearity. We recall here (11.3) and replace the complementarity conditions with the equivalent linear formulation using the auxiliary binary variables $s \in \{0, 1\}^q$:

$$\max_{\lambda, u, x_0, s} \Psi(G^T \epsilon, \lambda)$$

(11.14a)

$$H u + F^T x_0 + G^T \lambda = 0$$

(11.14b)

$$\lambda \geq 0$$

(11.14c)

$$\lambda \leq M^\lambda s$$

(11.14d)

$$G u - W - E x_0 \leq 0$$

(11.14e)

$$G u - W - E x_0 \geq -M^u(1 - s)$$

(11.14f)

where $M^\lambda$ and $M^u$ are diagonal matrices of appropriate dimensions and with sufficiently large values for the diagonal elements.
Note that the constraints (11.14e) are the constraints of the original MPC problem. Their presence means that we do not have to calculate the feasible region explicitly. This is a major advantage, since the projection operation involved in calculating explicitly the feasible region can be computationally very demanding for large systems.

However, the direct inclusion of \( \lambda \) as free variables in (11.14) allows for unnecessarily large solutions – bounded only by \( M^\lambda \) from (11.14d). In order to deal with this issue, in [62], an additional minimization level was introduced in (11.14). This approach has the drawback that it increases the number of auxiliary variables (due to the additional layer of minimization) and thus, adds to the computational difficulty:

\[
\begin{align*}
Gu - W - Ex_0 &\leq 0 \\
Gu - W - Ex_0 &\geq -M^u(1 - s) \\
\min_{\lambda} & \Psi(G^T_\epsilon, \lambda) \\
s.t. & \lambda \geq 0 \\
\lambda &\leq M^\lambda s \\
Hu + F^Tx_0 + G^T\lambda & = 0.
\end{align*}
\]

Note that the norm appearing in (11.15c) should be the same as the norm which is maximized at the upper level (11.14a), to ensure the accuracy of the solution.

To further reduce the computational effort we explicitly impose that the optimization problem takes into account only linearly independent combinations of constraints (as it will be detailed in the next subsection). This makes the addition of another level of optimization, as in [62] superfluous and permits to significantly reduce the computation time.

### 11.5.2. Explicit imposition of the LICQ condition

Let us recall the following notion characterizing an active set of constraints.

**Definition 11.2 ([165]).** For an active set, we say that the linear independence constraint qualification (LICQ) holds if the set of active constraint gradients are linearly independent.

As long as the set of active constraint gradients (i.e., the subset of rows from \( G \) saturated by the optimum \( u \) corresponding to a given \( x_0 \)) respects LICQ, the KKT problem (11.6) is well-posed in the sense that both the primal solution \( u \) and the dual solution \( \lambda \) are uniquely defined.

In [15], the construction of the explicit PWA control (and thus the determination of the Lagrangian multipliers) is considered only for selections which respect LICQ since only the constraints respecting this qualification determine a full-dimensional region in the state-space. On the other hand, (11.14) considers all feasible combinations of constraints regardless of their linear independence. Therefore, the Lagrangian multipliers will not be always uniquely determined which in turn leads to unnecessarily large solutions.

A first solution was proposed in (11.15) but at the cost of increased complexity. Instead, we aim to exploit the structure of constraint matrix \( G \) and use the information in order to impose that only subsets of constraints which respect LICQ are chosen. For future use let us define this collection of
Chapter 11. On the design of exact penalty functions for MPC using mixed integer programming

This bound and its subsequent refinements give a theoretical upper bound for the number of subproblems that the MI optimization problem may need to solve in a worst-case scenario.

Since we are interested only in candidate sets which respect LICQ it is clear that we can select at most \( N \cdot m \) constraints from the existing \( q \) without violating the linear independence requirement in the \( \mathbb{R}^{N \cdot m} \) space. This leads to an upper bound of

\[
\# \mathcal{J}_N^0 = \sum_{j=0}^{N \cdot m} \binom{q}{j}
\]

(11.16)

candidate subsets. The bound is obtained by selecting from the available \( q \) constraints \( 0, 1, \ldots, N \cdot m \) active constraints successively.

However, the resulting bound is conservative, and a better bound can be computed by making use of the special structure characterizing matrix \( G \): as seen in (11.4b), up to the time instant \( k \) only the initial state \( x_0 \) and the sequence of inputs \( u_0, \ldots, u_k \) appear in the constraints. Algebraically, this means that matrix \( G \) has a lower block-triangular structure:

\[
G = \begin{bmatrix}
R_1 & \cdots & q_1 \times m \\
R_2 & \cdots & q_2 \times 2m \\
\vdots & & \vdots \\
R_N & \cdots & q_N \times Nm
\end{bmatrix}
\]

(11.17)

where each block \( R_k \) describing the \( k \)th order constraints has \( q_k \) rows and \( k \cdot m \) columns (note that \( q_1 + q_2 + \ldots q_N = q \)).

In the following result we give conditions which restrict the number of rows selections from the constraint matrix:

**Proposition 11.2.** Let \( i_k \) denote the number of constraints selected from each block \( R_k \). Then the conditions

\[
0 \leq i_k \leq \min(q_k, k \cdot m), \sum_{j=0}^k i_j \leq k \cdot m, \forall k = 1, \ldots, N
\]

(11.18)

define all the selections of constraints which can be LICQ and the collection \( \mathcal{J}_N^0 \) of candidate active sets is bounded by

\[
\# \mathcal{J}_N^0 = \sum_{(i_1, \ldots, i_N) \text{ verifies } (11.18)} \left( \prod_{j=1}^N \binom{q_j}{i_j} \right).
\]

(11.19)

**Proof.** The reasoning for the conditions in (11.18) is based upon the fact that in a collection of ‘\( y \)’ rows where only the first ‘\( x \)’ elements in each row are non-zero, at most ‘\( \min(x, y) \)’ rows can be selected and still be linearly independent. This property leads directly to the first inequality in (11.18) whereas for the second we observe that the following selection rule applies: from the first \( q_1 \) rows we can select
at most \( m \), from the first \( q_1 + q_2 \) rows we can select at most \( 2 \cdot m \) and so forth until, from the first \( q_1 + \cdots + q_N = q \) rows we can select at most \( N \cdot m \) rows.

The bound (11.19) can be computed by simply counting the candidate sets of active constraints. For a selection \((i_1, \ldots, i_N)\) we take each block \( R_j \) and extract all the combinations of \( i_j \) rows from it, which means \( \binom{q_j}{i_j} \) possibilities. Each of these combinations is then considered with each of the other combinations from the other blocks.

The construction of \( \mathcal{J}_N^q \) was made assuming that any \( i_j \) subset of rows from the block \( R_j \) is linearly independent, assumption which may not hold in practical cases (e.g., when the bounds upon states/inputs are symmetrical). In this sense, consider the next definition:

**Definition 11.3.** We refer to an mpQP problem of the form (11.4) as input-symmetric if the involved constraints can be rewritten as

\[
\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} u - \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} - \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} x_0 \leq 0
\]

where \( G_2 = -G_1 \). Consequently, each block-matrix \( R_j \) with \( q_j \) rows given as in (11.17) can be written as \( R_j = \begin{bmatrix} \hat{R}_j \\ -\hat{R}_j \end{bmatrix} \) where \( \hat{R}_j \) has \( \frac{q_j}{2} \) rows.

