

SET-BASED HIERARCHICAL CONTROL FOR MULTI-TIMESCALE  
ENERGY MANAGEMENT

by

Vignesh Raghuraman

APPROVED BY SUPERVISORY COMMITTEE:

---

Justin Koeln, Chair

---

Justin Ruths

---

Mark W. Spong

---

Tyler Summers

Copyright © 2022

Vignesh Raghuraman

All rights reserved

*To my dear wife, my parents, relatives, and friends.*

SET-BASED HIERARCHICAL CONTROL FOR MULTI-TIMESCALE  
ENERGY MANAGEMENT

by

VIGNESH RAGHURAMAN, BE, MS

DISSERTATION

Presented to the Faculty of  
The University of Texas at Dallas  
in Partial Fulfillment  
of the Requirements  
for the Degree of

DOCTOR OF PHILOSOPHY IN  
MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

December 2022

## ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my advisor, Dr. Justin Koeln, for his great advising and continued support to help me to this stage in my career. His mentorship has instilled a tremendous growth in me being a researcher. I am really honored to have his mentorship over the last couple of years. I would also like to acknowledge the members of my doctoral examination committee, Dr. Justin Ruths, Dr. Mark Spong, Dr. Mohammad Saquib, and Dr. Tyler Summers, who have really helped guide and strengthen my research with their own expertise. I also would like to thank my professors Dr. Konstantinos Tsakalis, Dr. Armando Rodriguez, and Dr. Raja Ayyanar from graduate school and Ms. Mangayarkarasi, Dr. K. Renganathan, Ms. B.Sumathy, Ms. Lavanya Srinivasan, Ms. Lovecy Thomas, Mr. Chitrakannan, MS. Gowrishwari, and Mr. Karthikeyan from undergraduate school.

I must also thank my wife, parents, my father in-law, mother in-law, and my brother-in-law. My parents have been a great pillar of support during difficult times and I must thank them. My wife has been a great motivation over the past couple of months and especially during the final stages of the PhD. At this juncture, I am also grateful to my brother Swami for the motivation to come to the USA and pursue graduate school. Also, I would like to thank my brothers Shiva and Vivek and sister-in-laws Shobha and Deepa for their support.

I would also like to thank Energy Systems Control Laboratory (ESCL) lab mates Daniel, Wendy, and Trent. Special thanks to Daniel and Wendy, who I share a great friendship and have provided tremendous support and guidance in both research and personally as well. And I would like to thank my dear friend, Venkatraman Renganathan, who has provided great technical, moral support, and guidance through my graduate school at Arizona State University and here at The University of Texas at Dallas. Really glad to have you as my friend, Venkat. To my friend Zhongyou Wu, you have been a great support and provided guidance through my PhD.

I would also like to thank Benjamin, who has shared his technical expertise and knowledge in a couple of courses and in research as well. And, I would also love to express my thanks to Chang Liu, who I share a close friendship starting from our early PhD days. I would not be here if it weren't for your help while preparing for the Qualifying Exam. Thanks to Yi Guo, Karthik Ganapathy, Sleiman Safaoui, Sahand Hadizadeh, Navid Hashemi, Jonas Wagner, Aditya Bhat, Aryan, Mirell, Devesh, Zhichen, and Bhanu, who have really helped shape my graduate school experience. Special thanks to my friends Vigneshwaran Gurusamy, Avinash Chandrasekharan, Narayanan V. Padmanabhan, Vishal Iyer, Vishwa Krishnan, Aravind Ganesan, Siddarth Debnath, Bhargav, Reshma, and Arun Ramesh.

Additionally, I would like to acknowledge the support of the Air Force Research Laboratories (AFRL) and National Science Foundation (NSF) who have generously supported our research. Their support and flexibility has provided the opportunity to carry out research which interests me and provides value to our society. Last but not the least, I am extremely grateful to the almighty for giving me the courage to handle tough situations and be here in this position.

August 2022

SET-BASED HIERARCHICAL CONTROL FOR MULTI-TIMESCALE  
ENERGY MANAGEMENT

Vignesh Raghuraman, PhD  
The University of Texas at Dallas, 2022

Supervising Professor: Justin Koeln, Chair

The high-level goal of this PhD dissertation is to develop novel controller formulations and analysis techniques to provide provable closed-loop system behaviors for complex systems. Specifically, this dissertation focuses on the use of novel set-based mechanisms in the development of hierarchical Model Predictive Control (MPC) formulations for multi-timescale energy management systems. Many complex systems such as hybrid-electric vehicle energy systems, smart power grids, water distribution networks, and Heating Ventilation and Air-Conditioned (HVAC) systems are multi-timescaled and have long operation times. A single centralized MPC controller with fast update rates and a long prediction horizon might not be able to solve the control optimization problem within the allocated time and thus, real-time control actuation is not possible.

Alternatively, a hierarchical control architecture can be used to distribute control decisions among various controllers connected in a hierarchy where the upper-level controller plans coarse state and input trajectories at slow time steps while the lower-level controllers utilizes this information to plan state and input trajectories at fast time steps. However, existing hierarchical formulations are not well suited to maximize system transient performance subject

to state, input, and terminal constraints. The open research challenge is how to provide the flexibility to the lower-level controllers through novel coordination mechanisms to maximize control performance while guaranteeing state, input, and terminal constraint satisfaction.

To achieve a computationally efficient hierarchical MPC algorithm for multi-timescale energy management, the following research problems are addressed in this dissertation.

1. Development of a *multi-level vertical hierarchical MPC* framework for systems with additive known and unknown bounded disturbances. The proposed hierarchical control algorithm is proven to be recursively feasible and is scalable with increase in prediction horizon and number of states. The sub-optimality index of the hierarchical controller is enhanced through a *wayset* and *terminal cost*-based coordination. To facilitate a computationally efficient hierarchical control algorithm, set computations have been developed for *zonotopes* and *constrained zonotopes* with a focus on application to systems and control.
2. Development of a *tube-based robust MPC* with simultaneous optimization of uncertainty sets using *zonotopes*. The proposed control formulation guarantees recursive feasibility and constraint satisfaction to bounded additive disturbances from an uncertainty set optimized in real-time. The control formulation is extended to a *full hierarchical MPC* where the uncertainty is quantified based on difference in control decisions between hierarchical levels and between controllers in the same level. The hierarchical control framework is shown to be recursively feasible and guarantees state and input constraint satisfaction.
3. Development of a wayset-based *Stochastic MPC* framework that guarantees Mission-Wide Probability of Safety (MWPS) for systems with long duration. To enable longer missions under greater uncertainty, the wayset-based stochastic MPC allows for the



prediction horizon of the MPC to be significantly shorter than the length of the mission. A scenario-based approach is used to approximate the stochastic MPC formulation and recursive feasibility is proven.

## TABLE OF CONTENTS

ACKNOWLEDGMENTS . . . . .	v
ABSTRACT . . . . .	vii
LIST OF FIGURES . . . . .	xiv
LIST OF TABLES . . . . .	xviii
CHAPTER 1 INTRODUCTION . . . . .	1
1.1 Motivation and Background . . . . .	1
1.2 Research Objectives . . . . .	3
1.2.1 Problem Statement . . . . .	3
1.3 Dissertation Scope . . . . .	4
1.4 Organization of Dissertation . . . . .	4
CHAPTER 2 ZONOTOPE-BASED SET COMPUTATION . . . . .	6
2.1 Introduction . . . . .	6
2.2 Notation and Preliminaries . . . . .	7
2.3 Zonotope-Halfspace Intersection . . . . .	8
2.4 Constrained Zonotope-Halfspace Intersection . . . . .	11
2.5 Redundancy Removal . . . . .	15
2.6 Inner Approximations . . . . .	18
2.6.1 Motivation . . . . .	18
2.6.2 Zonotopes . . . . .	19
2.6.3 Constrained Zonotopes . . . . .	21
2.7 Convex Hulls . . . . .	25
2.8 Robust Positively Invariant (RPI) sets . . . . .	31
2.8.1 Iterative Method . . . . .	32
2.8.2 One-step Optimization Method . . . . .	32
2.9 Pontryagin Difference . . . . .	36
2.9.1 Iterative Method . . . . .	37
2.9.2 One-step Optimization Inner-Approximation Method . . . . .	38
2.10 Application to Reachability Analysis . . . . .	41

2.11	Conclusions . . . . .	44
CHAPTER 3 VERTICAL HIERARCHICAL MODEL PREDICTIVE CONTROL WITH ADDITIVE DISTURBANCES . . . . .		46
3.1	Introduction . . . . .	46
3.2	Notation and Preliminaries . . . . .	49
3.3	Problem Formulation . . . . .	50
3.4	Robust Vertical Hierarchical Control . . . . .	51
3.5	Set Definitions . . . . .	55
3.5.1	Output Constraint Tightening for $\mathbf{C}_M$ . . . . .	55
3.5.2	Output Constraint Tightening for $\mathbf{C}_i, i \in [1, M - 1]$ . . . . .	56
3.5.3	Waysets . . . . .	58
3.5.4	Initial Conditions . . . . .	60
3.6	Hierarchical Control Feasibility . . . . .	61
3.7	Set Computations . . . . .	64
3.7.1	Tightened Output Constraints . . . . .	64
3.7.2	Waysets . . . . .	64
3.7.3	Zonotope-based Set Calculation (See Chapter 2 for more details) . . . . .	66
3.8	Numerical Examples . . . . .	67
3.8.1	Vehicle Example . . . . .	67
3.8.2	Thermal Example . . . . .	72
3.9	Conclusions . . . . .	76
CHAPTER 4 HIERARCHICAL MPC WITH COORDINATING TERMINAL COSTS . . . . .		78
4.1	Introduction . . . . .	78
4.2	Notation and Preliminaries . . . . .	80
4.3	Problem Formulation . . . . .	80
4.3.1	Vertical Hierarchical Control . . . . .	81
4.3.2	Constraint Satisfaction . . . . .	84
4.4	Control Performance . . . . .	87
4.5	Terminal Cost Computation . . . . .	90

4.5.1	Upper-level Controller . . . . .	90
4.5.2	Lower-level Controller . . . . .	90
4.6	Numerical Example . . . . .	92
4.7	Conclusions . . . . .	96
4.8	Appendix . . . . .	97
CHAPTER 5 TUBE-BASED ROBUST MPC WITH ADJUSTABLE UNCERTAINTY SETS USING ZONOTOPES . . . . .		100
5.1	Introduction . . . . .	100
5.2	Notation and Preliminaries . . . . .	103
5.3	Robust MPC Background . . . . .	103
5.4	Robust MPC with Integrated RPI Set Computation and Constraint Tightening	106
5.5	Zonotopic Set Operations . . . . .	108
5.6	One-step RPI Set Computation and Constraint Tightening . . . . .	111
5.6.1	Containment Constraints . . . . .	112
5.6.2	Cost Function . . . . .	113
5.7	Numerical Example . . . . .	117
5.8	Conclusions . . . . .	120
CHAPTER 6 HIERARCHICAL MPC FOR COUPLED SUBSYSTEMS USING ADJUSTABLE TUBES . . . . .		123
6.1	Introduction . . . . .	123
6.2	Notation and Preliminaries . . . . .	126
6.3	Problem Formulation . . . . .	127
6.4	Hierarchical Control . . . . .	129
6.5	Nominal Trajectories and Error Propagation . . . . .	134
6.6	Hierarchical Control Feasibility . . . . .	138
6.7	Numerical Example . . . . .	141
6.8	Conclusions . . . . .	147
6.9	Appendix . . . . .	148
6.9.1	Inter-sample tightened output constraint set computation for $\mathbf{C}_0$ . . .	148
6.9.2	RPI set computation . . . . .	149

6.9.3	Output Constraint Tightening for $\mathbf{C}_0$ . . . . .	150
6.9.4	Terminal Constraint Tightening for $\mathbf{C}_0$ . . . . .	151
6.9.5	Set containment condition ( $\Delta\mathcal{Z} \subseteq \text{Pre}(\Delta\mathcal{Z})$ ) . . . . .	152
CHAPTER 7 LONG DURATION STOCHASTIC MPC WITH MISSION-WIDE PROBABILISTIC CONSTRAINTS USING WAYSETS . . . . .		153
7.1	Introduction . . . . .	153
7.2	Notation and Preliminaries . . . . .	155
7.3	Problem Formulation . . . . .	156
7.4	SMPC with MWPS Background . . . . .	157
7.4.1	Open-loop and Closed-loop MWPS . . . . .	157
7.4.2	Scenario-based SMPC . . . . .	158
7.4.3	Drawbacks . . . . .	159
7.5	Wayset-based SMPC with MWPS . . . . .	160
7.5.1	Error Dynamics . . . . .	161
7.5.2	Wayset Computation . . . . .	162
7.5.3	Scenario-based SMPC with Wayset . . . . .	163
7.5.4	Implementation and Set Computations . . . . .	164
7.6	Recursive Feasibility . . . . .	166
7.7	Numerical Example . . . . .	168
7.8	Conclusions . . . . .	172
CHAPTER 8 CONCLUSIONS AND FUTURE DIRECTIONS . . . . .		173
8.1	Future Research Directions: Theory . . . . .	175
8.2	Future Research Directions: Application . . . . .	176
REFERENCES . . . . .		177
BIOGRAPHICAL SKETCH . . . . .		185
CURRICULUM VITAE		

## LIST OF FIGURES

1.1	Schematic showing the individual chapters and relation to each other. . . . .	5
2.1	Left: The intersection of the zonotope $Z$ and the halfspace $H_-$ corresponding to the hyperplane $H$ results in the constrained zonotope $Z_h$ . The distances $d_1$ and $d_2$ , measured orthogonally to $H$ , are shown to provide a geometric interpretation of the equality constraints in (2.5). Right: An example where the constrained zonotope $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\}$ , with corresponding unconstrained zonotope $Z = \{\mathbf{G}, \mathbf{c}\}$ , where $Z$ intersects the hyperplane $H$ but $Z_c$ does not. . . . .	12
2.2	Zonotopes $Z_1$ and $Z_2$ , where $Z_1 \cap Z_2 = Z_2$ , used to demonstrate the ability to remove redundancy from constrained zonotopes that can arise from operations like the generalized intersection. . . . .	18
2.3	The inner-approximation of $Z$ with $n_g = 5$ by the reduced-order zonotope $Z_r$ with $n_r = 3$ . . . . .	21
2.4	Left: The inner-approximation of $Z_c$ by a constrained zonotope $Z_r$ with one less generator and constraint. Right: The inner-approximation of $Z_c$ by a zonotope $Z$ and an interval set $B$ . . . . .	26
2.5	The volume ratios for the inner-approximation of 100 randomly generated constrained zonotopes by a constrained zonotope $Z_r$ with one less generator and constraint, a zonotope $Z$ , and an interval set $B$ . The red crosses denote outliers that do not fit the box plot distribution. . . . .	26
2.6	Left: The convex hull $Z_h$ of zonotopes $Z_1$ and $Z_2$ . Right: The convex hull $Z_{ch}$ of constrained zonotopes $Z_{c1}$ and $Z_{c2}$ , where each constrained zonotope is a zonotope-halfspace intersection corresponding to the shown hyperplanes. . . . .	31
2.7	Comparison of volume ratio and computation time as a function of set complexity for outer-approximations of the mRPI set using iterative and 1-step approaches based on H-Rep or G-Rep. . . . .	35
2.8	Comparison of set complexity and computation time as a function of system order for outer-approximations of the mRPI set using iterative and 1-step approaches based on H-Rep or G-Rep. . . . .	36
2.9	Left: The Pontryagin difference $Z_d = Z_1 \ominus Z_2$ where $Z_1$ and $Z_2$ are zonotopes but $Z_d$ is not [3]. Right: The inner-approximation of $Z_d$ by a zonotope $\tilde{Z}_d \subseteq Z_d$ . . . . .	39
2.10	The evolution of backward reachable wayset $Z_c(k)$ for $k = 40$ and $N = 10$ time steps starting from $\mathbf{x}^*$ projected on the position and energy states. The sets $Z_c(k + j), \forall j \in \{7, 8, 9\}$ are zonotopes (evident from symmetry) while the sets $Z_c(k + j), \forall j \in \{0, \dots, 6\}$ , are constrained zonotopes. The constrained zonotope wayset $Z_c(k)$ contains $\mathbf{x}_*$ ensuring the control feasibility from [48, 49]. . . . .	43

2.11	Top: The wayset $Z_c$ , inner-approximating interval set $B$ with $V_r = 0.35$ , and $CH(B \cup \mathbf{x}_-^*)$ with $V_r = 0.39$ are shown on the left and the projections on to the position and velocity states are shown on the right. Bottom: The wayset $Z_c$ , inner-approximating interval set $B$ containing $\mathbf{x}_-^*$ with $V_r = 0.30$ shown on the left with the projection shown on the right. . . . .	45
3.1	The relationship between prediction horizons $N_i$ and time step sizes $\Delta t_i, i \in [1, 3]$ , for a three-level, $M = 3$ , hierarchical controller operating a system for $t \in [0, 27]$ . . . . .	51
3.2	The hierarchical MPC structure with $M$ levels, where controllers $\mathbf{C}_i, i \in [1, M]$ are formulated based on (3.6), the known disturbances $\hat{d}_i(k_i)$ are computed based on (3.7), the waysets $\hat{\mathcal{S}}_i(k_i + N_i(k_i)), i \in [2, M]$ , are used to coordinate controllers $\mathbf{C}_i$ and $\mathbf{C}_{i-}$ , and the static feedback control law (3.8) bounds the effect of the unknown disturbance $\Delta d(k) \in \mathcal{D}$ . . . . .	54
3.3	(a) Controller $\mathbf{C}_i$ plans a feasible state trajectory (large blue dots) at the slow time step $k_i$ but the resulting trajectory (small black dots) violates output constraints (dashed black line) at the inter-sample system time steps. (b) Tightening the output constraint set (dashed blue line) for $\mathbf{C}_i$ ensures that any trajectory at the slow time index is also feasible at the faster time indices. . . . .	57
3.4	Schematic showing the need for the initial condition option in (3.6e) where $x(3)$ satisfies the wayset constraint $x(3) \in \hat{\mathcal{S}}_2(3)$ but $x(3) \notin \mathbb{X}_{\mathcal{T}}^{(2)}$ . . . . .	61
3.5	Schematic showing how <b>Assumption 6</b> and <b>Lemmas 7-9</b> are used to establish feasibility of each controller at every time step for a three-level hierarchical controller. . . . .	64
3.6	References for state and load power. . . . .	69
3.7	Disturbance profile consisting of a large known pulse and small unknown deviations. . . . .	69
3.8	Example of constraint tightening with projections of various output constraint sets on the position and velocity states to the left and the acceleration and deceleration inputs to the right. Black denotes the original output constraints $\mathcal{Y}$ while gray denotes the robust output constraints $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_2(k_2), \forall k_2$ . Blue denotes the time-varying tightened output constraint set $\hat{\mathcal{Y}}_1(k_1), \forall k_1 \neq 5$ , while semi-transparent red denotes $\hat{\mathcal{Y}}_1(k_1), k_1 = 5$ . . . . .	69
3.9	Simulation results comparing the shrinking horizon centralized controller, receding horizon centralized controller with a short prediction horizon, and a two-level hierarchical controller. . . . .	71
3.10	Thermal system comprised of a disturbance heat input $Q_0$ and controllable heat transfer $Q_i, i \in [1, n]$ , between $n$ thermal elements of temperature $T_i$ and ambient surroundings of temperature $T_\infty$ . . . . .	73
3.11	Known disturbance profile for the heat input $Q_0$ . . . . .	73

3.12	Simulation results comparing the shrinking horizon centralized controller (top), receding horizon centralized controller with a short prediction horizon (middle) and a three-level hierarchical controller (bottom). . . . .	74
3.13	Average computation time for the centralized and hierarchical controller as a function of operating duration. . . . .	76
3.14	Average computation time for the centralized and hierarchical controller as a function of system order. . . . .	77
4.1	Notional example of the combined use of waysets and terminal costs for coordination between controllers at different levels of the hierarchy. . . . .	85
4.2	References for position and load power. . . . .	93
4.3	Simulation results for initial condition $x(0) = [0 \ 0 \ 150]^T$ comparing the proposed hierarchical MPC with terminal costs (Hier-T) to centralized MPC (Cent), hierarchical MPC without terminal costs (Hier-NT), and hierarchical MPC with only the upper-level controller (Hier-Up). . . . .	94
4.4	Simulation results for initial condition $x(0) = [0 \ 0 \ 100]^T$ comparing the proposed hierarchical MPC with terminal costs (Hier-T) to centralized MPC (Cent), hierarchical MPC without terminal costs (Hier-NT), and hierarchical MPC with only the upper-level controller (Hier-Up). . . . .	95
4.5	Comparison of system operation cost for both initial conditions and each of the four controllers. . . . .	96
4.6	Notional state trajectories used to demonstrate the operating cost relationships among feasible trajectories. . . . .	98
5.1	Left: Given zonotopes $\mathcal{Z}_1, \mathcal{Z}_2$ . Right: The Pontryagin difference $\mathcal{Z}_d$ is shown in green and the inner-approximating Pontryagin differences computed using <b>Theorem 14</b> for $p = 1, 2$ , and $\infty$ are shown in yellow, cyan, and brown. . . . .	116
5.2	Simulation results for uncertainty weightings $\lambda \in \{10^1, 2 \times 10^5, 10^6\}$ with the dashed lines denoting the reference trajectories for the position state and the acceleration/deceleration input. . . . .	119
5.3	Relationship between the chosen value of $\lambda$ and the resulting uncertainty set size captured by $\Phi_w$ . . . . .	120
5.4	Relationship between the chosen value of $\lambda$ and the resulting operational cost computed using (5.25). . . . .	120
5.5	Tightened state and input constraint sets implicitly showing the effect of the increasing uncertainty set size for $\lambda \in \{10^1, 2 \times 10^5, 10^6\}$ . . . . .	121
5.6	Comparison of $\tilde{\mathcal{E}}(\Phi_\varepsilon)$ and mRPI set $\mathcal{E}$ for $\lambda \in \{2 \times 10^5, 10^6\}$ . . . . .	122



6.1	Two-level hierarchical MPC where $\mathbf{C}_0$ is formulated based on (6.11) and $\mathbf{C}_i$ , $i \in \mathcal{N}$ , based on (6.14). The $\mathbf{C}_0$ -optimal trajectories $\hat{y}_i^*(j)$ and $\hat{w}_i^*(j)$ are computed using (6.16) and (6.17). The optimal output deviations $\delta_i^*(k_0)$ are used to coordinate controllers $\mathbf{C}_0$ and $\mathbf{C}_i$ , $i \in \mathcal{N}$ , and the static feedback control law (6.15) computes the inputs to each subsystem $\mathbf{S}_i$ . . . . .	136
6.2	Schematic showing how <b>Assumption 15</b> and <b>Lemmas 13-15</b> are used to establish feasibility of two-level hierarchical controller with coupling between subsystems in the lower-level. . . . .	140
6.3	Thermal system with two subsystems $\mathbf{S}_1$ and $\mathbf{S}_2$ that are dynamically coupled by active power flows $Q_2$ and $Q_4$ . . . . .	141
6.4	The desired reference trajectories for $Q_1$ and $Q_3$ . . . . .	144
6.5	Simulation results comparing the shrinking horizon centralized controller, receding horizon centralized controller with a short prediction horizon, receding horizon centralized controller with a terminal constraint, two-level hierarchical controller with no subsystem deviations, and a two-level hierarchical controller with subsystem deviations. . . . .	145
6.6	Relation between the chosen value of $\Lambda$ and normalized cost relative to Hier ( $\Lambda = 0$ ) computed using (6.36). . . . .	147
7.1	Top subplot shows the error set $\mathcal{E}(k_F k)$ computed using scenario optimization from [89] and terminal set $\mathcal{T}$ . Bottom subplot shows the stochastic wayset $\mathcal{S}(N(0))$ computed per <b>Algorithm 8</b> , error set $\mathcal{E}(N(0) 0)$ determined per (7.17) and tightened stochastic wayset $\mathcal{S}(N(0)) \ominus \mathcal{E}(N(0) 0)$ . . . . .	170
7.2	Figure showing the state trajectories for 900 missions starting from $x(0) = [-10 \ 0]^T$ for $p = 0.95$ . Note that crosses denote mission failures. . . . .	171
7.3	Figure showing the trend between MWPS and mission index for $p = 0.85$ (solid blue) and $p = 0.95$ (solid green). Note that MWPS thresholds are shown in dashed green and dashed blue, respectively. . . . .	171

## LIST OF TABLES

2.1	Pontryagin difference set complexity and computation time (seconds) . . . . .	40
2.2	Complexity and Computation Time of Waysets . . . . .	44
3.1	Complexity and Computation Time of Waysets . . . . .	72
6.1	Controller computation times . . . . .	146

# CHAPTER 1

## INTRODUCTION

### 1.1 Motivation and Background

For the control of many complex systems, the ability to satisfy both input and state constraints is critical to maintaining safe and reliable system operation. Model Predictive Control(MPC) is very well suited to achieve desired performance and control of such systems where constraints can be explicitly enforced in the control optimization problem. Examples include the control of aircraft power systems [80, 20], on- and off-road hybrid vehicles [72, 41, 88], smart grids [42, 33, 93], and water distribution networks [63, 64].

However, for these systems, operation time is normally long and the time and computational resources needed to solve the MPC control optimization problem with such a large prediction horizon can make it difficult to achieve real-time control actuation. Moreover, these systems are multi-timescaled with each subsystem operating at different timescales. For example in an aircraft power system, the dynamics of the electrical system can evolve over milliseconds while the dynamics of a thermal system can evolve over hundreds of seconds. To overcome these challenges, a hierarchical controller can be used to decompose control decisions across multiple levels of controllers [74]. Upper-level controllers update at large time steps to achieve long prediction horizons with fewer discrete time steps. Lower-level controllers with small time steps use short prediction horizons to minimize computational cost and thus, enabling real-time implementation. Moreover, each controller level can be tailored for each subsystem and can be operated at the required timescale. To handle the timescale separation, several two-level hierarchical MPC formulations have been developed [25, 26, 75, 74, 8, 9, 86].

However, existing hierarchical formulations are not well suited to maximize system performance subject to state, input, and terminal constraints for a finite system operation. Most

formulations are formulated with the goal of driving the system to an equilibrium within the interior of state and input constraints. However, for systems operating under a finite duration, such equilibrium might not exist as in the case of systems whose operation is based on the consumption of a finite energy resource (e.g. fuel in an aircraft [80, 20] or battery state of charge in an electric vehicle [72, 41, 88]). Additionally, existing hierarchical approaches are formulated with the upper-level controllers robust to the decisions of lower-level controllers and overall control authority is divided among each hierarchical level. While existing coordination mechanisms based on reference tracking achieve desired control performance, it does not incentivize the lower-level controllers to deviate from the upper-level control plan and utilizing the fast system dynamics to further maximize system performance. Finally, while most hierarchical MPC formulations are tailored for two levels of controllers, many complex systems are multi-timescaled with more than two timescales and a M-level hierarchical MPC would be more effective for each timescale.

While vertical hierarchical controller with one controller in each level significantly reduces computational cost, a *full* hierarchical architecture can be employed to further reduce computational cost with one controller at lower-levels tailored for each subsystem. The works in [24, 26] developed a full two-level hierarchical controller with one upper-level controller and multiple controllers at the lower-level, one for each subsystem operating at the same timescale in [24] and different timescales in [26], and guarantees closed-loop stability and input constraint satisfaction while driving the system to a desired steady state. While [24] drives the system to a desired set around a steady-state equilibrium, the works in [26, 9, 86] guarantee convergence to the exact steady-state equilibrium. As previously mentioned, a steady-state equilibrium might not exist for systems such as aircrafts and electric vehicles and whose operation is based on the utilization of a finite energy resource.

Future hierarchical control design must adopt a generic hierarchical architecture with more than two levels that can handle multi-timescale nature of complex systems. Additionally, novel coordination mechanisms that overcomes the limitations of existing reference

tracking coordination by providing additional flexibility to the lower-level controllers is also needed. While these developments will help reduce computational cost and achieve real-time control actuation for a wide range of systems, further research is needed to develop a full hierarchical architecture for systems with complex interconnection between subsystems that provides similar control flexibility to the lower-level controllers. While this is more challenging, this development could tremendously help achieve improved control performance for complex systems with dynamically coupled subsystems.

## 1.2 Research Objectives

### 1.2.1 Problem Statement

Due to a large number of systems having long prediction horizons and multiple timescales, hierarchical control architectures have wide applicability. However, achieving closed-loop constraint satisfaction and control performance close to a centralized controller requires novel coordination mechanisms.

The **primary objective** of this dissertation is the development and evaluation of a MPC based hierarchical control algorithm specifically designed to optimize control performance throughout the systems and subsystems over multiple timescales and operating for long duration. This work focuses on the theoretical development and analysis of novel set-based hierarchical MPC formulations and simplified numerical examples are used to highlight the key features and capabilities of these controllers. The proposed hierarchical control framework is developed to be

- *widely applicable* to complex multi-timescale systems such as aircraft energy systems, HVAC systems, water distribution networks, and smart power grids,
- *scalable* to large systems with many actuators, states, measurements, control objectives, and long prediction horizons,

- *robust* to model and signal uncertainty,
- *computationally efficient* to achieve real-time implementation.

### 1.3 Dissertation Scope

In order to meet the primary objective, multiple secondary objectives define the scope of the dissertation and, when achieved, provide a generic hierarchical control framework that can be adopted to improve the control performance of a wide range of complex systems. These secondary objectives are:

1. the development of set-based methods using *zonotopes* and *constrained zonotopes* to achieve computational performance applicable to a hierarchical controller,
2. the formulation of a generic *wayset*-based vertical hierarchical MPC framework for computationally efficient control of complex systems across different timescales,
3. tube-based *full* hierarchical MPC framework for systems with dynamically coupled subsystems,
4. closed-loop analysis of the vertical and full hierarchical controller with respect to recursive feasibility and closed-loop constraint satisfaction, and
5. simulation-based evaluation of hierarchical control performance on simple examples such as vehicle and thermal systems.

### 1.4 Organization of Dissertation

Fig. 1.1 shows an outline of the chapters including the methods developed in each and the relation between them used to achieve the goal of this dissertation. Chapter 2 introduces the zonotope-based set computation including methods such as halfspace intersection, Robust

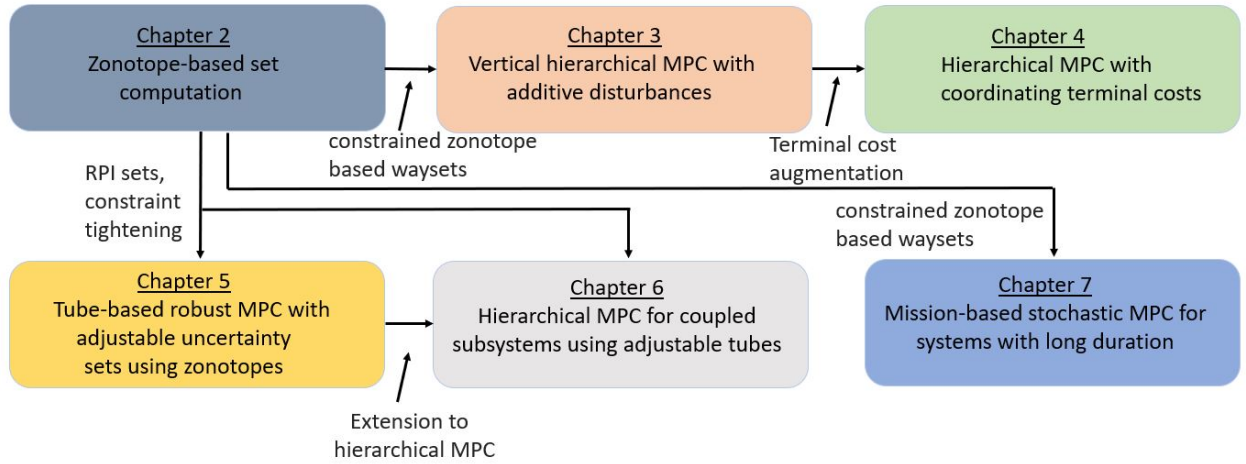


Figure 1.1. Schematic showing the individual chapters and relation to each other.

Positively Invariant (RPI) sets, Pontryagin difference, and reachable sets, which will be used throughout the following chapters.

Chapter 3 presents the development of a robust multi-level *vertical* hierarchical model predictive controller for linear systems and the associated control feasibility, while Chapter 4 modifies this hierarchical MPC formulation with coordinating terminal costs to achieve guaranteed control performance bounds. Chapter 5 proposes a tube-based robust MPC with simultaneous uncertainty set optimization and online RPI set computation and constraint tightening. Chapter 6 discusses the *full* hierarchical MPC formulation for systems with coupled subsystems, where control coordination is achieved using adjustable tubes. Additionally, recursive control feasibility of the *full* hierarchical controller is shown. Finally, Chapter 7 discusses long duration stochastic MPC with mission-wide probability of safety using waysets, where scenario optimization is used to approximate the underlying stochastic MPC optimization problem.

## CHAPTER 2

### ZONOTOPE-BASED SET COMPUTATION <sup>1</sup>

#### 2.1 Introduction

Sets are widely used in a variety of control theory and applications including reachability analysis for system verification [7, 28, 29, 53], robust Model Predictive Control (MPC) [60, 54, 15], and state estimation [19, 2, 55]. However, the sets used in control theory are not always practical to compute in application. For example, the minimal Robust Positively Invariant (mRPI) set [68] is widely used in robust MPC [61, 69, 56]. However, in general, mRPI sets are not finitely represented and must be approximated. Furthermore, existing techniques for determining finite approximations of the mRPI set do not scale well with the dimension of the state space. Such scalability issues are found in many set computations [83], motivating the need for alternative set representations and efficient approximation algorithms.

A *zonotope* is the Minkowski sum of a finite set of line segments or, equivalently, the image of a hypercube under an affine transformation [27, 59]. Due to their computational efficiency, zonotopes have been widely used in reach set calculations for hybrid system verification, estimation, and MPC [59, 6, 79, 15]. As with the iterative algorithm in [78], computing these reach sets utilizes linear transformation and Minkowski sum operations. Zonotopes are closed under these operations (i.e. the Minkowski sum of two zonotopes is a zonotope) and the number of generators grows linearly with the number of Minkowski sum operations, compared to the potential exponential growth of the number of halfspaces in H-Rep. Unfortunately, zonotopes in general are not closed under intersection and the conversion from G-Rep to H-Rep for intersection operations is inefficient.

---

<sup>1</sup>This chapter is based on work supported by the National Science Foundation under grant 1849500. Chapter 2 in part is a reprint of material published in: © Automatica. Reprinted, with permission, from Raghuraman, V. and Koeln, J.P., 2022. Set operations and order reductions for constrained zonotopes. Automatica, 139, p.110204.



*Constrained zonotopes* were developed in [79] to overcome the limitations caused by the inherent symmetry of zonotopes. Constrained zonotopes are closed under linear transformation, Minkowski sum, and generalized intersection and can be used to represent any convex polytope. Constrained zonotopes provide the computational advantages of zonotopes while enabling exact computations of a much wider class of sets. In [48], reach set computations using constrained zonotopes were shown to be several orders-of-magnitude faster than the same set computations using H-Rep, enabling the on-line computation of these reach sets for use in a hierarchical MPC formulation.

## 2.2 Notation and Preliminaries

For sets  $Z, W \subset \mathbb{R}^n$ ,  $Y \subset \mathbb{R}^m$ , and matrix  $\mathbf{R} \in \mathbb{R}^{m \times n}$ , the linear transformation of  $Z$  under  $\mathbf{R}$  is  $\mathbf{R}Z = \{\mathbf{R}\mathbf{z} \mid \mathbf{z} \in Z\}$ , the Minkowski sum of  $Z$  and  $W$  is  $Z \oplus W = \{\mathbf{z} + \mathbf{w} \mid \mathbf{z} \in Z, \mathbf{w} \in W\}$ , and the generalized intersection of  $Z$  and  $Y$  under  $\mathbf{R}$  is  $Z \cap_{\mathbf{R}} Y = \{\mathbf{z} \in Z \mid \mathbf{R}\mathbf{z} \in Y\}$ . The standard intersection, corresponding to the identity matrix  $\mathbf{R} = \mathbf{I}_n$ , is simply denoted as  $Z \cap Y$ .

The convex polytope  $H \subset \mathbb{R}^n$  in H-Rep is defined as  $H = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{H}\mathbf{x} \leq \mathbf{f}\}$  where  $\mathbf{H} \in \mathbb{R}^{n_h \times n}$ ,  $\mathbf{f} \in \mathbb{R}^{n_h}$ , and  $n_h$  is the number of halfspaces. A centrally symmetric set  $Z \subset \mathbb{R}^n$  can be represented as a zonotope in G-Rep where  $Z = \{\mathbf{G}\boldsymbol{\xi} + \mathbf{c} \mid \|\boldsymbol{\xi}\|_{\infty} \leq 1\}$ . The vector  $\mathbf{c} \in \mathbb{R}^n$  is the center and the  $n_g$  generators, denoted  $\mathbf{g}_i$ , form the columns of the generator matrix  $\mathbf{G} \in \mathbb{R}^{n \times n_g}$ . A constrained zonotope  $Z_c \subset \mathbb{R}^n$  is defined in CG-Rep as  $Z = \{\mathbf{G}\boldsymbol{\xi} + \mathbf{c} \mid \|\boldsymbol{\xi}\|_{\infty} \leq 1, \mathbf{A}\boldsymbol{\xi} = \mathbf{b}\}$ . With  $\mathbf{A} \in \mathbb{R}^{n_c \times n_g}$  and  $\mathbf{b} \in \mathbb{R}^{n_c}$ , constrained zonotopes include  $n_c$  equality constraints that break the symmetry of zonotopes and allow any convex polytope to be written in CG-Rep. The complexity of a zonotope is captured by its order,  $o = \frac{n_g}{n}$  while the complexity of a constrained zonotope is captured by the degrees-of-freedom order,  $o_d = \frac{n_g - n_c}{n}$ . Zonotopes and constrained zonotopes are denoted as  $Z = \{\mathbf{G}, \mathbf{c}\}$  and  $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\}$ , respectively.

As shown in [79], constrained zonotopes are closed under linear transformation, Minkowski sum, and generalized intersection where

$$\mathbf{R}Z = \{\mathbf{R}\mathbf{G}_z, \mathbf{R}\mathbf{c}_z, \mathbf{A}_z, \mathbf{b}_z\}, \quad (2.1)$$

$$Z \oplus W = \left\{ [\mathbf{G}_z \ \mathbf{G}_w], \mathbf{c}_z + \mathbf{c}_w, \begin{bmatrix} \mathbf{A}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_w \end{bmatrix}, \begin{bmatrix} \mathbf{b}_z \\ \mathbf{b}_w \end{bmatrix} \right\}, \quad (2.2)$$

$$Z \cap_{\mathbf{R}} Y = \left\{ [\mathbf{G}_z \ \mathbf{0}], \mathbf{c}_z, \begin{bmatrix} \mathbf{A}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_y \\ \mathbf{R}\mathbf{G}_z & -\mathbf{G}_y \end{bmatrix}, \begin{bmatrix} \mathbf{b}_z \\ \mathbf{b}_y \\ \mathbf{c}_y - \mathbf{R}\mathbf{c}_z \end{bmatrix} \right\}. \quad (2.3)$$

Additional notation is defined as follows. The set of non-negative real numbers is denoted as  $\mathbb{R}_+$ . The matrix  $\mathbf{T} \in \mathbb{R}^{n \times m}$  with values  $t_{i,j}$  in the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column is denoted as  $\mathbf{T} = [t_{i,j}]$ . A  $n \times m$  matrix of zeros is denoted as  $\mathbf{0}_{n \times m}$  or simply  $\mathbf{0}$  if the dimension can be readily determined from context. Similarly, a vector of ones is denoted as  $\mathbf{1}$ . For a matrix  $\mathbf{A}$ , the null space is denoted  $\mathcal{N}(\mathbf{A})$  and the pseudoinverse is denoted  $\mathbf{A}^\dagger$ . Parallel vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are denoted as  $\mathbf{v}_1 \parallel \mathbf{v}_2$ . The unit hypercube in  $\mathbb{R}^n$  is defined as  $B_\infty = \{\boldsymbol{\xi} \mid \|\boldsymbol{\xi}\|_\infty \leq 1\}$  while  $B_\infty(\mathbf{A}, \mathbf{b}) = \{\boldsymbol{\xi} \in B_\infty \mid \mathbf{A}\boldsymbol{\xi} = \mathbf{b}\}$ . With the volume of a set  $X$  denoted as  $V(X)$ , the volume ratio for sets  $X, Y \in \mathbb{R}^n$  is defined as  $V_r = \left(\frac{V(X)}{V(Y)}\right)^{1/n}$ . All numerical examples were generated using MATLAB on a desktop computer with a 3.6 GHz i7 processor and 16 GB of RAM. All optimization problems were formulated and solved with YALMIP [57] and Gurobi [34].

### 2.3 Zonotope-Halfspace Intersection

This section presents methods for determining if a zonotope intersects a given halfspace along with the exact representation of this intersection in CG-Rep. The need for computing this intersection arises in reachability analysis [5] and in MPC when determining the set

of feasible initial conditions [78]. The use of CG-Rep enables exact representations unlike existing techniques that rely on zonotopic approximations of the intersection [30].

For a zonotope in  $\mathbb{R}^n$  with  $n_g$  generators, the intersection between a zonotope and a hyperplane can be tested algebraically with complexity  $O(nn_g)$ .

**Lemma 1.** (Section 5.1 of [28]) *The zonotope  $Z = \{\mathbf{G}, \mathbf{c}\} \subset \mathbb{R}^n$  intersects the hyperplane  $H = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{h}^T \mathbf{x} = f\}$  if and only if*

$$|f - \mathbf{h}^T \mathbf{c}| \leq \sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i|. \quad (2.4)$$

If a zonotope intersects a hyperplane, the intersection between the zonotope and the corresponding halfspace can be represented in CG-Rep by the addition of exactly one generator and one equality constraint.

**Theorem 1.** *If the zonotope  $Z = \{\mathbf{G}, \mathbf{c}\} \subset \mathbb{R}^n$  intersects the hyperplane  $H = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{h}^T \mathbf{x} = f\}$  corresponding to the halfspace  $H_- = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{h}^T \mathbf{x} \leq f\}$ , then the intersection  $Z_h = Z \cap H_-$  is a constrained zonotope where*

$$Z_h = \{[\mathbf{G} \ \mathbf{0}], \mathbf{c}, [\mathbf{h}^T \mathbf{G} \ \frac{d_m}{2}], f - \mathbf{h}^T \mathbf{c} - \frac{d_m}{2}\}, \quad (2.5)$$

and  $d_m = f - \mathbf{h}^T \mathbf{c} + \sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i|$ .

*Proof.* Considering any element  $\mathbf{x} \in Z_h$ , it is to be proven that  $\mathbf{x} \in Z \cap H_-$ . From the definition of  $Z_h$  in (2.5),  $\exists \boldsymbol{\xi} \in \mathbb{R}^{n_g}$  and  $\xi_{n_g+1} \in \mathbb{R}$  such that

$$\begin{aligned} \mathbf{x} &= \mathbf{G}\boldsymbol{\xi} + \mathbf{0}\xi_{n_g+1} + \mathbf{c}, \quad \|\boldsymbol{\xi}\|_\infty \leq 1, \quad |\xi_{n_g+1}| \leq 1, \\ \mathbf{h}^T \mathbf{G}\boldsymbol{\xi} + \frac{d_m}{2}\xi_{n_g+1} &= f - \mathbf{h}^T \mathbf{c} - \frac{d_m}{2}. \end{aligned} \quad (2.6)$$

By the assumption that  $Z \cap H \neq \emptyset$ , the definition of  $d_m$  and (2.4) ensure  $d_m \geq 0$ . If  $d_m = 0$ , then (2.6) results in  $\mathbf{h}^T \mathbf{G}\boldsymbol{\xi} = f - \mathbf{h}^T \mathbf{c}$ , which can be rewritten as  $\mathbf{h}^T (\mathbf{G}\boldsymbol{\xi} + \mathbf{c}) = f$ . Therefore,  $\mathbf{x} \in Z_h \subset Z$  and  $\mathbf{x} \in H \subset H_-$ . If  $d_m > 0$ , (2.6) can be solved for  $\xi_{n_g+1}$  as

$$\xi_{n_g+1} = \frac{2}{d_m} (f - \mathbf{h}^T \mathbf{c} - \frac{d_m}{2} - \mathbf{h}^T \mathbf{G}\boldsymbol{\xi}). \quad (2.7)$$

Combining (2.7) and the inequality constraint  $-1 \leq \xi_{n_g+1}$  results in

$$\begin{aligned} -1 &\leq \xi_{n_g+1} = \frac{2}{d_m} \left( f - \mathbf{h}^T \mathbf{c} - \frac{d_m}{2} - \mathbf{h}^T \mathbf{G} \boldsymbol{\xi} \right), \\ -\frac{d_m}{2} &\leq f - \mathbf{h}^T \mathbf{c} - \frac{d_m}{2} - \mathbf{h}^T \mathbf{G} \boldsymbol{\xi}, \\ \mathbf{h}^T (\mathbf{c} + \mathbf{G} \boldsymbol{\xi}) &\leq f. \end{aligned}$$

Therefore,  $\mathbf{x} \in Z$  and  $\mathbf{x} \in H_-$ . Next, considering any  $\mathbf{x} \in Z \cap H_-$ , it is to be proven that  $\mathbf{x} \in Z_h$ . For  $\mathbf{x} \in Z \cap H_-$ ,  $\exists \boldsymbol{\xi} \in \mathbb{R}^{n_g}$  such that

$$\mathbf{x} = \mathbf{G} \boldsymbol{\xi} + \mathbf{c}, \quad \|\boldsymbol{\xi}\|_\infty \leq 1, \quad \mathbf{h}^T \mathbf{x} \leq f. \quad (2.8)$$

To show that  $\mathbf{x} \in Z_h$  requires proving the existence of  $\xi_{n_g+1} \in \mathbb{R}$  such that

$$\mathbf{x} = \mathbf{G} \boldsymbol{\xi} + \mathbf{0} \xi_{n_g+1} + \mathbf{c}, \quad |\xi_{n_g+1}| \leq 1,$$

and (2.6) holds. If  $d_m = 0$ , then by definition of  $d_m$ ,  $f - \mathbf{h}^T \mathbf{c} = -\sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i|$  is satisfied.

Rearranging (2.8) results in  $\mathbf{h}^T \mathbf{G} \boldsymbol{\xi} \leq f - \mathbf{h}^T \mathbf{c}$  and thus,

$$\mathbf{h}^T \mathbf{G} \boldsymbol{\xi} \leq -\sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i|. \quad (2.9)$$

Combining (2.9) with the fact that

$$-\sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i| \leq \mathbf{h}^T \mathbf{G} \boldsymbol{\xi} \leq \sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i|, \quad \forall \boldsymbol{\xi} \text{ s.t. } \|\boldsymbol{\xi}\|_\infty \leq 1,$$

results in  $\mathbf{h}^T \mathbf{G} \boldsymbol{\xi} = -\sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i| = f - \mathbf{h}^T \mathbf{c}$ , which satisfies (2.6) independently of  $\xi_{n_g+1}$  for  $d_m = 0$ , and thus  $\xi_{n_g+1}$  can be chosen arbitrarily such that  $|\xi_{n_g+1}| \leq 1$ . If  $d_m > 0$ , let  $\xi_{n_g+1}$  be chosen as in (2.7), which satisfies (2.6). To prove  $|\xi_{n_g+1}| \leq 1$ , consider  $\mathbf{x}$  as in (2.8). Since,  $f - \mathbf{h}^T \mathbf{x} \geq 0$ ,  $\xi_{n_g+1}$  satisfies

$$\xi_{n_g+1} = \frac{2}{d_m} \left( f - \mathbf{h}^T \mathbf{x} - \frac{d_m}{2} \right) \geq -1. \quad (2.10)$$

Finally, using (2.7), the fact that  $-\mathbf{h}^T \mathbf{G} \boldsymbol{\xi} \leq \sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i|$ , and the definition of  $d_m$  results in

$$\begin{aligned}\xi_{n_g+1} &\leq \frac{2}{d_m} \left( f - \mathbf{h}^T \mathbf{c} + \sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i| - \frac{d_m}{2} \right), \\ \xi_{n_g+1} &\leq \frac{2}{d_m} \left( d_m - \frac{d_m}{2} \right) = 1.\end{aligned}$$

Thus,  $\forall \mathbf{x} \in Z \cap H_-, \mathbf{x} \in Z_h$ . □

**Example 1.** The left subplot in Fig. 2.1 shows the zonotope  $Z$  and halfspace  $H_-$  where

$$Z = \left\{ \left[ \begin{array}{cc} 1 & 1 \\ 0 & 2 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \end{array} \right] \right\}, \quad H_- = \{ \mathbf{x} \in \mathbb{R}^2 \mid [3 \ 1] \mathbf{x} \leq 3 \}.$$

From **Lemma 1**,  $Z$  intersects the associated hyperplane  $H$  since (2.4) evaluates to  $3 \leq 8$ .

From **Theorem 1**, the intersection  $Z \cap H_-$  is a constrained zonotope and (2.5) evaluates to

$$Z_h = \left\{ \left[ \begin{array}{ccc} 1 & 1 & 0 \\ 0 & 2 & 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \end{array} \right], \left[ 3 \ 5 \ 5.5 \right], -2.5 \right\}.$$

The left subplot in Fig. 2.1 also shows the physical interpretation of  $d_m$  where  $d_m = d_1 + d_2$ .

With  $d_1 = f - \mathbf{h}^T \mathbf{c}$ ,  $d_1$  captures the orthogonal distance from the hyperplane  $H$  to the center,  $\mathbf{c}$ , of the zonotope. With  $d_2 = \sum_{i=1}^{n_g} |\mathbf{h}^T \mathbf{g}_i|$ ,  $d_2$  captures the orthogonal distance from center of the zonotope to the point in  $Z$  farthest from  $H$ .

## 2.4 Constrained Zonotope-Halfspace Intersection

For the intersection  $Z_h = Z_c \cap H_-$  of a constrained zonotope  $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\}$  and a halfspace  $H_-$ , **Theorem 1** is readily modified where

$$Z_h = \left\{ [\mathbf{G} \ \mathbf{0}], \mathbf{c}, \left[ \begin{array}{cc} \mathbf{A} & \mathbf{0} \\ \mathbf{h}^T \mathbf{G} & \frac{d_m}{2} \end{array} \right], \left[ \begin{array}{c} \mathbf{b} \\ f - \mathbf{h}^T \mathbf{c} - \frac{d_m}{2} \end{array} \right] \right\}. \quad (2.11)$$

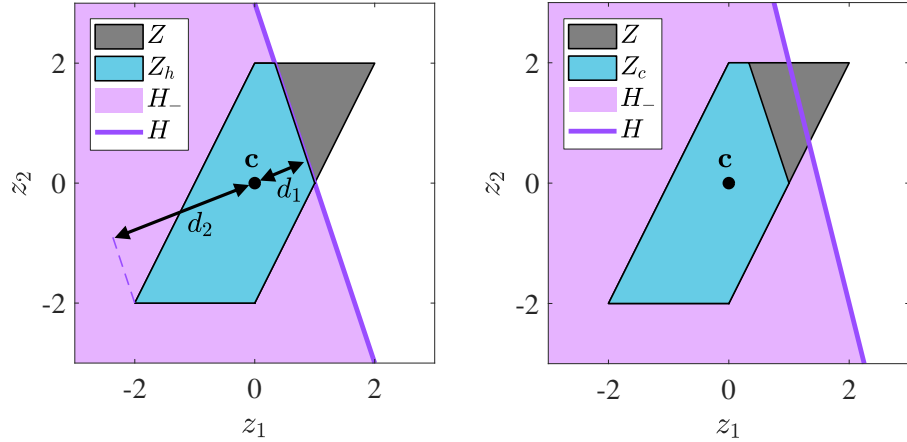


Figure 2.1. Left: The intersection of the zonotope  $Z$  and the halfspace  $H_-$  corresponding to the hyperplane  $H$  results in the constrained zonotope  $Z_h$ . The distances  $d_1$  and  $d_2$ , measured orthogonally to  $H$ , are shown to provide a geometric interpretation of the equality constraints in (2.5). Right: An example where the constrained zonotope  $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\}$ , with corresponding unconstrained zonotope  $Z = \{\mathbf{G}, \mathbf{c}\}$ , where  $Z$  intersects the hyperplane  $H$  but  $Z_c$  does not.

However, if the constrained zonotope is completely contained in the halfspace,  $Z_c \subset H_-$ , and does not intersect the corresponding hyperplane  $H$ , then  $Z_h = Z_c$  and the addition of the  $n_g + 1$  generator and  $n_c + 1$  constraint is redundant and increases the order of  $Z_h$  unnecessarily.

However, when determining if a constrained zonotope  $Z_c$  intersects a hyperplane  $H$ , the inequality (2.4) is necessary but not sufficient. The equality constraints  $\mathbf{A}\boldsymbol{\xi} = \mathbf{b}$  impose restrictions such that  $Z_c \subset Z = \{\mathbf{G}, \mathbf{c}\}$ . Thus, the parent zonotope  $Z$  may intersect  $H$  while  $Z_c$  does not (as shown in right subplot of Fig. 2.1). The intersection of a constrained zonotope with a hyperplane can be checked by solving two Linear Programs (LPs), each with  $n_g$  decision variables.

**Lemma 2.** *The constrained zonotope  $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\} \subset \mathbb{R}^n$  intersects the hyperplane  $H = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{h}^T \mathbf{x} = f\}$  if  $f_{min} \leq f \leq f_{max}$ , where*

$$f_{min} \triangleq \min\{\mathbf{h}^T(\mathbf{c} + \mathbf{G}\boldsymbol{\xi}) \mid \|\boldsymbol{\xi}\|_\infty \leq 1, \mathbf{A}\boldsymbol{\xi} = \mathbf{b}\},$$

$$f_{max} \triangleq \max\{\mathbf{h}^T(\mathbf{c} + \mathbf{G}\boldsymbol{\xi}) \mid \|\boldsymbol{\xi}\|_\infty \leq 1, \mathbf{A}\boldsymbol{\xi} = \mathbf{b}\}.$$

*Proof.* From the definition of  $f_{min}$  and  $f_{max}$ , if  $f_{min} \leq f \leq f_{max}$ , then there exists  $\mathbf{x}_{min}$ ,  $\mathbf{x}_{max} \in Z_c$  satisfying  $\mathbf{h}^T \mathbf{x}_{min} = f_{min}$  and  $\mathbf{h}^T \mathbf{x}_{max} = f_{max}$ , respectively. By the convexity of constrained zonotopes [79],  $\forall \lambda \in [0, 1]$ ,  $\exists \mathbf{x}_\lambda = \lambda \mathbf{x}_{min} + (1 - \lambda) \mathbf{x}_{max}$  such that  $\mathbf{x}_\lambda \in Z_c$ . For the case where  $f_{min} = f_{max} = f$ , any choice of  $\lambda \in [0, 1]$  results in  $\mathbf{h}^T \mathbf{x}_\lambda = f$ . Otherwise, if  $f_{min} \neq f_{max}$ , choosing  $\lambda = \frac{f - f_{max}}{f_{min} - f_{max}} \in [0, 1]$  results in  $\mathbf{h}^T \mathbf{x}_\lambda = f$ . Thus  $\mathbf{x}_\lambda \in H$  and  $\mathbf{x}_\lambda \in Z_c$ , proving  $Z_c \cap H \neq \emptyset$ .  $\square$

Note that  $f_{min}$  and  $f_{max}$  obtained using **Lemma 2** represent the largest orthogonal distance between a point in  $Z_c$  and either side of the hyperplane providing additional insight to the location of constrained zonotope with respect to the hyperplane.

**Remark 1.** *While the knowledge of  $f_{min}$  and  $f_{max}$  can be useful, checking for the non-empty intersection of a constrained zonotope and a hyperplane can be achieved by assessing the feasibility of a single LP with constraints*

$$\mathbf{h}^T(\mathbf{c} + \mathbf{G}\boldsymbol{\xi}) \leq f, \quad \mathbf{A}\boldsymbol{\xi} = \mathbf{b}, \quad \|\boldsymbol{\xi}\|_\infty \leq 1.$$

When solving these LPs is undesirable, an iterative method based on interval arithmetic from [79] provides an approach for checking constrained zonotope-halfspace intersection with complexity  $O(n_c n_g^2)$ . Reproduced from [79], **Algorithm 1** computes the interval set  $E = [\boldsymbol{\xi}^L, \boldsymbol{\xi}^U]$  such that  $B_\infty(\mathbf{A}, \mathbf{b}) \subset E \subset [-\mathbf{1}, \mathbf{1}]$  and  $R = [\boldsymbol{\rho}^L, \boldsymbol{\rho}^U] \subset \mathbb{R}^{n_g}$  where

$$R_j \supset \{\xi_j \mid \mathbf{A}\boldsymbol{\xi} = \mathbf{b}, |\xi_i| \leq 1, \forall i \neq j\}, \quad \forall j \in [1, n_g].$$

As discussed in [79], this iterative method has the potential to detect empty constrained zonotopes without solving a LP. Specifically, if  $E \cap R = \emptyset$ , then  $Z_c = \emptyset$ . Since  $E, R$  are intervals,  $E \cap R = \emptyset$  if  $\xi_j^U < \rho_j^L$  or  $\xi_j^L > \rho_j^U$  for any  $j \in [0, n_g]$ .

---

**Algorithm 1:** [79] Constrained zonotope intervals.

---

**Input** :  $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\}$   
**Output:**  $E_j, R_j, \forall j \in [1, n_g]$

- 1 Initialize  $E_j \leftarrow [-1, 1], R_j \leftarrow [-\infty, \infty], i, j \leftarrow 1$
- 2 **while**  $i \leq n_c$  **do**
- 3     **while**  $j \leq n_g$  **do**
- 4         **if**  $a_{ij} \neq 0$  **then**
- 5              $R_j \leftarrow R_j \cap (a_{ij}^{-1}b_i - \sum_{k \neq j} a_{ij}^{-1}a_{ik}E_k);$
- 6              $E_j \leftarrow E_j \cap R_j;$
- 7         **end**
- 8          $j \leftarrow j + 1;$
- 9     **end**
- 10     $i \leftarrow i + 1, j \leftarrow 1;$
- 11 **end**

---

The goal is to detect if  $Z_c \subset H_-$ , resulting in  $Z_h = Z_c$  and thus avoiding the unnecessary addition of generators and constraints from the application of (2.11). The proposed approach uses the fact that  $Z_c \subset H_-$  if and only if  $Z_c \cap H_+ = \emptyset$ , where  $H_+ = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{h}^T \mathbf{x} \geq f\}$  is the complement of  $H_-$ . By modifying (2.11) such that  $Z_{h^+} = Z_c \cap H_+$ , **Algorithm 1** can then be applied to  $Z_{h^+}$  to check if  $Z_{h^+} = \emptyset$ . Specifically, if  $E \cap R = \emptyset$ , then  $Z_{h^+} = \emptyset$  and  $Z_c \subset H_-$ . Note that applying **Algorithm 1** does not guarantee the detection of  $Z_{h^+} = \emptyset$ . As discussed in [79], **Algorithm 1** can be applied iteratively to refine the interval set  $E$ . In fact, two iterations of **Algorithm 1** were required to detect that  $Z_c \subset H_-$  for the example shown on the right subplot of Fig. 2.1.

**Remark 2.** *To provide an unbiased evaluation of constrained-zonotope hyperplane intersection using **Algorithm 1**, the intersection of  $Z_h$  (from **Example 1**) with 100 randomly chosen hyperplanes is checked. Note that for all instances, the parent zonotope  $Z$  satisfying  $Z \supset Z_h$  intersected the random hyperplanes. The constrained zonotope  $Z_h$  intersected these random hyperplanes 61 times and did not intersect for the remaining 39 times. In all cases, **Algorithm 1** accurately detected the intersection/non-intersection of the constrained zono-*



tope and randomly generated hyperplanes. Iteration of **Algorithm 1** to further refine  $E$  was only required in 13 of these 100 cases.

## 2.5 Redundancy Removal

It is important to recognize that certain set operations can create redundancy in the set representation. For example, the Minkowski sum can create redundancy in the resultant zonotope if the two operands have parallel generators. Additionally, the generalized intersection can create redundancy within the generators and constraints of a constrained zonotope. Detecting and removing this redundancy can provide order reduction without reducing the volume of the set. As proposed in [3], if a zonotope  $Z = \{\mathbf{G}, \mathbf{c}\}$  has parallel generators,  $\mathbf{g}_i \parallel \mathbf{g}_j$ , then the same set can be represented using one less generator by simply combining parallel generators through addition  $\mathbf{g}_i + \mathbf{g}_j$ . For a zonotope in  $\mathbb{R}^n$  with  $n_g$  generators, parallel generators can be detected and combined using a typical sorting algorithm with complexity  $O(nn_g^2)$ . To set a desired numerical precision, two generators are considered parallel if  $\frac{|\mathbf{g}_i^T \mathbf{g}_j|}{\|\mathbf{g}_i\|_2 \|\mathbf{g}_j\|_2} \geq 1 - \epsilon$ , where  $\epsilon > 0$  is a small number. The same is true for a constrained zonotope  $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\}$  if the lifted zonotope [79]

$$Z^+ = \left\{ \begin{bmatrix} \mathbf{G} \\ \mathbf{A} \end{bmatrix}, \begin{bmatrix} \mathbf{c} \\ -\mathbf{b} \end{bmatrix} \right\} = \{\mathbf{G}^+, \mathbf{c}^+\},$$

has parallel generators,  $\mathbf{g}_i^+ \parallel \mathbf{g}_j^+$ . In this case, the parallel generators can be similarly reduced but with higher complexity  $O(n + n_c)n_g^2$  due to the  $n_c$  constraints added to the rows of the lifted zonotope structure. Once the reduced lifted zonotope is obtained, it is transformed back to a reduced constrained zonotope with fewer generators.

For constrained zonotopes, redundancy can also come from the combination of constraints  $\mathbf{A}\boldsymbol{\xi} = \mathbf{b}$  and  $\|\boldsymbol{\xi}\|_\infty \leq 1$ . By representing these constraints as

$$\mathbf{A}\boldsymbol{\xi} = \mathbf{b} \iff \sum_{j \in \{1, \dots, n_g\}} a_{i,j} \xi_j = b_i, \forall i \in \{1, \dots, n_c\}, \quad (2.12)$$

and  $\|\boldsymbol{\xi}\|_\infty \leq 1 \Leftrightarrow |\xi_j| \leq 1, \forall j \in \{1, \dots, n_g\}$ , the following theorem provides a condition for detecting redundancy and a method for removing one generator and one constraint with complexity  $O(n_c n_g^2)$  based on an approach originally presented in [79].

**Theorem 2.** For  $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\} \subset \mathbb{R}^n$  with  $n_g$  generators and  $n_c$  constraints, if there exists indices  $r \in \{1, \dots, n_g\}$  and  $c \in \{1, \dots, n_c\}$  such that  $a_{r,c} \neq 0$  and

$$R_{r,c} \triangleq a_{r,c}^{-1} b_r - a_{r,c}^{-1} \sum_{k \neq c} a_{r,k} E_k \subseteq [-1, 1], \quad (2.13)$$

with  $E_k$  computed using **Algorithm 1**, then  $Z_c$  can be exactly represented by a constrained zonotope  $Z_r$  with  $n_g - 1$  generators and  $n_c - 1$  constraints.

