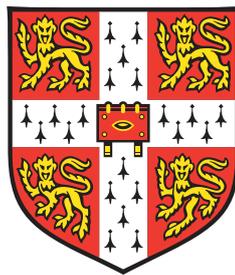


# Variable Horizon Model Predictive Control: Robustness & Optimality

Rohan C. Shekhar

Pembroke College



Control Group  
Department of Engineering  
University of Cambridge

A thesis submitted for the degree of  
*Doctor of Philosophy*

16th April 2012  
(Revised 8th June 2012)



## Abstract

### Variable Horizon Model Predictive Control: Robustness & Optimality

Variable Horizon Model Predictive Control (VH-MPC) is a form of predictive control that includes the horizon length as a decision variable in the constrained optimisation problem solved at each iteration. It has been recently applied to *completion* problems, where the system state is to be steered to a closed set in finite time. The behaviour of the system once completion has occurred is not considered part of the control problem.

This thesis is concerned with three aspects of robustness and optimality in VH-MPC completion problems. In particular, the thesis investigates robustness to well defined but unpredictable changes in system and controller parameters, robustness to bounded disturbances in the presence of certain input parameterisations to reduce computational complexity, and optimal robustness to bounded disturbances using tightened constraints. In the context of linear time invariant systems, new theoretical contributions and algorithms are developed.

Firstly, changing dynamics, constraints and control objectives are addressed by introducing the notion of feasible contingencies. A novel algorithm is proposed that introduces extra prediction variables to ensure that anticipated new control objectives are always feasible, under changed system parameters. In addition, a modified constraint tightening formulation is introduced to provide robust completion in the presence of bounded disturbances. Different contingency scenarios are presented and numerical simulations demonstrate the formulation's efficacy.

Next, complexity reduction is considered, using a form of input parameterisation known as move blocking. After introducing a new notation for move blocking, algorithms are presented for designing a move-blocked VH-MPC controller. Constraints are tightened in a novel way for robustness, whilst ensuring that guarantees of recursive feasibility and finite-time completion are preserved. Simulations are used to illustrate the effect of an example blocking scheme on computation time, closed-loop cost, control inputs and state trajectories.

Attention is now turned towards mitigating the effect of constraint tightening policies on a VH-MPC controller's region of attraction. An optimisation problem is formulated to maximise the volume of an inner approximation to the region of attraction, parameterised in terms of the tightening policy. Alternative heuristic approaches are also proposed to deal with high state dimensions. Numerical examples show that the new technique produces substantially improved regions of attraction in comparison to other proposed approaches, and greatly reduces the maximum required prediction horizon length for a given application.

Finally, a case study is presented to illustrate the application of the new theory developed in this thesis to a non-trivial example system. A simplified nonlinear surface excavation machine and material model is developed for this purpose. The model is stabilised with an inner-loop controller, following which a VH-MPC controller for autonomous trajectory generation is designed using a discretised, linearised model of the stabilised system. Realistic simulated trajectories are obtained from applying the controller to the stabilised system and incorporating the ideas developed in this thesis.

These ideas improve the applicability and computational tractability of VH-MPC, for both traditional applications as well as those that go beyond the realm of vehicle manoeuvring.

Rohan C. Shekhar  
Pembroke College



# Acknowledgements

I would firstly like to thank my supervisor, Professor Jan Maciejowski, for his guidance and support during my years of research in Cambridge. His knowledge of the field and passion for his research has been truly inspirational. It has been an honour to work with him as his student.

Additionally, I would like to thank Dr Edward Hartley for his advice and patience. His outstanding knowledge of all things MPC has helped me tremendously and I am grateful for our numerous fruitful discussions, from which emerged many of the ideas presented in this thesis.

I am also grateful to the Menzies Foundation and the Cambridge Commonwealth Trust for their financial support throughout my time in Cambridge. I am privileged to be a part of the community of scholars who have benefitted from these organisations.

Many thanks go to the members of the Control Group for providing an intellectually stimulating environment for research. I am especially grateful to my fellow students for their friendship and company throughout the long days prior to submission of this thesis. I wish them well in their future endeavours.

A special thanks goes to the Pembroke College community. In particular, I would like to acknowledge all of my friends in the Graduate Parlour, for making my time in Cambridge truly unforgettable.

Most of all, I would like to thank my family, for their love and unwavering support throughout the last four years.



# Declaration

As required by University Statute, I hereby declare that this dissertation is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other university. The dissertation is a result of my own work and includes nothing which is the outcome of work done in collaboration, except where specified explicitly in the text. I also declare that the length of this dissertation is less than 65,000 words and the number of figures is less than 150.



Rohan C. Shekhar  
Pembroke College  
Cambridge  
16th April 2012



# Contents

<b>Contents</b>	<b>ix</b>
<b>List of Figures</b>	<b>xv</b>
<b>List of Tables</b>	<b>xvii</b>
<b>Nomenclature</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Outline . . . . .	2
1.2 Notational Conventions . . . . .	4
<b>2 Background</b>	<b>5</b>
2.1 Overview of Model Predictive Control . . . . .	5
2.1.1 Principle of Operation . . . . .	5
2.1.2 The Receding Horizon Principle . . . . .	6
2.1.3 Constraint Handling . . . . .	7
2.2 Mathematical Formulation of MPC . . . . .	7
2.3 Analysis of Stability . . . . .	10
2.3.1 Sufficient Conditions from Lyapunov Theory . . . . .	10
2.3.2 Linear Quadratic MPC . . . . .	12
2.4 Variable Horizon MPC . . . . .	13
2.4.1 Completion . . . . .	14
2.4.2 Cost Functions . . . . .	15
2.4.3 An Alternative Formulation . . . . .	16
2.5 Robust MPC . . . . .	16
2.5.1 Min-Max Predictive Control . . . . .	16
2.5.2 Affine Feedback Policies . . . . .	17
2.5.3 Constraint Tightening . . . . .	17
2.5.4 Tube Model Predictive Control . . . . .	17

2.6	Robust VH-MPC with Constraint Tightening . . . . .	18
2.6.1	Formulation . . . . .	19
2.6.2	Controller Properties . . . . .	20
2.6.3	Disturbance Feedback Formulation . . . . .	21
2.7	Optimisation . . . . .	21
2.7.1	Mixed-Integer Formulation . . . . .	21
2.7.2	Implementation . . . . .	22
<b>3</b>	<b>Feasible Contingencies</b>	<b>25</b>
3.1	Introduction . . . . .	25
3.1.1	Nomenclature . . . . .	26
3.2	Feasible Contingencies . . . . .	26
3.2.1	Nominal Control Problem . . . . .	26
3.2.2	Robustness . . . . .	28
3.2.3	Linear Case . . . . .	28
3.3	Robust MPC with Contingencies . . . . .	29
3.3.1	MIP implementation . . . . .	37
3.4	Contingency Scenarios . . . . .	38
3.4.1	Costed Contingencies . . . . .	39
3.4.2	Multiple & Prioritised Contingencies . . . . .	41
3.4.3	State-Dependent Contingencies . . . . .	42
3.5	Simplifications . . . . .	43
3.5.1	Fixed Contingency Horizon . . . . .	44
3.5.2	Explicit Contingency Feasibility . . . . .	44
3.5.3	Prediction Variable Reduction . . . . .	44
3.6	Examples . . . . .	44
3.6.1	Double-Integrator Vehicle . . . . .	45
3.6.2	Cart-Pendulum . . . . .	53
3.7	Conclusion . . . . .	55
3.8	Acknowledgements . . . . .	57
<b>4</b>	<b>Move Blocking for Complexity Reduction</b>	<b>59</b>
4.1	Introduction . . . . .	59
4.1.1	Nomenclature . . . . .	60
4.2	Problem Definition . . . . .	60
4.3	Move Blocking Formulation . . . . .	61

4.4	Blocked VH-MPC . . . . .	66
4.4.1	Nominal Controller . . . . .	67
4.4.2	Robustness . . . . .	70
4.5	Implementation . . . . .	74
4.5.1	Input Condensed Prediction Model . . . . .	74
4.5.2	State & Input Condensed Prediction Model . . . . .	75
4.6	Example . . . . .	76
4.6.1	The Partitioned Horizon . . . . .	76
4.6.2	Formulation . . . . .	76
4.6.3	Simulation . . . . .	77
4.6.4	Analysis . . . . .	78
4.7	Conclusion . . . . .	82
4.8	Acknowledgements . . . . .	82
<b>5</b>	<b>Optimal Constraint Tightening Policies</b>	<b>83</b>
5.1	Introduction . . . . .	83
5.1.1	Nomenclature . . . . .	84
5.2	Problem Formulation . . . . .	84
5.3	Affine Parameterisation of Tightened Sets . . . . .	86
5.4	Maximising the Region of Attraction . . . . .	88
5.4.1	Parameterisation . . . . .	88
5.4.2	Optimisation . . . . .	91
5.5	Examples . . . . .	93
5.5.1	Symmetric Target Region Placement . . . . .	94
5.5.2	Horizon Length Reduction . . . . .	97
5.5.3	Asymmetric Target Region Placement . . . . .	97
5.5.4	Effect on Cost . . . . .	100
5.5.5	Heuristic Optimisation . . . . .	100
5.6	Discussion . . . . .	102
5.6.1	Curse of Dimensionality . . . . .	102
5.6.2	Computational Complexity . . . . .	102
5.6.3	Volume Calculation . . . . .	103
5.7	Conclusion . . . . .	104
5.8	Acknowledgements . . . . .	104

<b>6</b>	<b>Case Study - Surface Excavation</b>	<b>105</b>
6.1	Introduction . . . . .	105
6.1.1	Nomenclature . . . . .	107
6.2	Mechanism Model . . . . .	107
6.2.1	Kinematic Equations . . . . .	108
6.2.2	Equations of Motion . . . . .	111
6.2.3	External & Dissipative Forces . . . . .	112
6.3	Material Interaction Model . . . . .	113
6.3.1	Implementation . . . . .	115
6.4	System Identification . . . . .	115
6.4.1	Pre-Stabilising Controller . . . . .	116
6.4.2	Identification of Trajectory Tracking System . . . . .	117
6.5	VH-MPC Controller Design . . . . .	118
6.5.1	Constraints . . . . .	119
6.5.2	Cost Function . . . . .	121
6.6	Simulation . . . . .	121
6.6.1	Normal Operation . . . . .	121
6.6.2	Stall Contingency . . . . .	122
6.6.3	Move Blocking . . . . .	126
6.7	Conclusion . . . . .	126
6.8	Acknowledgements . . . . .	126
<b>7</b>	<b>Conclusion</b>	<b>129</b>
7.1	Contributions . . . . .	129
7.1.1	Feasible Contingencies . . . . .	129
7.1.2	Complexity-Reduced VH-MPC with Move Blocking . . . . .	130
7.1.3	Optimal Constraint Tightening Policies . . . . .	130
7.1.4	Case Study on Surface Excavation . . . . .	130
7.2	Future Research Directions . . . . .	130
7.2.1	Uncertain Contingencies . . . . .	131
7.2.2	Optimised & Multiplexed Move Blocking . . . . .	131
7.2.3	Cost Optimal Constraint Tightening Policies . . . . .	131
7.2.4	Combinations of Strategies . . . . .	131
7.2.5	Shovel & Material Model Generalisation . . . . .	132
	<b>Bibliography</b>	<b>133</b>

---

<b>A</b>	<b>Constraint Tightening Addenda</b>	<b>139</b>
A.1	Derivation of the Max-Dist Policy . . . . .	139
A.2	VH-ROA Plots for Different Policies . . . . .	140
<b>B</b>	<b>Shovel Model &amp; Controller Details</b>	<b>147</b>
B.1	Equations of Motion . . . . .	147
B.2	Controller Parameters . . . . .	148



# List of Figures

2.1	The receding horizon principle . . . . .	6
2.2	Qualitative illustration of MPC's constraint awareness . . . . .	8
3.1	Illustration of MPC with Contingency Predictions . . . . .	31
3.2	Trajectories with contingency . . . . .	48
3.3	Primary trajectory velocities . . . . .	48
3.4	Primary trajectory control inputs . . . . .	49
3.5	Primary trajectory fuel use . . . . .	49
3.6	Trajectories with mean-costed contingency . . . . .	50
3.7	Trajectories for two contingency sets . . . . .	51
3.8	Trajectories for two contingency sets with priority . . . . .	51
3.9	Trajectories for state-dependent contingency . . . . .	52
3.10	Fuel use for state-dependent contingency . . . . .	52
3.11	Diagram of cart-pendulum system . . . . .	53
3.12	Step response for cart states . . . . .	56
3.13	Step response for pendulum states . . . . .	56
3.14	Control input to cart . . . . .	57
4.1	Values of functions $\Lambda(\cdot, \cdot)$ and $\mu(\cdot, \cdot)$ for blocking matrix corresponding to $N = 9$ and block lengths $\{2, 4, 3\}$ . . . . .	63
4.2	Illustration of the partitioned horizon regime . . . . .	77
4.3	Cost comparison of different block lengths in the partitioned horizon . . . . .	79
4.4	Computation time comparison different block lengths in the partitioned horizon . . . . .	79
4.5	Closed-loop trajectories for $\lambda = 1$ . . . . .	80
4.6	Closed-loop trajectories for $\lambda = 4$ . . . . .	80
4.7	Closed-loop trajectories for $\lambda = 40$ . . . . .	80
4.8	Control inputs for $\lambda = 1$ . . . . .	81
4.9	Control inputs for $\lambda = 4$ . . . . .	81
4.10	Control inputs for $\lambda = 40$ . . . . .	81

5.1	Four-step ROA for $\mathcal{T}_1$ using $\mathcal{P}_{\max\text{-vol}}$ . . . . .	95
5.2	Total variable horizon ROA for $\mathcal{T}_1$ using $\mathcal{P}_{\max\text{-vol}}$ . . . . .	96
5.3	Four-step ROA comparison over different policies for $\mathcal{T}_1$ . . . . .	96
5.4	Comparison of 4-step ROA for $\mathcal{T}_1$ using $\mathcal{P}_{\max\text{-vol}}$ with 14-step ROA for $\mathcal{T}_1$ using $\mathcal{P}_{2\text{-step-nil}}$ . . . . .	97
5.5	Four-step ROA for $\mathcal{T}_2$ using $\mathcal{P}_{\max\text{-vol}}$ compared with nominal ROA . . . . .	98
5.6	Total variable horizon ROA for $\mathcal{T}_2$ using $\mathcal{P}_{\max\text{-vol}}$ . . . . .	99
5.7	Four-step ROA comparison over different policies for $\mathcal{T}_2$ . . . . .	99
6.1	The rope shovel . . . . .	106
6.2	Rope shovel excavation trajectory . . . . .	106
6.3	Kinematic vectors . . . . .	108
6.4	Mechanism model showing generalised coordinates . . . . .	109
6.5	Static balance of material failure forces . . . . .	114
6.6	Performance of trajectory tracking controller . . . . .	117
6.7	Performance of identified linear model for crowd length and angle . . . . .	118
6.8	Performance of identified linear model for payload . . . . .	119
6.9	Actual trajectory and prediction from linear model . . . . .	122
6.10	Crowd position and angle compared to reference . . . . .	123
6.11	Payload acquisition . . . . .	123
6.12	Hoist and crowd forces from the trim point . . . . .	124
6.13	Cutting resistance force . . . . .	124
6.14	Trajectory comparison with crowd stall contingency . . . . .	125
6.15	Payload comparison with crowd stall contingency . . . . .	125
6.16	Trajectory comparison with move blocking . . . . .	127
6.17	Payload comparison with move blocking . . . . .	127
A.1	Nominal VH-ROA for $\mathcal{T}_1$ . . . . .	140
A.2	VH-ROA for $\mathcal{T}_1$ using max-vol . . . . .	141
A.3	VH-ROA for $\mathcal{T}_1$ using max-dist . . . . .	141
A.4	VH-ROA for $\mathcal{T}_1$ using 3-step nil . . . . .	142
A.5	VH-ROA for $\mathcal{T}_1$ using 2-step nil . . . . .	142
A.6	Nominal VH-ROA for $\mathcal{T}_2$ . . . . .	143
A.7	VH-ROA for $\mathcal{T}_2$ using max-vol . . . . .	143
A.8	VH-ROA for $\mathcal{T}_2$ using max-dist . . . . .	144
A.9	VH-ROA for $\mathcal{T}_2$ using 3-step-nil . . . . .	144
A.10	VH-ROA for $\mathcal{T}_2$ using 2-step-nil . . . . .	145

# List of Tables

5.1	ROA volumes over different policies for $\mathcal{T}_1$ . . . . .	95
5.2	ROA volumes over different policies for $\mathcal{T}_2$ . . . . .	98
5.3	Mean costs for Monte Carlo simulation using different policies . . . . .	100
5.4	ROA volumes over different policies for 3-state system . . . . .	101
6.1	Kinematic parameters . . . . .	109
6.2	Dynamic parameters . . . . .	110
6.3	Material properties . . . . .	115



# Nomenclature

## Variables, Functions & Constants

$0_{n \times m}$	$n \times m$ matrix of zeros
$\mathbf{1}_n$	$n$ -dimensional vector of ones
$I_n$	$n \times n$ identity matrix
$\delta[i]$	discrete unit impulse at time $i$
$J^*(\cdot)$	value function
$\hat{J}(\cdot)$	suboptimal value function
$N$	prediction horizon length
$u$	input vector
$w$	disturbance vector
$x$	state vector
$y$	output vector

## Sets

$\mathbb{B}_p^n(\epsilon)$	$n$ -dimensional $p$ -norm ball with radius $\epsilon$
$\emptyset$	empty set
$Q\mathcal{W}$	image of set $\mathcal{W}$ under matrix $Q$ ( $\{x \mid x = Qw, \forall w \in \mathcal{W}\}$ )
$\mathbb{N}$	set of natural numbers
$\mathbb{N}_{[a,b]}$	set of natural numbers in the range $a, \dots, b$
$\mathbb{R}$	set of real numbers
$\mathbb{R}^+$	set of positive real numbers
$\mathcal{T}$	terminal / target constraint set
$\mathcal{U}$	input constraint set
$\mathcal{W}$	state disturbance set
$\mathcal{X}$	state constraint set

$\mathcal{Y}$  output constraint set

$\mathbb{Z}$  set of integers

$\mathbb{Z}_{[a,b]}$  set of integers in the range  $a, \dots, b$

**Operators**

$\mathcal{C}$  matrix horizontal concatenation operator

$|\cdot|$  row-wise absolute value, set cardinality

$\lceil \cdot \rceil$  ceiling function

$\triangleq$  defined as

$\text{diag}\{\cdot\}$  diagonal matrix

$\text{diam}(\cdot)$  polytope diameter

$\lfloor \cdot \rfloor$  floor function

iff,  $\iff$  if and only if

$\implies$  implies

$\cap$  intersection

$\otimes$  Kronecker product

$\mapsto$  maps to

$\|\cdot\|_p$  p-norm

$\oplus$  Minkowski sum (Chapter 3), direct sum (Chapter 4)

$\ominus$  Pontryagin difference

$\mathbb{P}_n(\cdot)$  projection onto  $\mathbb{R}^n$

$\subset$  subset

$\subseteq$  subset or equal to

$\supset$  superset

$\supseteq$  superset or equal to

$\cup$  union

$\text{vol}(\cdot)$  polytope volume

**Acronyms**

CT Constraint Tightening

DARE Discrete Algebraic Riccati Equation

DLE Discrete Lyapunov Equation

LMI Linear Matrix Inequality

LP Linear Program  
LQR Linear Quadratic Regulator  
MILP Mixed-Integer Linear Program  
MIP Mixed-Integer Programming  
MPC Model Predictive Control  
QP Quadratic Program  
RMPC Robust Model Predictive Control  
ROA Region of Attraction  
SDP Semidefinite Program  
SOCP Second Order Cone Program  
VH-MPC Variable Horizon Model Predictive Control  
VH-ROA Variable Horizon Region of Attraction

**Definitions**

*Compact Set:* a set which is closed and bounded

*Completion:* the time at which the system state first enters a specified target set

*Move Blocking:* an input parameterisation strategy where groups of adjacent-in-time inputs, or perturbations to a stabilising control law, are constrained to have the same value

*Output Admissibility:* a property of the state and control input of the system that ensures satisfaction of output constraints

*Polyhedron:* a closed, but not necessarily bounded set formed by the intersection of half-spaces

*Polytope:* a compact set formed by the intersection of half-spaces

*Recursive Feasibility:* the property of an MPC controller that ensures the existence of a feasible solution to the optimisation problem at all time steps from an initial feasible solution

*Region of Attraction:* the set of states for which there exists an initial feasible solution to the MPC optimisation problem

*Subspace:* a subset of a vector space that is closed under addition and scalar multiplication



# Chapter 1

## Introduction

Model Predictive Control (MPC) is an optimisation-based control paradigm that has been used successfully in the process industries for many years at a supervisory level. Its intuitive formulation and unique constraint-handling ability permit system operation closer to constraint boundaries, which can improve performance. Recent advances in computational power have made it possible to implement MPC directly at a low level on systems requiring fast update rates. This has been further aided by algorithmic advances, which have improved solution times for the constrained finite-horizon optimisation problem solved at each iteration of the MPC controller [WB10]. Conditions that guarantee convergence of predictive controllers for regulation problems are well established in the literature [May+00; RM09]. In such problems the system's output is required to asymptotically track some reference value.

However, regulation problems are not always an appropriate description of the desired control objectives. For applications like vehicle manoeuvring, the control objective may be to simply reach a specified target region, whilst minimising some cost function. Hence, the notion of stability is replaced with that of *completion* [RH03]; once the target region is reached (at completion), the problem ends. In essence, it is the transient that is of importance in problems of this nature. It is assumed that, if required, another controller is deployed after the completion time. For most applications of completion problems, it is desirable to have a finite completion time. Whilst stability is no longer required, it is also desirable to maintain a recursively feasible optimisation problem at each controller iteration up to completion.

Variable Horizon Model Predictive Control (VH-MPC) [MM93] has been shown to provide these guarantees. In this form of MPC, the horizon length is a decision variable in the constrained optimisation problem solved at each iteration. Recent applications of VH-MPC have addressed completion problems, where the requirements on terminal set invariance, that are central to the stability proofs of classical MPC, are relaxed. With Constraint Tightening (CT) applied for robustness, the guarantees of recursive feasibility and finite-time completion can be shown to hold for linear systems controlled by VH-MPC even in the presence of bounded state disturbances [RH03; RH06a].

Interesting research challenges are posed when considering robustness and optimality in real

world applications of VH-MPC. One problem is ensuring robust finite-time completion under changing system dynamics and target sets. Such changes to the original control problem could be caused by faults or changes in operational priorities. When the probable changes are well defined, it is prudent to ensure the existence of contingency trajectories to the new target sets, under the new dynamics and constraints. If it is unknown precisely when the control problem will change, the contingencies need to be robustly available at all times prior to completion of the original control problem.

Another research opportunity arises when considering the application of robust VH-MPC to systems requiring prediction over long time horizons. Given that predictions of system dynamics are required until predicted completion of the control manoeuvre, computational complexity can be an issue for real-time implementation. Techniques like move blocking [Cag+07] can reduce the computational burden; however, naive application of this technique will not provide the required robust completion guarantees. New formulations are required to achieve the aim of reducing complexity and providing the desired theoretical guarantees.

A third research problem presents itself when analysing the effect of robustness modifications on optimality measures of controller performance. In particular, the manner in which constraints are tightened directly affects the size of a VH-MPC controller's Region of Attraction (ROA), which is the set of states for which a feasible initial solution exists to the MPC optimisation problem. In many applications, the size of the ROA is an important measure of control performance, as it determines the ability of the controller to deal with large ranges of initial conditions. Tightening the constraints in such a way as to ensure the largest possible ROA size is desirable.

This thesis develops novel solutions to these three research challenges. It presents a new robust formulation of VH-MPC with feasible contingencies; develops a robust, flexible move blocking scheme that preserves recursive feasibility guarantees; and formulates a new algorithm for constraint tightening that enlarges the controller's ROA. Numerical examples are used to illustrate the efficacy of the proposed solutions. A case study is presented to demonstrate the effect of the techniques on a nontrivial application, namely a model surface excavation machine.

### 1.1 Outline

The main body of this dissertation is structured as follows:

**Chapter 2** provides a comprehensive overview of the theory that underpins the contributions in this thesis. It begins with a qualitative overview of model predictive control and introduces classical formulations for regulation problems. The stability properties of these formulations are also reviewed. Variable horizon completion problems are introduced next, with an emphasis on how these differ from the classical formulations. An overview

of popular robustness modifications is then presented, with a detailed outline of constraint tightening as applicable to VH-MPC completion problems. The chapter concludes with details on implementing the variable horizon with mixed-integer programming.