Under these circumstances, the following corollary applies.

**Corollary 11.3.** Consider that (11.4) is input-symmetrical and let \( i_k \) denote the number of constraints selected from each block \( \hat{R}_k \). Then, the LICQ constraint selections have to respect relations

\[
0 \leq i_k \leq \min\left(\frac{q_k}{2}, k \cdot m\right), \sum_{j=0}^{k} i_j \leq (k \cdot m), \forall k = 1, \ldots, N
\]

and the collection \( \mathcal{J}_N^q \) of candidate active sets is bounded by

\[
\# \mathcal{J}_N^q = \sum_{(i_1, \ldots, i_N) \text{ verifies (11.21)}} \left( \prod_{j=1}^{N} \binom{q_j}{i_j} \cdot 2^{i_j} \right).
\]

**Proof.** The proof is constructive. Relations (11.21) describe the same conditions as in (11.18) with the difference that at most \( \frac{q_j}{2} \) rows can be selected from a block \( R_j \).

For the bound (11.22) the reasoning is as follows: we select \( i_j \) rows from the \( \frac{q_j}{2} \) rows which define the sub-block \( \hat{R}_j \); subsequently, we consider all the possible sign permutations for the selected constraints (i.e., \( 2^{i_j} \)); the rest of the procedure follows the proof of Proposition 11.2.

**Remark 11.6.** The conditions (11.18) and (11.21) describe an integral polytope (i.e., a polytope whose extreme vertices are integers) and the selections \((i_1, \ldots, i_N)\) are points contained in this polytope. These notions can be described formally using the Ehrhart polynomial and the associated theory [39]. Using these elements it is possible to determine the number of integer points which respect (11.18) or (11.21) and to enumerate them. The actual computation of the Ehrhart polynomial coefficients and
the enumeration of the selections \((i_1, \ldots, i_N)\) can be done with the Ehrhart routines of the PolyLib library (see, [30, 31] for details).

Finally, using Proposition 11.2 we can reformulate (11.14) more efficiently by adding constraints which force the selection of only LICQ sets of constraints:

\[
Hu + F^T x_0 + G^T \lambda = 0 \tag{11.23a}
\]
\[
\lambda \geq 0 \tag{11.23b}
\]
\[
\lambda \leq M^\lambda s \tag{11.23c}
\]
\[
Gu - W - Ex_0 \leq 0 \tag{11.23d}
\]
\[
Gu - W - Ex_0 \geq -M^u(1 - s) \tag{11.23e}
\]
\[
0 \leq \sum_{i=\tau_{k-1}+1}^{\tau_k} s_i \leq \min(q_k, k \cdot m), \sum_{i=1}^{\tau_k} s_i \leq k \cdot m \tag{11.23f}
\]

where \(\tau_k = q_1 + \cdots + q_k\) for any \(k = 1, \ldots, N\) (with the convention that \(\tau_0 = 0\)).

Thus, the computation time is significantly reduced for (11.23) with respect to the original formulation (11.14).

Remark 11.7. Evidently, the construction (11.23) can be adapted for the case of input-symmetric constraints. By using the first part of Corollary 11.3 and the notation of Definition 11.3 we observe that to the constraints of (11.23) an additional condition can be added:

\[
s_1 + s_2 \leq 1 \tag{11.24}
\]

where \(s_1\) and \(s_2\) denote respectively the binary variables associated with \((G_1, E_1, W_1)\) and \((G_2, E_2, W_2)\). The condition simply states that any two binary variables which correspond to the same row from \(G_1\), respectively \(G_2\) cannot be “1” simultaneously.

Remark 11.8. Note that even if structurally a set of constraints can be linearly independent (i.e., it respects the selection constraints given in Proposition 11.2 and Corollary 11.3), it may happen that the actual constraints selected do not respect LICQ. In this case, the formulations (11.23) and (11.24) will fail. Whenever the constraints are input-symmetric and conditions (11.23) are applied, the problem still gives a correct solution since even if non-LICQ sets of constraints are possible, it is not possible that both can be saturated for the same input and initial state.

Finally, we can apply formulation (11.23) into the maximization problem (11.12):

\[
k^* = \max_{\lambda, u, x_0, s} \max_{j=1, \ldots, d} (G^T_s \lambda)_j \tag{11.25a}
\]
\[
s.t. \quad Hu + F^T x_0 + G^T \lambda = 0 \tag{11.25b}
\]
\[
\lambda \geq 0 \tag{11.25c}
\]
\[
\lambda \leq M^\lambda s \tag{11.25d}
\]
\[
Gu - W - Ex_0 \leq 0 \tag{11.25e}
\]
\[
Gu - W - Ex_0 \geq -M^u(1 - s) \tag{11.25f}
\]
Sets in control. Applications to fault tolerant control and motion planning.

\[ 0 \leq \sum_{i=k_{t-1}+1}^{\tau_k} s_i \leq \min(q_k, k \cdot m), \sum_{i=1}^{\tau_k} s_i \leq k \cdot m \]  

(11.25g)

with \( d_{\epsilon} \) denoting the number of rows in \( G_{\epsilon}^T \). The second “max” operator comes from the way the polyhedral norm \( \Psi(\cdot, \cdot) \) is defined.

In the general case when \( G_{\epsilon} \) is not a vector (e.g., for the \( \ell_1 \) norm we have \( G_{\epsilon} = 1_N \), as seen in Remark 11.4) we need to recast (11.25) as follows:

\[
k^* = \max_{j=1\ldots d_{\epsilon}, \lambda, u, x_0, s} \max_j (G_{\epsilon}^T \lambda)_j \\
\text{s.t.} \quad Hu + F^T x_0 + G^T \lambda = 0 \\
\quad \lambda \geq 0 \\
\quad \lambda \leq M^\lambda s \\
\quad Gu - W - Ex_0 \leq 0 \\
\quad Gu - W - Ex_0 \geq -M^u(1 - s) \\
0 \leq \sum_{i=k_{t-1}+1}^{\tau_k} s_i \leq \min(q_k, k \cdot m), \sum_{i=1}^{\tau_k} s_i \leq k \cdot m
\]  

(11.26a)

(11.26b)

(11.26c)

(11.26d)

(11.26e)

(11.26f)

(11.26g)

where we switched the order of the “max” operators and basically have to compute \( d_{\epsilon} \) lower level maximizations in order to solve the overall problem.

Lastly, recall the relations (11.10) and (11.11). We note that the maximization in (11.12) is a linear program and can thus be replaced by its dual. It follows then that an alternative formulation of Corollary 11.2 exists:

\[
k^* \leq \min_{G_{\epsilon}^T \lambda \leq k \cdot 1} \lambda
\]  

(11.27)

for any \( \lambda \) verifying (11.6).