*Proof.* Following the procedure in [79], let

$$Z_r = \{\mathbf{G} - \boldsymbol{\Lambda}_G \mathbf{A}, \mathbf{c} + \boldsymbol{\Lambda}_G \mathbf{b}, \mathbf{A} - \boldsymbol{\Lambda}_A \mathbf{A}, \mathbf{b} - \boldsymbol{\Lambda}_A \mathbf{b}\},$$

where  $\boldsymbol{\Lambda}_G = \mathbf{G} \mathbf{E}_{c,r} a_{r,c}^{-1} \in \mathbb{R}^{n \times n_c}$ ,  $\boldsymbol{\Lambda}_A = \mathbf{A} \mathbf{E}_{c,r} a_{r,c}^{-1} \in \mathbb{R}^{n_c \times n_c}$ , and  $\mathbf{E}_{c,r} \in \mathbb{R}^{n_g \times n_c}$  is zero except for a one in the  $(c, r)$  position. With  $Z_r = \{\mathbf{G}_r, \mathbf{c}_r, \mathbf{A}_r, \mathbf{b}_r\}$ , this transformation uses the  $r^{\text{th}}$  of row of (2.12) to solve for  $\xi_c$  in terms of  $\xi_k, k \neq c$ . This results in the  $c^{\text{th}}$  column of  $\mathbf{G}_r$  and  $\mathbf{A}_r$  and the  $r^{\text{th}}$  row of  $\mathbf{A}_r$  to equal zero. Removing these columns and rows of zeros results in a constrained zonotope with  $n_g - 1$  generators and  $n_c - 1$  constraints. Through this transformation, the  $r^{\text{th}}$  constraint is still imposed in  $Z_r$  but the ability to constraint  $|\xi_c| \leq 1$  is lost. However, since  $R_{r,c} \subseteq [-1, 1]$ , this constraint is imposed by the remaining equality and norm constraints, and thus  $Z_r = Z_c$ .  $\square$

Note that this is the same approach presented in [79] used to achieve constraint reduction in the context of obtaining an outer-approximating constrained zonotope  $\tilde{Z}_c$  satisfying  $\tilde{Z}_c \supset Z_c$  with one less generator and one less constraint. However, as the following example demonstrates, it is important to realize that this approach can also be used to identify and

remove redundancy in the set representation while retaining the exact set. As in [79], Gauss-Jordan elimination with full pivoting should be applied to  $Z_c$  prior to applying **Algorithm 1** to determine the intervals  $E_k$  required to compute (2.13). The procedure discussed in the proof of **Theorem 2** can be applied iteratively until  $R_{r,c} \not\subseteq [-1, 1]$  for any indices. However, there is no guarantee that the resulting constrained zonotope will be without redundancy since **Theorem 2** only provides a sufficient condition.

**Example 2.** Consider the two zonotopes shown in Fig. 2.2

$$Z_1 = \left\{ \left[ \begin{array}{cc} 1 & 1 \\ 1 & -1 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \end{array} \right] \right\}, \quad Z_2 = \left\{ \left[ \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \end{array} \right] \right\},$$

and the constrained zonotope  $Z_c = Z_1 \cap Z_2$ . Applying (2.3) results in

$$Z_c = \left\{ \left[ \begin{array}{cccc} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \end{array} \right], \left[ \begin{array}{cccc} 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & -1 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \end{array} \right] \right\}, \quad (2.14)$$

with  $n_g = 4$  generators and  $n_c = 2$  constraints. However, since  $Z_2 \subset Z_1$ , the intersection is also represented exactly by  $Z_2$ . By applying Gauss-Jordan elimination with full pivoting and two iterations of the procedure from **Theorem 2**, two constraints and two generators are removed to reduce  $Z_c$  from (2.14) to  $Z_c = Z_2$  with  $n_g = 2$  and  $n_c = 0$ . To provide an unbiased evaluation of **Theorem 2**, the axis-aligned generators of  $Z_2$  above were replaced by randomly chosen generators. In each of the 45 out of 100 cases where  $Z_2 \subseteq Z_1$ ,  $Z_c$  was successfully reduced to  $Z_c = Z_2$  with  $n_g = 2$  and  $n_c = 0$ .

**Remark 3.** For a constrained zonotope  $Z_c$  with  $n_c$  constraints and  $n_g$  generators and a set  $H$  in  $H$ -Rep with  $n_h$  halfspaces, **Algorithm 1** can be applied in two different ways to either prevent or remove redundancy in the set representation of  $Z_c \cap H$ . The approach from Section 2.4 based on preventing the addition of unnecessary generators and constraints has a best-case complexity of  $O(n_h n_c n_g^2)$  if  $Z_c \subset H$  and a worst-case complexity of  $O(n_h(n_c + n_h)(n_g + n_h)^2)$

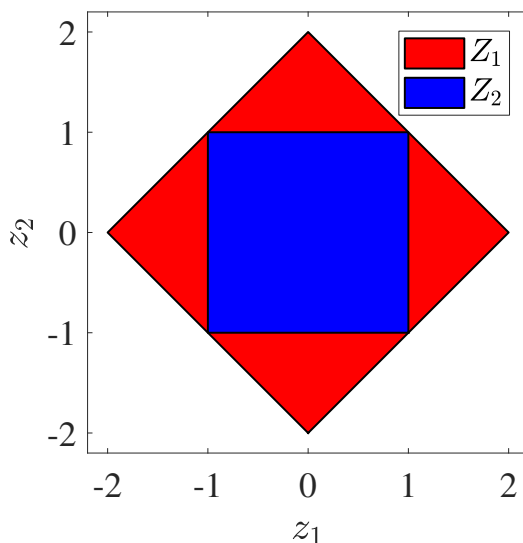


Figure 2.2. Zonotopes  $Z_1$  and  $Z_2$ , where  $Z_1 \cap Z_2 = Z_2$ , used to demonstrate the ability to remove redundancy from constrained zonotopes that can arise from operations like the generalized intersection.

if  $Z_c$  intersects each of the  $n_h$  halfspaces. Alternatively,  $n_h$  constraints and  $n_h$  generators can be directly added to  $Z_c$  using (2.11) and then **Theorem 2** can be applied to reduce set complexity. This approach has a best-case complexity of  $O((n_c + n_h)(n_g + n_h)^2)$  when no generators/constraints can be removed and a worst-case complexity  $O(n_h(n_c + n_h)(n_g + n_h)^2)$  when all of the added  $n_h$  constraints and  $n_h$  generators can be removed. Thus, both approaches have the same worst-case complexity but the preventative approach has the potential to require fewer computations in practice.

## 2.6 Inner Approximations

### 2.6.1 Motivation

Outer-approximations are widely used in the field of reachability analysis for system verification to determine if a system will always operate in a desired region of the state space [28, 29]. However, in many applications there is a need for computing reduced-order inner-

approximations. In general computing inner-approximations of sets is considered a more difficult problem [52]. Inner-approximations are particularly important when computing backward reachable sets that define a set of initial states for which a system will enter a specified target region after some allotted time [92]. While there are existing techniques for zonotopes [29, 35], inner-approximation techniques for constrained zonotopes are lacking. This section establishes inner-approximation order reduction for zonotopes and constrained zonotopes.

### 2.6.2 Zonotopes

The proposed reduced-order inner-approximation of a zonotope requires the following zonotope containment conditions.

**Lemma 3.** (Theorem 3 of [71]) *Given two zonotopes  $X = \{\mathbf{G}_x, \mathbf{c}_x\} \subset \mathbb{R}^n$  and  $Y = \{\mathbf{G}_y, \mathbf{c}_y\} \subset \mathbb{R}^n$ ,  $X \subseteq Y$  if there exists  $\mathbf{\Gamma} \in \mathbb{R}^{n_y \times n_x}$  and  $\boldsymbol{\beta} \in \mathbb{R}^{n_y}$  such that*

$$\mathbf{G}_x = \mathbf{G}_y \mathbf{\Gamma}, \quad \mathbf{c}_y - \mathbf{c}_x = \mathbf{G}_y \boldsymbol{\beta}, \quad |\mathbf{\Gamma}| \mathbf{1} + |\boldsymbol{\beta}| \leq \mathbf{1}. \quad (2.15)$$

**Theorem 3.** *The zonotope  $Z_r = \{\mathbf{G}_r, \mathbf{c}\} \subset \mathbb{R}^n$  is a reduced-order inner-approximation of  $Z = \{\mathbf{G}, \mathbf{c}\} \subset \mathbb{R}^n$  such that  $Z_r \subseteq Z$  with  $\mathbf{G}_r \in \mathbb{R}^{n \times n_r}$ ,  $\mathbf{G} \in \mathbb{R}^{n \times n_g}$ , and  $n_r < n_g$  if  $\mathbf{G}_r = \mathbf{G} \mathbf{T}$  where  $\mathbf{T} = [t_{i,j}] \in \mathbb{R}^{n_g \times n_r}$ ,  $t_{i,j} \in \{-1, 0, 1\}$ , and  $\sum_{j=1}^{n_r} |t_{i,j}| = 1, \forall i \in \{1, \dots, n_g\}$ .*

*Proof.* From **Lemma 3**,  $Z_r \subseteq Z$  if there exist  $\mathbf{\Gamma} \in \mathbb{R}^{n_g \times n_r}$  and  $\boldsymbol{\beta} \in \mathbb{R}^{n_g}$  such that

$$\mathbf{G} \mathbf{T} = \mathbf{G} \mathbf{\Gamma}, \quad \mathbf{c} - \mathbf{c} = \mathbf{G} \boldsymbol{\beta}, \quad |\mathbf{\Gamma}| \mathbf{1} + |\boldsymbol{\beta}| \leq \mathbf{1}.$$

The first two equations hold by setting  $\mathbf{\Gamma} = \mathbf{T}$  and  $\boldsymbol{\beta} = \mathbf{0}$ . The third equation holds since  $\sum_{j=1}^{n_r} |t_{i,j}| = 1, \forall i \in \{1, \dots, n_g\}$ , if and only if  $|\mathbf{T}| \mathbf{1} = \mathbf{1}$ .  $\square$

The specific definition of  $\mathbf{T}$  in **Theorem 3** produces an inner-approximation of  $Z$  by forming the generators of  $Z_r$  through the addition of the generators in  $Z$ . Typically, the

largest inner-approximation of  $Z$  is desired. The proposed method for determining  $\mathbf{T}$  is inspired by the methods for determining outer-approximations of zonotopes presented in [51]. First, let the generators  $\mathbf{g}_i$  of  $Z$  be arranged such that  $\|\mathbf{g}_i\|_2 \geq \|\mathbf{g}_{i+1}\|_2$ ,  $\forall i \in \{1, \dots, n_g - 1\}$ . Then partition the generator matrix such that  $\mathbf{G} = [\mathbf{G}_1 \ \mathbf{G}_2]$  where  $\mathbf{G}_1 \in \mathbb{R}^{n \times n_r}$  and  $\mathbf{G}_2 \in \mathbb{R}^{n \times (n_g - n_r)}$ . For each generator  $\mathbf{g}_{2,j}$  in  $\mathbf{G}_2$ , compute the magnitude of the dot product  $\alpha_{i,j} = |\mathbf{g}_{1,i}^T \mathbf{g}_{2,j}|$  with all generators  $\mathbf{g}_{1,i}$  in  $\mathbf{G}_1$ . The goal is to add the generators  $\mathbf{g}_{2,j}$  to the most aligned generator  $\mathbf{g}_{1,i}$ . Thus, let  $\mathbf{T} = [t_{i,j}]$  where

$$t_{i,j} = \begin{cases} 1 & \text{if } i = j \leq n_r \\ \frac{1}{\alpha_{i,j}} \mathbf{g}_{1,i}^T \mathbf{g}_{2,j} & \text{if } \alpha_{i,j} > \alpha_{i,k}, \forall k \neq j \\ 0 & \text{otherwise} \end{cases}. \quad (2.16)$$

Note that computing  $Z_r$  using **Theorem 3** and (2.16) has an overall complexity of  $O(nn_g^2 + nn_g n_r)$ , where the first term is associated with sorting the generators based on the 2-norm and the second term is associated with computing the product  $\mathbf{G}_r = \mathbf{G}\mathbf{T}$  in **Theorem 3**.

**Example 3.** Consider the zonotope

$$Z = \left\{ \begin{bmatrix} 4 & 3 & -2 & 0.2 & 0.5 \\ 0 & 2 & 3 & 0.6 & -0.3 \end{bmatrix}, \mathbf{0} \right\} \subset \mathbb{R}^2.$$

Note that the generators are already arranged in order of decreasing 2-norm. With  $n_g = 5$ , the goal is to determine  $Z_r \subseteq Z$  such that  $n_r = 3$ . From **Theorem 3** and (2.16), the matrix  $\mathbf{T}$  and the reduced-order zonotope  $Z_r$  are

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad Z_r = \left\{ \begin{bmatrix} 4.5 & 3.2 & -2 \\ -0.3 & 2.6 & 3 \end{bmatrix}, \mathbf{0} \right\}.$$

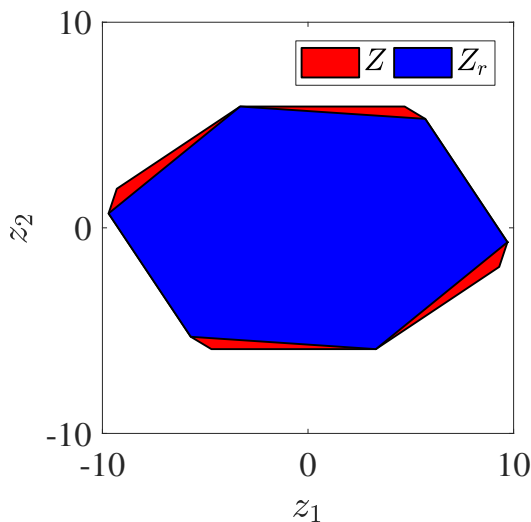


Figure 2.3. The inner-approximation of  $Z$  with  $n_g = 5$  by the reduced-order zonotope  $Z_r$  with  $n_r = 3$ .

*Fig. 2.3 confirms  $Z_r \subseteq Z$  with volume ratio  $V_r = 0.97$ . While this numerical example resulted in relatively large volume ratio, the reduction in volume is highly dependent on the distribution of generator lengths and the number of generators removed. For 100 randomly generated zonotopes in  $\mathbb{R}^2$  with  $n_g = 5$ , applying **Theorem 3** and (2.16), resulted in all reduced zonotopes satisfying  $Z_r \subseteq Z$  with  $n_r = 3$  and mean volume ratio  $V_r = 0.84$ .*

### 2.6.3 Constrained Zonotopes

For constrained zonotopes, a reduced-order inner-approximation  $Z_r$  of  $Z_c$  can be computed based on the set containment criteria for the affine transformation of polytopes in H-Rep (AH-polytopes) developed in [71] since AH-polytopes and constrained zonotopes are equivalent.

**Definition 1.** [71] *An AH-polytope  $X \subset \mathbb{R}^n$  is an affine transformation of a H-Rep polytope  $P \subset \mathbb{R}^m$  where*

$$X = \bar{\mathbf{x}} + \mathbf{X}P, \quad \mathbf{X} \in \mathbb{R}^{n \times m}, \quad \bar{\mathbf{x}} \in \mathbb{R}^n. \quad (2.17)$$

The following theorem proves the equivalency between constrained zonotopes and AH-polytopes in addition to providing a method to convert constrained zonotopes to AH-polytopes with complexity  $O(nn_g^2 + n_c^2n_g)$ , where the first term is associated with computing an affine transformation and the second term is associated with computing the basis of  $\mathcal{N}(\mathbf{A})$  for  $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\}$ .

**Theorem 4.** *A non-empty set  $Z_c \subset \mathbb{R}^n$  is a constrained zonotope if and only if it is an AH-polytope.*

*Proof.* To prove that every AH-polytope is a constrained zonotope, let  $P = \{\mathbf{z} \in \mathbb{R}^m \mid \mathbf{H}\mathbf{z} \leq \mathbf{k}\}$ . Per Theorem 1 in [79], the set  $P$  can always be represented as a constrained zonotope  $P = \{\mathbf{G}_p, \mathbf{c}_p, \mathbf{A}_p, \mathbf{b}_p\}$ . Thus, from (2.17) and the properties of constrained zonotopes (2.1) and (2.2),  $X$  is a constrained zonotope where  $X = \{\mathbf{X}\mathbf{G}_p, \bar{\mathbf{x}} + \mathbf{X}\mathbf{c}_p, \mathbf{A}_p, \mathbf{b}_p\}$ . To prove that every constrained zonotope is an AH-polytope, consider  $Z_c = \{\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}\}$  with  $n_g$  generators and  $n_c$  constraints. If  $n_c = 0$ ,  $Z_c = Z = \{\mathbf{G}, \mathbf{c}\}$  is a zonotope and can be represented in AH-polytope form of (2.17) with  $\bar{\mathbf{x}} = \mathbf{c}$ ,  $\mathbf{X} = \mathbf{G}$ , and  $P = B_\infty$ . For  $n_c > 0$ , assume that any rank deficiency in  $\mathbf{A}$  has been detected as a row of zeros in the reduced row echelon form achieved through Gauss-Jordan elimination with full pivoting (see [79] for details). Thus, the rank of  $\mathbf{A}$  is  $n_c$  and there exists  $\mathbf{s} = \mathbf{A}^\dagger \mathbf{b} \in \mathbb{R}^{n_g}$  and the matrix  $\mathbf{T} \in \mathbb{R}^{n_g \times (n_g - n_c)}$  with columns that form a basis for  $\mathcal{N}(\mathbf{A})$ . Using the change of variables  $\boldsymbol{\xi} = \mathbf{T}\bar{\boldsymbol{\xi}} + \mathbf{s}$ , the equality constraint  $\mathbf{A}\boldsymbol{\xi} = \mathbf{b}$  is satisfied for all  $\bar{\boldsymbol{\xi}} \in \mathbb{R}^{n_g - n_c}$ . Moreover, since  $\mathbf{T}$  has linearly independent columns, any solution  $\boldsymbol{\xi}$  of  $\mathbf{A}\boldsymbol{\xi} = \mathbf{b}$  can be achieved by a corresponding value of  $\bar{\boldsymbol{\xi}}$ . Specifically,  $\bar{\boldsymbol{\xi}} = \mathbf{T}^\dagger(\boldsymbol{\xi} - \mathbf{s})$ , where  $\mathbf{T}^\dagger$  is the left pseudoinverse of  $\mathbf{T}$ . Hence,  $Z_c$  can be expressed exactly as

$$Z_c = \{\mathbf{c} + \mathbf{G}\mathbf{s} + \mathbf{G}\mathbf{T}\bar{\boldsymbol{\xi}} \mid \|\mathbf{T}\bar{\boldsymbol{\xi}} + \mathbf{s}\|_\infty \leq 1\}.$$



Furthermore, the norm constraints  $\|\mathbf{T}\bar{\boldsymbol{\xi}} + \mathbf{s}\|_\infty \leq 1$  can be represented in H-Rep as  $P = \{\bar{\boldsymbol{\xi}} \mid \mathbf{H}\bar{\boldsymbol{\xi}} \leq \mathbf{k}\}$ , where

$$\mathbf{H} = \begin{bmatrix} \mathbf{T} \\ -\mathbf{T} \end{bmatrix}, \quad \mathbf{k} = \begin{bmatrix} \mathbf{1} - \mathbf{s} \\ \mathbf{1} + \mathbf{s} \end{bmatrix}.$$

Thus, with  $\bar{\mathbf{x}} = \mathbf{c} + \mathbf{G}\mathbf{s}$  and  $\mathbf{X} = \mathbf{G}\mathbf{T}$ ,  $Z_c$  is an AH-polytope of the form (2.17).  $\square$

**Remark 4.** *The convexity of the constrained zonotope  $Z_c = (\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b})$  also facilitates representation as a polynomial zonotope  $Z_p = (\mathbf{c}, \mathbf{G}, \mathbf{E})$  in Z-Rep [45]. However, for the reverse direction, an algorithm is yet to be found.*

**Lemma 4.** *(Theorem 1 of [71]) Given AH-polytopes  $X, Y \subset \mathbb{R}^n$  where  $X = \bar{\mathbf{x}} + \mathbf{X}P_x$ ,  $Y = \bar{\mathbf{y}} + \mathbf{Y}P_y$ ,  $P_x = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid \mathbf{H}_x\mathbf{x} \leq \mathbf{f}_x\}$ , and  $P_y = \{\mathbf{y} \in \mathbb{R}^{n_y} \mid \mathbf{H}_y\mathbf{y} \leq \mathbf{f}_y\}$ ,  $X \subseteq Y$  if there exists  $\boldsymbol{\Gamma} \in \mathbb{R}^{n_y \times n_x}$ ,  $\boldsymbol{\beta} \in \mathbb{R}^{n_y}$  and  $\boldsymbol{\Lambda} \in \mathbb{R}_+^{n_{hy} \times n_{hx}}$  such that*

$$\mathbf{X} = \mathbf{Y}\boldsymbol{\Gamma}, \quad \bar{\mathbf{y}} - \bar{\mathbf{x}} = \mathbf{Y}\boldsymbol{\beta}, \quad (2.18a)$$

$$\boldsymbol{\Lambda}\mathbf{H}_x = \mathbf{H}_y\boldsymbol{\Gamma}, \quad \boldsymbol{\Lambda}\mathbf{f}_x \leq \mathbf{f}_y + \mathbf{H}_y\boldsymbol{\beta}. \quad (2.18b)$$

To achieve a reduced-order inner-approximation  $Z_r$  of constrained zonotope  $Z_c$ , **Theorem 4** can be used to convert both  $Z_r$  and  $Z_c$  into AH-polytopes while **Lemma 4** can be used to ensure  $Z_r \subseteq Z_c$ . Assuming  $Z_c$  is known, consider  $Z_r = \{\mathbf{G}_r\boldsymbol{\Phi}, \mathbf{c}_r, \mathbf{A}_r, \mathbf{b}_r\}$  where  $\boldsymbol{\Phi} = \text{diag}(\boldsymbol{\phi})$  is a scaling matrix with  $\phi_i > 0, \forall i \in \{1, \dots, n_{gr}\}$ . Assuming  $\mathbf{G}_r$ ,  $\mathbf{A}_r$ , and  $\mathbf{b}_r$  are known, the following optimization problem can be formulated with  $4n_{gr}^2 + n_{gr} + (n_g - n_c)(1 + n_{gr} - n_{cr}) + n$  decision variables that maximizes the  $p = 1, 2$ , or  $\infty$  norm of the diagonal elements  $\boldsymbol{\phi}$  of the scaling matrix  $\boldsymbol{\Phi}$  by solving

$$\max_{\boldsymbol{\Phi}, \boldsymbol{\Gamma}, \boldsymbol{\beta}, \boldsymbol{\Lambda}, \mathbf{c}_r} \|\boldsymbol{\phi}\|_p, \quad (2.19a)$$

s.t.

$$(\mathbf{c} + \mathbf{G}\mathbf{s}) - (\mathbf{c}_r + \mathbf{G}_r\boldsymbol{\Phi}\mathbf{s}_r) = \mathbf{G}\mathbf{T}\boldsymbol{\beta}, \quad (2.19b)$$

$$\mathbf{G}_r \Phi \mathbf{T}_r = \mathbf{G} \mathbf{T} \mathbf{\Gamma}, \quad \mathbf{\Lambda} \begin{bmatrix} \mathbf{T}_r \\ -\mathbf{T}_r \end{bmatrix} = \begin{bmatrix} \mathbf{T} \\ -\mathbf{T} \end{bmatrix} \mathbf{\Gamma}, \quad (2.19c)$$

$$\mathbf{\Lambda} \begin{bmatrix} \mathbf{1} - \mathbf{s}_r \\ \mathbf{1} + \mathbf{s}_r \end{bmatrix} \leq \begin{bmatrix} \mathbf{1} - \mathbf{s} \\ \mathbf{1} + \mathbf{s} \end{bmatrix} + \begin{bmatrix} \mathbf{T} \\ -\mathbf{T} \end{bmatrix} \boldsymbol{\beta}, \quad (2.19d)$$

with parameters  $\mathbf{s} = \mathbf{A}^\dagger \mathbf{b} \in \mathbb{R}^{n_g}$ ,  $\mathbf{s}_r = \mathbf{A}_r^\dagger \mathbf{b}_r \in \mathbb{R}^{n_{gr}}$ , and matrices  $\mathbf{T} \in \mathbb{R}^{n_g \times (n_g - n_c)}$ ,  $\mathbf{T}_r \in \mathbb{R}^{n_{gr} \times (n_{gr} - n_{cr})}$  with columns that form bases for  $\mathcal{N}(\mathbf{A})$  and  $\mathcal{N}(\mathbf{A}_r)$ , respectively. Note that the majority of the decision variables in (2.19) come from the matrices  $\mathbf{\Gamma} \in \mathbb{R}^{(n_g - n_c) \times (n_{gr} - n_{cr})}$  and  $\mathbf{\Lambda} \in \mathbb{R}_+^{2n_{gr} \times 2n_{gr}}$ . While this procedure applies to any  $Z_r$ , the process discussed in Section 2.5 can be used to compute  $Z_r$  by removing exactly one constraint and one generator from  $Z_c$ . For the case where  $Z_c$  satisfies the conditions in **Theorem 2**, the  $r^{\text{th}}$  constraint and the  $c^{\text{th}}$  generators were chosen such that  $R_{r,c} \subseteq [-1, 1]$  and thus an exact reduced-order representation was achieved with  $Z_r = Z_c$ . To achieve further reduction through the inner-approximation of  $Z_c$ , the same procedure from Section 2.5 can be applied by choosing appropriate indices and scaling  $Z_r$  via optimization while enforcing  $Z_r \subseteq Z_c$  using the constraints from (2.19). Since  $R_j = [\rho_j^L, \rho_j^U]$  represents the range of  $\xi_j$  if the constraints  $|\xi_j| \leq 1$  were omitted [79], the  $c^{\text{th}}$  generator should be removed that minimizes  $\max(|\rho_j^L|, |\rho_j^U|)$ . Once  $c$  is chosen,  $r$  should be chosen such that the entry in the  $(r, c)$  position of  $\mathbf{A}_r$  has the largest absolute value of all entries in the  $c^{\text{th}}$  column.

**Example 4.** Consider the constrained zonotope  $Z_c$  shown in Fig. 2.4 where

$$Z_c = \left\{ \left[ \begin{array}{cccc} -1 & 3 & 4 & 0 & 0 \\ 4 & -2 & -5 & 0 & 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \end{array} \right], \left[ \begin{array}{cccc} -1 & 3 & 4 & 6.5 & 0 \\ 4 & -2 & -5 & 0 & 8 \end{array} \right], \left[ \begin{array}{c} -1.5 \\ -3 \end{array} \right] \right\}.$$

First, Gauss-Jordan elimination with full pivoting was applied to  $Z_c$ , followed by the transformation in **Theorem 2** by picking the  $c^{\text{th}}$  generator that minimizes  $\max(|\rho_j^L|, |\rho_j^U|)$  and the  $r^{\text{th}}$  row with the largest entry in  $c^{\text{th}}$  column of  $\mathbf{A}$ . Then an LP was formulated and solved

using the constraints from (2.18) and a cost function that maximized  $\|\phi\|_\infty$ . The resulting reduced-order zonotope  $Z_r$  is shown in Fig. 2.4 where

$$Z_r = \left\{ \begin{array}{l} \left[ \begin{array}{cccc} 0 & 3.17 & 2.38 & -0.79 \\ 0 & -3.97 & -1.59 & 3.17 \end{array} \right], \left[ \begin{array}{c} -1.34 \\ 1.03 \end{array} \right], \\ \left[ \begin{array}{cccc} 1 & -0.63 & -0.25 & 0.50 \end{array} \right], \left[ \begin{array}{c} -0.38 \end{array} \right] \end{array} \right\}.$$

Using a similar approach, Fig. 2.4 also shows the inner-approximations of  $Z_c$  by zonotope  $Z$  and interval set  $B$  where

$$Z = \left\{ \begin{array}{l} \left[ \begin{array}{ccc} 2.31 & 1.93 & -0.27 \\ -2.84 & -0.94 & 2.57 \end{array} \right], \left[ \begin{array}{c} 0.49 \\ -1.35 \end{array} \right] \end{array} \right\},$$

$$B = \left\{ \begin{array}{l} \left[ \begin{array}{c} 2 \ 0 \\ 0 \ 2 \end{array} \right], \left[ \begin{array}{c} 2.55 \\ -3.18 \end{array} \right] \end{array} \right\}.$$

To compute  $Z$ , the equality constraints from  $Z_c$  were removed via the same change of variables used in the proof of **Theorem 4**. Typically this would result in an outer-approximation of  $Z_c$ , however the scaling matrix  $\Phi$  is used to reduce the length of each generator such that  $Z \subseteq Z_c$ . For the interval set  $B$ , the generator matrix is initialized as the identity matrix and then scaled by  $\Phi$ . The resulting volume ratios with respect to  $Z_c$  are  $V_r = 0.86$ ,  $V_r = 0.83$ ,  $V_r = 0.46$  for  $Z_r$ ,  $Z$ , and  $B$ , respectively. Repeating this process for 100 randomly generated constrained zonotopes with  $4 \leq n_g \leq 20$  and  $1 \leq n_c \leq \frac{1}{2}n_g$ , Fig. 2.5 shows the volume ratios for constrained zonotope, zonotope, and interval set inner-approximations. Both constrained zonotopes and zonotopes provide better approximations compared to interval sets while constrained zonotopes provide only a slightly higher mean volume ratio.

## 2.7 Convex Hulls

This section computes the CG-Rep of the convex hull of two constrained zonotopes  $Z_1, Z_2 \subset \mathbb{R}^n$  with complexity  $O(n + n_{c1} + n_{c2})$  where  $n_{c1}$  and  $n_{c2}$  are the number of constraints in  $Z_1$

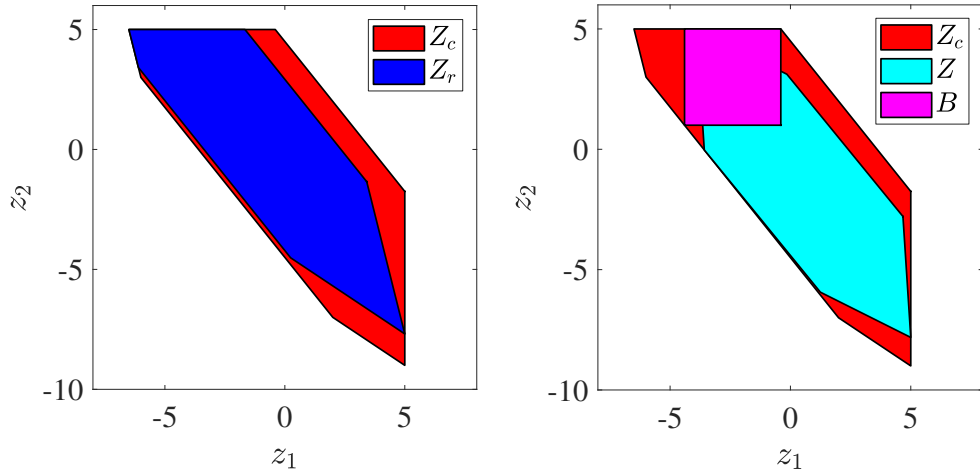


Figure 2.4. Left: The inner-approximation of  $Z_c$  by a constrained zonotope  $Z_r$  with one less generator and constraint. Right: The inner-approximation of  $Z_c$  by a zonotope  $Z$  and an interval set  $B$ .

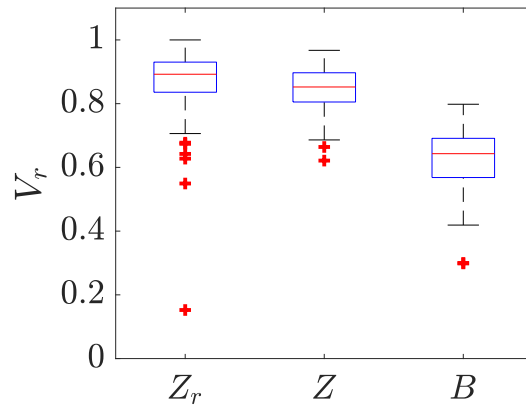


Figure 2.5. The volume ratios for the inner-approximation of 100 randomly generated constrained zonotopes by a constrained zonotope  $Z_r$  with one less generator and constraint, a zonotope  $Z$ , and an interval set  $B$ . The red crosses denote outliers that do not fit the box plot distribution.

and  $Z_2$ , respectively. Since zonotopes are a subset of constrained zonotopes with  $n_c = 0$ , the following result also applies to zonotopes.

**Definition 2.** [83] *The convex hull of the union of two polytopes  $P_1, P_2 \subset \mathbb{R}^n$  is defined as*

$$CH(P_1 \cup P_2) \triangleq \left\{ \begin{array}{l} \mathbf{x}_1 \in P_1, \\ \mathbf{x}_1\lambda + \mathbf{x}_2(1 - \lambda) \mid \mathbf{x}_2 \in P_2, \\ 0 \leq \lambda \leq 1 \end{array} \right\}.$$

**Theorem 5.** *The convex hull of the union of two constrained zonotopes  $Z_1 = \{\mathbf{G}_1, \mathbf{c}_1, \mathbf{A}_1, \mathbf{b}_1\} \subset \mathbb{R}^n$  and  $Z_2 = \{\mathbf{G}_2, \mathbf{c}_2, \mathbf{A}_2, \mathbf{b}_2\} \subset \mathbb{R}^n$  is a constrained zonotope  $Z_h = \{\mathbf{G}_h, \mathbf{c}_h, \mathbf{A}_h, \mathbf{b}_h\}$  where*

$$\begin{aligned} \mathbf{G}_h &= \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 & \frac{\mathbf{c}_1 - \mathbf{c}_2}{2} & \mathbf{0} \end{bmatrix}, & \mathbf{c}_h &= \frac{\mathbf{c}_1 + \mathbf{c}_2}{2}, \\ \mathbf{A}_h &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & -\frac{\mathbf{b}_1}{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \frac{\mathbf{b}_2}{2} & \mathbf{0} \\ \mathbf{A}_{3,1} & \mathbf{A}_{3,2} & \mathbf{A}_{3,0} & \mathbf{I} \end{bmatrix}, & \mathbf{b}_h &= \begin{bmatrix} \frac{1}{2}\mathbf{b}_1 \\ \frac{1}{2}\mathbf{b}_2 \\ -\frac{1}{2}\mathbf{1} \end{bmatrix}, \\ \mathbf{A}_{3,1} &= \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, & \mathbf{A}_{3,2} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \\ -\mathbf{I} \end{bmatrix}, & \mathbf{A}_{3,0} &= \begin{bmatrix} -\frac{1}{2}\mathbf{1} \\ -\frac{1}{2}\mathbf{1} \\ \frac{1}{2}\mathbf{1} \\ \frac{1}{2}\mathbf{1} \end{bmatrix}. \end{aligned}$$

*Proof.* Considering any element  $\mathbf{x} \in Z_h$ , it is to be proven that  $\mathbf{x} \in CH(Z_1 \cup Z_2)$ . By the definition of  $Z_h$ ,  $\exists \boldsymbol{\xi}_1 \in \mathbb{R}^{n_{g1}}, \boldsymbol{\xi}_2 \in \mathbb{R}^{n_{g2}}, \xi_0 \in \mathbb{R}$ , and  $\boldsymbol{\xi}_s \in \mathbb{R}^{2(n_{g1} + n_{g2})}$  such that

$$\mathbf{x} = \mathbf{G}_1\boldsymbol{\xi}_1 + \mathbf{G}_2\boldsymbol{\xi}_2 + \frac{\mathbf{c}_1 - \mathbf{c}_2}{2}\xi_0 + \mathbf{0}\boldsymbol{\xi}_s + \frac{\mathbf{c}_1 + \mathbf{c}_2}{2}, \quad (2.20a)$$

$$\|\boldsymbol{\xi}_1\|_\infty \leq 1, \|\boldsymbol{\xi}_2\|_\infty \leq 1, |\xi_0| \leq 1, \|\boldsymbol{\xi}_s\|_\infty \leq 1, \quad (2.20b)$$

$$\mathbf{A}_h[\boldsymbol{\xi}_1^T \ \boldsymbol{\xi}_2^T \ \xi_0 \ \boldsymbol{\xi}_s^T]^T = \mathbf{b}_h. \quad (2.20c)$$

To prove  $\mathbf{x} \in CH(Z_1 \cup Z_2)$  requires the existence of elements  $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^n$ ,  $\lambda \in \mathbb{R}$ ,  $\boldsymbol{\xi}'_1 \in \mathbb{R}^{n_{g1}}$ , and  $\boldsymbol{\xi}'_2 \in \mathbb{R}^{n_{g2}}$  such that

$$\mathbf{x} = \mathbf{z}_1\lambda + \mathbf{z}_2(1 - \lambda), \quad 0 \leq \lambda \leq 1, \quad (2.21a)$$

$$\mathbf{z}_1 = \mathbf{c}_1 + \mathbf{G}_1 \boldsymbol{\xi}'_1, \quad \|\boldsymbol{\xi}'_1\|_\infty \leq \mathbf{1}, \quad \mathbf{A}_1 \boldsymbol{\xi}'_1 = \mathbf{b}_1, \quad (2.21b)$$

$$\mathbf{z}_2 = \mathbf{c}_2 + \mathbf{G}_2 \boldsymbol{\xi}'_2, \quad \|\boldsymbol{\xi}'_2\|_\infty \leq \mathbf{1}, \quad \mathbf{A}_2 \boldsymbol{\xi}'_2 = \mathbf{b}_2. \quad (2.21c)$$

Note that if  $\lambda = 0$ ,  $\mathbf{x} = \mathbf{z}_2$ , and only (2.21c) needs to hold to prove  $\mathbf{x} \in Z_2 \subset CH(Z_1 \cup Z_2)$ . Similarly, if  $\lambda = 1$ , then  $\mathbf{x} = \mathbf{z}_1$ , and only (2.21b) needs to hold to prove  $\mathbf{x} \in Z_1 \subset CH(Z_1 \cup Z_2)$ . Finally, if  $0 < \lambda < 1$ , then both (2.21b) and (2.21c) must hold to show  $\mathbf{x} \in CH(Z_1 \cup Z_2)$ . Consider the following definitions of variables  $\lambda$ ,  $\boldsymbol{\xi}'_1$ , and  $\boldsymbol{\xi}'_2$  with

$$\lambda = \frac{1}{2}(1 + \xi_0), \quad \boldsymbol{\xi}_1 = \boldsymbol{\xi}'_1 \lambda, \quad \boldsymbol{\xi}_2 = \boldsymbol{\xi}'_2 (1 - \lambda). \quad (2.22)$$

By rearranging (2.20a), substituting using the variable definitions in (2.22), and then rearranging to simplify using the definitions for  $\mathbf{z}_1$  and  $\mathbf{z}_2$  from (2.21b) and (2.21c), the expression for  $\mathbf{x}$  from (2.20a) can be established as

$$\mathbf{x} = \frac{\mathbf{c}_1}{2}(1 + \xi_0) + \mathbf{G}_1 \boldsymbol{\xi}_1 + \frac{\mathbf{c}_2}{2}(1 - \xi_0) + \mathbf{G}_2 \boldsymbol{\xi}_2, \quad (2.23a)$$

$$= \mathbf{c}_1 \lambda + \mathbf{G}_1 \boldsymbol{\xi}'_1 \lambda + \mathbf{c}_2 (1 - \lambda) + \mathbf{G}_2 \boldsymbol{\xi}'_2 (1 - \lambda), \quad (2.23b)$$

$$= \mathbf{z}_1 \lambda + \mathbf{z}_2 (1 - \lambda). \quad (2.23c)$$

Since  $|\xi_0| \leq 1$ , the definition for  $\lambda$  in (2.22) results in  $0 \leq \lambda \leq 1$ . From the definition of  $\mathbf{A}_h$  and  $\mathbf{b}_h$ , the first two sets of equality constraints are

$$\mathbf{A}_1 \boldsymbol{\xi}_1 - \frac{\mathbf{b}_1}{2} \xi_0 = \frac{1}{2} \mathbf{b}_1, \quad \mathbf{A}_2 \boldsymbol{\xi}_2 + \frac{\mathbf{b}_2}{2} \xi_0 = \frac{1}{2} \mathbf{b}_2. \quad (2.24)$$

For  $\lambda = 0$ , (2.22) results in  $\xi_0 = -1$  and  $\boldsymbol{\xi}_2 = \boldsymbol{\xi}'_2$ . Since  $\|\boldsymbol{\xi}_2\|_\infty \leq 1$ ,  $\|\boldsymbol{\xi}'_2\|_\infty \leq 1$  is satisfied. Substituting  $\xi_0 = -1$  in (2.24) results in  $\mathbf{A}_2 \boldsymbol{\xi}_2 = \mathbf{b}_2$ , and since  $\boldsymbol{\xi}_2 = \boldsymbol{\xi}'_2$ , the equality constraint  $\mathbf{A}_2 \boldsymbol{\xi}'_2 = \mathbf{b}_2$  is satisfied. Similar arguments can be used to show the existence of  $\boldsymbol{\xi}'_1$  for  $\lambda = 1$  that satisfy the equality and infinity norm constraints from (2.21b). For  $0 < \lambda < 1$ , (2.24) simplifies to  $\mathbf{A}_1 \boldsymbol{\xi}'_1 = \mathbf{b}_1$  and  $\mathbf{A}_2 \boldsymbol{\xi}'_2 = \mathbf{b}_2$  using the definitions in (2.22).

To show that norm constraints  $\|\boldsymbol{\xi}'_1\|_\infty \leq 1$  and  $\|\boldsymbol{\xi}'_2\|_\infty \leq 1$  are satisfied, consider the third set of equality constraints from (2.20c) based on the definitions of  $\mathbf{A}_h$  and  $\mathbf{b}_h$ . Specifically, the first and second rows of  $\mathbf{A}_{3,1}$ ,  $\mathbf{A}_{3,2}$ ,  $\mathbf{A}_{3,0}$ , and  $\mathbf{I}$  result in

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \boldsymbol{\xi}_1 - \frac{1}{2} \mathbf{1} \xi_0 + \boldsymbol{\xi}_{s,1} = -\frac{1}{2} \mathbf{1}, \quad (2.25)$$

where  $\boldsymbol{\xi}_{s,1}$  corresponds to the first  $2n_{g1}$  elements of  $\boldsymbol{\xi}_s$ . By substituting the variable definitions from (2.22) and rearranging, (2.25) reduces to

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \boldsymbol{\xi}'_1 - \mathbf{1} = \frac{1}{\lambda} (-\mathbf{1} - \boldsymbol{\xi}_{s,1}). \quad (2.26)$$

Since  $\|\boldsymbol{\xi}_{s,1}\|_\infty \leq 1$ , (2.26) is upper-bounded by

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \boldsymbol{\xi}'_1 - \mathbf{1} \leq \frac{1}{\lambda} (-\mathbf{1} + \mathbf{1}) = \mathbf{0}, \quad (2.27)$$

and thus,  $\boldsymbol{\xi}'_1 \leq \mathbf{1}$  and  $\boldsymbol{\xi}'_1 \geq -\mathbf{1}$  which implies  $\|\boldsymbol{\xi}'_1\|_\infty \leq 1$ . Next, by considering the third and fourth rows of  $\mathbf{A}_{3,1}$ ,  $\mathbf{A}_{3,2}$ ,  $\mathbf{A}_{3,0}$ , and  $\mathbf{I}$ , and using similar arguments, it can be shown that  $\|\boldsymbol{\xi}'_2\|_\infty \leq 1$ . Thus,  $\mathbf{x} \in CH(Z_1 \cup Z_2)$ .

Next, considering any  $\mathbf{x} \in CH(Z_1 \cup Z_2)$ , it is to be proven that  $\mathbf{x} \in Z_h$ . By **Definition 2**, there exists elements  $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^n$ ,  $\lambda \in \mathbb{R}$ ,  $\boldsymbol{\xi}'_1 \in \mathbb{R}^{n_{g1}}$ , and  $\boldsymbol{\xi}'_2 \in \mathbb{R}^{n_{g2}}$  such that (2.21a)-(2.21c) hold. To prove  $\mathbf{x} \in Z_h$  requires the existence of variables  $\boldsymbol{\xi}_1 \in \mathbb{R}^{n_{g1}}$ ,  $\boldsymbol{\xi}_2 \in \mathbb{R}^{n_{g2}}$ ,  $\xi_0 \in \mathbb{R}$ ,  $\boldsymbol{\xi}_s \in \mathbb{R}^{2(n_{g1}+n_{g2})}$  such that (2.20a)-(2.20c) hold. Consider the following definitions for variables  $\boldsymbol{\xi}_1$ ,  $\boldsymbol{\xi}_2$ ,  $\xi_0$ , and  $\boldsymbol{\xi}_s$  with

$$\boldsymbol{\xi}_1 = \boldsymbol{\xi}'_1 \lambda, \quad \boldsymbol{\xi}_2 = \boldsymbol{\xi}'_2 (1 - \lambda), \quad \xi_0 = 2\lambda - 1, \quad (2.28a)$$

$$\boldsymbol{\xi}_s = -\frac{1}{2} \mathbf{1} - (\mathbf{A}_{3,1} \boldsymbol{\xi}_1 + \mathbf{A}_{3,2} \boldsymbol{\xi}_2 + \mathbf{A}_{3,0} \xi_0). \quad (2.28b)$$

Using (2.28a) and (2.28b), it can be readily shown that the equality constraints in (2.21a)-(2.21c) can be rewritten to achieve (2.20a) and (2.20c). Thus, all that remains is to show  $\| [\boldsymbol{\xi}_1^T \ \boldsymbol{\xi}_2^T \ \xi_0 \ \boldsymbol{\xi}_s^T]^T \|_\infty \leq 1$ . Since  $0 \leq \lambda \leq 1$  holds,

$$\|\boldsymbol{\xi}'_1\|_\infty \geq \|\boldsymbol{\xi}'_1 \lambda\|_\infty = \|\boldsymbol{\xi}_1\|_\infty,$$

is satisfied. By (2.21b),  $\|\boldsymbol{\xi}'_1\|_\infty \leq 1$  implies  $\|\boldsymbol{\xi}_1\|_\infty \leq 1$ . Similarly, it can be shown that  $\|\boldsymbol{\xi}_2\|_\infty \leq 1$ . Using the definition of  $\xi_0$  from (2.28a) and  $0 \leq \lambda \leq 1$  proves that  $|\xi_0| \leq 1$ . Next, to show that  $\|\boldsymbol{\xi}_s\|_\infty \leq 1$ , consider rearranging (2.26) as,

$$\boldsymbol{\xi}_{s,1} = -\mathbf{1} + \left( \mathbf{1} - \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \boldsymbol{\xi}'_1 \right) \lambda. \quad (2.29)$$

Since,  $0 \leq \lambda \leq 1$  and  $\|\boldsymbol{\xi}'_1\|_\infty \leq 1$ , (2.29) results in  $\|\boldsymbol{\xi}_{s,1}\|_\infty \leq 1$ . Finally, using the last two rows of  $\mathbf{A}_{3,1}$ ,  $\mathbf{A}_{3,2}$ ,  $\mathbf{A}_{3,0}$ , and  $\mathbf{I}$  from the definitions of  $\mathbf{A}_h$  and  $\mathbf{b}_h$ , and following similar arguments, it can be shown that  $\|\boldsymbol{\xi}_{s,2}\|_\infty \leq 1$ . Thus,  $\forall \mathbf{x} \in CH(Z_1 \cup Z_2)$ ,  $\mathbf{x} \in Z_h$ .  $\square$

The resulting constrained zonotope  $Z_h$  obtained using **Theorem 5** has  $n_{gh} = 3(n_{g1} + n_{g2}) + 1$  generators and  $n_{ch} = n_{c1} + n_{c2} + 2(n_{g1} + n_{g2})$  constraints.

**Example 5.** For the zonotopes

$$Z_1 = \left\{ \left[ \begin{array}{ccc} 0 & 1 & 0 \\ 1 & 1 & 2 \end{array} \right], \mathbf{0} \right\}, Z_2 = \left\{ \left[ \begin{array}{ccc} -0.5 & 1 & -2 \\ 0.5 & 0.5 & 1.5 \end{array} \right], \left[ \begin{array}{c} -5 \\ 0 \end{array} \right] \right\},$$

Fig. 2.6 shows the convex hull  $Z_h = CH(Z_1 \cup Z_2)$  with  $n_g = 19$  generators and  $n_c = 12$  constraints, as computed using **Theorem 5**. Fig. 2.6 also shows the convex hull  $Z_{ch} = CH(Z_{c1} \cup Z_{c2})$  with  $n_g = 25$  generators and  $n_c = 18$  constraints, where  $Z_{c1} = Z_1 \cap H_{1-}$ ,  $Z_{c2} = Z_2 \cap H_{2-}$ ,  $H_{1-} = \{\mathbf{z} \mid [1 \ 1]\mathbf{z} \leq 0\}$ , and  $H_{2-} = \{\mathbf{z} \mid [-2.5 \ 1]\mathbf{z} \leq 9.5\}$ .



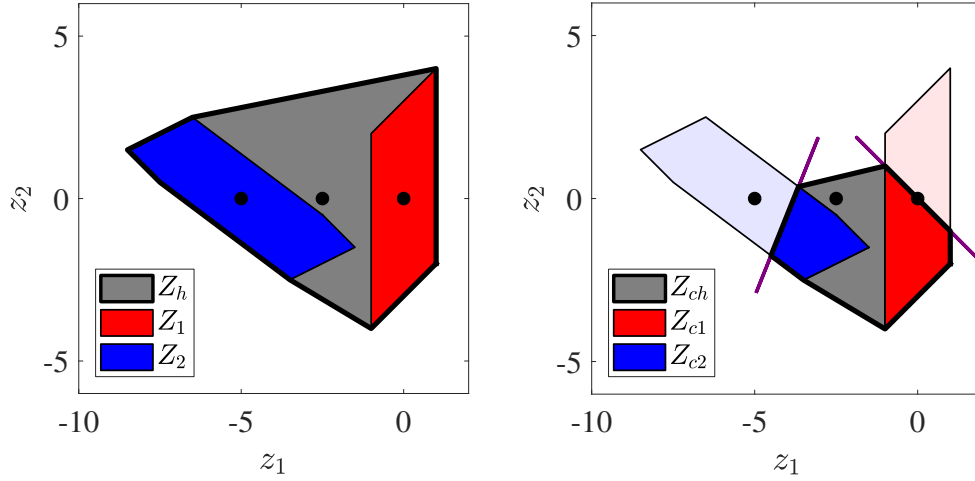


Figure 2.6. Left: The convex hull  $Z_h$  of zonotopes  $Z_1$  and  $Z_2$ . Right: The convex hull  $Z_{ch}$  of constrained zonotopes  $Z_{c1}$  and  $Z_{c2}$ , where each constrained zonotope is a zonotope-halfspace intersection corresponding to the shown hyperplanes.

## 2.8 Robust Positively Invariant (RPI) sets

This section provides both iterative and one-step optimization based methods for computing approximations of the minimal robust positively invariant set using zonotopes. Consider the autonomous discrete-time linear time-invariant system

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k, \quad (2.30)$$

where  $\mathbf{x}_k \in \mathbb{R}^n$ ,  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is a strictly stable matrix, and  $\mathbf{w}_k \in W \subset \mathbb{R}^n$ , where  $W$  is a convex and compact set containing the origin.

**Definition 3.** [13] *The set  $\Omega \subset \mathbb{R}^n$  is a robust positively invariant (RPI) set of (2.30) if and only if  $\mathbf{A}\Omega \oplus W \subseteq \Omega$ .*

**Definition 4.** [68] *The minimal RPI (mRPI) set  $F_\infty$  of (2.30) is the RPI set that is contained in every closed RPI set of (2.30) and is given by*

$$F_\infty = \bigoplus_{i=0}^{\infty} \mathbf{A}^i W. \quad (2.31)$$

### 2.8.1 Iterative Method

Unless specific conditions are met, such as  $\mathbf{A}$  being nilpotent, the infinite sequence of Minkowski sums in (2.31) makes it impossible to compute  $F_\infty$  exactly. Thus, outer- approximations of the mRPI set are typically used. An iterative approach is developed in [68] that computes the RPI set  $F(\alpha, s)$  such that  $F_\infty \subseteq F(\alpha, s) \subseteq F_\infty \oplus \epsilon B_\infty$ , where  $\epsilon$  is a user defined bound on the error of the approximation with  $s \in \mathbb{N}_+$ ,  $\alpha \in [0, 1)$  such that  $\mathbf{A}^s W \subseteq \alpha W$ . Starting at  $s = 0$ , the approach increments  $s$  until the approximation error is less than  $\epsilon$ , at which point  $F_s$  is computed as

$$F_s = \bigoplus_{i=0}^s \mathbf{A}^i W, \quad (2.32)$$

and  $F(\alpha, s) = (1 - \alpha)^{-1} F_s$ . The iterative algorithm in [68] requires use of multiple support functions at each iteration. When  $W$  is expressed in H-Rep, an LP must be solved for each support function calculation. As discussed in [84], computing  $F(\alpha, s)$  using this method may require the solution of thousands of LPs, even for a system with only two states. As briefly mentioned in Remark 3 in [68], if  $W$  is expressed in G-Rep, then the support function can be evaluated algebraically without the use of an LP, significantly reducing the computational cost. Thus, the use of zonotopes for RPI set calculations provides both improved scalability and reduced computational cost for the Minkowski sums in (2.32) and by removing the need to solve LPs.

### 2.8.2 One-step Optimization Method

As an alternative for the iterative method in [68], a one-step method for computing an outer- approximation of the mRPI set is presented in [84]. By expressing the RPI set in H-Rep, this method requires solving a single LP, assuming both the number and normal vectors of the hyperplanes associated with each halfspace inequality are provided *a priori*. Inspired by this approach, the following presents a similar one-step method for computing an outer- approximation of the mRPI set using G-Rep, where the generator vectors are predetermined.

**Theorem 6.** *The zonotope  $Z = \{\mathbf{G}\Phi, \mathbf{c}\} \subset \mathbb{R}^n$ , with  $\Phi = \text{diag}(\phi)$ ,  $\phi_i > 0, \forall i \in \{1, \dots, n_g\}$ , is an RPI set of (2.30) if  $W = \{\mathbf{G}_w, \mathbf{c}_w\}$  and there exists  $\Gamma_1 \in \mathbb{R}^{n_g \times n_g}$ ,  $\Gamma_2 \in \mathbb{R}^{n_g \times n_w}$ , and  $\beta \in \mathbb{R}^{n_g}$  such that*

$$\mathbf{A}\mathbf{G}\Phi = \mathbf{G}\Gamma_1, \quad (2.33a)$$

$$\mathbf{G}_w = \mathbf{G}\Gamma_2, \quad (2.33b)$$

$$(\mathbf{I} - \mathbf{A})\mathbf{c} - \mathbf{c}_w = \mathbf{G}\beta, \quad (2.33c)$$

$$|\Gamma_1|\mathbf{1} + |\Gamma_2|\mathbf{1} + |\beta| \leq \Phi\mathbf{1}. \quad (2.33d)$$

*Proof.* The proof requires showing that (2.33) enforces the zonotope containment conditions from **Lemma 3** such that  $X \subseteq Y$ , where  $X = \mathbf{A}Z \oplus W$  and  $Y = Z$ . Consider the change of variables  $\Gamma_1 = \Phi\tilde{\Gamma}_1$ ,  $\Gamma_2 = \Phi\tilde{\Gamma}_2$ ,  $\beta = \Phi\tilde{\beta}$  and define  $\tilde{\Gamma} = [\tilde{\Gamma}_1 \ \tilde{\Gamma}_2]$ . Then the zonotope containment conditions from (2.15) are satisfied by 1) rearranging and combining (2.33a) and (2.33b) to get  $[\mathbf{A}\mathbf{G}\Phi \ \mathbf{G}_w] = \mathbf{G}\Phi\tilde{\Gamma}$ , 2) rearranging (2.33c) to get  $\mathbf{c} - (\mathbf{A}\mathbf{c} + \mathbf{c}_w) = \mathbf{G}\Phi\tilde{\beta}$ , and 3) multiplying (2.33d) by  $\Phi^{-1}$ , since  $\phi_i > 0$ , to get  $|\tilde{\Gamma}|\mathbf{1} + |\tilde{\beta}| \leq \mathbf{1}$ .  $\square$

When using **Theorem 6** to determine the RPI set  $Z$  in G-Rep, the generator matrix  $\mathbf{G}$  is assumed to be known *a priori* in the same way that the normal vectors are chosen *a priori* in [84] for the one-step RPI set computation in H-Rep. Given a desired order of  $Z$ ,  $\mathbf{G}$  can be computed using (2.32) where  $\mathbf{G} = [\mathbf{G}_w \ \mathbf{A}\mathbf{G}_w \ \dots \ \mathbf{A}^s\mathbf{G}_w]$ , for some  $s \in \mathbb{N}_+$  that provides the desired order. Once  $\mathbf{G}$  is determined, the diagonal matrix  $\Phi$  provides the ability to scale the size of  $Z$  such that  $Z$  is an RPI set. Since the minimal RPI set is typically desired, an optimization problem can be formulated with the constraints from (2.33) and a objective function that minimizes the scaling variables in  $\Phi$ . With  $\mathbf{c}$ ,  $\Phi$ ,  $\Gamma_1$ ,  $\Gamma_2$ , and  $\beta$  as decision variables in this optimization problem, (2.33) consists of only linear constraints and thus an LP or QP can be formulated based on the norm used to minimize the vector  $\phi$ , where  $\Phi = \text{diag}(\phi)$ . In the following example, an LP is formulated by minimizing  $\|\phi\|_\infty$

subject to (2.33). Computing RPI set  $Z$  using **Theorem 6** requires solving an LP with  $n_g^2 + n_g(n_w + 2) + n$  decision variables.

**Example 6.** Consider the system from [84]

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u_k + \mathbf{w}_k, \quad (2.34)$$

with  $\mathbf{w}_k \in W = \{\mathbf{w} \in \mathbb{R}^2 \mid \|\mathbf{w}\|_\infty \leq 0.1\}$ . As in [84], the state feedback control law  $u_k = \mathbf{K}\mathbf{x}_k$ , where  $\mathbf{K}$  corresponds to the LQR solution with  $\mathbf{Q} = \mathbf{I}$  and  $\mathbf{R} = 1$ , converts (2.34) to an autonomous system of the form (2.30). For this system, four methods for computing outer-approximations of the mRPI set are compared in Fig. 2.7 with respect to volume ratio  $V_r$  and computation time  $\Delta t_{\text{calc}}$  as a function of set complexity ( $n_g$  for zonotopes in G-Rep,  $\frac{1}{2}n_h$  for polytopes in H-Rep). The seminal work from [68], denoted as  $\epsilon$ -mRPI (H-Rep), is the most computationally expensive since evaluating support functions for polytopes in H-Rep requires the solution of an LP. Using zonotopes in G-Rep, computational cost of this  $\epsilon$ -mRPI approach can be reduced by an order-of-magnitude since evaluating support functions for zonotopes is algebraic, as mentioned in Remark 3 of [68]. Alternatively, the 1-step approaches from [84] and **Theorem 6**, provide similar computational advantages. However, the 1-step approach from [84] is sensitive to the choice of hyperplanes. Using the same choice of hyperplanes from [84], Fig. 2.7 shows that the volume ratio does not decrease with increasing set complexity as quickly as the zonotope-based approach. Note that volume ratio is defined with respect to an approximation of the true mRPI set volume computed using the  $\epsilon$ -mRPI method with  $\epsilon = 10^{-9}$ .

To assess the scalability of these methods with respect to system order, Fig. 2.8 shows a comparison of these methods based on set complexity and computation time as a function of system order  $n$ . Note that the  $\epsilon$ -mRPI (H-Rep) method became impractical for higher system orders and is not included in Fig. 2.8. Similarly, the 1-step (H-Rep) method became

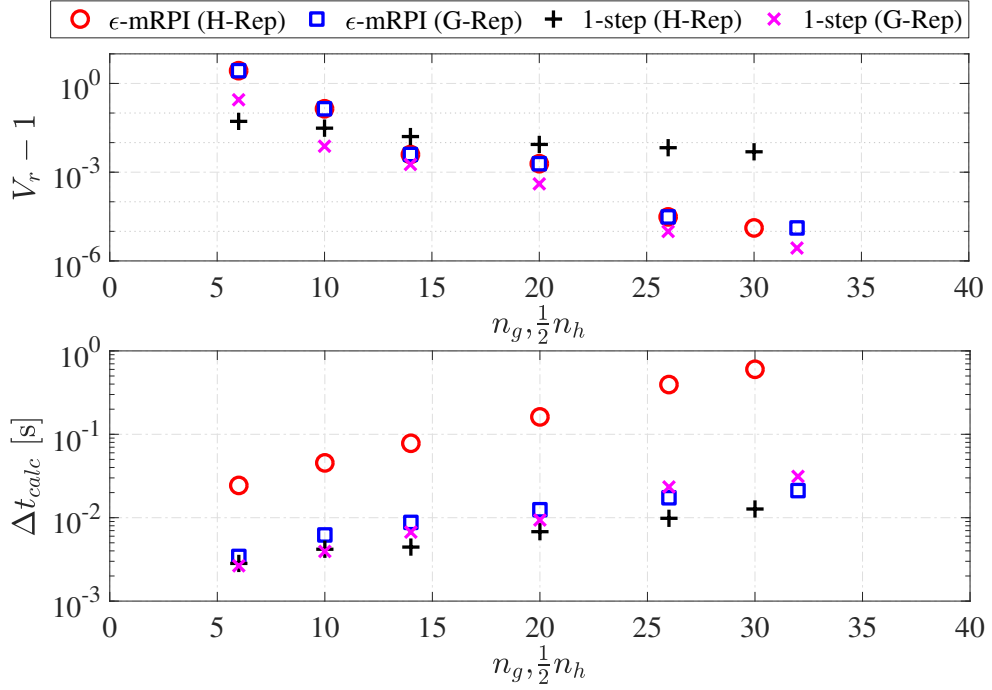


Figure 2.7. Comparison of volume ratio and computation time as a function of set complexity for outer-approximations of the mRPI set using iterative and 1-step approaches based on H-Rep or G-Rep.

impractical for  $n > 6$ . These results are generated using a  $n^{\text{th}}$ -order integrator system similar to that of (2.34). While the  $\epsilon$ -mRPI method in G-Rep provides the lowest computational cost, the complexity of the resulting set is roughly ten times larger than the set used for the 1-step approach. While scaling better than the 1-step H-Rep approach, the 1-step G-Rep approach requires solving a linear program with the constraints from (2.33) which includes the large decision variable  $\mathbf{\Gamma}_1 \in \mathbb{R}^{n_g \times n_g}$ . To manage this computational cost for higher order systems, the number of steps  $s \in \mathbb{N}_+$  in (2.32) can be chosen to balance set complexity and accuracy.