**Chapter 3** introduces the notion of feasible contingencies in the context of completion problems. The sufficient conditions for the design of a controller with nominal and robust contingency availability are detailed first. Subsequently, a VH-MPC controller is formulated to guarantee contingency availability at all time steps prior to completion. A modified form of constraint tightening is used to achieve the required robustness to bounded disturbances during normal operation and under contingency conditions. The controller formulation is shown to guarantee robust recursive feasibility, robust finite-time completion and robust contingency availability. Details are also provided for implementation of the resulting quasi-convex optimisation problems using mixed-integer programming.

Various contingency scenarios are then presented, including multiple, costed and state-dependent contingencies. In each of these scenarios, sufficient conditions are derived for preserving the guarantee of robust finite-time completion. Simplifications to reduce the computational burden of the optimisation problem are also outlined. The chapter concludes with numerical examples on two systems that demonstrate the power of the controller formulation and contingency scenarios.

**Chapter 4** presents a technique for reducing the computational complexity of VH-MPC, whilst still maintaining robust completion guarantees. A new move-blocked formulation of VH-MPC is developed, which allows the number of input decision variables in the optimisation problem to be reduced. The variable horizon is achieved by solving multiple optimisation problems at each time step corresponding to different horizon lengths and move blocking patterns. A novel constraint tightening methodology is developed to guarantee robustness to bounded disturbances in the presence of blocking constraints. Implementational techniques to ensure reductions in computation time with the new methodology are also outlined.

An example blocking scheme is then proposed, which allows fine-grained control action during the start and end of a control manoeuvre, but applies move blocking in the middle to reduce complexity. This kind of scheme is suited to vehicle control problems, in particular, spacecraft rendezvous, where robust constraint satisfaction is potentially required over long prediction horizons. A simple numerical example is used to illustrate the effect on computation time and closed loop cost under disturbances of applying the scheme with varying degrees of complexity reduction.

**Chapter 5** presents a way of tightening constraints to improve the region of attraction of a VH-MPC controller. It shows how the constraint tightening policy, parameterised in terms of direct disturbance feedback, can be formulated to maximise the volume of an inner approximation to the controller's true ROA. Through numerical examples, the technique is shown to produce vastly improved ROA volumes over other proposed techniques

for designing constraint tightening policies. It is also shown how the improvement in ROA volume can drastically decrease the maximum horizon length required to achieve an initial feasible solution. Simplifications for higher state dimensions and prediction horizons are also discussed.

**Chapter 6** illustrates the application of VH-MPC and the techniques developed in the previous chapters to a non-trivial system, namely a surface excavation machine. A simplified non-linear kinematic and dynamic model is developed for a two-degree-of-freedom machine used widely in surface mining operations. A material model is also developed to simulate the effect of excavation forces. The machine model is then pre-stabilised with an inner loop controller, following which a discretised and linearised model of the full machine and material system is obtained for the application of VH-MPC to trajectory generation. By appropriately designing the cost function and terminal constraints, realistic digging trajectories are produced. Further simulations show the effect of including contingency availability and applying move blocking to the linearised model. The results demonstrate the applicability of VH-MPC and the theory developed in this thesis to problems other than vehicle control.

**Chapter 7** summarises the conclusions and contributions of the thesis. Future research directions are also proposed.

**Appendix A** provides some derivations and additional plots pertaining to Chapter 5.

**Appendix B** lists the equations of motion and controller parameters for the model developed in Chapter 6.

## 1.2 Notational Conventions

Whilst the chapters in this thesis are all related to optimality and robustness in VH-MPC, they each require different mathematical tools. Hence, adopting a completely coherent system of notation is impractical. Within each chapter, coherency is preserved; however, some symbols take on different roles in subsequent chapters. Common nomenclature has been listed at the start of this thesis. In addition, some of the listed symbols are given a more detailed introduction at the start of each chapter where appropriate.

## Chapter 2

# Background

This chapter provides an overview of background theory that underlies all of the contributions in this thesis. To fulfil this aim, it presents a comprehensive review of relevant background literature. A brief introduction to Model Predictive Control is provided first, following which, standard formulations of MPC for regulation problems are described and their stability properties analysed. The notions of a variable horizon and completion are then introduced as an extension of these formulations.

Robust model predictive control is discussed next, where a brief review is presented of the most popular techniques for ensuring a priori robustness to bounded disturbances. Emphasis is given to the constraint tightening approach, which is used extensively in this thesis. In particular, the formulation of constraint tightening in conjunction with a variable horizon controller is presented, as applicable to completion problems.

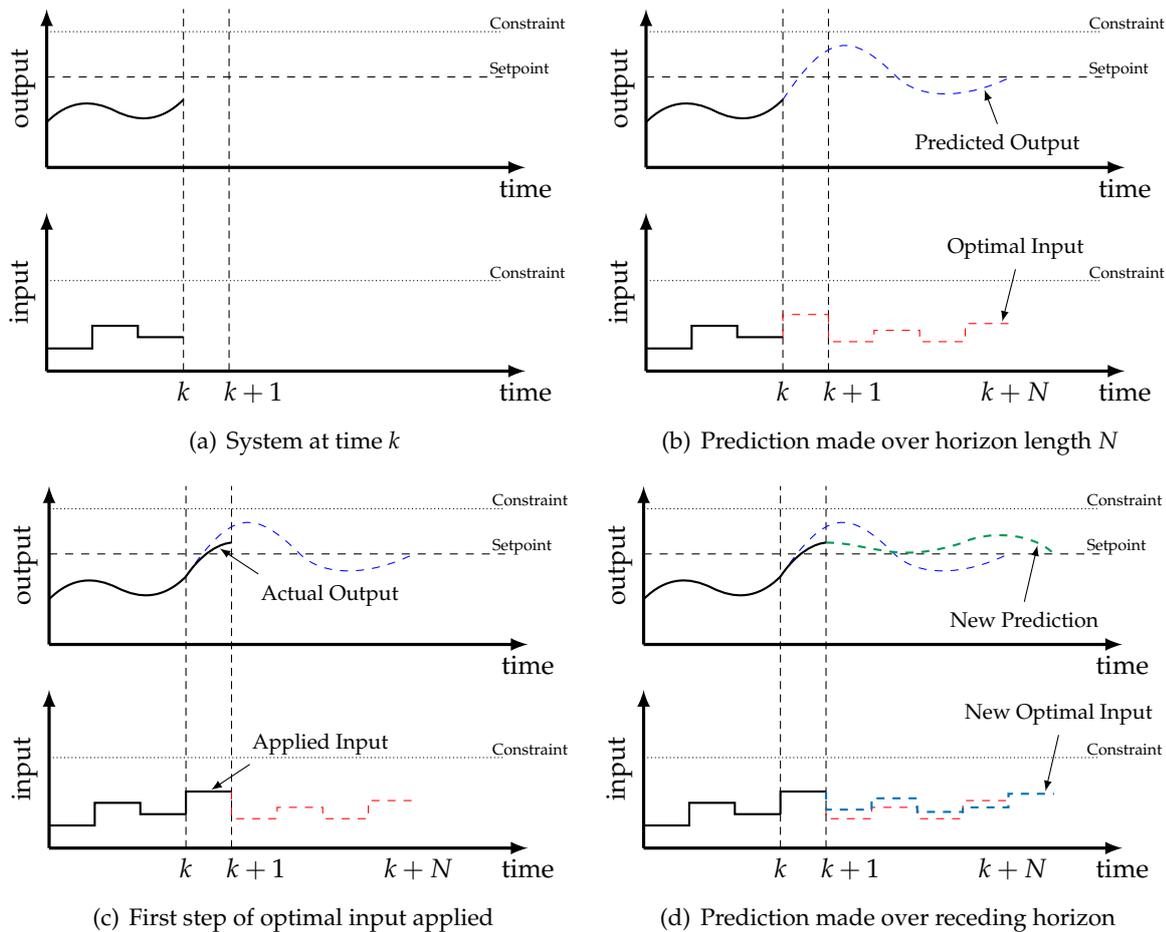
### 2.1 Overview of Model Predictive Control

This section provides an introduction to Model Predictive Control. A general overview of its principle of operation is presented first. As part of this overview, the *receding horizon* optimisation concept is introduced, which is at the heart of MPC. A qualitative description of MPC's constraint handling ability is also given.

#### 2.1.1 Principle of Operation

The basic mechanism of MPC is simple and intuitive. In most common formulations, it is a discrete-time control methodology, so it only generates control actions at discrete time intervals. Furthermore, it requires an explicit system model, capable of predicting behaviour over a number of time steps into the future. This time period is called the *prediction horizon*. An MPC controller seeks to find the optimal control input sequence over the prediction horizon that minimises some desired cost function, using the system model to make predictions about future behaviour. The salient characteristic of its operation is that it applies the first optimal

input to the system and repeats the process again over the prediction horizon, which is now shifted forward by a time step. This is the reason that MPC is also known as receding horizon control. As explained in the survey paper by Mayne et al. [May+00], the essential difference from standard optimal control is that in general, an optimisation problem is solved at each time step. This is advantageous, as constraints can be included in the formulation by simply solving numerically a constrained optimisation problem.



**Figure 2.1:** The receding horizon principle (adapted from [Cam09])

### 2.1.2 The Receding Horizon Principle

Figure 2.1 illustrates the idea of the receding horizon for a SISO system. The system input and response up to step  $k$  is illustrated in Plot (a). The MPC controller then calculates the input over the horizon of  $N$  steps required to optimise a given cost function, as shown in Plot (b). Sometimes, the input will be assumed to vary only over the first few steps of the horizon (known as the *control horizon*), and then be assumed constant, which can reduce the number of decision variables over which optimisation is required. Sample operating constraints are indicated on the figures.

After calculating the optimal input sequence, the predictive controller applies the first element

of this sequence, as shown in Plot (c). In general, the resulting output will be different to the predicted output, due to disturbances and modelling errors. The notion of a receding horizon is pictured in Plot (d), where the optimisation process is repeated again to calculate a new input signal to  $k + N + 1$  steps. The horizon length, being constant, recedes by one time step after each subsequent control action.

### 2.1.3 Constraint Handling

Traditional control methods take ad hoc approaches towards handling constraints. Often, this involves choosing operating points far enough away from constraint boundaries to decrease the likelihood of violating them. The drawback of such a strategy is that these operating points may not be optimal. Figure 2.2 illustrates this in a qualitative manner, assuming a linear system and Gaussian-distributed disturbances.

Graph (a) shows the distribution of outputs for a poorly tuned linear controller. Note that the mean of the distribution must be kept significantly far from the constraint boundary to have an acceptably low probability of violation. Graph (b) shows the distribution of outputs that may be achieved by using an optimal linear controller. The variance is vastly improved, however the symmetric nature of the distribution still limits the proximity of the operating point to the constraint boundary. A possible distribution of outputs from an MPC controller is shown in Graph (c). Note the salient feature of this distribution: it is asymmetric, allowing the operating point to move closer to the constraint boundary and thereby improving efficiency of operation, whilst still retaining an acceptably low probability of constraint violation. In effect, the controller is taking different actions when a disturbance causes the output to approach the constraint boundary. The resulting control law, however, is nonlinear. Hence, MPC allows superior handling of constraints at the expense of a nonlinear control law.

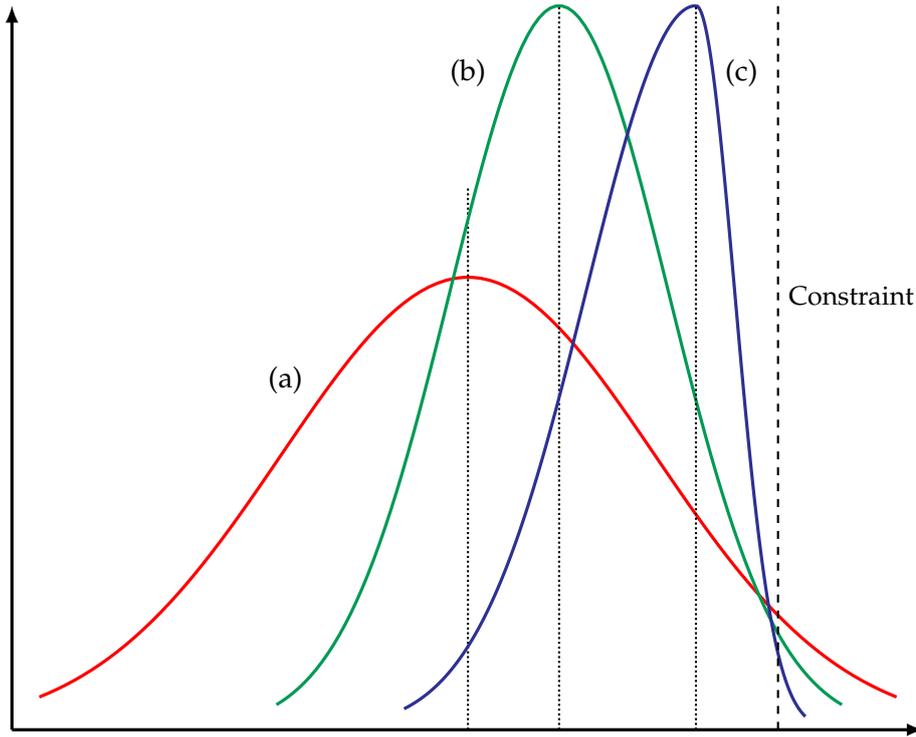
## 2.2 Mathematical Formulation of MPC

This section presents the classical state space formulation of MPC for regulation to the origin, given a general  $m$ -input,  $n$ -state and  $p$ -output time-invariant system. The derivations that follow are based on [May+00]. It will be assumed that the system can be approximated by a system of explicit difference equations

$$x(k+1) = f(x(k), u(k)) \quad (2.1)$$

$$y(k) = g(x(k)), \quad (2.2)$$

where  $f(\cdot, \cdot)$  will be taken to have an equilibrium point at the origin. The vector of states is denoted  $x$ , the input vector is written as  $u$  and the function  $f(\cdot, \cdot)$  defines the successor state in terms of the current state and input. The function  $g(\cdot)$  represents the system output as a function of the states alone, assuming there is no input feedforward. Constraints on the input



**Figure 2.2:** Qualitative illustration of MPC's constraint awareness (adapted from [Mac02])

and state vectors are assumed to be of the form

$$u(k) \in \mathcal{U} \subset \mathbb{R}^m \quad (2.3)$$

$$x(k) \in \mathcal{X} \subset \mathbb{R}^n, \quad (2.4)$$

where  $\mathcal{U}$  is usually a convex, compact subset of  $\mathbb{R}^m$  and  $\mathcal{X}$  is usually a convex, closed subset of  $\mathbb{R}^n$ , where both sets include the origin.

**Remark 2.1.** In the derivations that follow and indeed in the remainder of this thesis, full state information is assumed to be available.

The aim of designing a controller is usually to drive the system either to the origin or to an equilibrium state  $x_r$  for which  $g(x_r) = r$ , where  $r$  is the desired output value at equilibrium. It will be assumed without loss of generality that  $x_r = 0$ , as an affine transformation can be used to shift the equilibrium state to the origin for a general system. Then, define a *prediction model* with dynamics

$$x(k|k) = x(k) \quad (2.5)$$

$$x(k+j+1|k) = f(x(k+j|k), u(k+j|k)) \quad (2.6)$$

$$y(k+j|k) = g(x(k+j|k)), \quad (2.7)$$

subject to the constraints

$$u(k+j|k) \in \mathcal{U} \quad (2.8)$$

$$x(k + j|k) \in \mathcal{X} \quad (2.9)$$

$$x(k + N|k) \in \mathcal{T}, \quad (2.10)$$

for all  $j \in \mathbb{Z}_{[0, N-1]}$ . The notation  $x(k + j|k)$  denotes a prediction  $j$  steps into the future of the state  $x$  from the current time  $k$ . Constraint (2.10) requires some terminal constraint set  $\mathcal{T}$  to be entered at the end of the prediction horizon of length  $N$ .

A finite horizon cost function for the prediction model is then typically specified in the form

$$J_N(x(k), \mathbf{u}(k)) = \sum_{j=0}^{N-1} \ell(u(k + j|k), x(k + j|k)) + F(x(k + N|k)), \quad (2.11)$$

where

$$\mathbf{u}(k) = \{u(k|k), u(k + 1|k), \dots, u(k + N - 1|k)\} = \{u(k + j|k)\}_{j=0}^{N-1} \quad (2.12)$$

is the sequence of inputs applied over the prediction horizon,  $x(k)$  is the current state,  $\ell(\cdot, \cdot)$  is the stage cost and  $F(\cdot)$  is some terminal cost function.

**Remark 2.2.** For technical reasons, it is assumed that the stage cost is strictly greater than zero everywhere except for the origin, where it has the value zero, and decreases with the proximity of the state to the origin. Mathematically, this requires the existence of a  $\mathcal{K}_\infty$ -function  $c(\cdot)$  such that

$$\ell(x, u) \geq c(\|(x, u)\|). \quad (2.13)$$

The model predictive control problem can now be formally defined in terms of the cost function. For the given state of the system at time step  $k$ , the aim is to minimise  $J_N(\cdot, \cdot)$  whilst satisfying all constraints on the inputs, intermediate states and terminal state. Given that  $\ell(\cdot, \cdot)$  and  $F(\cdot)$  are time invariant, the optimal cost or value function depends only on  $x(k)$ . This value function is given by

$$\begin{aligned} J_N^*(x(k)) &= \min_{\mathbf{u}(k)} J_N(x(k), \mathbf{u}(k)) \\ &= \min_{\mathbf{u}(k)} \sum_{j=0}^{N-1} \ell(x(k + j|k), u(k + j|k)) + F(x(k + N|k)), \end{aligned} \quad (2.14)$$

with the minimisation subject to the constraints (2.8) - (2.10). The corresponding input sequence that minimises the value function is given by the expression

$$\begin{aligned} \mathbf{u}^*(k) &= \arg \min_{\mathbf{u}(k)} J_N(x(k), \mathbf{u}(k)) \\ &= \{u^*(k + j|k)\}_{j=0}^{N-1}. \end{aligned} \quad (2.15)$$

Note the state prediction at time  $k + j$  resulting from applying  $\mathbf{u}^*(k)$  is denoted  $x^*(k + j|k)$ . Only the first optimal input from (2.15) is applied to the system, resulting in the time-invariant model predictive control law

$$u(k) = \kappa_N(x(k)), \quad (2.16)$$

where  $\kappa_N(\cdot)$  is defined by

$$\kappa_N(x(k)) = u^*(k|k). \quad (2.17)$$

## 2.3 Analysis of Stability

Having formally defined the action of predictive control on a general dynamical system, its stability properties will be analysed. As mentioned in Section 2.1.3, a nonlinear control law results from MPC with active constraints. This means that finding necessary conditions for stability is, in general, difficult. Instead, the majority of literature focuses on finding sufficient conditions for stability through Lyapunov methods. In particular, Mayne et al. [May+00] outlines the sufficient conditions on the terminal cost  $F(\cdot)$ , the terminal constraint set  $\mathcal{T}$  and a candidate terminal state feedback control law  $\kappa_f(x(k))$  that guarantee stability. These sufficient conditions can be derived by utilising the value function  $J_N^*(\cdot)$  as a Control Lyapunov Function (CLF), and placing an upper bound on this function at the following time step (the so-called “direct” method). In the derivations that follow, the value function will be assumed continuous. Following the derivations, techniques for satisfying the sufficient conditions for stability will be discussed.

### 2.3.1 Sufficient Conditions from Lyapunov Theory

This method of deriving stability conditions seeks to find an upper bound on the value function at the successor state, namely  $J_N^*(x(k+1)) = J_N^*(f(x(k), \kappa_N(x(k))))$ . The upper bound is calculated by defining a feasible control sequence applied at time  $k+1$  of the form

$$\hat{u}(k+1) = \{u^*(k+1|k), u^*(k+2|k), \dots, u^*(k+N-1|k), \hat{u}(k+N|k+1)\}, \quad (2.18)$$

which is simply the optimal input sequence (2.15) shifted, with an additional input term as the new tail. To ensure that the sequence  $\hat{u}(\cdot)$  is feasible, the tail can be defined by some terminal control law

$$\hat{u}(k+N|k+1) = \kappa_f(x^*(k+N|k)),$$

where  $\kappa_f(\cdot)$  is both input and state admissible. This will be the case provided the terminal constraint set satisfies

$$\mathcal{T} \subseteq \mathcal{X}$$

and the terminal control law satisfies, for all  $x \in \mathcal{X}$ ,

$$\begin{aligned} \kappa_f(x) &\in \mathcal{U} \\ f(x, \kappa_f(x)) &\in \mathcal{T}. \end{aligned}$$

The resulting state trajectory from applying this sequence is then

$$\hat{x}(k+1) = \{x(k+1), x^*(k+2|k), \dots, x^*(k+N|k), \hat{x}(k+N+1|k+1)\},$$

where perfect model matching is assumed, so  $x(k+1) = x^*(k+1|k)$  under the action of MPC. Then, the new terminal state prediction is given by

$$\hat{x}(k+N+1|k+1) = f(x^*(k+N|k), \kappa_f(x^*(k+N|k))). \quad (2.19)$$

The cost at time  $k+1$  associated with applying the input sequence can be found by subtracting the old zeroth-stage and terminal costs from  $J_N^*(\cdot)$  and then adding the new  $(N-1)$ th stage and terminal costs, giving

$$\begin{aligned} J_N(x(k+1), \hat{u}(k+1)) &= J_N^*(x(k)) - \ell(x(k), \kappa_N(x(k))) - F(x^*(k+N|k)) \\ &\quad + \ell(x^*(k+N|k), \kappa_f(x^*(k+N|k))) + F(\hat{x}(k+N+1|k+1)). \end{aligned} \quad (2.20)$$

As the input sequence  $\hat{u}(\cdot)$  is feasible, but not necessarily optimal, this cost is an upper bound for the value function  $J_N^*(x(k+1))$ . By requiring the sum of the last three terms of (2.20) to be less than or equal to zero, closed-loop stability can be ensured. This requires that

$$F(\hat{x}(k+N+1|k+1)) - F(x^*(k+N|k)) + \ell(x^*(k+N|k), \kappa_f(x^*(k+N|k))) \leq 0. \quad (2.21)$$

Observe that (2.21), as well as the technical requirements on  $\ell(\cdot, \cdot)$  imply that

$$\begin{aligned} J_N^*(x(k+1)) - J_N^*(x(k)) + \ell(x(k), \kappa_N(x(k))) &\leq J_N(x(k+1), \hat{u}(k+1)) - J_N^*(x(k)) \\ &\quad + \ell(x(k), \kappa_N(x(k))) \\ &\leq 0. \end{aligned} \quad (2.22)$$

Hence, the value function is a CLF if (2.21) is satisfied. All of these requirements are summarised in Proposition 2.3 [May+00].

**Proposition 2.3.** *If the conditions C1-C4 are satisfied by the terminal set, terminal cost function and terminal controller, where*

C1:  $\mathcal{T} \subset \mathcal{X}$ ,  $\mathcal{T}$  is closed and  $0 \in \mathcal{T}$  ( $\mathcal{T}$  is state admissible)

C2:  $\kappa_f(x) \in \mathcal{U}, \forall x \in \mathcal{T}$  (the terminal control law is input admissible for all terminal states)

C3:  $x^+ \triangleq f(x, \kappa_f(x)) \in \mathcal{T}, \forall x \in \mathcal{T}$  ( $\mathcal{T}$  is positively invariant under  $\kappa_f(\cdot)$ )

C4:  $F(x^+) - F(x) + \ell(x, \kappa_f(x)) \leq 0, \forall x \in \mathcal{T}$  ( $F(\cdot)$  is a local CLF),

*asymptotic stability of the origin is ensured.*

**Remark 2.4.** Mathematically, conditions C1-C4 ensure that for all  $x \in \mathcal{X}$  for which an initial feasible solution to the  $N$ -step optimisation problem exists, as  $k \rightarrow \infty$ ,  $x \rightarrow 0$  in the closed loop.

The fact that given a feasible solution at time  $k$ , another one exists at time  $k + 1$  is known as the *recursive feasibility* property.

Mayne's survey paper [May+00] describes the evolution of stabilising modifications to the finite horizon problem to satisfy C1-C4. The addition of a terminal equality constraint, terminal cost function, terminal constraint sets or combinations of these have all been shown to theoretically guarantee stability.