Applying the same line of reasoning as for (11.25) we obtain the following “max-min” problem:

\[
k^* = \max_{u, x_0, s} k^o \\
\text{s.t.} \quad Hu + F^T x_0 + G^T \lambda = 0 \\
\quad \lambda \geq 0 \\
\quad \lambda \leq M^\lambda s \\
\quad Gu - W - Ex_0 \leq 0 \\
\quad Gu - W - Ex_0 \geq -M^u(1 - s) \\
0 \leq \sum_{i=k_{t-1}+1}^{\tau_k} s_i \leq \min(q_k, k \cdot m), \sum_{i=1}^{\tau_k} s_i \leq k \cdot m \\
k^o = \min_k k \\
\text{s.t.} \ G_{\epsilon}^T \lambda \leq k \cdot 1.
\]  

(11.28a)

(11.28b)

(11.28c)

(11.28d)

(11.28e)

(11.28f)

(11.28g)

(11.28h)

(11.28i)

Note that in the lower-order problem (11.28h)–(11.28i) the only variable is ‘k’ (a slack variable) whereas ‘\( \lambda \)’ is only a parameter, since the upper-order problem (11.28a)–(11.28g) uniquely defines for a given
triplet \((u, x_0, s)\) the associate Lagrangian multipliers (due to the presence of the LICQ conditions). Consequently, we reformulate (11.28) where the optimization (11.28h)–(11.28i) is replaced by its KKT conditions (where the complementarity condition is again linearized through the addition of auxiliary binary variables \(\nu \in \{0, 1\}^d\)):

\[
\begin{align*}
k^* &= \max_{u, x_0, s, \nu, k^0} k^0 \\
\text{s.t.} \quad & H u + F^T x_0 + G^T \lambda = 0 \\
& \lambda \geq 0 \\
& \lambda \leq M^\lambda s \\
& Gu - W - Ex_0 \leq 0 \\
& Gu - W - Ex_0 \geq -M^u(1 - s) \\
& 0 \leq \sum_{i=\tau_k-1+1}^{\tau_k} s_i \leq \min(q_k, k \cdot m), \quad \sum_{i=1}^{\tau_k} s_i \leq k \cdot m \\
& 1 - 1^T \cdot \delta = 0 \\
& \delta \geq 0 \\
& \delta \leq M^\delta \nu \\
& G^T_\epsilon \lambda - 1 \cdot k \leq 0 \\
& G^T_\epsilon \lambda - 1 \cdot k \geq -M^\delta(1 - \nu).
\end{align*}
\] (11.29a)

With this last formulation we have two alternatives, (11.26) and (11.29), for solving the original optimization problem (11.13). It remains at the latitude of the reader to choose between these representations, depending on the specifics of the problem at hand. For example, in the case of the \(\ell_1\) norm, the formulation (11.26) has a slightly more compact form whereas for the \(\ell_\infty\) norm, the formulation (11.29) proves superior.

### 11.5.3. Numerical considerations

Some conditions need to be imposed on the numerical values of parameters \(M^\lambda\) and \(M^u\) such as not to alter the original result (we use as basis for this analysis (11.23) without any loss of generality since the other variants can be treated similarly). These conditions can be summarized as follows:

i) \(M^u\) sufficiently large such that the discarded inequalities from (11.23e) do not influence the ones in (11.23d);

ii) \(M^\lambda\) sufficiently large such that there exists a \(\lambda\) permitting an optimum solution in (11.23) identical to one the which can be obtained in (11.6);

iii) \(M^u\) and \(M^\lambda\) small enough such that numerical problems are minimized and/or avoided.

Item (iii)) forces us to take the minimal values of parameters \(M^u\) and \(M^\lambda\) which respect the requirements given in items (ii)) and (iii)).

Item (i)) has a simple geometrical meaning: whenever an inequality from (11.23e) is discarded (the associated binary variable is \(s_i = 0\)), it should be redundant with respect to the feasible region defined
by the constraints in (11.23d). Without entering into details, this can be tested using a variant of the Farkas lemma (as explained in [170]) which leads to a LP problem whose solution gives the minimal $M^u$.

For item (ii)) the reasoning applied to the first item cannot be applied. The solution is to solve (11.23) iteratively: as long as the $\lambda$ found as result of the optimization problem saturates any of the constraints (11.23c) we increase $M^\lambda$ and repeat the computation of (11.23). We mentioned that the various elements of $M^\lambda$ can be scaled independently, only increasing the ones which correspond to variables saturated by constraints (11.23c).

It is worth noting that by adding constraint (11.23f) we reduce the complexity of the problem. A MI problem might, in the worst case scenario, pass through all the admissible combinations of binary variables. For conditions (11.14) this means that at most $2^N$ sub-problems may need to be solved. For comparison, in the case of (11.23), the maximal number of sub-problems is less than $\sum_{i=0}^{d} (\binom{N}{i})$.

### 11.6. Illustrative examples

We consider here three examples, from [66], [64] and [63], respectively.

We consider the first, a double integrator, to be described by

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0.3 \end{bmatrix}$$

with constraints

$$-1 \leq u_k \leq 1, \quad \begin{bmatrix} -5 \\ -5 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}.$$ 

The weight matrices used are $Q = I_2$ and $R = 1$, whereas the prediction horizon $N = 15$ is used, resulting in 58 constraints in the MPC formulation (11.5) (and the same number of binary variables in the optimization formulation (11.23)).

We choose here to particularize the optimization problem (11.23) to the $\ell_\infty$-norm, as shown in the second item of Remark 11.4. That is, $\theta(\epsilon) = 1_N \cdot \epsilon$ and $\psi(\epsilon) = k \cdot \epsilon$ with $\epsilon \in \mathbb{R}$. Maximizing the $\ell_1$-norm of the Lagrangian multipliers, we find that the maximum is achieved at $x_0^* = \begin{bmatrix} 9 \\ -3 \end{bmatrix}$ (as shown in Figure 11.1) and the corresponding value of the $\ell_1$-norm and thus of the penalty function weight is $k = ||\lambda||_1 = 911.2465$. This value, and the location of the maximum is verified by solving the MPC problem at all vertices of the feasible region.

In Figure 11.1 we illustrate the feasible domain of $x_0 \in \mathbb{R}^2$ for problem (11.5) – the region filled with red, and contrast it with projections of the unbounded feasible domain characterizing the soft problem (11.7) – contours drawn with blue for different bounds upon $\epsilon$. For a meaningful comparison we have projected along $\epsilon = 0.75$ and $\epsilon = 2$ values (e.g., inside the contour defined by $\epsilon = 2$, the penalty cost is less than 911.2465 · 2).

In Figure 11.2 we depict the cost difference between solving optimization problems (11.15) and (11.23), respectively, for a prediction horizon ranging from $N = 2$ to $N = 20$. As expected, we observe that the presence of the constraints (11.23f) in (11.23) reduces the computation time significantly.

Lastly, in Figure 11.3 we compare the trajectories generated by the soft/hard constrained MPC. As expected we note that inside the feasible region the trajectories coincide and that in the case of an
Chapter 11. On the design of exact penalty functions for MPC using mixed integer programming

Figure 11.1: Feasible region of problem (11.5) and the point where the $\ell_1$-norm of the Lagrangian multipliers is maximized.