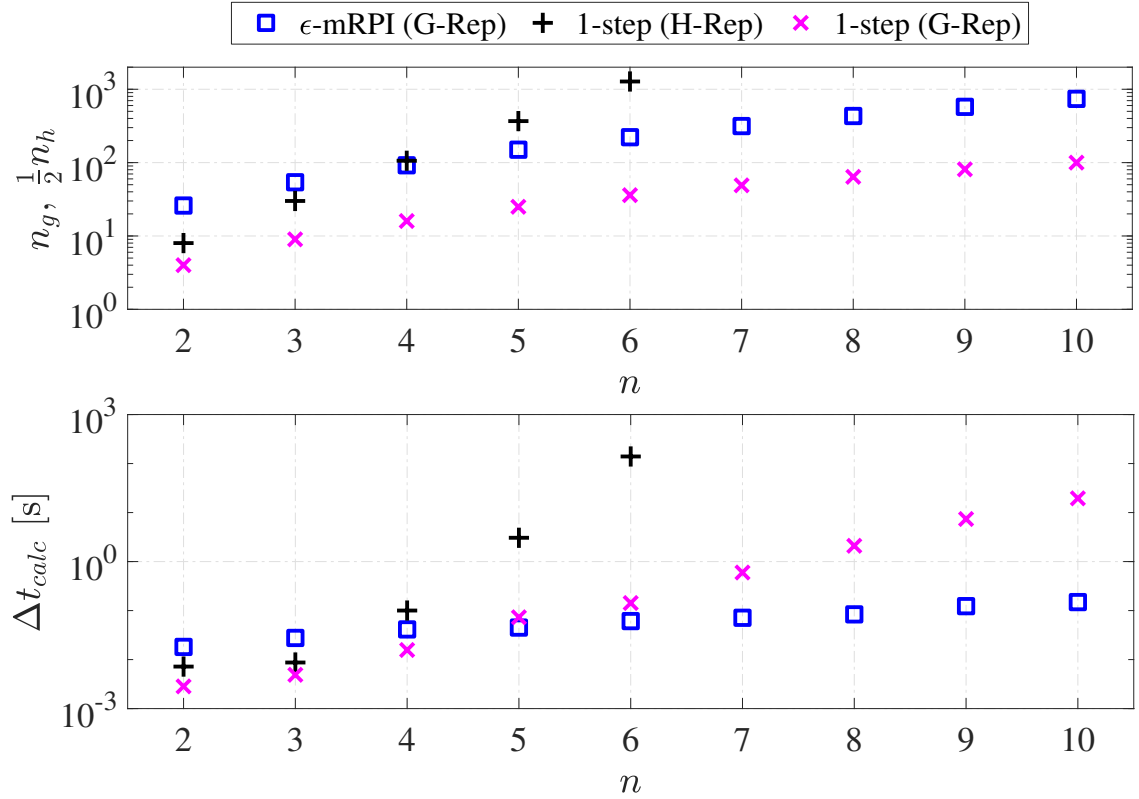


Figure 2.8. Comparison of set complexity and computation time as a function of system order for outer-approximations of the mRPI set using iterative and 1-step approaches based on H-Rep or G-Rep.

## 2.9 Pontryagin Difference

This subsection provides an iterative method for computing the constrained zonotope representation of the Pontryagin difference of two zonotopes and a one-step optimization method for computing the zonotopic inner-approximation of the Pontryagin difference.

**Definition 5.** [3] *Given two sets  $Z_1, Z_2 \subset \mathbb{R}^n$ , the Pontryagin difference  $Z_d = Z_1 \ominus Z_2$  is defined as*

$$Z_d = \{z \in \mathbb{R}^n \mid z \oplus Z_2 \subseteq Z_1\}. \quad (2.35)$$

The Pontryagin difference is also referred to as the Minkowski difference or the erosion of set  $Z_1$  by  $Z_2$ .

### 2.9.1 Iterative Method

If  $Z_1$  and  $Z_2$  are zonotopes, then [3] provides the following iterative method for computing  $Z_d$ .

**Lemma 5.** (Theorem 1 of [3]) *If  $Z_1 = \{\mathbf{G}_1, \mathbf{c}_1\}$  and  $Z_2 = \{\mathbf{G}_2, \mathbf{c}_2\}$ , then the Pontryagin difference  $Z_d = Z_1 \ominus Z_2$  is computed using the  $n_{g2}$  generators  $\mathbf{g}_{2,i}$  of  $Z_2$  by applying the following recursion:*

$$Z_{int}^{(0)} = Z_1 - \mathbf{c}_2, \quad (2.36a)$$

$$Z_{int}^{(i)} = (Z_{int}^{(i-1)} + \mathbf{g}_{2,i}) \cap (Z_{int}^{(i-1)} - \mathbf{g}_{2,i}), \quad (2.36b)$$

$$Z_d = Z_{int}^{(n_{g2})}. \quad (2.36c)$$

As shown in [3], zonotopes are not closed under the Pontryagin difference. Thus, the methods in [3] require the use of a combination of G-Rep and H-Rep to compute approximations of  $Z_d$  in G-Rep. While this combination results in faster calculations than methods that solely use H-Rep, the majority of computation time comes from the conversion from G-Rep to H-Rep, which scales exponentially with the number of generators.

However, since  $Z_d$  is computed via the intersection of zonotopes,  $Z_d$  can be exactly represented as a constrained zonotope. Thus, (2.36b) can be directly computed using the generalized intersection from (2.3) without the need for H-Rep. Note that iterative method from **Lemma 5** is also applicable if  $Z_1 = \{\mathbf{G}_1, \mathbf{c}_1, \mathbf{A}_1, \mathbf{b}_1\}$  is a constrained zonotope, since (2.36) only requires  $Z_2$  to be the Minkowski sum of generators  $\mathbf{g}_{2,i}$ . For a constrained zonotope  $Z_1$  in  $\mathbb{R}^n$  with  $n_{c1}$  constraints and  $n_{g1}$  generators and a zonotope  $Z_2$  in  $\mathbb{R}^n$  with  $n_{g2}$  generators,  $Z_d = Z_1 \ominus Z_2$  is a constrained zonotope with  $n_{gd} = 2^{n_{g2}}n_{g1}$  generators and  $n_{cd} = 2^{n_{g2}}n_{c1} + n(2^{n_{g2}} - 1)$  constraints.

### 2.9.2 One-step Optimization Inner-Approximation Method

As an alternative to the iterative method from **Lemma 5**, the following theorem presents a one-step method for computing an zonotopic inner-approximation of the Pontryagin difference  $\tilde{Z}_d \subseteq Z_d = Z_1 \ominus Z_2$  using a single LP.

**Theorem 7.** *Given  $Z_1 = \{\mathbf{G}_1, \mathbf{c}_1\}$  and  $Z_2 = \{\mathbf{G}_2, \mathbf{c}_2\}$ , then  $\tilde{Z}_d = \{[\mathbf{G}_1 \ \mathbf{G}_2]\Phi, \mathbf{c}_d\}$ , with  $\Phi = \text{diag}(\phi), \phi_i > 0, \forall i \in \{1, \dots, n_{g1} + n_{g2}\}$ , is an inner-approximation of the Pontryagin difference such that  $\tilde{Z}_d \subseteq Z_1 \ominus Z_2$  if there exists  $\Gamma \in \mathbb{R}^{n_{g1} \times (n_{g1} + 2n_{g2})}$  and  $\beta \in \mathbb{R}^{n_{g1}}$ , such that*

$$\begin{bmatrix} [\mathbf{G}_1 \ \mathbf{G}_2]\Phi & \mathbf{G}_2 \end{bmatrix} = \mathbf{G}_1\Gamma, \quad (2.37a)$$

$$\mathbf{c}_1 - (\mathbf{c}_d + \mathbf{c}_2) = \mathbf{G}_1\beta, \quad (2.37b)$$

$$|\Gamma|\mathbf{1} + |\beta| \leq \mathbf{1}. \quad (2.37c)$$

*Proof.* By viewing (2.37) in the context of the zonotope containment conditions from

**Lemma 3**, it is clear that (2.37) enforces the Pontryagin difference condition  $\tilde{Z}_d \oplus Z_2 \subset Z_1$  from (2.35).  $\square$

When using **Theorem 7** to compute  $\tilde{Z}_d \subset Z_d$  in G-Rep, the generator matrix  $[\mathbf{G}_1 \ \mathbf{G}_2]\Phi$  is assumed to be comprised of the generators from both  $Z_1$  and  $Z_2$  scaled by the diagonal matrix  $\Phi$ . Since maximizing the size of  $\tilde{Z}_d$  is typically desired, an optimization problem can be formulated with the constraints from (2.37) and an objective function that maximizes the scaling variables in  $\Phi$ . With  $\mathbf{c}_d$ ,  $\Phi$ ,  $\Gamma$ , and  $\beta$  as decision variables in this optimization problem, (2.37) consists of only linear constraints and thus an LP or QP can be formulated based on the norm used to maximize the vector  $\phi$ , where  $\Phi = \text{diag}(\phi)$ . Computing  $\tilde{Z}_d$  using **Theorem 7** requires solving a LP with  $n_{g1}^2 + 2n_{g1}n_{g2} + 2n_{g1} + n_{g2} + n$  decision variables.



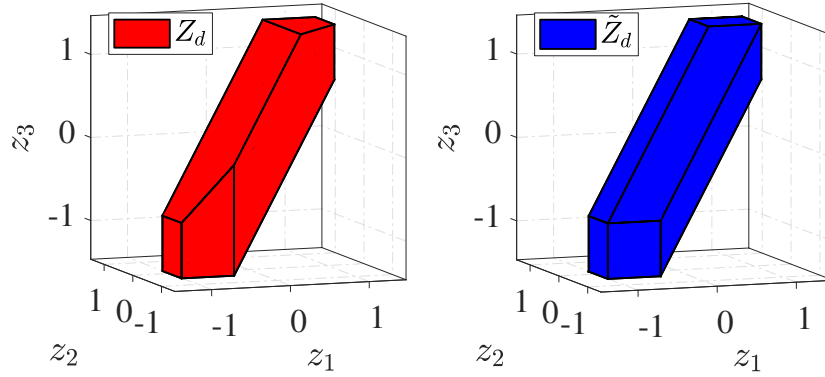


Figure 2.9. Left: The Pontryagin difference  $Z_d = Z_1 \ominus Z_2$  where  $Z_1$  and  $Z_2$  are zonotopes but  $Z_d$  is not [3]. Right: The inner-approximation of  $Z_d$  by a zonotope  $\tilde{Z}_d \subseteq Z_d$ .

**Example 7.** Consider the zonotopes from [3]

$$Z_1 = \left\{ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \mathbf{0} \right\}, \quad Z_2 = \left\{ \frac{1}{3} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \mathbf{0} \right\}.$$

Fig. 2.9 shows the Pontryagin difference  $Z_d = Z_1 \ominus Z_2$  with  $n_g = 64$  and  $n_c = 45$  computed using **Lemma 5**. As discussed in [3], zonotopes are not closed under the Pontryagin difference, which can be seen in Fig. 2.9 by the asymmetric facets of  $Z_d$ . Using **Theorem 7**, the inner-approximation of the Pontryagin difference  $\tilde{Z}_d$  is also shown in Fig. 2.9. Choosing to maximize  $\|[\mathbf{G}_1 \ \mathbf{G}_2]\Phi\|_\infty$  subject to (2.37) produced  $\tilde{Z}_d \subset Z_d$  with a volume ratio of  $V_r = 0.924$ .

**Example 8.** Similar to [3], the scalability of exact constrained zonotope representations of the Pontryagin difference via **Lemma 5** and zonotopic inner-approximations via **Theorem 7** is compared with the standard H-Rep approach provided in the Multi-Parametric Toolbox [37]. Table 2.1 shows the complexity and computational time for computing the Pontryagin difference  $Z_d = Z_1 \ominus Z_2$  using each of the three methods for zonotopes in  $\mathbb{R}^2$ ,

Table 2.1. Pontryagin difference set complexity and computation time (seconds)

		$Z_1 \ominus Z_2$							
	$Z_1$	$Z_2$	H-Rep		CG-Rep	1-Step (G-Rep)			
$n$	$n_g$		$n_h$	$t_h$	$n_c \times n_g$	$t_h/t_{cg}$	$n_g$	$V_r$	$t_h/t_g$
2	4	4	16	0.01	$30 \times 64$	33.2	2.5	0.64	3.3
2	8	4	32	0.01	$30 \times 128$	48.2	3.0	0.54	3.5
2	4	8	16	0.01	$510 \times 1024$	17.6	2.6	0.67	3.3
2	8	8	32	0.02	$510 \times 2048$	20.9	3.2	0.53	3.1
3	6	6	60	0.03	$189 \times 384$	68.8	3.7	0.55	7.5
3	12	6	264	0.16	$189 \times 768$	261	4.8	0.46	16.0
3	6	12	60	0.13	$12,285 \times 24,576$	18.6	3.8	0.54	21.5
3	12	12	264	0.40	$12,285 \times 49,152$	27.2	4.9	0.43	28.2
4	8	8	224	0.38	$1,020 \times 2,048$	359	4.9	0.50	46.2
4	16	8	2,240	41.0	$1,020 \times 4,096$	2,370	6.2	0.45	1,890
4	8	16	224	59.0	$262,140 \times 524,288$	271	4.7	0.43	4,078
4	16	16	2,240	243	$262,140 \times 1,048,576$	556	6.2	0.48	6,510

$\mathbb{R}^3$ , and  $\mathbb{R}^4$ . Each entry in Table 2.1 represents an average of 100 computations using randomly generated zonotopes  $Z_1$  and  $Z_2$ . These random zonotopes are generated using the procedure provided in [3] and the CORA toolbox [4]. Cases where  $Z_d = \emptyset$  were disregarded and not considered in the set of 100 computations. For CG-Rep and G-Rep, the ratio of computation times relative to that of H-Rep is presented. Since the G-Rep approach is an inner-approximation, the average volume ratio is also provided. From these results, it is clear that both the set complexity  $n_h$  and the computation time  $t_h$  for the H-Rep approach increase by approximately an order-of-magnitude as the set dimension  $n$  increases. While the CG-Rep approach increases the computation speed by approximately two orders-of-magnitude, the set complexity increases exponentially. Sparse matrices were used to reduce the memory requirements for these computations. The redundancy removal approach presented in Section was not able to detect the high-degree of redundancy in these set representations. Alternatively, the one-step G-Rep approximation approach also provided significant reductions in computational cost while maintaining a small number of generators. However, for these randomly

generated zonotopes, the inner-approximation only captures approximately 50% of the volume of  $Z_d$ . While these methods will likely work well for many practical applications, future work is needed to improve redundancy detection and removal for the CG-Rep approach and improved optimization formulations are needed for the G-Rep approach to further maximize volume ratio.

## 2.10 Application to Reachability Analysis

To demonstrate the applicability of algorithms developed in this chapter, this section considers the exact and approximate computations of backwards reachable sets of a constrained linear system in the context of the two-level hierarchical MPC framework developed in [48, 49]. The high-level goal is to compute a *wayset*  $Z_c(k)$  at discrete time step  $k$  that captures all of the initial states  $\mathbf{x}(k) \in Z_c(k) \subset \mathbb{R}^n$  for which there are state and input trajectories  $\mathbf{x}(k+j)$  and  $\mathbf{u}(k+j)$  that satisfy, for all  $j \in \{0, \dots, N-1\}$ , *i*) the dynamics  $\mathbf{x}(k+j+1) = \mathbf{A}\mathbf{x}(k+j) + \mathbf{B}\mathbf{u}(k+j)$ , *ii*) the state and input constraints  $\mathbf{x}(k+j) \in \mathcal{X}$  and  $\mathbf{u}(k+j) \in \mathcal{U}$ , and *iii*) the terminal constraint  $\mathbf{x}(k+N) = \mathbf{x}^*$  for some predetermined target  $\mathbf{x}^* \in \mathbb{R}^n$ . In the context of the hierarchical MPC framework from [48, 49],  $\mathbf{x}^*$  is a future state on the optimal trajectory determined by an upper-level controller and  $Z_c(k)$  is a terminal constraint imposed on a lower-level controller. Since  $\mathbf{x}^*$  is updated at every evaluation of the upper-level controller,  $Z_c(k)$  must be recomputed in real-time, which is enabled through the use of constrained zonotopes.

**Algorithm 2** shows a simplified version of the backward reachable wayset algorithms presented in [48, 49]. Fig. 2.10 shows the results of this algorithm when applied to the simplified vehicle system model from [48, 49] with discretization time step size  $\Delta t = 1$  second. The state and input constraints defining  $\mathcal{X}$  and  $\mathcal{U}$  are  $[-1 \ -20 \ 0]^T \leq \mathbf{x}(k) \leq [105 \ 20 \ 100]^T$  and  $[0 \ 0 \ 0]^T \leq \mathbf{u}(k) \leq [1 \ 1 \ 1]^T$ .

To demonstrate the halfspace intersection results from Section 2.3 and 2.4, Table 2.2 compares the set representation complexity and computation time of four different CG-Rep methods with those using H-Rep via the Multi-Parametric Toolbox [37]. All computation times are averaged over 100 runs. Overall, the CG-Rep methods result in significantly less set complexity and computation time. The CG-Rep methods differ in the computation of  $\hat{Z}_c(k+j-1) \cap \mathcal{X}$  in **Algorithm 2**. Specifically, this intersection is computed using 1) the zonotope-hyperplane (ZH) method from **Lemma 1** based on the parent zonotope  $\hat{Z}(k+j-1) \supset \hat{Z}_c(k+j-1)$  and the H-Rep of  $\mathcal{X}$ , 2) the generalized intersection (GI) (from (2.3)) of the constrained zonotope wayset and the G-Rep of  $\mathcal{X}$ , 3) the linear program (LP) method from **Lemma 2** for checking the intersection of a constrained zonotope and a hyperplane, and 4) the interval arithmetic (IA) approach using **Algorithm 1** to detect empty sets when  $\hat{Z}_c(k+j-1) \subset \mathcal{X}$ . In the ZH, LP, and IA methods, if the wayset intersects the hyperplanes associated with the halfspaces of  $\mathcal{X}$ , generators and constraints are added using (2.11) to exactly compute  $\hat{Z}_c(k+j-1) \cap \mathcal{X}$  in CG-Rep.

As expected, the GI approach resulted in the highest set complexity since generators and constraints are added even if  $\hat{Z}_c(k+j-1) \subset \mathcal{X}$ . The LP approach results in the lowest complexity by only adding generators and constraints when needed to exactly define the intersection. In this application, the ZH method also achieves this low set complexity and requires significantly less computation time. However, achieving this low complexity is not expected in general. Finally, the IA approach did not perform as well in this application, resulting in unnecessary generators and constraints and a large computation time. However, in practice, the zonotope-halfspace check from **Theorem 1** would be applied first so that **Algorithm 1** is only used in cases where the parent zonotope intersects the hyperplane.

To demonstrate redundancy removal results from Section 2.5, **Algorithm 2** and **Theorem 2** were applied to successfully remove all unnecessary generators and constraints resulting in the irredundant constrained zonotope wayset  $\tilde{Z}_c$  in Table 2.2. Overall, when

---

**Algorithm 2:** Wayset  $Z_c(k)$  for target  $\mathbf{x}^*$ .

---

**Input** :  $\mathbf{x}^*$   
**Output:**  $Z_c(k)$

- 1 initialize  $j \leftarrow N$
- 2  $Z_c(k + j) = \mathbf{x}^*$ ;
- 3 **while**  $j \geq 1$  **do**
- 4      $\hat{Z}_c(k + j - 1) = A^{-1}Z_c(k + j) \oplus (-A^{-1}B)\mathcal{U}$ ;
- 5      $Z_c(k + j - 1) = \hat{Z}_c(k + j - 1) \cap \mathcal{X}$ ;
- 6      $j \leftarrow j - 1$ ;
- 7 **end**
- 8  $Z_c(k) = Z_c(k + j)$

---

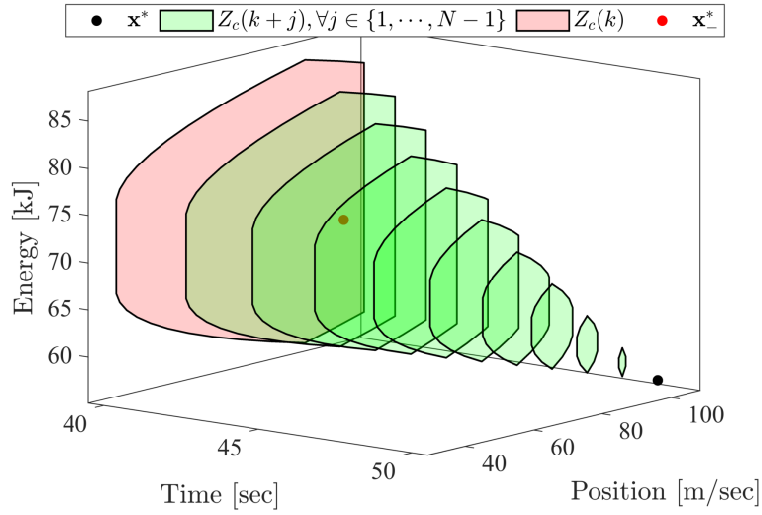


Figure 2.10. The evolution of backward reachable wayset  $Z_c(k)$  for  $k = 40$  and  $N = 10$  time steps starting from  $\mathbf{x}^*$  projected on the position and energy states. The sets  $Z_c(k + j), \forall j \in \{7, 8, 9\}$  are zonotopes (evident from symmetry) while the sets  $Z_c(k + j), \forall j \in \{0, \dots, 6\}$ , are constrained zonotopes. The constrained zonotope wayset  $Z_c(k)$  contains  $\mathbf{x}_*$  ensuring the control feasibility from [48, 49].

compared to H-Rep, any of the four CG-Rep approaches are computationally efficient with less set complexity and the preferred CG-Rep approach is likely to be application dependent.

When computing these waysets for complex systems, it is likely that inner-approximations are needed to restrict the complexity of the set to satisfy a predetermined upper bound on the number of generators and constraints. Demonstrating the inner-approximations from

Table 2.2. Complexity and Computation Time of Waysets

Method	$Z_c$	$t_{calc}$	$\tilde{Z}_c$	$t_{calc}$
	$n_c \times n_g$	sec	$\tilde{n}_c \times \tilde{n}_g$	sec
ZH	$7 \times 37$	$1e^{-3}$	$7 \times 37$	$4e^{-3}$
GI	$30 \times 60$	$2e^{-3}$	$7 \times 37$	$2e^{-1}$
LP	$7 \times 37$	$1e^{-1}$	$7 \times 37$	$2e^{-3}$
IA	$15 \times 45$	$1e^{-1}$	$7 \times 37$	$4e^{-2}$
H-Rep	$n_h = 5047$	161	$n_h = 153$	333

Section 2.6.3 and the convex hull operation from Section 2.7, the top row of plots in Fig. 2.11 shows the inner-approximating interval set  $B \subset Z_c$  computed using the method described in **Example 4** with  $n_g = 3$  and  $n_c = 0$ . However, in the hierarchical MPC framework from [48, 49] the wayset must also include a key element denoted here as  $\mathbf{x}_-^*$ . Since  $\mathbf{x}_-^* \notin B$ , the wayset can be computed as  $CH(B \cup \mathbf{x}_-^*)$  resulting in  $n_g = 10$  and  $n_c = 6$ . If this increase in set complexity is undesirable for a particular application, the point containment  $\mathbf{x}_-^* \in B \subseteq Z_c$  can be readily added to the LP defined in (2.19). The resulting inner-approximating interval set with this point containment is shown in the bottom row of plots in Fig. 2.11. The computation time for these inner-approximating interval sets are approximately 0.18 and 0.25 seconds for the top and bottom rows, respectively.

## 2.11 Conclusions

The use of zonotopes and constrained zonotopes for set operations provides significant computational advantages that improve the practicality of set-based techniques commonly used in systems and control theory. Operations such as halfspace intersections, convex hulls, invariant sets, and Pontryagin differences have been shown to benefit from zonotope and constrained zonotope set representations. Complexity reduction techniques were developed based on redundancy removal and inner-approximations to further improve the practical-

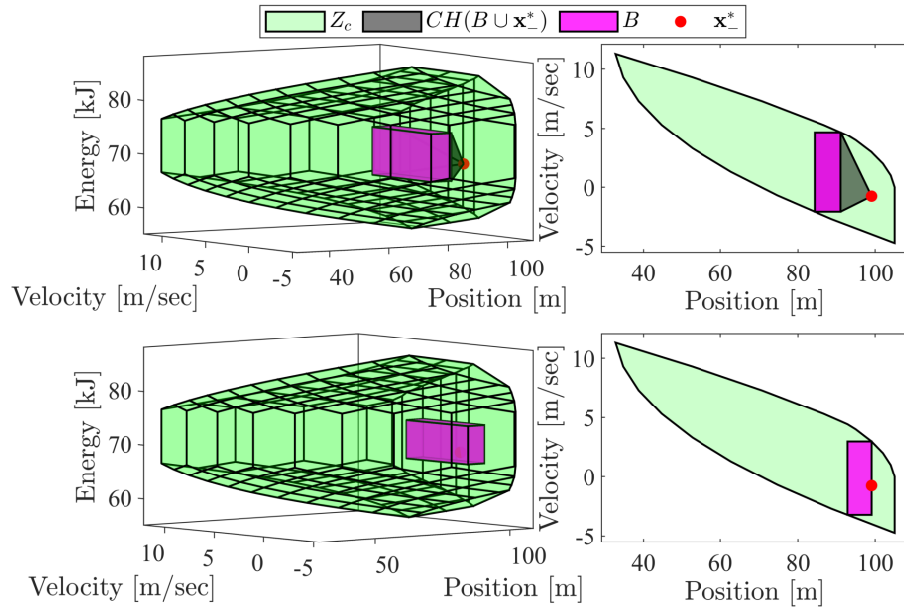


Figure 2.11. Top: The wayset  $Z_c$ , inner-approximating interval set  $B$  with  $V_r = 0.35$ , and  $CH(B \cup \mathbf{x}_-^*)$  with  $V_r = 0.39$  are shown on the left and the projections on to the position and velocity states are shown on the right. Bottom: The wayset  $Z_c$ , inner-approximating interval set  $B$  containing  $\mathbf{x}_-^*$  with  $V_r = 0.30$  shown on the left with the projection shown on the right.

ity of these set representations. Future work will focus on improved redundancy detection algorithms and optimization formulations that more accurately capture the volume of the approximated set. The following chapter will discuss about the development of a vertical hierarchical model predictive control algorithm for systems with additive disturbances.

### Supplementary Material

The source code for all the zonotope and constrained zonotope operations and numerical examples is provided at <https://github.com/ESCL-at-UTD/ConZono>.

# CHAPTER 3

## VERTICAL HIERARCHICAL MODEL PREDICTIVE CONTROL WITH ADDITIVE DISTURBANCES<sup>1</sup>

### 3.1 Introduction

For the control of many complex systems, the ability to satisfy both input and state constraints is critical to maintaining safe and reliable system operation. Additionally, with increasing demand for performance and efficiency, optimal system operation is characterized by both transient and steady-state input and state trajectories that approach these constraints. Examples include the control of aircraft power systems [80, 20], on- and off-road hybrid vehicles [72, 41, 88], smart grids [42, 33, 93], and water distribution networks [63, 64].

For the control of input and state constrained systems, system operation is not always indefinite and the desired behavior is not always characterized by driving the system to steady-state or from one equilibrium to another. This idea is discussed in [70] for vehicle maneuvering problems where the notion of stability is replaced by the notion of completion. Similarly, this work focuses on the control of systems under finite operation, with the goal of guaranteeing state and input constraints during operation and terminal state constraints at the end of operation.

Model Predictive Control (MPC) is well-suited for the control of constrained systems since input and state constraints are directly imposed in the underlying optimization problem. Feasibility of these constraints and stability of the closed-loop system are well understood for the case of a single centralized controller [62]. However, for systems that require fast control update rates and long prediction horizons, the time required to solve the resulting large optimization problem may prevent real-time implementation.

---

<sup>1</sup>This chapter is based on work supported by the United States Air Force Research Laboratory.

Chapter 3 in part is a reprint of material published in: © Automatica. Reprinted, with permission, from Koeln, J., Raghuraman, V. and Hancey, B., 2020. Vertical hierarchical MPC for constrained linear systems. *Automatica*, 113, p.108817.



Alternatively, hierarchical MPC can be used to decompose control decisions across multiple levels of controllers [73]. Upper-level controllers use large time steps to achieve long prediction horizons with fewer discrete steps. Lower-level controllers with small time steps use short prediction horizons to minimize computational cost and enable real-time implementation. To handle the timescale separation between the system and actuator dynamics, several two-level hierarchical MPC formulations have been developed [25, 26, 75, 74, 8, 9, 86].

However, existing hierarchical formulations are not well suited to maximize the performance of a system subject to input, state, and terminal constraints under finite operation. Most hierarchical MPC approaches are formulated with the goal of stabilizing the system to an equilibrium in the interior of state and input constraint sets. However, for systems with finite operation, such equilibrium might not exist as in the case of systems whose operation is based on the consumption of a finite resource (e.g. fuel in an aircraft [80, 20] or battery state of charge in an electric vehicle [72, 41, 88]). Moreover, existing approaches [25, 26, 75, 74, 9, 8, 86] are typically formulated where upper-level controllers are robust to the control decisions of lower-level controllers and overall control authority is divided among each control level. Existing reference tracking based coordination mechanisms require lower-level controllers to track state and input trajectories determined by upper-level controllers, preventing the hierarchical controller from utilizing the fast system dynamics to maximize system performance. Finally, while most hierarchical MPC formulations are designed to two controller levels, many systems have more than two timescales and an  $M$ -level hierarchical MPC would be more effective in controlling each timescale. To date, there does not exist a  $M$ -level hierarchical MPC framework that provides guaranteed state and input constraint satisfaction, even for linear systems.

To develop a constructive hierarchical MPC framework that guarantees input and state constraint satisfaction, this chapter focuses on a vertical hierarchy, with one controller per level, for discrete-time linear systems. This work replaces the conventional reference tracking

based coordination between controllers of the hierarchy with a novel coordination mechanism using *waysets*. A wayset defines a subset of states at a future point in time from which there exist feasible state and input trajectories for the remainder of system operation. Thus, driving the system states to a wayset provides a short-term control objective that guarantees long-term constraint satisfaction. Within the proposed hierarchical MPC framework, waysets are computed based on optimal state trajectories determined by upper-level controllers and imposed as terminal constraints for lower-level controllers. Wayset-based coordination overcomes the limitations of existing hierarchical MPC frameworks by removing the need for constraint-feasible equilibrium, removing the conservatism that stems from requiring upper-level controllers to be robust to the lower-level control decisions, and allowing lower-level controllers to utilize the fast system dynamics to further improve system performance.

Similar coordination mechanisms have been used in the literature. For wastewater treatment systems, the hierarchical controller in [16] uses “interlayer targets” to achieve coordination between controllers at different levels. These interlayer targets inspired the use of *waypoints* as the coordination mechanism for a two-level hierarchy in [47]. These waypoints are imposed as terminal constraints on lower-level controllers within the hierarchy. For vehicle path-planning, waypoint tracking control is used in [1] to split long planning horizons into multiple shorter horizons by creating intermediate goals. This idea was extended to wayset tracking in [81], where waysets represent a region of the state space instead of a single point. However, in both cases the waypoint/wayset generation is performed off-line in a feed-forward fashion.

To enable on-line calculation of waysets, computational efficiency is vastly improved by representing the waysets as constrained zonotopes [79]. Zonotopes are widely used due to their computational efficiency in reach set calculations for hybrid system verification, estimation, and MPC [59, 6, 79, 15]. As will be shown, the proposed wayset calculations are similar to the computation of reach sets and utilize linear transformation, Minkowski sum, and intersection operations.

*Contributions:* To achieve guaranteed input and state constraint satisfaction, this chapter develops a vertical hierarchical MPC framework with a novel wayset coordination mechanism. The specific contributions of this chapter are 1) the development of an  $M$ -level hierarchical MPC framework that incorporates known disturbances and is robust to bounded unknown disturbances, 2) the definition and use of waysets to prove robust closed-loop constraint satisfaction, 3) the representation and calculation of waysets as constrained zonotopes to achieve efficient on-line calculation, and 4) the numerical demonstration of performance and scalability of the hierarchical approach. Note that the nominal version of this hierarchical MPC formulation without accounting for disturbances was initially presented in [48].

The remainder of the section is organized as follows. Sections 3.3 and 3.4 present the class of constrained discrete-time linear systems and the proposed  $M$ -level hierarchical MPC formulation. Section 3.5 defines a robust output constraint tightening procedure and the wayset properties. Robust state and input constraint satisfaction is proved in Section 3.6. Section 3.7 details the calculation of waysets and the use of constrained zonotopes to achieve computational efficiency. Two numerical examples are provided in Section 3.8 to demonstrate the key features, performance, and scalability of the approach. Finally, Section 3.9 summarizes the conclusions of the chapter.

## 3.2 Notation and Preliminaries

For a discrete time system, the notation  $x(k)$  denotes the state  $x$  at time step  $k$ . For MPC, the double index notation  $x(k+j|k)$  denotes the predicted state at future time  $k+j$  determined at time step  $k$ . The bracket notation  $k \in [0, k_F]$  denotes all integer values of  $k$  from 0 to  $k_F$ . The state trajectory over these time indices is denoted  $\{x(k)\}_{k=0}^{k_F}$ . The set of positive integers is  $\mathbb{Z}_+$ . The weighted norm is defined as  $\|x\|_\Lambda^2 = x^T \Lambda x$ , where  $\Lambda$  is a positive definite diagonal matrix. The subscript  $i$  is used to denote the  $i^{\text{th}}$  controller in the hierarchy,  $\mathbf{C}_i$ , and  $i^- = i - 1$  is shorthand used to reference the controller directly above,  $\mathbf{C}_{i^-}$ . For

sets  $\mathcal{Z}, \mathcal{W} \subset \mathbb{R}^n$ ,  $\mathcal{Y} \subset \mathbb{R}^m$ , and matrix  $R \in \mathbb{R}^{m \times n}$ , the linear transformation of  $\mathcal{Z}$  under  $R$  is  $R\mathcal{Z} = \{Rz \mid z \in \mathcal{Z}\}$ , the Minkowski sum of  $\mathcal{Z}$  and  $\mathcal{W}$  is  $\mathcal{Z} \oplus \mathcal{W} = \{z + w \mid z \in \mathcal{Z}, w \in \mathcal{W}\}$ , and the generalized intersection of  $\mathcal{Z}$  and  $\mathcal{Y}$  under  $R$  is  $\mathcal{Z} \cap_R \mathcal{Y} = \{z \in \mathcal{Z} \mid Rz \in \mathcal{Y}\}$ . The standard intersection, corresponding to the identity matrix  $R = I_n$ , is simply denoted as  $\mathcal{Z} \cap \mathcal{W}$ . The Pontryagin difference is defined as  $\mathcal{Z} \ominus \mathcal{W} = \{z \in \mathbb{R}^n \mid z + w \in \mathcal{Z}, \forall w \in \mathcal{W}\}$ . The Cartesian product is defined as  $\mathcal{Z} \times \mathcal{Y} = \{[z^T \ y^T]^T \mid z \in \mathcal{Z}, y \in \mathcal{Y}\}$ . The projection of the set  $\mathcal{Y}$  on the first  $n$  dimensions is denoted  $\pi_n(\mathcal{Y})$ . The empty set is denoted as  $\emptyset$ .

### 3.3 Problem Formulation

Consider the discrete linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k) + d(k), \quad (3.1)$$

with states  $x \in \mathbb{R}^n$ , inputs  $u \in \mathbb{R}^m$ , disturbances  $d \in \mathbb{R}^n$ , and where  $A \in \mathbb{R}^{n \times n}$  is invertible,  $B \in \mathbb{R}^{n \times m}$ , and the pair  $(A, B)$  is stabilizable.

**Assumption 1.** *With a fixed time step size  $\Delta t$ , the system operates for a finite length of time starting from  $t = 0$  and ending at  $t = t_F = k_F \Delta t$  with discrete time steps indexed by  $k \in [0, k_F]$ .*

**Assumption 2.** *The disturbance is  $d(k) = \hat{d}(k) + \Delta d(k)$ , where  $\hat{d}(k)$  is known a priori for all  $k \in [0, k_F]$  and  $\Delta d(k)$  is unknown but bounded to a convex and compact set such that  $\Delta d(k) \in \mathcal{D} \subset \mathbb{R}^n$ .*

Starting from an initial condition  $x(0)$ , the goal is to develop a vertical hierarchical MPC approach that plans and executes an input trajectory  $\{u(k)\}_{k=0}^{k_F-1}$  and corresponding state trajectory  $\{x(k)\}_{k=0}^{k_F}$  which i) satisfies the system dynamics from (3.1); ii) satisfies the state and input constraints

$$x(k) \in \mathcal{X} \subset \mathbb{R}^n, \quad u(k) \in \mathcal{U} \subset \mathbb{R}^m, \quad \forall k \in [0, k_F - 1]; \quad (3.2)$$



**Assumption 4.** For each controller  $\mathbf{C}_i, i \in [1, M]$ , the prediction horizon  $N_i \in \mathbb{Z}_+$  and time step size  $\Delta t_i > 0$  satisfy

$$i) \Delta t_M = \Delta t;$$

$$ii) \Delta t_{i-} = N_i \Delta t_i;$$

$$iii) \Delta t_1 = \frac{t_F}{N_1}.$$

These assumptions indicate i) the lowest-level controller  $\mathbf{C}_M$  and the system (3.1) have the same time step size, ii) each controller  $\mathbf{C}_i$  predicts state and input trajectories between consecutive updates of the controller  $\mathbf{C}_{i-}$  directly above in the hierarchical controller, and iii) the highest-level controller  $\mathbf{C}_1$  predicts to the end of system operation. Additionally, let  $\nu_i \triangleq \frac{\Delta t_i}{\Delta t} \in \mathbb{Z}_+$ , be defined as a time scaling factor for each controller. The time steps for  $\mathbf{C}_i$  are indexed by  $k_i$ , where  $k_i \triangleq \frac{k}{\nu_i}$  and  $k_M = k$ . Let  $k_{i,F} \triangleq \frac{k_F}{\nu_i}$  such that  $k_i \in [0, k_{i,F}]$ . Fig. 3.1 shows how the conditions of **Assumption 4** determine the relationships between time step sizes and prediction horizons for a three-level hierarchical controller.

Each controller  $\mathbf{C}_i$  updates only when  $k = \nu_i k_i$  (i.e. when  $k \bmod \nu_i = 0$ ), by solving the constrained optimization problem  $\mathbf{P}_i(x(k))$  defined as

$$J_i^*(x(k)) = \min_{\substack{\hat{x}_i(k_i|k_i) \\ \hat{u}_i(k_i)}} \sum_{j=k_i}^{k_i+N_i(k_i)} \ell(\hat{x}_i(j|k_i), \hat{u}_i(j|k_i), r_i(j)), \quad (3.6a)$$

$$\text{s.t. } \forall j \in [k_i, k_i + N_i(k_i)]$$

$$\hat{x}_i(j+1|k_i) = A_i \hat{x}_i(j|k_i) + B_i \hat{u}_i(j|k_i) + \hat{d}_i(j), \quad (3.6b)$$

$$\hat{y}_i(j|k_i) = C \hat{x}_i(j|k_i) + D \hat{u}_i(j|k_i) \in \hat{\mathcal{Y}}_i(j), \quad (3.6c)$$

$$\hat{x}_i(k_i + N_i(k_i)|k_i) \in \hat{\mathcal{S}}_i(k_i + N_i(k_i)), \quad (3.6d)$$

$$x(k) - \hat{x}_i(k_i|k_i) \in \mathcal{E} \vee \hat{x}_i(k_i|k_i) = \hat{x}_i^*(k_i|k_i - 1). \quad (3.6e)$$

First, note that  $\mathbf{P}_i(x(k))$  has a *shrinking horizon*, based on the summation limits in (3.6a), with horizon length  $N_i(k_i) \triangleq N_i - (k_i \bmod N_i)$  where  $N_i$  satisfies **Assumption 4ii**. Thus,

$\mathbf{C}_i$  predicts between the current time step and the time step of the next update of  $\mathbf{C}_{i-}$ , at which point  $(k_i \bmod N_i) = 0$  and prediction horizon resets back to  $N_i(k_i) = N_i$ . The nominal input sequence over this horizon is defined as  $\hat{U}_i(k_i) = \{\hat{u}_i(j|k_i)\}_{j=k_i}^{k_i+N_i(k_i)-1}$ , with the optimal nominal sequence denoted as  $\hat{U}_i^*(k_i)$ . In (3.6b), the model used by  $\mathbf{C}_i$  assumes a piecewise constant nominal control input over the time step size  $\Delta t_i$  and thus  $A_i = A^{\nu_i}$  and  $B_i = \sum_{j=0}^{\nu_i-1} A^j B$  (as in [74]). Since the known disturbance is time varying over the time step size  $\Delta t_i$ , the known disturbance used in (3.6b) is

$$\hat{d}_i(k_i) = \sum_{j=0}^{\nu_i-1} A^{\nu_i-1-j} \hat{d}(\nu_i k_i + j), \quad (3.7)$$

which ensures that  $\hat{d}_i(k_i)$  captures the accumulated effect of the known time-varying disturbances,  $\hat{d}(k), k \in [\nu_i k_i, \nu_i k_i + \nu_i - 1]$ , during this slow time step. The nominal outputs  $\hat{y}_i(j|k_i)$  in (3.6c) are constrained to the time-varying tightened output constraint set  $\hat{\mathcal{Y}}_i(j)$ , with details provided in Sections 3.5.1 and 3.5.2. The time-varying terminal state constraint in (3.6d) corresponds to the waysets  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$  used as the only coordination mechanism between controllers  $\mathbf{C}_i$  and  $\mathbf{C}_{i-}$ . The properties of these waysets are defined in Section 3.5.3 and the calculation of the waysets is provided in Section 3.7.2. Finally, (3.6e) provides  $\mathbf{C}_i$  the choice of nominal initial condition,  $\hat{x}_i(k_i|k_i)$ , which is a decision variable following the tube-based MPC formulation in [60]. The reasoning for this specific treatment of the initial condition is detailed in Section 3.5.4.

As shown in Fig. 3.2, coordination is achieved among the controllers through the use of waysets imposed as terminal constraints (3.6d). Within this hierarchical control framework, only the lowest level controller  $\mathbf{C}_M$  directly affects the system. Once  $\mathbf{C}_M$  has solved for the optimal nominal control input trajectory  $\hat{U}_M^*(k_M)$  and optimal nominal initial condition  $\hat{x}_M^*(k_M|k_M)$ , the input to the system  $u(k)$  is calculated based on the control law

$$u(k) = \hat{u}_M^*(k_M|k_M) + K[x(k) - \hat{x}_M^*(k_M|k_M)], \quad (3.8)$$

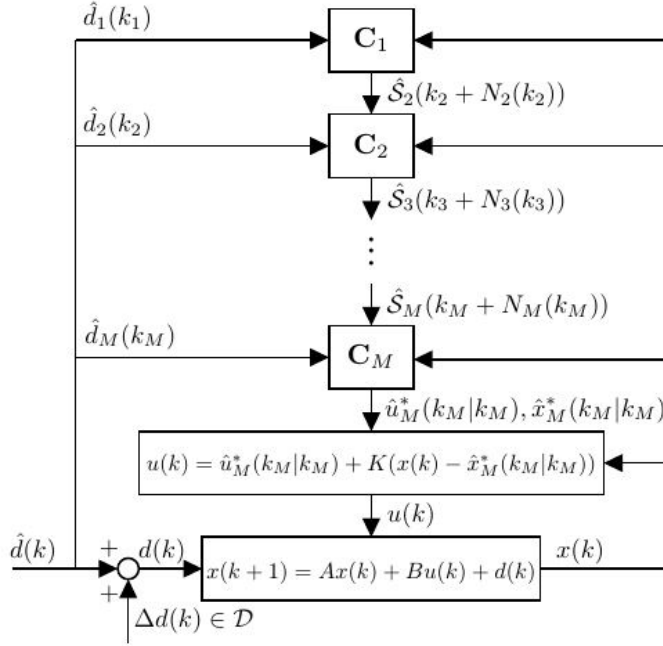


Figure 3.2. The hierarchical MPC structure with  $M$  levels, where controllers  $\mathbf{C}_i, i \in [1, M]$  are formulated based on (3.6), the known disturbances  $\hat{d}_i(k_i)$  are computed based on (3.7), the waysets  $\hat{\mathcal{S}}_i(k_i + N_i(k_i)), i \in [2, M]$ , are used to coordinate controllers  $\mathbf{C}_i$  and  $\mathbf{C}_{i-}$ , and the static feedback control law (3.8) bounds the effect of the unknown disturbance  $\Delta d(k) \in \mathcal{D}$ .

where  $K \in \mathbb{R}^{m \times n}$  is a static feedback control gain. This control law is used to bound the error between the nominal and true system state trajectories created by the unknown disturbances  $\Delta d(k)$ . Section 3.5.1 details the design of  $K$  and the resulting set  $\mathcal{E}$  that bounds this error.

In summary, the  $M$ -level hierarchical controller is implemented based on **Algorithm 3**. The constrained optimization problem  $\mathbf{P}_i(x(k))$  for each controller  $\mathbf{C}_i, i \in [1, M]$  is specifically designed with the nominal model of (3.6b), the time-varying tightened output constraints of (3.6c), the wayset terminal constraints of (3.6d), and the initial state condition of (3.6e) to establish guaranteed robust satisfaction of output and terminal constraints as proven in Section 3.6.

**Remark 5.** As discussed in [48], neither the references nor the exact formulation of cost function in (3.6a) affect the feasibility of any  $\mathbf{P}_i(x(k))$  in the hierarchical controller.



---

**Algorithm 3:**  $M$ -level hierarchical MPC

---

```
1 initialize  $k, k_i \leftarrow 0, \forall i \in [1, M]$ 
2 while  $k < k_F$  do
3   for  $i = 1$  to  $M - 1$  do
4     if  $k \bmod \nu_i = 0$  then
5       solve  $\mathbf{P}_i(x(k))$ ;
6       calculate  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$  and communicate to  $\mathbf{P}_{i+1}(x(k))$ ;
7        $k_i \leftarrow k_i + 1$ ;
8     end
9     solve  $\mathbf{P}_M(x(k))$  and apply the input  $u(k)$  to the system based on (3.8);
10     $k_M \leftarrow k_M + 1$ ;
11     $k \leftarrow k + 1$ ;
12  end
13 end
```

---

**Remark 6.** While the focus of this chapter is on hierarchical control for systems with finite operation per Assumption 1, indefinite system operation can be achieved by replacing the shrinking prediction horizon,  $N_1$ , of  $C_1$  with a receding horizon of fixed length. To guarantee recursive feasibility, the terminal set  $\mathcal{T}$  must be a Robust Positive Invariant (RPI) set based on pre-determined bounds of  $d(k)$ , as done in centralized MPC formulations [68].

## 3.5 Set Definitions

### 3.5.1 Output Constraint Tightening for $\mathbf{C}_M$

The proposed hierarchical MPC framework is robust to unknown bounded disturbances using the tube-based MPC formulation developed in [60]. For the lowest level controller  $\mathbf{C}_M$ , the only difference between the true system (3.1) and the model used for control (3.6b) is the unknown bounded disturbance  $\Delta d(k) \in \mathcal{D}$ . Using the control law (3.8), and comparing (3.1) and (3.6b) for  $i = M$ , the error  $e(k) = x(k) - \hat{x}_M^*(k_M|k_M)$  satisfies

$$e(k+1) = (A + BK)e(k) + \Delta d(k). \quad (3.9)$$

Assuming  $K$  is designed to stabilize  $A + BK$  and  $\mathcal{E} \in \mathbb{R}^n$  is a disturbance invariant set for (3.9), then

$$(A + BK)\mathcal{E} \oplus \mathcal{D} \subseteq \mathcal{E}. \quad (3.10)$$

Thus, if  $e(k) \in \mathcal{E}$ , then  $e(k + 1) \in \mathcal{E}$  for all  $\Delta d(k) \in \mathcal{D}$ . As in [60], the constraint (3.6e) on the initial condition allows  $\mathbf{C}_M$  to choose  $\hat{x}_M(k_M|k_M)$  such that  $e(k) \in \mathcal{E}$ . The minimal disturbance invariant set [50] is

$$\mathcal{E} = \bigoplus_{i=0}^{\infty} (A + BK)^i \mathcal{D}, \quad (3.11)$$

and should be as small as possible to reduce conservatism of the controller. However, due to the infinite sum, computing  $\mathcal{E}$  is difficult and a outer approximation  $\tilde{\mathcal{E}}$  is typically used where  $\mathcal{E} \subseteq \tilde{\mathcal{E}}$ ,  $\tilde{\mathcal{E}}$  satisfies (3.10), and  $\tilde{\mathcal{E}}$  can be represented as a polytope [68]. For the remainder of the paper  $\mathcal{E}$  and  $\tilde{\mathcal{E}}$  are used interchangeably.

From [60], when designing a robust MPC controller for  $\mathbf{C}_M$  with the nominal system model (3.6b), tightened state, input, and terminal constraint sets are used where

$$\hat{\mathcal{X}} \triangleq \mathcal{X} \ominus \mathcal{E}, \quad \hat{\mathcal{U}} \triangleq \mathcal{U} \ominus K\mathcal{E}, \quad \hat{\mathcal{T}} \triangleq \mathcal{T} \ominus \mathcal{E}. \quad (3.12)$$

It is assumed that  $\mathcal{D}$ , and thus  $\mathcal{E}$ , are small relative to the state and input constraint sets such that  $\hat{\mathcal{X}}, \hat{\mathcal{U}}, \hat{\mathcal{T}} \neq \emptyset$ . Based on (3.5), the tightened output constraint set is

$$\hat{\mathcal{Y}} \triangleq \mathcal{Y} \ominus (\mathcal{E} \times K\mathcal{E}). \quad (3.13)$$

Since  $\mathbf{C}_M$  has the same time step size as the system, i.e.  $\nu_M = 1$ ,  $\hat{\mathcal{Y}}_M(k_M) = \hat{\mathcal{Y}}$  for all  $k_M$ . However, for  $\mathbf{C}_i, i \in [1, M - 1]$ , additional time-varying constraint tightening is required to account for inter-sample behavior between the slow updates of these upper-level controllers.

### 3.5.2 Output Constraint Tightening for $\mathbf{C}_i, i \in [1, M - 1]$

As shown in Fig. 3.3(a), constraining the slow nominal output trajectory  $\hat{y}_i(k_i) \in \hat{\mathcal{Y}}$  planned by  $\mathbf{C}_i, i \in [1, M - 1]$  does not guarantee that  $\hat{y}(k) \in \hat{\mathcal{Y}}$  during the inter-sample updates

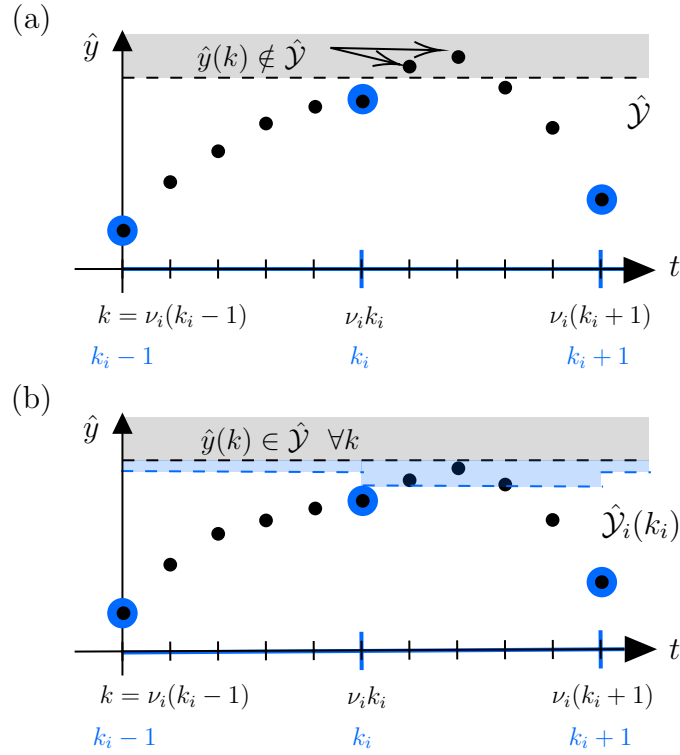


Figure 3.3. (a) Controller  $\mathbf{C}_i$  plans a feasible state trajectory (large blue dots) at the slow time step  $k_i$  but the resulting trajectory (small black dots) violates output constraints (dashed black line) at the inter-sample system time steps. (b) Tightening the output constraint set (dashed blue line) for  $\mathbf{C}_i$  ensures that any trajectory at the slow time index is also feasible at the faster time indices.

when  $k = [\nu_i k_i + 1, \nu_i(k_i + 1) - 1]$ . Thus, it is important to further tighten the constraint set  $\hat{\mathcal{Y}}$  to account for transient state trajectories between the slow updates, as shown in Fig. 3.3(b).

**Definition 6.** *The time-varying tightened nominal output constraint set  $\hat{\mathcal{Y}}_i(k_i)$  is the set of all initial nominal state and nominal input combinations that results in an output trajectory satisfying (3.13) if the input is held constant for  $\nu_i$  steps, i.e.*

$$\hat{\mathcal{Y}}_i(k_i) \triangleq \left\{ \hat{y}_i(k_i) = \begin{bmatrix} \hat{x}_i(k_i) \\ \hat{u}_i(k_i) \end{bmatrix} \mid \forall k \in [\nu_i k_i, \nu_i(k_i + 1) - 1], \right.$$

$$\begin{aligned} \hat{y}(k) &= \begin{bmatrix} \hat{x}(k) \\ \hat{u}(k) \end{bmatrix} \in \hat{\mathcal{Y}}, \quad \hat{u}(k) = \hat{u}_i(k_i), \\ \hat{x}(k+1) &= A\hat{x}(k) + B\hat{u}(k) + \hat{d}(k), \\ \hat{x}(\nu_i k_i) &= \hat{x}_i(k_i) \}. \end{aligned}$$

Similar to the procedure presented in [48],  $\hat{\mathcal{Y}}_i(k_i)$  is calculated based on the tightened output constraint set  $\hat{\mathcal{Y}}$  represented in H-Rep as

$$\hat{\mathcal{Y}} = \left\{ (\hat{x}(k), \hat{u}(k)) \mid \hat{P} \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{u}(k) \end{bmatrix} \leq \hat{q} \right\}. \quad (3.14)$$

For  $i \in [1, M-1]$ ,  $\hat{\mathcal{Y}}_i(k_i)$  is time-varying due to the dependence on  $\hat{d}_i(k_i)$  and may be computed as

$$\hat{\mathcal{Y}}_i(k_i) = \hat{\mathcal{Y}} \cap \hat{\mathcal{Y}}(\nu_i k_i + 1) \cap \cdots \cap \hat{\mathcal{Y}}(\nu_i k_i + \nu_i - 1), \quad (3.15)$$

where, for all  $j \in [1, \nu_i - 1]$ ,

$$\hat{\mathcal{Y}}(\nu_i k_i + j) = \left\{ (\hat{x}(k), \hat{u}(k)) \mid \hat{P} \begin{bmatrix} C_j & D_j \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{u}(k) \end{bmatrix} \leq \hat{q}(\nu_i k_i + j) \right\}, \quad (3.16)$$

with  $C_j = CA^j$ ,  $D_j = D + C \sum_{l=0}^{j-1} A^l B$ , and

$$\hat{q}(\nu_i k_i + j) = \hat{q} - \hat{P}C \sum_{l=0}^{j-1} A^{j-1-l} \hat{d}(\nu_i k_i + l). \quad (3.17)$$

**Lemma 6.** For all  $i < j$ ,  $i, j \in [1, M]$ , at time step  $k = \nu_i k_i = \nu_j k_j$ ,  $\hat{\mathcal{Y}}_i(k_i) \subseteq \hat{\mathcal{Y}}_j(k_j) \subseteq \hat{\mathcal{Y}}$ .

*Proof.* See [48]. □

### 3.5.3 Waysets

**Definition 7.** The wayset  $\mathcal{S}(k) \subset \mathcal{X}$  denotes a set of states at time step  $k$  such that for any  $x(k) \in \mathcal{S}(k)$  there exists a future input trajectory  $\{u(k)\}_{k=k}^{k_F-1}$  and corresponding state trajectory  $\{x(k)\}_{k=k}^{k_F}$  satisfying (3.1-3.3).

In this work, the waysets imposed as terminal constraints in (3.6d) are the sole coordination mechanism between levels of the hierarchical controller. Since (3.6d) imposes a constraint on the nominal predicted state, nominal waysets are used and denoted as  $\hat{\mathcal{S}}$ . Since the waysets are used to guarantee feasibility of constraints beyond the prediction horizon of lower-level controllers, each controller  $\mathbf{C}_i$  has a different wayset denoted as  $\hat{\mathcal{S}}_i$ . Finally, the waysets are time-varying, denoted as  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$ , where the time step always corresponds to the time step of the next update of  $\mathbf{C}_{i-}$ . Within the context of **Definition 7**, there are many possible ways to formulate  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$ . In this paper, waysets are formulated to satisfy the following assumptions in order to prove feasibility of the  $M$ -level hierarchical controller.

**Assumption 5.** *The waysets  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$  in (3.6d) satisfy the following:*

1. for  $\mathbf{C}_1$ ,  $\hat{\mathcal{S}}_1(k_1 + N_1(k_1)) = \hat{\mathcal{T}}$ ,
2. for  $\mathbf{C}_i$ ,  $i \in [2, M]$ ,
  - (a)  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$  is only recomputed at updates of  $\mathbf{C}_{i-}$  per **Algorithm 3**,
  - (b)  $\hat{x}_{i-}^*(k_{i-} + 1|k_{i-}) \in \hat{\mathcal{S}}_i(k_i + N_i(k_i))$ ,
  - (c) if  $\hat{x}_{i-}^*(k_{i-} + 2|k_{i-})$  exists, for each state in  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$  there exists a trajectory satisfying (3.6b) and (3.6c) that drives the system to  $\hat{x}_{i-}^*(k_{i-} + 2|k_{i-})$ ,
  - (d) if  $\hat{x}_{i-}^*(k_{i-} + 2|k_{i-})$  does not exist,  $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{S}}_{i-}(k_{i-} + N_{i-}(k_{i-}))$ ,
  - (e) if  $k_i + N_i(k_i) = k_{i,F}$ ,  $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{T}}$ .

Conceptually, these assumptions state: 1) for the highest level controller, the wayset equals the terminal constraint set, since  $\mathbf{C}_1$  always predicts to the final time step per **Assumption 4iii**; 2a) since the wayset for  $\mathbf{C}_i$  depends on the state trajectory determined by  $\mathbf{C}_{i-}$ , waysets for  $\mathbf{C}_i$  are only recomputed when  $\mathbf{C}_{i-}$  updates; 2b) noting that the time

index of the wayset for  $\mathbf{C}_i$  corresponds to the time index for the second optimal nominal state in the trajectory determined by  $\mathbf{C}_{i-}$ , this optimal state exists in the wayset; 2c) since the prediction horizon for each controller shrinks over time, if the third optimal nominal state in the trajectory determined by  $\mathbf{C}_{i-}$  exists, then the wayset for  $\mathbf{C}_i$  is defined as all the nominal states such that there exists feasible nominal input and state trajectories that drive the nominal system to this state; 2d) if the third optimal nominal state in the trajectory determined by  $\mathbf{C}_{i-}$  does not exist, then the wayset for  $\mathbf{C}_i$  is set equal to the wayset for  $\mathbf{C}_{i-}$ ; and 2e) if  $\mathbf{C}_i$  predicts to the final time step, the wayset is set equal to the tightened terminal constraint set.

### 3.5.4 Initial Conditions

In the formulation of  $\mathbf{P}_i(x(k))$ , (3.6e) provides  $\mathbf{C}_i$  with two options for the choice of nominal initial condition,  $\hat{x}_i(k_i|k_i)$ . The first option,  $x(k) - \hat{x}_i(k_i|k_i) \in \mathcal{E}$ , comes from the tube-based MPC formulation presented in [60]. The second option,  $\hat{x}_i(k_i|k_i) = \hat{x}_i^*(k_i|k_i - 1)$ , similar to [38] allows the nominal initial condition to equal the optimal nominal state for this time step determined by  $\mathbf{C}_i$  at the previous time step. To understand the role of this initial condition option, consider the following definition and assumption.

**Definition 8.** [78] *The feasible set  $\mathbb{X}_{\mathcal{T}}^{(N_1)} \subset \mathcal{X}$  denotes the set of states such that  $\mathbf{P}_1(x(k))$  admits a solution, with prediction horizon  $N_1$  and terminal set  $\mathcal{T}$ .*

**Assumption 6.** *The initial condition satisfies  $x(0) \in \mathbb{X}_{\mathcal{T}}^{(N_1)}$  and thus there exists a feasible solution to  $\mathbf{P}_1(x(0))$  at time step  $k = k_1 = 0$ .*

For clarity of exposition, Fig. 3.4 demonstrates the need for the initial condition option in (3.6e) for the case where  $M = 2$  and  $\Delta d(k) = 0$ . With a prediction horizon of  $N_1 = 3$ , Fig. 3.4 shows how  $x(0) \in \mathbb{X}_{\mathcal{T}}^{(3)}$  by **Assumption 6**. Therefore,  $\mathbf{C}_1$  has a feasible state trajectory  $\{\hat{x}_1^*(j|0)\}_{j=0}^3$ . The feasibility of this trajectory implies  $\hat{x}_1^*(1|0) \in \mathbb{X}_{\mathcal{T}}^{(2)}$  and  $\hat{x}_1^*(2|0) \in \mathbb{X}_{\mathcal{T}}^{(1)}$ . Per

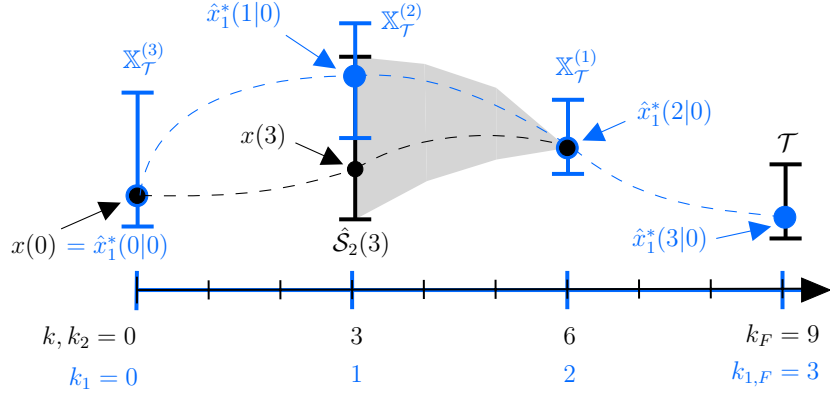


Figure 3.4. Schematic showing the need for the initial condition option in (3.6e) where  $x(3)$  satisfies the wayset constraint  $x(3) \in \hat{\mathcal{S}}_2(3)$  but  $x(3) \notin \mathbb{X}_{\mathcal{T}}^{(2)}$ .

**Assumption 5**,  $\hat{\mathcal{S}}_2(3)$  denotes the set of states such that there exists a trajectory satisfying (3.6b) and (3.6c) which drives the system to  $\hat{x}_1^*(2|0)$ . As proven in the following section, the lower-level controller(s) will drive the system from  $x(0)$  to  $x(3) \in \hat{\mathcal{S}}_2(3)$ , however there is no guarantee that  $x(3) \in \mathbb{X}_{\mathcal{T}}^{(2)}$ . If  $x(3) \notin \mathbb{X}_{\mathcal{T}}^{(2)}$ , then the second initial condition option in (3.6e) is required to maintain feasibility of  $\mathbf{P}_1(x(3))$ . The following section details how the initial condition option and properties of the waysets defined in **Assumption 5** establish feasibility of all controllers.

### 3.6 Hierarchical Control Feasibility

The following lemmas establish feasibility of individual controllers within the hierarchy starting with the highest-level controller  $\mathbf{C}_1$ .