One of the simplest methods is the use of a terminal equality constraint, as outlined by Keerthi and Gilbert [KG88]. This essentially requires the system to be at the equilibrium state by the end of the prediction horizon (i.e.  $x(k + N|k) = 0$ ). In terms of the conditions C1-C4, the terminal set contains only one element:  $\mathcal{T} = \{0\}$ . As  $0 \in \mathcal{X}$ , C1 is satisfied. The terminal controller is not required to take any action, since the predicted state trajectory reaches equilibrium in finite time, so  $\kappa_f(x(k)) = 0$ . Hence, C2 is satisfied, given  $\kappa_f(0) = 0 \in \mathcal{U}$ . The third condition C3 is then also satisfied, given that the equilibrium point at the origin is a positively invariant set by definition i.e.  $f(0, 0) = 0 \in \mathcal{T}$ . No terminal cost is used in this case, hence  $F(\cdot) = 0$ . In addition, the technical conditions on  $\ell(\cdot, \cdot)$  guarantee that  $\ell(0, 0) = 0$ . These ensure that C4 is satisfied:  $F(f(0, \kappa_f(0))) - F(\kappa_f(0)) - \ell(0, \kappa_f(0)) = 0$ . It is clear, therefore, that a terminal equality constraint guarantees stability.

A more general method for guaranteeing stability is known as *dual mode* MPC [CLM96]. In this paradigm, the controller imposes a terminal set constraint  $\mathcal{T}$  satisfying C1 which is invariant with respect to the terminal control law  $\kappa_f(x)$ . This ensures satisfaction of C2-C3. The terminal cost is then chosen to satisfy C4. Once the actual state of the system enters  $\mathcal{T}$ , the controller is switched to  $\kappa_f(\cdot)$ , which stabilises the system at the origin. Other exotic variants such as *triple mode* MPC have also been proposed [RKC00].

### 2.3.2 Linear Quadratic MPC

The linear unconstrained case has a particularly nice form. Given the linear system

$$f(x, u) = Ax + Bu, \tag{2.23}$$

the stage cost is usually defined by the quadratic form

$$\ell(x, u) = x^T Qx + u^T Ru, \tag{2.24}$$

where  $Q \succeq 0$  and  $R \succ 0$ . Since the system is unconstrained, the conditions C1-C3 are trivially satisfied. A constant gain terminal controller of the form  $\kappa_f(x) = K_f x$  is again chosen, where the gain is selected such that the eigenvalues of  $A + BK_f$  lie within the unit circle. This requires that the pair  $(A, B)$  is stabilisable. Then, the terminal cost is chosen to have the form

$$F(x) = x^T Px, \tag{2.25}$$

where  $P$  is the positive definite solution to the Discrete Lyapunov Equation (DLE)

$$(A + BK_f)^T P (A + BK_f) - P + Q + K_f^T R K_f = 0. \quad (2.26)$$

Choosing the terminal cost in this way ensures that C4 is satisfied, making the origin asymptotically stable with region of attraction  $\mathbb{R}^n$ . If the terminal controller is chosen to be

$$K_f = -(B^T P B + R)^{-1} B^T P A, \quad (2.27)$$

then (2.26) becomes a Discrete Algebraic Riccati Equation (DARE), with a unique solution for  $P$ , given the conditions on  $Q$  and  $R$ . The resulting controller then produces the same action as an infinite horizon LQR controller.

In the linear constrained case, it is explained in [May+00] that for a stable system with no state constraints and convex, compact input constraints, there is no requirement for a terminal constraint set, so the reasoning presented above can be used again to prove asymptotic convergence. However, when state and input constraints are enforced, or the system is unstable, a terminal cost is not sufficient to ensure stability.

To ensure asymptotic convergence under constraints, a dual mode scheme can be used. Define the terminal constraint set to satisfy C1 and be positively invariant under the linear terminal control law  $K_f$ . This requires the terminal set to satisfy

$$(A + BK_f)x \in \mathcal{T} \quad (2.28)$$

$$K_f x \in \mathcal{U}, \quad (2.29)$$

for all  $x \in \mathcal{T}$ . Then, C1 - C3 are satisfied. By choosing  $P$  to satisfy (2.26), C4 is satisfied, ensuring asymptotic stability of the origin. The terminal controller can be chosen to be the solution of the infinite horizon unconstrained problem, as in (2.27), which will ensure that the behaviour of the system once inside the terminal set will be equivalent to the infinite horizon (unconstrained) optimal solution. With  $K_f$  chosen in this way, a suitable positively-invariant  $\mathcal{T}$  is given by a constraint-admissible level set of  $x^T P x$ .

## 2.4 Variable Horizon MPC

Having defined fixed horizon predictive control problems and analysed their stability properties, the notion of a variable horizon is now introduced. In this variety of MPC, the value function takes the form

$$J^*(x(k)) = \min_{u^{(k), N(k)}} \sum_{j=0}^{N(k)-1} \ell(x(k+j|k), u(k+j|k)), \quad (2.30)$$

subject to the pointwise-in-time constraints

$$u(k+j|k) \in \mathcal{U} \quad (2.31)$$

$$x(k + j|k) \in \mathcal{X} \tag{2.32}$$

$$x(k + N(k)|k) \in \mathcal{T}, \tag{2.33}$$

for all  $j \in \mathbb{Z}_{[0, N(k)-1]}$ , where the horizon length  $N(k)$  is a decision variable in the constrained optimisation problem solved at each  $k$ . Essentially,  $N(k)$  denotes the *first* time step at which the set  $\mathcal{T}$  is entered. The corresponding optimal horizon length is denoted  $N^*(k)$ . No terminal cost is applied in this case.

One of the earliest mentions of a varying prediction horizon is in the paper by Michalska and Mayne [MM93], where a dual mode scheme is used for control of nonlinear systems. The observation is made that recursive feasibility is easier to achieve with a variable horizon. In (2.18), a tail term has to be added to the shifted input sequence to ensure recursive feasibility, as the horizon length must stay the same. However, allowing the prediction horizon to vary means that the input sequence

$$\hat{u}(k + 1) = \{u^*(k + 1|k), u^*(k + 2|k), \dots, u^*(k + N^*(k) - 1|k)\} \tag{2.34}$$

provides a feasible solution to the optimisation problem solved at time  $k + 1$ , with a horizon length  $N^*(k) - 1$ .

### 2.4.1 Completion

Inherent in the definition of the variable horizon is the idea of completion, which denotes that the state of the system is driven into the terminal set  $\mathcal{T}$  in  $N(k)$  steps. This can form the first part of a dual mode scheme, where the control action will be switched once the target set is entered to stabilise at an equilibrium point. It can be shown that with an initially feasible solution and the stage cost defined to satisfy (2.13), completion is achieved in finite time. The technical requirement (2.13) ensures that for all  $j < N^*(k)$ , there exists some constant  $c \in \mathbb{R}^+$  such that

$$\ell(x(k + j|k), u(k + j|k)) \geq c, \tag{2.35}$$

as the state prediction  $x(k + j|k)$  is outside the terminal set. It follows that, using the shifted feasible input at time  $k + 1$  with horizon length  $N^*(k) - 1$ ,

$$\begin{aligned} J^*(x(k + 1)) - J^*(x(k)) &\leq \hat{J}(x(k + 1)) - J^*(x(k)) \\ &= -\ell(x(k), u^*(k|k)) \\ &\leq -c. \end{aligned} \tag{2.36}$$

This means that the value function at each time step decreases by at least  $c > 0$ . As the value function is non-negative by construction, this guarantees that the set  $\mathcal{T}$  must be entered in at most  $\lfloor J^*(x(k))/c \rfloor$  steps, from current state  $x(k)$ . Note that this guarantee does not require the satisfaction of C1-C4, which pertain to stability.

**Remark 2.5.** The value function in this case is discontinuous, as for any state within the

terminal state, the horizon length, and therefore the value function, is zero. For this reason,  $J^*(\cdot)$  will be referred to as a *Lyapunov-like* function when the variable horizon is used.

### 2.4.2 Cost Functions

For a stage cost chosen to satisfy (2.13), if an initial feasible solution exists, completion is achieved in finite time. However, this stage cost has been designed with the objective of stabilising at the origin. In some applications, there is no requirement for stability; enforcing the invariance requirement on  $\mathcal{T}$  (C3) can be overly restrictive for such applications [RH03; RH06a]. In vehicle manoeuvring problems, the control objective is often to simply achieve completion in some *target set*  $\mathcal{T}$ , with the control problem ending after this occurs. This thesis is concerned mainly with control problems of this form.

The previous subsection showed that the key requirement for finite-time completion is a positive lower bound on the stage cost outside  $\mathcal{T}$ . Hence, different kinds of cost functions can be defined, which still guarantee finite-time completion, but have no requirement on  $\mathcal{T}$  containing the origin. Since the horizon length is a variable in the optimisation problem, a simple minimum-time value function [SM98] of the form

$$J^*(x(k)) = \min_{N(k)} N(k), \quad (2.37)$$

can be defined, which will optimise the time to completion. Of course, this may result in aggressive control action, so the cost function can be modified to penalise a weighted sum of the control inputs over the prediction and the time to completion. The value function then takes the form

$$J^*(x(k)) = \min_{u(k), N(k)} \sum_{j=0}^{N(k)-1} 1 + \gamma \|u(k+j|k)\|_p, \quad (2.38)$$

for some  $p$ -norm of the inputs weighted by  $\gamma > 0$ . In [RH03; RH06a], a 1-norm is chosen, as the application in question is spacecraft rendezvous, where this quantity provides a measure of fuel use during a control manoeuvre. Note that no terminal cost has been included, as the control problem ends at completion.

**Remark 2.6.** From here on, the  $p$  subscript on the norm will be dropped unless a particular norm is implied.

It is easy to see that when the time to entry into  $\mathcal{T}$  is included in the cost function, finite-time completion is guaranteed. From the feasible input sequence at time  $k+1$  (2.34), the cost function is guaranteed to reduce by at least unity, which guarantees finite-time entry into  $\mathcal{T}$ , given that the cost function is non-zero by construction. Another important observation is that these new cost functions make it possible to define the completion problem directly in terms of the original target set, without requiring a change of coordinates to have the origin contained within  $\mathcal{T}$ . The benefits of being able to do this will become apparent in Chapter 3, where multiple target sets are defined within the same MPC problem.

**Remark 2.7.** Note that the new cost functions do not prevent VH-MPC being used for regulation problems. By appropriately choosing  $\mathcal{T}$  as a control invariant set, a dual mode scheme can still be used to guarantee asymptotic convergence to a given point within the terminal set.

### 2.4.3 An Alternative Formulation

The dual mode schemes detailed in [MM93] and [SMR99] also use a variable horizon, albeit formulated in a slightly different way. A fixed horizon length is used for the optimisation; however, the schemes multiply the original stage cost by the indicator function  $\alpha(\cdot)$  for the state being outside the terminal set, giving the new stage cost

$$\bar{\ell}(x, u) = \alpha(x)\ell(x, u), \quad (2.39)$$

where

$$\alpha(x) = \begin{cases} 0 & x \in \mathcal{T} \\ 1 & \text{otherwise.} \end{cases} \quad (2.40)$$

This ensures that for a regulation problem, C4 is satisfied and the system is stabilised. Essentially, the trajectory is only costed whilst the state is outside the terminal set, so this is equivalent to applying costs over a variable horizon up to the completion time. The main difference is that constraints are still applied after this time, so if a completion problem is considered with no subsequent stabilisation, the closed loop behaviour may differ from a controller that explicitly optimises over the horizon length.

## 2.5 Robust MPC

The preceding sections have presented a formulation of MPC where no a priori anticipation of disturbances is incorporated. Hence, simply using a shifted, truncated input sequence will not guarantee recursive feasibility under the action of disturbances. This is due to the fact that at the next prediction step, the system state will not necessarily be at its one-step prediction from the current time. To address this, many forms of Robust Model Predictive Control (RMPC) have been proposed. This section summarises the main contributions, for an additive state disturbance  $w(k) \in \mathcal{W} \subset \mathbb{R}^n$ , where the disturbance set  $\mathcal{W}$  is compact and contains the origin.

### 2.5.1 Min-Max Predictive Control

Min-max robustness modifications centre around the idea of minimising the cost that would be incurred by the worst case disturbance realisation over the prediction horizon. By finding an input sequence that achieves this objective and results in a constraint-admissible trajectory for all disturbance realisations, robust feasibility is ensured by construction. This so-called open-loop approach can be excessively conservative, hence a modification that optimises over

state feedback policies instead is outlined in [SM98], and subsequently extended in [KM04]. Essentially, predicted trajectory sequences are formulated for every disturbance realisation, which is made tractable if  $\mathcal{W}$  is a polytope. The variable horizon case is considered at the end of [SM98]. An alternative approach using Linear Matrix Inequalities (LMIs) is described by [KBM96], where the pair  $(A, B)$  is taken to lie within a known polytope or an ellipsoid. An upper bound on the cost function is then minimised over state feedback policies.

### 2.5.2 Affine Feedback Policies

Optimisation over affine feedback policies is described in [GKM06]. This approach differs from the min-max strategies, as it includes an extra affine term in the control policy at each prediction step. The set of affine state feedback policies is non-convex; however, it is shown that the set of affine disturbance feedback policies is equivalent, which results in a convex optimisation problem. These policies are shown to guarantee robust constraint satisfaction for bounded disturbances when applied in a receding horizon manner.

### 2.5.3 Constraint Tightening

This robustness modification for linear time-invariant systems was first elucidated by [GKR97] and subsequently developed by [CRZ01] and [RH03; Ric05b; RH06a]. A formulation for LTV systems is described in [Ric05a], and a disturbance feedback parameterisation is introduced by [KRH07]. In all cases, the essence of Constraint Tightening (CT) is that constraint sets at future steps into the prediction horizon are shrunk, to allow enough control authority to correct for the action of an unknown but bounded state disturbance. This can be seen as a partitioning of the control authority; one part is used to steer the system state into  $\mathcal{T}$  and the remainder is held in reserve to mitigate the effect of the error between the predicted and actual state evolution at each step. By designing  $\mathcal{T}$  as a robust control invariant set, robust asymptotic convergence can be guaranteed [RH06b]. For completion problems, robust finite-time completion can be guaranteed for an arbitrary choice of closed terminal set [RH06a]. A detailed overview of constraint tightening for linear systems will be presented in the following section.

**Remark 2.8.** The disturbance feedback formulation of CT is identical to applying the affine feedback formulation of [GKM06] with time invariant disturbance feedback gains.

### 2.5.4 Tube Model Predictive Control

This form of robustness modification is closely related to constraint tightening. The basic formulation, known as disturbance invariant tube MPC, is outlined in [ML01]. It involves partitioning each term of the input sequence into a feedforward term and a feedback term. Specifically, the input to the system at time  $k$  takes the form

$$u(k) = u^*(k|k) + K(x(k) - x^*(k|k)), \quad (2.41)$$

where the constraint  $x^*(k|k) = x(k)$  specified in (2.5) is relaxed, and  $K$  is a stabilising controller for  $(A, B)$ . The relaxation means that the predicted states trajectory now defines a “tube centre”, which does not necessarily need to originate at the current state. This nominal predicted trajectory is constrained to reach the terminal set in  $N$  steps under the predicted input sequence  $\mathbf{u}^*(k)$ . The additional feedback term regulates the true state of the system to the predicted tube centre.

To ensure that  $u(k)$  is feasible, the constraints applied to the nominal trajectory need to be tightened. Generalisations to the basic formulation can be found in [Lan+04], where the variable horizon case is addressed. Tube MPC has also been applied to nonlinear systems, as described in [May+11]. In the case of linear systems, CT with a fixed state feedback gain is in general, less conservative than tube MPC [Tro09].

**Remark 2.9.** Observe that the former two robustness modifications presented in this section both increase online computational complexity, whereas the latter two do not.

## 2.6 Robust VH-MPC with Constraint Tightening

In this section, an  $n$ -state,  $m$ -input linear system of the form

$$x(k+1) = Ax(k) + Bu(k) + w(k) \quad (2.42)$$

will be assumed, where

$$w(k) \in \mathcal{W} \quad (2.43)$$

is a state disturbance, known to lie within the compact set  $\mathcal{W}$ , which contains the origin. Constraints are now generalised to be in terms of an output vector

$$y(k) = Cx(k) + Du(k), \quad (2.44)$$

where  $C$  and  $D$  are a design choice. Hence, the system is subject to constraints of the form

$$y(k) \in \mathcal{Y}, \quad (2.45)$$

where  $\mathcal{Y}$  is closed. In the remainder of this thesis, whenever a linear system is introduced with outputs  $y(\cdot)$ , it is assumed that these have been designed for the application of constraints. They do not represent measurements, as full state information is assumed.

**Remark 2.10.** Individual state and input constraints can be easily cast into output constraint form. To see this, given the separate state and input constraint sets defined in (2.3) - (2.4), it is clear that

$$(x, u) \in \mathcal{X} \times \mathcal{U}. \quad (2.46)$$

By choosing

$$C = \begin{bmatrix} I_n & 0_{m \times n} \end{bmatrix}^T \quad (2.47)$$

$$D = \begin{bmatrix} 0_{n \times m} & I_m \end{bmatrix}^T \quad (2.48)$$

$$\mathcal{Y} = \mathcal{X} \times \mathcal{U}, \quad (2.49)$$

the constraint (2.46) can be written in the form (2.45). The output constrained form is more general, as it permits cross-constraints between states and inputs, which could not be achieved with individual state and input constraint sets.

### 2.6.1 Formulation

Using a similar notation to [RH06a], define the matrices

$$L(0) = I_n \quad (2.50a)$$

$$L(j+1) = (A + BK(j))L(j) \quad (2.50b)$$

$$Q(j) = (C + DK(j))L(j) \quad (2.50c)$$

and the corresponding sets

$$\mathcal{Y}(0) = \mathcal{Y} \quad (2.51a)$$

$$\mathcal{T}(0) = \mathcal{T} \quad (2.51b)$$

$$\mathcal{Y}(j+1) = \mathcal{Y}(j) \ominus Q(j)\mathcal{W} \quad (2.51c)$$

$$\mathcal{T}(j+1) = \mathcal{T}(j) \ominus L(j)\mathcal{W}. \quad (2.51d)$$

The sequence of matrices

$$\{K(j) \in \mathbb{R}^{n \times m}\}_{j=0}^{\infty}, \quad (2.52)$$

is known as the state feedback constraint tightening *policy*. In [RH06a], nilpotent policies are suggested, as they minimise the number of tightened sets to be calculated. For a degree- $h$  nilpotent policy,  $L(h) = 0$ , so there is no additional tightening applied for  $j > h$ .

The prediction model is then defined by the dynamics

$$x(k+j+1|k) = Ax(k+j|k) + Bu(k+j|k) \quad (2.53a)$$

$$y(k+j|k) = Cx(k+j|k) + Du(k+j|k), \quad (2.53b)$$

subject to the tightened constraints

$$y(k+j|k) \in \mathcal{Y}(j) \quad (2.54a)$$

$$x(k+N(k)|k) \in \mathcal{T}(N(k)). \quad (2.54b)$$

The operator ‘ $\ominus$ ’ denotes the Pontryagin set difference, or P-difference, which is also known as the set erosion operation used in mathematical morphology [Rak+06]. It is defined for sets  $\mathcal{A}$  and  $\mathcal{B}$  by

$$\mathcal{A} \ominus \mathcal{B} = \{x \mid x + b \in \mathcal{A}, \forall b \in \mathcal{B}\}. \quad (2.55)$$

The properties of the P-difference are described in [KG95]. Implementation of the operator for polyhedral sets is discussed in [Ker00]. The robust model predictive controller now minimises a cost function of the form (2.38), subject to the dynamics (2.53) and the operating constraints (2.54).

### 2.6.2 Controller Properties

The following theorems describe the recursive feasibility and finite-time completion guarantees provided by this constraint tightened controller.

**Theorem 2.11** (Robust Recursive Feasibility). *Given the optimal input, state and output sequences to the system at time  $k$ , namely  $\{u^*(k+j|k)\}_{j=0}^{N^*(k)-1}$ ,  $\{x^*(k+j|k)\}_{j=0}^{N^*(k)}$  and  $\{y^*(k+j|k)\}_{j=0}^{N^*(k)-1}$ , with optimal horizon length  $N^*(k)$ , a candidate suboptimal, but feasible trajectory at time  $k+1$  is defined by the relationships*

$$\hat{u}(k+j+1|k+1) = u^*(k+j+1|k) + K(j)L(j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]} \quad (2.56a)$$

$$\hat{x}(k+j+1|k+1) = x^*(k+j+1|k) + L(j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-1]} \quad (2.56b)$$

$$\hat{y}(k+j+1|k+1) = u^*(k+j+1|k) + (C + DK(j))L(j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]} \quad (2.56c)$$

$$\hat{N}(k+1) = N^*(k) - 1. \quad (2.56d)$$

*Proof.* The full proof is provided in [RH06a]. It shows that the candidate solution satisfies the dynamics constraints, and is admissible with respect to the tightened constraints. Then, by induction, this implies that the optimisation is feasible until completion, given an initial feasible solution. ■

**Theorem 2.12** (Robust Finite-Time Completion). *If a feasible solution exists at time  $k$  and the weighting on the cost function (2.38) satisfies*

$$0 < \eta = 1 - \gamma \left( \max_{w \in \mathcal{W}} \sum_{j=0}^{\infty} \|K(j)L(j)w\| \right), \quad (2.57)$$

*then the target set  $\mathcal{T}$  will be reached in at most  $\lfloor J^*(x(k))/\eta \rfloor$  steps, from the current state.*

*Proof.* See [RH06a]. The cost of the candidate solution (2.56) is compared to the optimal cost at time  $k$ . Using the triangle inequality, this places a lower bound on the amount by which the cost decreases as in (2.36). Treating the cost as a Lyapunov-like function, if the property (2.57) is satisfied, the set  $\mathcal{T}$  must be entered in finite time as  $J^*(\cdot)$  is non-negative by construction and takes on the value zero within  $\mathcal{T}$ . The maximum number of steps to completion is then given by  $\lfloor J^*(x(k))/\eta \rfloor$ . ■

**Remark 2.13.** A  $p$ -norm cost is used rather than a quadratic cost, as the lower bound on the value function under the action of disturbances in Theorem 2.12 requires the stage cost to satisfy the triangle inequality. The quadratic cost, being a squared norm rather than a norm, does not satisfy this property.

**Corollary 2.14** (Lower Bound on Value Function). *For any state  $x \notin \mathcal{T}$  for which an initial feasible solution exists,*

$$J^*(x) \geq \eta \quad (2.58)$$

*Proof.* The result follows from the fact that the value function must reduce by at least  $\eta$  at each step and is zero within the terminal set. ■

### 2.6.3 Disturbance Feedback Formulation

In [KRH07], a constraint tightening policy parameterised in terms of direct feedback on the disturbance is presented. It is shown that all state feedback tightening policies can be represented in this form, but not all disturbance feedback policies have an equivalent state feedback representation. Furthermore, it is shown that the expressions for the tightening matrices are now convex functions of the disturbance feedback policy. For these reasons, the disturbance feedback approach is used exclusively in this thesis. The detailed formulation is introduced in subsequent chapters, so is omitted here.

## 2.7 Optimisation

It can be shown that the standard linear-quadratic fixed horizon MPC problem can be expressed in the form of a Quadratic Program (QP), if the constraint sets are formed by the intersection of linear half-spaces [Mac02]. Similarly, a 1-norm cost function will result in a linear program. However, the variable horizon introduces an additional complication, as the cost function is not, in general, a convex function of the horizon length. Two main methods are employed to handle this non-convexity. It is possible to simply evaluate the fixed horizon value function corresponding to each horizon length and choose the one with a minimum cost. Alternatively, Mixed-Integer Programming (MIP) may be used to solve a quasi-convex optimisation problem [RH06a; RH03; BM99]. This section provides an overview of the latter. To ensure a tractable formulation, a maximum horizon length needs to be specified, which will be denoted  $\bar{N}$ .