Figure 11.2: Time necessary to solve problem (11.15) – in solid blue, vs. for solving problem (11.23) – in dashed red.

Infeasible starting point, we can still compute a trajectory.

The second example is taken from [64]. The discrete-time model is given by

$$A = \begin{bmatrix} 2 & -1.45 & 0.35 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$
Sets in control. Applications to fault tolerant control and motion planning.

Figure 11.3: Depiction of trajectories generated with/out soft constraints (dashed red and solid blue respectively).

\[ C = \begin{bmatrix} -1 & 0 & 2 \end{bmatrix}, \quad D = 0 \]

and the constraints are given by

\[ -1 \leq Cx \leq 1, \quad -1 \leq u \leq 1. \]

The state weight is given by \( Q = C^T C \), the input weight is \( R = 10 \), and a prediction horizon \( N = 5 \) with a terminal set is used.

We apply the penalty and cost function to the original problem and obtain a soft-MPC in both \( \ell_1 \) and \( \ell_\infty \) formulation when starting from an infeasible state \( (x_0 = \begin{bmatrix} 1.5 & 1.5 & 21.5 \end{bmatrix}^T) \). For illustration, we depict in Figure 11.4 the resulting output and corresponding input values for both norm implementations. As it can be seen, for these particular dynamics, the use of the \( \ell_1 \) norm (solid blue line) in the penalty function results in a large overshoot whereas the \( \ell_\infty \) norm (dashed red line) has a smaller overshoot but takes longer to converge to the origin. For illustration purposes we have used different scales in Figure 11.4 (b) and (c) but it can be seen that the \( \ell_\infty \) input has a significantly smaller magnitude - but displays strong oscillations. A technique to mitigate such oscillations was proposed in [64]. Finally, the inputs resulting from the use of the \( \ell_1 \) norm behave better, but this is to be traded with the additional difficulty of solving online a larger MPC problem. The issue can be alleviated by the use of special-purpose qp solvers as the one described in [141] which is able to efficiently utilize the problem structure.

The third and last example is taken from [63]. The discrete-time model is given by

\[
A = \begin{bmatrix}
.928 & .002 & -.003 & -.004 \\
.041 & .954 & .012 & .006 \\
-.052 & -.046 & .896 & -.003 \\
-.069 & .051 & .032 & .935 \\
\end{bmatrix}, \quad B = \begin{bmatrix}
.000 & .336 \\
.183 & .007 \\
.090 & -.009 \\
.042 & .012 \\
\end{bmatrix}
\]
Chapter 11. On the design of exact penalty functions for MPC using mixed integer programming

Figure 11.4.: Depiction of inputs/outputs generated with soft constraints for a $\ell_1$ and $\ell_\infty$-norm penalty functions when starting from an infeasible point.

\[
C = \begin{bmatrix}
0.000 & 0.000 & -0.098 & 0.269 \\
0.000 & 0.000 & 0.080 & 0.327
\end{bmatrix}, \quad D = \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}
\]

and the constraints are given by

\[
\begin{bmatrix}
-1 \\
-1
\end{bmatrix} \leq C x \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix}
-1 \\
-1
\end{bmatrix} \leq u \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}.
\]

The state weight is given by $Q = C^T C$, the input weight is $R = I_2$, and a prediction horizon $N = 10$ is used. This problem has 116 constraints in the MPC formulation, and hence requires 116 binary variables in the MILP formulation for calculating $||\lambda||_1$. We find that the maximum value of the norm is achieved at $x = \begin{bmatrix}
25.5724 & 25.3546 & 9.7892 & 0.2448
\end{bmatrix}^T$, and has the value $||\lambda||_1 = 38907$. For this example, calculating the feasible region is subject to numerical errors, and the result has therefore not been verified by checking the vertices of the feasible region. We have executed a partial verification by finding the active constraints which correspond to the $\lambda$ given by the LICQ problem and using them to describe the Lagrangian multipliers over the current critical region. Then, we took all the vertices of the critical region and selected from them the Lagrangian with the highest $\ell_1$ norm.
11.7. Conclusions

In this chapter, procedures for calculating the maximum values of the polyhedral norm of the Lagrangian multipliers of standard mp-QP problems have been developed. The procedures are intended for designing penalty functions for soft constraints in MPC, to find the required weights for making the constraints exact. The calculation procedures are formulated as MILP problems, which are known in general to be NP-hard and thus very computationally demanding to solve. By forcing the selection of active constraints which are LICQ we simplify the problem and reduce the computation times.
Part III.

Open issues and future directions
12. Open issues in the state of the art

One of the chief motivations of continued interest in control topics lies in the obvious observation that many well-known algorithms in robust control become exponentially more difficult to implement when applied to complex systems. “Complex” stands here as a byword for any system which is in some sense difficult to solve (e.g., distributed systems with heterogeneous parts, strongly non-linear dynamics, communication delays and so forth). The result is that for some problems albeit the existence of a solution is claimed and proved in a general setting, analytic or sound numerical constructions are merely provided under confining hypotheses.

12.1. Set theoretic issues for fault tolerant control

While complex dynamics are in general difficult to tackle, most models coming from real-life (like infrastructure management and power generation industries) do not obey “nice” assumptions and, not infrequently, are the most non-standard imaginable cases from a theoretical standpoint.

In what follows we identify two major problems which to a large extent are open in the literature and who particularly suffer when considered for complex systems.

12.1.1. Numerical and theoretical difficulties in set constructions related to control problems

Set-theoretic methods have become in the last decade(s) a mainstay in a large number of control areas [19, 40, 54]. In particular, invariance and stability analysis [22, 170], reachability analysis [49, 85, 107, 169], robust state estimation [5, 129], control of hybrid systems [12], navigation, localization and trajectory tracking [121] have benefited greatly. Historically, the ellipsoidal sets [86] were the tool of choice due to their simple definition; closer to the present, the polyhedral sets [22, 181] emerged to the fore due to their dual representation (as developed in the Bronn-Minkowski algebra [35, 150]) and reliable numerical algorithms.

While conceptually powerful, set-theoretic notions have a somewhat reduced presence due to:

i) In many areas (robust estimation, constrained optimization, robust invariance) the theory is incomplete or amounts to impractical implementations due to confining assumptions.

ii) The “dimensionality curse”: for large dimensions the numerical algorithms become unreliable or have a prohibitive resolution time.

Current approaches in the literature have several important limitations:

i) Theoretical results and their implementations concentrate on ellipsoids, polyhedrons or spectahedra which are not well-suited to non-convex set-characterizations.

ii) Solutions are often reached iteratively [139] or by making conservative assumptions [80]. Algorithms for non-standard dynamics (switched, with delays or distributed) require either strong simpli-
fications (convex boundings, linearizations) or are too general to be of any practical use [13].

iii) Many operations involving sets become unreliable or are simply impossible to implement at larger dimensions (set projection [73], Minkowski addition, etc.).

iv) Invariance, reachability or robust estimation generally assume bounded parameter variation, perturbations and disturbances and neglect that these are usually stochastic [22].