**Lemma 7.** *If  $\mathbf{P}_1(x(k))$  is feasible at  $k = \nu_1 k_1$ , then  $\mathbf{P}_1(x(k))$  is feasible at  $k = \nu_1(k_1 + 1)$ .*

*Proof.* As discussed in [48], the optimal solution at  $k = \nu_1(k_1 + 1)$  is the tail of trajectories determined at the previous time step  $k = \nu_1 k_1$ . To show that this candidate solution satisfies the constraints in (3.6), first note that the nominal system model is time-invariant and that while  $\hat{d}_1(k_1)$  is time-varying, the trajectory of  $\hat{d}_1(k_1)$  is known per **Assumption 2** and does

not change during system operation. Thus, the candidate solution satisfies (3.6b). Similarly, while  $\hat{\mathcal{Y}}_1(k_1)$  varies with  $k_1$ ,  $\hat{\mathcal{Y}}_1(k_1)$  at a particular  $k_1$  remains constant since the time step dependency only comes from the dependence of  $\hat{\mathcal{Y}}_1(k_1)$  on  $\hat{d}(k)$  per (3.15)-(3.17). Thus, the candidate solution satisfies (3.6c). Since  $\hat{\mathcal{S}}_1(k_1 + N_1(k_1)) = \hat{\mathcal{T}}$  is time-invariant, the candidate solution satisfies (3.6d). Finally, (3.6e) provides the option to let  $\hat{x}_1(k_1|k_1) = \hat{x}_1^*(k_1|k_1 - 1)$ . Note, this candidate solution at  $k = \nu_1(k_1 + 1)$  is optimal if  $x(k) - \hat{x}_1(k_1 + 1|k_1 + 1) \in \mathcal{E}$  does not admit a feasible solution.  $\square$

Next, if  $\mathbf{C}_i$  is feasible at the time step of the upper-level controller  $\mathbf{C}_{i-}$  update, then  $\mathbf{C}_i$  remains feasible for all time steps until the next update of  $\mathbf{C}_{i-}$ .

**Lemma 8.** *If  $\mathbf{P}_i(x(k))$  is feasible at  $k = \nu_i k_i$ , where  $k \bmod \nu_{i-} = 0$  (i.e at the time of a  $\mathbf{C}_{i-}$  update), then  $\mathbf{P}_i(x(k))$  is feasible at each time step  $k = \nu_i(k_i + 1)$  through  $k = \nu_i(k_i + N_i - 1)$ .*

*Proof.* See [48].  $\square$

Finally, at the time step of the upper-level controller  $\mathbf{C}_{i-}$  update, feasibility of  $\mathbf{C}_{i-}$  guarantees feasibility of  $\mathbf{C}_i$ .

**Lemma 9.** *If  $\mathbf{P}_{i-}(x(k))$  has a feasible solution at  $k = \nu_{i-} k_{i-}$  and  $\mathbf{P}_M(x(k - 1))$  had a feasible solution at the previous time step  $k - 1$ , then  $\mathbf{P}_i(x(k))$  has a feasible solution at this time step.*

*Proof.* The proof for the robust case presented in this paper is similar to the nominal case presented in [48] where two cases must be considered based on

1.  $x(k) - \hat{x}_{i-}^*(k_{i-}|k_{i-}) \in \mathcal{E}$  or,
2.  $x_{i-}^*(k_{i-}|k_{i-}) = x_{i-}^*(k_{i-}|k_{i-} - 1)$ .



For (1), the feasible solutions to  $\mathbf{P}_i(x(k))$ , comes directly from the feasible solutions determined by  $\mathbf{P}_{i-}(x(k))$  as shown in [48].

For (2), a feasible solution to  $\mathbf{P}_i(x(k))$  exists with a nominal state trajectory satisfying  $x(k) - \hat{x}_i(k_i|k_i) \in \mathcal{E}$ . If  $\mathbf{P}_M(x(k-1))$  had a feasible solution at the previous time step  $k-1$ , then  $x(k-1) - \hat{x}_M^*(k_M-1|k_M-1) \in \mathcal{E}$  due to (3.6e) and  $x(k) - \hat{x}_M^*(k_M|k_M-1) \in \mathcal{E}$  due to the invariance of  $\mathcal{E}$  under control law (3.8). Thus,  $\hat{x}_M^*$  is a feasible initial condition for  $\mathbf{P}_i(x(k))$  that satisfies constraints (3.6b)-(3.6e) as shown in [48].  $\square$

The main results of this work guarantees constraint satisfaction for the  $M$ -level hierarchical controller.

**Theorem 8.** *Following **Algorithm 3** for an  $M$ -level hierarchical controller, all control problems  $\mathbf{P}_i(x(k))$ ,  $i \in [1, M]$  are feasible when solved at  $k \bmod \nu_i = 0$ , resulting in system state and input trajectories satisfying constraints (3.2) and (3.3).*

*Proof.* Using **Assumption 6** and **Lemmas 7-9**, Fig. 3.5 shows how feasibility is established for each  $\mathbf{C}_i$ ,  $i \in [1, M]$ . Due to the constraint tightening presented in Section 3.5.1, feasibility of  $\mathbf{P}_M(x(k))$  and the use of control law (3.8) guarantees the satisfaction of (3.2). By the wayset properties defined in **Assumption 5.2e**,  $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{T}}$  once  $k_i + N_i(k_i) = k_{i,F}$  and thus feasibility of  $\mathbf{P}_M(x(k))$  also guarantees the satisfaction of (3.3).  $\square$

**Remark 7.** *With all states and inputs constrained in (3.2), the constraint satisfaction established in **Theorem 8** also provides bounded input bounded output (BIBO) stability. For many applications, BIBO stability is preferred over asymptotic stability so that the dynamics of the system can be used to maximize performance as in the case of completion-based MPC [70].*



from (3.6c). Finally, since waysets define a set of states, the projection operation is used to project the calculated output wayset into the first  $n$  dimensions.

---

**Algorithm 4:** Wayset  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$  computation for  $i \in [2, M]$  at time step  $k = \nu_{i-} k_{i-}$ .

---

```

1 initialize  $j \leftarrow N_i$ 
2 if  $N_{i-}(k_{i-}) < 2$  then
3    $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{S}}_{i-}(k_{i-} + N_{i-}(k_{i-}))$ ;
4 else
5    $\hat{\mathcal{S}}_i(j) = \hat{x}_{i-}^*(k_{i-} + 2|k_{i-})$ ;
6   while  $j \geq 1$  do
7      $\tilde{\mathcal{Y}}_i^{\hat{\mathcal{S}}_i(j)} = \left\{ (\hat{x}_i, \hat{u}_i) \mid \hat{x}_i^+ \in \hat{\mathcal{S}}_i(j), \hat{x}_i^+ = A_i \hat{x}_i + B_i \hat{u}_i + \hat{d}_i(k_i + N_i + j - 1) \right\}$ ;
8      $\hat{\mathcal{S}}_i(j - 1) = \pi_n(\tilde{\mathcal{Y}}_i^{\hat{\mathcal{S}}_i(j)} \cap_I \hat{\mathcal{Y}}_i(k_i + N_i + j - 1))$ ;
9      $j \leftarrow j - 1$ ;
10  end
11   $\hat{\mathcal{S}}_i(k_i + N_i(k_i)) = \hat{\mathcal{S}}_i(j)$ 
12 end

```

---

While the steps in **Algorithm 4** conceptually define the wayset calculations, these steps can be simplified using the notion of generalized intersection. Note that line 7 is equivalent to

$$[A_i \ B_i] \hat{y}_i \in \hat{\mathcal{S}}_i(j) \ominus \hat{d}_i(k_i + N_i + j - 1), \quad (3.18)$$

where the Pontryagin difference simply shifts the center of  $\hat{\mathcal{S}}_i(j)$  since  $\hat{d}_i(k_i + N_i + j - 1)$  is a vector and not a set. This condition and the tightened output constraint condition from (3.6c),  $\hat{y}_i \in \hat{\mathcal{Y}}_i(k_i + N_i + j - 1)$  must be satisfied. The generalized intersection can be used to enforce both conditions as

$$\hat{y}_i \in \hat{\mathcal{Y}}_i(k_i + N_i + j - 1) \cap_{[A_i \ B_i]} \left( \hat{\mathcal{S}}_i(j) \ominus \hat{d}_i(k_i + N_i + j - 1) \right). \quad (3.19)$$

Projection can then be used to transform this output constraint set to a state constraint set.

As discussed in [78], the iterative approach provides improved computational efficiency compared to projection-based methods. However, when polytopic constraint sets and waysets

are represented in H-Rep or as a convex hull of vertices (V-Rep), on-line wayset calculation may still be limited by computational cost. As discussed in [79] and the references therein, the worst-case complexity of linear transformation, Minkowski sum, and generalized intersection scales exponentially in the set dimension. Therefore, **Algorithm 4** is likely to be very computationally expensive and potentially numerically unstable for  $n, m$  greater than about five (potentially less than 5 if the number of halfspaces or vertices defining the polytopes is large). In fact, the numerical examples in [78] are restricted to  $n \leq 4$ ,  $m \leq 2$ , and less than ten iteration steps.

Under **Algorithm 3**,  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$  is recomputed on-line at every update of  $\mathbf{C}_{i-}$ . Thus, efficient set computations are critical to the wayset-based hierarchical control. The following section demonstrates how zonotopes can be used to significantly reduce the cost of computing waysets, enabling the proposed approach.

### 3.7.3 Zonotope-based Set Calculation (See Chapter 2 for more details)

To perform the wayset calculations using zonotopes, it is necessary to convert sets from H-Rep to CG-Rep (See Section 2.4 for additional details). Note the following details when executing **Algorithm 4** using constrained zonotopes. Line 5 initializes the wayset as a point from the optimal nominal state trajectory determined by  $\mathbf{C}_{i-}$  as  $\hat{\mathcal{S}}_i(j) = \{[\cdot], \hat{x}_{i-}^*(k_{i-} + 2|k_{i-}), [\cdot], [\cdot]\}$ . As discussed in Section 3.7.2, Line 7 and the intersection operation in Line 8 are expressed as the generalized intersection in (3.19). The projection operation in Line 8 is equivalent to a linear transformation in CG-Rep where  $R$  from (2.1) is  $R = [I_n \ 0_{n \times m}]$ .

Based on **Algorithm 4**, the number of generators  $n_g$  and number of constraints  $n_c$  required to represent  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$  grows linearly with the prediction horizon  $N_i$ . Assuming  $n$  states and that the CG-Rep of  $\hat{\mathcal{Y}}_i(k_i + N_i + j - 1)$  has  $n_{g, \hat{\mathcal{Y}}_i}$  generators and  $n_{c, \hat{\mathcal{Y}}_i}$  constraints, the generalized intersection in (3.19) adds  $n_{g, \hat{\mathcal{Y}}_i}$  generators and  $n + n_{c, \hat{\mathcal{Y}}_i}$  constraints to the CG-Rep of  $\hat{\mathcal{S}}_i(j)$ . Thus, for a prediction horizon of  $N_i$ , the wayset  $\hat{\mathcal{S}}_i(k_i + N_i(k_i))$  has  $n_{g, \hat{\mathcal{Y}}_i} N_i$  generators and  $(n + n_{c, \hat{\mathcal{Y}}_i}) N_i$  constraints.

As shown in the following numerical examples, CG-Rep reduces the time required to compute waysets by several orders-of-magnitude compared to H-Rep.

### 3.8 Numerical Examples

To demonstrate the formulation and use of waysets in hierarchical MPC, this section presents two numerical examples. The first example is the same vehicle system from [47, 48] and highlights the wayset and tightened output constraint set calculations, the resulting robust constraint satisfaction, and the overall closed-loop performance and computational cost of the two-level hierarchy compared to centralized MPC. The second example presents a three-level hierarchy for a linearized thermal system to demonstrate the scalability of the approach. All results were generated using MATLAB on a desktop computer with a 3.6 GHz i7 processor and 16 GB of RAM and all MPC optimization problems were formulated and solved with YALMIP [57] and Gurobi [34].

#### 3.8.1 Vehicle Example

Consider the simplified vehicle system model

$$x(k+1) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix} u(k) + d(k),$$

where the states  $x(k) \in \mathbb{R}^3$  represent position, velocity, and on-board stored energy, and the inputs  $u(k) \in \mathbb{R}^3$  represent acceleration, deceleration, and power to an on-board load, all of which deplete the stored energy. Per Assumption 2, the disturbances  $d(k) \in \mathbb{R}^3$  with  $d(k) = \hat{d}(k) + \Delta d(k)$ .

The system and lowest level controller have time step sizes of  $\Delta t = \Delta t_M = 1$  second. Finite operation is defined for 100 seconds, thus  $k_F = 100$ . Choosing  $\Delta t_1 = 10$  seconds

results in  $\nu_1 = 10$  and maximum prediction horizons of  $N_1 = N_2 = 10$  steps. The output constraints defining  $\mathcal{Y}$  and  $\mathcal{T}$  are

$$\begin{bmatrix} -1 \\ -20 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq y(k) \leq \begin{bmatrix} 105 \\ 20 \\ 100 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} \leq x(k_F) \leq \begin{bmatrix} 1 \\ 1 \\ 100 \end{bmatrix}.$$

Given an initial state of  $x(0) = [0 \ 0 \ 100]^T$ , the desired operation, defined by  $\{r(k)\}_{k=0}^{k_F}$ , is shown in Fig. 3.6 for the first state (position), and third input (load power). References for the first and second inputs (acceleration and deceleration) are 0 for the entire operation, and thus are not shown in Fig. 3.6. These references are used to define (3.4) as the weighted quadratic cost function

$$\ell(x(j), u(j), r(j)) = \|r(j) - y_r(j)\|_{\Lambda}^2, \quad (3.20)$$

where  $y_r(j) = \begin{bmatrix} [1 \ 0 \ 0]x(j) \\ u(j) \end{bmatrix}$ ,  $\Lambda = \text{diag}([10^2 \ 10^0 \ 10^0 \ 10^2])$ .

The disturbances  $d(k)$  are shown in Fig. 3.7, which consist of a known pulse to the second state of magnitude 1.5 from 45 to 55 seconds. Note that the known disturbances are permitted to change between updates of the upper level controller. This unique feature of the proposed approach is enabled by calculating the equivalent known disturbances in (3.7) and the time-varying constraint tightening in (3.15). The unknown disturbances are independently generated from a uniformly distributed random signal bounded such that  $\|\Delta d(k)\|_{\infty} \leq 0.01$ . The static feedback control gain  $K \in \mathbb{R}^{m \times n}$  from (3.8) compensates for these unknown disturbances and was designed as a discrete-time linear-quadratic regulator with weighting matrices  $Q = I_n, R = I_m$ .

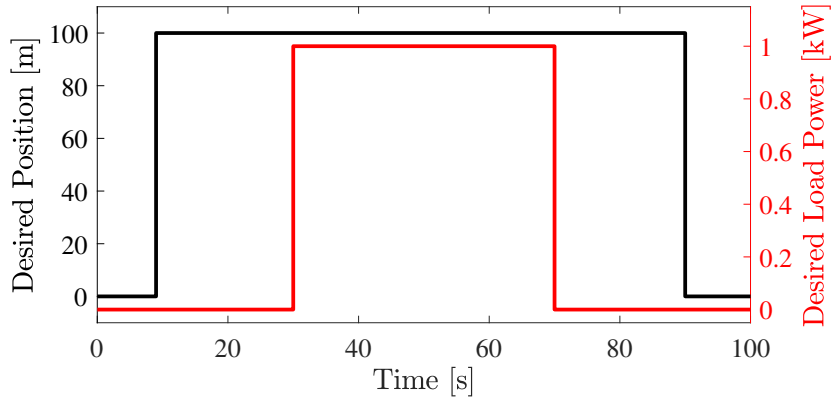


Figure 3.6. References for state and load power.

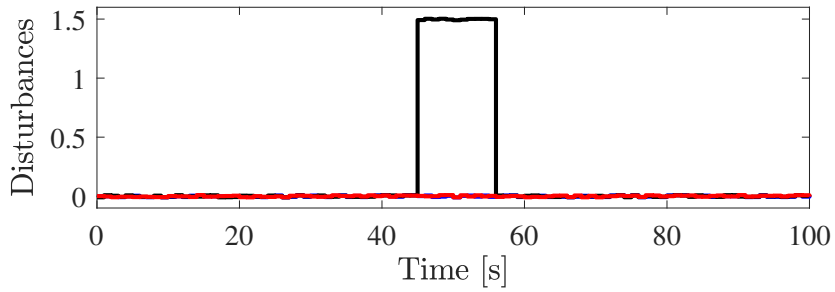


Figure 3.7. Disturbance profile consisting of a large known pulse and small unknown deviations.

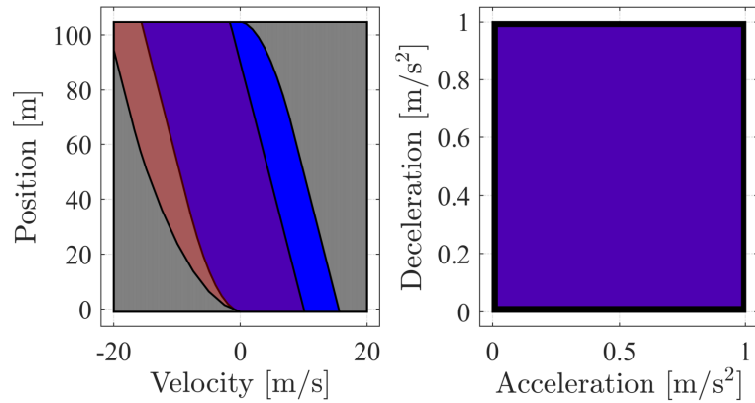


Figure 3.8. Example of constraint tightening with projections of various output constraint sets on the position and velocity states to the left and the acceleration and deceleration inputs to the right. Black denotes the original output constraints  $\mathcal{Y}$  while gray denotes the robust output constraints  $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_2(k_2), \forall k_2$ . Blue denotes the time-varying tightened output constraint set  $\hat{\mathcal{Y}}_1(k_1), \forall k_1 \neq 5$ , while semi-transparent red denotes  $\hat{\mathcal{Y}}_1(k_1), k_1 = 5$ .

Fig. 3.8 provides an example of the constraint tightening used by the two controllers of the hierarchy. Given the output constraint set  $\mathcal{Y}$ , the time-invariant tightened output constraint set  $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_2(k_2), \forall k_2$ , used by the lower-level controller  $\mathbf{C}_2$ , is calculated based on (3.13), where an outer approximation of  $\mathcal{E}$  is calculated using the results from [68]. Two examples of  $\hat{\mathcal{Y}}_1(k_1)$  are shown to demonstrate **Lemma 6** and the dependency of  $\hat{\mathcal{Y}}_1(k_1)$  on the time-varying known disturbance.

Fig. 3.9 shows simulation results using the proposed wayset-based hierarchical controller compared to a shrinking horizon centralized controller that predicts to the end of system operation and a receding horizon centralized controller (Cent Short) with a short prediction horizon of 10 time steps. Since tracking a desired position of 100 meters from 10 to 90 seconds is a major objective for the operation of the system, the first subplot in Fig. 3.9 highlights this part of operation. Both the centralized and hierarchical controllers track the desired reference while compensating for the known and unknown disturbances and satisfy the output constraints, shown here by keeping position below 105 meters. Alternatively, the optimization problem for the 10-step receding horizon centralized controller becomes infeasible at  $t = 11$  seconds when the velocity of the vehicle is too high to avoid violating the position constraint. The second and third subplots show that the wayset-based hierarchy satisfies the terminal constraint while maintaining a positive amount of stored energy. Finally, the fourth subplot shows the trajectory for input 3, from 30 to 70 seconds. Due to the limited amount of on-board energy, neither controller is able to track the desired load power and must shed some of the load to satisfy the terminal constraint. Centralized MPC evenly distributes this load shedding while  $\mathbf{C}_2$  of the hierarchy is more greedy and only load sheds once it is required to satisfy the wayset constraint determined by  $\mathbf{C}_1$ .

Table 3.1 shows the maximum, mean, and minimum complexity of the waysets for this example simulation using CG-Rep and H-Rep along with the computation time. Both H-Rep and min H-Rep calculations were performed using the Multi-Parametric Toolbox (MPT) [37].



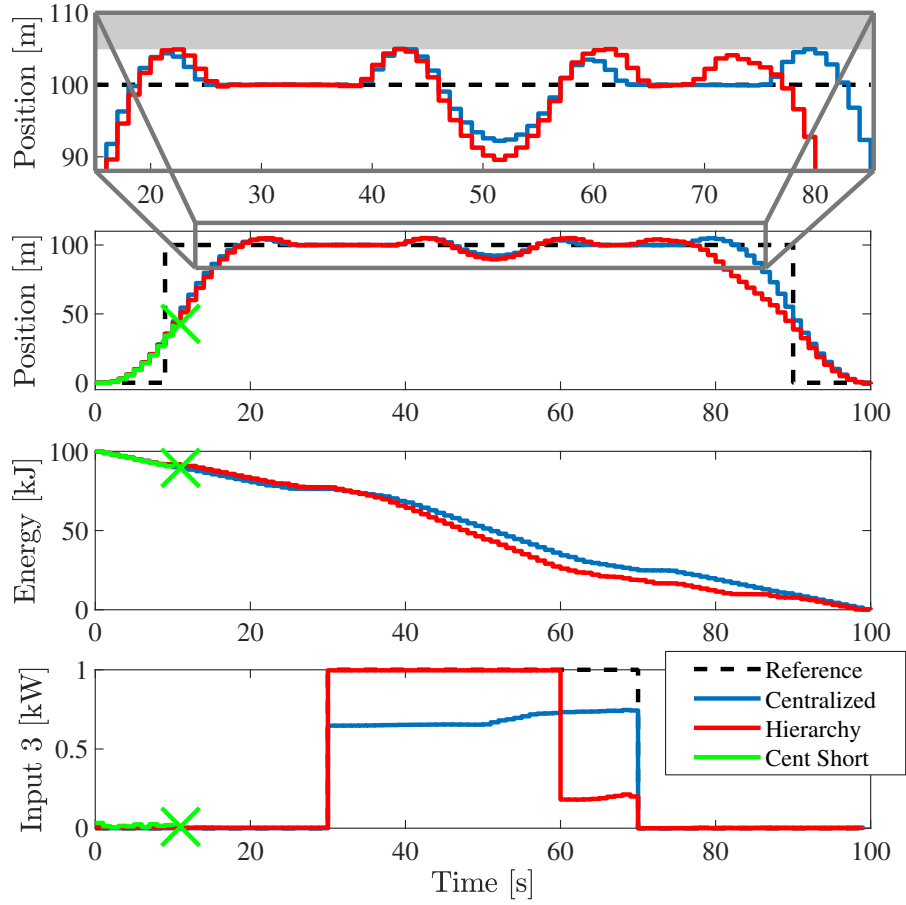


Figure 3.9. Simulation results comparing the shrinking horizon centralized controller, receding horizon centralized controller with a short prediction horizon, and a two-level hierarchical controller.

For CG-Rep and H-Rep, no attempt was made to remove redundant constraints. Overall, the CG-Rep achieves three to five orders-of-magnitude reduction in computation time, enabling on-line calculation of waysets. Note that 60 generators and 30 equality constraints were used to represent each wayset in CG-Rep regardless of the shape of each wayset. For scalability of the CG-Rep wayset calculations, it will be important to maintain a desired degree-of-freedom order,  $o_d \leq o_d^{des}$ .

Table 3.1. Complexity and Computation Time of Waysets  
Complexity

Wayset	Complexity		
	CG-Rep	H-Rep	min H-Rep
	$n_c \times n_g$	$n_h$	$n_h$
Maximum	$30 \times 60$	6913	197
Mean	$30 \times 60$	3120	108
Minimum	$30 \times 60$	38	21
Approximate Computation Time (seconds)			
	CG-Rep	H-Rep	min H-Rep
Maximum	0.006	393	353
Mean	0.002	120	114
Minimum	0.0006	0.51	0.38

### 3.8.2 Thermal Example

To evaluate the scalability of the hierarchical approach, consider the thermal system shown in Fig. 3.10, where  $T_i, i \in [1, n]$  are the temperatures of  $n$  thermal elements arranged in a chain, each with a thermal capacitance  $C_i$ . Heat transfer,  $Q_i$  between thermal elements  $T_i$  and  $T_{i+1}$  is controlled by the coolant mass flow rate  $\dot{m}_i$  resulting in  $Q_i = \dot{m}_i c_p (T_i - T_{i+1})$ , where  $c_p$  is the specific heat of the coolant. Disturbances consist of the heat input  $Q_0$  and the ambient temperature  $T_\infty$ . From conservation of energy, the nonlinear, continuous-time dynamics are  $C_i \dot{T}_i = Q_{i-1} - Q_i, \forall i \in [1, n]$ . For the following results,  $C_i = 10^4$  J/K and  $c_p = 4181$  J/(kg K) are assumed.

To represent this system in the form of (3.1), these dynamics are linearized about a nominal mass flow rate  $\dot{m}_i = \dot{m}^o = 0.25$  kg/sec and temperature difference  $T_i - T_{i+1} = \Delta T^o = 50$ K and discretized with a time step size of 1 second. The corresponding steady-state heat input is  $Q_0^o = \dot{m}^o c_p \Delta T^o$ . From this linearization, the states  $x(k) \in \mathbb{R}^n$  represent temperature deviations from nominal and the inputs  $u(k) \in \mathbb{R}^n$  represent mass flow rate deviations from nominal. The disturbances  $d(k) = \hat{d}(k) + \Delta d(k) \in \mathbb{R}^2$  represent deviations from the nominal heat input and nominal ambient temperature, satisfying **Assumption 2**. Finite operation

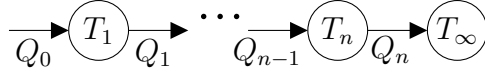


Figure 3.10. Thermal system comprised of a disturbance heat input  $Q_0$  and controllable heat transfer  $Q_i, i \in [1, n]$ , between  $n$  thermal elements of temperature  $T_i$  and ambient surroundings of temperature  $T_\infty$ .

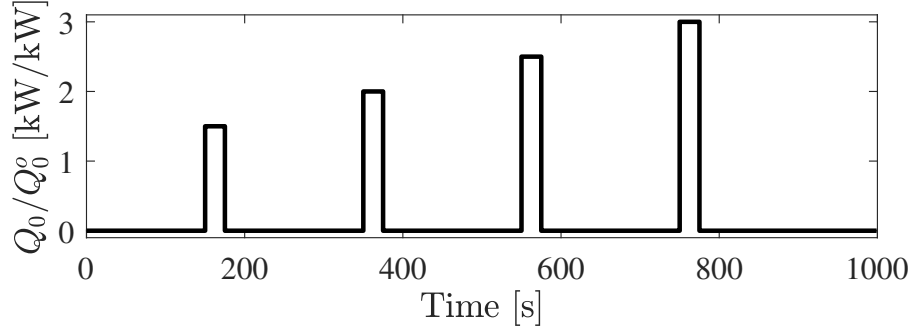


Figure 3.11. Known disturbance profile for the heat input  $Q_0$ .

is defined for 1000 seconds ( $k_F = 1000$ ) with the known ambient temperature remaining at nominal and the known heat input consisting of four pulses as shown in Fig. 3.11. The unknown disturbances are independently generated from a uniformly distributed random signal bounded such that  $|\Delta d(k)| \leq [0.1Q_0^% \ 5]^T$ . As with the previous vehicle example, the static feedback control gain  $K$  from (3.8) was designed as a discrete-time linear-quadratic regulator with weighting matrices  $Q = R = I_n$ .

For a three-level hierarchical controller, the system and lowest level controller have time step sizes of  $\Delta t = \Delta t_M = 1$  second while the middle and upper level controllers have time step sizes of  $\Delta t_2 = 5$  and  $\Delta t_1 = 40$  seconds, respectively. As a result,  $\nu_1 = 40$ ,  $\nu_2 = 5$ , and the maximum prediction horizons are  $N_1 = 25$ ,  $N_2 = 8$ , and  $N_3 = 5$  steps. The output constraints  $\mathcal{Y}$  are defined such that  $\|x(k)\|_\infty \leq 100$  and  $\|u(k)\|_\infty \leq 0.25, \forall k \in [0, k_F - 1]$ . The terminal constraint simply enforces  $\|x(k_F)\|_\infty \leq 100$ . Given an initial state of  $x(0) = 0$ , the desired operation is to satisfy the output and terminal constraints while minimizing the

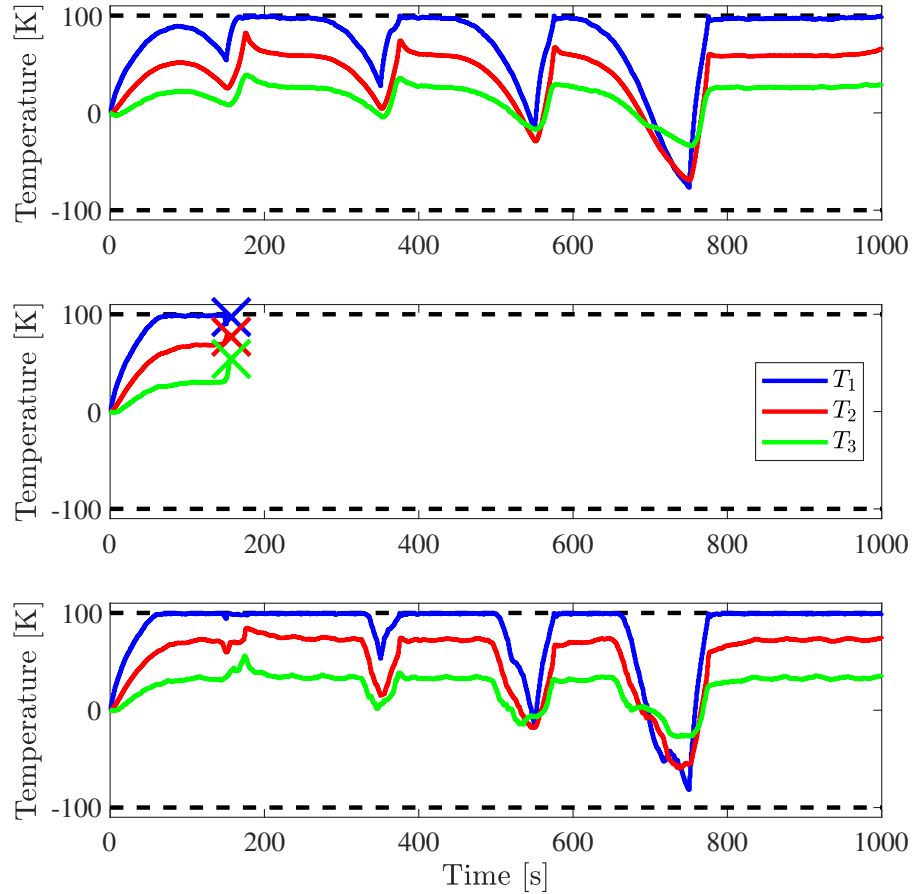


Figure 3.12. Simulation results comparing the shrinking horizon centralized controller (top), receding horizon centralized controller with a short prediction horizon (middle) and a three-level hierarchical controller (bottom).

control inputs to their lower bounds ( $u(k) = -0.25$ ). A quadratic cost function is used as in (3.20) with  $\Lambda = I_n$ .

Fig. 3.12 shows the state trajectories for a three element system ( $n = 3$ ) under the proposed wayset-based hierarchical controller compared to a shrinking horizon centralized controller that predicts to the end of system operation and a receding horizon centralized controller with a short prediction horizon of 5 time steps. For the disturbances shown in Fig. 3.11, the heat pulses are so large that there does not exist a steady state under these heat loads that satisfies the specified output constraints. Thus, the controller needs to pre-cool each thermal element to take advantage of the thermal capacitance in the system. Both

the shrinking horizon controller and the hierarchical controller achieve this pre-cooling effectively, demonstrating that waysets can effectively constrain short-term system operation to guarantee long-term output constraint satisfaction beyond the prediction horizon of the lower-level controllers. Alternatively, the receding horizon centralized controller becomes infeasible at  $t = 157$  seconds where, given the short prediction horizon, the controller cannot pre-cool the system enough prior to the heat load to avoid violating the temperature constraint.

The value of the proposed hierarchical control approach is the scalability with respect to prediction horizon and system order. For the results shown in Fig. 3.12 with  $n = 3$  states and  $k_F = 1000$  steps, the average computation time is 5.96 seconds for the centralized controller while the average computation times, including wayset calculations, for the hierarchical controller  $C_1, C_2$  and  $C_3$  are 0.12, 0.03, and 0.02 seconds, respectively. To demonstrate scalability with respect to prediction horizon, Fig. 3.13 shows average computation times for the centralized and hierarchical controllers for prediction horizons ranging from  $k_F = 200$  to  $k_F = 1000$  steps for a three element system ( $n = 3$ ). As indicated by the dashed line, the centralized controller is able to maintain real-time calculation speed,  $t_{calc} \leq \Delta t$ , for prediction horizons of  $k_F \leq 360$  steps. In these cases, a centralized approach is practical and preferable to the hierarchical controller. However, for longer prediction horizons,  $k_F > 360$ , a centralized approach is no longer viable under the available computational resources, warranting the proposed hierarchical approach with two orders-of-magnitude faster computation times. For these results, only the prediction horizon  $N_1$  of the upper level controller was varied to accommodate the change in  $k_F$ . For the hierarchical MPC formulation, real-time execution only requires  $t_{calc,i} \leq \Delta t_i$  for each controller, and thus the upper-level controllers are allotted more time to solve their optimization problems. Extending the proposed theoretical hierarchical MPC formulation to directly account for computational delay, as in [82], is the focus of future work.

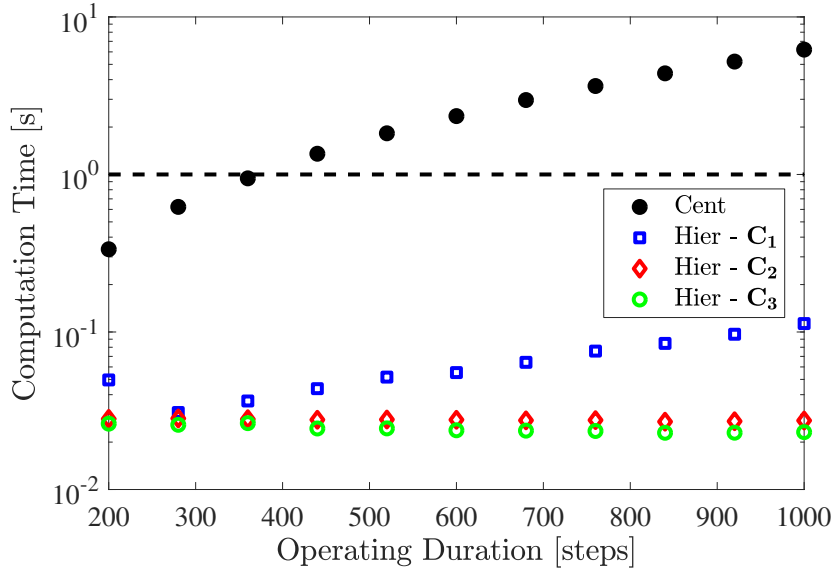


Figure 3.13. Average computation time for the centralized and hierarchical controller as a function of operating duration.

To demonstrate scalability with respect to system order, Fig. 3.14 shows average computation times for the centralized and hierarchical controller for system orders ranging from  $n = 3$  to  $n = 10$  with  $k_F = 360$  steps. While the complexity of the waysets grows linearly with system order, the computation times of the controllers within the hierarchy remain significantly faster than that of the centralized controller. However, for the tenth order system, the waysets used by  $C_2$  in CG-Rep had 800 generators and 720 constraints, thus motivating future research in lower-complexity inner approximations of waysets for improved scalability of the proposed hierarchical approach.

### 3.9 Conclusions

A multi-level vertical hierarchical MPC formulation was presented for constrained linear systems. A robust MPC formulation was used for each controller to guarantee state and input constraint satisfaction in the presence of known and unknown disturbances. Waysets were developed as a novel coordination mechanism between controllers at different levels of the hierarchy. These waysets served as terminal constraints for lower-level controllers, providing

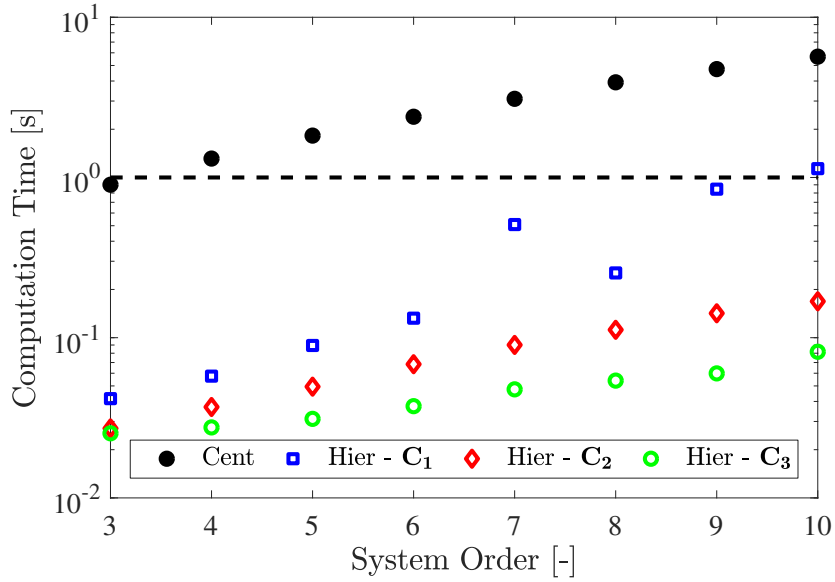


Figure 3.14. Average computation time for the centralized and hierarchical controller as a function of system order.

flexibility in short-term operation of the system while guaranteeing long-term ability to satisfy output and terminal constraints. Using a constrained zonotope representation, waysets were efficiently computed on-line based on the state trajectories determined by upper-level controllers. Numerical examples demonstrated the performance and scalability of the wayset-based hierarchy compared to centralized MPC approaches. Future work will focus on the efficient calculation of lower complexity inner-approximations of the waysets for improved scalability and the extension of this work to hybrid and nonlinear systems. The following chapter will discuss about the development of a hierarchical MPC with coordinating terminal costs for reduced control suboptimality which will be discussed in the following chapter.

# CHAPTER 4

## HIERARCHICAL MPC WITH COORDINATING TERMINAL COSTS <sup>1</sup>

### 4.1 Introduction

Model Predictive Control (MPC) is well-suited for the control of constrained systems since input and state constraints are directly imposed in the underlying optimization problem [62, 60]. However, for systems that require fast control update rates and long prediction horizons, real-time implementation of centralized MPC is hindered by the time required to solve large optimization problems. In such situations, hierarchical MPC reduces computational costs by decomposing control decisions across multiple levels of controllers operating in different timescales [73].

Coordination between controllers at different levels of the hierarchical controller is typically achieved via reference tracking, where state and input trajectories determined by upper-level controllers are communicated down the hierarchy to be tracked by lower-level controllers. Since the closed-loop system behavior is heavily dependent on the weightings used to add reference tracking to the cost function of each controller, guaranteeing constraint satisfaction and control performance is very difficult.

To guarantee input and state constraint satisfaction, coordination mechanisms based on terminal constraints were introduced in [48, 49]. Specifically, *waysets* were defined based on reachability analysis that represent a subset of states at a future point in time from which there exist feasible state and input trajectories for the remainder of system operation. Thus, driving the system states to a wayset provides a short-term control objective that guarantees

---

<sup>1</sup>This chapter is based on work supported by the National Science Foundation under grant 1849500. Chapter 4 in part is a reprint of material published in: © 2020 IEEE. Reprinted, with permission, from Raghuraman, V., Renganathan, V., Summers, T.H., and Koeln, J.P., 2020. Hierarchical MPC with coordinating terminal costs. In 2020 American Control Conference (ACC) (pp. 4126-4133).



long-term constraint satisfaction. While similar wayset-based coordination strategies have been used in [81], those waysets are computed off-line in a feed-forward fashion.

To provide improved control performance in the presence of disturbances, [49] computes waysets on-line based on the optimal state trajectories determined by upper-level controllers. To achieve the computational efficiency required for on-line computation, constrained zonotopes [79] were shown in [48, 49] to provide several orders-of-magnitude reduction in wayset computation time compared to conventional set representations.

While using waysets to achieve guaranteed constraint satisfaction, this research focuses on improving the coordination between controllers within a hierarchy using specifically designed terminal costs to provide closed-loop control performance guarantees. Terminal costs are widely used to guarantee MPC stability by quantifying system operation cost beyond the finite prediction horizon [62]. Within the proposed hierarchical MPC formulation, terminal costs are imposed on the lower-level controller to quantify a specific state transition cost subject to constraints. For a controller with quadratic stage costs, capturing this state transition cost as a function of a terminal state would result in a time-varying piece-wise quadratic cost [11]. However, the present research shows that it is possible to efficiently compute the desired terminal costs on-line in terms of the same constrained zonotope variables used to define the wayset terminal constraint. Thus, the proposed addition of terminal costs to a wayset-based hierarchical MPC controller provides provable control performance guarantees without any additional complexity.

*Contributions:* The contributions of this research are 1) the development of a two-level hierarchical MPC framework with guaranteed constraint satisfaction and control performance, 2) the formulation of terminal costs that allow the lower-level controller to balance both short- and long-term control performance, and 3) the novel representation of the terminal cost using the same variables that define the wayset terminal constraints as constrained zonotopes to achieve computational efficiency.

## 4.2 Notation and Preliminaries

For a discrete time system, the notation  $x(k)$  denotes the state  $x$  at time step  $k$ . For MPC, the double index notation  $x(k+j|k)$  denotes the predicted state at future time  $k+j$  determined at time step  $k$ . The bracket notation  $k \in [0, k_F]$  denotes all integer values of  $k$  from 0 to  $k_F$ . The state trajectory over these time indices is denoted  $\{x(k)\}_{k=0}^{k_F}$ . The set of positive integers is  $\mathbb{Z}_+$ . The weighted norm is defined as  $\|x\|_\Lambda^2 = x^T \Lambda x$ , where  $\Lambda$  is a positive definite diagonal matrix. For sets  $\mathcal{Z}, \mathcal{W} \subset \mathbb{R}^n$ ,  $\mathcal{Y} \subset \mathbb{R}^m$ , and matrix  $R \in \mathbb{R}^{m \times n}$ , the linear transformation of  $\mathcal{Z}$  under  $R$  is  $R\mathcal{Z} = \{Rz \mid z \in \mathcal{Z}\}$ , the Minkowski sum of  $\mathcal{Z}$  and  $\mathcal{W}$  is  $\mathcal{Z} \oplus \mathcal{W} = \{z + w \mid z \in \mathcal{Z}, w \in \mathcal{W}\}$ , and the generalized intersection of  $\mathcal{Z}$  and  $\mathcal{Y}$  under  $R$  is  $\mathcal{Z} \cap_R \mathcal{Y} = \{z \in \mathcal{Z} \mid Rz \in \mathcal{Y}\}$ . The standard intersection, corresponding to the identity matrix  $R = I_n$ , is simply denoted as  $\mathcal{Z} \cap \mathcal{W}$ .

## 4.3 Problem Formulation

As in [48], consider the discrete linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k), \quad (4.1)$$

where  $x \in \mathbb{R}^n$  are the states,  $u \in \mathbb{R}^m$  are the inputs,  $A \in \mathbb{R}^{n \times n}$  is invertible,  $B \in \mathbb{R}^{n \times m}$ , and the pair  $(A, B)$  is stabilizable.

**Assumption 7.** *With a fixed time step  $\Delta t$ , the system operates for a finite length of time starting from  $t = 0$  and ending at  $t = t_F = k_F \Delta t$  with time steps indexed by  $k \in [0, k_F]$ .*

Starting from an initial condition  $x(0)$ , the goal is to plan and execute an input trajectory  $\{u(k)\}_{k=0}^{k_F-1}$  and corresponding state trajectory  $\{x(k)\}_{k=0}^{k_F}$  satisfying the system dynamics from (4.1), the state and input constraints

$$x(k) \in \mathcal{X} \subset \mathbb{R}^n, \quad u(k) \in \mathcal{U} \subset \mathbb{R}^m, \quad \forall k \in [0, k_F - 1], \quad (4.2)$$

and the terminal constraint

$$x(k_F) \in \mathcal{T} \subseteq \mathcal{X}. \quad (4.3)$$

**Assumption 8.** *The sets  $\mathcal{X}, \mathcal{U}, \mathcal{T}$  are compact and convex.*

For notational simplicity, the state and input constraints from (4.2) are represented as the output constraints

$$y(k) \triangleq \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} = Cx(k) + Du(k) \in \mathcal{Y} \triangleq \mathcal{X} \times \mathcal{U}, \quad (4.4)$$

where  $[C \ D] = I_{n+m}$ .

A generic cost function defines the desired system operation using a pre-determined reference trajectory  $\{r(k)\}_{k=0}^{k_F}$  where

$$J^*(x(0)) = \min_{\{u(k)\}_{k=0}^{k_F-1}} \sum_{j=0}^{k_F-1} \ell(j) + \ell_T(k_F), \quad (4.5)$$

with stage costs  $\ell(j) = \ell(x(j), u(j), r(j))$  and terminal cost  $\ell_T(k_F) = \ell_T(x(k_F))$ .

Considering the system (4.1), terminal constraint (4.3), output constraints (4.4), and cost function (4.5), this research extends the wayset-based vertical hierarchical MPC approach developed in [48] to include terminal costs for guaranteed control performance in addition to guaranteed constraint satisfaction.

### 4.3.1 Vertical Hierarchical Control

While [48, 49] provide vertical hierarchical MPC formulations with  $M$  levels of controllers  $\mathbf{C}_i, i \in [1, M]$ , this research will focus on the two-level case,  $M = 2$ , for clarity of exposition. The prediction horizons and time steps for the upper-level controller,  $\mathbf{C}_1$ , and the lower-level controller,  $\mathbf{C}_2$ , satisfy the following assumptions.

**Assumption 9.** *For each controller  $\mathbf{C}_i, i \in [1, 2]$ , the prediction horizon  $N_i \in \mathbb{Z}_+$  and time step  $\Delta t_i > 0$  satisfy*

$$i) \Delta t_2 = \Delta t;$$

$$ii) \Delta t_1 = N_2 \Delta t_2;$$

$$iii) \Delta t_1 = \frac{t_F}{N_1}.$$

These assumptions indicate i) the lower-level controller  $\mathbf{C}_2$  and the system (4.1) have the same time step, ii)  $\mathbf{C}_2$  predicts state and input trajectories between consecutive updates of the upper-level controller  $\mathbf{C}_1$ , and iii)  $\mathbf{C}_1$  predicts to the end of system operation. Additionally, let  $\nu_i \triangleq \frac{\Delta t_i}{\Delta t} \in \mathbb{Z}_+$ , be the time scaling factor where  $\nu_1 = N_2$  and  $\nu_2 = 1$ . The time steps for  $\mathbf{C}_1$  are indexed by  $k_1$ , where  $k_1 \triangleq \frac{k}{\nu_1}$  and the time steps for  $\mathbf{C}_2$  are the same as those for the system,  $k_2 = k$ . Let  $k_{1,F} \triangleq \frac{k_F}{\nu_1}$  such that  $k_1 \in [0, k_{1,F}]$ .

While similar, the optimization problems for each of the two controllers are presented separately to highlight the key differences. Details of these controller formulations are further explained and used to make constraint satisfaction and performance guarantees in Sections 4.3.2 and 4.4.

The upper-level controller  $\mathbf{C}_1$  updates when  $k = \nu_1 k_1$  (i.e. when  $k \bmod \nu_1 = 0$ ), by solving the constrained optimization problem  $\mathbf{P}_1(x(k))$  defined as

$$J_1^*(x(k)) = \min_{\substack{x_1(k_1|k_1) \\ U_1(k_1)}} \sum_{j=k_1}^{k_{1,F}-1} \ell(j|k_1) + \ell_T(k_{1,F}) + \ell_p(k_1), \quad (4.6a)$$

$$\text{s.t. } \forall j \in [k_1, k_{1,F} - 1]$$

$$x_1(j+1|k_1) = A_1 x_1(j|k_1) + B_1 u_1(j|k_1), \quad (4.6b)$$

$$y_1(j|k_1) = C x_1(j|k_1) + D u_1(j|k_1) \in \mathcal{Y}_1, \quad (4.6c)$$

$$x_1(k_{1,F}|k_1) \in \mathcal{T}, \quad (4.6d)$$

$$x_1(k_1|k_1) = x(k) \vee x_1^*(k_1|k_1 - 1). \quad (4.6e)$$

First, note that  $\mathbf{C}_1$  has a *shrinking horizon*, based on the summation limits in (4.6a), since  $\mathbf{C}_1$  predicts to the end of system operation. The input sequence is defined as  $U_1(k_1) = \{u_1(j|k_1)\}_{j=k_1}^{k_{1,F}-1}$ , with the optimal sequence denoted as  $U_1^*(k_1)$ . In (4.6a), the stage cost is a function of states, inputs, and references such that  $\ell(j|k_1) = \ell(x_1(j|k_1), u_1(j|k_1), r_1(j))$ . The penalty cost  $\ell_p(k_1)$  is described in Section 4.5.1 and is used to guarantee the performance of the hierarchical controller. In (4.6b), the model used by  $\mathbf{C}_1$  assumes a piecewise constant control input over the time step  $\Delta t_1$  and thus  $A_1 = A^{\nu_1}$  and  $B_1 = \sum_{j=0}^{\nu_1-1} A^j B$ . In (4.6c), the outputs are constrained to the tightened output constraint set  $\mathcal{Y}_1$ , as discussed in Section 4.3.2. In (4.6d), the terminal state  $x_1(k_{1,F}|k_1)$  is constrained to the terminal set  $\mathcal{T}$  from (4.3). Finally, (4.6e) provides  $\mathbf{C}_1$  the choice of initial condition,  $x_1(k_1|k_1)$ , as either the current state of the system,  $x(k)$ , or the optimal state at this time step determined by  $\mathbf{C}_1$  at the previous time step,  $x_1^*(k_1|k_1 - 1)$ . This choice of initial condition is important to guaranteeing recursive feasibility, as discussed in Section 4.3.2.

The lower-level controller  $\mathbf{C}_2$  updates at every time step of the system, by solving the constrained optimization problem  $\mathbf{P}_2(x(k))$  defined as

$$J_2^*(x(k)) = \min_{U_2(k_2)} \sum_{j=k_2}^{k_2+N_2(k_2)-1} \ell(j|k_2) + \ell_T(k_2 + N_2(k_2)), \quad (4.7a)$$

$$\text{s.t. } \forall j \in [k_2, k_2 + N_2(k_2) - 1]$$

$$x_2(j+1|k_2) = Ax_2(j|k_2) + Bu_2(j|k_2), \quad (4.7b)$$

$$y_2(j|k_2) = Cx_2(j|k_2) + Du_2(j|k_2) \in \mathcal{Y}, \quad (4.7c)$$

$$x_2(k_2 + N_2(k_2)|k_2) \in \mathcal{S}_2(k_2 + N_2(k_2)), \quad (4.7d)$$

$$x_2(k_2|k_2) = x(k). \quad (4.7e)$$

Note that  $\mathbf{C}_2$  has a shrinking and resetting horizon. The prediction horizon length is defined as  $N_2(k_2) \triangleq N_2 - (k_2 \bmod N_2)$  where  $N_2$  satisfies **Assumption 9ii**. This allows  $\mathbf{C}_2$  to predict between the current time step and the time step of the next update of  $\mathbf{C}_1$ ,

at which point  $(k_2 \bmod N_2) = 0$  and prediction horizon resets back to  $N_2(k_2) = N_2$ . The input sequence  $U_2(k_2)$  is defined similarly to  $U_1(k_1)$ . In (4.7a), the stage cost is a function of states, inputs, and references such that  $\ell(j|k_2) = \ell(x_2(j|k_2), u_2(j|k_2), r_2(j))$ . The terminal cost  $\ell_T(k_2 + N_2(k_2))$  is described in Section 4.5.2 and represents operational costs beyond the prediction horizon of  $\mathbf{C}_2$  to improve the performance of the hierarchical controller. In (4.7b), the lower-level controller has an exact model of the system. In (4.7c), the outputs are constrained to the output constraint set  $\mathcal{Y}$  from (4.4). In (4.7d), the time-varying terminal constraint corresponds to the waysets used to coordinate between controllers  $\mathbf{C}_2$  and  $\mathbf{C}_1$ . Finally, (4.7e) defines the initial condition as the current state of the system.

The two-level hierarchical controller is implemented throughout system operation based on **Algorithm 5**.

---

**Algorithm 5:** Two-level hierarchical MPC.

---

```

1 initialize  $k, k_1, k_2 \leftarrow 0$ 
2 while  $k < k_F$  do
3   if  $k \bmod \nu_1 = 0$  then
4     calculate  $\ell_p(k_1)$ ;
5     solve  $\mathbf{P}_1(x(k))$ ;
6     calculate  $\ell_T(k_2 + N_2(k_2))$ ,  $\mathcal{S}_2(k_2 + N_2(k_2))$  and communicate to  $\mathbf{P}_2(x(k))$ ;
7      $k_1 \leftarrow k_1 + 1$ ;
8   end
9   solve  $\mathbf{P}_2(x(k))$  for  $U_2^*(k_2)$  and apply the first input in the sequence,  $u_2^*(k_2|k_2)$ , to
   the system;
10   $k_2 \leftarrow k_2 + 1$ ;
11   $k \leftarrow k + 1$ ;
12 end

```

---

### 4.3.2 Constraint Satisfaction

As discussed in detail in [48], the controller formulations (4.6) and (4.7) are specifically designed to guarantee recursive feasibility for  $\mathbf{P}_1(x(k))$  and  $\mathbf{P}_2(x(k))$  when implementing the hierarchical controller using **Algorithm 5**. Furthermore, recursive feasibility guarantees

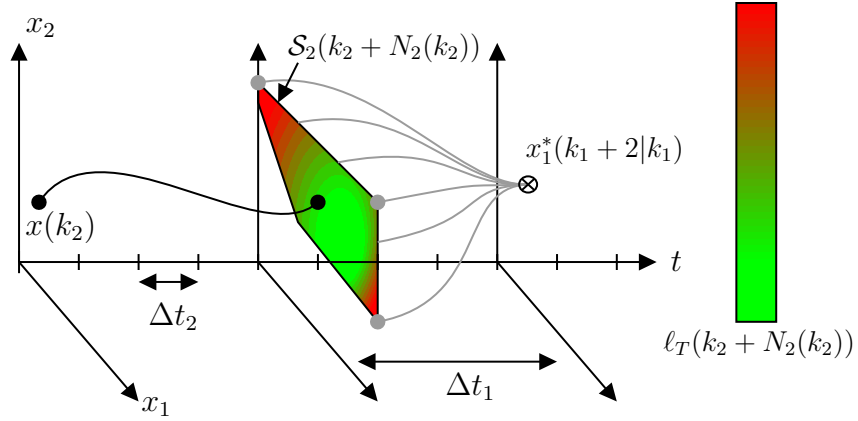


Figure 4.1. Notional example of the combined use of waysets and terminal costs for coordination between controllers at different levels of the hierarchy.

the satisfaction of output constraints (4.4) and terminal constraints (4.3). This constraint satisfaction guarantee relies on i) output constraint tightening, ii) wayset formulation, and iii) initial condition options. With additional details provided in [48], these three features are summarized as follows.

While the lower-level controller  $\mathbf{C}_2$  is formulated with the original output constraint set  $\mathcal{Y}$ , the upper-level controller  $\mathbf{C}_1$  requires the tightened output constraint set  $\mathcal{Y}_1 \subseteq \mathcal{Y}$ . This constraint tightening prevents  $\mathbf{C}_1$  from computing state trajectories that violate state constraints during the inter-sample time steps between the slow updates of  $\mathbf{C}_1$ . As a result, any optimal state and input trajectory determined by  $\mathbf{C}_1$  is a feasible solution for  $\mathbf{C}_2$ .

As shown in Fig. 4.1, the wayset  $\mathcal{S}_2(k_2 + N_2(k_2))$ , used as a terminal constraint in the formulation of  $\mathbf{C}_2$ , represents a backward reach set starting from the point  $x_1^*(k_1 + 2|k_1)$  along the optimal state trajectory determined by  $\mathbf{C}_1$ . Thus, for any state within this wayset, there exists feasible state and input trajectories that steer the system back to the trajectory determined by  $\mathbf{C}_1$ . The wayset provides  $\mathbf{C}_2$  the flexibility to deviate from the trajectories determined by  $\mathbf{C}_1$ . This flexibility allows  $\mathbf{C}_2$  to further minimize short-term operational cost, by using its faster update rate, while guaranteeing long-term constraint satisfaction. **Algorithm 6** outlines the backward reach set computation used to define the wayset.

---

**Algorithm 6:** Wayset  $\mathcal{S}_2(k_2 + N_2(k_2))$  computation at time step  $k = \nu_1 k_1$ .

---

```

1 initialize  $j \leftarrow N_2$ 
2 if  $k_1 \geq k_{1,F} - 1$  then
3   |  $\mathcal{S}_2(k_2 + N_2(k_2)) = \mathcal{T}$ ;
4 else
5   |  $\mathcal{S}_2(j) = x_1^*(k_1 + 2|k_1)$ ;
6   | while  $j \geq 1$  do
7     |  $\tilde{\mathcal{S}}_2(j-1) = A^{-1}\mathcal{S}_2(j) \oplus (-A^{-1}B)\mathcal{U}$ ;
8     |  $\mathcal{S}_2(j-1) = \tilde{\mathcal{S}}_2(j-1) \cap \mathcal{X}$ ;
9     |  $j \leftarrow j - 1$ ;
10  | end
11  |  $\mathcal{S}_2(k_2 + N_2(k_2)) = \mathcal{S}_2(j)$ 
12 end

```

---

Since the wayset  $\mathcal{S}_2(k_2 + N_2(k_2))$  is time-varying and depends on the state  $x_1^*(k_1 + 2|k_1)$ , each wayset needs to be computed on-line immediately following each update of  $\mathbf{C}_1$ . Constrained zonotopes [79] are used to provide the computational efficiency necessary to perform these set computations on-line. Assuming zonotopic input and state sets,

$$\mathcal{X} = \{G_x \xi_x + c_x \mid \|\xi_x\|_\infty \leq 1\},$$

$$\mathcal{U} = \{G_u \xi_u + c_u \mid \|\xi_u\|_\infty \leq 1\},$$

where  $G_x \in \mathbb{R}^{n \times n}$ ,  $G_u \in \mathbb{R}^{m \times m}$ ,  $c_x \in \mathbb{R}^n$ , and  $c_u \in \mathbb{R}^m$ . By applying **Algorithm 6** with Minkowski sum and intersection operations defined in [79], the wayset is a constrained zonotope such that

$$\mathcal{S}_2(k_2 + N_2(k_2)) = \{G\xi + c \mid \|\xi\|_\infty \leq 1, A\xi = b\}, \quad (4.8)$$

where  $G \in \mathbb{R}^{n \times (n+m)N_2}$ ,  $c \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{nN_2 \times (n+m)N_2}$  and  $b \in \mathbb{R}^{nN_2}$ .

Finally, the initial condition option (4.6e) is used to ensure recursive feasibility of  $\mathbf{C}_1$ . This initial condition option is similar to that used in robust MPC [38]. Due to the use of waysets,  $\mathbf{C}_2$  may drive the system to a point in the wayset that is not a feasible initial condition for  $\mathbf{C}_1$ . Thus  $\mathbf{C}_1$  is given the option to start its state trajectory from the current



state of the system, if a feasible solution exists, or from the state along the optimal trajectory determined by  $\mathbf{C}_1$  at the previous time step. While this first option is not guaranteed to be feasible, the second option always provides a feasible solution to both  $\mathbf{C}_1$  and  $\mathbf{C}_2$  [48].

Using output constraint tightening, waysets, and the initial condition option, the following theorem from [48] establishes guaranteed constraint satisfaction for the hierarchical controller.

**Theorem 9.** [48] *Following **Algorithm 5** for a two-level hierarchy, both the upper- and lower-level controller problems,  $\mathbf{P}_1(x(k))$  and  $\mathbf{P}_2(x(k))$ , are feasible when solved at  $k \bmod \nu_i = 0$ , resulting in system state and input trajectories satisfying constraints (4.2) and (4.3).*

*Proof.* See Appendix. □

#### 4.4 Control Performance

With constraint satisfaction established, this section presents the main results of this research by focusing on the control performance of the hierarchical controller.

In this research, control performance is quantified by the cost function (4.5), which captures the sum of system operation costs at each discrete time step starting from  $t = 0$  and ending at  $t = t_F = k_F \Delta t$ . With a time step of  $\Delta t$  and an initial prediction horizon of  $N = k_F$ , a centralized MPC approach produces the optimal (minimal) cost  $J^*(x(0))$ , as defined in (4.5). Since this research assumes an exact model of the system (4.1) without disturbances and finite operation, the optimal solution determined by the centralized MPC controller any time  $k \in [0, k_F - 1]$  is simply the tail of the optimal solution determined at time  $k = 0$  [47].

With the optimal system operation cost denoted  $J^*(x(0))$ , any other control formulation results in a cost  $J(x(0)) \geq J^*(x(0))$ . First, consider the case where only the upper-level

controller of the hierarchy is applied to the system. In this case, the inputs determined by  $\mathbf{C}_1$  are applied directly to the system with a slow update period of  $\Delta t_1 = N_2 \Delta t$ . With an initial prediction horizon of  $N_1 = k_{1,F}$ , the minimal operational cost for this controller is denoted as  $J^{up}(x(0))$ , which is the sum of the  $N_1 - 1$  coarse stage costs where

$$J^{up}(x(0)) = \sum_{k_1=0}^{N_1-1} J^{up}(k_1|k_1).$$

The double index notation  $J^{up}(k_1|k_1)$  is used to denote the optimal operation cost at coarse time step  $k_1$  determined by  $\mathbf{C}_1$  at time  $k_1$ . As with the centralized MPC controller, if only the upper-level controller is applied directly to the system, then the optimal solution for  $\mathbf{C}_1$  at any time step  $k_1 \in [0, N_1 - 1]$  is the tail of the optimal solution determined at time  $k = 0$ . Therefore,  $J^{up}(k_1|k_1) = J^{up}(k_1|0)$ ,  $\forall k_1$ .

For a hierarchical controller with multiple levels, it is natural to expect the lower-level controllers to further reduce the total operation cost. Thus, denoting the minimal operating cost for the hierarchical controller as  $J^h(x(0))$ , it is expected that

$$J^*(x(0)) \leq J^h(x(0)) \leq J^{up}(x(0)). \quad (4.9)$$

However guaranteeing (4.9) requires effective coordination between the controller at different levels of the hierarchy.

An initial coordination strategy introduced in [47] utilized the notion of *waypoints*. A waypoint corresponds to a point along the optimal state trajectory determined by  $\mathbf{C}_1$ . This waypoint is treated as a terminal constraint in the formulation of the optimization problem for  $\mathbf{C}_2$ . Therefore, the lower-level controller has the flexibility to further improve system performance over the fast time steps between slow updates of the upper-level controller. Due to the waypoint constraint, operating costs for the waypoint-based hierarchy and the upper-level only controller can be directly compared at each coarse time step, where

$$J^h(k_1|k_1) \leq J^{up}(k_1|k_1), \quad \forall k_1 \in [0, N_1 - 1].$$

Therefore,  $J^h(x(0)) \leq J^{up}(x(0))$ , which guarantees that the lower-level controller can only help to improve the control performance of the hierarchy.

An improved coordination strategy introduced in [48] expanded the idea of coordination via terminal constraints through the use of *waysets*. As shown in Fig. 4.1, a wayset represents a backward-reachable set from a point along the optimal state trajectory determined by  $\mathbf{C}_1$ . Waysets provide even greater flexibility to the lower-level controller while still guaranteeing controller feasibility and system constraint satisfaction. However, this additional flexibility introduces the potential for greedy behavior, where the  $\mathbf{C}_2$  minimizes its own cost function over its short horizon while unknowingly increasing the long-term operational cost beyond its prediction horizon. This greedy behavior could lead to an increase in total operation cost where  $J^h(x(0)) \geq J^{up}(x(0))$ . In this case, the lower-level controller actually degrades the control performance. In Section 4.6, a numerical example shows how a wayset-based hierarchy can greedily utilize a finite resource too quickly, leading to significant performance degradation during later system operation.