### 2.7.1 Mixed-Integer Formulation

The prediction model is augmented with the binary variable sequence

$$\mathbf{b}(k) = \{b(k+j|k) \in \{0,1\}\}_{j=0}^{\bar{N}}, \quad (2.59)$$

where the variable  $b(k + j|k)$  is set to unity when completion is predicted at step  $j$ . Then, the VH-MPC optimisation problem can be defined as solving

$$J^*(x(k)) = \min_{\mathbf{u}^{(k)}, \mathbf{b}^{(k)}} \sum_{j=0}^{\bar{N}-1} j b(k + j|k) + \gamma \|u(k + j|k)\|, \quad (2.60)$$

subject to the tightened mixed logic and dynamics constraints

$$\sum_{l=0}^j b(k + l|k) = 0 \implies y(k + j|k) \in \mathcal{Y}(j) \quad (2.61a)$$

$$b(k + j|k) = 1 \implies x(k + j|k) \in \mathcal{T}(j) \quad (2.61b)$$

$$\sum_{l=0}^{\bar{N}} b(k + l|k) = 1. \quad (2.61c)$$

Constraint (2.61a) enforces the output constraints only whilst completion has not been reached, constraint (2.61b) ensures that the binary variable is set to unity only when the state is within the terminal set and (2.61c) mandates that completion is achieved within  $\bar{N}$  steps.

**Remark 2.15.** The constraint (2.61c) imposes a particular form on the optimal solution, namely that completion is reached at precisely one step along the prediction horizon. However, this does not require the state to leave the terminal set after this time, as (2.61b) is only a one-way implication. Furthermore, if the (2.61c) is relaxed to allow the sum of the binary variables to be greater than or equal to unity, the trajectory that results from solving (2.60) will still have the same form by optimality.

## 2.7.2 Implementation

If the output and terminal constraints are in half-space form, a ‘‘big-M’’ formulation can be used to implement the logical implications [Sch+01]. If the output constraint sets can be expressed in the form

$$\mathcal{Y}(j) = \{y \mid E(j)y \leq f(j)\}, \quad (2.62)$$

for matrices  $E(\cdot)$  and vectors  $f(\cdot)$ , then (2.61a) can be written as

$$E(j)y(k + j|k) \leq f(j) + M \left( \sum_{l=0}^j b(k + l|k) \right) \mathbf{1}, \quad (2.63)$$

for some suitably large number  $M$ . This ensures relaxation of the constraint once completion is reached. Similarly, if the terminal constraint sets can be expressed in the form

$$\mathcal{T}(j) = \{x \mid G(j)x \leq h(j)\}, \quad (2.64)$$

then (2.61b) can be written in the form

$$G(j)x(k + j|k) \leq h(j) + M(1 - b(k + j|k))\mathbf{1}, \quad (2.65)$$

which ensures that if  $b(k + j|k)$  is set to unity, the appropriate terminal constraint must be satisfied.

With a 1-norm objective function (2.38), the resulting optimisation problem solved at each iteration is a Mixed-Integer Linear Program (MILP). If a 2-norm is used instead, the resulting problem is a Mixed-Integer Second Order Cone Program (MISOCP). These optimisation methods rely on branch-and-bound techniques [Sch86] to intelligently solve the convex problems corresponding to fixed sequences of values for  $\mathbf{b}(k)$ .



## Chapter 3

# Feasible Contingencies

### 3.1 Introduction

In real-world systems, it is important to consider the robustness of a controller to changes in the control objective, system dynamics, constraints, or a combination of these. In all cases, it is desirable to have one or more *contingencies* available, if the changing objectives, dynamics and constraints are well defined. An example of such a scenario is fault tolerance, where system parameters under fault may be known, but the exact time at which the fault will occur is not. If and when the changes occur, the appropriate contingencies can then be *activated*, that is, a new control problem can be posed under the changed conditions. The new problem will only be feasible, however, if the system state at the time of contingency activation lies within the modified region of attraction.

It is therefore prudent to consider the design of a controller that ensures the availability of feasible contingencies at all times prior to completion of the original control problem. For practical applicability, it is also important for the controller to be robust to disturbances. To design such a controller, this chapter builds upon ideas from prior work on nominal MPC formulations having additional safety constraints. Passive and active fault-tolerance formulations for spacecraft rendezvous that guarantee collision free abort trajectories in case of thruster failure are presented in [BH08] and [BH07] respectively. The strategy detailed in [SHF04], on the other hand, ensures the existence of a safe periodic trajectory at the end of every prediction horizon, as applied to vehicle path planning. A third idea is presented in [Car+08] for a continuous time predictive controller with a reactive *safety mode*, where the ability to track some safety reference trajectory is always available. Robustness is provided with an adapted tube formulation.

In combining elements of these strategies, the formulation outlined in the following sections extends the idea of fault-tolerant escape trajectories to more general contingency scenarios, by ensuring that the system state can be steered to one or more well defined target regions if contingency activation is required. It does so by explicitly guaranteeing the existence of controlled trajectories, or feasible contingencies, at every future state prediction. The approach

in this chapter differs from the active and passive abort strategies by using receding horizon optimisation as opposed to single-shot optimisation for path planning. This allows robustness to bounded disturbances to be provided through a modified constraint tightening approach. Various scenarios, including costed, prioritised and state-dependent contingencies are considered. Two example systems are then used to demonstrate the efficacy of the formulation.

### 3.1.1 Nomenclature

In this chapter, the symbol ' $\oplus$ ' denotes the Minkowski sum, also known as the dilation operation in mathematical morphology. It is defined for sets  $\mathcal{A}$  and  $\mathcal{B}$  by

$$\mathcal{A} \oplus \mathcal{B} = \{x \mid x = a + b, \forall a \in \mathcal{A}, b \in \mathcal{B}\}. \quad (3.1)$$

## 3.2 Feasible Contingencies

This section poses the control problem for a general discrete-time system and states the conditions on a control law under which a contingency is guaranteed to be available, for the nominal and robust cases respectively.

### 3.2.1 Nominal Control Problem

Consider a switched discrete-time system of the form

$$x(k+1) = \begin{cases} f(x(k), u(k)) & k < k_s \\ \tilde{f}^{(\alpha)}(x(k), u(k)) & k \geq k_s, \end{cases} \quad (3.2)$$

$$y(k) = \begin{cases} g(x(k), u(k)) & k < k_s \\ \tilde{g}^{(\alpha)}(x(k), u(k)) & k \geq k_s, \end{cases} \quad (3.3)$$

having state vector  $x(\cdot) \in \mathbb{R}^n$ , input vector  $u(\cdot) \in \mathbb{R}^m$  and output vector  $y(\cdot) \in \mathbb{R}^p$ , where  $k_s$  is some unknown switching time and  $\alpha \in \mathbb{N}_{[1, n_c]}$  is an unknown parameter, assuming there are  $n_c$  possible contingencies that may need to be activated. The output vector is subject to the pointwise-in-time switched constraints

$$y(k) \in \begin{cases} \mathcal{Y} & k < k_s \\ \tilde{\mathcal{Y}}^{(\alpha)} & k \geq k_s, \end{cases} \quad (3.4)$$

where  $\mathcal{Y}$  and  $\tilde{\mathcal{Y}}^{(\alpha)}$  are closed, for all  $\alpha$ . The primary control problem is to find an output-admissible feedback policy  $u(k) = \kappa(x(k))$  that drives the system state to some target set  $\mathcal{T}$  in finite time  $k_c \leq \bar{N}$ , given a maximum horizon length of  $\bar{N}$  steps. If the dynamics switch and the contingency is activated before  $\mathcal{T}$  has been reached, that is, if  $k_s < k_c$ , the new control problem

is to drive the state to a contingency target set  $\tilde{\mathcal{T}}^{(\alpha)}$  in at most  $\bar{M}^{(\alpha)}$  steps after contingency activation, with the new dynamics, constraints and disturbance input specified by the particular value of  $\alpha$ .

**Remark 3.1.** A technical assumption is imposed on the sets  $\mathcal{Y}$  and  $\tilde{\mathcal{Y}}^{(\alpha)}$ , namely that the intersection of the output admissible sets of states corresponding to both sets is non-empty. This ensures that there exists a feasible region within which the state at the time of contingency activation is compatible with both sets of constraints.

**Remark 3.2.** In the control problem, the dynamics and constraints are assumed to only switch once, at the time  $k_s$ .

For notational convenience, define the input vector  $\mu(i)$ , state vector  $\phi(i)$  and output vector  $\psi(i)$ , for  $i \geq 0$  as

$$\mu(i) = u(k_s + i) \quad (3.5)$$

$$\phi(i) = x(k_s + i) \quad (3.6)$$

$$\psi(i) = y(k_s + i), \quad (3.7)$$

which will be used to represent the dynamics of the system after the activation of contingency  $\alpha$ . The non-negativity of  $i$  implies that these variables only have meaning after contingency activation. Using this notation, define the  $q$ -step controllable set [Ker00]

$$\mathcal{K}_q(\tilde{\mathcal{T}}^{(\alpha)}) = \left\{ \phi(0) \in \mathbb{R}^n \mid \exists \{ \mu(i) \}_0^{q-1} : \{ \psi(i) \in \tilde{\mathcal{Y}}^{(\alpha)} \}_0^{q-1}, \phi(q) \in \tilde{\mathcal{T}}^{(\alpha)} \right\}. \quad (3.8)$$

This is the set of all states for which there exists an admissible input sequence after activation of contingency  $\alpha$  that guides the system state admissibly to some target set  $\tilde{\mathcal{T}}^{(\alpha)}$  in exactly  $q$  steps.

**Definition 3.3** (Nominal Contingency Feasibility). For the contingency  $\alpha$  to be nominally feasible, the set

$$\mathcal{F}_{\bar{M}^{(\alpha)}}(\tilde{\mathcal{T}}^{(\alpha)}) = \bigcup_{q=1}^{\bar{M}^{(\alpha)}} \mathcal{K}_q(\tilde{\mathcal{T}}^{(\alpha)}) \quad (3.9)$$

must be nonempty.

**Criterion 3.4** (Nominal Contingency Availability). For a contingency to be nominally available, it must be nominally feasible and the control input  $u(k)$  applied to the system must ensure that

$$x(k) \in \mathcal{F}_{\bar{M}^{(\alpha)}}(\tilde{\mathcal{T}}^{(\alpha)}) \quad (3.10)$$

for all  $0 < k < k_c$ .

**Remark 3.5.** If there is no time limit on contingency completion, it is possible to relax the completion time constraint by letting  $\bar{M}^{(\alpha)} \rightarrow \infty$  in (3.9), if the limit exists.

### 3.2.2 Robustness

A bounded state disturbance is now introduced into the formulation. Using a slight abuse of notation, the state dynamics are altered to

$$x(k+1) = \begin{cases} f(x(k), u(k), w(k)) & k < k_s \\ \tilde{f}^{(\alpha)}(x(k), u(k), w(k)) & k \geq k_s, \end{cases} \quad (3.11)$$

where the disturbance satisfies the pointwise-in-time switched constraints

$$w(k) \in \begin{cases} \mathcal{W} & k < k_s \\ \tilde{\mathcal{W}}^{(\alpha)} & k \geq k_s. \end{cases} \quad (3.12)$$

The control problem is now to find an admissible feedback policy  $u(k) = \kappa(x(k))$  that drives the system state to the target set  $\mathcal{T}$  in finite time  $k_c$ , whilst allowing the state to be driven to the set  $\tilde{\mathcal{T}}^{(\alpha)}$  in  $\bar{M}^{(\alpha)}$  steps or fewer, for any allowable disturbance realisation. Again, for notational convenience, define the disturbance after the dynamics switch as

$$\omega(i) = w(k_s + i). \quad (3.13)$$

Now define the robust  $q$ -step controllable set [Ker00]

$$\begin{aligned} \tilde{\mathcal{K}}_q(\tilde{\mathcal{T}}^{(\alpha)}) = \{ & \phi(0) \in \mathbb{R}^n \mid \exists \{\mu(i) = \tilde{\kappa}(\phi(j))\}_0^{q-1} : \\ & \{\psi(i) \in \tilde{\mathcal{Y}}^{(\alpha)}\}_0^{q-1}, \phi(q) \in \tilde{\mathcal{T}}^{(\alpha)}, \forall \omega(i) \in \tilde{\mathcal{W}}^{(\alpha)} \}. \end{aligned} \quad (3.14)$$

This is the set of all states for which there exists a control law  $\tilde{\kappa}(\cdot)$  after activation of contingency  $\alpha$  that guides the system state admissibly to the target set  $\tilde{\mathcal{T}}^{(\alpha)}$  in exactly  $q$  steps.

**Definition 3.6** (Robust Contingency Feasibility). For the contingency  $\alpha$  to be robustly feasible, the set

$$\tilde{\mathcal{F}}_{\bar{M}^{(\alpha)}}(\tilde{\mathcal{T}}^{(\alpha)}) = \bigcup_{q=1}^{\bar{M}^{(\alpha)}} \tilde{\mathcal{K}}_q(\tilde{\mathcal{T}}^{(\alpha)}) \quad (3.15)$$

must be nonempty.

**Criterion 3.7** (Robust Contingency Availability). For the contingency  $\alpha$  to be robustly available, it must be robustly feasible and the control law applied in the closed loop  $u(k) = \kappa(x(k))$  must ensure that for all  $0 < k < k_c$ ,

$$x(k) \in \tilde{\mathcal{F}}_{\bar{M}^{(\alpha)}}(\tilde{\mathcal{T}}^{(\alpha)}). \quad (3.16)$$

### 3.2.3 Linear Case

Having introduced the feasible contingency criteria for a general system, the particular case of a linear system will be analysed. The dynamics prior to contingency activation are described

by the equations

$$f(x(k), u(k)) = Ax(k) + Bu(k) + w(k) \quad (3.17)$$

$$g(x(k), u(k)) = Cx(k) + Du(k), \quad (3.18)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ , and  $D \in \mathbb{R}^{p \times m}$ . The ideal control objective is to admissibly reach the set  $\mathcal{T}$  in finite time, from the initial state  $x(0)$ , whilst minimising a p-norm cost function of the form

$$J(x(0), \mathbf{y}) = \sum_{k=0}^{k_c-1} (1 + \|\Theta(\mathbf{y}(k) - \mathbf{y}_r)\|), \quad (3.19)$$

where  $\mathbf{y} = \{y(0), y(1), \dots, y(N-1)\}$ ,  $\Theta \in \mathbb{R}^{p \times p}$  and  $\mathbf{y}_r$  is some reference output, which is not required to correspond to an equilibrium state. It is required that

$$x(k_c) \in \mathcal{T}. \quad (3.20)$$

The cost function  $J(\cdot, \cdot)$  penalises a weighted sum of output-dependent operating costs and time to completion. Using this form of cost function allows more general costs to be encapsulated than 2.38.

Assuming linear time-invariant dynamics, the system after switching can be similarly represented by the equations

$$\begin{aligned} \tilde{f}^{(\alpha)}(x(k), u(k)) &= \tilde{A}^{(\alpha)}x(k) + \tilde{B}^{(\alpha)}u(k) + w(k) \\ \tilde{g}^{(\alpha)}(x(k), u(k)) &= \tilde{C}^{(\alpha)}x(k) + \tilde{D}^{(\alpha)}u(k), \end{aligned} \quad (3.21)$$

where  $\tilde{A}^{(\alpha)} \in \mathbb{R}^{n \times n}$ ,  $\tilde{B}^{(\alpha)} \in \mathbb{R}^{n \times m}$ ,  $\tilde{C}^{(\alpha)} \in \mathbb{R}^{p \times n}$ , and  $\tilde{D}^{(\alpha)} \in \mathbb{R}^{p \times m}$ .

**Remark 3.8.** It is assumed that the pairs  $(A, B)$  and  $(\tilde{A}^{(\alpha)}, \tilde{B}^{(\alpha)})$  are controllable.

### 3.3 Robust MPC with Contingencies

An MPC controller will now be formulated to ensure reachability of the target set  $\tilde{\mathcal{T}}^{(\alpha)}$  within  $\bar{M}^{(\alpha)}$  steps to satisfy Criterion 3.7. Only the robust case is considered as the nominal problem is a special case with  $w(k) = 0$  for all  $k$ . Define the nominal prediction model under normal operation by

$$x(k|k) = x(k) \quad (3.22)$$

$$x(k+j+1|k) = Ax(k+j|k) + Bu(k+j|k) \quad (3.23)$$

$$y(k+j|k) = Cx(k+j|k) + Du(k+j|k). \quad (3.24)$$

It is assumed that the maximum prediction horizon is  $\bar{N}$ . Also define the prediction model under a change in dynamics at prediction step  $j \geq 1$  by the equations

$$\phi_j^{(\alpha)}(0|k) = x(k+j|k) \quad (3.25)$$

$$\phi_j^{(\alpha)}(i+1|k) = \tilde{A}^{(\alpha)}\phi_j^{(\alpha)}(i|k) + \tilde{B}^{(\alpha)}\mu_j^{(\alpha)}(i|k) \quad (3.26)$$

$$\psi_j^{(\alpha)}(i|k) = \tilde{C}^{(\alpha)}\phi_j^{(\alpha)}(i|k) + \tilde{D}^{(\alpha)}\mu_j^{(\alpha)}(i|k). \quad (3.27)$$

These variables predict the behaviour of the system for one particular contingency  $\alpha$ .

Recursively define the matrices

$$L(0) = I_n \quad (3.28)$$

$$L(j+1) = AL(j) + BP(j) \quad (3.29)$$

$$Q(j) = CL(j) + DP(j) \quad (3.30)$$

for the primary trajectory and

$$\tilde{L}^{(\alpha)}(0) = I_n \quad (3.31)$$

$$\tilde{L}^{(\alpha)}(i+1) = \tilde{A}^{(\alpha)}\tilde{L}^{(\alpha)}(i) + \tilde{B}^{(\alpha)}\tilde{P}^{(\alpha)}(i) \quad (3.32)$$

$$\tilde{Q}^{(\alpha)}(i) = \tilde{C}^{(\alpha)}\tilde{L}^{(\alpha)}(i) + \tilde{D}^{(\alpha)}\tilde{P}^{(\alpha)}(i) \quad (3.33)$$

for contingency  $\alpha$ , as well as the corresponding sets

$$\mathcal{Y}(0) = \mathcal{Y}, \quad \mathcal{Y}(j+1) = \mathcal{Y}(j) \ominus Q(j)\mathcal{W} \quad (3.34)$$

$$\mathcal{T}(0) = \mathcal{T}, \quad \mathcal{T}(j+1) = \mathcal{T}(j) \ominus L(j)\mathcal{W} \quad (3.35)$$

for the primary trajectory and

$$\tilde{\mathcal{Y}}_0^{(\alpha)}(0) = \tilde{\mathcal{Y}}^{(\alpha)}, \quad \tilde{\mathcal{Y}}_0^{(\alpha)}(i+1) = \tilde{\mathcal{Y}}_0^{(\alpha)}(i) \ominus \tilde{Q}^{(\alpha)}(i)\tilde{\mathcal{W}}^{(\alpha)} \quad (3.36)$$

$$\tilde{\mathcal{T}}_0^{(\alpha)}(0) = \tilde{\mathcal{T}}^{(\alpha)}, \quad \tilde{\mathcal{T}}_0^{(\alpha)}(i+1) = \tilde{\mathcal{T}}_0^{(\alpha)}(i) \ominus \tilde{L}^{(\alpha)}(i)\tilde{\mathcal{W}}^{(\alpha)} \quad (3.37)$$

$$\tilde{\mathcal{Y}}_{j+1}^{(\alpha)}(i) = \tilde{\mathcal{Y}}_j^{(\alpha)}(i) \ominus \tilde{Q}^{(\alpha)}(i)L(j)\mathcal{W} \quad (3.38)$$

$$\tilde{\mathcal{T}}_{j+1}^{(\alpha)}(i) = \tilde{\mathcal{T}}_j^{(\alpha)}(i) \ominus \tilde{L}^{(\alpha)}(i)L(j)\mathcal{W} \quad (3.39)$$

for the contingency, where  $j \in \mathbb{Z}_{[0, \bar{N}]}$  and  $i \in \mathbb{Z}_{[0, \bar{M}^{(\alpha)}]}$ .

In a similar manner to the state-feedback policy introduced in Section 2.6, the sequence of matrices  $\{P(j)\}$  defines the constraint-tightening policy in disturbance-feedback form. Additionally, the sequence  $\{\tilde{P}^{(\alpha)}(i)\}$  denotes the policy corresponding to contingency  $\alpha$ . Tightening constraints to ensure that these policies can be feasibly implemented guarantees robustness to the bounded disturbances before and after contingency activation.

The MPC optimisation problem is then to solve

$$J^*(x(k)) = \min_{\theta} \sum_{j=0}^{N(k)-1} (1 + \|\Theta(y(k+j|k) - y_r)\|) \quad (3.40)$$

subject to

$$y(k+j|k) \in \mathcal{Y}(j) \quad (3.41a)$$

$$x(k+N(k)|k) \in \mathcal{T}(N(k)) \quad (3.41b)$$

$$0 \leq N(k) \leq \bar{N} \quad (3.41c)$$

$$\psi_j^{(\alpha)}(i|k) \in \tilde{\mathcal{Y}}_j^{(\alpha)}(i) \quad (3.41d)$$

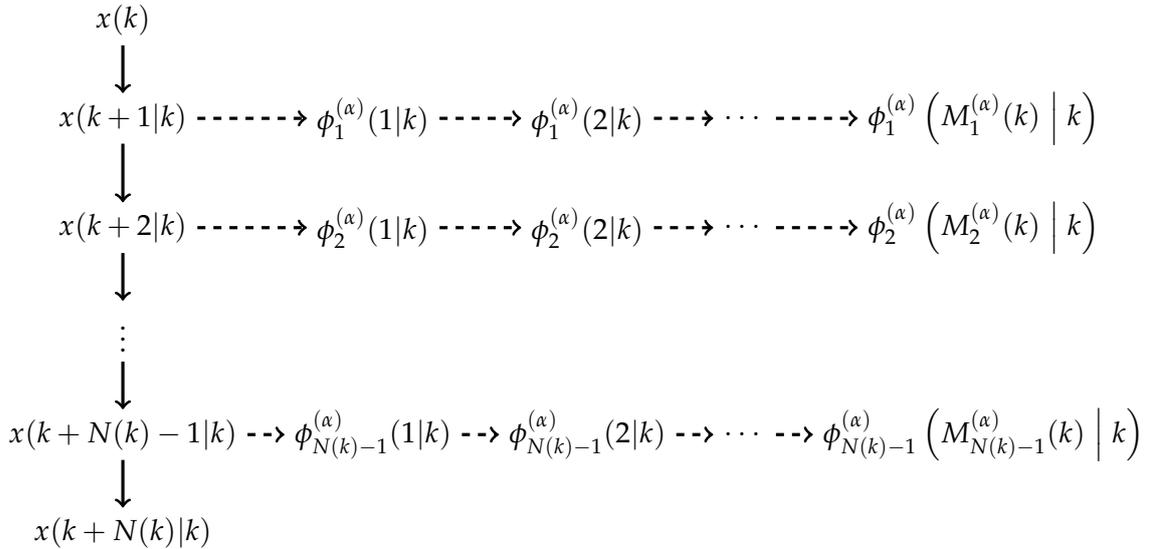
$$\phi_j^{(\alpha)}(M_j^{(\alpha)}(k)|k) \in \tilde{\mathcal{T}}_j^{(\alpha)}(M_j^{(\alpha)}(k)) \quad (3.41e)$$

$$0 \leq M_j^{(\alpha)}(k) \leq \bar{M}^{(\alpha)}, \quad (3.41f)$$

where  $j \in \mathbb{Z}_{[0, N(k)-1]}$  for the primary prediction variables,  $j \in \mathbb{N}_{[1, N(k)-1]}$  for the contingency prediction variables,  $i \in \mathbb{Z}_{[0, M_j^{(\alpha)}(k)-1]}$  and  $\Theta \in \mathbb{R}^{p \times p}$  is a weighting matrix. The optimisation is with respect to the set of decision variables,

$$\theta = \bigcup_{i,j} \left\{ u(k+j|k), \mu_j(i|k), N(k), M_j^{(\alpha)}(k) \right\}, \quad (3.42)$$

where it is assumed that  $n_c = 1$ . Figure 3.1 illustrates the relationships between the state prediction variables, where the solid arrows indicate dynamics under normal operation and the dashed arrows indicate the dynamics of contingency  $\alpha$ .



**Figure 3.1:** Illustration of MPC with Contingency Predictions

**Remark 3.9.** The tightening of the constraint sets in (3.34)-(3.39) is essentially carried out twice, to account for the effect of the disturbance before and after contingency activation.

---

**Algorithm 3.1:** MPC with Feasible Contingency
 

---

```

1 while  $x(k) \notin \mathcal{T}$  do
2   Solve (3.40) subject to (3.41);
3   Apply the first element of the resulting optimal input sequence to the system;
4    $k \leftarrow k + 1$ ;
5   if contingency  $\alpha$  requires activation then
6     Switch to an auxiliary controller to handle the contingency;
7     exit;
8   end
9 end
    
```

---

**Remark 3.10.** The states and inputs after anticipated contingency activation are only constrained for  $j \geq 1$  as the current state, which corresponds to  $j = 0$ , cannot be modified within the MPC optimisation problem. However, for  $k > 0$ , a feasible trajectory for  $j = 0$  will exist, as will be subsequently shown.