12.1.2. Fault tolerant control design for complex systems

Due to the proliferation of cheap sensors, increase in complexity and the emerging intelligence of day-to-day appliances the chances of fault occurrences which might not only degrade the nominal performance but lead in fact to catastrophic failures have greatly increased [90, 112, 140, 160]. In many of these situations the systemic failures occurred due to unchecked fault propagation, use of faulty information or human error even if the system was still salvageable. Hence, fault tolerant control (FTC) strategies [25, 178] are required to attenuate/cancel the negative effects of a fault.

Active FTC reacts to a detected fault and reconfigures the control actions so that the stability and performances specifications are kept. Fault detection and isolation (FDI) is done via a multitude of methods (various Kalman filter variants, parameter estimation, parity equations, etc.) and reconfiguration of the control (RC) is done either by using a pre-computed law [179] or by synthesizing a new one on-line [124]. Albeit reduced with respect to the mainstream, set-membership approaches have made a breakthrough in recent years [102, 129]. In particular, invariant set constructions [154, 155], [J13] have reduced the online computations and offered stability guarantees [J12].

Several difficulties are apparent:

i) Fault detection and control reconfiguration are difficult to consider simultaneously (usually one of them is assumed to work flawlessly, i.e., missed faults, false alarms or delays are ignored).

ii) Subject to the problem’s structure, localized detection and reconfiguration need to be considered (i.e., limitations on the controllability and observability of the problem).

Current approaches in the literature have several important limitations:

i) Fault propagation and subsequent corrective measures are difficult to estimate prior to the fault occurrence in complex systems (due to problem structure and parameter variation [70, 171, 172]).

ii) Distributed control strategies (e.g., via multi-agent formulations) do not usually have stability and performance guarantees (consensus is sought usually only for nominal functioning [92]).

iii) Set-membership constructions require conservative set approximations or offline complex computations. Detection guarantees imposed in the control design lead to non-convex constraints [69].

iv) Fault detection and isolation depends on sensor placement which is difficult to ascertain optimally in distributed/hierarchical systems [135].

12.2. Issues for motion planning strategies

Many tasks in today’s industrial practice are either expensive, dangerous, unsuited or downright impossible to handle by a human operator. Consequently, (semi)autonomous control strategies which tackle complex situations, are aware of the environment and, in general, are reliable with respect to various disturbances, need to be considered [76].

A prime example of these challenges is the field of motion planning trajectory / path tracking. Aerial,
maritime or ground robots are already utilized for critical tasks such as infrastructure surveillance, damage assessment or victim search. We consider that this field is ripe for approaches which exploit offline design with flexible control reconfiguration strategies under realistic conditions. In particular, there are still open issues with obstacle avoidance, trajectory reconfiguration (either as selection from a predefined bundle of trajectories or via on-the-fly generation), model variation (either due to “normal” disturbances or faults affecting the dynamics). Moreover, it is often the case that a task cannot be (efficiently) solved by a single vehicle, rather a team of autonomous vehicles has to collaborate such that the mission is accomplished. This leads to discussions about task assignment, distributed control and communication issues (line of sight, variable delays, etc.) [117].

In what follows we identify three major problems which, to a large extent, are open in the literature and which particularly suffer when considered under challenging operational constraints.

12.2.1. Numerical issues of trajectory generation for nonlinear dynamics

While offline trajectory generation procedures seem a straightforward choice (i.e., the difficult part is relocated offline), several numerical issues have reduced their appeal throughout the community:

i) The “dimensionality curse”: for large dimensions the numerical algorithms become unreliable or have a prohibitive resolution time (e.g., for stability analysis and constrained optimization).

ii) The design procedures reduce to difficult (nonlinear cost and non-convex feasible domain) optimization problems. Subsequent computations are cumbersome and prone to numerical errors.

Current approaches in the literature have several important limitations:

i) Stability and performance guarantees can be tackled via robust positive / controlled invariance notions [24] but these are again sensitive to dimension and handle badly nonlinear dynamics.

ii) Collision avoidance, target tracking, formation control, lead to non-convex constraints which require either nonlinear solvers or mixed-integer formulations (cumbersome to formulate and solve).

iii) Heuristic approaches are often used and dispatch some of the numerical issues but usually lack convergence guarantees and are difficult to tune (e.g., parameters for fuzzy control).

12.2.2. Discretization issues for trajectory generation and subsequent tracking

Typically, the constrained optimization problem resulting from a design procedure is first discretized and then solved. This introduces equivalence issues between the discrete and “real” continuous cases:

i) Constraints and costs are usually sampled along the trajectory thus either requiring conservative assumptions or leading to imprecise formulations which cannot handle runtime functioning.

ii) Often there is a disparity between the sampling time of the trajectory and that of the runtime controller which assures stability and trajectory tracking.

Current approaches in the literature have several important limitations:

i) Some effort has been put into showing that discrete constraints can provide sufficient guarantees for the continuous model (e.g., the “corner cutting” problem [144] or safety regions around agents).

ii) Hierarchical control strategies have been proposed for systems with multiple sampling scales but usually the upper level ignores or greatly simplifies the closed-loop dynamics of the lower level.

iii) Order reduction and discretization methods are employed but these do not take into account the trajectory generation requirements.
12.2.3. Reliable trajectory generation under model variation and disturbances

Usually a trajectory is generated for a nominal model (known, with no disturbances, etc). This makes the construction fragile from several points of view:

i) Guarantees of trajectory feasibility, constraint validation and cost optimization in the design stage are difficult to consider explicitly (especially perturbations and model variations) [42].

ii) Large model variations (e.g., mode switching due to fault occurrences which change the dynamics) make the trajectory either impossible to follow or reduce unacceptably the performance.

iii) Disturbances and measurement noises impede trajectory tracking. Additive errors lead to increasing tracking errors which in the end require to recompute the reference trajectory from scratch.

Current approaches in the literature have several important limitations:

i) Procedures which implicitly guarantee feasibility [164] cannot easily account for constraint validation and cost reduction. Discrete constrained optimization problems (e.g., model predictive control) explicitly account for constraints and cost but are sensitive to dimension and dynamics.

ii) Fault detection and control reconfiguration are not usually considered simultaneously [26]. One or the other is assumed to work flawlessly (i.e., missed faults, false alarms or delays are ignored) thus leading to unrealistic trajectory reconfiguration strategies.

iii) The strategies usually employed are “robust” rather than “adaptive”: the trajectory can withstand certain variations via conservative limitations (e.g., tighter input bounds) but it is not parametrized explicitly after the varying parameters nor are structured uncertainties usually considered.
13. Future directions

Based on the issues highlighted in Chapter 12 I propose several directions of interest.