By imposing terminal costs, denoted as  $\ell_T(k_2 + N_2(k_2))$  for the formulation of  $\mathbf{C}_2$  in (4.7), the lower-level controller can only improve control performance, resulting in  $J^h(x(0)) \leq J^{up}(x(0))$ . As shown in Fig. 4.1, this terminal cost represents the constrained state transition cost from the terminal state  $x_2(k_2 + N_2(k_2)|k_2)$  to the optimal state  $x_1^*(k_1 + 2|k_1)$  determined by the upper-level controller. Note that this optimal state is exactly the state used to define the wayset in **Algorithm 5**. Therefore, while the wayset constraint (4.7d) guarantees that there is a feasible trajectory from  $x_2(k_2 + N_2(k_2)|k_2)$  to  $x_1^*(k_1 + 2|k_1)$ , the terminal cost now represents the exact operational cost for this state transition. Using this terminal cost at coarse time step  $k_1$ ,  $\mathbf{C}_2$  now minimizes the cost function  $J^h(k_1|k_1) + J^h(k_1 + 1|k_1)$ , effectively doubling the prediction horizon of the lower-level controller.

While Section 4.5 demonstrates how to compute the terminal cost  $\ell_T(k_2 + N_2(k_2))$ , the following theorem states that using terminal costs results in a lower-level controller that can only improve the performance of the system.

**Theorem 10.** *Following **Algorithm 5** for a two-level hierarchy with penalty cost  $\ell_p(k_1)$  and terminal cost  $\ell_T(k_2 + N_2(k_2))$  as computed in Section 4.5, the hierarchical controller results in a reduced operational cost compared to only applying the upper-level controller such that  $J^h(x(0)) \leq J^{up}(x(0))$ .*

*Proof.* See Appendix. □

## 4.5 Terminal Cost Computation

### 4.5.1 Upper-level Controller

From (4.6a), the cost function for the upper-level controller consists of stage costs, a terminal cost, and a penalty cost. Since  $\mathbf{C}_1$  predicts to the end of system operation, the terminal cost is the same as that used in (4.5). The penalty cost is used to ensure that the hierarchical controller provides improved control performance as stated in **Theorem 10**. Specifically,

$$\ell_p(k_1) = \begin{cases} 0 & \text{if } x_1(k_1|k_1) = x_1^*(k_1|k_1 - 1), \\ \Delta J(k_1 - 1|k_1 - 1) & \text{if } x_1(k_1|k_1) = x(k), \end{cases}$$

where  $\Delta J(j|j) \triangleq \max(0, J^h(j|j) - J^{up}(j|j))$ . Therefore,  $\ell_p(k_1)$  penalizes  $\mathbf{C}_1$  only when using the current state as its initial condition and this penalty is based on the difference between operational costs determined by  $\mathbf{C}_2$  and  $\mathbf{C}_1$ . If  $\mathbf{C}_2$  chose a higher operating cost at the previous coarse time step  $k_1 - 1$ , then  $\Delta J(k_1 - 1|k_1 - 1) \geq 0$ . This formulation of the penalty cost ensures that if  $\mathbf{C}_1$  chooses to start at  $x(k)$  then the corresponding optimal trajectory has an operating cost at least  $\Delta J(k_1 - 1|k_1 - 1)$  less than the optimal trajectory starting at  $x_1^*(k_1|k_1 - 1)$ . This property is used in the proof of **Theorem 10** in the Appendix.

### 4.5.2 Lower-level Controller

From (4.7a), the cost function for the lower-level controller consists of stage costs and a time-varying terminal cost. Traditionally, terminal costs in MPC are formulated as a function

of the terminal state  $x_2(k_2 + N_2(k_2)|k_2)$ . However, the proposed terminal cost represents the state transition cost from  $x_2(k_2 + N_2(k_2)|k_2)$  to  $x_1^*(k_1 + 2|k_1)$  subject to linear output constraints (4.4). If the cost function in (4.5) is quadratic, the resulting terminal cost would be piecewise-quadratic, a known result from the field of explicit MPC [11]. However, this cost is dependent on  $x_1^*(k_1 + 2|k_1)$  as well as references  $r(j)$ . Therefore, the terminal cost is time-varying and would be very difficult to parameterize with respect to both  $x_1^*(k_1 + 2|k_1)$  and references  $r(j)$ . For these reasons, formulating the terminal cost as a function of the terminal state is impractical.

Alternatively, the terminal cost can be formulated in terms of  $\xi$ , the variables used to define the wayset as a constrained zonotope in (4.8). First note, if  $k_2 + N_2(k_2) = k_F$ ,  $\mathbf{C}_2$  predicts to the end of system operation and therefore  $\ell_T(k_2 + N_2(k_2)) = \ell_T(k_F)$ . Otherwise, for all  $k_2 + N_2(k_2) < k_F$ ,

$$\ell_T(k_2 + N_2(k_2)) = \sum_{j=k_2+N_2(k_2)}^{k_2+2N_2(k_2)-1} \ell(j|k_2). \quad (4.10)$$

If the stage costs are in the form of a weighted quadratic function, then let

$$\ell(j|k_2) = \|r(j) - z(j|k_2)\|_{\Lambda}^2, \quad (4.11)$$

where  $z(j|k_2) = Ex(j|k_2) + Fu(j|k_2)$ .

**Theorem 11.** *The terminal cost (4.10) with the quadratic stage costs (4.11) is the constrained state transition cost from  $x_2(k_2 + N_2(k_2)|k_2)$  to  $x_1^*(k_1 + 2|k_1)$  and can be exactly represented as a quadratic function of  $\xi$  from (4.8) where*

$$\ell_T(k_2 + N_2(k_2)) = \xi^T P_T \xi + 2q_T \xi + r_T, \quad (4.12)$$

and  $P_T$  is time-invariant while  $q_T$  and  $r_T$  are time-varying due to dependence on references  $r(j)$  and state  $x_1^*(k_1 + 2|k_1)$ .

*Proof.* See Appendix. □

## 4.6 Numerical Example

To demonstrate the benefits of using terminal costs for hierarchical MPC coordination, this section develops a two-level hierarchy with terminal costs for the simplified vehicle system from [48]. The system model is

$$x(k+1) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix} u(k),$$

where the three states represent position, velocity, and on-board stored energy, the three inputs represent acceleration, deceleration, and power to an on-board load, all of which deplete the stored energy. The system and lower-level controller have time step sizes of  $\Delta t = \Delta t_2 = 1$  second. Finite operation is defined for  $k_F = 200$  seconds. Choosing  $\Delta t_1 = 10$  seconds results in  $\nu_1 = 10$  and prediction horizons of  $N_1 = 20$  and  $N_2 = 10$  steps.

The desired operation, defined by  $\{r(k)\}_{k=0}^{k_F}$ , is shown in Fig. 4.2 for the first state (position), and third input (load power). References for the first and second inputs (acceleration and deceleration) are 0 for the entire mission, and are not shown in Fig. 4.2. These references are used to define (4.5) as the weighted quadratic cost function from (4.11) where  $z(j) = \begin{bmatrix} [1 \ 0 \ 0] x(j) \\ u(j) \end{bmatrix}$  and  $\Lambda = \text{diag} \left( \begin{bmatrix} 10^{-2} & 10^0 & 10^0 & 10^2 \end{bmatrix} \right)$ .

Given an initial on-board stored energy,  $E(0)$ , the output and terminal constraints defining  $\mathcal{Y}$  and  $\mathcal{T}$  are

$$\begin{bmatrix} -1 \\ -20 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq y(k) \leq \begin{bmatrix} 105 \\ 20 \\ E(0) \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} \leq x(k_F) \leq \begin{bmatrix} 1 \\ 1 \\ E(0) \end{bmatrix}.$$

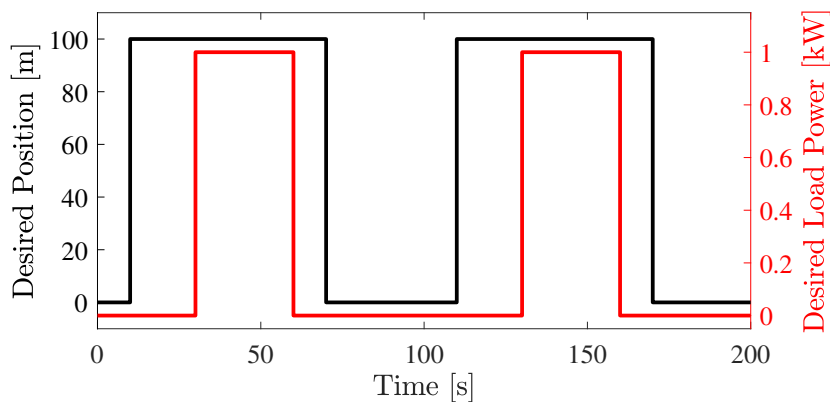


Figure 4.2. References for position and load power.

For two different initial conditions,  $x(0) = [0 \ 0 \ 150]^T$  and  $x(0) = [0 \ 0 \ 100]^T$ , Figs. 4.3 and 4.4 show simulation results using the proposed hierarchical MPC with terminal costs (Hier-T) compared to centralized MPC (Cent), hierarchical MPC without terminal costs (Hier-NT), and hierarchical MPC with only the upper-level controller (Hier-Up). The three subplots show the position reference tracking, the depletion of on-board energy, and the load power reference tracking.

First, note that all controllers satisfy the output and terminal constraints constraints. For the hierarchical controllers, this is achieved using constraint tightening and wayset terminal constraints as discussed in Section 4.3.2. While Figs. 4.3 and 4.4 show that the four controllers result in qualitatively similar performance, the quantitative differences in control performance are clearly shown in Fig. 4.5. For both initial conditions, the centralized controller provides the lowest cost (best performance), as expected. Guaranteed by **Theorem 10**, the hierarchy with terminal costs results in an operating cost less than the upper-level only controller. The performance of the wayset-based hierarchical controller without terminal costs is significantly different for the two different initial conditions. When viewed as a resource distribution problem, an initial on-board stored energy of  $E(0) = 150\text{kJ}$  represents a case where there is enough energy to operate the system as desired. In this case, there is enough energy to support the short-sighted, greedy behavior of a hierarchy without terminal

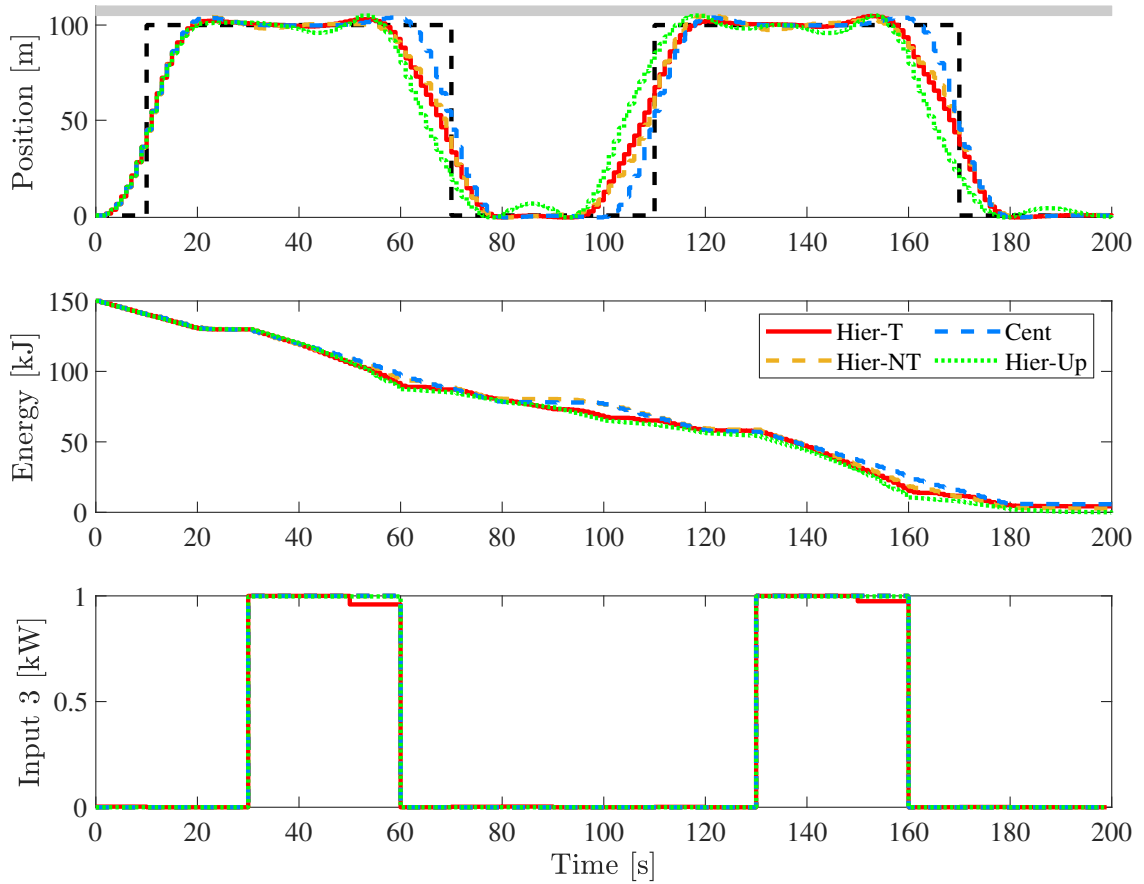


Figure 4.3. Simulation results for initial condition  $x(0) = [0 \ 0 \ 150]^T$  comparing the proposed hierarchical MPC with terminal costs (Hier-T) to centralized MPC (Cent), hierarchical MPC without terminal costs (Hier-NT), and hierarchical MPC with only the upper-level controller (Hier-Up).

costs. However, if the initial on-board stored energy is reduced to  $E(0) = 100\text{kJ}$ , there is insufficient energy to operate as desired and an intelligent controller must ration this resource throughout operation. Now, the greedy behavior of the lower-level controller results in an operating cost greater than if only the upper-level controller was used. Figs. 4.3 and 4.4 show that the majority of this increase in operating costs comes from the inability to closely track the load power reference trajectory (input 3).

Using a desktop computer with a 3.2 Ghz i7 processor and 16 GB of RAM, all controllers were formulated and solved with YALMIP [57] and Gurobi [34]. While the addition of ter-



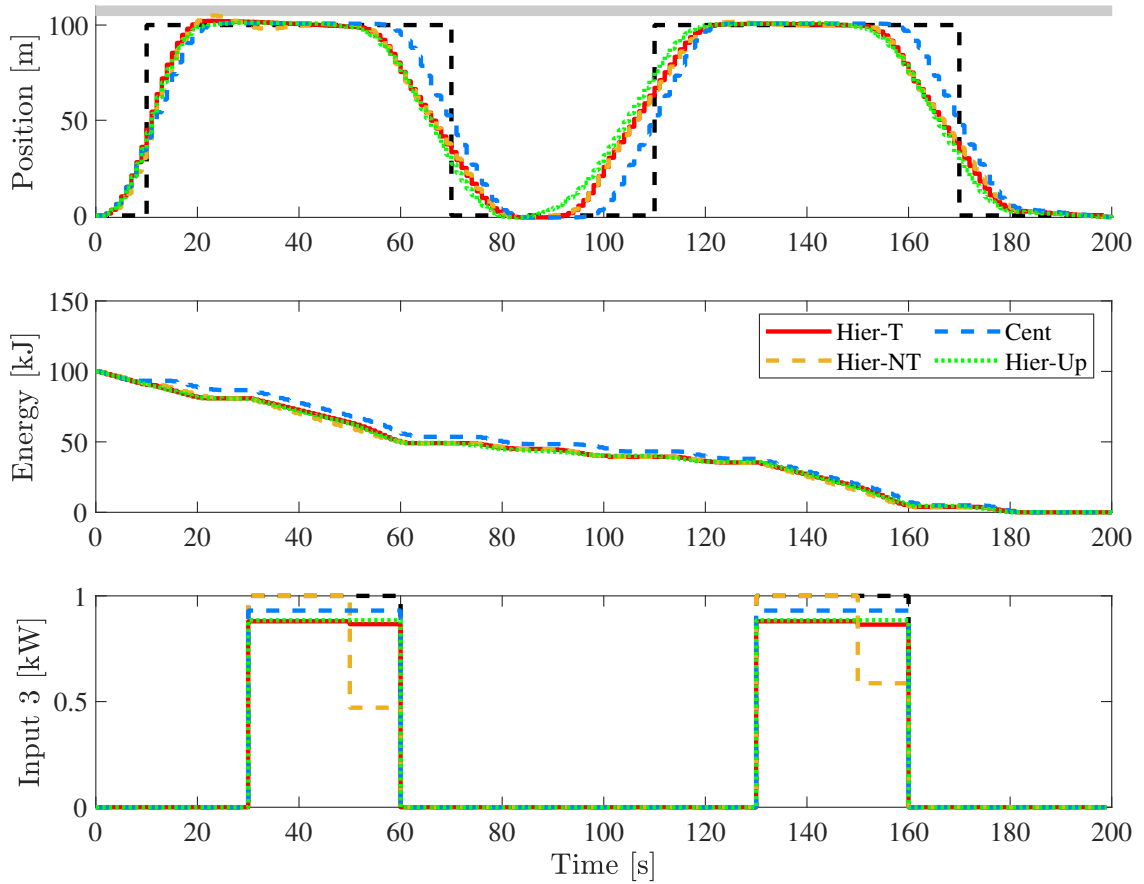


Figure 4.4. Simulation results for initial condition  $x(0) = [0 \ 0 \ 100]^T$  comparing the proposed hierarchical MPC with terminal costs (Hier-T) to centralized MPC (Cent), hierarchical MPC without terminal costs (Hier-NT), and hierarchical MPC with only the upper-level controller (Hier-Up).

terminal costs does not increase the number of decision variables for wayset-based hierarchical MPC, the simulation results shows a modest increase in average computation time for the lower-level controller from  $\Delta t_{calc} = 0.066$  seconds without terminal costs to  $\Delta t_{calc} = 0.106$  seconds. This is likely due to how the time-varying  $q_T$  and  $r_T$  terms from (4.12) are implemented in YALMIP. For systems with long prediction horizons and a large number of states and inputs, the number of  $\xi$  variables required to represent the waysets and terminal costs might pose challenges to real-time control execution. Therefore, future work will explore the

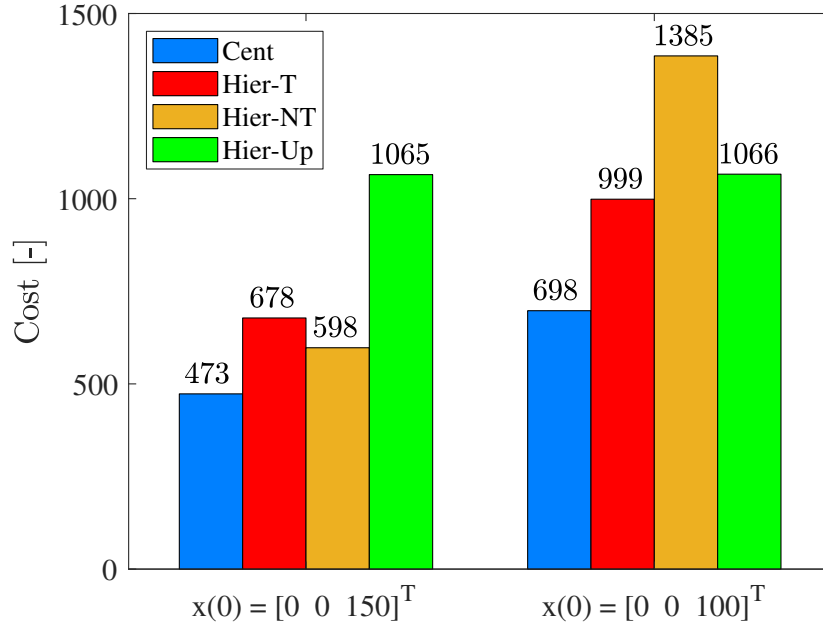


Figure 4.5. Comparison of system operation cost for both initial conditions and each of the four controllers.

use of reduced-order inner-approximations of waysets and the corresponding approximations of terminal costs to provide greater scalability of the proposed approach.

#### 4.7 Conclusions

A two-level hierarchical MPC formulation was presented with coordination between controllers through the use of both terminal constraints and terminal costs. Wayset-based terminal constraints guaranteed constraint satisfaction while terminal costs guaranteed hierarchical control performance. The terminal costs were specifically formulated to balance both short- and long-term control performance without the need for controller tuning. As a result, the hierarchical controller was proven to provide better control performance compared to only applying the upper-level controller. A numerical example demonstrated the merits of including of terminal costs as a coordination mechanism for hierarchical MPC. Future work will focus on the efficient calculation of lower complexity inner-approximations of the waysets and corresponding terminal cost approximations for improved scalability. The following

chapter will discuss about the development of a tube-based robust MPC with adjustable uncertainty set optimization using zonotopes.

## 4.8 Appendix

*Proof of Theorem 10.* Achieving the desired reduction in operational cost is equivalent to

$$\sum_{k_1=0}^{N_1-1} J^h(k_1|k_1) \leq \sum_{k_1=0}^{N_1-1} J^{up}(k_1|0), \quad (4.13)$$

where the left-hand side represent the operational cost of the two-level hierarchy as the sum of operational costs over every coarse time step.

As shown in Fig. 4.6a, at time step  $k_1 = 0$ ,  $\mathbf{C}_2$  plans a trajectory such that the total operational cost satisfies

$$J^h(0|0) + J^h(1|0) + \sum_{j=2}^{N_1-1} J^{up}(j|0) \leq \sum_{j=0}^{N_1-1} J^{up}(j|0), \quad (4.14)$$

where  $J^h(0|0)$  and  $J^h(1|0)$  are the stage and terminal costs for  $\mathbf{C}_2$ . Note that (4.14) holds since the trajectory determined by  $\mathbf{C}_1$  is always a feasible trajectory for  $\mathbf{C}_2$ .

At each time step  $k_1 \in [1, N_1 - 2]$ , the upper-level controller has a choice of initial condition from (4.6e) such that  $x_1(k_1|k_1) = x(k) \vee x_1^*(k_1|k_1 - 1)$ . As shown in Fig. 4.6b, if  $\mathbf{C}_1$  chooses  $x_1(k_1|k_1) = x_1^*(k_1|k_1 - 1)$ , then there exists a feasible trajectory for  $\mathbf{C}_2$  such that

$$J^h(k_1|k_1) + J^h(k_1 + 1|k_1) + \sum_{j=k_1+2}^{N_1-1} J^{up}(j|k_1) \leq J^h(k_1|k_1 - 1) + \sum_{j=k_1+1}^{N_1-1} J^{up}(j|k_1 - 1). \quad (4.15)$$

Due to the penalty cost imposed on  $\mathbf{C}_1$  from Section 4.5.1, if  $\mathbf{C}_1$  chooses  $x_1(k_1|k_1) = x(k)$ , then there exists a feasible trajectory for  $\mathbf{C}_2$ , as shown in Fig. 4.6c, such that

$$J^h(k_1|k_1) + J^h(k_1 + 1|k_1) + \sum_{j=k_1+2}^{N_1-1} J^{up}(j|k_1) \leq \sum_{j=k_1}^{N_1-1} J^{up}(j|k_1 - 1) - \Delta J(k_1 - 1|k_1 - 1). \quad (4.16)$$

At the time step  $k_1 = N_1 - 1$ , there exists a feasible trajectory  $\mathbf{C}_2$ , as shown in Fig. 4.6d, such that

$$J^h(N_1 - 1|N_1 - 1) \leq J^h(N_1 - 1|N_1 - 2). \quad (4.17)$$

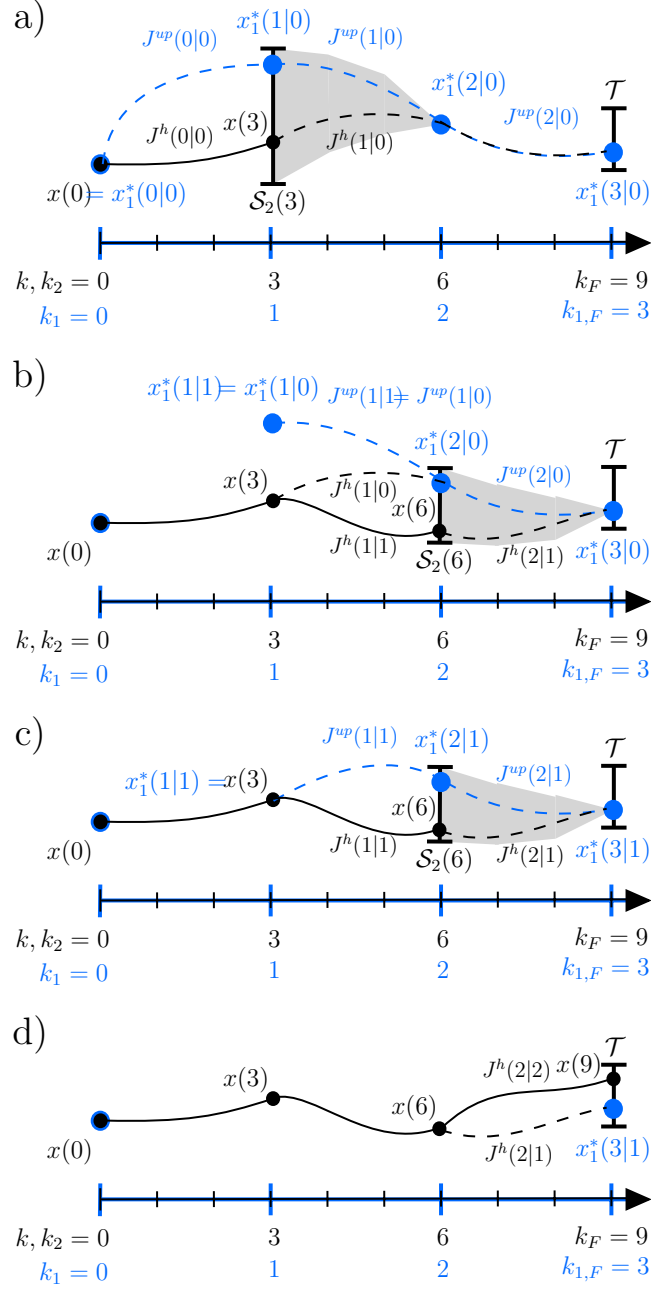


Figure 4.6. Notional state trajectories used to demonstrate the operating cost relationships among feasible trajectories.

From the definition of  $\Delta J(j|j)$  and combination of (4.14)-(4.16), it can be shown that there exists feasible trajectories at each time step  $k_1 \in [1, N_1 - 2]$  such that

$$\sum_{j=0}^{k_1} J^h(j|j) + J^h(k_1 + 1|k_1) + \sum_{j=k_1+2}^{N_1-1} J^{up}(j|k_1) \leq \sum_{j=0}^{N_1-1} J^{up}(j|0). \quad (4.18)$$

Adding (4.18) for  $k_1 = N_1 - 2$  and (4.17) results in the desired operational cost relationship from (4.13), proving the theorem.

*Proof of **Theorem 11**.* Let  $R$ ,  $Z$ , and  $U$  denote trajectories such that, for  $k = k_2 + N_2$ ,

$$\begin{aligned} R &= \begin{bmatrix} r^T(k + N_2 - 1) & r^T(k + N_2 - 2) & \dots & r^T(k) \end{bmatrix}^T, \\ Z &= \begin{bmatrix} z^T(k + N_2 - 1) & z^T(k + N_2 - 2) & \dots & z^T(k) \end{bmatrix}^T, \\ U &= \begin{bmatrix} u^T(k + N_2 - 1) & u^T(k + N_2 - 2) & \dots & u^T(k) \end{bmatrix}^T. \end{aligned}$$

Given (4.11), (4.10) can now be re-written as

$$\ell_T(k_2 + N_2(k_2)) = (R - Z)^T \hat{\Lambda} (R - Z),$$

where  $\hat{\Lambda}$  is block diagonal with  $\Lambda$  repeated  $N_2$  times. There exist matrices  $\hat{A}, \hat{B}$  where  $Z = \hat{A}x_1^*(k_1 + 2|k_1) + \hat{B}U$ . Furthermore, with  $\xi$  from (4.8),  $U = \hat{c}_u + \hat{G}_u \hat{T} \xi$  where

$$\hat{c}_u = \begin{bmatrix} c_u \\ c_u \\ \vdots \\ c_u \end{bmatrix}, \hat{G}_u = \begin{bmatrix} G_u & 0 & \dots & 0 \\ 0 & G_u & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & G_u \end{bmatrix},$$

and

$$\hat{T} = \begin{bmatrix} [I \ 0] & 0 & \dots & 0 \\ 0 & [I \ 0] & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & [I \ 0] \end{bmatrix}.$$

Now, as in (4.12), the terminal cost is quadratic in  $\xi$  where

$$\begin{aligned} P_T &= (\hat{B} \hat{G}_u \hat{T})^T \hat{\Lambda} (\hat{B} \hat{G}_u \hat{T}), \\ q_T &= (\hat{A}x_1^*(k_1 + 2|k_1) + \hat{B} \hat{c}_u - R)^T \hat{\Lambda} (\hat{B} \hat{G}_u \hat{T}), \\ r_t &= \|\hat{A}x_1^*(k_1 + 2|k_1) + \hat{B} \hat{c}_u - R\|_{\hat{\Lambda}}^2. \end{aligned}$$

Note that both  $q_T$  and  $r_T$  are time-varying due to their dependence on references  $R$  and state  $x_1^*(k_1 + 2|k_1)$  while  $P_T$  is time-invariant.

**CHAPTER 5**  
**TUBE-BASED ROBUST MPC WITH ADJUSTABLE**  
**UNCERTAINTY SETS USING ZONOTOPES**<sup>1</sup>

## 5.1 Introduction

Model Predictive Control (MPC) is widely used for the control of constrained systems where state and input constraints are explicitly embedded in the optimization problem. For the control of systems with unknown but bounded additive disturbances, there exists many different robust MPC formulations designed to guarantee robust state and input constraint satisfaction in the presence of these uncertainties [62, 10]. In these formulations, the bounded disturbances are typically restricted to a uncertainty set of predetermined shape and size.

A specific form of robust MPC known as *tube-based* robust MPC is widely used to provide robustness with only a slight increase in computational complexity [54, 60]. Tubes bound the true system trajectories within a neighborhood centered around nominal state and input trajectories optimized by the robust MPC controller. The sizes of these tubes depend on the size of the bounded uncertainty set and are typically computed using a Robust Positively Invariant (RPI) set. This RPI set is then used to tighten the state and input constraint sets such that if nominal trajectories satisfy the tightened constraints, then the true system trajectories are guaranteed to satisfy the original constraints. Typically, the size of the uncertainty set is known *a priori* and both the RPI set and the constraint tightening are computed offline.

Recently, there has been a growing interest to compute the uncertainty set as a part of the robust MPC optimization problem and these problems are referred to as robust MPC

---

<sup>1</sup>This chapter is based on work supported by the National Science Foundation under grant 1849500. Chapter 5 in part is a reprint of material published in: © 2021 IEEE. Reprinted, with permission, from Raghuraman, V. and Koeln, J.P., 2021. Tube-based robust MPC with adjustable uncertainty sets using zonotopes. In 2021 American Control Conference (ACC) (pp. 462-469).

with adjustable uncertainty sets (RMPC-AU) [94, 44]. Applications that can benefit from RMPC-AU formulation include economic reserve capacity optimization [95], robust input tracking [87, 31], distributed MPC [85], and hierarchical MPC [49]. In the reserve capacity problem, the adjustable uncertainty set is characterized as a reserve capacity which can be provided by an operator to third-parties for some monetary benefits without violating its own operational constraints [95]. While in robust input tracking from [87] and [31], the adjustable uncertainty set is quantified as the largest input set that can be tracked without violating the system constraints. In these applications, maximization of uncertainty sets is desired.

However, in distributed MPC, where the coupling between neighboring subsystems is treated as a bounded disturbance, minimizing the size of disturbance sets provides reduced conservatism and improved performance [85]. Finally, in hierarchical MPC, differences between control decisions by controllers at different levels of the hierarchy can be viewed as bounded disturbances [49]. In this case, the optimal size of bounding sets is to be determined while solving the upper-level optimization problem to provide a time-varying optimal balance between performance at the upper-level and flexibility at the lower-level.

In the RMPC-AU design from [94] and [44], the adjustable uncertainty sets are defined as the affine transformation of an *a priori* chosen convex primitive set in the form of ellipsoids, hyper-rectangles, or polytopes. Thus, the shape and size of these uncertainty sets are optimized by determining these affine transformation variables from the feasible domain set.

Specifically in [94], control policies are optimized from a chosen class of piece-wise affine policy functions to achieve computational tractability and feasibility for a finite horizon operation. This work is extended to an infinite horizon operation in [44] by enforcing a terminal constraint set, which is an inner-approximation of the positive invariant terminal set computed based on the chosen primitive set. Moreover, when norm-balls are used to represent these uncertainty sets, as in [94], the underlying optimization requires solving a semi-definite

program which could pose computational challenges with increase in system dimension and complexity of the primitive set. Similarly in [44], the construction of robust precursor set through repetitive intersections particularly for a long prediction horizon is likely to result in large number of inequalities using H-Rep which adds complexity to the underlying optimization problem. Additionally, both [94] and [44] use a min-max optimization approach for dealing with uncertainty which is generally more computationally complex than tube-based robust MPC [60].

Specifically with regards to tube-based robust MPC, RPI sets are traditionally computed offline using the iterative procedure from [68] in H-Rep which is an outer-approximation of the minimal RPI (mRPI) set [50]. To improve computational performance and scalability, one-step optimization-based techniques have been developed in [84]. To further improve computational performance, similar one-step approaches using *zonotopes* with a corresponding Generator-Representation (G-Rep) have been developed in [66] for computing both outer-approximations of the mRPI set and inner-approximations of the Pontryagin (Minkowski) difference used for constraint tightening. In both cases, set computations are based on set containment conditions which can be reformulated as linear constraints using the techniques in [71].

This research leverages the recent developments in one-step RPI and Pontryagin difference set computations to embed the computation of these sets within a robust tube-based MPC formulation that simultaneously optimizes nominal state and input trajectories along with the size of the uncertainty set. In addition to presenting this novel tube-based MPC formulation, this research shows how all of the set containment constraints of robust MPC can be imposed as linear constraints and how the formulation of the cost function plays a critical role in minimizing conservatism introduced through inner- and outer-approximations of these sets.



## 5.2 Notation and Preliminaries

For a discrete time system, the notation  $\mathbf{x}(k)$  denotes the state  $\mathbf{x}$  at time step  $k$ . The brace notation  $j \in \{k, \dots, k + N - 1\}$  denotes all integers from  $k$  to until  $k + N - 1$ . For MPC, the double-index notation  $\hat{\mathbf{x}}(k + j|k)$  denotes the predicted state  $\hat{\mathbf{x}}$  at future time step  $k + j$  determined at  $k$ . For notational convenience, vectors and matrices are bolded and sets are shown in caligraphic font. The weighted norm is defined as  $\|\mathbf{x}\|_{\mathbf{Q}}^2 = \mathbf{x}^\top \mathbf{Q} \mathbf{x}$  where  $\mathbf{Q}$  is a positive definite diagonal weighting matrix. A vector of ones of appropriate dimension is given by  $\mathbf{1}$ . For a matrix  $\mathbf{A}$ ,  $|\mathbf{A}|$  denotes the element-wise absolute value. The notation  $\|\cdot\|_p$  denotes the  $p$ -norm of a vector. The sets  $\mathbb{N}_+$  and  $\mathbb{R}_+$  denote the set of natural numbers and positive real numbers, respectively. For sets  $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^n$ ,  $\mathcal{X} \oplus \mathcal{Y}$  denotes the Minkowski sum and  $\mathcal{X} \ominus \mathcal{Y}$  denotes the Pontryagin/Minkowski difference of  $\mathcal{Y}$  from  $\mathcal{X}$ . The volume ratio of  $\mathcal{X}$  with respect to  $\mathcal{Y}$  is given by  $V_r = \left(\frac{V(\mathcal{X})}{V(\mathcal{Y})}\right)^{\frac{1}{n}}$  where  $V(\cdot)$  denotes the volume of a set. The convex polytope  $\mathcal{S} \subset \mathbb{R}^n$  in H-Rep is defined as  $\mathcal{S} = \{\mathbf{s} \in \mathbb{R}^n \mid \mathbf{F}\mathbf{s} \leq \mathbf{h}\}$  such that  $\mathbf{F} \in \mathbb{R}^{n_h \times n}$  and  $\mathbf{h} \in \mathbb{R}^{n_h}$  where  $n_h$  denotes the number of halfspaces. A zonotope  $\mathcal{Z} = \{\mathbf{G}, \mathbf{c}\} \subset \mathbb{R}^n$  defined by  $\mathcal{Z} = \{\mathbf{G}\boldsymbol{\xi} + \mathbf{c} \mid \|\boldsymbol{\xi}\|_\infty \leq 1\}$ , where  $n_g$  denotes the number of generators such that  $\mathbf{G} \in \mathbb{R}^{n \times n_g}$  and  $\mathbf{c} \in \mathbb{R}^n$ . For zonotopes  $\mathcal{X} = \{\mathbf{G}_x, \mathbf{c}_x\} \subset \mathbb{R}^n, \mathcal{Y} = \{\mathbf{G}_y, \mathbf{c}_y\} \subset \mathbb{R}^n$ , the linear transformation by a matrix  $\mathbf{R} \in \mathbb{R}^{m \times n}$  is defined as  $\mathbf{R}\mathcal{X} = \{\mathbf{R}\mathbf{G}_x, \mathbf{R}\mathbf{c}_x\}$  and the Minkowski sum  $\mathcal{X} \oplus \mathcal{Y} = \left\{ \begin{bmatrix} \mathbf{G}_x & \mathbf{G}_y \end{bmatrix}, \mathbf{c}_x + \mathbf{c}_y \right\}$ . A hypercube with edge length  $d$  is denoted  $d\mathcal{B}$ , where  $\mathcal{B}$  is the unit hypercube defined by  $\mathcal{B} = \{b \mid \|b\|_\infty \leq 1\}$ .

## 5.3 Robust MPC Background

Consider the discrete linear time-invariant system

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k), \quad (5.1)$$

where  $\mathbf{x} \in \mathbb{R}^n$  are the states,  $\mathbf{u} \in \mathbb{R}^m$  are the inputs, and  $\mathbf{w} \in \mathbb{R}^n$  are the additive disturbances. It is assumed that the pair  $(\mathbf{A}, \mathbf{B})$  is stabilizable. The states, inputs, and distur-

bances are subject to the constraints

$$\mathbf{x}(k) \in \mathcal{X}, \quad \mathbf{u}(k) \in \mathcal{U}, \quad \mathbf{w}(k) \in \mathcal{W}, \quad (5.2)$$

assuming  $\mathcal{X}$ ,  $\mathcal{U}$ , and  $\mathcal{W}$  are compact and convex polytopes containing the origin in their interiors.

The widely-used tube-based robust MPC from [60] solves the following constrained optimization problem at every time step  $k \geq 0$ ,

$$J^*(\mathbf{x}(k)) = \min_{\hat{\mathbf{x}}(k|k), \hat{\mathbf{U}}(k)} \sum_{j=k}^{k+N-1} \ell(j|k) + \ell_f(k+N|k), \quad (5.3a)$$

$$\text{s.t.} \quad \forall j \in [k, k+N-1],$$

$$\hat{\mathbf{x}}(j+1|k) = \mathbf{A}\hat{\mathbf{x}}(j|k) + \mathbf{B}\hat{\mathbf{u}}(j|k), \quad (5.3b)$$

$$\hat{\mathbf{x}}(j|k) \in \hat{\mathcal{X}} \triangleq \mathcal{X} \ominus \mathcal{E}, \quad (5.3c)$$

$$\hat{\mathbf{u}}(j|k) \in \hat{\mathcal{U}} \triangleq \mathcal{U} \ominus K\mathcal{E}, \quad (5.3d)$$

$$\mathbf{x}(k) - \hat{\mathbf{x}}(k|k) \in \mathcal{E}, \quad (5.3e)$$

$$\hat{\mathbf{x}}(k+N|k) \in \hat{\mathcal{T}}. \quad (5.3f)$$

This MPC formulation has a prediction horizon of  $N$  steps and optimizes  $\hat{\mathbf{U}}(k) \triangleq \{\hat{\mathbf{u}}(j|k)\}_{j=k}^{k+N-1}$  to minimize the cost function (5.3a) which is typically formulated with stage costs  $\ell(j|k) = \|\hat{\mathbf{x}}(j|k)\|_{\mathbf{Q}}^2 + \|\hat{\mathbf{u}}(j|k)\|_{\mathbf{R}}^2$  and terminal cost  $\ell_f(k+N|k) = \|\hat{\mathbf{x}}(k+N|k)\|_{\mathbf{P}}^2$ , where  $\mathbf{Q}$  and  $\mathbf{R}$  are design parameters and  $\mathbf{P}$  satisfies the discrete-time algebraic Riccati equation. First introduced in [60], this tube-based robust MPC formulation also allows the initial nominal state  $\hat{\mathbf{x}}(k|k)$  to be a decision variable.

Based on the solution to (5.3), the control input applied to (5.1) is

$$\mathbf{u}(k) = \hat{\mathbf{u}}^*(k|k) + \mathbf{K}(\mathbf{x}(k) - \hat{\mathbf{x}}^*(k|k)), \quad (5.4)$$

where  $\hat{\mathbf{x}}^*(k|k)$  and  $\hat{\mathbf{u}}^*(k|k)$  denote the optimal state and input at time step  $k$  and  $\mathbf{K}$  is a stabilizing feedback control law, often chosen as the infinite-horizon, discrete-time LQR

controller. This stabilizing control law ensures that the difference,  $\mathbf{x}(k) - \hat{\mathbf{x}}(k|k)$ , between the true and nominal state trajectories satisfying (5.1) and (5.3b) respectively, always stays within a bounded set,  $\mathcal{E}$ , given the bounded disturbances,  $\mathbf{w}(k) \in \mathcal{W}$ .

This bounding set,  $\mathcal{E}$ , is typically computed as an outer-approximation of the mRPI set satisfying

$$\mathbf{A}_K \mathcal{E} \oplus \mathcal{W} \subseteq \mathcal{E}, \quad (5.5)$$

where  $\mathbf{A}_K = \mathbf{A} + \mathbf{B}\mathbf{K}$ . Thus, if the nominal state starts off close to the true state, as enforced by (5.3e), then the difference between the two state trajectories will always remain within  $\mathcal{E}$ . This enables the constraint tightening approach used in (5.3c) and (5.3d), where if the nominal trajectories satisfy the tightened constraints, the true trajectories will satisfy the original state and input constraints from (5.2).

The terminal set  $\hat{\mathcal{T}}$  in (5.3f) is typically included to guarantee robust stability by choosing  $\hat{\mathcal{T}}$  such that

$$\mathbf{A}_K \hat{\mathcal{T}} \subseteq \hat{\mathcal{T}}, \quad \hat{\mathcal{T}} \subseteq \hat{\mathcal{X}}, \quad \mathbf{K} \hat{\mathcal{T}} \subseteq \hat{\mathcal{U}}. \quad (5.6)$$

Traditionally, tube-based robust MPC development starts with the offline processes of: 1) determining the uncertainty bounding set,  $\mathcal{W}$ ; 2) computing an outer-approximation of the mRPI set,  $\mathcal{E}$ ; and 3) computing the tightened constraint sets  $\hat{\mathcal{X}}$ ,  $\hat{\mathcal{U}}$ , and  $\hat{\mathcal{T}}$ . The online process then consists of solving (5.3) and applying the input defined in (5.4).

However, there is growing interest in developing MPC-based control strategies [94, 44] where the size of the uncertainty set,  $\mathcal{W}$ , is included in the optimization. This prohibits the offline computation of the mRPI sets and constraint tightening which are usually time-consuming calculations that scale poorly with the number of states and inputs.

Therefore, this research presents a scalable robust MPC approach where the size of  $\mathcal{W}$ , and the corresponding sizes of  $\mathcal{E}$ ,  $\hat{\mathcal{X}}$ ,  $\hat{\mathcal{U}}$ , and  $\hat{\mathcal{T}}$ , are all computed online as part of the overall optimization problem.

## 5.4 Robust MPC with Integrated RPI Set Computation and Constraint Tightening

This research assumes that the shape of the uncertainty set is predetermined but the size is variable such that

$$\mathbf{w}(k) \in \mathcal{W}(\Phi_w), \quad (5.7)$$

where  $\Phi_w$  is a scaling variable with positive entries. Similar scaling variables have been used in [85] and [77]. In [85], a scaling variable  $\Phi_f$  is used to scale the offset vector of the uncertainty set represented in H-Rep as

$$\mathcal{W}(\Phi_f) = \{\mathbf{w} \mid \mathbf{H}\mathbf{w} \leq \Phi_f\}, \quad \Phi_f \in \mathbb{R}^{n_h}. \quad (5.8)$$

Alternatively, in [77], the scaling variable  $\Phi_m$  scales the magnitude of a nominal uncertainty set  $\mathcal{W}^*$  where

$$\mathcal{W}(\Phi_m) = \Phi_m \mathcal{W}^*, \quad \Phi_m \in \mathbb{R}_+. \quad (5.9)$$

Since the scaling variable  $\Phi_w$  in (7) is a decision variable, the sets  $\mathcal{E}$ ,  $\hat{\mathcal{X}}$ ,  $\hat{\mathcal{U}}$ , and  $\hat{\mathcal{T}}$  cannot be precomputed. Instead, each of these sets will have a nominal shape and a corresponding scaling variable such that

$$\mathcal{E} \subseteq \tilde{\mathcal{E}}(\Phi_\varepsilon), \quad \tilde{\mathcal{X}}(\Phi_x) \subseteq \hat{\mathcal{X}}, \quad \tilde{\mathcal{U}}(\Phi_u) \subseteq \hat{\mathcal{U}}, \quad \tilde{\mathcal{T}}(\Phi_t) \subseteq \hat{\mathcal{T}}. \quad (5.10)$$

With the inclusion of these scalable sets, (5.3) is reformulated as

$$J^*(\mathbf{x}(k)) = \min_{\substack{\Phi, \hat{\mathbf{x}}(k|k), j=k \\ \hat{\mathbf{U}}(k)}}^{\quad k+N-1} \sum \ell(j|k) + \ell_f(k+N|k) + \ell_\Phi, \quad (5.11a)$$

$$\text{s.t.} \quad \forall j \in [k, k+N-1],$$

$$\hat{\mathbf{x}}(j+1|k) = \mathbf{A}\hat{\mathbf{x}}(j|k) + \mathbf{B}\hat{\mathbf{u}}(j|k), \quad (5.11b)$$

$$\hat{\mathbf{x}}(j|k) \in \tilde{\mathcal{X}}(\Phi_x), \quad \tilde{\mathcal{X}}(\Phi_x) \oplus \tilde{\mathcal{E}}(\Phi_\varepsilon) \subseteq \mathcal{X}, \quad (5.11c)$$

$$\hat{\mathbf{u}}(j|k) \in \tilde{\mathcal{U}}(\Phi_u), \quad \tilde{\mathcal{U}}(\Phi_u) \oplus \mathbf{K}\tilde{\mathcal{E}}(\Phi_\varepsilon) \subseteq \mathcal{U}, \quad (5.11d)$$

$$\mathbf{x}(k) - \hat{\mathbf{x}}(k|k) \in \tilde{\mathcal{E}}(\Phi_\varepsilon), \quad (5.11e)$$

$$\mathbf{A}_\mathbf{K}\tilde{\mathcal{E}}(\Phi_\varepsilon) \oplus \mathcal{W}(\Phi_w) \subseteq \tilde{\mathcal{E}}(\Phi_\varepsilon), \quad (5.11f)$$

$$\hat{\mathbf{x}}(k+N|k) \in \tilde{\mathcal{T}}(\Phi_t), \quad \mathbf{A}_\mathbf{K}\tilde{\mathcal{T}}(\Phi_t) \subseteq \tilde{\mathcal{T}}(\Phi_t), \quad (5.11g)$$

$$\tilde{\mathcal{T}}(\Phi_t) \subseteq \tilde{\mathcal{X}}(\Phi_x), \quad \mathbf{K}\tilde{\mathcal{T}}(\Phi_t) \subseteq \tilde{\mathcal{U}}(\Phi_u). \quad (5.11h)$$

By allowing the RPI and tightened constraint sets to vary in size, the scaling matrices  $\Phi = \{\Phi_w, \Phi_\varepsilon, \Phi_x, \Phi_u, \Phi_t\}$  are included as decision variables in (5.11). The sizes of these corresponding sets are prioritized in (5.11a) using a generic cost function  $\ell_\Phi = \ell_\Phi(\Phi_w, \Phi_\varepsilon, \Phi_x, \Phi_u, \Phi_t)$  as discussed in more detail in Section VIB. The set containment condition in (5.11c) has been added to ensure that

$$\tilde{\mathcal{X}}(\Phi_x) \subseteq \mathcal{X} \ominus \tilde{\mathcal{E}}(\Phi_\varepsilon) \subseteq \hat{\mathcal{X}} \triangleq \mathcal{X} \ominus \mathcal{E}, \quad (5.12)$$

is satisfied based on the extensive property of the closing of  $\tilde{\mathcal{X}}(\Phi_x)$  by  $\tilde{\mathcal{E}}(\Phi_\varepsilon)$ . A similar set containment condition has been added for the input set in (5.11d) to ensure satisfaction of

$$\tilde{\mathcal{U}}(\Phi_u) \subseteq \mathcal{U} \ominus \mathbf{K}\tilde{\mathcal{E}}(\Phi_\varepsilon) \subseteq \hat{\mathcal{U}} \triangleq \mathcal{U} \ominus \mathbf{K}\mathcal{E}. \quad (5.13)$$

The set containment condition in (5.11f) has been added to ensure that  $\tilde{\mathcal{E}}(\Phi_\varepsilon)$  satisfies the definition of an RPI set from (5.5). Finally, the set containment conditions in (5.11g) and (5.11h) have been added to ensure that  $\tilde{\mathcal{T}}(\Phi_t)$  satisfies the properties of a terminal set from (5.6). Note that the size of the uncertainty set, determined by the scaling matrix  $\Phi_w$ , only appears in (5.11f) but directly affects the size of the RPI set and the state, input, and terminal set constraint tightening. The following theorem mathematically proves the recursive feasibility of the proposed tube-based robust MPC framework.

**Theorem 12.** *If the MPC optimization problem in (5.11) is feasible at time step  $k$ , then it is feasible at time step  $k+1$ .*

*Proof.* The proof is the same as that of tube-based robust MPC from Proposition 3 in [60] with the addition that the candidate solution for the scaling variables at time step  $k + 1$  is equal to the optimal solution from time step  $k$  such that  $\Phi(k + 1) = \Phi^*(k)$ .  $\square$

Usually, in constrained robust MPC, desired performance is achieved by letting the state and input trajectories operate close to the bounds of the tightened state and input constraints. A typical example is the thermal management of aircraft electro-thermal systems where the temperatures of the system components are allowed to approach the respective upper-bounds to minimize coolant flow and achieve energy-efficient operation [46]. Usually, the larger the uncertainty set, the smaller the tightened state, input, and terminal sets, which results in loss of desired system performance. Thus, there exists a trade-off between the size of the uncertainty set and system performance, which must be optimized.

The main contribution of this chapter lies in showing that despite the introduction of scaling variables,  $\Phi$ , as decision variables, all of the point and set containment conditions in (5.11c)-(5.11h) can be represented as linear constraints. Therefore, if (5.3) is formulated as a Quadratic Program (QP), then (5.11) is also a QP. This key feature of the proposed approach is enabled through the use of *zonotopes* (See Chapter 2 for details).

## 5.5 Zonotopic Set Operations

The results in this research heavily rely on the following zonotope containment condition from [71].

**Lemma 10.** (Corollary 4 of [71]) *Given two zonotopes  $\mathcal{F} = \{\mathbf{G}_f, \mathbf{c}_f\} \subset \mathbb{R}^n$  and  $\mathcal{H} = \{\mathbf{G}_h, \mathbf{c}_h\} \subset \mathbb{R}^n$ ,  $\mathcal{F} \subseteq \mathcal{H}$  if there exists  $\mathbf{\Gamma} \in \mathbb{R}^{n_h \times n_f}$  and  $\boldsymbol{\beta} \in \mathbb{R}^{n_h}$  such that*

$$\mathbf{G}_f = \mathbf{G}_h \mathbf{\Gamma}, \quad \mathbf{c}_h - \mathbf{c}_f = \mathbf{G}_h \boldsymbol{\beta}, \quad |\mathbf{\Gamma}| \mathbf{1} + |\boldsymbol{\beta}| \leq \mathbf{1}. \quad (5.14)$$

Using this zonotope containment condition, the main idea is to scale a zonotope such that set containment conditions from (5.11c) and (5.11d) and RPI set condition from (5.11f) holds.

**Definition 9.** *The zonotope  $\mathcal{Z}(\Phi) = \{\mathbf{G}\Phi, \mathbf{c}\} \subset \mathbb{R}^n$  is a scaled version of the nominal zonotope  $\mathcal{Z} = \{\mathbf{G}, \mathbf{c}\}$  with the generator matrix  $\mathbf{G}$  scaled by a diagonal matrix  $\Phi \in \mathbb{R}^{n_g \times n_g}$ ,  $\Phi = \text{diag}(\phi_i), \phi_i > 0, \forall i \in \{1, \dots, n_g\}$ .*

**Assumption 10.** *For the chosen  $\mathbf{G}$  and the system  $(\mathbf{A}_K, \mathcal{W})$ , there exists a  $\Phi \in \mathbb{R}^{n_g \times n_g}$  that scales the RPI set  $\mathcal{E}$  such that (5) holds.*

While *Assumption 10* is needed to exclude certain systems that do not admit RPI sets [84], this assumption is typically mild in practice through the proper choice of  $\mathbf{G}$ , as discussed below.

**Theorem 13.** *(Theorem 6 of [66]) The zonotope  $\mathcal{Z}(\Phi) = \{\mathbf{G}\Phi, \mathbf{c}\} \subset \mathbb{R}^n$  is an RPI set of  $\mathbf{x}(k+1) = \mathbf{A}_K \mathbf{x}(k) + \mathbf{w}(k)$  if  $\mathbf{w}(k) \in \mathcal{W} = \{\mathbf{G}_w, \mathbf{c}_w\}$ , and there exists  $\Gamma_1 \in \mathbb{R}^{n_g \times n_g}$ ,  $\Gamma_2 \in \mathbb{R}^{n_g \times n_w}$ , and  $\beta \in \mathbb{R}^{n_g}$  such that*

$$\mathbf{A}_K \mathbf{G} \Phi = \mathbf{G} \Gamma_1, \quad (5.15a)$$

$$\mathbf{G}_w = \mathbf{G} \Gamma_2, \quad (5.15b)$$

$$(\mathbf{I} - \mathbf{A}_K) \mathbf{c} - \mathbf{c}_w = \mathbf{G} \beta, \quad (5.15c)$$

$$|\Gamma_1| \mathbf{1} + |\Gamma_2| \mathbf{1} + |\beta| \leq \Phi \mathbf{1}. \quad (5.15d)$$

**Theorem 13** can be applied to determine the RPI set  $\mathcal{Z}(\Phi)$  for a predetermined generator matrix  $\mathbf{G}$ . For a desired order of  $\mathcal{Z}$ ,  $\mathbf{G}$  can be determined using  $\mathbf{G}_w$  and  $\mathbf{A}_K$  as  $\mathbf{G} = [\mathbf{G}_w \ \mathbf{A}_K \mathbf{G}_w \ \dots \ \mathbf{A}_K^s \mathbf{G}_w]$ , for some  $s \in \mathbb{N}_+$  that provides the desired order. Thus  $\mathbf{G}$  is a truncated version of the infinite sum used to compute the mRPI set in [68]. However, for

an improper choice of  $\mathbf{G}$  or with an inadequate number of generators in  $\mathbf{G}$ , there might not exist a  $\Phi$  that scales  $\mathcal{Z}(\Phi)$  to satisfy (5.15), resulting in an infeasible optimization problem. Assuming a properly chosen  $\mathbf{G}$ , the size of  $\mathcal{Z}$  can be scaled by the diagonal matrix  $\Phi$  such that  $\mathcal{Z}(\Phi)$  is an approximation of the mRPI set, where  $\Phi$  is determined by solving an optimization problem with the constraints from (5.15) and an objective function that minimizes the scaling variables in  $\Phi$ . This approach is an indirect attempt to minimize the volume of  $\mathcal{Z}(\Phi)$ , since directly optimizing the volume of a zonotope is a nonconvex problem [39]. With  $\mathbf{c}$ ,  $\Phi$ ,  $\Gamma_1$ ,  $\Gamma_2$ , and  $\beta$  as decision variables, (5.15) consists of only linear constraints and thus a LP or QP can be formulated based on the  $p$ -norm used to minimize the vector  $\phi$ , where  $\Phi = \text{diag}(\phi)$ .

**Theorem 14.** (Theorem 7 of [66]) Given  $\mathcal{Z}_1 = \{\mathbf{G}_1, \mathbf{c}_1\}$  and  $\mathcal{Z}_2 = \{\mathbf{G}_2, \mathbf{c}_2\}$ , then  $\tilde{\mathcal{Z}}_d = \{\mathbf{G}_d \Phi, \mathbf{c}_d\}$ , with  $\Phi = \text{diag}(\phi)$ ,  $\phi_i > 0, \forall i \in \{1, \dots, n_{g_d}\}$ , is an inner-approximation of the Pontryagin difference  $\mathcal{Z}_d = \mathcal{Z}_1 \ominus \mathcal{Z}_2$  such that  $\tilde{\mathcal{Z}}_d \subseteq \mathcal{Z}_d$  if there exists  $\Gamma \in \mathbb{R}^{n_{g_1} \times (n_{g_d} + n_{g_2})}$  and  $\beta \in \mathbb{R}^{n_{g_1}}$  such that

$$\begin{bmatrix} \mathbf{G}_d \Phi & \mathbf{G}_2 \end{bmatrix} = \mathbf{G}_1 \Gamma, \quad (5.16a)$$

$$\mathbf{c}_1 - (\mathbf{c}_d + \mathbf{c}_2) = \mathbf{G}_1 \beta, \quad (5.16b)$$

$$|\Gamma| \mathbf{1} + |\beta| \leq \mathbf{1}. \quad (5.16c)$$

**Theorem 14** can be applied to compute  $\tilde{\mathcal{Z}}_d \subseteq \mathcal{Z}_d$  and it is practical to assume the generator matrix  $\mathbf{G}_d$  is comprised of the generators from both  $\mathcal{Z}_1$  and  $\mathcal{Z}_2$  such that  $\mathbf{G}_d = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 \end{bmatrix}$ . The set  $\tilde{\mathcal{Z}}_d$  with maximal volume is typically desired and can be computed by solving an optimization problem formulated with the constraints from (5.16) and an objective function that maximizes the scaling variables in  $\Phi$ . With  $\mathbf{c}_d$ ,  $\Phi$ ,  $\Gamma$ , and  $\beta$  as decision variables in this optimization problem, (5.16) consists of only linear constraints and



thus a LP or QP can be formulated based on the  $p$ -norm used to maximize the vector  $\phi$ , where  $\Phi = \text{diag}(\phi)$ .

## 5.6 One-step RPI Set Computation and Constraint Tightening

When analyzing (5.11), the addition of scalable sets introduces the need to 1) enforce point containment within a scaled tightened constraint set (e.g.  $\hat{\mathbf{x}}(j|k) \in \tilde{\mathcal{X}}(\Phi_x)$ ) and 2) set containment for inner-approximations of the tightened constraint set (e.g.  $\tilde{\mathcal{X}}(\Phi_x) \oplus \tilde{\mathcal{E}}(\Phi_\varepsilon) \subseteq \mathcal{X}$ ) and the outer-approximation of the mRPI set (e.g.  $\mathbf{A}_K \tilde{\mathcal{E}}(\Phi_\varepsilon) \oplus \mathcal{W}(\Phi_w) \subseteq \tilde{\mathcal{E}}(\Phi_\varepsilon)$ ). To reduce unnecessary complexity and highlight the ability to simultaneously formulate these containment conditions as linear constraints, the following optimization problem is introduced for some predefined point  $\hat{\mathbf{x}}(j|k)$  and sets  $\mathcal{X}$ ,  $\tilde{\mathcal{X}}$ ,  $\tilde{\mathcal{E}}$ , and  $\mathcal{W}$ .

$$\min_{\Phi_x, \Phi_\varepsilon, \Phi_w} \ell_\Phi, \quad (5.17a)$$

s.t.

$$\hat{\mathbf{x}}(j|k) \in \tilde{\mathcal{X}}(\Phi_x), \quad (5.17b)$$

$$\tilde{\mathcal{X}}(\Phi_x) \oplus \tilde{\mathcal{E}}(\Phi_\varepsilon) \subseteq \mathcal{X}, \quad (5.17c)$$

$$\mathbf{A}_K \tilde{\mathcal{E}}(\Phi_\varepsilon) \oplus \mathcal{W}(\Phi_w) \subseteq \tilde{\mathcal{E}}(\Phi_\varepsilon). \quad (5.17d)$$

The following subsections use **Theorems 13** and **14** to convert the point and set containment conditions from (5.17b)-(5.17d) to linear constraints and discuss how to formulate the cost function in (5.17a) to optimize the size of these sets. Note that similar linear constraints and cost functions designs can be formulated for the input and terminal constraint sets conditions from (5.11d), (5.11g), and (5.11h) to recast all of the constraints in (5.11) as linear constraints.