Algorithm 3.1 describes how this optimisation problem is used to control the system to the target set, whilst ensuring that contingency  $\alpha$  is available if required.

**Theorem 3.11** (Robust Recursive Feasibility). *Given the optimal solution to the MPC optimisation in Algorithm 3.1 at time  $k$ , namely the sequences  $\{u^*(k+j|k)\}$ ,  $\{x^*(k+j|k)\}$ ,  $\{y^*(k+j|k)\}$ ,  $\{\phi_j^{*(\alpha)}(i|k)\}$ ,  $\{\mu_j^{*(\alpha)}(i|k)\}$  and  $\{\psi_j^{*(\alpha)}(i|k)\}$ , as well as the completion times  $N^*(k)$  and  $M_j^{*(\alpha)}(k)$ , a feasible solution to the optimisation problem at time  $k+1$  is given by*

$$\hat{u}(k+j+1|k+1) = u^*(k+j+1|k) + P(j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]} \quad (3.43a)$$

$$\hat{x}(k+j+1|k+1) = x^*(k+j+1|k) + L(j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-1]} \quad (3.43b)$$

$$\hat{y}(k+j+1|k+1) = y^*(k+j+1|k) + Q(j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]} \quad (3.43c)$$

$$\hat{\mu}_j^{(\alpha)}(i|k+1) = \mu_{j+1}^{*(\alpha)}(i|k) + \tilde{P}^{(\alpha)}(i)L(j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]}, i \in \mathbb{Z}_{[0, M_{j+1}^{*(\alpha)}(k)-1]} \quad (3.43d)$$

$$\hat{\phi}_j^{(\alpha)}(i|k+1) = \phi_{j+1}^{*(\alpha)}(i|k) + \tilde{L}^{(\alpha)}(i)L(j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]}, i \in \mathbb{Z}_{[0, M_{j+1}^{*(\alpha)}(k)]} \quad (3.43e)$$

$$\hat{\psi}_j^{(\alpha)}(i|k+1) = \psi_{j+1}^{*(\alpha)}(i|k) + \tilde{Q}^{(\alpha)}(i)L(j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]}, i \in \mathbb{Z}_{[0, M_{j+1}^{*(\alpha)}(k)-1]} \quad (3.43f)$$

$$\hat{N}(k) = N^*(k) - 1 \quad (3.43g)$$

$$\hat{M}_j^{(\alpha)}(k+1) = M_{j+1}^{*(\alpha)}(k). \quad (3.43h)$$

*Proof.* The proof follows a similar methodology to [RH06a], albeit modified to include the contingency prediction variables and disturbance-feedback constraint tightening.

Satisfaction of the initial conditions (3.22) and (3.25) can be proven as follows. For the primary trajectory prediction, (3.23) and Algorithm 3.1 can be used to show that

$$\hat{x}(k+1|k+1) = x^*(k+1|k) + w(k) \quad (3.44)$$

$$= Ax(k) + Bu^*(k|k) + w(k) \quad (3.45)$$

$$= Ax(k) + Bu(k) + w(k) \quad (3.46)$$

$$= x(k+1). \quad (3.47)$$

In a similar way, (3.43b) and (3.43e) can be used to show that

$$\hat{\phi}_j^{(\alpha)}(0|k+1) = \phi_{j+1}^{*(\alpha)}(0|k) + \tilde{L}^{(\alpha)}(0)L(j)w(k) \quad (3.48)$$

$$= x^*(k+j+1|k) + L(j)w(k) \quad (3.49)$$

$$= \hat{x}(k+j+1|k+1). \quad (3.50)$$

For the system dynamics, (3.23) and (3.29) can be used to show that the candidate solution satisfies

$$\hat{x}(k+j+2|k+1) = x^*(k+j+2|k) + L(j+1)w(k) \quad (3.51)$$

$$= Ax^*(k+j+1|k) + Bu^*(k+j+1|k) + L(j+1)w(k) \quad (3.52)$$

$$\begin{aligned} &= A(\hat{x}(k+j+1|k+1) - L(j)w(k)) + B(\hat{u}(k+j+1|k+1) - P(j)w(k)) \\ &\quad + L(j+1)w(k) \end{aligned} \quad (3.53)$$

$$\begin{aligned} &= A\hat{x}(k+j+1|k+1) + B\hat{u}(k+j+1|k+1) - (AL(j) + BP(j))w(k) \\ &\quad + L(j+1)w(k) \end{aligned} \quad (3.54)$$

$$\begin{aligned} &= A\hat{x}(k+j+1|k+1) + B\hat{u}(k+j+1|k+1) - L(j+1)w(k) \\ &\quad + L(j+1)w(k) \end{aligned} \quad (3.55)$$

$$= A\hat{x}(k+j+1|k+1) + B\hat{u}(k+j+1|k+1). \quad (3.56)$$

For the contingency predictions, (3.26) and (3.32) show that

$$\hat{\phi}_j^{(\alpha)}(i+1|k+1) = \phi_{j+1}^{*(\alpha)}(i+1|k) + \tilde{L}^{(\alpha)}(i+1)L(j)w(k) \quad (3.57)$$

$$= \tilde{A}^{(\alpha)}\phi_{j+1}^{*(\alpha)}(i|k) + \tilde{B}^{(\alpha)}\mu_{j+1}^{*(\alpha)}(i|k) + \tilde{L}^{(\alpha)}(i+1)L(j)w(k) \quad (3.58)$$

$$\begin{aligned} &= \tilde{A}^{(\alpha)}(\hat{\phi}_j^{(\alpha)}(i|k+1) - \tilde{L}^{(\alpha)}(i)L(j)w(k)) + \tilde{B}^{(\alpha)}(\hat{\mu}_j^{(\alpha)}(i|k+1) \\ &\quad - \tilde{P}^{(\alpha)}(i)L(j)w(k)) + \tilde{L}^{(\alpha)}(i+1)L(j)w(k) \end{aligned} \quad (3.59)$$

$$\begin{aligned} &= \tilde{A}^{(\alpha)}\hat{\phi}_j^{(\alpha)}(i|k+1) + \tilde{B}^{(\alpha)}\hat{\mu}_j^{(\alpha)}(i|k+1) \\ &\quad - (\tilde{A}^{(\alpha)}\tilde{L}^{(\alpha)}(i) + \tilde{B}^{(\alpha)}\tilde{P}^{(\alpha)}(i))L(j)w(k) + \tilde{L}^{(\alpha)}(i+1)L(j)w(k) \end{aligned} \quad (3.60)$$

$$\begin{aligned} &= \tilde{A}^{(\alpha)}\hat{\phi}_j^{(\alpha)}(i|k+1) + \tilde{B}^{(\alpha)}\hat{\mu}_j^{(\alpha)}(i|k+1) - \tilde{L}^{(\alpha)}(i+1)L(j)w(k) \\ &\quad + \tilde{L}^{(\alpha)}(i+1)L(j)w(k) \end{aligned} \quad (3.61)$$

$$= \tilde{A}^{(\alpha)}\hat{\phi}_j^{(\alpha)}(i|k+1) + \tilde{B}^{(\alpha)}\hat{\mu}_j^{(\alpha)}(i|k+1). \quad (3.62)$$

For the output dynamics, (3.24) and (3.30) can be used to show that the candidate solution

satisfies

$$\hat{y}(k+j+1|k+1) = y^*(k+j+1|k) + Q(j)w(k) \quad (3.63)$$

$$= Cx^*(k+j+1|k) + Du^*(k+j+1|k) + Q(j)w(k) \quad (3.64)$$

$$\begin{aligned} &= C(\hat{x}(k+j+1|k+1) - L(j)w(k)) + D(\hat{u}(k+j+1|k+1) - P(j)w(k)) \\ &\quad + Q(j)w(k) \end{aligned} \quad (3.65)$$

$$\begin{aligned} &= C\hat{x}(k+j+1|k+1) + D\hat{u}(k+j+1|k+1) - (CL(j) + DP(j))w(k) \\ &\quad + Q(j)w(k) \end{aligned} \quad (3.66)$$

$$\begin{aligned} &= C\hat{x}(k+j+1|k+1) + D\hat{u}(k+j+1|k+1) - Q(j)w(k) + Q(j)w(k) \\ &\quad (3.67) \end{aligned}$$

$$= C\hat{x}(k+j+1|k+1) + D\hat{u}(k+j+1|k+1) \quad (3.68)$$

For the contingency predictions, (3.27) and (3.33) show that

$$\hat{\psi}_j^{(\alpha)}(i|k+1) = \psi_{j+1}^{*(\alpha)}(i|k) + \tilde{Q}^{(\alpha)}(i)L(j)w(k) \quad (3.69)$$

$$= \tilde{C}^{(\alpha)}\phi_{j+1}^{*(\alpha)}(i|k) + \tilde{D}^{(\alpha)}\mu_{j+1}^{*(\alpha)}(i|k) + \tilde{Q}^{(\alpha)}(i)L(j)w(k) \quad (3.70)$$

$$\begin{aligned} &= \tilde{C}^{(\alpha)}(\hat{\phi}_j^{(\alpha)}(i|k+1) - \tilde{L}^{(\alpha)}(i)L(j)w(k)) + \tilde{D}^{(\alpha)}(\hat{\mu}_j^{(\alpha)}(i|k+1) \\ &\quad - \tilde{P}^{(\alpha)}(i)L(j)w(k)) + \tilde{Q}^{(\alpha)}(i)L(j)w(k) \end{aligned} \quad (3.71)$$

$$\begin{aligned} &= \tilde{C}^{(\alpha)}\hat{\phi}_j^{(\alpha)}(i|k+1) + \tilde{D}^{(\alpha)}\hat{\mu}_j^{(\alpha)}(i|k+1) - (\tilde{C}^{(\alpha)}\tilde{L}^{(\alpha)}(i) + \tilde{D}^{(\alpha)}\tilde{P}^{(\alpha)}(i))L(j)w(k) \\ &\quad + \tilde{Q}^{(\alpha)}(i)L(j)w(k) \end{aligned} \quad (3.72)$$

$$\begin{aligned} &= \tilde{C}^{(\alpha)}\hat{\phi}_j^{(\alpha)}(i|k+1) + \tilde{D}^{(\alpha)}\hat{\mu}_j^{(\alpha)}(i|k+1) - \tilde{Q}^{(\alpha)}(i)L(j)w(k) + \tilde{Q}^{(\alpha)}(i)L(j)w(k) \\ &\quad (3.73) \end{aligned}$$

$$= \tilde{C}^{(\alpha)}\hat{\phi}_j^{(\alpha)}(i|k+1) + \tilde{D}^{(\alpha)}\hat{\mu}_j^{(\alpha)}(i|k+1) \quad (3.74)$$

Satisfaction of the output constraints can be verified by using (3.38) and the definition of the P-difference (2.55).

$$\begin{aligned} y^*(k+j+1|k) \in \mathcal{Y}(j+1) &= \mathcal{Y}(j) \ominus Q(j)\mathcal{W} \\ \implies y^*(k+j+1|k) + Q(j)\mathcal{W} &= \hat{y}(k+j+1|k+1) \in \mathcal{Y}(j). \end{aligned} \quad (3.75)$$

Similarly,

$$\begin{aligned} \psi_{j+1}^{*(\alpha)}(i|k) \in \tilde{\mathcal{Y}}_{j+1}^{(\alpha)}(i) &= \tilde{\mathcal{Y}}_j^{(\alpha)}(i) \ominus \tilde{Q}^{(\alpha)}(i)L(j)\mathcal{W} \\ \implies \psi_{j+1}^{*(\alpha)}(i|k) + \tilde{Q}^{(\alpha)}(i)L(j)w(k) &= \hat{\psi}_j^{(\alpha)}(i|k+1) \in \tilde{\mathcal{Y}}_j^{(\alpha)}(i). \end{aligned} \quad (3.76)$$

In the same manner, (3.39) can be used to verify that the candidate solution satisfies the terminal constraints

$$\begin{aligned} x^*(k + N^*(k)|k) &\in \mathcal{T}(N^*(k)) = \mathcal{T}(N^*(k) - 1) \ominus L(N^*(k) - 1)\mathcal{W} \\ \implies x^*(k + N^*(k)|k) + L(N^*(k) - 1)\mathcal{W} &= \hat{x}(k + \hat{N}(k + 1)|k + 1) \in \mathcal{T}(\hat{N}(k + 1)). \end{aligned} \quad (3.77)$$

Similarly

$$\begin{aligned} \phi_{j+1}^{*(\alpha)}(M_{j+1}^{*(\alpha)}(k)|k) &\in \tilde{\mathcal{T}}_{j+1}^{(\alpha)}(M_{j+1}^{*(\alpha)}(k)) = \tilde{\mathcal{T}}_j^{(\alpha)}(M_{j+1}^{*(\alpha)}(k)) \ominus \tilde{L}^{(\alpha)}(M_{j+1}^{*(\alpha)}(k))L(j)\mathcal{W} \\ \implies \phi_{j+1}^{*(\alpha)}(M_{j+1}^{*(\alpha)}(k)|k) + \tilde{L}^{(\alpha)}(M_{j+1}^{*(\alpha)}(k))L(j)\omega(k) & \\ &= \hat{\phi}_j^{(\alpha)}(\hat{M}_j^{(\alpha)}(k + 1)|k + 1) \in \tilde{\mathcal{T}}_j^{(\alpha)}(\hat{M}_j^{(\alpha)}(k + 1)). \end{aligned} \quad (3.78)$$

■

By induction, this means that, given an initial feasible solution at time  $k$ , there exists a feasible solution until completion is reached, in the absence of contingency activation. The case of contingency activation is considered in the following theorem.

**Theorem 3.12** (Guaranteed Contingency Feasibility). *Given an initial feasible solution, Algorithm 3.1 ensures that, if contingency  $\alpha$  is activated at time  $k_s$ , there exists a control law that drives the state of the system to  $\tilde{\mathcal{T}}^{(\alpha)}$  in  $\bar{M}$  steps or fewer, for any admissible sequence of disturbances after contingency activation.*

*Proof.* Given an initial feasible solution, Theorem 3.11 guarantees that there exists a nominal input sequence

$$\mu_0^{*(\alpha)}(i|k_s), i \in \mathbb{Z}_{[0, M_0^{*(\alpha)}(k_s) - 1]} \quad (3.79)$$

and resulting trajectory to the contingency set at any time  $0 < k_s < k_c$  that satisfies

$$\psi_0^{*(\alpha)}(i|k_s) \in \tilde{\mathcal{Y}}_0^{(\alpha)}(i) \quad (3.80)$$

$$\phi_0^{*(\alpha)}(M_0^{*(\alpha)}(k_s)|k_s) \in \tilde{\mathcal{T}}_0^{(\alpha)}(M_0^{*(\alpha)}(k_s)), \quad (3.81)$$

where  $M_0^{*(\alpha)}(k_s) \leq \bar{M}^{(\alpha)}$  as enforced by (3.41). Notice also that the sets  $\tilde{\mathcal{Y}}_0^{(\alpha)}(i)$  and  $\tilde{\mathcal{T}}_0^{(\alpha)}(i)$  have been defined to ensure that, given the feasible input at time  $k_s$ , a feasible input to the system at time  $k_s + i$  that guarantees satisfaction of the original (i.e. non-tightened) output constraints  $\psi(i) \in \tilde{\mathcal{Y}}$  and terminal constraint  $\phi(M_0^{*(\alpha)}(k_s)) \in \tilde{\mathcal{T}}^{(\alpha)}$  is given by

$$\mu(i) = \begin{cases} \mu_0^{*(\alpha)}(0|k_s) & i = 0 \\ \mu_0^{*(\alpha)}(i|k_s) + \sum_{l=0}^{i-1} \tilde{P}^{(\alpha)}(i-l-1)\omega(l) & 0 < i < M_0^{*(\alpha)}(k_s). \end{cases} \quad (3.82)$$

This result follows from modifying Theorem 3.11 to consider the contingency trajectory, and applying it inductively to show that there exists a feasible solution at time  $k_s + i + 1$  given any feasible (not necessarily optimal) solution at time  $k_s + i$ .

Using  $q$  to index the predictions that would be made after contingency activation, the predicted optimal contingency input trajectory at time  $k_s + i$  is denoted  $\{\mu_0^{*(\alpha)}(i + q|k_s)\}$ . Then, a feasible predicted input trajectory at time  $k_s + 1$  is given by

$$\hat{u}(k_s + 1 + q|k_s + 1) = \mu_0^{*(\alpha)}(q + 1|k_s) + \tilde{P}^{(\alpha)}(q)\omega(0), q \in \mathbb{Z}_{[0, M_0^{*(\alpha)}(k_s) - 2]} \quad (3.83)$$

Now, assume for any  $i$  that a feasible input trajectory is given by

$$\hat{u}(k_s + i + q|k_s + i) = \mu_0^{*(\alpha)}(q + i|k_s) + \sum_{l=0}^{i-1} \tilde{P}^{(\alpha)}(q + i - l - 1)\omega(l), q \in \mathbb{Z}_{[0, M_0^{*(\alpha)}(k_s) - i - 1]} \quad (3.84)$$

Then applying the modified Theorem 3.11 to this trajectory shows that

$$\hat{u}(k_s + i + 1 + q|k_s + i + 1) = \hat{u}(k_s + i + 1 + q|k_s + i) + \tilde{P}^{(\alpha)}(q)\omega(i) \quad (3.85)$$

$$= \mu_0^{*(\alpha)}(q + 1 + i|k_s) + \sum_{l=0}^{i-1} \tilde{P}^{(\alpha)}(q + i - l)\omega(l) + \tilde{P}^{(\alpha)}(q)\omega(i) \quad (3.86)$$

$$= \mu_0^{*(\alpha)}(q + 1 + i|k_s) + \sum_{l=0}^i \tilde{P}^{(\alpha)}(q + i - l)\omega(l). \quad (3.87)$$

where  $q \in \mathbb{Z}_{[0, M_0^{*(\alpha)}(k_s) - i - 1]}$ . Given that the MPC algorithm will, at each time step, apply the first input, that is, at  $q = 0$ , the actual inputs are found by substituting this value into (3.84) for each  $i$ .

Hence, (3.82) defines a time-varying affine control law that applies direct feedback to the disturbances  $\omega(\cdot)$  entering after contingency activation. This implies that there exists a control law that drives the system from the current state to the target set in  $\bar{M}$  steps or fewer, for any admissible disturbance realisation, as required by Criterion 3.7.  $\blacksquare$

**Remark 3.13.** Theorem 3.12 provides a candidate control law to be used under contingency activation. This means that, in the event of an optimiser failure, the law could actually be applied to the system to guide it to the contingency terminal set.

**Theorem 3.14** (Robust Finite-Time Completion). *If the cost weighting matrix  $\Theta$  in (3.19) is chosen to satisfy  $\eta > 0$ , where*

$$\eta = 1 - \|\Theta\| \max_{w \in \mathcal{W}} \sum_{j=0}^{\bar{N}-2} \|Q(j)w\|, \quad (3.88)$$

*the target set will be reached in at most  $H$  steps, where*

$$H = \left\lceil \frac{J^*(x(0))}{\eta} \right\rceil. \quad (3.89)$$

*Proof.* Denote the cost of the candidate solution (3.43) as  $\hat{J}(\cdot)$ . Then, applying the triangle

inequality and maximising over all allowable disturbances and horizon lengths,

$$J^*(x(k)) - \hat{J}(x(k+1)) = \sum_{j=0}^{N^*(k)-1} (1 + \|\Theta(y^*(k+j|k) - y_r)\|) \quad (3.90)$$

$$\begin{aligned} & - \sum_{j=0}^{N^*(k)-2} (1 + \|\Theta(y^*(k+j+1|k) + Q(j)w(k) - y_r)\|) \\ & \geq 1 - \sum_{j=0}^{N^*(k)-2} \|\Theta Q(j)w(k)\| \end{aligned} \quad (3.91)$$

$$\geq 1 - \|\Theta\| \max_{w \in \mathcal{W}} \sum_{j=0}^{\bar{N}-2} \|Q(j)w\| \quad (3.92)$$

$$= \eta \quad (3.93)$$

From optimality, the cost  $\hat{J}(\cdot)$  is an upper bound on the optimal cost at time  $k+1$ , hence

$$J^*(x(k)) - J^*(x(k+1)) \geq J^*(x(k)) - \hat{J}(x(k+1)) \quad (3.94)$$

$$\geq \eta. \quad (3.95)$$

The value function is therefore a Lyapunov-like function, which reduces at each time step by at least  $\eta$ . As the cost is non-negative by construction, even within the terminal set, this implies that completion must be reached in at most  $\lfloor J^*(x(0))/\eta \rfloor$  steps.  $\blacksquare$

**Remark 3.15.** Note that  $\eta > 0$  is sufficient to guarantee finite-time completion, but is not a necessary condition, as there may be other choices of cost weighting that still maintain the guarantee for a given system.

With an appropriate choice of cost function after contingency activation, or simply applying the control law detailed in Theorem 3.12, finite-time entry of the contingency target set is also guaranteed.

### 3.3.1 MIP implementation

As stated in Chapter 2, the cost function in (3.40) is, in general, a non-convex function of the horizon length. Mixed-integer programming is one method that can be used to solve the non-convex optimisation problem and incorporate the horizon lengths before and after switching as decision variables. The prediction model is augmented with the binary variable sequences  $\{b(k+j|k) \in \{0,1\}\}$ ,  $\{\beta_j^{(\alpha)}(i|k) \in \{0,1\}\}$  and  $\{v(\alpha, j) \in \{0,1\}\}$ . The variables  $b(\cdot)$  and  $\beta_j^{(\alpha)}(\cdot)$  take the value 1 at the time at which completion is achieved on the primary and contingency trajectory predictions respectively. The variables  $v(\cdot, \cdot)$  indicate whether or not the contingency  $\alpha$  is enabled, (i.e. required to be feasible) at prediction step  $j$ , and will be used in subsequent sections of this chapter.

The binary variables can be used to represent the value function (3.40), where the summation is now over a fixed horizon length. This function takes the form

$$J^*(x(k)) = \min_{\theta_b} \sum_{j=0}^{\bar{N}-1} (jb(k+j|k) + \|\Theta(y(k+j|k) - y_r)\|), \quad (3.96)$$

where the optimisation is now with respect to the set of decision variables

$$\theta_b = \bigcup_{i,j} \left\{ u(k+j|k), \mu_j^{(\alpha)}(i|k), b(k+j|k), \beta_j^{(\alpha)}(i|k) \right\}. \quad (3.97)$$

The optimisation is carried out subject to the mixed logic and dynamics constraints

$$b(k+j|k) = 1 \implies x(k+j|k) \in \mathcal{T}(j) \quad (3.98)$$

$$\beta_j^{(\alpha)}(i|k) = 1 \implies \phi_j^{(\alpha)}(i|k) \in \tilde{\mathcal{T}}_j^{(\alpha)}(i) \quad (3.99)$$

$$\sum_{l=0}^j b(k+l|k) = 0 \implies y(k+j|k) \in \mathcal{Y}(j) \quad (3.100)$$

$$\sum_{l=0}^j b(k+l|k) + \sum_{l=0}^i \beta_j^{(\alpha)}(l|k) + (1 - v(\alpha, j)) = 0 \implies \psi_j^{(\alpha)}(i|k) \in \tilde{\mathcal{Y}}_j^{(\alpha)}(i) \quad (3.101)$$

$$\sum_{l=0}^{\bar{N}} b(k+l|k) = 1 \quad (3.102)$$

$$\sum_{l=0}^{\bar{M}^{(\alpha)}} \beta_j^{(\alpha)}(l|k) + \sum_{l=0}^j b(k+l|k) - v(\alpha, j) \geq 0, \quad (3.103)$$

where  $j \in \mathbb{Z}_{[0, \bar{N}]}$  for the primary trajectory variables,  $j \in \mathbb{N}_{[1, \bar{N}]}$  for the contingency variables and  $i \in \mathbb{Z}_{[0, \bar{M}^{(\alpha)}]}$ . Note that the condition (3.102) ensures that completion occurs somewhere along the primary trajectory and (3.103) guarantees that if primary completion has not occurred and the contingency is active at step  $j$ , completion must be reached somewhere along the contingency prediction propagated from  $x(k+j|k)$ . Also note that the output-based operating costs are relaxed once contingency completion is achieved by virtue of (3.101). If the constraint sets are polyhedral, a “big- $M$ ” formulation can be used to implement these logical implications in a similar way to Section 2.7. By replacing the optimisation problem and constraints in Algorithm 3.1 with (3.96) and (3.98) - (3.103) respectively, the resulting optimisation problem is a MIP.