13.1. Improvements in set descriptions. Numerical issues

Let us consider dynamics \( x^+ = f(x, u, \delta) \), assume bounds on state, inputs and disturbances \( (x \in \mathcal{X}, u \in \mathcal{U}, \delta \in \Delta) \) and a starting set \( x \in R_0 \). Then I can define recursively the \( k \)-step reachable set \( R_k = f(R_{k-1}, \mathcal{U}, \Delta) \) which is useful in feasibility analysis and robust set-estimations. Robust positive invariance (RPI) and robust control led invariance (RCI):

\[
x \in \Omega \rightarrow f(x, k(x), \Delta) \in \Omega, \quad x \in \Omega \rightarrow \exists u \in \mathcal{U} \text{ s.t. } f(x, u, \Delta) \in \Omega
\]

characterize the stability and controllability of the dynamics. The minimal/maximal RPI (mRPI/MRPI) sets are also of interest. All these set constructions are fairly easy to write and between themselves largely cover the usage of sets in control theory. Nonetheless, any slight change in the assumptions has tremendous influence in the numerical implementations or even the mathematical descriptions.

As detailed earlier there are still gaps in the theory and application of robust set-estimation, reachability and invariance. “No-go” zones are complex dynamics (nonlinear, switched, with delay), stochastic disturbances, large dimensions, piecewise affine control, etc. I detail below how I propose to tackle these issues and improve the state of the art.

13.1.1. Robust estimation and invariance constructions for complex systems

I plan to provide analytic (as opposed to recursive) descriptions for reachable and minimal RPI sets (and maximal RPI sets using duality properties [114]). I have preliminary results for discrete-time systems with zonotopic disturbances and plan to expand them to the continuous case (using null-controllability results [67, 68]) and to prove equivalence and generalization to ultimate bounds constructions [80]. Lastly, I will expand these issues to controlled invariance (e.g., “given a set of constrains find both the fixed gain and the shape which can be made robustly invariant by the gain feedback”, one of the few places where ellipsoidal sets still have the ‘upper hand’ [110]).

Having analytic constructions codified in compact representation (for which zonotopic sets excel) will open large-scale systems to set approaches. The other half of the coin is the description of non-convex regions (coming from either nonlinear dynamics or opposing constraints). One way to characterize such regions is via disjunctive constraints (union of convex sets) which can be modeled as mixed-integer problems. Unfortunately, this framework is sensitive to dimension and number of decision variables [120, 173]. The best way to reduce the computational burden is to provide a compact description of the region of interest. I have preliminary results with hyperplane arrangement.
decompositions [J11], [C10] and plan to expand them to parametrized hyperplane arrangements and improve upon merging procedures by using Boolean algebra notions [52].

Another direction envisaged is represented by star-shaped sets [108] which are inherently non-convex but preserve enough structure to be of practical use. They have appeared in non-smooth optimization [148] but otherwise are relatively unused in control theory and certainly, not for characterization of non-convex regions. I propose to use these sets for efficient representations by exploiting some of the deeper notions treated in the literature: i) set separability for star-shaped bodies [146, 157]; ii) star-shaped distance and its minimization with respect to another set [147].

13.1.2. Extensions to chance constrained dynamics

The previous results assume bounded disturbances which is not always a reasonable assumption. As such, I plan to adapt the set estimation, reachability and invariance notions discussed a priori to the stochastic case, i.e., to take into account ‘chance constraints’. A typical optimization problem can be formulated as:

$$\min_x c^T x, \quad \text{s.t. } P(Ax \geq \zeta) \geq 1 - \tau \text{ and } x \in \mathcal{X},$$

where a cost is minimized over chance constraints ($Ax \geq \zeta$ constraints have a probability $\tau$ of not being valid).

I am interested in reformulations of chance-constrained programs with joint probabilistic constraints for a finite discrete distribution [96, 149] and by probabilistic set coverings [17] in the context of mixed integer reformulations [61, 83]. In addition, I plan to generalize probabilistic invariance notions [79, 130] to arbitrary-good approximations of the minimal RPI set (assuming zonotopic constraints).

13.1.3. Elements of originality and innovation. Impact

The directions proposed in this manuscript go beyond the state of the art in several important ways. The emphasis on analytic descriptions for sets and on classes of sets which are relatively unknown in the control community (zonotopes and star-shaped sets) will broaden the appeal of set-theoretic methods. Additionally, compact hyperplane arrangement descriptions via mixed integer programming (with parameterizations for increased flexibility) will allow to characterize non-convex regions efficiently. Not in the least, the expansion to chance constraints will bring immediate benefits through the generalization of invariance notions.

Impact

The expected improvements in set constructions for complex systems (with nonlinear dynamics, large dimension, distributed components) will have a large impact in the control community as a whole by making theoretical algorithms numerically tractable and by opening new avenues in the characterization of non-convex bodies and chance constraints.

13.2. Fault tolerant control with set-theoretic methods

For the fault detection and isolation (FDI) part of a fault tolerant control (FTC) scheme the vast majority of the model-based methods rely on probabilistic approaches. In contrast, what I propose
here is to use set-theoretic methods to define healthy and faulty functioning \([J13], [155]\): take a residual signal \(r_i\) associated to a fault \(f_i\) and consider the “healthy” \(R^H_i\) and “faulty” \(R^F_i\) residual sets characterizing it. As long as a separation condition (partially) holds, the FDI mechanism can be characterized through set inclusions:

\[
R^H_i \cap R^F_i = \emptyset \quad \Rightarrow \quad r_i \in R^H_i \xrightarrow{\text{FAULT}} r_i \in R^F_i.
\]

Building upon these constructions I can tackle the issues highlighted earlier. In particular, complex systems which might be geographically distributed and have heterogeneous components will pose problems in the analysis and design of fault detection and isolation, control reconfiguration and fault accommodation.

**13.2.1. Fault detection and isolation analysis**

I plan to provide a complete set framework for residual signals characterization and for subsequent fault detection and isolation conditions. First of all, I aim to provide extended residual formulations (rather than asymptotic observers or parity equations). I will employ moving horizon estimation with a horizon long enough to recover the entire state but short enough to discard earlier fault occurrences. I will consider the system’s structure in conjunction with sensor placement such that state and faults are observable in both nominal and faulty functioning (expand on graph theoretic results \([135]\)).

Overall, I plan to improve the existing set approaches by redefining the underlying sets (zonotopic constraints, non-convex representations, chance constraints, etc) such that I can efficiently estimate the fault model’s parameters when they are unknown or within an incertitude interval.

Finally, I will analyze non-zero fault detection and isolation times. This is a major drawback for invariant set constructions as either the invariance properties are broken or artificial control delays have to be imposed. I envisage calculating upper bounds for the delays appearing in detection/isolation \([154]\), using them to provide robust detection and isolation guarantees and, lastly, to take into account explicitly the transitory behavior of the system while the fault is not yet detected/isolated.

The carried analysis will allow to either characterize exact fault detection conditions (as separation conditions upon bounded sets) or to provide probabilities of missed/false fault alarms (e.g., in terms of probabilistic invariant sets).