### 5.6.1 Containment Constraints

**Theorem 15.** Given  $\mathcal{X} = \{\mathbf{G}_x, \mathbf{c}_x\}$ , let  $\tilde{\mathcal{X}} = \{\tilde{\mathbf{G}}_x \Phi_x, \tilde{\mathbf{c}}_x\}$  and  $\tilde{\mathcal{E}} = \{\mathbf{G}_\varepsilon \Phi_\varepsilon, \mathbf{c}_\varepsilon\}$ . Then (5.17b)-(5.17d) are satisfied if there exists  $\boldsymbol{\xi}_x \in \mathbb{R}^{n_x}, \boldsymbol{\Gamma}_1 \in \mathbb{R}^{n_x \times (n_x + n_\varepsilon)}, \boldsymbol{\beta}_1 \in \mathbb{R}^{n_x}, \boldsymbol{\Gamma}_2 \in \mathbb{R}^{n_\varepsilon \times n_\varepsilon}, \boldsymbol{\Gamma}_3 \in \mathbb{R}^{n_\varepsilon \times n_w}$ , and  $\boldsymbol{\beta}_2 \in \mathbb{R}^{n_\varepsilon}$  such that

$$\hat{\mathbf{x}}(j|k) = \tilde{\mathbf{G}}_x \boldsymbol{\xi}_x + \tilde{\mathbf{c}}_x, \quad -\Phi_x \leq \boldsymbol{\xi}_x \leq \Phi_x, \quad (5.18a)$$

$$\begin{bmatrix} \tilde{\mathbf{G}}_x \Phi_x & \mathbf{G}_\varepsilon \Phi_\varepsilon \end{bmatrix} = \mathbf{G}_x \boldsymbol{\Gamma}_1, \quad (5.18b)$$

$$\mathbf{c}_x - (\tilde{\mathbf{c}}_x + \mathbf{c}_\varepsilon) = \mathbf{G}_x \boldsymbol{\beta}_1, \quad (5.18c)$$

$$|\boldsymbol{\Gamma}_1| \mathbf{1} + |\boldsymbol{\beta}_1| \leq \mathbf{1}, \quad (5.18d)$$

$$\mathbf{A}_K \mathbf{G}_\varepsilon \Phi_\varepsilon = \mathbf{G}_\varepsilon \boldsymbol{\Gamma}_2, \quad (5.18e)$$

$$\mathbf{G}_w = \mathbf{G}_\varepsilon \boldsymbol{\Gamma}_3, \quad (5.18f)$$

$$(\mathbf{I} - \mathbf{A}_K) \mathbf{c}_\varepsilon - \mathbf{c}_w = \mathbf{G}_\varepsilon \boldsymbol{\beta}_2, \quad (5.18g)$$

$$|\boldsymbol{\Gamma}_2| \mathbf{1} + |\boldsymbol{\Gamma}_3| \mathbf{1} + |\boldsymbol{\beta}_2| \leq \Phi_\varepsilon \mathbf{1}. \quad (5.18h)$$

*Proof.* The proof requires showing that (5.18a) enforces the point containment condition from (5.17b), (5.18b)-(5.18d) enforce the Pontryagin difference definition (5.17c), and (5.18e)-(5.18h) enforce the definition of an RPI set from (5.5). Consider the change of variables  $\boldsymbol{\xi}_x = \Phi_x \hat{\boldsymbol{\xi}}_x$ ,  $\boldsymbol{\Gamma}_2 = \Phi_\varepsilon \tilde{\boldsymbol{\Gamma}}_2$ ,  $\boldsymbol{\Gamma}_3 = \Phi_\varepsilon \tilde{\boldsymbol{\Gamma}}_3$ ,  $\boldsymbol{\beta}_2 = \Phi_\varepsilon \tilde{\boldsymbol{\beta}}_2$  and define  $\tilde{\boldsymbol{\Gamma}}_{23} = \begin{bmatrix} \tilde{\boldsymbol{\Gamma}}_2 & \tilde{\boldsymbol{\Gamma}}_3 \end{bmatrix}$ . Then, (5.18a) readily satisfies the definition of a zonotope with  $\hat{\boldsymbol{\xi}}_x \in [-\mathbf{1}, \mathbf{1}]$ . The Pontryagin difference containment conditions from **Theorem 14** are satisfied by (5.18b)-(5.18d). Then the zonotope containment conditions from **Lemma 10** are satisfied by 1) rearranging and combining (5.18e) and (5.18f) to get  $\begin{bmatrix} \mathbf{A}_K \mathbf{G}_\varepsilon \Phi_\varepsilon & \mathbf{G}_w \end{bmatrix} = \mathbf{G}_\varepsilon \Phi_\varepsilon \tilde{\boldsymbol{\Gamma}}_{23}$ , 2) rearranging (5.18g) to get  $\mathbf{c}_\varepsilon - (\mathbf{A}_K \mathbf{c}_\varepsilon + \mathbf{c}_w) = \mathbf{G}_\varepsilon \Phi_\varepsilon \tilde{\boldsymbol{\beta}}_2$ , and 3) multiplying (5.18h) by  $\Phi_\varepsilon^{-1}$ , since  $\phi_i > 0$ , to get  $|\tilde{\boldsymbol{\Gamma}}_{23}| \mathbf{1} + |\tilde{\boldsymbol{\beta}}_2| \leq \mathbf{1}$ .  $\square$

## 5.6.2 Cost Function

In determining the cost function  $\ell_{\Phi}$ , it is typically desired to let  $\ell_{\Phi}$  be a linear or a quadratic function of  $\Phi_x, \Phi_\varepsilon, \Phi_w$  so that (5.17), and therefore (5.11), remains a linear or quadratic program. Conceptually,  $\ell_{\Phi}$  should be defined to maximize the size of the uncertainty set  $\mathcal{W}(\Phi_w)$  while minimizing the outer-approximation of the mRPI set  $\tilde{\mathcal{E}}(\Phi_\varepsilon)$  and maximizing the inner-approximation of the tightened constraint set  $\tilde{\mathcal{X}}(\Phi_x)$ . Interestingly, maximizing  $\tilde{\mathcal{X}}(\Phi_x)$  automatically incentivizes minimizing  $\tilde{\mathcal{E}}(\Phi_\varepsilon)$ . Thus,  $\ell_{\Phi}$  should be designed to maximize  $\Phi_w$  while also maximizing  $\Phi_x$ . However, maximizing  $\Phi_x$  alone would minimize  $\Phi_w$ , and vice-versa, and thus there is a trade-off.

With  $\Phi_w = \text{diag}(\phi_w)$  and  $\Phi_x = \text{diag}(\phi_x)$ , one approach to formulating  $\ell_{\Phi}$  is based on the norms of the scaling variables such that

$$\ell_{\Phi} = -\|\phi_x\|_p - \lambda\|\phi_w\|_p, \quad \lambda \geq 0, \quad (5.19)$$

where the  $p$ -norms,  $\|\cdot\|_p$ , can be used to express the 1-norm, 2-norm, or  $\infty$ -norm. The weighting term  $\lambda$  can be used to tune the priority between maximizing the uncertainty and tightened constraint sets. Similar scaling variables have been used in [39] to outer-approximate asymmetric polytopes by scaled zonotopes obtained by maximizing the 1- and  $\infty$ -norm of the generators. However, maximizing the  $p$ -norm of the scaling variables may not effectively maximize the *volume* of these approximating sets, as will be shown in *Example 9*.

Alternatively, the Hausdorff distance can be used to effectively maximize the size of the tightened constraint set such that  $\tilde{\mathcal{X}}(\Phi_x)$  inner-approximates  $\hat{\mathcal{X}}$ . As defined in [71], the Hausdorff distance  $d$  for two sets  $\mathcal{P}$  and  $\mathcal{Q}$  is

$$d = \min_{0 \leq d \in \mathbb{R}} \{ \mathcal{Q} \subseteq \mathcal{P} \oplus d\mathcal{B}, \quad \mathcal{P} \subseteq \mathcal{Q} \oplus d\mathcal{B} \}. \quad (5.20)$$

Using this Hausdorff distance metric, the optimization problem from (5.17) is modified

as

$$\min_{d_x} \ell_d, \quad (5.21a)$$

s.t.

$$\mathcal{X} \subseteq \tilde{\mathcal{X}}(\Phi_x) \oplus d_x \mathcal{B}, \quad (5.21b)$$

$$\hat{\mathbf{x}}(j|k) \in \tilde{\mathcal{X}}(\Phi_x), \quad (5.21c)$$

$$\tilde{\mathcal{X}}(\Phi_x) \oplus \tilde{\mathcal{E}}(\Phi_\varepsilon) \subseteq \mathcal{X}, \quad (5.21d)$$

$$\mathbf{A}_K \tilde{\mathcal{E}}(\Phi_\varepsilon) \oplus \mathcal{W}(\Phi_w) \subseteq \tilde{\mathcal{E}}(\Phi_\varepsilon). \quad (5.21e)$$

**Theorem 16.** *Given  $\mathcal{X} = \{\mathbf{G}_x, \mathbf{c}_x\}$  and  $\tilde{\mathcal{X}} = \{\tilde{\mathbf{G}}_x \Phi_x, \tilde{\mathbf{c}}_x\}$ , the containment property in (5.21b) is satisfied if there exists  $\Gamma_4 \in \mathbb{R}^{n_{\tilde{x}} \times n_x}$ ,  $\Gamma_5 \in \mathbb{R}^{n \times n_x}$ ,  $\beta_3 \in \mathbb{R}^{n_{\tilde{x}}}$ ,  $\beta_4 \in \mathbb{R}^n$  such that*

$$\mathbf{G}_x = \tilde{\mathbf{G}}_x \Gamma_4 + \Gamma_5, \quad (5.22a)$$

$$\tilde{\mathbf{c}}_x - \mathbf{c}_x = \tilde{\mathbf{G}}_x \beta_3 + \beta_4, \quad (5.22b)$$

$$|\Gamma_4| \mathbf{1} + |\beta_3| \leq \Phi_x \mathbf{1}, \quad (5.22c)$$

$$|\Gamma_5| \mathbf{1} + |\beta_4| \leq d_x \mathbf{1}. \quad (5.22d)$$

*Proof.* The proof requires using **Lemma 10** to show that (5.22a)-(5.22d) enforce the zonotope containment condition from (5.21b). Consider the change of variables  $\Gamma_4 = \Phi_x \tilde{\Gamma}_4$ ,  $\beta_3 = \Phi_x \tilde{\beta}_3$ ,  $\Gamma_5 = d_x \tilde{\Gamma}_5$ ,  $\beta_4 = d_x \tilde{\beta}_4$ , and let  $\Gamma_{45} = \begin{bmatrix} \Gamma_4^\top & \Gamma_5^\top \end{bmatrix}^\top$  and  $\beta_{34} = \begin{bmatrix} \beta_3^\top & \beta_4^\top \end{bmatrix}^\top$ . Then the zonotope containment conditions from **Lemma 10** are readily satisfied by expressing (5.22a) in terms of  $\Gamma_{45}$  to get  $\mathbf{G}_x = \begin{bmatrix} \tilde{\mathbf{G}}_x & \mathbf{I} \end{bmatrix} \Gamma_{45}$  and expressing (5.22b) in terms of  $\beta_{34}$  to get  $\tilde{\mathbf{c}}_x - \mathbf{c}_x = \begin{bmatrix} \tilde{\mathbf{G}}_x & \mathbf{I} \end{bmatrix} \beta_{34}$ . Then, (5.22c) is written in terms of  $\tilde{\Gamma}_4$  and  $\tilde{\beta}_3$  to get  $|\Phi_x \tilde{\Gamma}_4| \mathbf{1} + |\Phi_x \tilde{\beta}_3| \leq \Phi_x \mathbf{1}$ , while (5.22d) is written in terms of  $\tilde{\Gamma}_5$  and  $\tilde{\beta}_4$  to get  $|d_x \tilde{\Gamma}_5| \mathbf{1} + |d_x \tilde{\beta}_4| \leq d_x \mathbf{1}$ . Cancelling  $\Phi_x$  and  $d_x$  in these equations results in  $|\tilde{\Gamma}_4| \mathbf{1} + |\tilde{\beta}_3| \leq \mathbf{1}$

and  $|\tilde{\Gamma}_5| \mathbf{1} + |\tilde{\beta}_4| \leq \mathbf{1}$ . Finally, letting  $\tilde{\Gamma}_{45} = \begin{bmatrix} \tilde{\Gamma}_4^\top & \tilde{\Gamma}_5^\top \end{bmatrix}^\top$  and  $\tilde{\beta}_{34} = \begin{bmatrix} \tilde{\beta}_3^\top & \tilde{\beta}_4^\top \end{bmatrix}^\top$ , these equations are concatenated vertically to obtain the final result  $|\tilde{\Gamma}_{45}| \mathbf{1} + |\tilde{\beta}_{34}| \leq \mathbf{1}$ .  $\square$

The cost function in (5.21a) minimizes the Hausdorff distance between the state constraint set  $\mathcal{X}$  and the inner-approximating tightened state constraint set  $\tilde{\mathcal{X}}(\Phi_x)$ . Similar to (5.17a), minimizing the Hausdorff distance maximizes  $\tilde{\mathcal{X}}(\Phi_x)$  which incentivizes minimizing  $\tilde{\mathcal{E}}(\Phi_\varepsilon)$ . The cost  $\ell_d$  is formulated based on the Hausdorff distance  $d_x$  such that

$$\ell_d = d_x - \lambda \|\phi_w\|_p. \quad (5.23)$$

Similar to (5.19), the weighting term  $\lambda$  provides the desired tradeoff between maximizing the uncertainty and tightened constraint sets where increasing  $\lambda$  should increase the size of the uncertainty set. The following example demonstrates the performance of minimizing the Hausdorff distance metric over maximizing the norm of the scaling variables.

**Example 9.** *Consider the zonotopes*

$$\mathcal{Z}_1 = \left\{ \begin{bmatrix} 5 & 2 & 1 \\ 3 & -1 & 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}, \quad \mathcal{Z}_2 = \left\{ \begin{bmatrix} 1 & 0.2 & 0.5 \\ -0.3 & -0.1 & 0.3 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\},$$

shown in the left subplot of Fig. 5.1. First, the Pontryagin difference  $\mathcal{Z}_d = \mathcal{Z}_1 \ominus \mathcal{Z}_2$  is computed in *H-Rep* using the *Multi-Parametric Toolbox* [37] as a benchmark for volume ratio comparison, as shown in green in the right subplot. The inner-approximations of the Pontryagin difference  $\tilde{\mathcal{Z}}_d \subseteq \mathcal{Z}_d$  computed using **Theorem 14** are shown in the right subplot for  $p = 1, 2$ , and  $\infty$ . The best approximation is obtained with  $p = 2$  by solving a QP instead of a LP with  $V_r = 0.80$ . For  $p = 1$  and  $\infty$ , the volume ratios are 0.29 and 0.38.

However, choosing to compute  $\tilde{\mathcal{Z}}_d$  by minimizing the Hausdorff distance based cost function in (5.23) results in the exact set  $\mathcal{Z}_d$  with  $V_r = 1$  while still solving a LP.

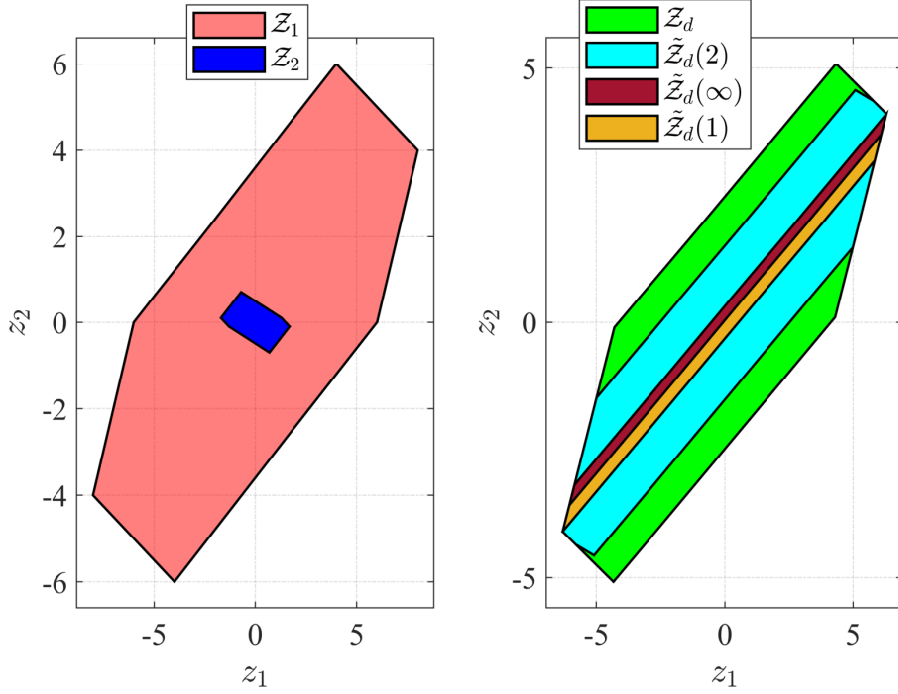


Figure 5.1. Left: Given zonotopes  $Z_1, Z_2$ . Right: The Pontryagin difference  $Z_d$  is shown in green and the inner-approximating Pontryagin differences computed using **Theorem 14** for  $p = 1, 2$ , and  $\infty$  are shown in yellow, cyan, and brown.

Using **Theorems 15** and **16** along with the Hausdorff distance based cost function from (5.23), the tightened input and terminal state constraint set conditions in (5.11d), (5.11g) and (5.11h) can also be formulated as linear constraints. Thus, the sizes of the uncertainty set, state, input, and terminal constraint sets can all be simultaneously optimized online along with the nominal state and input trajectories enabling the proposed tube-based robust MPC framework. By optimizing  $\Phi_w$  at every update of the controller, the proposed approach provides the desired time-varying balance between flexibility and performance.

The following numerical example demonstrates the key features of the proposed approach.

## 5.7 Numerical Example

A simplified vehicle system model is considered with

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{w}(k), \quad (5.24)$$

where the states  $\mathbf{x}(k) \in \mathbb{R}^2$  represent position and velocity, input  $\mathbf{u}(k) \in \mathbb{R}^1$  represents acceleration/deceleration, and the uncertainty  $\mathbf{w}(k) \in \mathcal{W}(\Phi_w) = \{\|\mathbf{w}(k)\|_\infty \leq \Phi_w\}$ ,  $\Phi_w \in \mathbb{R}$  affects only the velocity state. Note that the scaling matrix  $\Phi_w$  of the uncertainty set  $\mathcal{W}(\Phi_w) = \{\mathbf{G}_w \Phi_w, \mathbf{c}_w\}$  is computed as part of the robust MPC optimization problem in (5.11) with  $\mathbf{c}_w = 0$ , and  $\mathbf{G}_w = 1$ . The system and controller have time step size  $\Delta t = 1$  second. The prediction horizon is  $N = 100$  steps.

Starting from an initial condition  $\mathbf{x}(0) = [0 \ 0]^\top$ , the desired operation is defined by tracking references  $\{\mathbf{r}(k)\}_{k=0}^N$  for the position state and the acceleration/deceleration input (as shown in Fig. 5.2) using the weighted quadratic cost function

$$\ell(\mathbf{x}(j), \mathbf{u}(j), \mathbf{r}(j)) = \|\mathbf{r}(j) - \mathbf{z}(j)\|_{\mathbf{I}}^2, \quad (5.25)$$

where  $\mathbf{z}(j) = \begin{bmatrix} [1 \ 0]\mathbf{x}(j) \\ \mathbf{u}(j) \end{bmatrix}$ . The state constraint set  $\mathcal{X}$  and input constraint set  $\mathcal{U}$  are defined as

$$\begin{bmatrix} -1 \\ -5 \end{bmatrix} \leq \mathbf{x}(k) \leq \begin{bmatrix} 60 \\ 5 \end{bmatrix}, \quad -0.5 \leq \mathbf{u}(k) \leq 0.5.$$

The position reference was intentionally designed to be on the boundary of  $\mathcal{X}$  to clearly demonstrate the tradeoff between cost performance and uncertainty set optimization. The terminal set for the robust MPC is  $\hat{\mathcal{T}} = \tilde{\mathcal{T}}(\Phi_t) = [0 \ 0]^\top$ .

Different values of uncertainty weighting  $\lambda$  from (5.23) ranging from  $10^1$  to  $5 \times 10^7$  are considered to analyze the relationship between the uncertainty scaling variable  $\Phi_w$ , the corresponding RPI and tightened constraint sets, and the vehicle performance. Fig.

5.3 shows the trend between  $\lambda$  and  $\Phi_w$  which determines the size of  $\mathcal{W}$ . Note that the uncertainty set size increases with  $\lambda$ . Similarly, Fig. 5.4 shows that the total operational cost based on (5.25) also increases with  $\lambda$ . For clarity of exposition, three key data points, corresponding to  $\lambda \in \{10^1, 2 \times 10^5, 10^6\}$  were identified to more closely study the relationship between uncertainty set size and system operation.

Fig. 5.2 shows the simulation results using the proposed robust MPC for the chosen values of  $\lambda$ . The first subplot shows the reduction in position reference tracking performance corresponding to increasing values of  $\lambda$ . This reduced performance is a result of the reduced vehicle velocities shown in the second subplot and the reduced acceleration and deceleration shown in the third subplot. Specifically, the magnitude of the inputs decreases to compensate for the growing uncertainty set size. Intuitively, with increasing uncertainty set size, more of the control input is allocated to disturbance rejection, leaving less of the control input to be used for nominal acceleration and deceleration of the vehicle. Note that position reference tracking performance is also reduced under increasing uncertainty set size since the reference position of 60 m is no longer in the tightened state constraint set  $\tilde{\mathcal{X}}$ .

Fig. 5.5 shows the tightened state constraint set  $\tilde{\mathcal{X}}(\Phi_x)$  and tightened input constraint set  $\tilde{\mathcal{U}}(\Phi_u)$  from (5.10) for different uncertainty set sizes realized for the chosen values of  $\lambda$ . Clearly, the volume of the tightened state and input constraint sets decrease as the uncertainty set size increases.

An important limitation of the proposed approach is the conservatism introduced by the one-step approximation of the RPI set and corresponding constraint tightening. As shown in Fig. 5.6,  $\tilde{\mathcal{E}}(\Phi_\varepsilon)$  computed by solving (5.11) is an outer-approximation of the mRPI set  $\mathcal{E}$  computed using the iterative approach from [68]. For both  $\lambda \in \{2 \times 10^5, 10^6\}$ , the volume ratio of  $\tilde{\mathcal{E}}(\Phi_\varepsilon)$  compared to  $\mathcal{E}$  is  $V_r = 1.26$ . Additionally, the tightened state and input constraint sets are inner-approximations of  $\hat{\mathcal{X}}$  and  $\hat{\mathcal{U}}$  from (5.3c) and (5.3d) computed directly using the Pontryagin difference. For  $\lambda \in \{2 \times 10^5, 10^6\}$ , the volume ratios of  $\tilde{\mathcal{X}}(\Phi_x)$



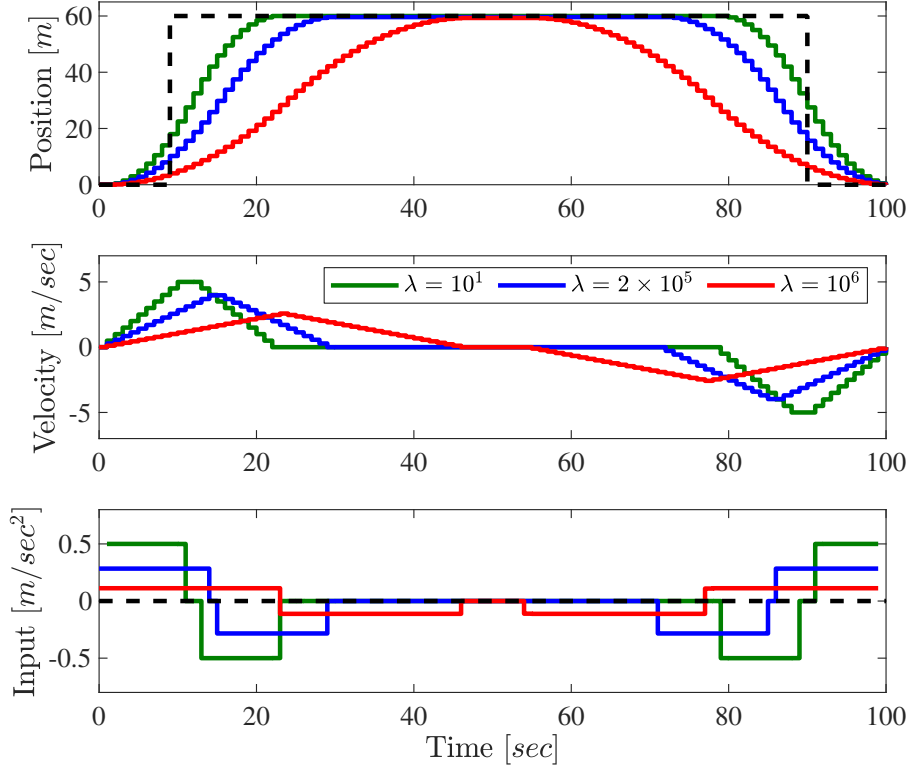


Figure 5.2. Simulation results for uncertainty weightings  $\lambda \in \{10^1, 2 \times 10^5, 10^6\}$  with the dashed lines denoting the reference trajectories for the position state and the acceleration/deceleration input.

compared to  $\hat{\mathcal{X}}$  are  $V_r = 0.99$  and  $V_r = 0.987$  while the volume ratios of  $\tilde{\mathcal{U}}(\Phi_u)$  compared to  $\hat{\mathcal{U}}$  are  $V_r = 0.94$  and  $V_r = 0.78$ .

Using YALMIP [57] and Gurobi [34] to formulate and solve (5.11), the mean and maximum computation times are 0.26 and 0.42 seconds over 100 runs on a laptop with a 2.2 GHz i5 processor with 8 GB of RAM. By comparison, solving (5.3) has a mean and maximum computation times of 0.09 and 0.34 seconds with an additional 1.19 seconds required for the offline computation of RPI sets and corresponding constraint tightening. While including the uncertainty set size as a decision variable clearly increases the computation time, the proposed approach is expected to remain computationally viable for systems with significantly more states and inputs.

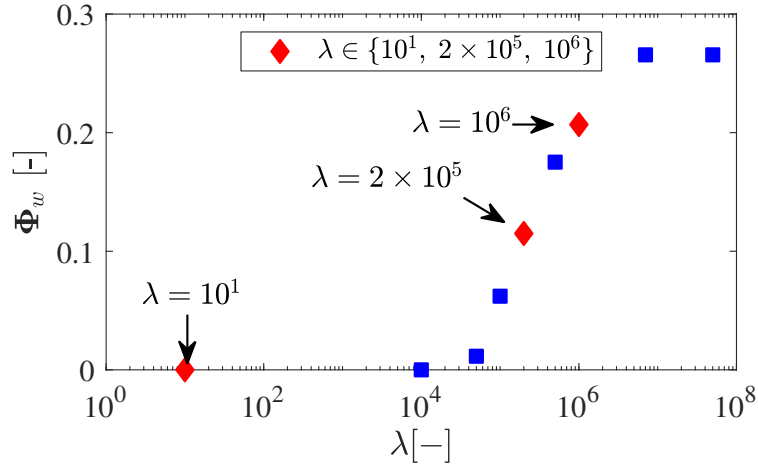


Figure 5.3. Relationship between the chosen value of  $\lambda$  and the resulting uncertainty set size captured by  $\Phi_w$ .

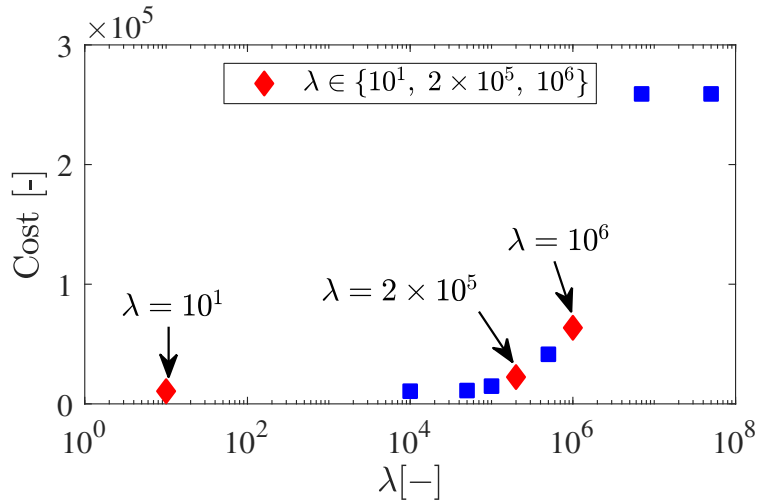


Figure 5.4. Relationship between the chosen value of  $\lambda$  and the resulting operational cost computed using (5.25).

## 5.8 Conclusions

A tube-based robust MPC formulation with integrated set computation is presented for constrained linear systems. The size of the uncertainty set is computed online in the underlying control optimization problem. One-step RPI set and Pontryagin difference methods formu-

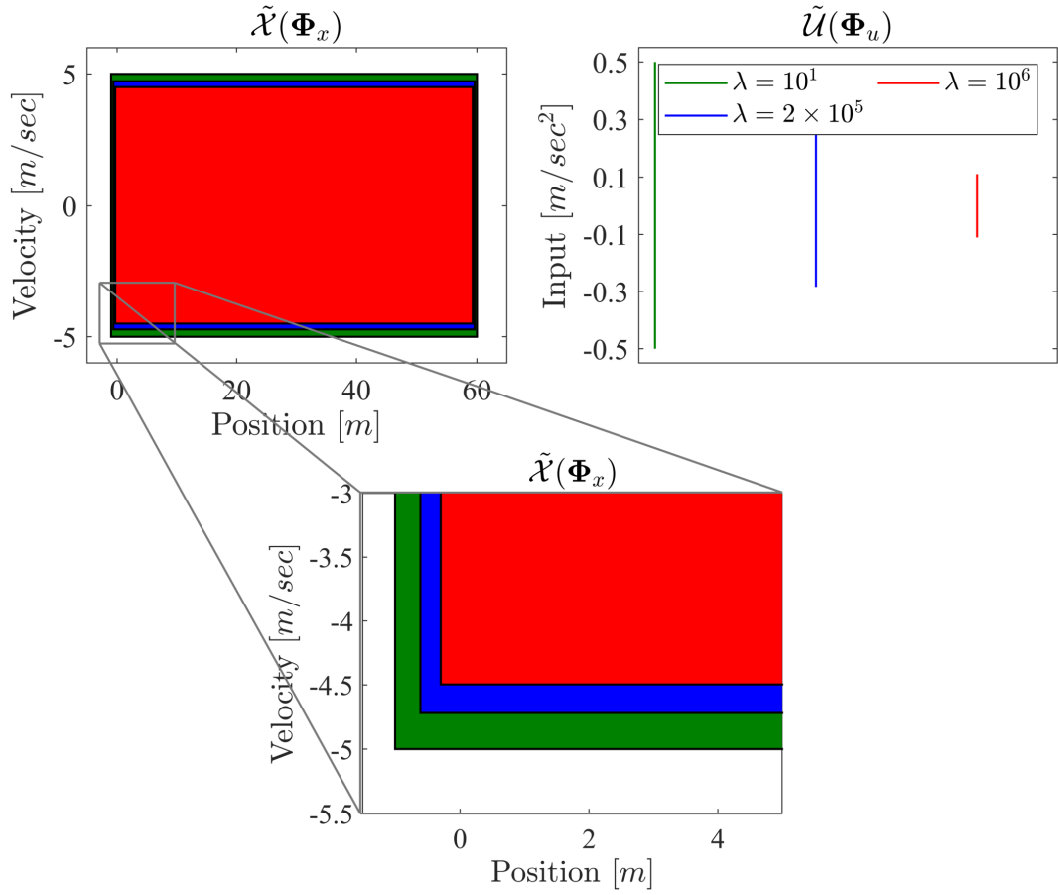


Figure 5.5. Tightened state and input constraint sets implicitly showing the effect of the increasing uncertainty set size for  $\lambda \in \{10^1, 2 \times 10^5, 10^6\}$ .

lated based on zonotopes and Hausdorff distance enabled online computation of RPI sets and tightened state and input constraint sets. A numerical example demonstrated the performance of the proposed tube-based robust MPC formulation and highlighted the benefits and limitations of embedding set calculations in the optimization problem. Future work will focus on the extension of the proposed tube-based robust MPC formulation to hierarchical MPC for online computation of disturbances between subsystems and between control decision at each level of the hierarchy which will be discussed in the following chapter.

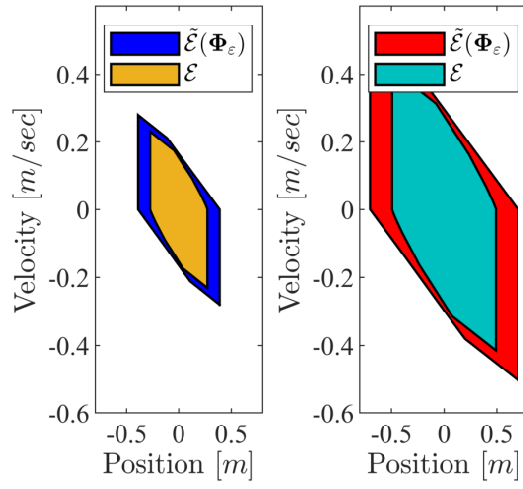


Figure 5.6. Comparison of  $\tilde{\mathcal{E}}(\Phi_\varepsilon)$  and mRPI set  $\mathcal{E}$  for  $\lambda \in \{2 \times 10^5, 10^6\}$ .

**CHAPTER 6**  
**HIERARCHICAL MPC FOR COUPLED SUBSYSTEMS**  
**USING ADJUSTABLE TUBES <sup>1</sup>**

**6.1 Introduction**

Model Predictive Control (MPC) of constrained dynamic systems provides the ability to satisfy both input and state constraints to guarantee safe and reliable system operation. This is particularly important for systems where the desired operation requires both transient and steady-state input and state trajectories to approach these constraints. Examples include the control of water distribution networks [90], aircraft power systems [80], smart power grids [43, 40], and hybrid electric vehicles [21, 91].

However, centralized MPC approaches are not well-suited for the control of these complex multi-timescale systems, where the system is comprised of many dynamically coupled subsystems and achieving the desired operation requires both fast control update rates and long prediction horizons.

For these complex systems, hierarchical MPC can be used to decompose control decision across multiple levels of controllers [73]. Typically, upper-level controllers are designed with large time step sizes to optimize system operation over long prediction horizons while lower-level controllers use small time step sizes to resolve the fast dynamics of the system over short prediction horizons. With a single controller per level, *vertical* hierarchical MPC is a computationally efficient approach for controlling multi-timescale systems with a relatively low number of states and inputs [49]. For more complex systems, comprised of multiple dynamically-coupled subsystems, *full* hierarchical MPC utilizes multiple controllers at each of the lower-levels to reduce the number of control decisions per controller [9, 24, 26].

---

<sup>1</sup>This chapter is based on work supported by the National Science Foundation under grant 1849500. Chapter 6 in part is a reprint of material published in: © 2022 Automatica. Reprinted, with permission, from Raghuraman, V. and Koeln, J.P., 2021. Hierarchical MPC for coupled subsystems using adjustable tubes. Automatica, 143, p.110435.

Many hierarchical MPC formulations [26, 24, 9, 86] have been developed to provide theoretical guarantees for the closed-loop system. Specifically, the two-level hierarchical controller in [9] with an upper-level MPC and a lower-level linear controller achieves state and input constraint satisfaction through communication of optimal references and reference rate changes between controller levels and also guarantees closed-loop stability. The controller developed in [86] provides guaranteed persistent controller feasibility and closed-loop stability for a cascaded system with actuator dynamics subject to input constraints. In this case, coordination is achieved through the appropriate choice of contractive terminal constraint sets and terminal control laws, which overall guarantee stability of the error between inner-loop and outer-loop reference models to the origin. The works in [24, 26] extend the vertical hierarchical architecture to a full two-level hierarchical controller with one upper-level controller and multiple controllers at the lower-level, one for each subsystem operating at the same timescale in [24] and different timescales in [26], and guarantees closed-loop stability and input constraint satisfaction while driving the system to a desired steady state. While [24] drives the system to a desired set around a steady-state equilibrium, the works in [26, 9, 86] guarantee convergence to the exact steady-state equilibrium. However, for systems with finite operation, such steady-state equilibrium might not exist, as in the case of systems whose operation is based on the utilization of a finite resource (e.g. battery state of charge in an electric vehicle [91, 72] or fuel in an aircraft [20]).

Similar to [70, 49], this work focuses on the notion of *completion*, with the goal of maximizing transient performance by satisfying state, input, and terminal constraints during system operation. While the multi-rate hierarchical MPC proposed in [26] achieves real-time computational performance using a *full* hierarchical MPC architecture with a reduced-order model at the upper-level, guarantees on closed-loop state constraint satisfaction are not shown explicitly. Additionally, the amount of control flexibility provided to the upper- and lower-level controllers along with the resulting uncertainty sets, Robust Positive Invariant

(RPI) sets, and tightened constraint sets are determined offline and might not be the optimal choice for systems that need a time-varying control flexibility. Moreover, guaranteed convergence might not be possible for a wide range of systems due to underlying assumptions on the slow timescale of the upper-level controller. To address these challenges, this work focuses on development of a set-based hierarchical MPC architecture for linear systems of dynamically-coupled subsystems that guarantees state and input constraint satisfaction.

One of the fundamental considerations for coordination in hierarchical MPC is how to provide lower-level controllers the flexibility to use their fast update rates and the fast dynamics of the system to improve upon the control decisions made by upper-level controllers without introducing unnecessary conservatism to account for this flexibility. In the authors' prior work [49], set-based vertical hierarchical MPC was proposed, where *waysets* were used as the primary coordination mechanism to provide both control performance and guaranteed constraint satisfaction. Strategically designed terminal costs were added to complement the waysets to guarantee that the lower-level controllers can only improve control performance compared to the upper-level controller trajectories [67].

For full hierarchical MPC of systems of dynamically-coupled subsystems, providing lower-level controllers the flexibility to deviate from the trajectories planned by upper-level controllers introduces uncertainty between subsystems. Therefore, the desired degree of flexibility balances the benefits of allowing lower-level controller to improve control performance within their own subsystem with the cost of creating unknown disturbances for neighboring subsystems. This trade off can be time-varying, where certain system operations might require a high level of coordination between subsystems, resulting in very little flexibility for lower-level subsystem controllers to deviate from the upper-level system-wide control plan. Alternatively, other system operations might not require much coordination between subsystems and lower-level controllers should be permitted a high degree of flexibility to further improve control performance.

The proposed two-level hierarchical MPC framework provides this time-varying subsystem coordination flexibility using an adjustable tube set-based coordination mechanism. Specifically, while planning system state and input trajectories, the upper-level controller simultaneously optimizes the permissible deviations from these trajectories provided to the lower-level subsystem controllers and the corresponding constraint tightening needed to be robust to these deviations. These time-varying permissible deviations are communicated to the lower-level controllers that use this flexibility to further optimize subsystem operation. The ability to embed the optimization of these permissible deviation bounds within the upper-level MPC optimization problem is enabled by *zonotopes* and the recent work on computing RPI sets and Pontryagin difference set operations using linear constraints [66, 65].

*Contributions:* The specific contributions of this research are: (1) the development a two-level hierarchical framework with  $M$  lower-level controllers, one for each of the  $M$  dynamically-coupled subsystems; (2) the definition and the use of adjustable tubes to provide time-varying bounds on permissible deviations between upper-level and lower-level planned trajectories; (3) the closed-loop analysis of the hierarchical controller to prove controller feasibility and guarantee constraint satisfaction; and (4) a numerical demonstration of the capabilities of the proposed approach. Note that the proposed work extends the tube-based robust MPC with uncertainty set optimization from [65] to a hierarchical MPC framework with optimal allocation of uncertainty quantified as the differences in control decisions between controller levels and between subsystems. Similar to [65], RPI, tightened output, and tightened terminal sets corresponding to the optimized uncertainty are computed online while solving the control optimization problem.

## 6.2 Notation and Preliminaries

For a system comprised of multiple subsystems, system-level vectors are denoted in bold, e.g. state  $\mathbf{x}$  and input  $\mathbf{u}$ , while vectors of the  $i^{th}$  subsystem have the subscript  $i$ , e.g. state



$x_i$  and input  $u_i$ . The system state vector is formed by the concatenation of subsystem state vectors as  $\mathbf{x} = [x_i]$ . Alternatively, the states of subsystem  $i$  can be extracted from the system state vector as  $x_i = \Pi_i \mathbf{x}$ . For a discrete-time system,  $\mathbf{x}(k)$  denotes the state  $\mathbf{x}$  at time step  $k$ . With  $[k, k + N - 1]$  denoting the integers from  $k$  to  $k + N - 1$ , the input trajectory over these time steps is denoted  $\{\mathbf{u}(j)\}_{j=k}^{k+N-1}$ . For MPC, the double index notation  $\mathbf{x}(k + l|k)$  denotes the predicted state at future time  $k + l$  determined at time step  $k$ . The block-diagonal matrix  $K$  with blocks  $K_i$  is denoted  $K = \text{diag}(K_i)$ . The  $p$ -norm of a vector is denoted  $\|\cdot\|_p$  and the weighted norm is  $\|\mathbf{x}\|_\Lambda^2 = \mathbf{x}^T \Lambda \mathbf{x}$ , where  $\Lambda$  is a positive-definite diagonal matrix. All sets are shown in caligraphic font. For sets  $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^n$ ,  $\mathcal{X} \oplus \mathcal{Y}$  denotes the Minkowski sum and  $\mathcal{X} \ominus \mathcal{Y}$  denotes the Minkowski/Pontryagin difference of  $\mathcal{Y}$  from  $\mathcal{X}$ . The Cartesian product of sets is denoted as  $\mathcal{X} \times \mathcal{Y}$ . The projection of  $\mathcal{X}$  on the  $n_i$  dimensions of subsystem  $i$  is denoted as  $\mathcal{X}_i = \Pi_i \mathcal{X}$ .

### 6.3 Problem Formulation

Consider a linear discrete time-invariant system composed of  $M$  dynamically-coupled subsystems,  $\mathbf{S}_i$ , where  $i \in \mathcal{N} \triangleq [1, M]$ . The dynamics of subsystem  $\mathbf{S}_i$  are

$$x_i(k + 1) = A_{ii}x_i(k) + B_{ii}u_i(k) + w_i(k), \quad (6.1a)$$

$$y_i(k) = C_i x_i(k) + D_i u_i(k), \quad (6.1b)$$

where  $x_i \in \mathbb{R}^{n_i}$  are the states,  $u_i \in \mathbb{R}^{m_i}$  are the inputs, and  $y_i \in \mathbb{R}^{n_i + m_i}$  are the outputs.

The coupling between subsystems is captured by the disturbance vector

$$w_i(k) = \sum_{j \in \mathcal{N}_i} (A_{ij}x_j(k) + B_{ij}u_j(k)), \quad (6.2)$$

where  $\mathcal{N}_i$  is the set of neighboring subsystems such that

$$\mathcal{N}_i \triangleq \{j \in \mathcal{N} \setminus \{i\} : [A_{ij} \ B_{ij}] \neq 0\}. \quad (6.3)$$

The outputs are defined to include all states and inputs such that  $y_i(k) \triangleq [x_i(k)^\top u_i(k)^\top]^\top$  and  $[C_i D_i] \triangleq I_{n_i+m_i}$ . The subsystem states, inputs, and outputs are constrained such that

$$x_i(k) \in \mathcal{X}_i, u_i(k) \in \mathcal{U}_i, y_i(k) \in \mathcal{Y}_i \triangleq \mathcal{X}_i \times \mathcal{U}_i. \quad (6.4)$$

Based on (6.1) and (6.2), the full system dynamics are

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k), \quad (6.5a)$$

$$\mathbf{y}(k) = C\mathbf{x}(k) + D\mathbf{u}(k), \quad (6.5b)$$

where  $\mathbf{x} = [x_i] \in \mathbb{R}^n$ ,  $\mathbf{u} = [u_i] \in \mathbb{R}^m$ , and  $\mathbf{y} = [y_i] \in \mathbb{R}^{n+m}$ , such that  $n = \sum_{i=1}^M n_i$  and  $m = \sum_{i=1}^M m_i$ . The system constraints are

$$\mathbf{x}(k) \in \mathcal{X} \triangleq \mathcal{X}_1 \times \cdots \times \mathcal{X}_M, \quad (6.6a)$$

$$\mathbf{u}(k) \in \mathcal{U} \triangleq \mathcal{U}_1 \times \cdots \times \mathcal{U}_M, \quad (6.6b)$$

$$\mathbf{y}(k) \in \mathcal{Y} \triangleq \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_M. \quad (6.6c)$$

Let  $A_D \triangleq \text{diag}(A_{ii})$  and  $B_D \triangleq \text{diag}(B_{ii})$  be block diagonal matrices while  $A_C \triangleq A - A_D$  and  $B_C \triangleq B - B_D$  are off-diagonal matrices that capture the coupling between subsystems.

**Assumption 11.** *There exists a static feedback control gain  $K_i \in \mathbb{R}^{m_i \times n_i}$  for each subsystem  $\mathbf{S}_i$ ,  $i \in \mathcal{N}$ , such that  $A_{ii} + B_{ii}K_i$  is Schur stable and  $A + BK$  is Schur stable, where  $K = \text{diag}(K_i)$  is a block-diagonal matrix.*

**Remark 8.** *For systems with weak dynamic subsystem coupling, the control gain  $K_i$ ,  $\forall i \in \mathcal{N}$ , satisfying **Assumption 11** can often be obtained using decentralized control design methods such as LQR or pole placement. For systems comprised of more strongly coupled subsystems, control gains satisfying **Assumption 11** may potentially be found by solving a set of Linear Matrix Inequalities (LMIs) based on [12]. However, for highly-coupled systems, it may not be possible to satisfy **Assumption 11** and a control approach that requires the decomposition of the system into subsystems may not be practical.*

**Assumption 12.** *With a fixed time step  $\Delta t$ , the system operates for a finite length of time starting from  $t = 0$  and ending at  $t = t_F = k_F \Delta t$  with time steps indexed by  $k \in [0, k_F]$ .*

Starting from an initial condition  $\mathbf{x}(0)$ , the goal is to plan and execute an input trajectory and corresponding state and output trajectories satisfying the system dynamics from (6.5), the constraints from (6.6) for all  $k \in [0, k_F - 1]$ , and the terminal constraint

$$\mathbf{x}(k_F) \in \mathcal{T} \triangleq \mathcal{T}_1 \times \cdots \times \mathcal{T}_M \subseteq \mathcal{X}. \quad (6.7)$$

**Assumption 13.** *The sets  $\mathcal{X}_i$ ,  $\mathcal{U}_i$ , and  $\mathcal{T}_i$ ,  $i \in \mathcal{N}$ , are zonotopes.*

The generic cost function

$$J(\mathbf{x}(0)) = \sum_{j=0}^{k_F-1} \ell(j) + \ell_F(k_F), \quad (6.8)$$

defines the cost of system operation using a pre-determined reference trajectory  $\{\mathbf{r}(k)\}_{k=0}^{k_F}$  with stage costs  $\ell(j) = \ell(\mathbf{x}(j), \mathbf{u}(j), \mathbf{r}(j))$  and terminal cost  $\ell_F(k_F) = \ell_F(\mathbf{x}(k_F), \mathbf{r}(k_F))$ .

Considering the full system (6.5), operational constraints (6.6), terminal constraint (6.7), and cost function (6.8), this research develops a two-level hierarchical control approach with  $M$  controllers at the lower-level that guarantees constraint satisfaction and provides computational efficiency in the case of a large number of subsystems  $M$ , small time step size  $\Delta t$ , and large operating duration  $t_F$ .

## 6.4 Hierarchical Control

The proposed hierarchical control formulation consists of a single controller  $\mathbf{C}_0$  in the upper-level and  $M$  controllers  $\mathbf{C}_i, i \in \mathcal{N}$ , in the lower-level, where  $\mathbf{C}_i$  controls subsystem  $\mathbf{S}_i$ .

**Assumption 14.** *The controller  $\mathbf{C}_0$  has a time step size  $\Delta t_0$  and maximum prediction horizon  $\bar{N}_0$  such that  $\Delta t_0 \bar{N}_0 = t_F$ . Each controller  $\mathbf{C}_i, i \in \mathcal{N}$ , has a time step size  $\Delta t$  and maximum prediction horizon  $\bar{N}$  such that  $\Delta t \bar{N} = \Delta t_0$ .*

Let  $\nu_0 \triangleq \frac{\Delta t_0}{\Delta t} = \bar{N} \in \mathbb{Z}_+$  be defined as a time scaling factor for  $\mathbf{C}_0$ . The time steps for  $\mathbf{C}_0$  are indexed by  $k_0$ , with  $k_0 \triangleq \frac{k}{\nu_0}$ , and let  $k_{0,F} \triangleq \frac{k_F}{\nu_0} = \bar{N}_0$  denote the terminal step of  $\mathbf{C}_0$  such that  $k_0 \in [0, k_{0,F}]$ . Thus, the upper-level controller  $\mathbf{C}_0$  has a *shrinking horizon*, with time-varying horizon length  $N_0(k_0) \triangleq \bar{N}_0 - k_0$ . Each lower-level controller  $\mathbf{C}_i$  has a *shrinking and resetting horizon*, with horizon length  $N(k) \triangleq \bar{N} - (k \bmod \bar{N})$ . This allows  $\mathbf{C}_i$  to predict between updates of  $\mathbf{C}_0$ , at which point  $(k \bmod \bar{N} = 0)$  and the prediction horizon resets back to  $N(k) = \bar{N}$ .

Similar to [48, 49],  $\mathbf{C}_0$  predicts coarse state and input trajectories at time indices  $k_0$  with a large time step size  $\Delta t_0$ . Lower-level controllers  $\mathbf{C}_i$  are permitted bounded deviations from the trajectories planned by  $\mathbf{C}_0$  to further improve control performance using a smaller time step size  $\Delta t$ . Unlike [48, 49], this work addresses the coupling between subsystems. If the lower-level controller  $\mathbf{C}_i$  chooses to deviate from the state and input trajectories planned by  $\mathbf{C}_0$ , these deviations create unknown disturbances that could lead to constraint violations in neighboring subsystems. Therefore, instead of using waysets as in [48, 49], a tube-based coordination mechanism is used to bound the permissible deviations between the trajectories planned by  $\mathbf{C}_0$  and those planned by  $\mathbf{C}_i$ . Moreover, the size of these permissible deviations is optimized online by  $\mathbf{C}_0$  to balance the flexibility given to lower-level controllers with the potentially time-varying need for close coordination among subsystems.

Specifically, for each subsystem, the sets  $\Delta\mathcal{Z}_i(\delta_i^z(k_0))$  and  $\Delta\mathcal{V}_i(\delta_i^v(k_0))$  denote scaled zonotopes that bound the permissible state and input deviations between the trajectories planned by  $\mathbf{C}_0$  and those planned by  $\mathbf{C}_i$ . The scaling vectors can be collected to form the output deviation vector  $\delta_i(k_0) = [\delta_i^z(k_0)^\top \delta_i^v(k_0)^\top]^\top$  and the permissible output deviation set

$$\Delta\mathcal{Y}_i(\delta_i(k_0)) = \Delta\mathcal{Z}_i(\delta_i^z(k_0)) \times \Delta\mathcal{V}_i(\delta_i^v(k_0)). \quad (6.9)$$

To reduce notational complexity, the shorthand  $\Delta\mathcal{Y}_i(k_0) = \Delta\mathcal{Y}_i(\delta_i(k_0))$  is used when explicitly stating the dependency on  $\delta_i(k_0)$  is unnecessary. The system state, input, and output

deviation vectors are  $\boldsymbol{\delta}^z(k_0) = [\delta_i^z(k_0)]$ ,  $\boldsymbol{\delta}^v(k_0) = [\delta_i^v(k_0)]$ , and  $\boldsymbol{\delta}(k_0) = [\boldsymbol{\delta}^z(k_0)^\top \boldsymbol{\delta}^v(k_0)^\top]^\top$  and the scaled subsystem deviation sets combine to form the scaled system deviation sets

$$\Delta\mathcal{Z}(\boldsymbol{\delta}(k_0)) = \Delta\mathcal{Z}_1(k_0) \times \cdots \times \Delta\mathcal{Z}_M(k_0), \quad (6.10a)$$

$$\Delta\mathcal{V}(\boldsymbol{\delta}(k_0)) = \Delta\mathcal{V}_1(k_0) \times \cdots \times \Delta\mathcal{V}_M(k_0), \quad (6.10b)$$

$$\Delta\mathcal{Y}(\boldsymbol{\delta}(k_0)) = \Delta\mathcal{Z}(\boldsymbol{\delta}(k_0)) \times \Delta\mathcal{V}(\boldsymbol{\delta}(k_0)). \quad (6.10c)$$

The controller  $\mathbf{C}_0$  updates only when  $k = \nu_0 k_0$  (i.e. when  $k \bmod \nu_0 = 0$ ), by solving the constrained optimization problem  $\mathbf{P}_0(\mathbf{x}(k))$  defined as

$$J_0^*(\mathbf{x}(k)) = \min_{\substack{\hat{\mathbf{x}}(k_0|k_0), \hat{\mathbf{u}}(k_0), \\ \boldsymbol{\delta}(k_0)}} \sum_{j=k_0}^{k_{0,F}-1} \ell(j|k_0) + \ell_F(k_{0,F}), \quad (6.11a)$$

$$\text{s.t. } \forall j \in [k_0, k_{0,F} - 1],$$

$$\hat{\mathbf{x}}(j+1|k_0) = A_0 \hat{\mathbf{x}}(j|k_0) + B_0 \hat{\mathbf{u}}(j|k_0), \quad (6.11b)$$

$$\hat{\mathbf{y}}(j|k_0) = C \hat{\mathbf{x}}(j|k_0) + D \hat{\mathbf{u}}(j|k_0) \in \hat{\mathcal{Y}}_0(\boldsymbol{\delta}(k_0)), \quad (6.11c)$$

$$\hat{\mathbf{x}}(k_{0,F}|k_0) \in \hat{\mathcal{T}}_0(\boldsymbol{\delta}(k_0)), \quad (6.11d)$$

$$\mathbf{x}(k) - \hat{\mathbf{x}}(k_0|k_0) \in \Delta\mathcal{Z}(\boldsymbol{\delta}(k_0)) \oplus \mathcal{E}_0(\boldsymbol{\delta}(k_0)), \quad (6.11e)$$

$$\Delta\mathcal{Z}(\boldsymbol{\delta}(k_0)) \subseteq \text{Pre}(\Delta\mathcal{Z}(\boldsymbol{\delta}(k_0))). \quad (6.11f)$$

The shrinking horizon of  $\mathbf{P}_0(\mathbf{x}(k))$  is reflected in the summation limits in (6.11a). The stage costs are defined as  $\ell(j|k_0) = \ell(\mathbf{x}(k), \hat{\mathbf{x}}(j|k_0), \hat{\mathbf{u}}(j|k_0), \boldsymbol{\delta}(k_0), \mathbf{r}_0(j))$  to be a function of the measured state, nominal state, nominal input, permissible deviations for lower-level controllers, and the reference trajectory. The reference trajectory  $\mathbf{r}_0(j)$  can be obtained by downsampling the predetermined reference trajectory  $\mathbf{r}(j)$  either using averaging or zero order hold [48]. The terminal cost  $\ell_F(k_{0,F})$  is the same as in (6.8). Note that the system performance can be balanced with the maximization of  $\boldsymbol{\delta}$  through an additional cost function term  $\Lambda \|\bar{\boldsymbol{\delta}} - \boldsymbol{\delta}\|_p$ , where  $\Lambda$  is a scalar weighting term and  $\bar{\boldsymbol{\delta}}$  is a user-specified upper-bound on

$\delta$ . The nominal input trajectory is defined as  $\hat{\mathbf{U}}(k_0) = \{\hat{\mathbf{u}}(j|k_0)\}_{j=k_0}^{k_0, F-1}$ . The permissible deviation scaling vector  $\delta(k_0)$  affects the sizes of the tightened output constraint set  $\hat{\mathcal{Y}}_0(\delta(k_0))$ , the tightened terminal constraint set  $\hat{\mathcal{T}}_0(\delta(k_0))$ , the state deviation constraint set  $\Delta\mathcal{Z}(\delta(k_0))$ , and the RPI set  $\mathcal{E}_0(\delta(k_0))$ , as discussed in Section 6.5. In (6.11b), the model used by  $\mathbf{C}_0$  assumes a piecewise constant control input over the time step size  $\Delta t_0$  and thus  $A_0 = A^{\nu_0}$  and  $B_0 = \sum_{j=0}^{\nu_0-1} A^j B$  (as in [74]). In (6.11c) and (6.11d), the outputs and terminal state are constrained to the time-varying tightened output and terminal constraint sets. Similar to tube-based MPC [60], (6.11e) allows  $\mathbf{C}_0$  flexibility in the choice of initial condition  $\hat{\mathbf{x}}(k_0|k_0)$ , which is used to prove recursive feasibility of  $\mathbf{P}_0(\mathbf{x}(k))$  in Section 6.6. Finally, (6.11f) constrains the time-varying permissible state deviation set to be a subset of its own precursor set. Based on the definition from [14], the precursor set is defined specifically as

$$\text{Pre}(\Delta\mathcal{Z}(k_0)) = \{\mathbf{z} \mid \exists \mathbf{v} \in \Delta\mathcal{V}(k_0) \text{ s.t. } A_D \mathbf{z} + B_D \mathbf{v} \in \Delta\mathcal{Z}(k_0)\}, \quad (6.12)$$

and is used to establish feasibility of the lower-level controllers in Section 6.6. Note that the RPI set  $\mathcal{E}_0(\delta(k_0))$  is assumed to be a *structured* RPI set such that

$$\mathcal{E}_0(\delta(k_0)) = \mathcal{E}_1(\delta_1(k_0)) \times \cdots \times \mathcal{E}_M(\delta_M(k_0)), \quad (6.13)$$

and is formulated in more detail in Section 6.5. This structure has been used in distributed robust MPC [85] and imposes an inherent limit on the coupling between subsystems.

The lower-level controllers  $\mathbf{C}_i$ ,  $i \in \mathcal{N}$ , update at each time index  $k$  by each solving, in parallel, the constrained optimization problems  $\mathbf{P}_i(x_i(k))$ , defined as

$$J_i^*(x_i(k)) = \min_{\substack{z_i(k|k), j=k \\ v_i(k)}} \sum_{j=k}^{k+N(k)-1} \ell_i(j|k) + \ell_{i,F}(k+N(k)), \quad (6.14a)$$

$$\text{s.t. } \forall j \in [k, k+N(k)-1],$$

$$z_i(j+1|k) = A_{ii} z_i(j|k) + B_{ii} v_i(j|k) + \hat{w}_i^*(j), \quad (6.14b)$$

$$y_i(j|k) = C_i z_i(j|k) + D_i v_i(j|k), \quad (6.14c)$$

$$y_i(j|k) - \hat{y}_i^*(j) \in \Delta\mathcal{Y}_i(\delta_i^*(k_0)), \quad (6.14d)$$

$$z_i(k + N(k)|k) - \hat{x}_i^*(k + N(k)) \in \Delta\mathcal{Z}_i(\delta_i^*(k_0)), \quad (6.14e)$$

$$x_i(k) - z_i(k|k) \in \mathcal{E}_i(\delta_i^*(k_0)). \quad (6.14f)$$

The shrinking and resetting horizon of  $\mathbf{P}_i(x_i(k))$  is reflected in the summation limits in (6.14a). The stage costs are defined as  $\ell_i(j|k) = \ell_i(x_i(k), z_i(j|k), v_i(j|k), r_i(j))$  to be a function of the measured subsystem state, nominal subsystem state, nominal subsystem input, and subsystem reference trajectory. The terminal cost is defined as  $\ell_{i,F}(k + N(k)) = \ell_{i,F}(z_i(k + N(k)|k), r_i(k + N(k)))$ . The nominal input trajectory is defined as  $V_i(k) = \{v_i(j|k)\}_{j=k}^{k+N(k)-1}$ . In (6.14b), the subsystem dynamics from (6.1a) are used with a time-varying  $\mathbf{C}_0$ -optimal disturbance  $\hat{w}_i^*(j)$  that is communicated from  $\mathbf{C}_0$  (details in Section 6.5). Nominal subsystem outputs are defined in (6.14c) and the differences between these outputs and the  $\mathbf{C}_0$ -optimal outputs  $\hat{y}_i^*(j)$  are constrained in (6.14d) to the time-varying permissible output deviation set  $\Delta\mathcal{Y}_i(\delta_i^*(k_0))$  (details in Section 6.5). Similarly, the difference between the nominal terminal state and the  $\mathbf{C}_0$ -optimal terminal state is constrained to the time-varying permissible state deviation set  $\Delta\mathcal{Z}_i(\delta_i^*(k_0))$  in (6.14e). Finally, (6.14f) provides flexibility in initial condition  $z_i(k|k)$  based on the RPI set from (6.13).

As shown in Fig. 6.1, coordination between the upper-level controller  $\mathbf{C}_0$  and lower-level controllers  $\mathbf{C}_i$ ,  $i \in \mathcal{N}$ , is achieved through the communication of the  $\mathbf{C}_0$ -optimal trajectories  $\hat{y}_i^*(j)$  and  $\hat{w}_i^*(j)$ ,  $j \in [k, k + N(k) - 1]$ , terminal state  $\hat{x}_i^*(k + N(k))$ , and the time-varying permissible deviation vectors  $\delta_i^*(k_0)$ . In this hierarchical control architecture, only the lower-level controllers  $\mathbf{C}_i$  directly affect the system through inputs to the subsystems  $\mathbf{S}_i$ . Once each  $\mathbf{C}_i$  has solved for the optimal nominal input trajectories  $V_i^*(k)$  and optimal nominal initial condition  $z_i^*(k|k)$ , the input to the system is  $\mathbf{u}(k) = [u_i(k)]$  where

$$u_i(k) = v_i^*(k|k) + K_i(x_i(k) - z_i^*(k|k)). \quad (6.15)$$

The two-level hierarchical controller is implemented based on **Algorithm 7**. The specific formulation of the sets in (6.11) and (6.14) are presented in Section 6.5 and the corresponding constraints are used to guarantee satisfaction of the state, input, and terminal constraints from (6.6) and (6.7) in Section 6.6.

---

**Algorithm 7:** Two-level Hierarchical MPC with subsystem coupling.

---

```

1 Initialize  $k, k_0 \leftarrow 0$ 
2 if  $k \bmod \nu_0 = 0$  then
3   | solve  $\mathbf{P}_0(\mathbf{x}(k))$ ;
4   | communicate  $\{\hat{y}_i^*(j)\}_{j=k}^{k+N(k)-1}$ ,  $\{\hat{w}_i^*(j)\}_{j=k}^{k+N(k)-1}$ ,  $\hat{x}_i^*(k+N(k))$ , and  $\delta_i^*(k_0)$  to
5   |    $\mathbf{P}_i(x_i(k)), \forall i \in \mathcal{N}$ ;
6   |  $k_0 \leftarrow k_0 + 1$ ;
7 end
7 solve  $\mathbf{P}_i(x_i(k)), \forall i \in \mathcal{N}$ , and apply the input  $\mathbf{u}(k) = [u_i(k)]$  to the system based
   on (6.15);
8  $k \leftarrow k + 1$ ;
```

---

## 6.5 Nominal Trajectories and Error Propagation

Following the tube-based MPC framework in [60], the goal of this section is to explicitly bound the differences between the nominal trajectories planned by the controllers  $\mathbf{C}_0$  and  $\mathbf{C}_i, i \in \mathcal{N}$ , and true system trajectories.

First, since  $\mathbf{C}_0$  has a larger time step size than  $\mathbf{C}_i$  and system dynamics (i.e.  $\Delta t_0 > \Delta t$ ), the input and state trajectories determined by  $\mathbf{C}_0$  must be upsampled. Let  $\hat{\mathbf{u}}^*(k)$  and  $\hat{\mathbf{x}}^*(k)$  be the upsampled input and state trajectories corresponding to the optimal trajectories determined by  $\mathbf{C}_0$ . Since the model (6.11b) assumed a piecewise constant input, the upsampled trajectories are computed as the forward simulation of (6.5a) such that

$$\hat{\mathbf{u}}^*(k) = \hat{\mathbf{u}}^*(k_0|k_0), \quad (6.16a)$$

$$\hat{\mathbf{x}}^*(k) = A^{k-\nu_0 k_0} \hat{\mathbf{x}}^*(k_0|k_0) + \sum_{j=0}^{k-\nu_0 k_0-1} A^j B \hat{\mathbf{u}}^*(k_0|k_0), \quad (6.16b)$$



for  $k \in [\nu_0 k_0, \nu_0(k_0+1) - 1]$ . These trajectories create the  $\mathbf{C}_0$ -optimal output and disturbance trajectories  $\hat{y}_i^*(k)$  and  $\hat{w}_i^*(k)$  used in (6.14d) and (6.14b), where  $\hat{u}_i^*(k) = \Pi_i \hat{\mathbf{u}}^*(k)$ ,  $\hat{x}_i^*(k) = \Pi_i \hat{\mathbf{x}}^*(k)$ , and

$$\hat{y}_i^*(k) = [\hat{x}_i^*(k)^\top \hat{u}_i^*(k)^\top]^\top, \quad (6.17a)$$

$$\hat{w}_i^*(k) = \sum_{j \in \mathcal{N}_i} (A_{ij} \hat{x}_j^*(k) + B_{ij} \hat{u}_j^*(k)). \quad (6.17b)$$

Let  $\Delta \mathbf{x}(k) = [\Delta x_i(k)]$ ,  $\Delta \mathbf{u}(k) = [\Delta u_i(k)]$ , and  $\Delta \mathbf{y}(k) = [\Delta y_i(k)]$  denote the state, input, and output prediction errors for  $\mathbf{C}_0$ , where

$$\begin{aligned} \Delta x_i(k) &\triangleq x_i(k) - \hat{x}_i^*(k), & \Delta u_i(k) &\triangleq u_i(k) - \hat{u}_i^*(k), \\ \Delta y_i(k) &\triangleq y_i(k) - \hat{y}_i^*(k) = [\Delta x_i(k)^\top \Delta u_i(k)^\top]^\top. \end{aligned}$$

These upper-level prediction errors consist of two parts, corresponding to the planned deviations by lower-level controllers  $\mathbf{C}_i$  and the resulting lower-level prediction errors due to coupling between subsystems. Specifically,

$$\Delta x_i(k) = \Delta z_i(k) + e_i(k), \quad (6.19a)$$

$$\Delta u_i(k) = \Delta v_i(k) + K_i e_i(k), \quad (6.19b)$$

where

$$\Delta z_i(k) \triangleq z_i(k) - \hat{x}_i^*(k), \quad \Delta v_i(k) \triangleq v_i(k) - \hat{u}_i^*(k), \quad (6.20)$$

are the planned deviations and  $e_i(k) \triangleq x_i(k) - z_i(k)$ , are lower-level prediction errors due to the coupling between subsystems. Note that  $K_i e_i(k) = u_i(k) - v_i(k)$  based on the control law from (6.15).