### 3.4 Contingency Scenarios

The algorithm outlined in the previous section guarantees that a single contingency  $\alpha$  is feasible. This section extends the basic case to other contingency scenarios, namely costed contingencies, multiple contingencies, prioritised contingencies and state dependent contingencies. It shows how these scenarios can be formulated using the mixed-integer framework previously

developed.

### 3.4.1 Costed Contingencies

It is possible that there could be some freedom in the choice of the nominal trajectory. For an appropriately sized set  $\tilde{\mathcal{F}}_M^{(\alpha)}(\cdot)$ , the formulation presented in the previous section will ensure only the existence of a feasible trajectory to the contingency set, whilst attempting to make the cost of the primary trajectory as small as possible. However, if the cost of the contingency trajectory is also important, this could affect the shape of the primary trajectory. The value function can be modified in two different ways to account for this.

#### Worst-case contingency cost

It is possible to minimise a weighted sum of the cost to the primary target and the worst case cost for contingency  $\alpha$ , from each state prediction. This is achieved by the value function

$$J_I^*(x(k)) = \min_{\theta} \left( \sum_{j=0}^{N(k)-1} (1 + \|\Theta(y(k+j|k) - y_r)\|) + \rho \max_{j \in \mathbb{N}_{[1, N-1]}} \sum_{i=0}^{M_j^{(\alpha)}(k)-1} \left( 1 + \left\| \tilde{\Theta}^{(\alpha)} \left( \psi_j^{(\alpha)}(i|k) - \psi_r^{(\alpha)} \right) \right\| \right) \right), \quad (3.104)$$

where  $\tilde{\Theta}^{(\alpha)}$  is an appropriately sized cost weighting matrix,  $\rho \in \mathbb{R}^+$  is the weighting on the contingency cost and  $\psi_r^{(\alpha)}$  is some contingency output reference. Assuming that 1-norms are used, this can be written in mixed-integer form as

$$\min_{\theta_b, s(k)} \sum_{j=0}^{\bar{N}-1} (jb(k+j|k) + \|\Theta(y(k+j|k) - y_r)\|_1) + \rho s(k), \quad (3.105)$$

where the slack variable  $s(k) \geq 0$  satisfies the constraint

$$s(k) \geq \sum_{i=0}^{\bar{M}^{(\alpha)}-1} \left( i\beta_j^{(\alpha)}(i|k) + \left\| \tilde{\Theta}^{(\alpha)} \left( \psi_j^{(\alpha)}(i|k) - \psi_r^{(\alpha)} \right) \right\|_1 \right), \forall j \in \mathbb{N}_{[1, \bar{N}-1]}. \quad (3.106)$$

The 1-norm in (3.106) can be represented by extra slack vectors to bound each component from above and below, and subsequently summing these together. By optimality, this will provide a tight bound.

Also by optimality,  $s(k)$  will be an upper bound only on those contingency trajectories that originate prior to completion, since all output constraints are relaxed upon completion. It can be shown that for an appropriate choice of cost weightings, the value function remains a Lyapunov-like function, and thus preserves the guarantee of finite-time completion.

Denoting the cost of the feasible solution (3.43) at time  $k + 1$  by  $\hat{J}_I(\cdot)$ , observe that

$$J_I^*(x(k)) - \hat{J}_I(x(k+1)) = J^*(x(k)) - \hat{J}(x(k+1)) + \rho \max_{j \in \mathbb{N}_{[1, N^*(k)-1]}} \sum_{i=0}^{M_j^{*(\alpha)}(k)-1} \left( 1 + \left\| \tilde{\Theta}^{(\alpha)}(\psi_j^{*(\alpha)}(i|k) - \psi_r^{(\alpha)}) \right\| \right) \quad (3.107)$$

$$- \rho \max_{j \in \mathbb{N}_{[1, N^*(k)-2]}} \sum_{i=0}^{M_{j+1}^{*(\alpha)}(k)-1} \left( 1 + \left\| \tilde{\Theta}^{(\alpha)}(\psi_{j+1}^{*(\alpha)}(i|k) + \tilde{Q}^{(\alpha)}(i)L(j)w(k) - \psi_r^{(\alpha)}) \right\| \right) \geq \eta + \rho \max_{j \in \mathbb{N}_{[1, N^*(k)-1]}} \sum_{i=0}^{M_j^{*(\alpha)}(k)-1} \left( 1 + \left\| \tilde{\Theta}^{(\alpha)}(\psi_j^{*(\alpha)}(i|k) - \psi_r^{(\alpha)}) \right\| \right) \quad (3.108)$$

$$- \rho \max_{j \in \mathbb{N}_{[1, N^*(k)-1]}} \sum_{i=0}^{M_{j+1}^{*(\alpha)}(k)-1} \left( 1 + \left\| \tilde{\Theta}^{(\alpha)}(\psi_{j+1}^{*(\alpha)}(i|k) - \psi_r^{(\alpha)}) \right\| \right) - \rho \max_{j \in \mathbb{N}_{[1, N^*(k)-2]}} \sum_{i=0}^{M_{j+1}^{*(\alpha)}(k)-1} \left\| \tilde{\Theta}^{(\alpha)} \tilde{Q}^{(\alpha)}(i)L(j)w(k) \right\| \geq \eta - \rho \left\| \tilde{\Theta}^{(\alpha)} \right\| \max_{\substack{j \in \mathbb{N}_{[1, N-2]} \\ w \in \mathcal{W}}} \sum_{i=0}^{\bar{M}^{(\alpha)}-1} \left\| \tilde{Q}^{(\alpha)}(i)L(j)w \right\|. \quad (3.109)$$

Hence, by choosing the cost weightings  $\Theta$ ,  $\tilde{\Theta}^{(\alpha)}$  and  $\rho$  such that the RHS of (3.109) is greater than zero, the finite-time completion guarantee still holds.

### Mean contingency cost

In addition to the primary trajectory cost, the mean contingency cost can also be penalised with a value function of the form

$$J_{II}^*(x(k)) = \min_{\theta} \left( \sum_{j=0}^{N(k)-1} (1 + \|\Theta(y(k+j|k) - y_r)\|) + \frac{\rho}{(N(k)-1)} \sum_{j=1}^{N(k)-1} \sum_{i=0}^{M_j^{*(\alpha)}(k)-1} \left( 1 + \left\| \tilde{\Theta}^{(\alpha)}(\psi_j^{(\alpha)}(i|k) - \psi_r^{(\alpha)}) \right\| \right) \right). \quad (3.110)$$

With the addition of extra variables, this can be cast into mixed-integer form if a 1-norm cost function is used. Define

$$\sigma_1(k) \geq \sum_{i=0}^{\bar{M}^{(\alpha)}-1} \left( i\beta_1^{(\alpha)}(i|k) + \left\| \tilde{\Theta}^{(\alpha)}(\psi_1^{(\alpha)}(i|k) - \psi_r^{(\alpha)}) \right\|_1 \right) \quad (3.111)$$

$$\sigma_j(k) \geq \sigma_{j-1}(k) + \frac{1}{j} \left( \sum_{i=0}^{\bar{M}^{(\alpha)}-1} \left( i\beta_j^{(\alpha)}(i|k) + \left\| \tilde{\Theta}^{(\alpha)}(\psi_j^{(\alpha)}(i|k) - \psi_r^{(\alpha)}) \right\|_1 \right) - \sigma_{j-1}(k) \right), \quad (3.112)$$

for  $j \in \mathbb{N}_{[2, \bar{N}-1]}$ . Then by using a value function of the form

$$\min_{\theta_b, s(k)} \sum_{j=0}^{\bar{N}-1} (jb(k+i|k) + \|\Theta(y(k+j|k) - y_r)\|_1) + \rho s(k), \quad (3.113)$$

where the slack variable  $s(k) \geq 0$  is redefined to satisfy the logical implications

$$b(k+j+1|k) = 1 \implies s(k) \geq \sigma_j(k), \forall j \in \mathbb{N}_{[1, \bar{N}-1]}, \quad (3.114)$$

the mean contingency cost is now penalised in addition to the normal operating cost. Slack vectors can again be used to replace the 1-norm in (3.111)-(3.112). In this case the optimal cost  $J_{\Pi}^*(\cdot)$  cannot be used as a Lyapunov-like function, as the mean is not, in general, non-increasing. However, it is easy to show that

$$J_{\Pi}^*(x(k)) \leq J_I^*(x(k)), \quad (3.115)$$

for any  $x(k)$ . Hence, by choosing  $\rho$  and the cost weightings to satisfy the conditions (3.109) for the worst-case contingency cost, finite time completion is also guaranteed for the mean contingency cost, as an upper bound on  $J_{\Pi}^*(\cdot)$  reduces by at least some fixed amount at each time step.

**Remark 3.16.** The cost weightings on the contingencies only consider *nominal* costs. Under disturbances, the actual closed loop cost after contingency activation will differ. This is, of course, true of the primary trajectory cost as well.

### 3.4.2 Multiple & Prioritised Contingencies

Algorithm 3.1 can be extended in a straightforward manner to ensure feasibility of multiple contingencies. To do so, extra prediction variables need to be augmented to the system. Specifically, sequences of states, inputs and outputs are required for each  $\alpha \in \mathbb{N}_{[1, n_c]}$ . Then constraints will be enforced on each of these sequences according to (3.41). Define the set  $\mathcal{C} \subseteq \mathbb{N}_{[1, n_c]}$ , indicating the indices of the contingencies that are required to be available. Then, by enforcing the constraint

$$v(\alpha, j) = 1, \forall \alpha \in \mathcal{C}, j \in \mathbb{N}_{[1, \bar{N}-1]}, \quad (3.116)$$

there must exist feasible trajectories from each primary state prediction in the event of any of these contingencies being activated. This may, of course, make finding an initial feasible solution more difficult.

Another manner in which the contingencies could be used is to ensure the availability of at least one contingency within the set  $\mathcal{C}$  at all times  $0 < k < k_s$ . By enforcing the constraints

$$\sum_{\alpha \in \mathcal{C}} v(\alpha, j) \geq 1, \forall j \in \mathbb{N}_{[1, \bar{N}-1]} \quad (3.117)$$

at least one of the contingencies in  $\mathcal{C}$  will be available at each time step. Furthermore, by modifying the cost function, it is also possible to prioritise the availability of contingencies at a given prediction step if multiple options are feasible. Hence, a contingency that makes the primary trajectory cost worse can be given a higher priority if demanded by operational requirements. The new value function is given by

$$J_{\text{III}}^*(x(k)) = \min_{\theta_b, v(\cdot, \cdot)} \sum_{j=0}^{\bar{N}-1} \left( jb(k+i|k) + \|\Theta(y(k+j|k) - y_r)\| + \sum_{\alpha \in \mathcal{C}} \gamma_\alpha v(\alpha, j) \right), \quad (3.118)$$

where  $\gamma_\alpha \in \mathbb{R}^+$  is the weighting on contingency  $\alpha$ . Note that this preserves the guarantee of finite-time completion, as the feasible solution at the next time step still has the same contingency availability, so the cost term involving the variables  $v(\cdot, \cdot)$  reduces.

### 3.4.3 State-Dependent Contingencies

It may be the case that a particular contingency is required to be available depending on the current state of the system. Given the output constraint set  $\mathcal{Y}$ , the output-admissible set of states is given by

$$\mathcal{O} = \{x \mid \exists u : Cx + Du \in \mathcal{Y}\}. \quad (3.119)$$

Now define a set  $\mathcal{A} \subseteq \mathcal{O}$ , which is a set that defines a region of the state space in which one or more contingencies are required to be available. A partition of  $\mathcal{A}$  is then specified by the sequence of sets  $\{\mathcal{A}_\alpha\}_{\alpha \in \mathcal{C}}$ , which satisfy

$$\bigcup_{\alpha \in \mathcal{C}} \mathcal{A}_\alpha = \mathcal{A} \quad (3.120)$$

$$\mathcal{A}_{\alpha_1} \cap \mathcal{A}_{\alpha_2} = \emptyset, \forall \alpha_1, \alpha_2 \in \mathcal{C}, \alpha_1 \neq \alpha_2. \quad (3.121)$$

The ideal requirement is that contingency  $\alpha$  is available if the state of the system is within  $\mathcal{A}_\alpha$ . However, in order to maintain recursive feasibility, the sets  $\mathcal{A}_\alpha$  need to be enlarged. This is due to the fact that if  $x^*(k+j+1|k) \in \mathcal{A}_{\alpha_0}$ , for some  $\alpha_0$  there is no guarantee that  $\hat{x}(k+j+1|k+1)$  will also lie in  $\mathcal{A}_{\alpha_0}$ , due to the action of the disturbance at time  $k$ . Therefore, define the enlarged sets

$$\bar{\mathcal{A}}_\alpha = \mathcal{A}_\alpha \oplus \bigoplus_{j=0}^{\bar{N}-1} L(j)\mathcal{W}. \quad (3.122)$$

Then, recursively define the sets

$$\bar{\mathcal{A}}_\alpha(0) = \bar{\mathcal{A}}_\alpha \quad (3.123)$$

$$\bar{\mathcal{A}}_\alpha(j+1) = \bar{\mathcal{A}}_\alpha(j) \ominus L(j)\mathcal{W}. \quad (3.124)$$

By now enforcing the constraints

$$x(k+j|k) \in \bar{\mathcal{A}}_\alpha(j) \implies v(\alpha, j) = 1, \quad (3.125)$$

the availability of the appropriate contingency in the MPC optimisation problem is ensured, depending on the predicted state. This is due to the fact that, from the design of sets  $\mathcal{A}_\alpha$  as a partition of the entire feasible state-space, and given that the relationship between the Pontryagin difference and Minkowski sum [KG95] ensures that  $\bar{\mathcal{A}}_\alpha(j) \supseteq \mathcal{A}_\alpha$  for all  $j \in \mathbb{Z}_{[0, \bar{N}]}$ , the system state is always guaranteed to be in at least one of the sets  $\bar{\mathcal{A}}_\alpha(j)$ . This will, however, require multiple contingencies to be available in regions of overlap.

**Remark 3.17.** The implication (3.125) is reversed from all of the others encountered so far. Extra binary variables are needed to represent this reverse implication. If the sets can be represented by  $r$  half space constraints as

$$\bar{\mathcal{A}}_\alpha(j) = \left\{ x \mid G^{(\alpha)}(j)x \leq h^{(\alpha)}(j), G(j) \in \mathbb{R}^{r \times n}, h(j) \in \mathbb{R}^{n \times 1} \right\}, \quad (3.126)$$

by introducing the binary vector  $\delta^{(\alpha)}(j) \in \{0, 1\}^{r \times 1}$ , the implication can be expressed in mixed-integer form as

$$G^{(\alpha)}(j)x \geq f^{(\alpha)}(j) - M(1 - \delta^{(\alpha)}(j)) \quad (3.127)$$

$$1 - v(\alpha, j) \leq \sum_{i=1}^r \delta_i^{(\alpha)}(j), \quad (3.128)$$

for all  $j \in \mathbb{Z}_0^{\bar{N}-1}$ , where  $\delta_i^{(\alpha)}(j)$  denotes the  $i$ th binary element of  $\delta^{(\alpha)}(j)$  and  $M$  is a suitably large number.

Recursive feasibility of the resulting optimisation problem is then readily apparent, as

$$x^*(k+j+1|k) \in \bar{\mathcal{A}}_\alpha(j+1) \implies x^*(k+j+1|k) + L(j)\mathcal{W} = \hat{x}(k+j+1|k+1) \in \bar{\mathcal{A}}_\alpha(j). \quad (3.129)$$

Hence, a shifted version of the optimal solution at time  $k$  with the disturbance feedback term added still requires the same contingency availability. This essentially means that the disturbance feedback term doesn't cause the state to "jump" into another region where a different contingency is required to be feasible, or worse still, move into the set  $\mathcal{A}$  from being outside it.

**Remark 3.18.** It is possible to combine costed contingencies with multiple contingency availability and state dependent contingencies, through appropriate addition of prediction variables and modification of the cost function.

## 3.5 Simplifications

The formulation presented in the previous two sections uses additional prediction variables for contingency dynamics, as well as integer variables for indicating completion on the primary and contingency trajectories. This can become very computationally expensive if there are many possible contingencies. There are three simplifications that can be applied to reduce complexity.

### 3.5.1 Fixed Contingency Horizon

By assuming a fixed horizon on the contingency trajectories, the need for binary variables on these trajectory predictions is eliminated. However, this may make it more difficult to find an initially feasible solution, and will produce a more conservative primary trajectory. A fixed contingency horizon may be appropriate when the contingency terminal set  $\tilde{\mathcal{T}}_0^{(\alpha)}$  is robust control invariant and upon contingency activation, the control problem is to remain within the contingency terminal set or stabilise to a setpoint within this set.

### 3.5.2 Explicit Contingency Feasibility

If a fixed horizon is used, it is possible to further simplify the problem by explicitly calculating the set  $\tilde{\mathcal{K}}_M^{(\alpha)}(\tilde{\mathcal{T}}^{(\alpha)})$ , assuming that the target set after contingency activation is to be entered in  $M^{(\alpha)}$  steps. For the variable horizon case, it is possible to calculate the set  $\tilde{\mathcal{F}}_M^{(\alpha)}(\tilde{\mathcal{T}}^{(\alpha)})$ , but this will, in general, be non convex, as it will be the union of the controllable sets for each possible horizon length after contingency activation. As each state on the primary trajectory only has to be in one of these sets, binary variables will be needed to enforce the constraints.

It may be asked why the online solution is ever required if the robust controllable sets for contingency feasibility can be computed. After all, even with explicit variable horizon contingencies, the resulting mixed-integer program contains fewer decision variables. However, without parameterising the contingency trajectories, it is not possible to place costs on them and incorporate these into the optimisation problem. Furthermore, with non-convex collision avoidance constraints, many sets will need to be calculated for every combination of avoidance constraints over the whole prediction horizon. Finally, the contingency predictions provide a feasible solution from Theorem 3.12 that can be applied in the case of optimiser failure under contingency activation.

### 3.5.3 Prediction Variable Reduction

When considering the availability of at least one contingency as enforced by the condition (3.117), it may be the case that all contingencies in  $\mathcal{C}$  have the same dynamics and operating constraints, but different terminal sets. In this case, only a single set of prediction variables is needed for the states, inputs and outputs under contingency activation. The binary variables indicating completion cannot be reduced, however. They are still required to distinguish completion amongst all of the contingency target sets.

## 3.6 Examples

Examples illustrating the different contingency scenarios are now presented. A simple double-integrator system is introduced first, with a contingency requirement to provide fault-tolerance. This system is used to demonstrate the effect of costed, prioritised and state-based contingencies.

A cart-pendulum is introduced next, with an emergency stop contingency. It demonstrates the feasible contingency concept applied to a setpoint regulation problem, with a fixed contingency horizon.

### 3.6.1 Double-Integrator Vehicle

Consider a unit point mass moving in two dimensions, actuated by orthogonal forces in the horizontal and vertical directions. Treating the mass as a fictitious vehicle, fuel usage is quantified as the integral of the absolute actuator forces over time. Hence, there are two inputs representing the force components and five states representing the position, velocity and fuel. In order to measure the fuel usage, the inputs are split to capture the positive and negative components separately, giving four inputs in the resulting system, constrained to be non-negative.

Discretising with a sampling frequency of 1 Hz gives the state-space matrices

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \quad (3.130)$$

The  $C$  and  $D$  matrices are chosen to output all states and feed through all inputs of the system, that is, such that  $y = (x, u)$ . In addition to the positivity constraints on each individual input, in order to restrict the maximum actuation, the inputs satisfy the operating constraints

$$u(k) \in \mathcal{U} = \{u \mid |u| < 1\}, \quad (3.131)$$

where  $|\cdot|$  indicates the row-wise absolute value. The states satisfy the constraints

$$x(k) \in \mathcal{X} = \left\{ x \mid \begin{bmatrix} -20 & -20 & -2.5 & -2.5 & 0 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 20 & 20 & 2.5 & 2.5 & 10 \end{bmatrix}^T \right\}, \quad (3.132)$$

which restrict the maximum speed in each direction to 2.5 and ensure that the fuel level is always non-negative. The remaining constraints on position and an upper bound on fuel use simply ensure that the set  $\mathcal{X}$  is a polytope. The system is assumed to start with 8 units of fuel, which limits the overall use of control authority (or impulse applied to the vehicle).

Using VH-MPC, define the value function as

$$J^*(x(k)) = \min_{u(k), N(k)} \sum_{j=0}^{N(k)-1} 1 + 0.1 \|u(k+j|k)\|_1. \quad (3.133)$$

This function can be expressed in the form (3.40) by an appropriate choice of  $\Theta$  and  $y_r$ . The

primary target is now defined by

$$\mathcal{T} = \left\{ x \mid \begin{bmatrix} -1 & 8 & -2.5 & -2.5 & 0 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 1 & 10 & 2.5 & 2.5 & 10 \end{bmatrix}^T \right\}. \quad (3.134)$$

This set restricts the position of the vehicle to a translated square of side length 2 and ensures that the velocity components in each coordinate direction have a maximum magnitude of 2.5. It also ensures that the fuel is non-negative upon entry. The upper bound on the fuel is again to ensure that  $\mathcal{T}$  is a polytope.

The system is known to be vulnerable to a certain fault, where it begins to leak fuel at the rate of 10% of its current level at each time step. At the same time, it requires 10% more fuel to produce an equivalent force in the  $y$  actuator. This gives the new system dynamics defined by matrices

$$\tilde{A} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.9 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & -1 & -1.1 & -1.1 \end{bmatrix}. \quad (3.135)$$

When this fault occurs, the system is required, within 4 time steps, to reach another target set, subject to the same input and state constraints as under normal operation. The contingency set is given by

$$\tilde{\mathcal{T}}^{(1)} = \left\{ \phi \mid \begin{bmatrix} 4 & 3.5 & -2.5 & -2.5 & 0 \end{bmatrix}^T \leq \phi \leq \begin{bmatrix} 6 & 5.5 & 2.5 & 2.5 & 10 \end{bmatrix}^T \right\}, \quad (3.136)$$

which is another square, subject to the same velocity and fuel constraints as during normal operation. To test the efficacy of robust contingency availability, or robust fault-tolerance, a wind disturbance force, defined by the set

$$\mathcal{W} = \left\{ w \mid |w| \leq \begin{bmatrix} 0 & 0 & 0.2 & 0.2 & 0 \end{bmatrix}^T \right\}, \quad (3.137)$$

acts on the system both during normal operation and under fault.

Robustness is ensured by tightening these constraint sets according to (3.34) - (3.37). The process is simplified by choosing nilpotent disturbance feedback policies for constraint tightening, which greatly reduces the number of sets that need to be calculated. It can be verified that, for the system operating normally (3.130), the policy

$$P(j) = \begin{cases} \begin{bmatrix} -0.5 & 0 & -0.75 & 0 & 0.25 \\ 0.5 & 0 & 0.75 & 0 & 0.25 \\ 0 & -0.5 & 0 & -0.75 & 0.25 \\ 0 & 0.5 & 0 & 0.75 & 0.25 \end{bmatrix} & j = 0 \\ \begin{bmatrix} -0.5 & 0 & -0.75 & 0 & 0 \\ 0.5 & 0 & 0.75 & 0 & 0 \\ 0 & -0.5 & 0 & -0.75 & 0 \\ 0 & 0.5 & 0 & 0.75 & 0 \end{bmatrix} & j = 1 \\ 0_{4 \times 5} & j > 1, \end{cases}$$

renders the system degree-2 nilpotent, meaning that  $A^2 + ABP(0) + BP(1) = 0$ . For the system

under fault, the policy

$$\tilde{P}^{(1)}(i) = \begin{cases} \begin{bmatrix} -0.5 & 0 & -0.75 & 0 & 0.2036 \\ 0.5 & 0 & 0.75 & 0 & 0.2036 \\ 0 & -0.5 & 0 & -0.75 & 0.2240 \\ 0 & 0.5 & 0 & 0.75 & 0.2240 \end{bmatrix} & i = 0 \\ \begin{bmatrix} -0.5 & 0 & -0.75 & 0 & 0 \\ 0.5 & 0 & 0.75 & 0 & 0 \\ 0 & -0.5 & 0 & -0.75 & 0 \\ 0 & 0.5 & 0 & 0.75 & 0 \end{bmatrix} & i = 1 \\ 0_{4 \times 5} & i > 1, \end{cases}$$

also gives degree-2 nilpotency.