**13.2.2. Control reconfiguration and fault accommodation**

Using the information provided by the FDI block I will consider the post-fault model and analyze its characteristics (e.g., controllability and observability) in order to design adequate post-fault control laws or to reconfigure the control online. Depending on the characteristics of the post-fault model I can mitigate the fault effects (possible if there is enough redundancy in the system), conserve stability with degraded performance (if the system is still controllable and observable) and lastly, design graceful degradation strategies (if the system is irrevocably damaged) \([25]\).

I plan to consider explicitly the detection and reconfiguration blocks such that I generalize the fault tolerant control problem to a dual optimization problem \([43, 45]\) where the detection and isolation constraints provide persistent excitation to the residual signals \([57]\). With extended residual formulations I have set separation conditions which are parametrized (references, feedback control, etc).
etc) and which can be used in control design:

\[
\min_{u,x_0,p} \mathcal{G}(x,u), \quad \text{s.t.} \quad R^H_i(p) \cap R^F_i(p) = \emptyset.
\]

These degrees of freedom allow to explore various implementations: i) fixed gain design; ii) feed-forward control where a reference governor provides persistent references (such that the threshold of detection for the residual signal is above noise/parameter variation); iii) an integrated control where the fault detection and isolation conditions are part of the feedback control design (e.g., via moving estimation + predictive horizon implementations).

If, in addition I assume that the FDI block is unreliable, the control algorithms have to: i) withstand variation of the fault parameters, missed faults and so forth; ii) be able to minimize the “un-reliability” of the FDI mechanism.

I will consider realistic situations in which bounded/stochastic noises and perturbations appear as well as state and input constraints. These will lead to feasibility limitations which will have to be included in the control design. in particular, non-convex constraints (appearing from the persistence of excitation conditions) will be interesting/challenging.

### 13.2.3. Elements of originality and innovation. Impact

The use of set-theoretic tools will provide a coherent framework for fault tolerant control strategies for complex systems. Unlike the classical case were usually one of the tolerant control mechanisms (fault detection or control reconfiguration) is expected to work flawlessly and the other is kept as focus of research, here I consider the interactions between these mechanisms from the design stage onwards. Reciprocal influences will be analyzed and tweaked in order to provide a holistic approach under realistic assumptions (parameter variation, noises, state and input constraints). I will analyze the stability and performance properties through the prism of set characterizations. The resulted algorithms will be analyzed for their runtime performances and complexity (execution time and resource utilization).

**Impact**

Design of fault tolerant control strategies based on realist assumptions (parameter variance, unreliability of the fault detection and isolation) will allow functioning in more challenging conditions. Providing stability and performance guarantees will allow for safer systems and more aggressive control design.

For technological applications I may mention smartgrid systems, which are naturally heterogeneous, suffer contradictory constraints and have distributed control, as a perfect benchmark for the algorithms developed here. Improvements in overall functioning, performance and economic benefits are expected. As a conclusion, the findings of the research implemented in professionally written software libraries can lead to real economic benefits concrete engineering applications.

### 13.3. Motion planning strategies

Many of the issues raised in Section 12.2 are tackled by computing offline the trajectories. To this end I consider the notion of flatness, a generalization of controllability for nonlinear dynamics [91].
Dynamics $x^+ = f(x, u)$ where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state/input vectors are called \textit{differentially flat} if there exists a \textit{flat output} $z \in \mathbb{R}^m$ with $z(t) = \gamma(x(t), u(t), \dot{u}(t), \ldots, u^{(r)}(t))$ such that the states and inputs are algebraically expressed in terms of $z$ and a finite number of its higher-order derivatives:

$$x(t) = \Theta\left(z(t), \dot{z}(t), \cdots, z^{(q)}(t)\right), \quad u(t) = \Phi\left(z(t), \dot{z}(t), \cdots, z^{(q+1)}(t)\right).$$

Typically $z(t)$ is parametrized by a set of smooth basis functions $\Lambda_i(t)$: $z(t) = \sum_{i=0}^{n} \alpha_i \Lambda_i(t)$. This formulation reduces the problem of trajectory generation to finding an adequate flat output, i.e., finding the coefficients $\alpha_i$ which tune $z(t)$ such that certain constraints and cost minimizations, expressed in terms of $x(t), u(t)$ via mappings $\Theta(\cdot), \Phi(\cdot)$ are fulfilled. I propose B-splines \cite{95, 164}, as a relatively novel choice for basis functions. Their popularity lies in their geometrical and algebraic properties (continuity, local support, convexity in the flat output space, etc.).

\section{13.3.1. Offline trajectory generation via flatness with B-spline parametrizations}

Several directions of interest have become apparent.

\subsection*{Analytic descriptions of constraints and costs in the flat output space}

Usually constraints are checked and costs are evaluated at discrete moments of time. This raises obvious problems for continuous models. Here I propose to exploit B-splines properties in order to characterize continuous constraints (input bounds, way-point passing, collision avoidance, coverage and target tracking) through a finite number of (possibly non-convex) constraints involving control parameters $\alpha_i$. Similarly, I plan to express analytically integral costs (trajectory length, energy consumption, etc.) as (possibly nonlinear) mappings of the control parameters $\alpha_i$. Formally, I will study the equivalence between trajectory space and flat output space and provide explicit descriptions in terms of the parameters $\alpha_i$ which will be interpreted as decision variables in the resulting constrained optimization problems.

\subsection*{13.3.2. Improvements in reliability and stability guarantees under model variation}

Any realistic model has to account for perturbations (either structured or non-structured) and for model variations (e.g., fault events which change significantly the dynamics). The goal is to integrate these perturbations and variations into the design procedure such that the flat-derived trajectory will: i) handle them either robustly (up to some pre-defined bounds) or adaptively (by changing on-the-fly the reference trajectory); ii) respect performance and stability criteria a priori given (e.g., passing within a prescribed distance from the given way-points, keeping the tracking error bounded under bounded measurement and process noises, etc.).

\subsection*{Robust design which handles model variation}

The first step will be to consider unstructured perturbations and provide explicit characterizations of the tracking error dynamics via set invariance notions (either positive if the feedback control law is fixed or controlled is the feedback is subject to an optimization problem). Gathering the disturbances
into $\delta \in \Delta$, the dynamics become $x^+ = f(x,u,\delta)$ and I aim to have a reference trajectory $(\bar{x}, \bar{u})$ such that $x(t) - \bar{x}(t) \in \Omega$ $\Rightarrow$ $x^+(t) - \bar{x}^+(t) \in \Omega$, $\forall \delta \in \Delta$.

Further, I will explicitly introduce the uncertainties bounds $\Delta$ into the trajectory design procedure, i.e., the trajectory will respect the operational constraints for all the values of the parameter (e.g., passing near a way-point $\omega_i$ regardless of disturbance: $\exists t \in [t_i^-, t_i^+]$ s.t. $|x(t) - \omega_i| \leq \epsilon$, $\forall \delta \in \Delta$).