**Lemma 11.** *Let the disturbance error set be defined as*

$$\Delta \mathcal{W} = A_C \Delta \mathcal{Z} \oplus B_C \Delta \mathcal{V}. \quad (6.21)$$

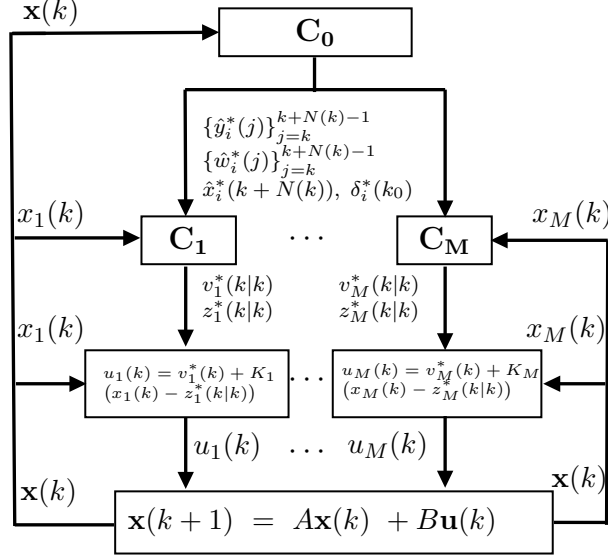


Figure 6.1. Two-level hierarchical MPC where  $\mathbf{C}_0$  is formulated based on (6.11) and  $\mathbf{C}_i$ ,  $i \in \mathcal{N}$ , based on (6.14). The  $\mathbf{C}_0$ -optimal trajectories  $\hat{y}_i^*(j)$  and  $\hat{w}_i^*(j)$  are computed using (6.16) and (6.17). The optimal output deviations  $\delta_i^*(k_0)$  are used to coordinate controllers  $\mathbf{C}_0$  and  $\mathbf{C}_i$ ,  $i \in \mathcal{N}$ , and the static feedback control law (6.15) computes the inputs to each subsystem  $\mathbf{S}_i$ .

Then the lower-level prediction errors  $\mathbf{e}(k) = [e_i(k)]$  are bounded to the RPI set  $\mathcal{E}_0 \subset \mathbb{R}^n$ , where  $\mathcal{E}_0$  satisfies

$$(A + BK)\mathcal{E}_0 \oplus \Delta\mathcal{W} \subseteq \mathcal{E}_0. \quad (6.22)$$

*Proof.* Using the true subsystem dynamics from (6.1a) and the nominal subsystem model from (6.14b), the lower-level prediction error dynamics for each subsystem are

$$e_i(k+1) = (A_{ii} + B_{ii}K_i)e_i(k) + \Delta w_i(k), \quad (6.23)$$

where  $\Delta w_i(k) = w_i(k) - \hat{w}_i^*(k)$ . Using the definitions of  $w_i(k)$  and  $\hat{w}_i^*(k)$  from (6.2) and (6.17b), this disturbance error is

$$\Delta w_i(k) = \sum_{j \in \mathcal{N}_i} (A_{ij}(x_j(k) - \hat{x}_j^*(k)) + B_{ij}(u_j(k) - \hat{u}_j^*(k))).$$

Based on (6.18) and (6.19), this disturbance error can be rewritten as

$$\Delta w_i(k) = \sum_{j \in \mathcal{N}_i} (A_{ij} + B_{ij}K_j)e_j(k) + A_{ij}\Delta z_j(k) + B_{ij}\Delta v_j(k). \quad (6.24)$$

Combining (6.23) and (6.24) for all subsystems  $i \in \mathcal{N}$  results in the system error dynamics

$$\mathbf{e}(k+1) = (A + BK)\mathbf{e}(k) + A_C \Delta \mathbf{z}(k) + B_C \Delta \mathbf{v}(k).$$

Since  $\Delta z_i(k) \in \Delta \mathcal{Z}_i$  and  $\Delta v_i(k) \in \Delta \mathcal{V}_i$ ,  $\forall i \in \mathcal{N}$ ,  $A_C \Delta \mathbf{z}(k) + B_C \Delta \mathbf{v}(k) \in \Delta \mathcal{W}$ , as defined in (6.21). Thus, if  $\mathbf{e}(k) \in \mathcal{E}_0$  and  $\mathcal{E}_0$  satisfies (6.22), then  $\mathbf{e}(k+1) \in \mathcal{E}_0$ .  $\square$

**Lemma 12.** *The upper-level prediction errors  $\Delta \mathbf{x}(k) = [\Delta x_i(k)]$  and  $\Delta \mathbf{u}(k) = [\Delta u_i(k)]$  are bounded such that*

$$\Delta \mathbf{x}(k) \in \Delta \mathcal{Z} \oplus \mathcal{E}_0, \quad \Delta \mathbf{u}(k) \in \Delta \mathcal{V} \oplus K\mathcal{E}_0. \quad (6.25)$$

*Proof.* The proof follows directly from the definitions of  $\Delta x_i(k)$  and  $\Delta u_i(k)$  from (6.19) and the result of **Lemma 11**.  $\square$

Based on the results of **Lemmas 11** and **12**, the nominal outputs determined by the upper-level controller in (6.11c) are constrained to the time-varying tightened output constraint set  $\hat{\mathcal{Y}}_0(\boldsymbol{\delta}(k_0))$ . Then the time-varying tightened output constraint set is defined as

$$\hat{\mathcal{Y}}_0(\boldsymbol{\delta}(k_0)) \triangleq \tilde{\mathcal{Y}}_0 \ominus [(\Delta \mathcal{Z} \oplus \mathcal{E}_0) \times (\Delta \mathcal{V} \oplus K\mathcal{E}_0)], \quad (6.26)$$

where  $\tilde{\mathcal{Y}}_0 \subseteq \mathcal{Y}$  is a tightened output constraint set used to prevent inter-sample constraint violations (see Appendix 6.9.1 for details on computing  $\tilde{\mathcal{Y}}_0$ ). Similarly, in (6.11d), the nominal terminal state is constrained to the time-varying tightened terminal constraint set  $\hat{\mathcal{T}}_0(\boldsymbol{\delta}(k_0))$  defined as

$$\hat{\mathcal{T}}_0(\boldsymbol{\delta}(k_0)) \triangleq \mathcal{T} \ominus (\Delta \mathcal{Z} \oplus \mathcal{E}_0). \quad (6.27)$$

**Remark 9.** *While the inter-sample constraint tightening for  $\mathbf{C}_0$  could be significant for underdamped higher-order systems, the resulting reduction in control performance can be alleviated through carefully choosing the time step size  $\Delta t_0$  of  $\mathbf{C}_0$ , while balancing the overall computational cost associated with a smaller time step size and the maximum prediction horizon  $\bar{N}_0$ .*

## 6.6 Hierarchical Control Feasibility

The following establishes recursive feasibility of each controller in the hierarchy and guarantees constraint satisfaction for the closed-loop system.

**Assumption 15.** *There exists a feasible solution to  $\mathbf{P}_0(\mathbf{x}(0))$  at time step  $k = k_0 = 0$  for the initial condition  $\mathbf{x}(0)$ .*

**Lemma 13.** *If  $\mathbf{P}_0(\mathbf{x}(k))$  is feasible time step  $k = \nu_0 k_0$ , then  $\mathbf{P}_i(x_i(k)), i \in \mathcal{N}$ , is feasible at this time step.*

*Proof.* Let  $\{\hat{\mathbf{x}}^*(j)\}_{j=k}^{k+N(k)}$ ,  $\{\hat{\mathbf{u}}^*(j)\}_{j=k}^{k+N(k)-1}$ , and  $\boldsymbol{\delta}^*(k_0)$  denote upsampled  $\mathbf{C}_0$ -optimal state and input trajectories and the optimal output deviation determined by  $\mathbf{C}_0$  at time step  $k = \nu_0 k_0$ . First, for all  $i \in \mathcal{N}$ , it is to be shown that there exists an initial condition  $\mathbf{z}(k|k) = [z_i(k|k)]$  that simultaneously satisfies the output deviation constraints from (6.14d) and the initial condition constraint from (6.14f). Specifically, the initial condition constraint (6.11e) ensures that  $\mathbf{x}(k) - \hat{\mathbf{x}}^*(k_0|k_0) \in \Delta\mathcal{Z}(\boldsymbol{\delta}^*(k_0)) \oplus \mathcal{E}_0(\boldsymbol{\delta}^*(k_0))$ . Based on the structure of these sets from (6.10a) and (6.13),  $x_i(k) - \hat{x}_i^*(k_0|k_0) \in \Delta\mathcal{Z}_i(\delta_i^*(k_0)) \oplus \mathcal{E}_i(\delta_i^*(k_0))$  for all  $i \in \mathcal{N}$ . Since these sets are all zonotopes, let  $\Delta\mathcal{Z}_i(\delta_i^*(k_0)) = \{G_i^z, 0\}$  and  $\mathcal{E}_i(\delta_i^*(k_0)) = \{G_i^\varepsilon, 0\}$ . Therefore,  $x_i(k) - \hat{x}_i^*(k_0|k_0) \in \Delta\mathcal{Z}_i(\delta_i^*(k_0)) \oplus \mathcal{E}_i(\delta_i^*(k_0))$  guarantees the existence of  $\xi_i^z$  and  $\xi_i^\varepsilon$  such that  $\|\xi_i^z\|_\infty \leq 1$ ,  $\|\xi_i^\varepsilon\|_\infty \leq 1$ , and

$$x_i(k) - \hat{x}_i^*(k_0|k_0) = G_i^z \xi_i^z + G_i^\varepsilon \xi_i^\varepsilon. \quad (6.28)$$

Choosing  $z_i(k|k) = \hat{x}_i^*(k_0|k_0) + G_i^z \xi_i^z$ , ensures that this initial condition satisfies the output deviation constraint from (6.14d). Solving for  $\hat{x}_i^*(k_0|k_0)$  and plugging into (6.28) results in  $x_i(k) - z_i(k|k) = G_i^\varepsilon \xi_i^\varepsilon$  and thus this choice of initial condition also satisfies the initial condition constraint from (6.14f).

It remains to show the existence of a candidate solution starting from this initial condition  $z_i(k|k)$ , denoted by the nominal input sequence  $\{v_i(j|k)\}_{j=k}^{k+N(k)-1}$  and corresponding nominal

state sequence  $\{z_i(j|k)\}_{j=k}^{k+N(k)}$ , that satisfies the model (6.14b) and the constraints (6.14c)-(6.14e). Comparing the candidate solution satisfying the nominal subsystem dynamics from (6.14b) and the upsampled  $\mathbf{C}_0$ -optimal trajectories satisfying the the true subsystem dynamics from (6.1a) results in  $z_i(j+1|k) - \hat{x}_i^*(j+1) = A_{ii}(z_i(j|k) - \hat{x}_i^*(j)) + B_{ii}(v_i(j|k) - \hat{u}_i^*(j))$  for all  $j \in [k, k + N(k) - 1]$ . Since  $z_i(k|k)$  has already been shown to satisfy the output deviation constraint from (6.14d),  $z_i(k|k) - \hat{x}_i^*(k) \in \Delta\mathcal{Z}_i(\delta_i^*(k_0))$ . From (6.11f) and the definition of the precursor set from (6.12), this guarantees the existence of  $v_i(k|k)$  such that  $v_i(k|k) - \hat{u}_i^*(k) \in \Delta\mathcal{V}_i(\delta_i^*(k_0))$  and  $z_i(k+1|k) - \hat{x}_i^*(k+1) \in \Delta\mathcal{Z}_i(\delta_i^*(k_0))$ . This process is repeated to show that the output deviation constraints from (6.14d) can be satisfied for all time steps and that the terminal constraint from (6.14e) is satisfied at the final time step.  $\square$

**Lemma 14.** *For all  $i \in \mathcal{N}$ , if  $\mathbf{P}_i(x_i(k))$  is feasible at time step  $k$ , where  $k \bmod \nu_0 = 0$  (i.e. at the time of  $\mathbf{C}_0$  update), then  $\mathbf{P}_i(x_i(k))$  is feasible at each time step  $k+1$  through  $k+N(k)-1$ .*

*Proof.* Let the feasible solution for  $\mathbf{P}_i(x_i(k))$  at time step  $k$  be defined by the optimal nominal input sequence  $\{v_i^*(j|k)\}_{j=k}^{k+N(k)-1}$  and corresponding optimal nominal state sequence  $\{z_i^*(j|k)\}_{j=k}^{k+N(k)}$  satisfying (6.14b). While (6.14f) guarantees  $x_i(k) - z_i^*(k|k) \in \mathcal{E}_i(\delta_i^*(k_0))$ , the feasibility of  $\mathbf{P}_j(x_j(k))$ ,  $j \in \mathcal{N} \setminus \{i\}$ , ensures that the disturbances  $\Delta w_i$  from (6.24) due to subsystem coupling are bounded to  $\Delta\mathcal{W}$  used to define  $\mathcal{E}_0 = \mathcal{E}_1 \times \dots \times \mathcal{E}_M$  in **Lemma 11**. Thus  $x_i(k+1) - z_i^*(k+1|k) \in \mathcal{E}_i(\delta_i^*(k_0))$ , and  $\{z_i^*(j|k)\}_{j=k+1}^{k+N(k)}$  and  $\{v_i^*(j|k)\}_{j=k+1}^{k+N(k)-1}$  are feasible nominal state and input sequences, which are the tails of sequences determined at previous time step  $k$ . Thus,  $\mathbf{P}_i(x_i(k+1))$  is feasible and by induction,  $\mathbf{P}_i(x_i(j))$ ,  $\forall j \in [k+1, k+N(k)-1]$  is recursively feasible.  $\square$

**Lemma 15.** *If  $\mathbf{P}_i(x_i(k-1)) \forall i \in \mathcal{N}$  had feasible solutions at the previous time step  $k-1$ , then  $\mathbf{P}_0(\mathbf{x}(k))$  has a feasible solution at current time step  $k = \nu_0 k_0$ .*

*Proof.* Let the candidate solution to  $\mathbf{P}_0(\mathbf{x}(k))$  be the optimal nominal state and input sequences  $\{\hat{\mathbf{x}}^*(j|k_0-1)\}_{j=k_0}^{k_0+N_0(k_0)}$ ,  $\{\hat{\mathbf{u}}^*(j|k_0-1)\}_{j=k_0}^{k_0+N_0(k_0)-1}$ , corresponding to the tails of the

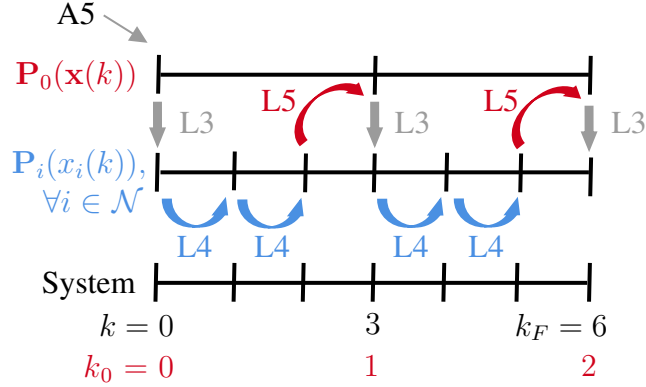


Figure 6.2. Schematic showing how **Assumption 15** and **Lemmas 13-15** are used to establish feasibility of two-level hierarchical controller with coupling between subsystems in the lower-level.

optimal solution determined at previous time step  $k_0 - 1$ , and the previously planned deviation bound  $\delta^*(k_0 - 1)$ . Since (6.11b), (6.11c), (6.11d), and (6.11f) are time-invariant, the candidate solution satisfies these constraints. To show that  $\hat{\mathbf{x}}^*(k_0|k_0 - 1)$  is a feasible initial condition, consider the following. Since,  $\mathbf{P}_i(x_i(k - 1))$  is feasible at time step  $k - 1$ , the terminal state  $z_i^*(k|k - 1) = z_i^*(k_0)$  satisfies  $z_i^*(k_0) - \hat{x}_i^*(k_0|k_0 - 1) \in \Delta \mathcal{Z}_i(\delta_i^*(k_0 - 1))$  for every  $\mathbf{S}_i$ . Additionally, using the invariance of  $\mathcal{E}_i(\delta_i^*(k_0 - 1))$  under control law (6.15),  $x_i(k) - z_i^*(k_0) \in \mathcal{E}_i(\delta_i^*(k_0 - 1))$ . Thus, by combining these statements for all subsystems,  $\hat{\mathbf{x}}^*(k_0|k_0 - 1)$  satisfies (6.11e).  $\square$

**Theorem 17.** *Following **Algorithm 7** for a two-level hierarchical controller with  $M$  controllers in the lower-level, all control problems,  $\mathbf{P}_0(\mathbf{x}(k))$  and  $\mathbf{P}_i(x_i(k))$ ,  $\forall i \in \mathcal{N}$ , are feasible, resulting in system state and input trajectories satisfying state, input, and output constraints from (6.6) and terminal constraint from (6.7).*

*Proof.* Using **Assumption 15** and **Lemmas 13-15**, Fig. 6.2 shows how feasibility is established for  $\mathbf{C}_0$  and  $\mathbf{C}_i$ ,  $\forall i \in \mathcal{N}$ . For notational convenience, let

$$\Omega \triangleq \Delta \mathcal{Y}(\delta^*(k_0)) \oplus \left( \mathcal{E}_0(\delta^*(k_0)) \times K \mathcal{E}_0(\delta^*(k_0)) \right).$$

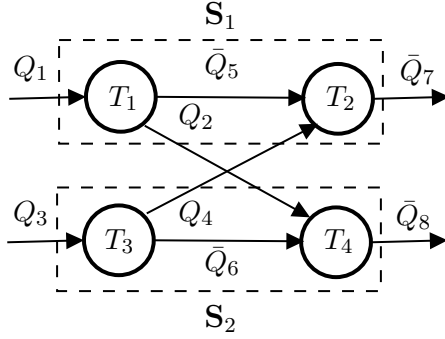


Figure 6.3. Thermal system with two subsystems  $S_1$  and  $S_2$  that are dynamically coupled by active power flows  $Q_2$  and  $Q_4$ .

Since,  $\mathbf{P}_0(\mathbf{x}(k))$  is feasible, the output trajectory  $\hat{\mathbf{y}}^*(j)$  satisfies

$$\hat{\mathbf{y}}^*(j) \in \hat{\mathcal{Y}}_0^*(k_0) \triangleq \tilde{\mathcal{Y}}_0 \ominus \Omega \subseteq \mathcal{Y} \ominus \Omega,$$

based on (6.11c), (6.26), and the fact that  $\tilde{\mathcal{Y}}_0 \subseteq \mathcal{Y}$ . Similarly, the feasibility of  $\mathbf{P}_i(x_i(k))$  guarantees that the output trajectory  $y_i^*(k|k)$  generated by  $\mathbf{C}_i$  satisfies

$$y_i^*(k|k) - \hat{y}_i^*(k) \in \Delta\mathcal{Y}_i(\delta_i^*(k_0)), \quad (6.29a)$$

$$y_i(k) - y_i^*(k|k) \in \mathcal{E}_i^*(\delta_i^*(k_0)) \times K_i\mathcal{E}_i(\delta_i^*(k_0)), \quad (6.29b)$$

based on (6.14d) and (6.14f). Thus, adding (6.29a) and (6.29b) for each system results in  $\mathbf{y}(k) \in \hat{\mathbf{y}}^*(k) \oplus \Omega$ . Since  $\hat{\mathbf{y}}^*(k) \in \mathcal{Y} \ominus \Omega$ ,  $\mathbf{y}(k) \in (\mathcal{Y} \ominus \Omega) \oplus \Omega$ . Finally, using the anti-extensive property of the set opening operation,  $\mathbf{y}(k) \in \mathcal{Y}$ . Note that satisfaction of the terminal constraint from (6.7) can be proven similarly.  $\square$

## 6.7 Numerical Example

Consider the four component thermal system shown in Fig. 6.3 where  $T_i, \forall i \in [1, 4]$ , are the temperatures of thermal components, each with thermal capacitance  $C_i$ . The power flows

(heat) into the system  $Q_1$  and  $Q_3$  are controlled directly. Two active power flows  $Q_2$  and  $Q_4$  are controlled by coolant mass flow rates  $\dot{m}_2$  and  $\dot{m}_4$  and satisfy

$$Q_2 = \dot{m}_2 c_p (T_1 - T_4), \quad Q_4 = \dot{m}_4 c_p (T_3 - T_2), \quad (6.30)$$

where  $c_p$  is the specific heat of the coolant. Additionally, four passive power flows  $\bar{Q}_5$ ,  $\bar{Q}_6$ ,  $\bar{Q}_7$ , and  $\bar{Q}_8$  have constant coolant mass flow rates  $\dot{m}_{p1}$  and  $\dot{m}_{p2}$  and satisfy

$$\bar{Q}_5 = \dot{m}_{p1} c_p (T_1 - T_2), \quad \bar{Q}_6 = \dot{m}_{p1} c_p (T_3 - T_4), \quad (6.31a)$$

$$\bar{Q}_7 = \dot{m}_{p2} c_p (T_2 - T_\infty), \quad \bar{Q}_8 = \dot{m}_{p2} c_p (T_4 - T_\infty). \quad (6.31b)$$

From conservation of energy, the nonlinear, continuous-time dynamics are

$$C_1 \dot{T}_1 = Q_1 - Q_2 - \bar{Q}_5, \quad C_2 \dot{T}_2 = \bar{Q}_5 + Q_4 - \bar{Q}_7, \quad (6.32a)$$

$$C_3 \dot{T}_3 = Q_3 - Q_4 - \bar{Q}_6, \quad C_4 \dot{T}_4 = \bar{Q}_6 + Q_2 - \bar{Q}_8. \quad (6.32b)$$

For the following results,  $C_1 = C_2 = C_3 = C_4 = 15 \times 10^4$  J/K,  $c_p = 4181$  J/(kgK), and  $T_\infty = 300$  K.

To represent (6.32a)-(6.32b) as a linear discrete-time invariant system in the form of (6.5a), these dynamics are first linearized about nominal mass flow rates  $\dot{m}_2^o = \dot{m}_4^o = 0.036$  kg/s and  $\dot{m}_{p1}^o = \dot{m}_{p2}^o = 0.108$  kg/s, nominal power flow rates  $Q_1^o = Q_3^o = 30$  kW, and nominal temperature differences between adjacent components  $\Delta T^o = 50$  K and then discretized with a time step size of  $\Delta t = 1$  s. As shown in Fig. 6.3, the system is partitioned into  $M = 2$  subsystems  $\mathbf{S}_1$  and  $\mathbf{S}_2$  with state-input pairs  $[(x_1, x_2), (u_1, u_2)]$  and  $[(x_3, x_4), (u_3, u_4)]$ , respectively. The resulting discrete-time subsystem state and input matrices from (6.1a) and coupling matrices from (6.2) are

$$A_{11} = A_{22} = \begin{bmatrix} 0.996 & 0.003 \\ 0.003 & 0.993 \end{bmatrix}, \quad B_{11} = B_{22} = \begin{bmatrix} 7e^{-6} & -1.39 \\ 1e^{-8} & -0.002 \end{bmatrix},$$



$$A_{12} = A_{21} = \begin{bmatrix} 0 & 0.001 \\ 0.001 & 0 \end{bmatrix}, \quad B_{12} = B_{21} = \begin{bmatrix} 6e^{-12} & 0.002 \\ 3e^{-9} & 1.39 \end{bmatrix}.$$

The static feedback control gains  $K_1$  and  $K_2$  are designed as discrete-time linear-quadratic regulators with weighting matrices  $Q_i = I_{n_i}$  and  $R_i = 10^5 I_{m_i}$ . The poles of each of the resulting closed-loop subsystems  $\mathbf{S}_1$  and  $\mathbf{S}_2$  are  $\{0.27, 0.99\}$ , while the closed-loop poles of the overall system are  $\{0.12, 0.42, 0.99, 0.99\}$ , and thus, **Assumption 11** is satisfied.

For a two-level hierarchical controller with two subsystems, the system and lower-level controllers  $\mathbf{C}_1$  and  $\mathbf{C}_2$  have time step sizes of  $\Delta t = \Delta t_1 = \Delta t_2 = 1$  s while the upper-level controller  $\mathbf{C}_0$  has time step size of  $\Delta t_0 = 10$  s, which results in  $\nu_0 = 10$ . Thus, the maximum prediction horizons are  $\bar{N}_0 = 200$  and  $\bar{N} = 10$ . The output constraint set  $\mathcal{Y}$  is defined such that  $\|\mathbf{x}(k)\|_\infty \leq 50$  and  $\mathbf{u}(k)$  satisfies

$$\begin{bmatrix} -Q_1^o & -\dot{m}_2^o & -Q_3^o & -\dot{m}_4^o \end{bmatrix}^T \leq \mathbf{u}(k) \leq \begin{bmatrix} 200 & 0.14 & 200 & 0.14 \end{bmatrix}^T. \quad (6.34)$$

The terminal constraint set  $\mathcal{T}$  enforces  $\|x(k_F)\|_\infty \leq 50$ . Using the procedure from Appendix 6.9.1, inter-sample constraint satisfaction for trajectories planned by  $\mathbf{C}_0$  is achieved using the tightened output constraint set  $\tilde{\mathcal{Y}}_0 = \tilde{\mathcal{X}}_0 \times \mathcal{U}$  where only the state constraints need to be tightened. Minimal tightening is required where  $\tilde{\mathcal{X}}_0 = \{\mathbf{x} \in \mathbb{R}^n \mid \underline{x} \leq \mathbf{x} \leq \bar{x}\}$  is computed such that

$$\underline{x} = \begin{bmatrix} -49.93 & -49.86 & -49.93 & -49.86 \end{bmatrix}^\top, \\ \bar{x} = \begin{bmatrix} 49.88 & 49.91 & 49.88 & 49.91 \end{bmatrix}^\top.$$

Given an initial state of  $\mathbf{x}(0) = \mathbf{0}$ , the desired operation defined by  $\{r(k)\}_{k=0}^{k_F}$ , is shown in Fig. 6.4 for the first input (power flow  $u_1$ ) and third input (power flow  $u_3$ ). References for the second and fourth inputs (mass flow rates  $u_2$  and  $u_4$ ) are the corresponding lower bounds from (6.34) for the entire operation. The primary objective is to track the desired power flows

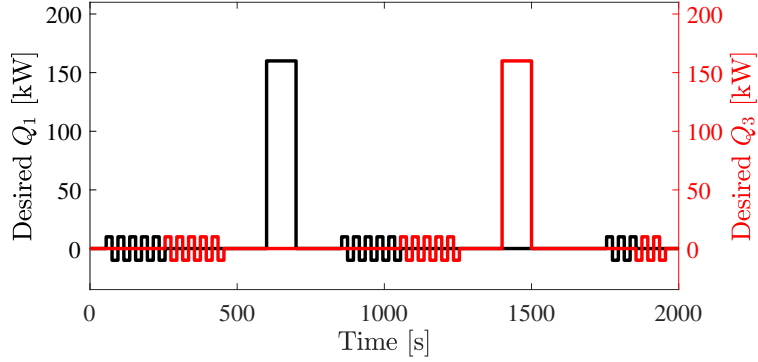


Figure 6.4. The desired reference trajectories for  $Q_1$  and  $Q_3$ .

$(u_1, u_3)$  into the system while minimizing mass flow rates  $(u_2, u_4)$  and satisfying state and input constraints from (6.6a) and (6.6b). For this example, the references for  $\mathbf{C}_0$  are obtained by downsampling the references using averaging. Note that the small high-frequency pulse references for power flows  $u_1$  and  $u_3$  vary in-between the updates of  $\mathbf{C}_0$ .

The weighted quadratic cost function in (6.8) is defined based on these references and rate of input change as

$$\ell(\mathbf{x}(j), \mathbf{u}(j), \mathbf{r}(j)) = \|\mathbf{r}(j) - \mathbf{u}(j)\|_{\Gamma_1}^2 + \|\mathbf{u}(j) - \mathbf{u}(j-1)\|_{\Gamma_2}^2, \quad (6.36)$$

Fig. 6.5 shows closed-loop simulation results using the linearized system model and the proposed two-level hierarchical controller (Hier ( $\Lambda = 10$ )), where  $\Lambda$  is a cost function weighting term used to incentivize maximizing the permissible deviation scaling vector  $\boldsymbol{\delta}(k_0)$ , compared to a two-level hierarchical controller with no subsystem deviations (Hier ( $\Lambda = 0$ )), a shrinking horizon centralized controller (Cent) that predicts to the end of system operation, a receding horizon centralized controller with a control invariant terminal set and a prediction horizon of  $N = 100$  time steps (Cent T), and a receding horizon centralized controller (Cent Short) with a short prediction horizon of  $N = 10$  time steps. As expected, all the controllers except the Cent Short controller satisfy the state constraints. Since there does not exist a steady-state operating condition that satisfies the state and input constraints while tracking the desired large pulsed power flows  $Q_1$  and  $Q_3$  shown in the third row of subplots in

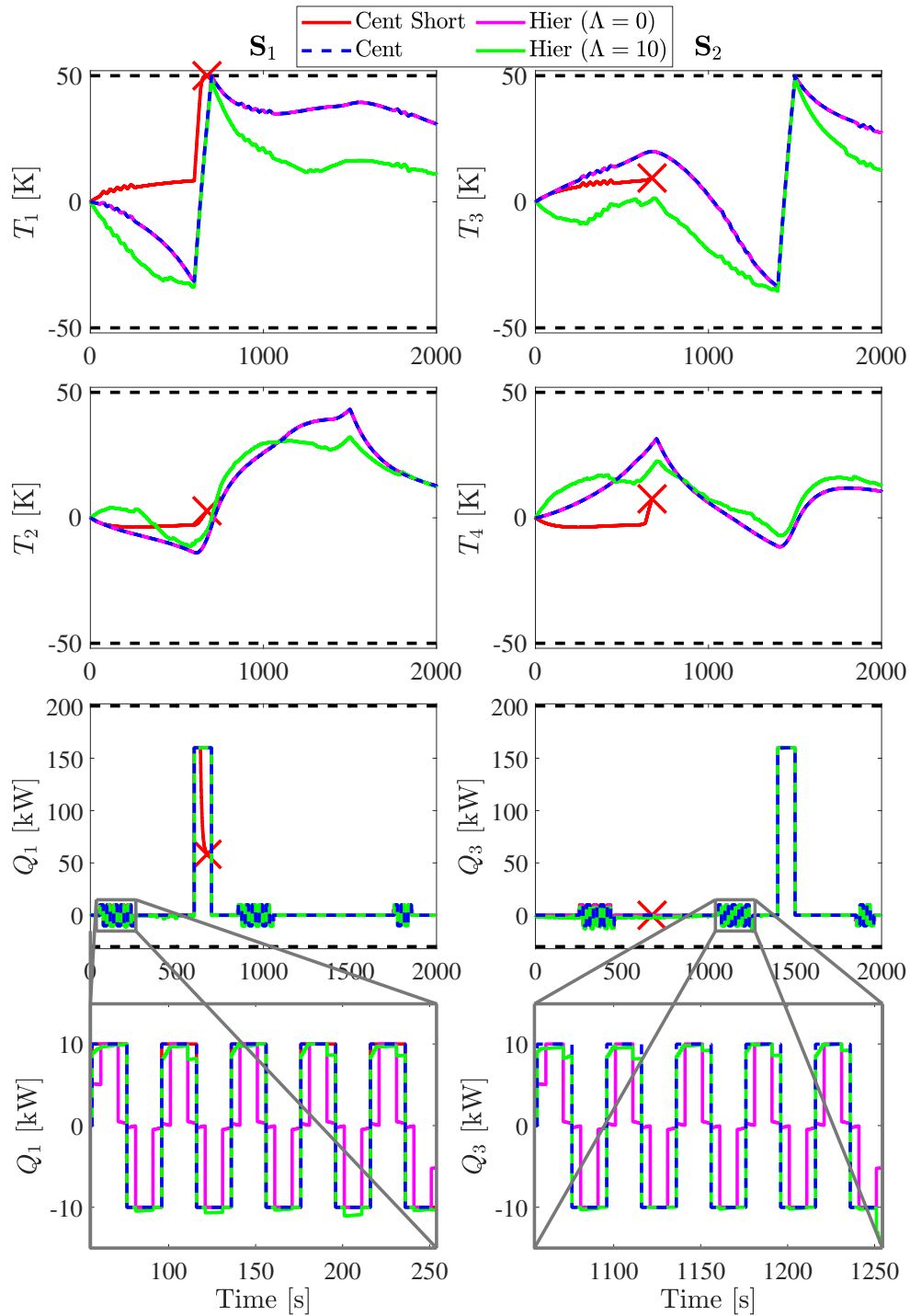


Figure 6.5. Simulation results comparing the shrinking horizon centralized controller, receding horizon centralized controller with a short prediction horizon, receding horizon centralized controller with a terminal constraint, two-level hierarchical controller with no subsystem deviations, and a two-level hierarchical controller with subsystem deviations.

Table 6.1. Controller computation times

Controller	Computation time (s)		
	Minimum	Mean	Maximum
Cent	13.569	23.144	1092.3
Cent Short	0.009	0.01	0.05
Cent T	0.08	0.09	0.137
Hier ( $\mathbf{C}_0$ )	0.757	2.262	4.105
Hier ( $\mathbf{C}_1, \mathbf{C}_2$ )	0.011	0.014	0.08

Fig. 6.5, the controllers strategically *precool* the system temperatures to utilize the thermal capacitance of the system. Note that due to a short prediction horizon, the Cent T controller is unable to achieve the necessary level of precooling and thus, significantly reduces the power flow into the system to avoid temperature constraint violations. While the Cent controller tracks all pulses of the reference power flows, Hier ( $\Lambda = 0$ ) tracks only the large pulse references. The smaller, high-frequency, pulse references are not tracked perfectly by Hier ( $\Lambda = 0$ ) due to step changes occurring between updates of  $\mathbf{C}_0$ . Alternatively, Hier ( $\Lambda = 10$ ) achieves perfect tracking of the large pulsed power flows and nearly perfect tracking of the small, high-frequency, pulse references by allowing the lower-level controllers to deviate from the upper-level prediction, as shown in bottom row of subplots in Fig. 6.5 during the time intervals [55, 254] s and [1055, 1254] s, respectively.

Fig. 6.6 shows the total operating costs computed using (6.36) when using the proposed hierarchical controller for  $\Lambda = \{10, 10^5, 5 \times 10^5, 10^6\}$  normalized by the cost when using Hier ( $\Lambda = 0$ ). Initially, the normalized costs decreases due to the additional flexibility provided to  $\mathbf{C}_1$  and  $\mathbf{C}_2$  by the increasing size of output deviation sets. However, as shown by Hier ( $\Lambda = 5 \times 10^5$ ), there is a point where further increasing the size of the output deviation sets leads to significant constraint tightening that degrades overall control performance.

Using YALMIP [57] and Gurobi optimizer 8.5 [34] to formulate and solve the controller optimization problems, Table 6.1 shows the mean, minimum, and maximum computation

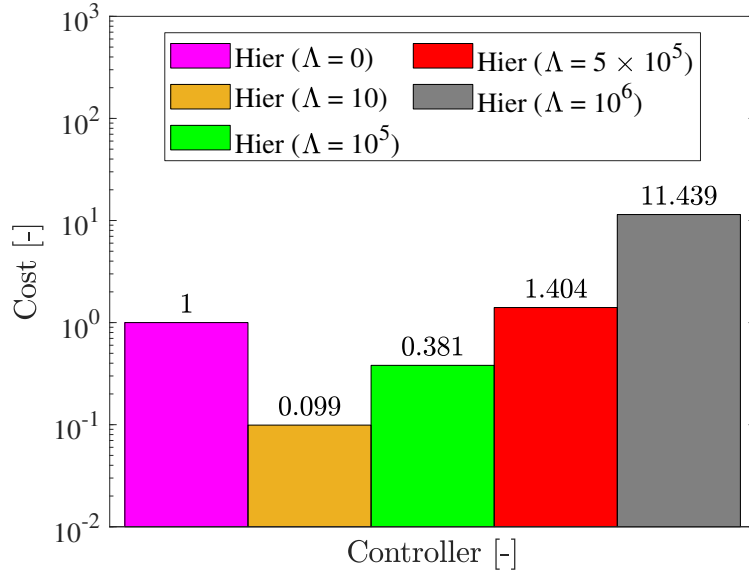


Figure 6.6. Relation between the chosen value of  $\Lambda$  and normalized cost relative to Hier ( $\Lambda = 0$ ) computed using (6.36).

times for each controller. Note that despite the additional complexity of simultaneous uncertainty set optimization and constraint tightening at  $\mathbf{C}_0$ , the computation time is smaller than that of the Cent controller. Furthermore, the average computation time of  $\mathbf{C}_1$  and  $\mathbf{C}_2$  is similar to Cent Short and less than Cent T. Overall, the proposed hierarchical approach is expected to remain computationally efficient for systems with significantly more states, inputs, and subsystems.

## 6.8 Conclusions

A two-level hierarchical MPC formulation was presented for linear systems of dynamically-coupled subsystems. Adjustable tubes are used to bound permissible deviations between the system trajectories planned by the upper- and lower-level controllers. A tube-based robust MPC formulation with simultaneous uncertainty set optimization and constraint tightening guaranteed constraint satisfaction to bounded disturbances between subsystem controllers. A numerical example demonstrated the performance of the proposed two-level hierarchical

MPC. Future work will focus on the extension of the proposed set-based hierarchical MPC formulation to nonlinear systems and include more than two levels of controllers with application to systems of greater complexity. The following chapter will discuss about the development of a wayset-based stochastic MPC framework with guaranteed Mission-Wide Probability of Safety (MWPS) for systems with long duration.

## 6.9 Appendix

This appendix provides the details necessary to implement the proposed hierarchical MPC controller.

### 6.9.1 Inter-sample tightened output constraint set computation for $\mathbf{C}_0$

The tightened output constraint set  $\tilde{\mathcal{Y}}_0 \subseteq \mathcal{Y}_0$  introduced in (6.26) is used to ensure that the coarse trajectories planned by  $\mathbf{C}_0$  produce upsampled trajectories in (6.16) and (6.17a) that satisfy the original output constraints. While there are many ways to achieve this, the approach used in this research is based on bounding the difference between the upsampled trajectories and the linear interpolations of the coarse trajectories planned by  $\mathbf{C}_0$ . Specifically, let  $\hat{\mathbf{x}}(k_0)$  and  $\hat{\mathbf{x}}(k_0 + 1)$  denote the first two states predicted by  $\mathbf{C}_0$  corresponding to the first input  $\hat{\mathbf{u}}(k_0)$ , where  $\hat{\mathbf{x}}(k_0 + 1) = A^{\nu_0} \hat{\mathbf{x}}(k_0) + \sum_{j=0}^{\nu_0-1} A^j B \hat{\mathbf{u}}(k_0)$ . The linearly interpolated trajectories  $\mathbf{x}_l(k + i)$ ,  $\forall i \in [1, \nu_0 - 1]$ , can be computed between  $\hat{\mathbf{x}}(k_0)$  and  $\hat{\mathbf{x}}(k_0 + 1)$  as

$$\mathbf{x}_l(k + i) = \hat{\mathbf{x}}(k_0) + \frac{i}{\nu_0} (\hat{\mathbf{x}}(k_0 + 1) - \hat{\mathbf{x}}(k_0)).$$

By the convexity of  $\mathcal{X}$ ,  $\hat{\mathbf{x}}(k_0), \hat{\mathbf{x}}(k_0 + 1) \in \mathcal{X}$  implies  $\mathbf{x}_l(k + i) \in \mathcal{X}$ . However, it is not guaranteed that the upsampled trajectory satisfies  $\hat{\mathbf{x}}(k + i) \in \mathcal{X}$ ,  $\forall i \in [1, \nu_0 - 1]$ . Defined as  $\mathbf{e}(k + i) = \hat{\mathbf{x}}(k + i) - \mathbf{x}_l(k + i)$ , the difference between these trajectories can be computed as

$$\mathbf{e}(k + i) = A_e(i) \hat{\mathbf{x}}(k) + B_e(i) \hat{\mathbf{u}}(k), \quad (6.37)$$

where  $A_e(i) = A^i - \frac{i}{\nu_0} A^{\nu_0} - (1 - \frac{i}{\nu_0}) I_n$  and  $B_e(i) = \sum_{j=0}^{i-1} A^j B - \frac{i}{\nu_0} \sum_{j=0}^{\nu_0-1} A^j B$ . Since  $\hat{\mathbf{x}}(k) \in \mathcal{X}$  and  $\hat{\mathbf{u}}(k) \in \mathcal{U}$ , these differences are bounded such that  $\mathbf{e}(k+i) \in \mathcal{F}_i = A_e(i)\mathcal{X} \oplus B_e(i)\mathcal{U}$ . Thus, defining  $\mathcal{F} = \mathcal{F}_1 \cup \dots \cup \mathcal{F}_{\nu_0-1}$  ensures  $\mathbf{e}(k+i) \in \mathcal{F}, \forall i \in [1, \nu_0-1]$ . Finally, computing  $\tilde{\mathcal{Y}} = \tilde{\mathcal{X}} \times \mathcal{U}$  where  $\tilde{\mathcal{X}} = \mathcal{X} \ominus \mathcal{F}$  guarantees that the upsampled trajectories satisfy the original constraints. For ease of implementation, outer-approximating bounding boxes of  $\mathcal{F}_i$  are computed and thus,  $\mathcal{F}$  is also a bounding box.

### 6.9.2 RPI set computation

This section presents how the RPI set  $\mathcal{E}_0(\boldsymbol{\delta}(k_0))$ , used in (6.11e), is computed through the addition of linear constraints and cost function terms in the formulation of  $\mathbf{P}_0(\mathbf{x}(k))$  using the approach from [65]. Before proceeding with the computation of  $\mathcal{E}_0(\boldsymbol{\delta}(k_0))$ , consider the following definition.

**Definition 10.** [65] *The zonotope  $\mathcal{Z}(\Phi) = \{G\Phi, \mathbf{c}\} \subset \mathbb{R}^n$  is a scaled version of the nominal zonotope  $\mathcal{Z} = \{G, \mathbf{c}\}$  with the generator matrix  $G$  scaled by a diagonal matrix  $\Phi \in \mathbb{R}^{n_g \times n_g}$ ,  $\Phi = \text{diag}(\phi_i), \phi_i \geq 0, \forall i \in [1, n_g]$ .*

Let the RPI set be a scaled zonotope such that  $\mathcal{E}_0(\boldsymbol{\delta}) = \{G_\varepsilon \Phi_\varepsilon, \mathbf{0}\}$  with an *a priori* chosen nominal generator matrix  $G_\varepsilon \in \mathbb{R}^{n \times n_\varepsilon}$ . The permissible state and input deviation sets are defined as scaled zonotopes with centers at the origin such that  $\Delta\mathcal{Z} = \{G_z \text{diag}(\boldsymbol{\delta}^z), \mathbf{0}\}$  and  $\Delta\mathcal{V} = \{G_v \text{diag}(\boldsymbol{\delta}^v), \mathbf{0}\}$ , where  $G_z \in \mathbb{R}^{n \times n_z}$  and  $G_v \in \mathbb{R}^{m \times n_v}$ . From (6.21), the resulting disturbance error set is a scaled zonotope such that  $\Delta\mathcal{W} = \{G_w \text{diag}(\boldsymbol{\delta}), \mathbf{0}\}$ , where  $G_w = [A_C G_z \ B_C G_v] \in \mathbb{R}^{n \times n_w}$  and  $n_w = n_z + n_v$ . In the numerical results from Section 6.7,  $G_z = I$  and  $G_v = I$  such that  $n_z = n$  and  $n_v = m$ .

Following the approach from [65], based on the one-step RPI computation from [66] and the zonotope containment conditions from [71], the decision variables  $\Phi_\varepsilon \in \mathbb{R}^{n_\varepsilon \times n_\varepsilon}$ ,  $\Gamma_{\varepsilon,1} \in \mathbb{R}^{n_\varepsilon \times n_\varepsilon}$ , and  $\Gamma_{\varepsilon,2} \in \mathbb{R}^{n_\varepsilon \times n_w}$  are added to  $\mathbf{P}_0(\mathbf{x}(k))$  with linear constraints

$$(A + BK)G_\varepsilon \Phi_\varepsilon = G_\varepsilon \Gamma_{\varepsilon,1}, \quad (6.38a)$$

$$G_w \text{diag}(\boldsymbol{\delta}) = G_\varepsilon \Gamma_{\varepsilon,2}, \quad (6.38b)$$

$$|\Gamma_{\varepsilon,1}| \mathbf{1} + |\Gamma_{\varepsilon,2}| \mathbf{1} \leq \Phi_\varepsilon \mathbf{1}. \quad (6.38c)$$

The cost function for  $\mathbf{P}_0(\mathbf{x}(k))$  is modified to balance system performance with the maximization of  $\boldsymbol{\delta}$  through the addition of the term  $\Lambda \|\bar{\boldsymbol{\delta}} - \boldsymbol{\delta}\|_p$ , where  $\Lambda$  is a scalar weighting term and  $\bar{\boldsymbol{\delta}}$  is a user-specified upper-bound on  $\boldsymbol{\delta}$ . In the numerical results from Section 6.7 showing the effects of  $\Lambda$ ,  $\bar{\boldsymbol{\delta}}$  is chosen to equal the upper bounds on the states and inputs and the 1-norm is used ( $p = 1$ ).

For the linear constraints (6.38) to admit a feasible solution, the generator matrix  $G_\varepsilon$  needs to be chosen carefully. As in [66, 65], an intuitive choice of generators is based on  $G_w$  and  $\bar{A}_K = \text{diag}(A_{ii} + B_{ii}K_i)$  such that

$$G_\varepsilon = [G_w \bar{A}_K G_w \cdots \bar{A}_K^{n_s} G_w], \quad (6.39)$$

where  $n_s \in \mathbb{Z}_+$  is a parameter that can be increased to promote the feasibility of (6.38) at the cost of set complexity and the number of decision variables. In the numerical results from Section 6.7,  $G_\varepsilon = [G_w \bar{A}_K G_w]$  is chosen with  $n_s = 1$ .

Note that the block-diagonal structure of  $\bar{A}_K$  and structure of  $G_w$  ensures that  $G_\varepsilon$  from (6.39) is separable and thus,  $\mathcal{E}_0$  is a structured RPI set satisfying (6.13). Once  $\mathcal{E}_0(\boldsymbol{\delta}^*(k_0)) = \{G_\varepsilon \Phi_\varepsilon, c_\varepsilon\}$  is computed, the subsystem-level RPI sets  $\mathcal{E}_i = \{G_\varepsilon^i, c_\varepsilon^i\}$ ,  $i \in \mathcal{N}$ , satisfying (6.13) can be obtained by projection.

### 6.9.3 Output Constraint Tightening for $\mathbf{C}_0$

This section presents how the output constraint tightening used to compute  $\hat{\mathcal{Y}}_0(\boldsymbol{\delta}(k_0))$ , based on (6.26) and used in (6.11c), is integrated into the formulation of  $\mathbf{P}_0(\mathbf{x}(k))$  through the addition of linear constraints and cost function terms.

In (6.26), let the inter-sample tightened output constraint set be a zonotope such that  $\tilde{\mathcal{Y}}_0 = \{\tilde{G}_y, \tilde{c}_y\}$  with known generator matrix  $\tilde{G}_y \in \mathbb{R}^{(n+m) \times n_{\tilde{y}}}$  and center  $\tilde{c}_y \in \mathbb{R}^{n+m}$ . Let



the tightened output constraint set be a scaled zonotope such that  $\hat{\mathcal{Y}}_0(\boldsymbol{\delta}(k_0)) = \{\hat{G}_y \Phi_y, \hat{c}_y\}$ , where  $\hat{G}_y \in \mathbb{R}^{(n+m) \times n_{\hat{y}}}$  is an *a priori* chosen nominal generator matrix and  $\Phi_y$  is a diagonal scaling matrix satisfying  $\Phi_y = \text{diag}(\phi_y)$  with  $\phi_{y,i} \geq 0, \forall i \in [1, n_{\hat{y}}]$ .

Following the approach from [65], based on the one-step Pontryagin difference computation from [66] and the zonotope containment conditions from [71], the decision variables  $\hat{c}_y \in \mathbb{R}^{n+m}$ ,  $\Phi_y \in \mathbb{R}^{n_{\hat{y}} \times n_{\hat{y}}}$ ,  $\Gamma_y \in \mathbb{R}^{n_{\hat{y}} \times (n_{\hat{y}} + n_w + 2n_\varepsilon)}$ , and  $\beta_y \in \mathbb{R}^{n_{\hat{y}}}$  are added to  $\mathbf{P}_0(\mathbf{x}(k))$  with linear constraints

$$\begin{bmatrix} \hat{G}_y \Phi_y & \begin{bmatrix} [G_z \text{diag}(\boldsymbol{\delta}^z) G_\varepsilon \Phi_\varepsilon] & \mathbf{0} \\ \mathbf{0} & [G_v \text{diag}(\boldsymbol{\delta}^v) K G_\varepsilon \Phi_\varepsilon] \end{bmatrix} \end{bmatrix} = \tilde{G}_y \Gamma_y, \quad (6.40a)$$

$$\tilde{c}_y - \hat{c}_y = \tilde{G}_y \beta_y, \quad (6.40b)$$

$$|\Gamma_y| \mathbf{1} + |\beta_y| \leq \mathbf{1}. \quad (6.40c)$$

The cost function for  $\mathbf{P}_0(\mathbf{x}(k))$  is modified to balance system performance with maximizing the size of  $\hat{\mathcal{Y}}_0(\boldsymbol{\delta}(k_0))$  through the addition of the term  $-||\phi_y||_p$ , where  $\phi_y$  is the vector of scaling variables along the diagonal of  $\Phi_y$ . In the numerical results from Section 6.7, the 1-norm is used ( $p = 1$ ).

Note that the choice of  $\hat{G}_y$  affects the quality of the inner-approximation of the Pontryagin difference from (6.26). While the addition of generators could improve the approximation, for the numerical results from Section 6.7,  $\hat{G}_y$  was simply chosen as  $\hat{G}_y = \tilde{G}_y$ .

#### 6.9.4 Terminal Constraint Tightening for $\mathbf{C}_0$

The terminal constraint tightening used to compute  $\hat{\mathcal{T}}_0(\boldsymbol{\delta}^*(k_0))$ , based on (6.27) and used in (6.11d), is integrated into the formulation of  $\mathbf{P}_0(\mathbf{x}(k))$  through the addition of linear constraints and cost function terms following the same approach used in the previous section for output constraint tightening, and thus is not repeated here for brevity.

### 6.9.5 Set containment condition ( $\Delta\mathcal{Z} \subseteq \text{Pre}(\Delta\mathcal{Z})$ )

This section presents how the set containment  $\Delta\mathcal{Z} \subseteq \text{Pre}(\Delta\mathcal{Z})$  in (6.11f) can be enforced using linear constraints. Assuming the invertibility of  $A_D$ , the precursor set is a zonotope defined as  $\text{Pre}(\Delta\mathcal{Z}) = \{G_p \text{diag}(\boldsymbol{\delta}), \mathbf{0}\}$ , with  $G_p = [A_D^{-1}G_z - A_D^{-1}B_D G_v]$ .

Using the zonotope containment conditions from [71], the decision variable  $\Gamma_p \in \mathbb{R}^{n_w \times n_z}$  is added to  $\mathbf{P}_0(\mathbf{x}(k))$  with linear constraints

$$G_z \text{diag}(\boldsymbol{\delta}^z) = G_p \Gamma_p, \quad (6.41a)$$

$$|\Gamma_p| \mathbf{1} \leq \text{diag}(\boldsymbol{\delta}) \mathbf{1}. \quad (6.41b)$$

## CHAPTER 7

# LONG DURATION STOCHASTIC MPC WITH MISSION-WIDE PROBABILISTIC CONSTRAINTS USING WAYSETS<sup>1</sup>

### 7.1 Introduction

Model Predictive Control (MPC) is well-suited for applications where the goal is to plan and execute closed-loop state and input trajectories that satisfy state and input constraints while driving the terminal state to a terminal constraint set. In applications such as spacecraft rendezvous, aircraft missions, entering the terminal constraint set in a finite time is required and several shrinking horizon MPC formulations have been developed [36, 22]. In the presence of disturbances bounded to a known set, robust MPC formulations based on constraint tightening have been proposed that guarantee recursive feasibility of the MPC optimization problem and constraint satisfaction of the closed-loop trajectories [60, 65].

If the disturbances are unbounded or if these bounds are relatively large, guaranteeing recursive feasibility and closed-loop constraint satisfaction can be challenging in the sense of being overly conservative at best and impossible at worst. Therefore, a Stochastic MPC (SMPC) approach can be adapted to guarantee a desired probability of constraint satisfaction. The majority of SMPC formulations focus on individual chance constraints on the probability of constraint violations at any particular point in time [23, 76]. Even if a joint probability constraint is imposed across multiple time steps, this joint probability is typically decomposed into individual chance constraints using Booles's inequality [32, 58]. Referring to these approaches as constraining the Stage-Wise Probability of Safety (SWPS), the authors in [89] show that a SWPS approach becomes very conservative for systems operating over a long duration.

---

<sup>1</sup>This chapter is based on work supported by the National Science Foundation under grant 1849500.

Alternatively, this chapter aims to constrain the Mission-Wide Probability of Safety (MWPS), first introduced in [89]. Specifically, the operation of the system from an initial condition to a terminal set in a specific amount of time, subject to state and input constraints, is referred to as a mission. A mission is successful if all state and input constraints are satisfied and the terminal state lies in the terminal set. Constraining the MWPS to exceed some probability  $p$  means that when running a large number of missions, the percentage of successful missions will converge to at least  $p$ .

The novel approach to achieving a desired MWPS proposed in [89] leverages a Markov chain perspective to show that SMPC formulations with open-loop MWPS constraints are recursively feasible and result in the closed-loop system achieving a designed MWPS. A scenario-based approach is used to convert the SMPC problem into a deterministic robust MPC formulation that is easily solved. The authors of [89] show that a MWPS is fundamentally less conservative than a SWPS approach.

While the efficacy of this approach is demonstrated in [89] for a discrete-time linear system operating for eleven time steps, a number of limitations exist. First, only state constraints are considered in [89] and the terminal set is assumed to equal the state constraint set. To increase the applicability of this approach, input constraints need to be considered along with terminal sets that are significantly smaller than the state constraint set. The latter poses a challenge for the approach in [89] where constraint tightening based on the forward propagation of uncertainty due to the disturbances could lead to an empty tightened terminal set.

To overcome these limitations, a wayset-based approach to SMPC with MWPS constraints is proposed for systems with state, input, and terminal state constraints. While the fundamental approaches from [89] are adopted, the proposed use of waysets allows for the prediction horizon to be significantly shorter than the mission length and significantly reduces the degree of constraint tightening required for long missions. Thus, the main contributions of this chapter are: (1) the formulation of waysets that preserve the desired MWPS

constraint, (2) the proof of recursive feasibility of the proposed wayset-based MPC formulation, and (3) the demonstration that the use of wayset increases the feasibility of SMPC with MWPS constraint to longer missions and/or larger disturbances.

The remainder of the section is organized as follows. Section II introduces the problem formulation, while Section III provides the background on Stochastic MPC with MWPS. Section IV discusses the wayset-based stochastic MPC with MWPS and Section V proves recursive feasibility of the corresponding scenario-based optimization problem. Section VI highlights the applicability to a numerical example and finally, Section VII provides conclusive remarks.

## 7.2 Notation and Preliminaries

Probabilities and conditional probabilities are denoted as  $\mathbb{P}[a]$  and  $\mathbb{P}[a|b]$ . The expected value of  $x$  over random variable  $w$  is denoted  $\mathbb{E}_w[x]$ . The set of integers in the interval  $[j_1, j_2] \in \mathbb{R}$  is denoted  $\mathbb{I}_{[j_1, j_2]}$ . Vectors are denoted in lowercase (e.g.  $x \in \mathbb{R}^n$ ), sets are denoted in calligraphic uppercase (e.g.  $\mathcal{X} \subset \mathbb{R}^n$ ), and matrices are denoted in uppercase (e.g.  $A \in \mathbb{R}^{(n \times n)}$ ). Additionally, uppercase is used to denote trajectories of vectors where, for example,  $U_{[j_1, j_2]} = \{u(j)\}_{j=j_1}^{j_2}$  denotes the trajectory of the vector  $u(j)$  over time steps  $j \in \mathbb{I}_{[j_1, j_2]}$ . The notation  $U_{[j_1, j_2]} \in \mathcal{U}$  denotes that each vector  $u(j)$  in the trajectory  $U_{[j_1, j_2]}$  is constrained as  $u(j) \in \mathcal{U}$ . For a discrete-time system,  $x(k)$  denotes the state  $x$  at time step  $k$ . For MPC, the double-index notation  $x(j|k)$  denotes the predicted state  $x$  at future time step  $j$  determined at  $k$ . Given the sets  $\mathcal{Z}, \mathcal{W} \subset \mathbb{R}^n$ ,  $\mathcal{Y} \subset \mathbb{R}^m$ , and matrix  $R \in \mathbb{R}^{m \times n}$  the linear mapping of  $\mathcal{Z}$  by  $R$  is  $R\mathcal{Z} = \{Rz \mid z \in \mathcal{Z}\}$ , the Minkowski sum of  $\mathcal{Z}$  and  $\mathcal{W}$  is  $\mathcal{Z} \oplus \mathcal{W} = \{z + w \mid z \in \mathcal{Z} \wedge w \in \mathcal{W}\}$ , the generalized intersection of  $\mathcal{Z}$  and  $\mathcal{Y}$  under  $R$  is  $\mathcal{Z} \cap_R \mathcal{Y} = \{z \in \mathcal{Z} \mid Rz \in \mathcal{Y}\}$ , and the Pontryagin difference of  $\mathcal{W}$  from  $\mathcal{Z}$  is  $\mathcal{Z} \ominus \mathcal{W} = \{z \mid z \oplus \mathcal{W} \subseteq \mathcal{Z}\}$ .

### 7.3 Problem Formulation

Consider the discrete linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k) + w(k), \quad (7.1)$$

with states  $x \in \mathbb{R}^n$ , inputs  $u \in \mathbb{R}^m$ , disturbances  $w \in \mathbb{R}^n$ , and the stabilizable pair  $(A, B)$ ,  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ .

**Assumption 16.** *With a fixed time step size  $\Delta t$ , the system operates over a mission with a finite length of time starting from  $t = 0$  and ending at  $t = t_F = k_F \Delta t$ , with discrete time steps indexed by  $k \in \mathbb{I}_{[0, k_F]}$ .*

**Assumption 17.** *The disturbance trajectory  $W_{[0, k_F-1]} = \{w(j)\}_{j=0}^{k_F-1}$  is a random variable with known probability distribution such that  $N_s$  independent random disturbance trajectories  $W_{[0, k_F-1]}^{(i)} = \{w^{(i)}(j)\}_{j=0}^{k_F-1}$ ,  $i \in \mathbb{I}_{[1, N_s]}$  can be generated from this probability distribution.*

Starting from an initial condition  $x(0)$ , the goal is to formulate a SMPC such that the resulting closed-loop input trajectory  $U_{[0, k_F-1]} = \{u(j)\}_{j=0}^{k_F-1}$ , and corresponding state trajectory  $X_{[1, k_F]} = \{x(k)\}_{k=1}^{k_F}$  achieve a desired closed-loop MWPS  $p$  such that

$$\mathbb{P}[X_{[1, k_F-1]} \in \mathcal{X} \wedge U_{[0, k_F-1]} \in \mathcal{U} \wedge x(k_F) \in \mathcal{T} | x(0)] \geq p, \quad (7.2)$$

while minimizing the generic cost function

$$\min_{U_{[0, k_F-1]}} \mathbb{E}_w \left[ \sum_{j=0}^{k_F-1} \ell(x(j), u(j)) + \ell_F(x(k_F)) \right], \quad (7.3)$$

where  $\ell$  and  $\ell_F$  are given stage and terminal costs.

**Assumption 18.** *The state, input, and terminal constraint sets  $\mathcal{X}$ ,  $\mathcal{U}$ , and  $\mathcal{T}$  are compact and convex.*

## 7.4 SMPC with MWPS Background

### 7.4.1 Open-loop and Closed-loop MWPS

The approach provided in [89] establishes the theoretical foundation relating the MWPS of the closed-loop system to the MWPS constraints imposed in the open-loop planning of the SMPC controller. Specifically, let the MWPS of the open-loop trajectories planned at time step  $k$  be

$$\mathbb{P}[X_{[k,k_F]} \in \mathcal{X} | x(k), U_{[k,k_F-1]}^k], \quad (7.4)$$

where  $U_{[k,k_F-1]}^k$  denotes the input trajectory from time steps  $k$  to  $k_F - 1$  planned at time step  $k$ . Note that the input constraints are omitted and the terminal constraint is set as  $\mathcal{T} = \mathcal{X}$  for this subsection for brevity and to be consistent with the formulation in [89]. The main results of [89] (Proposition 1 and Corollary 1) show that constraining the initial open-loop MWPS at  $k = 0$  such that

$$\mathbb{P}[X_{[0,k_F]} \in \mathcal{X} | x(0), U_{[0,k_F-1]}^0] \geq p_0, \quad (7.5)$$

where

$$\prod_{k=1}^{k_F-1} \gamma_k p_0 = p, \quad (7.6)$$

with  $\gamma_k \in (0, 1] \forall k$ , achieves the desired closed-loop MWPS

$$\mathbb{P}[X_{[1,k_F]} \in \mathcal{X} | x(0)] \geq p, \quad (7.7)$$

as long as the open-loop MWPS is constrained to exceed the discounted MWPS from the previous time step

$$\mathbb{P}[X_{[k+1,k_F]} \in \mathcal{X} | x(k), U_{[k,k_F-1]}^k] \geq \gamma_k \mathbb{P}[X_{[k+1,k_F]} \in \mathcal{X} | x(k), U_{[k,k_F-1]}^{k-1}]. \quad (7.8)$$

This result is based on viewing the state trajectory as a Markov chain and leads to the conclusion that the probability of successful missions under SMPC with these MWPS constraints will asymptotically converge to at least the desired probability  $p$ . In [89], only state

constraints are considered while state, input, and terminal state constraints are included in the MWPS in this work. Since these additional constraints do not affect the results summarized above, the approach from [89] is adopted in this work. With a primary focus on long missions, the constraint from (7.6) requires  $\gamma_k$  to be close to one for all  $k$ , and therefore this discounting factor is not considered in this work ( $\gamma_k = 1$ ). This allows for a simplified implementation where  $p_0 = p$ . In summary, based on the results from [89], the goal of this work is to solve the following SMPC problem at each time  $k$  with an open-loop MWPS constraint

$$\min_{U_{[k, k_F-1]}^k} \mathbb{E}_w \left[ \sum_{j=k}^{k_F-1} \ell(x(j), u(j)) + \ell_F(x(k_F)) \right], \quad (7.9a)$$

$$\text{s.t. } \forall j \in \mathbb{I}_{[k, k_F-1]},$$

$$x(j+1|k) = Ax(j|k) + Bu(j|k) + w(j), \quad (7.9b)$$

$$x(k|k) = x(k), \quad (7.9c)$$

$$\mathbb{P}[X_{[k+1, k_F-1]}^k \in \mathcal{X} \wedge U_{[k, k_F-1]}^k \in \mathcal{U} \wedge x(k_F|k) \in \mathcal{T}|x(k), U_{[k, k_F-1]}^k] \geq p, \quad (7.9d)$$

to achieve a desired closed-loop MWPS of  $p$ . Note that this SMPC formulation has a shrinking horizon of length  $\bar{N}(k) = k_F - k$ .