Simulations of the controlled, discretised system are performed over the same 20 random sequences of allowable disturbances, for the various contingency scenarios detailed below.

### Single Contingency

Figure 3.2 shows the trajectories under normal operation and anticipated contingency activation at each time step. The cost function used once the contingency is activated takes the same form as (3.133), but has a sufficiently small cost weighting on the inputs (0.05) to ensure that completion does occur within four steps. On the contingency trajectories, the '+' symbols indicate the time steps.

It is clear that the distribution of trajectories entering the primary target is skewed towards the right, illustrating the effect of requiring robust reachability of  $\tilde{\mathcal{T}}^{(1)}$  from any state prediction. Figures 3.3 and 3.4 show the speeds and actuator forces respectively. These quantities satisfy the imposed constraints, indicated by the dashed lines, in the presence of the disturbance force. The fuel use is displayed in Figure 3.5.

### Costed Contingency

Using the same system and contingency requirement, a weighting can be placed on the mean contingency cost as in (3.110). Defining the contingency cost as above, the mean cost over all contingency predictions is weighted by 1.5 and added to the primary trajectory cost. This essentially gives preference to contingency trajectories that complete sooner than the four-step maximum. The effect of applying this weighting is evident in Figure 3.6. It can be seen that the primary trajectories have now moved closer to the contingency set.

### Multiple Prioritised Contingencies

Define a second contingency target set as

$$\tilde{\mathcal{T}}^{(2)} = \left\{ \phi \mid \begin{bmatrix} 4 & 6.5 & -2.5 & -2.5 & 0 \end{bmatrix}^T \leq \phi \leq \begin{bmatrix} 6 & 8.5 & 2.5 & 2.5 & 10 \end{bmatrix}^T \right\}. \quad (3.138)$$

The system is now required to reach either  $\tilde{\mathcal{T}}^{(1)}$  or  $\tilde{\mathcal{T}}^{(2)}$  if the fault occurs. Applying no priorities on the two contingency target sets gives the trajectories shown in Figure 3.7. When the original

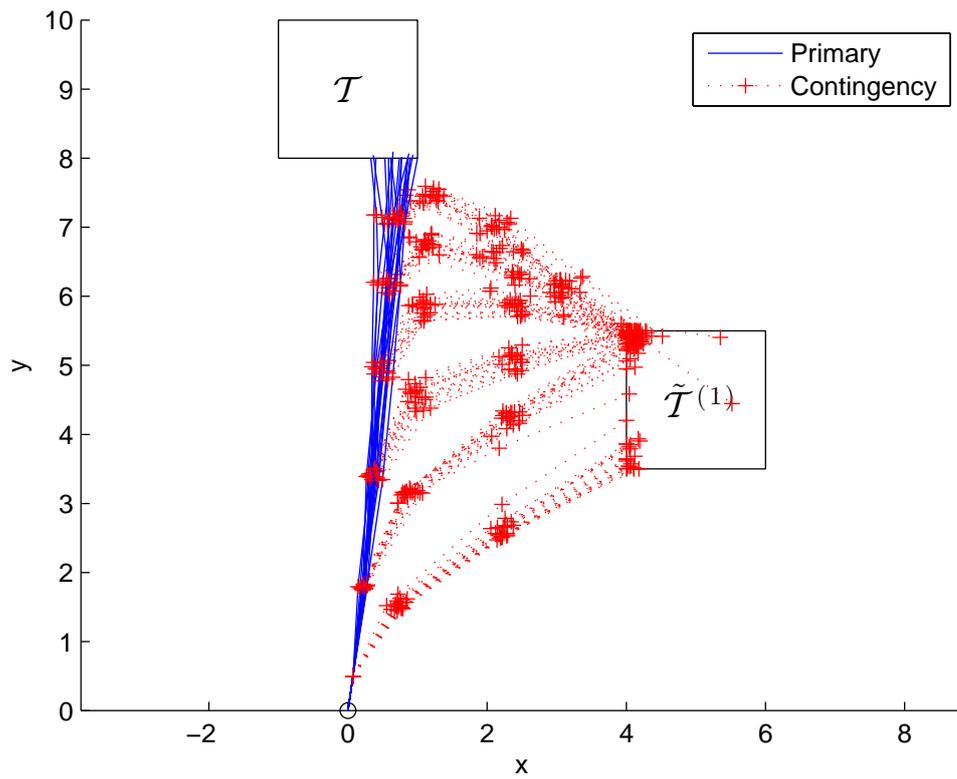


Figure 3.2: Trajectories with contingency

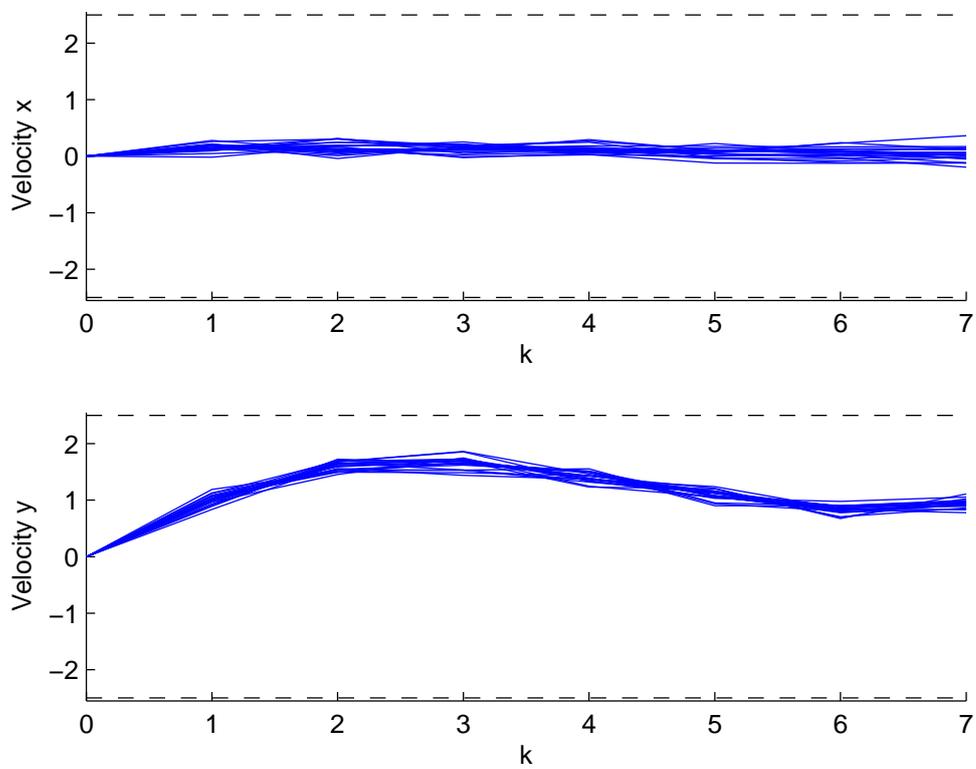


Figure 3.3: Primary trajectory velocities

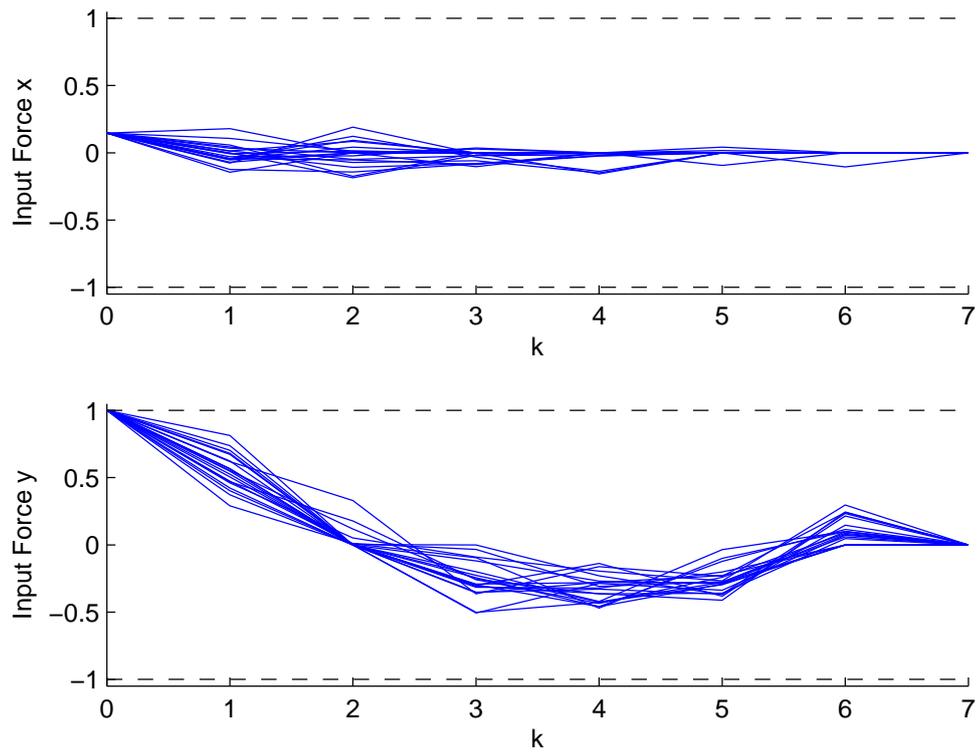


Figure 3.4: Primary trajectory control inputs

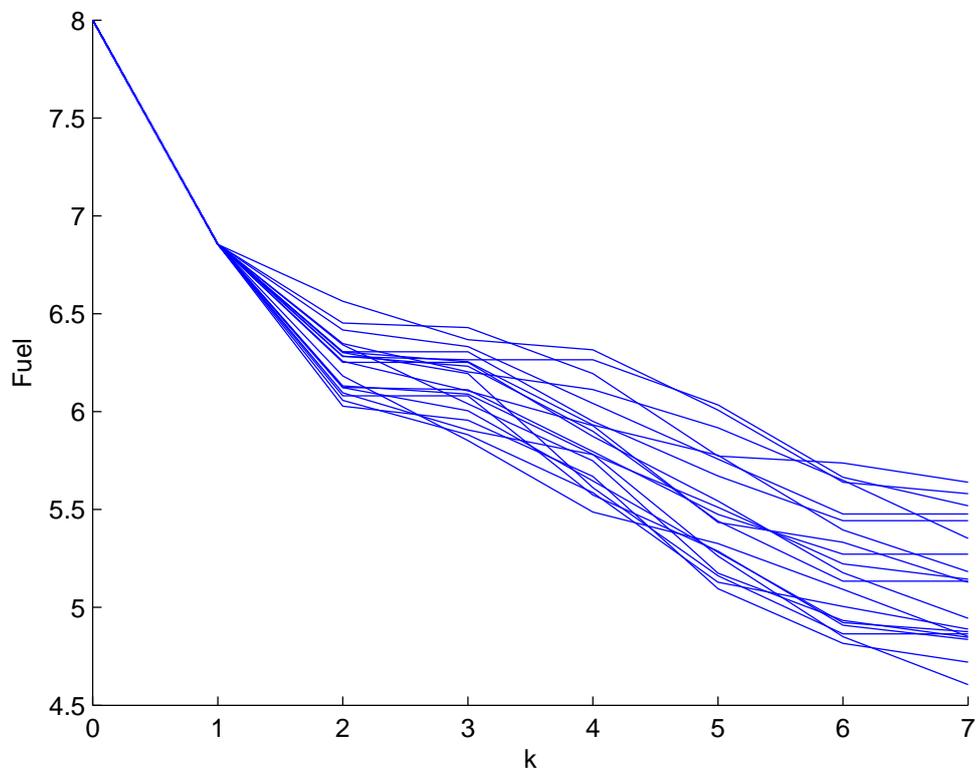
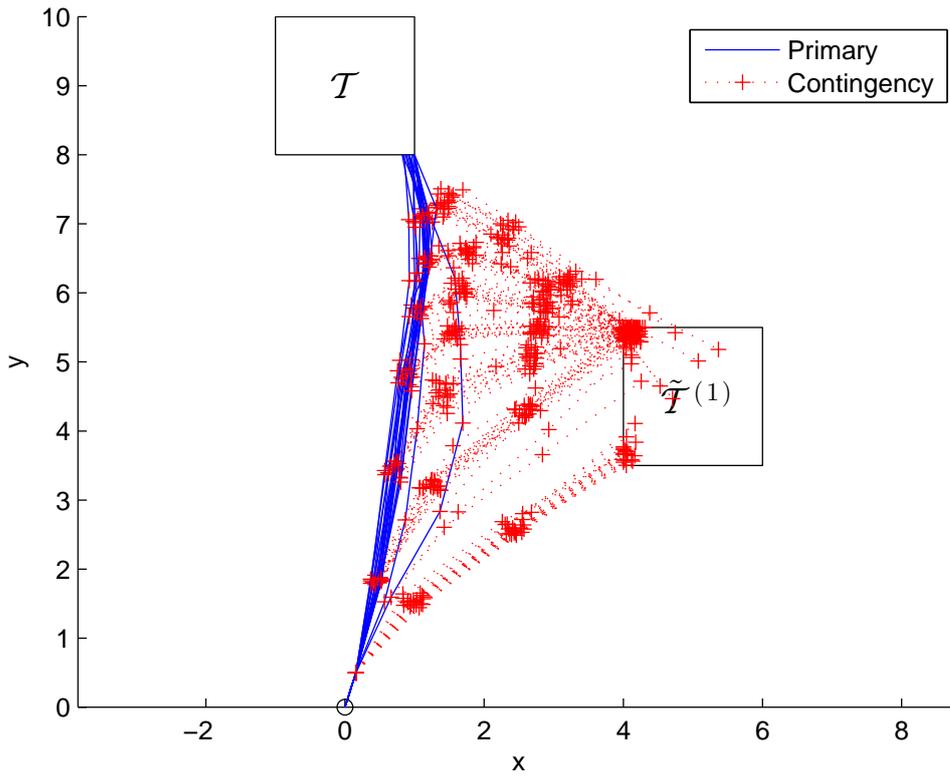


Figure 3.5: Primary trajectory fuel use



**Figure 3.6:** Trajectories with mean-costed contingency

contingency has a smaller cost weighting placed on it compared to the new contingency using (3.118), contingency completion in this set is now preferred. This is illustrated in Figure 3.8.

### State-dependent Contingencies

To clearly demonstrate the effect of a state-dependent contingency, define a new contingency target set as

$$\tilde{\mathcal{T}}^{(3)} = \left\{ \phi \mid \begin{bmatrix} 8 & 3.5 & -2.5 & -2.5 & 0 \end{bmatrix}^T \leq \phi \leq \begin{bmatrix} 10 & 5.5 & 2.5 & 2.5 & 10 \end{bmatrix}^T \right\}. \quad (3.139)$$

Assuming the same fault dynamics, the system is now required to reach  $\tilde{\mathcal{T}}^{(3)}$  in the event of the fault occurring, but only if the system state lies within the set

$$\mathcal{A} = \left\{ x \in \mathcal{O} \mid \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x \leq 4 \right\}, \quad (3.140)$$

which is the set of all output-admissible states such that the the y coordinate is less than 4. After expanding this set appropriately and applying the extra constraints (3.125), the system is once again simulated over 20 admissible disturbance realisations. As shown in Figure 3.9, the trajectories initially deviate to the right, to guarantee that  $\tilde{\mathcal{T}}^{(3)}$  can be robustly reached. Once this requirement is lifted, they moves back towards the left. Figure 3.10 shows how fuel use is restricted whilst the contingency is required to be available.

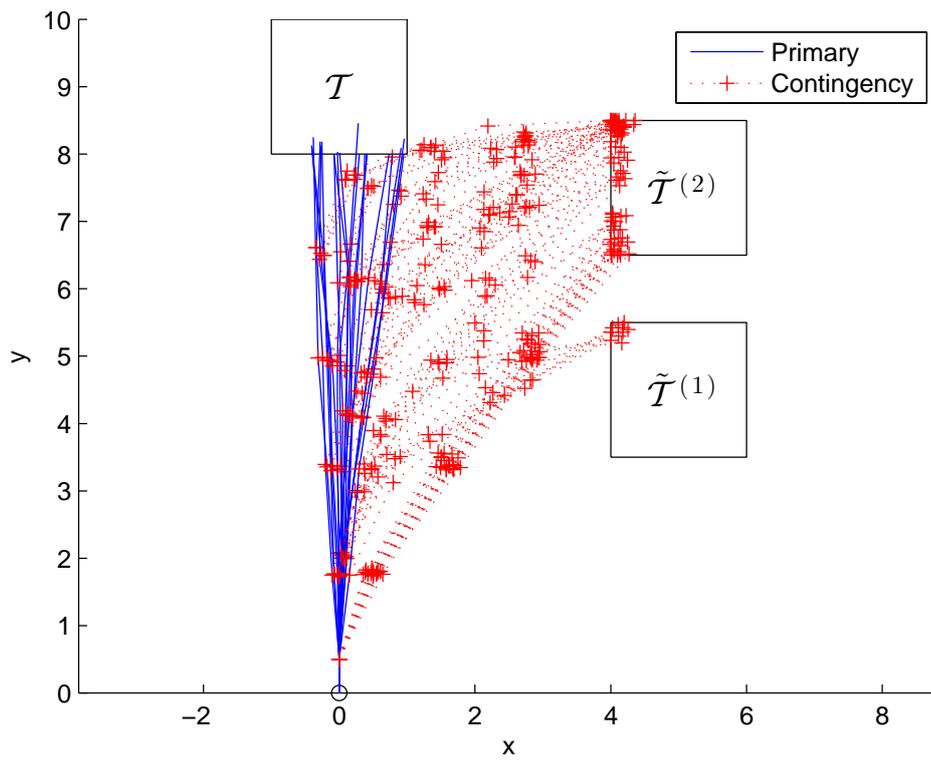


Figure 3.7: Trajectories for two contingency sets

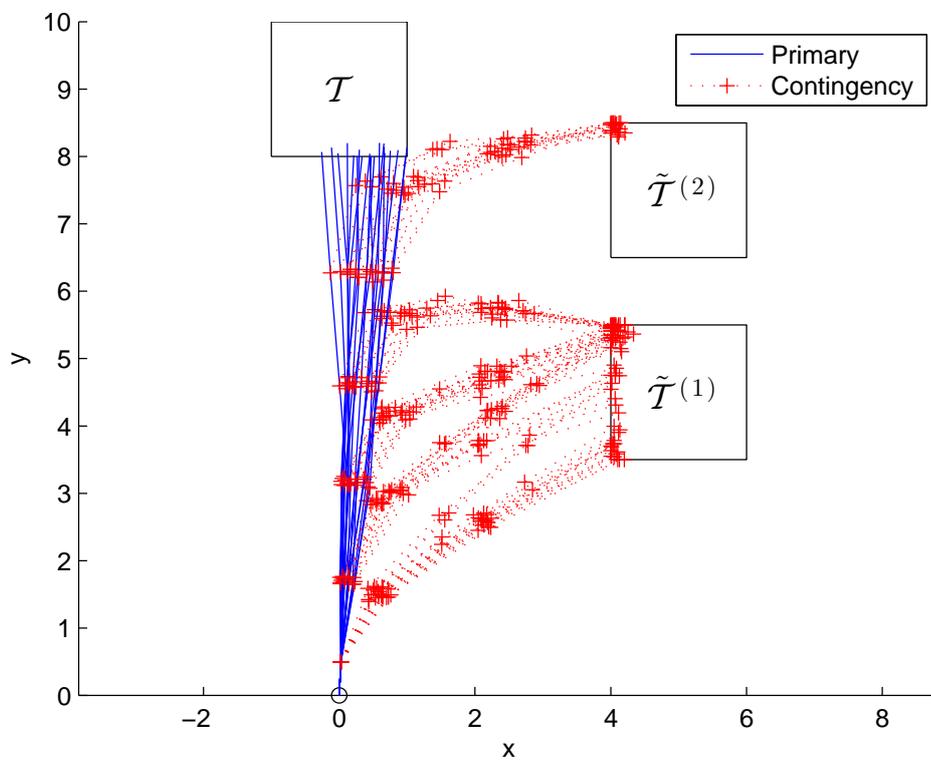


Figure 3.8: Trajectories for two contingency sets with priority

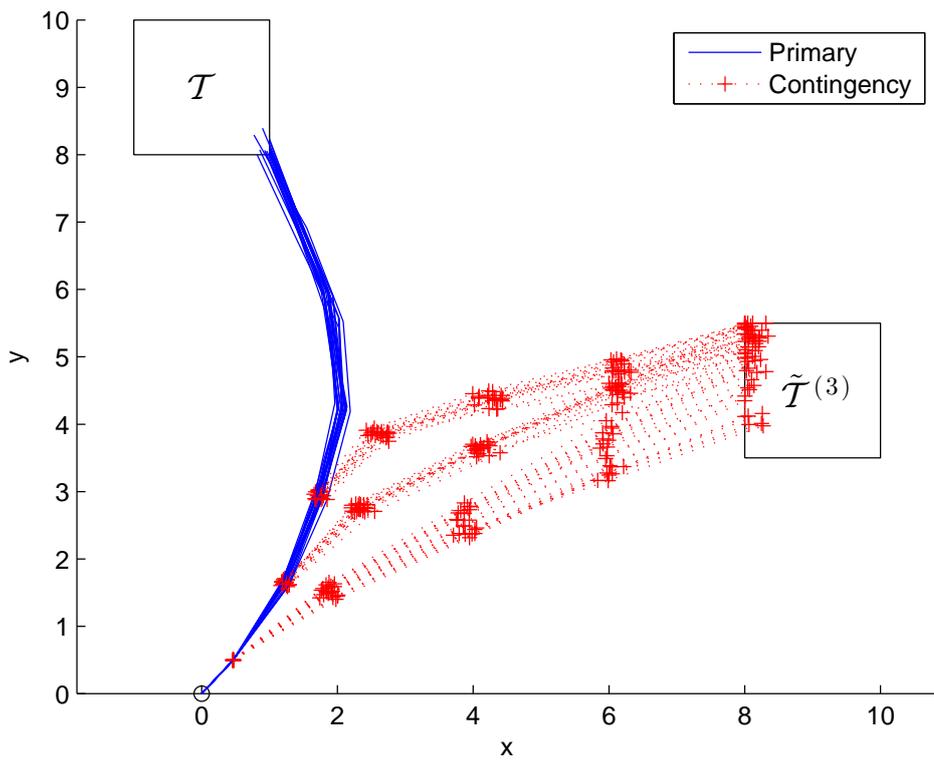


Figure 3.9: Trajectories for state-dependent contingency

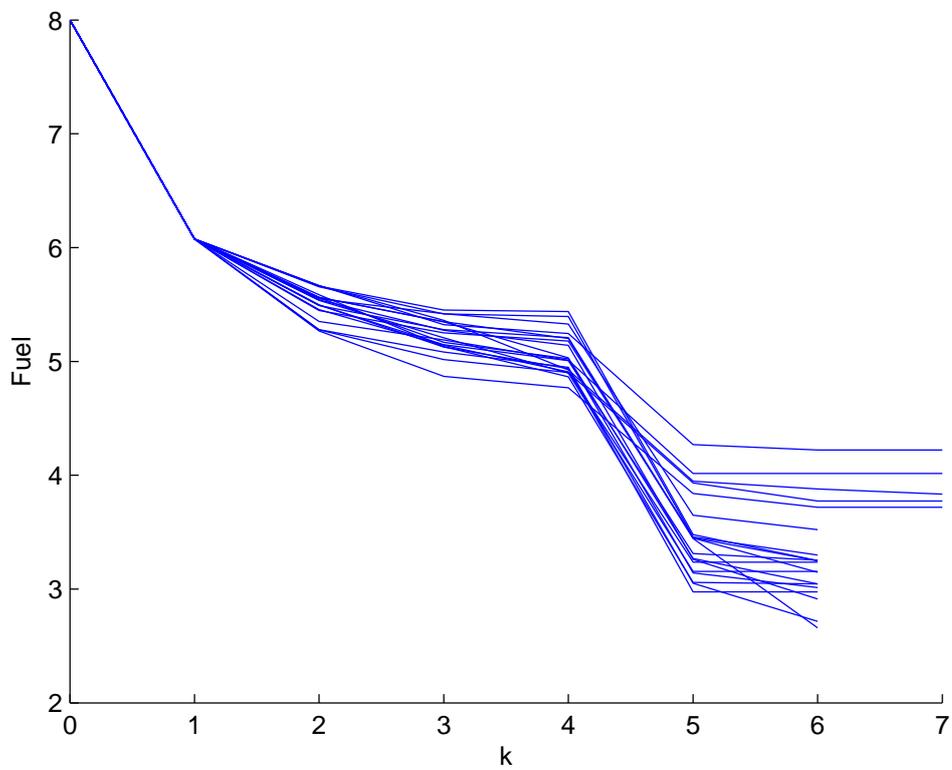
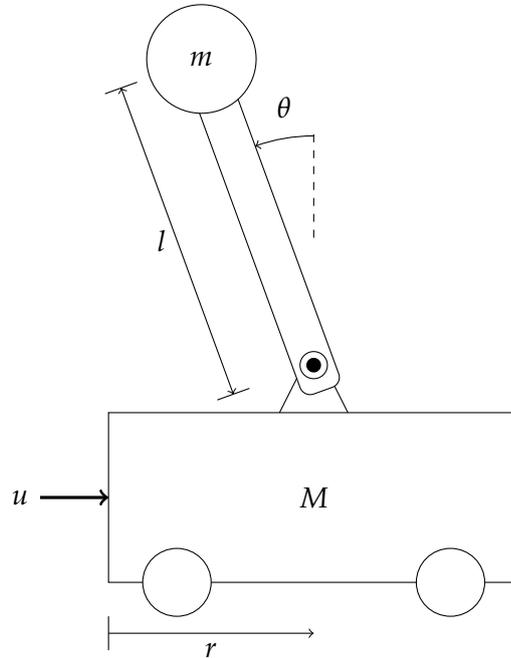


Figure 3.10: Fuel use for state-dependent contingency

### 3.6.2 Cart-Pendulum

To demonstrate the effect of contingency behaviour on a setpoint regulation problem, consider the idealised cart-pendulum shown in Figure 3.11. Given the applied force  $u$ , cart mass  $M$ ,



**Figure 3.11:** Diagram of cart-pendulum system

pendulum mass  $m$ , pendulum length  $l$ , the cart position  $r$  and pendulum angle  $\theta$ , the system satisfies the nonlinear differential equations,

$$(M + m \sin^2 \theta) \dot{r} + ml \dot{\theta}^2 \sin \theta - mg \sin \theta \cos \theta = u \quad (3.141)$$

$$((M + m)l - ml \cos^2 \theta) \ddot{\theta} + ml \dot{\theta}^2 \sin \theta \cos \theta - (M + m)g \sin \theta = u \cos \theta. \quad (3.142)$$

Linearising about  $\theta = 0$  and converting to state-space form gives the system

$$\begin{bmatrix} \dot{r} \\ \dot{\theta} \\ \ddot{r} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m}{M}g & 0 & 0 \\ 0 & \frac{M+m}{Ml}g & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{M} \\ \frac{1}{Ml} \end{bmatrix} u. \quad (3.143)$$

Taking  $M = m = 1$  kg,  $l = 1$  m and  $g = 9.81$  m s<sup>-2</sup>, this linearised system is discretised with a sampling period  $T_s = 0.2$  s, producing the discrete-time system with state space matrices,

$$A = \begin{bmatrix} 1 & 0.2094 & 0.2 & 0.0136 \\ 0 & 1.4187 & 0 & 0.2272 \\ 0 & 2.2289 & 1 & 0.2094 \\ 0 & 4.4578 & 0 & 1.4187 \end{bmatrix}, B = \begin{bmatrix} 0.0207 \\ 0.0213 \\ 0.2136 \\ 0.2272 \end{bmatrix}, \quad (3.144)$$

having linearised state vector  $x(k) = [r(kT_s) \ \theta(kT_s) \ \dot{r}(kT_s) \ \dot{\theta}(kT_s)]^T$  and input vector  $u(k) = u(kT_s)$ .