### Adaptive design which handles model variation

Assuming that the model switches between pre-defined modes ($x^+ \in f_\sigma(x,u,\delta) - \sigma$ can be seen as a fault signature which switches between healthy and faulty modes of functioning) I consider an adaptive approach. Hence, instead of having a single (conservatively computed to be valid for all $\sigma$) trajectory, I provide either: i) a bundle of pre-computed trajectories, one per each mode of functioning: $(\bar{x}_\sigma(t), \bar{u}_\sigma(t))$; or ii) an explicit parametrization after the switching parameter $\sigma$: $(\bar{x}(t,\sigma), \bar{u}(t,\sigma))$. The latter construction will be expanded for LPV (linear parameter varying) models. Both approaches lead to complex closed-loop dynamics and deserve particular attention (e.g., the invariant sets which characterize them become star-shaped).

### Analysis of time variation along the flat trajectory

Since usually the tracking mechanism is the result of a discrete optimization procedure, I will consider non-uniform discretization methods in order to sample points along the flat trajectory (e.g., to provide a correct reference for a model predictive control mechanism). Further, I will consider strategies which counteract mismatch between real and reference values of the states and inputs. E.g., instead of tracking $x(t) \rightarrow \bar{x}(t)$ I will have $x(t) \rightarrow \bar{x}(\tau)$ with $\tau$, the internal time along the trajectory, itself a decision variable.

### 13.3.3. Elements of originality and innovation

The analytic descriptions of constraints and costs is to my knowledge only tangentially treated in the literature (e.g., corner cutting problem in obstacle avoidance) and the combination of flatness and B-splines parametrization has never been used at its full potential. Going beyond the relatively simple numerical and continuity issues considered so far will expand the state of the art quantitatively (new design algorithms) and qualitatively (new theoretical results which exploit the algebraic and geometrical properties of B-splines in conjunction with flat representations).

I consider explicitly stability, performance and model variation elements into the trajectory design through the prism of set characterizations (positive and controlled invariance notions). Thus, I arrive at robust and / or adaptive formulations which fully capture the structure of the model under discussion.

Flat-derived trajectories almost always assume nominal dynamics which makes them susceptible to disturbances and model variation. Here, in contrast, the proposed design procedures provide reliable control schemes (both in the robust and adaptive senses) by analyzing reciprocal influences, thus providing a holistic approach under realistic assumptions (parameter variation, noises, state and input constraints).
Impact

The expected theoretical advances in trajectory generation (continuous constraint validation, offline stability guarantees, reduced computational load, etc.) will have a large impact in the control community as a whole by making advanced motion planning algorithms numerically tractable even for complex nonlinear dynamics. The design of reliable trajectories (w.r.t. model variation and fault occurrences) will allow functioning in more challenging conditions and providing stability and performance guarantees will allow for safer systems and more aggressive control design.

I expect that the improvements in overall functioning, performance and economic benefits will make the obtained control algorithms attractive for a large number of development-oriented companies. Since unmanned vehicle control is usually hierarchical (“autopilot” and “mission management” levels) and the proposed strategies mainly deal with the upper level it will be easy to adapt over existing controllers.

13.4. Validation of the results. Experimental setups

I will consider the results from Sections 13.2 and 13.3 for simulation, laboratory and experimental validation (the results of Section 13.1 apply implicitly to the other two sections).

13.4.1. Fault tolerant control implementations in complex systems

I will consider the various control algorithms obtained and apply them to relevant fault scenarios and system dynamics. I will validate different residual formulations in an iterative procedure: start with simpler formulations (e.g., parity equations, no input/state constraints) and go to more complex cases (e.g., residual generated as a moving horizon estimation, input/state constraints).

Validation of the control algorithms over detailed simulations

I will analyze the various control strategies through the prism of computational limits (computation time, memory limitations) and will analyze also the non-convex optimization problems which appear (e.g., when generating a reference which has to be sufficiently far away from the origin) such that they are solved efficiently and with a minimum computational load. Lastly, I will consider explicitly the balance between flexibility of characterization and numerical complexity for the existing families of sets. In particular, I will consider robust approximations of predefined complexity, analytic expressions for invariant sets and generalizations for stochastic perturbations.

Stability and performance analysis

Combining together the detection and reconfiguration mechanisms, I am interested in the overall closed-loop behavior. Specifically, I want to guarantee stability and maintain (in as much as it is possible) the performance specifications of the post-fault system. These guarantees will be tested for robust/stochastic set constructions and under predefined prerequisites (bounded/stochastic noise, parameter variation within pre-specified intervals and so forth). For a coherent approach I will consider a set framework such that any operation or testing involving either the detection of reconfiguration...
 mechanism will be recast as a set operation/membership testing. I will study robust constructions (iterative estimations), positive and controlled invariance in offline and during runtime.

### 13.4.2. Motion planning strategies in a multi-agent setting

The theoretical directions shown in Section 13.3 will be tested (first in simulation and latter in experimental setups) over multi-agent formations which employ autonomous vehicles. I will proceed gradually, starting with simpler cases (no constraints, quadratic costs, no disturbances, centralized control) and relax these up to a fully realistic multi-agent scheme (constraints on state / input / target tracking and obstacle avoidance, nonlinear costs, bounded disturbances and distributed control).

**Validation through simulations without control reconfiguration**

In here I will consider the flat-derived trajectories to be computed offline in a centralized manner (the tasks and their partitioning between agents are decided a priori, e.g., for covering a certain area, with known shape and positioning for the obstacles, unobstructed communication, etc.). These reference trajectories will be provided to various trajectory tracking controllers in order to “close the loop” and have a fully functioning simulation. Throughout these simulations I will analyze the boundedness of the tracking error dynamics and the robustness of the references w.r.t. bounded noises and disturbances.

**Validation through simulations with control reconfiguration**

I will consider that the dynamics can switch between various modes of functioning (e.g., due to a fault with known model but unknown magnitude) and I will implement various fault tolerant control strategies. First, I will consider a robust scheme by providing fault magnitude bounds and pre-computing a trajectory which can tolerate these faults and/or make use of actuator/sensor redundancy. Second, I will consider active schemes where faults are accommodated through an active reconfiguration mechanism (either through selection of pre-defined trajectories from within a bundle or though variation of the coefficient which parametrizes the trajectories). Subsequently, I will test these trajectories within closed-loop control schemes and observe the behavior.

**Validation through experimental setups with autonomous vehicles**

The algorithms tested earlier will be implemented experimentally on relevant benchmarks. First, I will use a large (10-20) multi-agent formation of laboratory autonomous ground robots (relatively simple dynamics, stability issues manageable). The main points of interest will be: i) validation of the algorithms; ii) checking the tracking error values w.r.t. the bounds obtained theoretically; iii) analysis of the numerical issues generated by the computational limitations (for the distributed control approach the tracking will be done on-board by the embedded controller).

Second, I will consider a medium (3-6) multi-agent formation of nano-quadcopters where I will test reconfiguration strategies. Mainly, I will consider stuck actuator faults which gravely affect the dynamics and hence require either selecting from pre-defined trajectories or online reconfiguration. First I will consider all auxiliary information known (fault detected without delay, fault magnitude known) and then I will relax these requirements (fault may be mis-diagnosed, its magnitude will lay within an interval).
Bibliography


Sets in control. Applications to fault tolerant control and motion planning.


Sets in control. Applications to fault tolerant control and motion planning.