#### 7.4.2 Scenario-based SMPC

A scenario-based approach to SMPC allows probabilistic constraints, such as (7.9d), to be replaced with a sufficiently large finite number of deterministic constraints based on sampled disturbance trajectories. Specifically, let a generic SMPC optimization problem be defined as

$$\min_{u \in \mathbb{R}^d} \ell(u), \quad (7.10a)$$

$$\text{s.t. } \mathbb{P}[u \in S_w] \geq p, \quad (7.10b)$$



where  $u \in \mathbb{R}^d$  is a vector of  $d$  decision variables (with notation chosen to reflect that the inputs are the decision variables in MPC problems),  $\ell$  is a generic cost function, and  $S_w$  is a closed and convex set based on the realization of the random variable  $w$ . By sampling  $N_s$  values of the random variable  $w$ , denoted as  $w^{(i)}$ ,  $i \in \mathbb{I}_{[1, N_s]}$ , the corresponding deterministic optimization problem is

$$\min_{u \in \mathbb{R}^d} \ell(u), \quad (7.11a)$$

$$\text{s.t. } u \in \bigcap_{i=1}^{N_s} S_{w^{(i)}}. \quad (7.11b)$$

If  $N_s$  is chosen to be sufficiently large, then the solution of (7.11) is a feasible solution of (7.10) with a probability level  $1 - \beta$ , where  $\beta$  is typically chosen to be very small, *e.g.*,  $\beta = 10^{-6}$ . While there are several explicit bounds on  $N_s$  with varying levels of conservatism [17, 18], this work uses the bound

$$N_s \geq \frac{2}{1-p} \left( \ln \frac{1}{\beta} + d \right). \quad (7.12)$$

When using a scenario-based approach to solve the SMPC with MWPS constraint in [89], the  $N_s$  sampled disturbance realizations are used to tighten the state constraints. A polytopic Halfspace-representation (H-rep) of these state constraints makes the constraint tightening computationally efficient, *i.e.* the MPC optimization problem complexity does not depend on the number of samples  $N_s$ .

### 7.4.3 Drawbacks

While [89] only considers state constraints, this work intends to include state, input, and terminal state constraints. The extension to include input constraints is relatively minor but the adaptation of the approach from [89] to include terminal state constraints presents a significant challenge. Specifically, the terminal constraint set  $\mathcal{T}$  is often significantly smaller than the state constraint set  $\mathcal{X}$ . As such, when including terminal constraints in the generic

constraint sets  $S_{w^{(i)}}$  from (7.11b), the intersection of these sets may be empty. Conceptually, this results from the case when there is not a single control sequence that can drive the terminal state to the terminal constraint set for all of the  $N_s$  sampled disturbance trajectories. To limit the effects of the unknown disturbances on the state trajectory, it is common in robust MPC formulations to augment the nominal inputs determined by the MPC with a static feedback control law, with stabilizing feedback matrix  $K$ , which is also done in [89]. However, this may not prevent the intersection in (7.11b) from being empty, particularly for long missions ( $k_F$  is large). Therefore, the following section proposes a wayset-based approach to overcome this issue.

## 7.5 Wayset-based SMPC with MWPS

The underlying idea of using waysets to overcome the infeasibility issue associated with long missions and small terminal constraint sets is to use a shorter prediction horizon  $N(k) < \bar{N}(k)$  and a wayset, denoted  $\mathcal{S}_{p,\mathcal{T},w}(k+N(k))$ , as a terminal constraint instead of  $x(k_F|k) \in \mathcal{T}$ . The notation for the wayset highlights the fact that the set  $\mathcal{S}$  depends on the probability level  $p$ , the terminal set  $\mathcal{T}$ , the disturbance trajectory  $\{w(j)\}_{j=k+N(k)}^{k_F-1}$ , and is imposed as a terminal constraint at time step  $k + N(k)$ . When these dependencies do not need to be explicitly stated, the short-hand  $\mathcal{S}(k + N(k))$  is used. The primary goal for the remainder of the chapter is to construct  $\mathcal{S}(k + N(k))$  and prove that by solving

$$\min_{U_{[k,k+N(k)-1]}^k} \mathbb{E}_w \left[ \sum_{j=k}^{k+N(k)-1} \ell(x(j), u(j)) + \ell_F(x(k_F)) \right], \quad (7.13a)$$

$$\text{s.t. } \forall j \in \mathbb{I}_{[k,k+N(k)-1]},$$

$$x(j+1|k) = Ax(j|k) + Bu(j|k) + w(j), \quad (7.13b)$$

$$x(k|k) = x(k), \quad (7.13c)$$

$$\begin{aligned} \mathbb{P}[X_{[k+1, k+N(k)-1]}^k \in \mathcal{X} \wedge U_{[k, k+N(k)-1]}^k \in \mathcal{U} \wedge \\ x(k+N(k)|k) \in \mathcal{S}(k+N(k)|x(k), U_{[k, k+N(k)-1]}^k)] \geq p, \end{aligned} \quad (7.13d)$$

there exists an input trajectory  $U_{[k+N(k), k_F-1]}^k$  such that the resulting input trajectory  $U_{[k, k_F-1]}^k$  satisfies (7.9d).

### 7.5.1 Error Dynamics

Given that a scenario-based approach will be used to approximate the solution of (7.13), a scenario-based approach is used to bound the effects of the unknown disturbance trajectory. Given the  $N_s$  randomly generated disturbance trajectories  $W_{[0, k_F-1]}^{(i)}$ ,  $i = \mathbb{I}_{[1, N_s]}$ , let each set  $\mathcal{W}(k)$ ,  $k = \mathbb{I}_{[0, k_F-1]}$ , be the convex hull of the  $N_s$  disturbances at time step  $k$  such that

$$\mathcal{W}(k) = \text{CH}(w^{(i)}(k)), \quad i = \mathbb{I}_{[1, N_s]}. \quad (7.14)$$

Let the true state at time step  $j$  based on the MPC solution at time step  $k$  be  $x(j|k) = \hat{x}(j|k) + e(j|k)$ , where  $\hat{x}$  is the nominal state planned by the controller and  $e$  is the error induced by the disturbances. To control this error, the planned input to the system  $u(j|k)$  is a combination of the nominal part  $\hat{u}(j|k)$  and stochastic error part  $Ke(j|k)$  such that

$$u(j|k) = \hat{u}(j|k) + Ke(j|k), \quad (7.15)$$

where  $K$  is a static feedback controller for the nominal system dynamics  $\hat{x}(j+1|k) = A\hat{x}(j|k) + B\hat{u}(j|k)$  such that  $(A + BK)$  is stable. Based on (7.15), the predicted error evolves  $\forall j \in \mathbb{I}_{[k, k_F]}$  as

$$e(j+1|k) = (A + BK)e(j|k) + w(j), \quad (7.16)$$

with  $e(k|k) = 0$ . Since  $w(j) \in \mathcal{W}(j)$ ,  $e(j|k) \in \mathcal{E}(j|k)$ , where  $\mathcal{E}(k|k) = 0$  and

$$\mathcal{E}(j+1|k) = (A + BK)\mathcal{E}(j|k) + \mathcal{W}(j), \quad \forall j \in \mathbb{I}_{[k, k_F]}. \quad (7.17)$$

Following the standard tube-based robust MPC formulation [60], constraining the nominal state and input trajectories to tightened constraint sets can ensure that the true state and input trajectories for each of the  $N_s$  disturbance trajectory realizations satisfy the original constraints. Specifically, the scenario-based approximation of (7.9) is

$$\min_{\hat{U}_{[k, k_F-1]}^k} \sum_{j=k}^{k_F-1} \ell(\hat{x}(j), \hat{u}(j)) + \ell_F(\hat{x}(k_F)), \quad (7.18a)$$

$$\text{s.t. } \forall j \in \mathbb{I}_{[k, k_F-1]},$$

$$\hat{x}(j+1|k) = A\hat{x}(j|k) + B\hat{u}(j|k), \quad (7.18b)$$

$$\hat{x}(k|k) = x(k), \quad (7.18c)$$

$$\hat{x}(j|k) \in \mathcal{X} \ominus \mathcal{E}(j|k), \quad (7.18d)$$

$$\hat{u}(j|k) \in \mathcal{U} \ominus K\mathcal{E}(j|k), \quad (7.18e)$$

$$\hat{x}(k_F|k) \in \mathcal{T} \ominus \mathcal{E}(k_F|k). \quad (7.18f)$$

To ensure that the minimization of (18a) minimizes the expected cost in (13a), it is assumed that the disturbances  $w(k)$  have zero-mean for all  $k \in \mathbb{I}_{[0, k_F-1]}$ .

## 7.5.2 Wayset Computation

As discussed in Section 7.4.3, this chapter proposes the use of waysets to overcome the fact that  $\mathcal{T} \ominus \mathcal{E}(k_F|k)$  may be an empty set if  $\mathcal{T}$  is relatively small and/or if  $\mathcal{E}(k_F|k)$  is large due to a large  $k_F$  or large disturbances. Fundamentally, the goal is to define  $\mathcal{S}(k+N(k))$  such that  $x(k+N(k)|k) \in \mathcal{S}(k+N(k))$  guarantees the existence of an input trajectory  $U_{[k+N(k), k_F-1]}^k$  where the resulting state and input trajectories and terminal state satisfy their constraints for all  $N_s$  disturbance trajectory realizations. By definition, this corresponds to  $\mathcal{S}(k+N(k))$  being a  $(k_F - k - N(k))$ -step robust controllable set [14], and is computed by **Algorithm 8** through iterative calculations of robust precursor sets starting from the terminal set  $\mathcal{T}$ .

---

**Algorithm 8:** Calc. waysets  $\mathcal{S}(k), \forall k \in \mathbb{I}_{[1, k_F-1]}$ .

---

```

1 Initialize  $k \leftarrow k_F, \mathcal{S}(k) = \mathcal{T}$ ;
2 while  $k \geq 1$  do
3    $\mathcal{S}(k-1) = \mathcal{X} \cap_A [(\mathcal{S}(k) \ominus W(k-1)) \oplus (-B\mathcal{U})]$ ;
4    $k \leftarrow k-1$ ;
5 end

```

---

### 7.5.3 Scenario-based SMPC with Wayset

The resulting sub-optimal approximation of (7.18) using waysets is

$$\min_{\hat{U}_{[k, k+N(k)-1]}^k} \sum_{j=k}^{k+N(k)-1} \ell(\hat{x}(j), \hat{u}(j)) + \ell_F(\hat{x}(k+N(k))), \quad (7.19a)$$

$$\text{s.t. } \forall j \in \mathbb{I}_{[k, k+N(k)-1]},$$

$$\hat{x}(j+1|k) = A\hat{x}(j|k) + B\hat{u}(j|k), \quad (7.19b)$$

$$\hat{x}(k|k) = x(k), \quad (7.19c)$$

$$\hat{x}(j|k) \in \mathcal{X} \ominus \mathcal{E}(j|k), \quad (7.19d)$$

$$\hat{u}(j|k) \in \mathcal{U} \ominus K\mathcal{E}(j|k), \quad (7.19e)$$

$$\hat{x}(k+N(k)|k) \in \mathcal{S}(k+N(k)) \ominus \mathcal{E}(k+N(k)|k). \quad (7.19f)$$

The prediction horizon  $N(k)$  can be varied to ensure the feasibility of (7.19) for all time steps  $k \in \mathbb{I}_{[0, k_F-1]}$ . Specifically, it is assumed that  $N(0)$  can be chosen to maximize control performance while satisfying the constraint  $\mathcal{S}(N(0)) \ominus \mathcal{E}(N(0)|0) \neq \emptyset$ . If  $k+N(k) = k_F$ , then a shrinking prediction horizon is used such that  $N(k+1) = N(k) - 1$ . If  $k+N(k) \neq k_F$ , then a receding or shrinking horizon can be used. As discussed in detail when formally proving recursive feasibility in Section 7.6, a sufficient condition is used to guarantee the feasibility of (7.19) at time step  $k+1$  based on the feasibility at time step  $k$  using a receding horizon,  $N(k+1) = N(k)$ . Specifically, a receding horizon is feasible if there exists a candidate nominal input  $\bar{u}(k+N(k)|k+1)$  that satisfies the tightened input constraints  $\mathcal{U} \ominus K\mathcal{E}(k+N(k)|k+1)$

and drives the candidate state to the tightened wayset  $\mathcal{S}(k+N(k)+1) \ominus \mathcal{E}(k+N(k)+1|k+1)$ .

This is equivalent to the set

$$\begin{aligned} \bar{\mathcal{U}}(k+N(k)) &= [\mathcal{U} \ominus K\mathcal{E}(k+N(k)|k+1)] \bigcap_B \\ &[\mathcal{S}(k+N(k)+1) \ominus \mathcal{E}(k+N(k)+1|k+1) - A\tilde{x}(k+N(k)|k+1)], \end{aligned} \quad (7.20)$$

being non-empty, where  $\tilde{x}(k+N(k)|k+1) = \hat{x}^*(k+N(k)|k) + (A+BK)^{(N(k)-1)}w(k)$  is a candidate terminal state for time step  $k+1$  and  $\hat{x}^*(k+N(k)|k)$  is the optimal nominal terminal state determined at time step  $k$ . Note that  $w(k) = x(k+1) - \hat{x}^*(k+1|k)$  is known when computing  $\bar{\mathcal{U}}(k+N(k))$  prior to solving (7.19) at time step  $k+1$ . If  $\bar{\mathcal{U}}(k+N(k)) = \emptyset$ , then a shrinking horizon is used where  $N(k+1) = \max(N(k)-1, 1)$ . The prediction horizon update is summarized as

$$N(k+1) = \begin{cases} N(k), & \text{if } k+N(k) < k_F \wedge \bar{\mathcal{U}}(k+N(k)) \neq \emptyset, \\ \max(N(k)-1, 1), & \text{if } k+N(k) = k_F \vee \bar{\mathcal{U}}(k+N(k)) = \emptyset. \end{cases} \quad (7.21)$$

#### 7.5.4 Implementation and Set Computations

**Algorithm 9** shows the implementation of the proposed scenario-based SMPC with MWPS. The majority of set computations can be performed offline. In **Line 1**, the Multi-Parametric Toolbox [37] is used to compute minimal representations of  $\mathcal{W}(k)$  in Vertex-representation (V-rep) and identify the corresponding minimal H-rep. To compute the error sets in (7.17) efficiently, the  $\mathcal{W}(k)$  are first converted from H-rep to constrained zonotopes in CG-rep based on the method proposed in [79]. Denoted as  $\text{Box}(\mathcal{W}(k))$ , interval hull outer-approximations of  $\mathcal{W}(k)$  are also computed by solving  $2n$  linear programs to make the Pontryagin difference operations more computationally efficient. With  $\mathcal{W}(k)$  in CG-rep, **Line 2** computes the error sets  $\mathcal{E}(j|k)$  in CG-rep using linear transformations and Minkowski sums based on (7.17). Once computed, these error sets are also outer-approximated with the interval hull as  $\text{Box}(\mathcal{E}(j|k))$ . In **Line 3**, inner-approximations of the waysets are computed using **Algorithm**

8, where  $\text{Box}(\mathcal{W}(k))$  is used instead of  $\mathcal{W}(k)$  to make the iterative method from [66] for computing the Pontryagin difference more computationally efficient. When computing **Line 3** of **Algorithm 8**, a zonotopic inner-approximation of  $\mathcal{S}(k-1)$  is computed using the method from [66] to increase computational efficiency. In **Lines 3** and **4** of **Algorithm 9**,  $\text{Box}(\mathcal{E}(j|k))$  is used instead of  $\mathcal{E}(j|k)$ . The tightened wayset constraint (7.19f) is implemented using a hypercube inner-approximation, computed using the method from [66], to simplify the MPC formulation and implementation.

In the online portion of **Algorithm 9**, similar approximations are used to compute  $\bar{\mathcal{U}}(k+N(k))$  from (7.20) to determine the length of the prediction horizon  $N(k)$  based on (7.21). Using zonotopes and constrained zonotopes and the strategic dependence on inner- and outer-approximations, the online constraint tightening for (7.19d)-(7.19f) in **Line 11** can be computed extremely quickly.

---

**Algorithm 9:** Wayset-based SMPC with MWPS

---

**Offline:**

- 1 Generate  $N_s$  dist. traj.  $W_{[0, k_F-1]}^{(i)}$ ,  $i = \mathbb{I}_{[1, N_s]}$ , and bounding sets  $\mathcal{W}(k)$ ,  $\forall k = \mathbb{I}_{[0, k_F-1]}$ ;
- 2 Compute error sets per (7.17),  $\forall k \in \mathbb{I}_{[0, k_F]}$ ;
- 3 Compute waysets  $\mathcal{S}(k)$ ,  $\forall k \in \mathbb{I}_{[1, k_F-1]}$  per **Algorithm 8**;
- 4 Choose  $N(0)$  s.t.  $\mathcal{S}(N(0)) \ominus \mathcal{E}(N(0)|0) \neq \emptyset$ ;
- 5 Compute tightened constraints (7.19d)-(7.19f) ;
- 6 Initialize  $k \leftarrow 0$ ;

**Online:**

- 7 **while**  $k \leq k_F - 1$  **do**
- 8     **if**  $k \neq 0$  **then**
- 9         | Compute  $N(k)$  using (7.21);
- 10     **end**
- 11     Compute tightened constraints (7.19d)-(7.19f);
- 12     Solve optimization problem (7.19);
- 13     Apply  $u(k) = \hat{u}^*(k|k)$  to the system;
- 14      $k \leftarrow k + 1$ ;
- 15 **end**

---

## 7.6 Recursive Feasibility

**Assumption 19.** *The optimization problem (7.19) is feasible at time step  $k = 0$ .*

**Lemma 16.** *If (7.19) is feasible at time step  $k \in \mathbb{I}_{[0, k_F-2]}$  and  $w(k) \in \mathcal{W}(k)$ , then (7.19) is feasible at time step  $k + 1$  when using the prediction horizon update law from (7.21).*

*Proof.* Let  $\{\hat{x}^*(j|k)\}_{j=k}^{k+N(k)}$  and  $\{\hat{u}^*(j|k)\}_{j=k}^{k+N(k)-1}$  be the optimal nominal state and input trajectories determined by solving (7.19) at time step  $k$  such that constraints (7.19b)-(7.19f) are satisfied. First, consider the shrinking horizon case where  $N(k+1) = N(k) - 1 \geq 1$  based on the conditions from (7.21). The candidate solution at time step  $k+1$  is initialized as  $\tilde{x}(k+1|k+1) = x(k+1) = \hat{x}^*(k+1|k) + e(k+1|k)$ , satisfying (7.19c). Note that  $e(k+1|k) = w(k)$  based on (7.16). The candidate input trajectory is defined as  $\tilde{u}(j|k+1) = \hat{u}^*(j|k) + Ke(j|k)$ ,  $\forall j \in \mathbb{I}_{[k+1, k+N(k)-1]}$ , where  $e(j|k) = \tilde{x}(j|k+1) - \hat{x}^*(j|k)$  and  $\tilde{x}(j+1|k+1) = A\tilde{x}(j|k+1) + B\tilde{u}(j|k+1)$ , satisfying (7.19b). From these definitions and system dynamics, the error evolves as  $e(j+1|k) = (A+BK)e(j|k)$ ,  $\forall j \in \mathbb{I}_{[k+1, k+N(k)-1]}$ , initialized with  $e(k+1|k) = w(k)$ . Therefore,  $e(j|k) = (A+BK)^{(j-k-1)}w(k)$ . Note that  $\mathcal{E}(j|k) = \mathcal{E}(j|k+1) \oplus (A+BK)^{(j-k-1)}\mathcal{W}(k)$ ,  $\forall j \in \mathbb{I}_{[k+1, k+N(k)]}$ , based on (7.17). From (7.19d) at time step  $k$ ,  $\hat{x}^*(j|k) \in \mathcal{X} \ominus \mathcal{E}(j|k)$ ,  $\forall j \in \mathbb{I}_{[k, k+N(k)-1]}$ . Therefore,

$$\begin{aligned}
\tilde{x}(j|k+1) &= \hat{x}^*(j|k) + e(j|k), \\
&\in \mathcal{X} \ominus \mathcal{E}(j|k) + e(j|k), \\
&\in \mathcal{X} \ominus \mathcal{E}(j|k) + (A+BK)^{(j-k-1)}w(k), \\
&\in \mathcal{X} \ominus (\mathcal{E}(j|k+1) \oplus (A+BK)^{(j-k-1)}\mathcal{W}(k)) \\
&\quad + (A+BK)^{(j-k-1)}w(k), \\
&\in \mathcal{X} \ominus \mathcal{E}(j|k+1),
\end{aligned}$$



which shows that the candidate solution satisfies (7.19d). Very similar steps can be used to prove that

$$\begin{aligned}\tilde{u}(j|k+1) &\in \mathcal{U} \ominus K\mathcal{E}(j|k+1), \\ \tilde{x}(k+N(k)|k+1) &\in \mathcal{S}(k+N(k)) \ominus \mathcal{E}(k+N(k)|k+1),\end{aligned}$$

satisfying (7.19e) and (7.19f).

Next, consider the receding horizon case where  $N(k+1) = N(k)$ . The same candidate solution as the shrinking horizon case is used for time steps  $j \in \mathbb{I}_{[k+1, k+N(k)-1]}$ . The same procedure can also be used to show that this solution satisfies (7.19b)-(7.19e) for time steps  $j \in \mathbb{I}_{[k+1, k+N(k)-1]}$ . To handle the receding horizon, it must be shown that 1)  $\tilde{x}(k+N(k)|k+1) \in \mathcal{X} \ominus \mathcal{E}(k+N(k)|k+1)$  and that 2) there exists  $\tilde{u}(k+N(k)|k+1) \in \mathcal{U} \ominus K\mathcal{E}(k+N(k)|k+1)$  such that  $\tilde{x}(k+N(k)+1|k+1) \in \mathcal{S}(k+N(k)+1) \ominus \mathcal{E}(k+N(k)+1|k+1)$ . To show 1),

$$\begin{aligned}\tilde{x}(k+N(k)|k+1) &= \hat{x}^*(k+N(k)|k) + e(k+N(k)|k), \\ &\in \mathcal{S}(k+N(k)) \ominus \mathcal{E}(k+N(k)|k) + e(k+N(k)|k), \\ &\in \mathcal{X} \ominus \mathcal{E}(k+N(k)|k) + e(k+N(k)|k), \\ &\in \mathcal{X} \ominus \mathcal{E}(k+N(k)|k+1),\end{aligned}$$

using similar arguments from before and the fact that  $\mathcal{S}(k+N(k)) \subseteq \mathcal{X}$  by definition. To show 2), note that  $\tilde{x}(k+N(k)|k+1) = \hat{x}^*(k+N(k)|k) + (A+BK)^{(N(k)-1)}w(k)$ . The existence of  $\tilde{u}(k+N(k)|k+1)$  with the desired properties is guaranteed if  $\bar{\mathcal{U}}(k+N(k)) \neq \emptyset$ , based on the definition in (7.20). Note that  $\bar{\mathcal{U}}(k+N(k)) \neq \emptyset$  is a necessary condition for a receding horizon in (7.21). Thus the candidate solution satisfies all the constraints in (7.19) at time step  $k+1$ .

Finally, if  $N(k) = 1$ , a receding horizon must be used such that  $N(k+1) = 1$  even if  $\bar{\mathcal{U}}(k+N(k)) = \emptyset$ . Let  $\hat{x}^*(k|k)$ ,  $\hat{x}^*(k+1|k)$ , and  $\hat{u}^*(k|k)$  be the optimal solutions from time step  $k$ . As before, the candidate solution at time step  $k+1$  is initialized as  $\tilde{x}(k+1|k+1) =$

$x(k+1)$ , which satisfies (7.19c) and (7.19d) following the same arguments presented in the shrinking horizon case. Note that  $\tilde{x}(k+1|k+1) = \hat{x}^*(k+1|k) + w(k)$ . Since  $\hat{x}^*(k+1|k) \in \mathcal{S}(k+1) \ominus \mathcal{E}(k+1|k)$  and  $w(k) \in \mathcal{E}(k+1|k) = \mathcal{W}(k)$ ,  $\tilde{x}(k+1|k+1) \in \mathcal{S}(k+1)$ . By the wayset definition,  $\tilde{x}(k+1|k+1) \in \mathcal{S}(k+1)$  guarantees there exists  $\tilde{u}(k+1|k+1) \in \mathcal{U}$  such that  $\tilde{x}(k+2|k+1) = A\tilde{x}(k+1|k+1) + B\tilde{u}(k+1|k+1) \in \mathcal{S}(k+1) \ominus \mathcal{W}(k+1)$ . Since  $\mathcal{W}(k+1) = \mathcal{E}(k+2|k+1)$ , (7.19f) is satisfied at time step  $k+1$ . Additionally, (7.19e) is satisfied since  $\mathcal{E}(k+1|k+1) = \emptyset$  and thus  $\tilde{u}(k+1|k+1) \in \mathcal{U} \ominus K\mathcal{E}(k+1|k+1) = \mathcal{U}$ .  $\square$

While the condition  $w(k) \in \mathcal{W}(k)$  in the statement of **Lemma 16** may seem restrictive and may not be satisfied in practice due to the potential infinite support for the random variable  $w(k)$ , it is important to show that the proposed constraint tightening, wayset formulation, and prediction horizon update law lead to recursive feasibility when  $w(k) \in \mathcal{W}(k)$ . As a result, if (7.19) was feasible at time step  $k$  but is *not* feasible at time step  $k+1$ , then  $w(k) \notin \mathcal{W}(k)$ . In this case, based on the scenario-based approach to satisfying the probabilistic constraint (7.13d) at time step  $k$ , the probability of the control policy planned at time step  $k$  failing to satisfy all of the mission constraints is at most  $1-p$ .

## 7.7 Numerical Example

Consider the discrete-time system with  $n = 2$  states,  $m = 1$  input, and matrices

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}.$$

The state, input, and terminal constraint sets are

$$\mathcal{X} = \left\{ x \mid \begin{bmatrix} -10 \\ -1 \end{bmatrix} \leq x \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}, \quad \mathcal{U} = \{u \mid \|u\|_\infty \leq 2\},$$

$$\mathcal{T} = \left\{ x \mid \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix} \leq x \leq \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right\}.$$

It is assumed that the system operates for 20 steps and thus,  $k_F = 20$ . The disturbance  $d$  is assumed to be Gaussian such that  $d \in \mathcal{N}(\mu_d, \Sigma_d)$  with  $\mu_d = \mathbf{0}_n$ ,  $\Sigma_d = 0.001\mathbf{I}_n$ . For  $\beta = 10^{-6}$ ,  $o_d = k_F m = 20$ , probability levels  $p = 0.85$  and  $p = 0.95$ , the number of samples  $N_s$  from (7.12) needed to achieve desired level of MWPS are  $N_s = 451$  and  $N_s = 1353$ , respectively.

For  $p = 0.95$ , the terminal error set  $\text{Box}(\mathcal{E}(k_F|0))$ , corresponding to the  $N_s = 1353$  randomly generated disturbance trajectories, and terminal set  $\mathcal{T}$  are shown in the top subplot of Fig. 7.1. Since  $\text{Box}(\mathcal{E}(k_F|0)) \not\subseteq \mathcal{T}$ , the tightened terminal set  $\mathcal{T} \ominus \mathcal{E}(k_F|0)$  is empty and thus, the existing mission-wide shrinking horizon stochastic MPC from [89] may be infeasible. However, the proposed scenario-based SMPC with waysets in (7.19) can be employed with a short prediction horizon  $N(0) = 10$ . The tightened terminal wayset  $\mathcal{S}(N(0)) \ominus \mathcal{E}(N(0)|0)$ , imposed as the terminal constraint in (7.19f), is shown in the bottom subplot of Fig. 7.1.

Fig. 7.2 shows the state trajectories for 900 missions starting from  $x(0) = [-10 \ 0]^T$  for probability level  $p = 0.95$ . Note that 889 of the 900 missions (98.8%) are successful and thus, the desired probability of MWPS is achieved as expected. The constraint violations are due to the state trajectory crossing the upper-bound of  $x_2$  and the terminal state violating the upper-bound of  $x_1$ . The input constraints are satisfied for all missions and thus, are not shown in Fig. 7.2.

Fig. 7.3 shows the trend between MWPS and mission index for both probability levels  $p = 0.85$  and  $p = 0.95$ . While the MWPS decreases initially due to mission failures, the achieved MWPS appears to converge to steady states of about 0.98 and 0.97, respectively. The observed conservatism is likely due to the conservative bound on  $N_s$  used in (7.12) and the approximations used in the computation of disturbance sets, waysets, and tightened state, input, and terminal constraint sets.

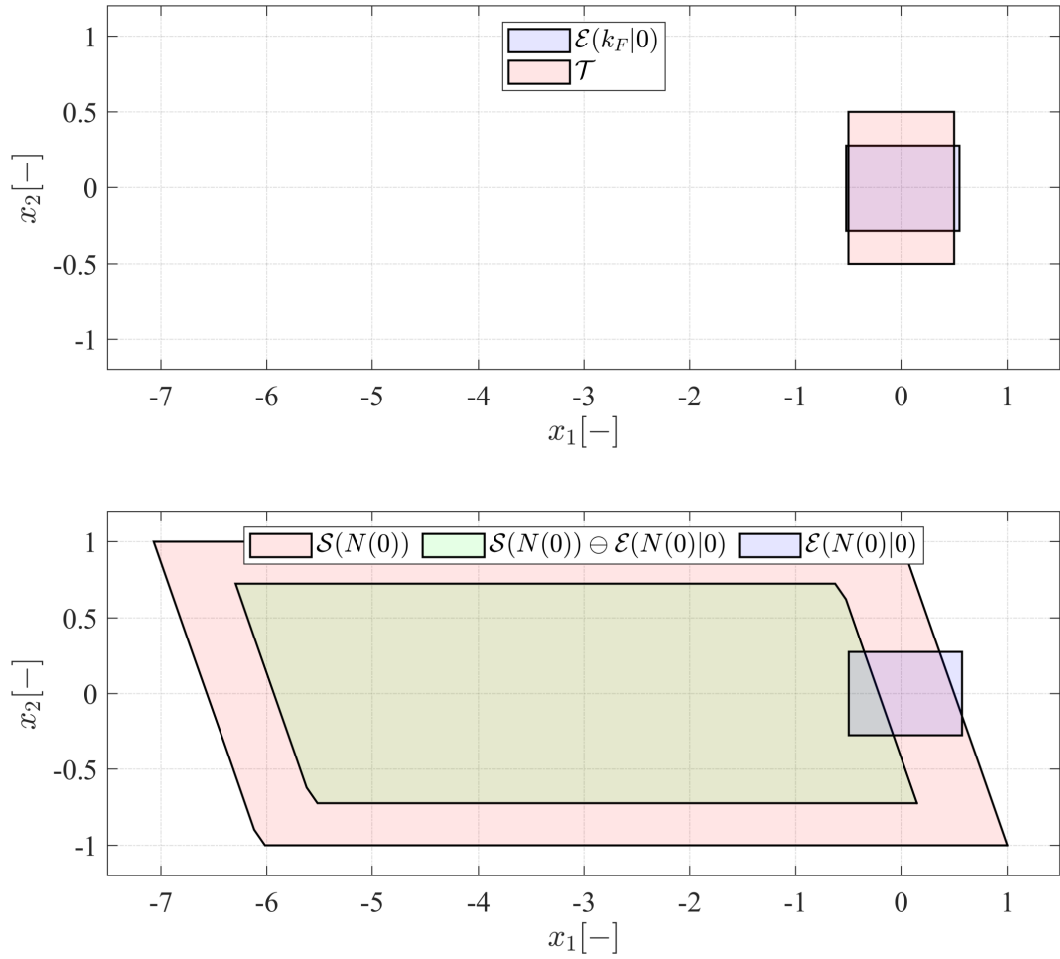


Figure 7.1. Top subplot shows the error set  $\mathcal{E}(k_F|k)$  computed using scenario optimization from [89] and terminal set  $\mathcal{T}$ . Bottom subplot shows the stochastic wayset  $\mathcal{S}(N(0))$  computed per **Algorithm 8**, error set  $\mathcal{E}(N(0)|0)$  determined per (7.17) and tightened stochastic wayset  $\mathcal{S}(N(0)) \ominus \mathcal{E}(N(0)|0)$ .

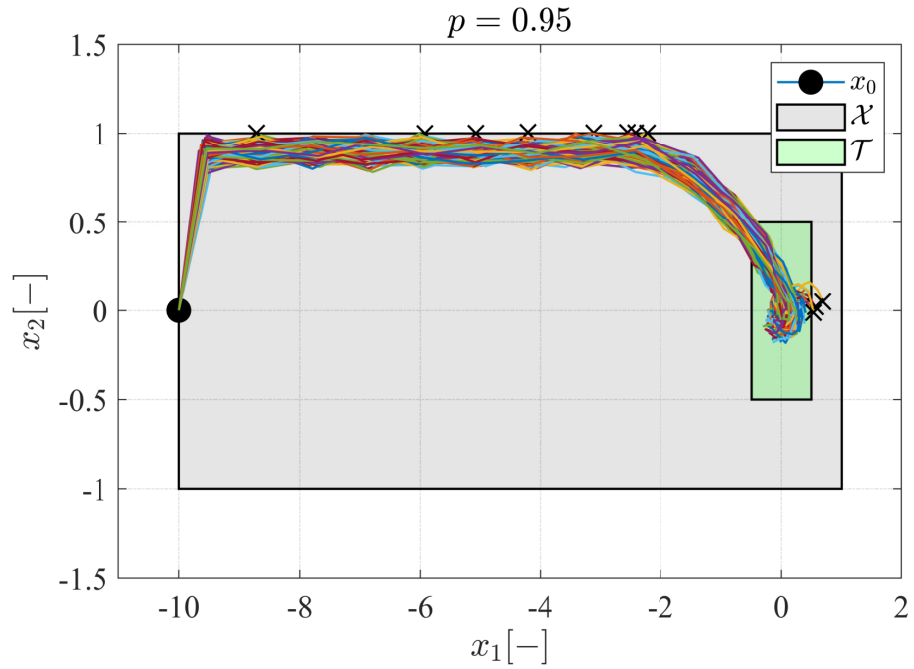


Figure 7.2. Figure showing the state trajectories for 900 missions starting from  $x(0) = [-10 \ 0]^T$  for  $p = 0.95$ . Note that crosses denote mission failures.

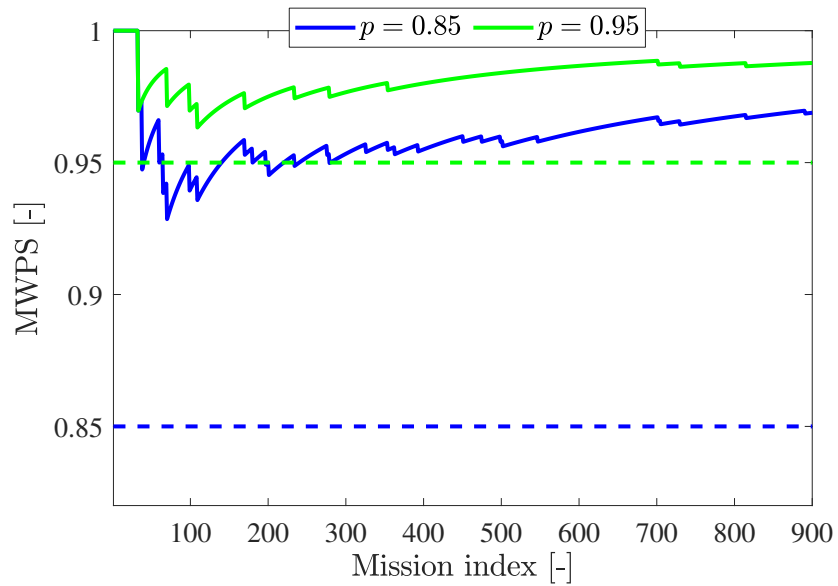


Figure 7.3. Figure showing the trend between MWPS and mission index for  $p = 0.85$  (solid blue) and  $p = 0.95$  (solid green). Note that MWPS thresholds are shown in dashed green and dashed blue, respectively.

## 7.8 Conclusions

A stochastic MPC formulation is presented for linear systems with additive stochastic disturbances under finite operation subject to constraints on the Mission-Wide Probability of Safety (MWPS). A wayset-based approach is proposed to enable missions with long duration while achieving a desired MWPS. A scenario-based approach is used to approximate the stochastic MPC optimization problem and constrained zonotopes are used to achieve efficient computation of the waysets. A numerical example demonstrates the benefits and key features of using this approach. Future work will focus on reducing the conservatism of the proposed wayset-based stochastic MPC framework and the extension to hierarchical stochastic MPC.

## CHAPTER 8

### CONCLUSIONS AND FUTURE DIRECTIONS

This dissertation presented the theoretical development, analysis, and demonstration of set-based hierarchical control frameworks for multi-timescale systems. Optimizing the *energy management* is very important to extract maximum performance of these complex systems with increasing demands for performance, efficiency, and reliability. With multiple systems and subsystems of various domains interacting over a wide range of timescales, and operating over long prediction horizons, these systems require modeling and control frameworks that are *scalable, robust, computationally efficient*, and are *widely applicable*.

This is addressed through developments in the following areas.

- Chapter 2 developed set-based methods using *zonotopes* and *constrained zonotopes* to compute sets resulting from halfspace intersections, convex hulls, robust positively invariant sets, and Pontryagin differences. Order reduction techniques are also presented that provide lower-complexity inner-approximations of zonotopes and constrained zonotopes. Numerical examples are used to demonstrate the efficacy and computational advantages of using zonotope-based set representations for dynamic system analysis and control.
- Chapter 3 utilized methods developed for *constrained zonotopes* to compute reachable sets within the context of a hierarchical control framework. Constraint satisfaction to disturbances is guaranteed using a *tube-based* robust MPC controller at every level of the hierarchy. Coordination is achieved using *waysets* and a numerical example demonstrated the key features, performance, and scalability of the proposed approach.
- Chapter 4 developed a two-level hierarchical MPC formulation with coordinating terminal cost using *constrained zonotopes*. In addition to waysets, significant improvement

in hierarchical control performance is achieved by using terminal costs in the control formulation. The proposed terminal costs are time-varying and are computed online using constrained zonotopes. A numerical example demonstrated the performance benefits of using the proposed terminal cost coordination mechanism.

- A *tube-based* robust MPC formulation with adjustable uncertainty set optimization using *zonotopes* was presented in Chapter 5. Constraint satisfaction to these uncertainties bounded to this set is achieved by using robust positive invariant sets and constraint tightening. These set computations are embedded online in the underlying control optimization problem. A *Hausdorff distance* metric is used to reduce conservatism when optimizing the size of these tightened constraint sets. A numerical example demonstrated the key features and limitations of this approach.
- Chapter 6 presented a *full* hierarchical MPC formulation for coupled discrete-time linear systems. Hierarchical control coordination is achieved using *adjustable tubes*, which are optimized online by the upper-level controller. The size of these adjustable tubes represent the degree of uncertainty between subsystems, which are computed efficiently using *zonotopes*. Recursive control feasibility is proven and a numerical example demonstrated the key features and performance of the approach.
- Chapter 7 discussed a stochastic MPC formulation for systems with finite operation subject to constraints on the *Mission-Wide Probability of Safety*. A wayset-based approach is proposed for systems with *long* missions and *scenario optimization* is used to approximate the stochastic MPC optimization problem. Recursive feasibility is guaranteed and the numerical example demonstrated the benefits and limitations of using the proposed stochastic MPC formulation.

In conclusion, this dissertation demonstrated that vertical hierarchical MPC control formulations using waysets and full hierarchical control formulations using adjustable tubes are



promising approaches to multi-timescale energy management of complex systems, and are worthy for future research and development both in theory and application.

## 8.1 Future Research Directions: Theory

Within the class of hierarchical control systems, there could be numerous extensions.

- Merging the *vertical* hierarchical MPC from Chapter 3 and *full* tube-based hierarchical controller to a hierarchical controller with multiple controllers in each level for each subsystem would be really useful. This would further the applicability to a wide-range of complex real-world systems. Showing recursive feasibility would help guarantee desired control performance and constraint satisfaction.
- The extension of the proposed mission-based hierarchical control to nonlinear system dynamics will achieve better performance for systems operating over a wide range of operating conditions and thereby reducing suboptimality introduced by linearization. Recursive feasibility of the nonlinear hierarchical controller will guarantee constraint satisfaction over the entire system mission.
- Extending the *stochastic MPC* from Chapter 7 to a hierarchical framework will widen applicability to complex systems with multiple subsystems and long prediction horizons. Furthermore, analyzing the *scalability* of the hierarchical controller with increase in system complexity and prediction horizon will be very useful to achieving real-time control actuation.
- With the ongoing data revolution, it would be useful to develop a *data-driven* hierarchical controller for linear systems. Using a strong theoretical foundation in control systems, showing similar guarantees and recursive feasibility for the data-driven hierarchical controller will help guarantee safe system operation.

## 8.2 Future Research Directions: Application

To further analyze the practical applicability of the proposed model-based vertical and full hierarchical controllers, it is important to perform experimental testing and validate the performance benefits achieved in simulations in Chapters 3 and 6.

- Testing the mission-based hierarchical controller on a multi-timescale vapor compression system would validate the benefits of using hierarchical controller for these complex systems with long prediction horizons. The computational performance of solving MPC optimization problems in real-time would provide more insight to bridging the gap between theory and practice.
- Implementing the *full* tube-based hierarchical controller developed in Chapter 6 on a more complex system, such as power grids and water distribution networks, would help demonstrate the practical benefits of optimizing uncertainty online while solving the control optimization problem. Analyzing the computational performance achieved using the *full* hierarchical controller would widen its applicability to other complex systems.

## REFERENCES

- [1] Afonso, R. J. M., R. K. H. Galvao, and K. H. Kienitz (2012). Predictive control with trajectory planning in the presence of obstacles. In *UKACC International Conference on Control*, pp. 508–514. IEEE.
- [2] Alamo, T., J. M. Bravo, and E. F. Camacho (2005). Guaranteed state estimation by zonotopes. *Automatica* *41*(6), 1035–1043.
- [3] Althoff, M. (2015). An introduction to cora 2015. In *Proc. of the workshop on applied verification for continuous and hybrid systems*, pp. 120–151.
- [4] Althoff, M., N. Kochdumper, and C. Arch (2018). CORA 2018 Manual. In <https://tumcps.github.io/CORA/data/Cora2018Manual.pdf>.
- [5] Althoff, M. and B. H. Krogh (2012). Avoiding geometric intersection operations in reachability analysis of hybrid systems. In *International Conference on Hybrid Systems: Computation and Control*.
- [6] Althoff, M., O. Stursberg, and M. Buss (2010). Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear Analysis: Hybrid Systems* *4*(2), 233–249.
- [7] Asarin, E., T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler (2006). Recent progress in continuous and hybrid reachability analysis. In *International Symposium on Intelligent Control*, pp. 1582–1587. IEEE.
- [8] Barcelli, D., A. Bemporad, and G. Ripaccioli (2011). Decentralized Hierarchical Multi-Rate Control of Constrained Linear Systems. *18th IFAC World Congress* *44*(1), 277–283.
- [9] Barcelliy, D., A. Bemporadz, and G. Ripaccioliy (2010). Hierarchical multi-rate control design for constrained linear systems. In *Conference on Decision and Control*, pp. 5216–5221. IEEE.
- [10] Bemporad, A. and M. Morari (1999). Robustness in identification and control. In *Lecture Notes in Control and Information Sciences*, pp. 207–226.
- [11] Bemporad, A., M. Morari, V. Dua, and E. N. Pistikopoulos (2002). The explicit linear quadratic regulator for constrained systems. *Automatica* *38*(1), 3–20.
- [12] Betti, G., M. Farina, and R. Scattolini (2014). Distributed Model Predictive Control Made Easy. *Springer* *69*, 421–435.
- [13] Blanchini, F. (1999). Set invariance in control. *Automatica* *35*, 1747–1767.

- [14] Borrelli, F., A. Bemporad, and M. Morari (2011). *Predictive control for linear and hybrid systems*. Cambridge University Press.
- [15] Bravo, J. M., T. Alamo, and E. F. Camacho (2006). Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica* 42(10), 1745–1751.
- [16] Brdys, M., M. Grochowski, T. Gminski, K. Konarczak, and M. Drewa (2008). Hierarchical predictive control of integrated wastewater treatment systems. *Control Engineering Practice* 16, 751–767.
- [17] Calafiore, G. C. (2009). A note on the expected probability of constraint violation in sampled convex programs. In *Control Applications & Intelligent Control*, pp. 1788–1791. IEEE.
- [18] Calafiore, G. C. and M. C. Campi (2006). The scenario approach to robust control design. *IEEE Transactions on Automatic Control* 51(5), 742–753.
- [19] Chisci, L., A. Garulli, and G. Zappa (1996). Recursive state bounding by parallelotopes. *Automatica* 32(7), 1049–1055.
- [20] Doman, D. B. (2015). Rapid mission planning for aircraft thermal management. In *AIAA guidance, navigation, and control conference*, pp. 1076.
- [21] Enang, W. and C. Bannister (2017). Modelling and control of hybrid electric vehicles (A comprehensive review). *Renewable and Sustainable Energy Reviews* 74, 1210–1239.
- [22] Farahani, S. S., R. Majumdar, V. S. Prabhu, and S. Soudjani (2018). Shrinking horizon model predictive control with signal temporal logic constraints under stochastic disturbances. *IEEE Transactions on Automatic Control* 64(8), 3324–3331.
- [23] Farina, M., L. Giulioni, and R. Scattolini (2016). Stochastic linear model predictive control with chance constraints—a review. *Journal of Process Control* 44, 53–67.
- [24] Farina, M., X. Zhang, and R. Scattolini (2017). A hierarchical MPC scheme for interconnected systems. *IFAC-PapersOnLine* 50(1), 12021–12026.
- [25] Farina, M., X. Zhang, and R. Scattolini (2018a). A Hierarchical MPC Scheme for Coordination of Independent Systems With Shared Resources and Plug-and-Play Capabilities. *IEEE Transactions on Control Systems Technology* 28(2), 521–532.
- [26] Farina, M., X. Zhang, and R. Scattolini (2018b). A hierarchical multi-rate MPC scheme for inter-connected systems. *Automatica* 90, 38–46.
- [27] Fukuda, K. (2004). From the zonotope construction to the Minkowski addition of convex polytopes. *Journal of Symbolic Computation* 38(4), 1261–1272.

- [28] Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. In M. Morari and L. Thiele (Eds.), *Hybrid Systems: Computation and Control*, pp. 291–305. Springer.
- [29] Girard, A., C. L. Guernic, and O. Maler (2006). Efficient computation of reachable sets of linear time-invariant systems with inputs. In *International workshop on hybrid systems: Computation and control*, pp. 257–271. Springer.
- [30] Girard, A. and C. Le Guernic (2008). Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
- [31] Gorecki, T. T., A. Bitlislioglu, G. Stathopoulos, and C. N. Jones (2015). Guaranteeing input tracking for constrained systems: Theory and application to demand response. In *American Control Conference*, pp. 232–237. IEEE.
- [32] Grosso, J., C. Ocampo-Martínez, V. Puig, and B. Joseph (2014). Chance-constrained model predictive control for drinking water networks. *Journal of process control* 24(5), 504–516.
- [33] Guerrero, J. M., M. Chandorkar, T.-L. Lee, and P. C. Loh (2013). Advanced Control Architectures for Intelligent Microgrids - Part I: Decentralized and Hierarchical Control. *IEEE Transactions on Industrial Electronics* 60(4), 1254–1262.
- [34] Gurobi Optimization Inc. (2020). Gurobi Optimizer Reference Manual. [www.gurobi.com](http://www.gurobi.com), 1 – 969.
- [35] Han, D., A. Rizaldi, A. El-Guindy, and M. Althoff (2016). On enlarging backward reachable sets via Zonotopic set membership. In *IEEE International Symposium on Intelligent Control*, pp. 1–8.
- [36] Hartley, E. N., P. A. Trodden, A. G. Richards, and J. M. Maciejowski (2012). Model predictive control system design and implementation for spacecraft rendezvous. *Control Engineering Practice* 20(7), 695–713.
- [37] Herceg, M., M. Kvasnica, C. N. Jones, and M. Morari (2013). Multi-parametric toolbox 3.0. In *European Control Conference*, pp. 502–510. IEEE.
- [38] Hewing, L. and M. N. Zeilinger (2018). Stochastic model predictive control for linear systems using probabilistic reachable sets. In *Conference on Decision and Control*, pp. 5182–5188. IEEE.
- [39] Ioan, D., I. Prodan, F. Stoican, S. Olaru, and S.-I. Niculescu (2019). Complexity bounds for obstacle avoidance within a zonotopic framework. In *American Control Conference*, pp. 335–340. IEEE.

- [40] Irfan, M., J. Iqbal, A. Iqbal, Z. Iqbal, R. A. Riaz, and A. Mehmood (2017). Opportunities and challenges in control of smart grids – Pakistani perspective. *Renewable and Sustainable Energy Reviews* 71, 652–674.
- [41] Joševski, M. and D. Abel (2014). Multi-time scale model predictive control framework for energy management of hybrid electric vehicles. In *Conference on Decision and Control*, pp. 2523–2528. IEEE.
- [42] Kennel, F., D. Görges, and S. Liu (2013). Energy Management for Smart Grids With Electric Vehicles Based on Hierarchical MPC. *IEEE Transactions on industrial informatics* 9(3), 1528–1537.
- [43] Khan, I., Z. Li, Y. Xu, and W. Gu (2016). Distributed control algorithm for optimal reactive power control in power grids. *International Journal of Electrical Power and Energy Systems* 83, 505–513.
- [44] Kim, Y., X. Zhang, J. Guanetti, and F. Borrelli (2018). Robust model predictive control with adjustable uncertainty sets. In *Conference on Decision and Control*, pp. 5176–5181. IEEE.
- [45] Kochdumper, N. and M. Althoff (2020). Sparse polynomial zonotopes: A novel set representation for reachability analysis. *IEEE Transactions on Automatic Control* 66(9), 4043–4058.
- [46] Koeln, J., H. Pangborn, M. Williams, M. Kawamura, and A. Alleyne (2020). Hierarchical Control of Aircraft Electro-Thermal Systems. *IEEE Transactions on Control Systems Technology* 28(4), 1218–1232.
- [47] Koeln, J. P. and A. G. Alleyne (2018). Two-level hierarchical mission-based model predictive control. In *American Control Conference*, pp. 2332–2337. IEEE.
- [48] Koeln, J. P. and B. M. Hencsey (2019). Constrained hierarchical mpc via zonotopic waysets. In *American Control Conference*, pp. 4237–4244. IEEE.
- [49] Koeln, J. P., V. Raghuraman, and B. M. Hencsey (2020). Vertical hierarchical MPC for constrained linear systems. *Automatica* 113, 108817.
- [50] Kolmanovsky, I. and E. G. Gilbert (1998). Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering* 4(4), 317–367.
- [51] Kopetzki, A.-K., B. Schürmann, and M. Althoff (2017). Methods for order reduction of zonotopes. In *Conference on Decision and Control*, pp. 5626–5633. IEEE.

- [52] Kurzhanski, A. B. and P. Varaiya (2000). Ellipsoidal techniques for reachability analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
- [53] Kurzhanskiy, A. A. and P. Varaiya (2007). Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Transactions on Automatic Control* *52(1)*, 26–38.
- [54] Langson, W., I. Chrysochoos, S. V. Raković, and D. Q. Mayne (2004). Robust model predictive control using tubes. *Automatica* *40*, 125–133.
- [55] Le, V. T. H., C. Stoica, T. Alamo, E. F. Camacho, and D. Dumur (2013). Zonotopic guaranteed state estimation for uncertain systems. *Automatica* *49(11)*, 3418–3424.
- [56] Limon, D., I. Alvarado, T. Alamo, and E. F. Camacho (2010). Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control* *20(3)*, 248–260.
- [57] Lofberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *International conference on robotics and automation*, pp. 284–289. IEEE.
- [58] Ma, Y., S. Vichik, and F. Borrelli (2012). Fast stochastic mpc with optimal risk allocation applied to building control systems. In *Conference on Decision and Control*, pp. 7559–7564. IEEE.
- [59] Maler, O. (2008). Computing reachable sets: An introduction. *Technical report*, 1–8.
- [60] Mayne, D., M. Seron, and S. Raković (2005). Robust Model Predictive Control of Constrained Linear Systems with Bounded Disturbances. *Automatica* *41*, 219–224.
- [61] Mayne, D. Q. (2014). Model predictive control: Recent developments and future promise. *Automatica* *50(12)*, 2967–2986.
- [62] Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. Scokaert (2000). Constrained model predictive control: Stability and optimality. *Automatica* *36*, 789–814.
- [63] Negenborn, R. R., A. Sahiir, Z. Lukszer, B. De Schutter, and M. Morari (2009). A non-iterative cascaded predictive control approach for control of irrigation canals. In *International Conference on Systems, Man and Cybernetics*, pp. 3552–3557. IEEE.
- [64] Ocampo-Martinez, C., D. Barcelli, V. Puig, and A. Bemporad (2012). Hierarchical and decentralised model predictive control of drinking water networks: application to Barcelona case study. *IET Control Theory & Applications* *6(1)*, 62–71.
- [65] Raghuraman, V. and J. P. Koeln (2021). Tube-based robust mpc with adjustable uncertainty sets using zonotopes. In *American Control Conference*, pp. 462–469. IEEE.

- [66] Raghuraman, V. and J. P. Koeln (2022). Set operations and order reductions for constrained zonotopes. *Automatica* 139, 110204.
- [67] Raghuraman, V., V. Renganathan, T. H. Summers, and J. P. Koeln (2020). Hierarchical mpc with coordinating terminal costs. In *American Control Conference*, pp. 4126–4133. IEEE.
- [68] Raković, S. V., E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne (2005). Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control* 50(3), 406–410.
- [69] Richards, A. and J. How (2006). Robust stable model predictive control with constraint tightening. In *American Control Conference*, pp. 6–11.
- [70] Richards, A. and J. P. How (2003). Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility. In *American Control Conference*, pp. 4034–4040. IEEE.
- [71] Sadraddini, S. and R. Tedrake (2019). Linear encodings for polytope containment problems. In *Conference on Decision and Control*, pp. 4367–4372. IEEE.
- [72] Sampathnarayanan, B., S. Onori, and S. Yurkovich (2014). An optimal regulation strategy with disturbance rejection for energy management of hybrid electric vehicles. *Automatica* 50, 128–140.
- [73] Scattolini, R. (2009). Architectures for distributed and hierarchical Model Predictive Control - A review. *Journal of Process Control* 19, 723–731.
- [74] Scattolini, R. and P. Colaneri (2007). Hierarchical model predictive control. In *Conference on Decision and Control*, pp. 4803–4808.
- [75] Scattolini, R., P. Colaneri, and D. D. Vito (2008). A switched MPC approach to hierarchical control. *17th IFAC World Congress* 41(2), 7790–7795.
- [76] Schildbach, G. and M. Morari (2015). Scenario mpc for linear time-varying systems with individual chance constraints. In *American Control Conference*, pp. 415–421. IEEE.
- [77] Schulze Darup, M., R. Schaich, and M. Cannon (2017). How scaling of the disturbance set affects robust positively invariant sets for linear systems. *International Journal of Robust and Nonlinear Control* 27(16), 3236–3258.
- [78] Scibilia, F., S. Olaru, and M. Hovd (2011). On feasible sets for MPC and their approximations. *Automatica* 47, 133–139.



- [79] Scott, J. K., D. M. Raimondo, G. R. Marseglia, and R. D. Braatz (2016). Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica* 69, 126–136.
- [80] Seok, J., I. Kolmanovsky, and A. Girard (2017). Coordinated model predictive control of aircraft gas turbine engine and power system. *Journal of Guidance, Control, and Dynamics* 40(10), 2538–2555.
- [81] Shekhar, R. C., M. Kearney, and I. Shames (2015). Robust Model Predictive Control of Unmanned Aerial Vehicles Using Waysets. *Journal of Guidance, Control, and Dynamics* 38, 1898–1907.
- [82] Su, Y., K. K. Tan, and T. H. Lee (2013). Computation delay compensation for real time implementation of robust model predictive control. *Journal of Process Control* 23(9), 1342–1349.
- [83] Tiwary, H. R. (2008). On the hardness of computing intersection, union and minkowski sum of polytopes. *Discrete and Computational Geometry* 40(3), 469–479.
- [84] Trodden, P. (2016). A One-Step Approach to Computing a Polytopic Robust Positively Invariant Set. *IEEE Transactions on Automatic Control* 61(12), 4100–4105.
- [85] Trodden, P. A. and J. M. Maestre (2017). Distributed predictive control with minimization of mutual disturbances. *Automatica* 77, 31–43.
- [86] Vermillion, C., A. Menezes, and I. Kolmanovsky (2014). Stable hierarchical model predictive control using an inner loop reference model and  $\lambda$ -contractive terminal constraint sets. *Automatica* 50, 92–99.
- [87] Vrettos, E., F. Oldewurtel, F. Zhu, and G. Andersson (2014). Robust provision of frequency reserves by office building aggregations. *IFAC Proceedings* 47, 12068–12073.
- [88] Wang, F., M. A. M. Zulkefli, Z. Sun, and K. A. Stelson (2013). Investigation on the Energy Management Strategy for Hydraulic Hybrid Wheel Loaders. *Dynamic Systems and Control Conference* 56123, p. V001T11A005.
- [89] Wang, K. and S. Gros (2021). Recursive Feasibility of Stochastic Model Predictive Control with Mission-Wide Probabilistic Constraints. In *Conference on Decision and Control*, pp. 2312–2317.
- [90] Wang, L., A. D. Ames, and M. Egerstedt (2017). Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics* 33(3), 661–674.
- [91] Wang, W. and J. P. Koeln (2020). Hierarchical Multi-Timescale Energy Management for Hybrid-Electric Aircraft. *ASME Dynamic Systems and Control Conference* 84270, p. V001T11A002.

- [92] Xue, B., A. Easwaran, and Z. She (2017). Under-Approximating Backward Reachable Sets by Semialgebraic Sets Verification and Analysis of Hybrid Systems View project Underapproximating Backward Reachable Sets by Semialgebraic Sets. *IEEE Transactions on Automatic Control* 62(10), 5185–5197.
- [93] Zamora, R. and A. K. Srivastava (2010). Controls for microgrids with storage: Review, challenges, and research needs. *Renewable and Sustainable Energy Reviews* 14(7), 2009–2018.
- [94] Zhang, X., M. Kamgarpour, A. Georghiou, P. Goulart, and J. Lygeros (2017). Robust optimal control with adjustable uncertainty sets. *Automatica* 75, 249–259.
- [95] Zhang, X., M. Kamgarpour, P. Goulart, and J. Lygeros (2014). Selling robustness margins: A framework for optimizing reserve capacities for linear systems. In *Conference on Decision and Control*, pp. 6419–6424. IEEE.

## BIOGRAPHICAL SKETCH

Vignesh Raghuraman received his BE degree in 2013 from Sri Sairam Engineering College (affiliated to Anna University, Chennai) in Instrumentation and Control Engineering. He received MS degree in 2018 from Arizona State University, Tempe in Electrical Engineering. He will receive his doctoral degree in Mechanical Engineering from The University of Texas at Dallas. He is a member of the Energy Systems Control Laboratory headed by Dr. Justin P. Koeln. He was also one of the finalists for the best student paper award at the American Control Conference, 2021. His research interests include model predictive control, hierarchical and distributed control for multi-timescale systems, and set theoretic methods for control.

## CURRICULUM VITAE

# Vignesh Raghuraman

### Contact Information:

Department of Mechanical Engineering  
The University of Texas at Dallas  
800 W. Campbell Rd.  
Richardson, TX 75080-3021, U.S.A.  
Email: vignesh.raghuraman@utdallas.edu

### Educational History:

PhD, Mechanical Engineering, The University of Texas at Dallas, Richardson, TX, USA	2018 - 2022
MS, Electrical Engineering, Arizona State University, Tempe, AZ, USA	2015 - 2018
BE, Instrumentation & Control Engineering, Sri Sairam Engineering College, Chennai, Anna University, India	2009 - 2013

### Employment History:

Controls Algorithm Intern, ASML, San Diego, California, USA	2021
Graduate Research Assistant, Mechanical Engineering, The University of Texas at Dallas, Richardson, TX, USA	2018 - 2022
Project Engineer, Wipro Technologies Private Limited, Chennai, Tamil Nadu, India	2014 - 2015

### Selected Awards:

<b>Finalist-Best Student Paper Award</b> , American Control Conference, USA	2021
<b>Second prize</b> , Poster Competition, University of Texas at Dallas, USA	2019
<b>Best Paper</b> , Paper Presentation, SRM University, India	2012

### Research Interests:

Model Predictive Control, Hierarchical and Distributed Control for multi-timescale systems, Set-based computing, and Set-theoretic methods for control.

### Publications:

*Journal Articles:*

1. **Raghuraman, Vignesh**, Koeln, J. P. “Long duration stochastic MPC with mission-wide probabilistic constraints using waysets”, *Letters to Control Systems Society*, 2023. (under preparation)
2. **Raghuraman, Vignesh**, Koeln, J. P. “Hierarchical MPC for coupled subsystems using adjustable tubes”, *Automatica*, 2022.
3. **Raghuraman, Vignesh**, Koeln, J. P. “Set operations and order reductions for constrained zonotopes”, *Automatica*, 139, 2022: 110204.
4. Koeln, J.P, **Raghuraman, Vignesh**, Hencey, Brandon M., “Vertical hierarchical MPC for constrained linear systems”, *Automatica*, 113, 2020: 108817.

#### *Conference Papers*

1. **Raghuraman, Vignesh**, Koeln, J.P, “Tube-based Robust MPC with adjustable uncertainty sets using zonotopes”, *ACC 2021*, pp. 462-469.
2. **Raghuraman, Vignesh**, Renganathan, V., Summers, T.H., and Koeln, J P., “Hierarchical MPC with Coordinating Terminal Costs.”, *ACC 2020*, pp. 4126-4133.

### **Teaching Experience:**

#### **Graduate Teaching Assistant, The University of Texas at Dallas**

MECH 3340: System Dynamics Modeling & Analysis Spring 2020

#### **Grader, Arizona State University**

EE 304: Signals and Systems - II Spring 2017-Spring 2018

EE 586: Nonlinear Systems Spring 2018

MAE 417: System Dynamics and Controls - II Spring 2018

EE 582: Linear Systems Fall 2017

EE 591: Computer Controlled Systems Spring 2017

#### **Laboratory Aide, Arizona State University**

EE 202: Electrical Circuits Fall 2017

### **Services:**

#### **Journal Article Reviewer**

IEEE Transactions on Automatic Control ◦ ASME-DSC letters

#### **Conference Paper Reviewer**

IEEE American Control Conference (ACC) ◦ IEEE Conference on Decision and Control (CDC) ◦ Modeling, Estimation, and Control Conference (MECC)