An LQR controller is designed to stabilise the pendulum about  $(r, \theta) = (r_r, 0)$ , where  $r_r$  is some reference position. The gain matrix is given by

$$K_f = [2.0245 \quad -41.3743 \quad 3.1303 \quad -10.6857]. \quad (3.145)$$

The system is subject to the pointwise-in-time state constraints (with respect to the reference)

$$x(k) - x_r \in \mathcal{X} = \left\{ x \mid \begin{bmatrix} -10 & -0.8 & -0.5 & -0.5 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 10 & 0.8 & 0.5 & 0.5 \end{bmatrix}^T \right\}, \quad (3.146)$$

where  $x_r = [r_r \ 0 \ 0 \ 0]^T$ , and input constraints

$$u(k) \in \mathcal{U} = \{u \mid \|u\|_\infty \leq 4.0\}. \quad (3.147)$$

These constraints are tightened by a small margin defined by the set

$$\mathcal{W} = \left\{ w \mid |w| \leq \begin{bmatrix} 0.001 & 0.001 & 0.0001 & 0.0001 \end{bmatrix}^T \right\} \quad (3.148)$$

to help mitigate the effect of linearisation error. Taking a 15-step prediction horizon, a 15-step nilpotent policy is used for tightening constraints. Given that this policy is nilpotent, the control objective is to steer the system state to the set  $\mathcal{R}$ , which is the largest nominally control invariant set with respect to  $K_f$  inside the terminal set given by

$$\mathcal{T} = \left\{ x \mid |x - x_r| \leq \begin{bmatrix} 0.1 & 0.5 & 0.5 & 0.5 \end{bmatrix}^T \right\}. \quad (3.149)$$

That is, for all  $x \in \mathcal{R} \subseteq \mathcal{T}$ ,

$$(A + BK_f)(x - x_r) + x_r \in \mathcal{R} \quad (3.150)$$

$$K_f(x - x_r) \in \mathcal{U}. \quad (3.151)$$

Then, a VH-MPC controller will be used for transient control and once the terminal set is entered, the controller  $K_f$  will be applied, which is a dual mode MPC scheme. The value function takes the form

$$J^*(x(k)) = \min_{u^{(k), N(k)}} \sum_{j=0}^{N(k)-1} 1 + 0.01 \|u(k)\|_1. \quad (3.152)$$

During the transition to the terminal set, a three-step fixed horizon emergency stop contingency is required at each sampling instant. This contingency must keep the pendulum balanced at the end of the emergency-stop manoeuvre. Using the same disturbance set and constraint tightening policy, the contingency requires the pendulum state to be steered into the largest

robust control invariant set  $\tilde{\mathcal{R}}^{(1)}$  with respect to  $K_f$  and the disturbance  $L(2)\mathcal{W}$  that lies within the contingency terminal set

$$\mathcal{T}^{(1)} = \left\{ \phi \mid |\phi - \phi_s| \leq [10 \ 0.1 \ 0.05 \ 0.05]^T \right\}, \quad (3.153)$$

where  $\phi_s = [r_s \ 0 \ 0 \ 0]^T$  is any stopping reference state and the stopping position  $r_s$  satisfies  $|r_s - r_r| \leq 10$ . This allows the cart at the time of contingency completion to be arbitrarily positioned within a specified range of stopping positions relative to the desired cart position. The contingency invariant set therefore ensures that, for all  $\phi \in \tilde{\mathcal{R}}^{(1)} \subseteq \mathcal{T}^{(1)}$ , there exists some  $r_s$  such that

$$(A + BK_f)(\phi - \phi_s) + L(2)\mathcal{W} + \phi_s \in \tilde{\mathcal{R}}^{(1)} \quad (3.154)$$

$$K_f(\phi - \phi_s) \in \mathcal{U}. \quad (3.155)$$

This set places a more stringent terminal requirement on the pendulum angle and angular velocity, as might be expected with an emergency stop constraint. Then, starting with the zero state, a step input is used as the reference (i.e.  $r_r = 1$ ). The state trajectories for the nonlinear pendulum dynamics with and without the imposition of contingency availability are shown in Figures 3.12 and 3.13, for the cart and pendulum respectively. The control inputs are displayed in Figure 3.14. Zero order hold discretisation is applied to the continuous states which are then fed into the MPC controller.

**Remark 3.19.** The set  $\tilde{\mathcal{R}}^{(1)}$  ensures that there exists some  $r_s$  at which the cart can be stabilised. To find a suitable value for this position upon contingency activation, a quadratic program can be solved to determine the  $\phi_s$  closest to the terminal predicted contingency state at the current time.

It can be seen that the contingency requirement produces a vastly different closed-loop behaviour. The only signal that approaches a constraint boundary is the cart velocity without contingency, and this constraint is respected in spite of the nonlinearity as a result of the tightening applied. It can be seen that when the contingency is required to be available, the cart velocity, pendulum angular velocity and input force are all reduced.

### 3.7 Conclusion

This chapter has introduced the notion of feasible contingencies and specified the criterion for a contingency to be robustly feasible. It has demonstrated how a variable horizon controller can be designed for linear systems to guarantee pointwise-in-time robust contingency availability. This controller has been shown to preserve the guarantees of robust recursive feasibility and robust finite-time completion with an appropriate choice of cost weightings. The basic formulation has then been extended to show how these guarantees can be maintained with costed,

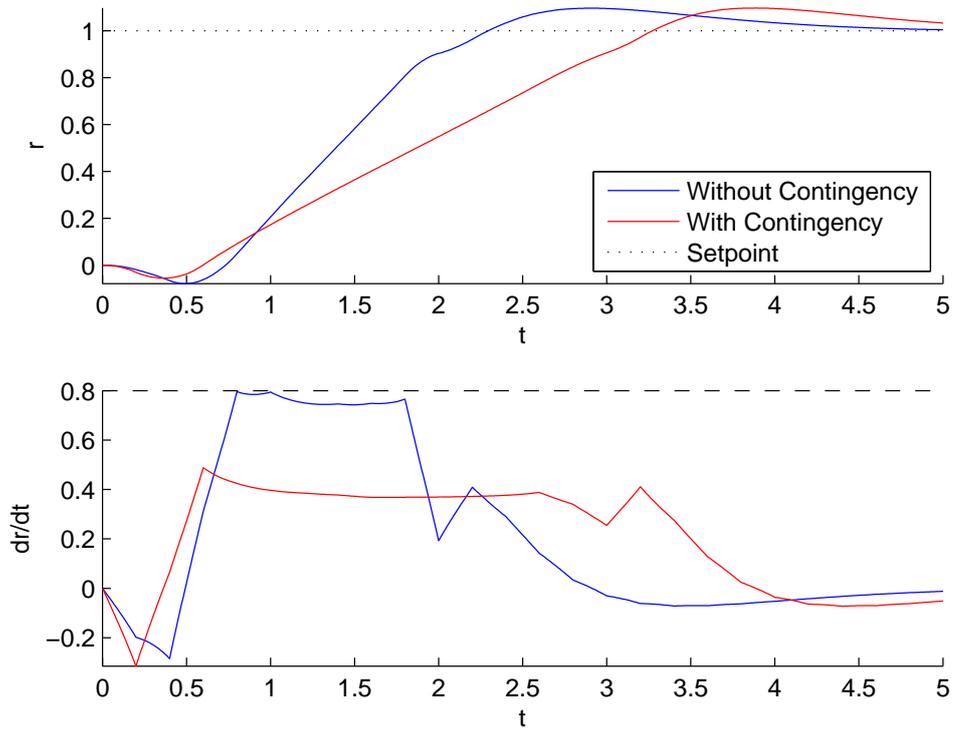


Figure 3.12: Step response for cart states

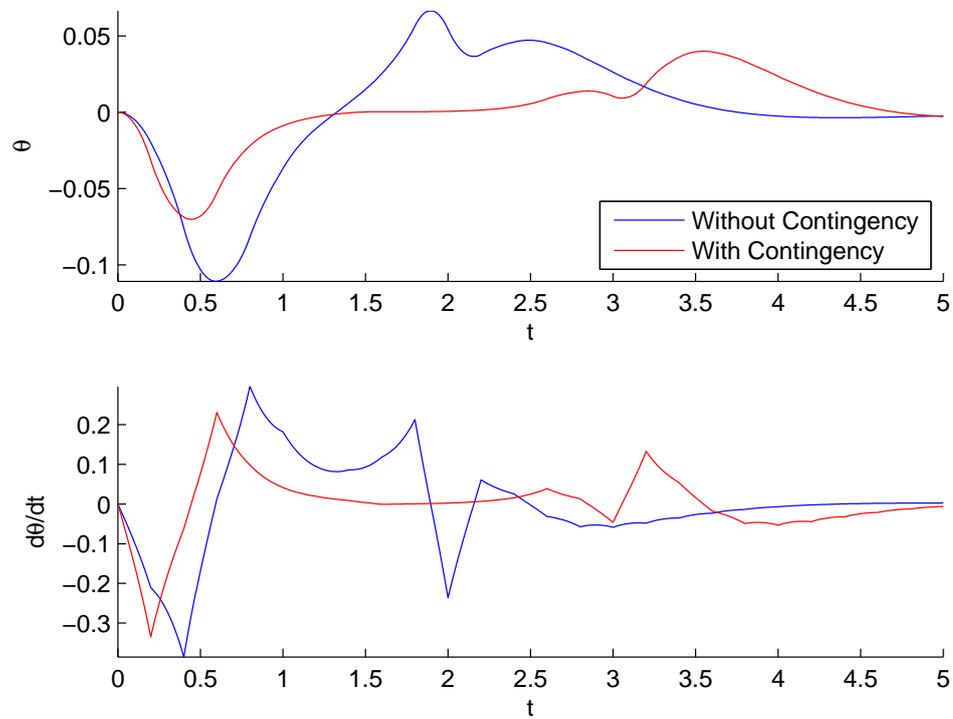
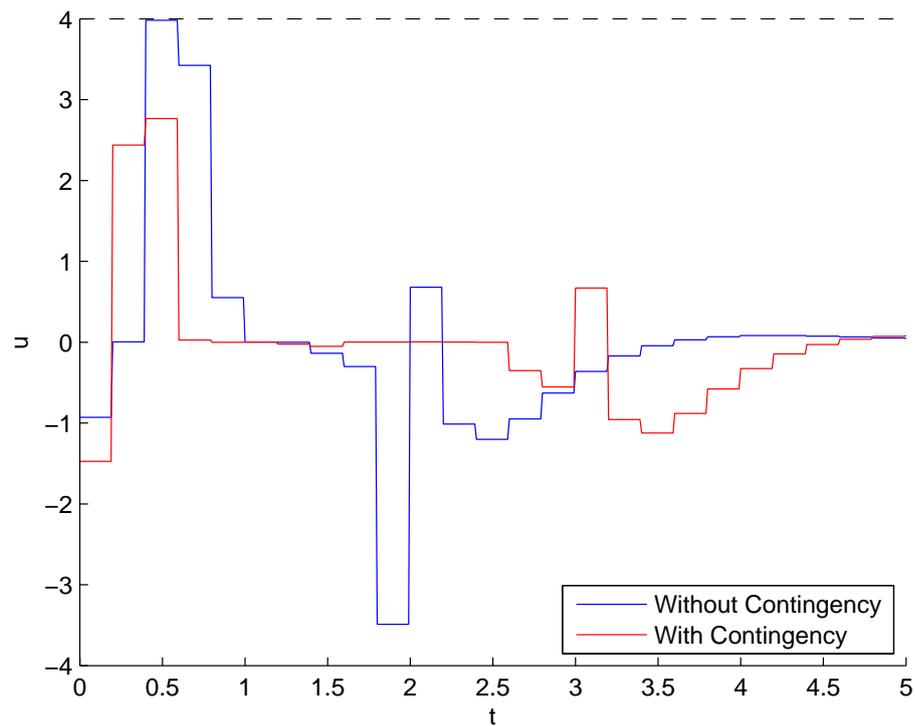


Figure 3.13: Step response for pendulum states



**Figure 3.14:** Control input to cart

multiple, prioritised and state-dependent contingencies. Two example systems demonstrate the power of the formulation.

Future work will consider the issue of further contingency activations along the contingency trajectories, as well as techniques for reducing the computational complexity of the approach, which depends on the use of mixed-integer programming in implementation. For the latter, this could include the use of move blocking strategies described in Chapter 4, or intelligently reducing the predicted states at which contingency availability is enforced.

### 3.8 Acknowledgements

This chapter builds upon a paper presented by the author at the 18th IFAC World Congress [SM11]. The simulations in this chapter were carried out using MATLAB R2011b, YALMIP [Löf04], CPLEX [CPL09] and the Invariant Set Toolbox [Ker00].



## Chapter 4

# Move Blocking for Complexity Reduction

### 4.1 Introduction

A major issue with Variable Horizon MPC, as mentioned in Chapter 2, is that the cost function optimised at each time step is not, in general, a convex function of the horizon length. Recall that there are two common strategies for achieving global optimality in spite of this non-convexity: evaluating multiple fixed horizon problems at each iteration or solving a mixed-integer programming problem. With both methods, a bound must be placed on the maximum horizon length to ensure a finite number of decision variables. However, this length must be long enough to encompass the entire control manoeuvre to completion, potentially requiring a large number of decision variables for a long range control problem.

Move blocking [Mac02; Cag+07] presents a way of reducing the number of decision variables, by curtailing the number of degrees of freedom in the MPC optimisation problem. It achieves this by constraining groups of adjacent-in-time input decision variables to have the same value. The manner in which such constraints are applied over time is termed the *blocking regime*. In order to retain recursive feasibility guarantees, the blocking regime must be time-varying [Cag+07], or additional constraints need to be enforced on the first prediction step [GI07; GIK09].

In this chapter, a new formulation of VH-MPC is presented, utilising time-varying move blocking to reduce complexity. It allows flexibility in the choice of blocking regime and, through the use of tightened constraints, guarantees robust recursive feasibility and finite-time completion in the presence of bounded disturbances. By using constraint tightening, the formulation is, in general, computationally simpler than solving for blocked affine feedback policies as outlined in [Old+09].

The contents of the chapter are structured as follows. Section 4.2 poses the control problem for a linear discrete-time system. Section 4.3 introduces move blocking notation and various utility

functions. The controller formulation is described in Section 4.4. An example move blocking regime and simulation results are presented in Section 4.6.

### 4.1.1 Nomenclature

Define the horizontal concatenation operator for matrices  $A(i)$  having the same row dimension as

$$\bigoplus_{i=a}^b A(i) = \begin{bmatrix} A(a) & A(a+1) & \cdots & A(b) \end{bmatrix}, \quad (4.1)$$

where  $a, b \in \mathbb{Z}$ ,  $a \leq b$ . Also, define the *direct* sum of matrices  $C \in \mathbb{R}^{n \times m}$  and  $D \in \mathbb{R}^{p \times q}$  by

$$C \oplus D = \begin{bmatrix} C & 0_{n \times q} \\ 0_{p \times m} & D \end{bmatrix}. \quad (4.2)$$

Note that this symbol has changed meaning from the previous chapter, where it represented the Minkowski sum.

## 4.2 Problem Definition

Consider once again a discrete-time linear system with states  $x(k)$ , inputs  $u(k)$  and outputs  $y(k)$  specified by the difference equation

$$x(k+1) = Ax(k) + Bu(k) \quad (4.3)$$

$$y(k) = Cx(k) + Du(k), \quad (4.4)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$  and  $D \in \mathbb{R}^{p \times m}$ . It is assumed that the pair  $(A, B)$  is controllable. The system is subject to pointwise-in-time input and state constraints of the form

$$u(k) \in \mathcal{U} \subset \mathbb{R}^m \quad (4.5)$$

$$x(k) \in \mathcal{X} \subset \mathbb{R}^n, \quad (4.6)$$

where the sets  $\mathcal{U}$  and  $\mathcal{X}$  are compact and closed respectively. The control objective is to steer the state to a target set  $\mathcal{T} \subset \mathbb{R}^n$ , which is also closed, in time  $N < \infty$ , expressed as the constraint

$$x(N) \in \mathcal{T}. \quad (4.7)$$

The control input and horizon length are ideally chosen to minimise a p-norm cost function that penalises inputs, states and time to completion. The value function takes the form

$$J^*(x(0)) = \min_{u, N} \sum_{k=0}^{N-1} (1 + \|\Theta(y(k) - y_r)\|), \quad (4.8)$$

for some matrix  $\Theta \in \mathbb{R}^{p \times p}$ , sequence of inputs  $u$  and some reference output  $y_r = Cx_r + Du_r$ , where  $x_r \in \mathcal{T}$  and  $u_r \in \mathcal{U}$  respectively. The behaviour of the system after reaching  $\mathcal{T}$  is not specified as part of the control problem, and therefore there is no terminal cost in (4.8).

### 4.3 Move Blocking Formulation

In order to represent different move blocking regimes, blocking matrices will be employed. These will be subsequently used in the development of a predictive controller to solve the control problem posed in the previous section.

**Definition 4.1.** A blocking matrix  $S \in \{0, 1\}^{N \times r}$  is one that takes the form

$$S = \begin{bmatrix} b_1 & & & \\ & b_2 & & \\ & & \ddots & \\ & & & b_r \end{bmatrix} = \bigoplus_{q=1}^r b_q, \quad (4.9)$$

where each  $b_q = \mathbf{1}_{l_q}$  is termed a blocking vector, having length  $l_q \in \mathbb{N}$ . The lengths satisfy

$$\sum_{q=1}^r l_q = N. \quad (4.10)$$

The set of all  $N \times r$  blocking matrices will be denoted  $\mathcal{A}(N, r)$ .

Given a sequence of elements  $v_i \in \mathbb{R}^{m \times 1}$ ,  $i \in \mathbb{N}_{[1, N]}$  and a corresponding sequence of block values  $\omega_q \in \mathbb{R}^{m \times 1}$ ,  $q \in \mathbb{N}_{[1, r]}$ , the blocking matrix  $S \in \mathcal{A}(N, r)$  relates the two sequences arranged in matrix form by the equation

$$\bigcup_{i=1}^N v_i = \left( \bigcup_{q=1}^r \omega_q \right) S^T = \left( \bigcup_{q \in \mathcal{S}} \omega_q \right) S^T, \quad (4.11)$$

where, with a slight abuse of notation, the symbol  $S$  has also been used to represent the ordered set of all block indices, namely  $\mathbb{N}_{[1, r]}$ . Now define the set-valued map of blocking matrices that subdivide  $N$  elements or fewer into  $r$  blocks or fewer by

$$\mathcal{S}(N, r) = \{S \in \mathcal{A}(i, q) \mid 0 \leq i \leq N, 0 \leq q \leq \min\{i, r\}\}. \quad (4.12)$$

**Definition 4.2.** A relaxation  $S'$  of a blocking matrix  $S \in \mathcal{A}(N, r)$  is another blocking matrix in which blocks can be amalgamated to give  $S$ . The set of all relaxations of  $S$  is given by

$$\mathcal{R}(S) = \left\{ S' \in \mathbb{R}^{N \times r'} \mid \exists M \in \mathcal{A}(r', r) : S = S'M, r' \geq r \right\}. \quad (4.13)$$

The identity blocking matrix (i.e. no blocking) is a relaxation of any blocking matrix. Each blocking matrix is also a relaxation of itself. The set-valued map  $\mathcal{R}(S, r')$  will be used to denote

all relaxations having exactly  $r'$  blocks.

Define the function  $\ell : \mathcal{A}(N, r) \times \mathbb{N}_{[1, r]} \mapsto \mathbb{N}_{[1, N-r+1]}$  for a given blocking matrix  $S \in \mathcal{A}(N, r)$ , which returns the length of the  $q$ th block.

$$\ell(S, q) = \|S\hat{e}_q\|_1, \quad (4.14)$$

where  $\hat{e}_q$  is the  $q$ th Cartesian unit vector in  $r$  dimensions, (i.e. the  $q$ th column of  $I_r$ ). Furthermore, define a cumulative block length function  $\Lambda(S, q)$  as

$$\Lambda(S, q) = \begin{cases} 0 & q = 0 \\ \sum_{h=1}^q \ell(S, h), & q \in \mathbb{N}_{[1, r]}. \end{cases} \quad (4.15)$$

A block-end indexing function is now defined in terms of  $\Lambda(\cdot, \cdot)$ .

$$\begin{aligned} \mu(S, j) &= \begin{cases} \Lambda(S, 1) - 1 & \Lambda(S, 0) \leq j < \Lambda(S, 1) \\ \Lambda(S, 2) - 1 & \Lambda(S, 1) \leq j < \Lambda(S, 2) \\ \vdots & \\ \Lambda(S, r) - 1 & \Lambda(S, r-1) \leq j < \Lambda(S, r) \end{cases} \\ &= \left\{ \Lambda(S, q) - 1 \quad \Lambda(S, q-1) \leq j < \Lambda(S, q), q \in \mathbb{N}_{[1, r]} \right\}. \end{aligned} \quad (4.16)$$

A function is also defined that returns the block number corresponding to the zero-based element index.

$$\begin{aligned} \sigma(S, j) &= \begin{cases} 1 & \Lambda(S, 0) \leq j < \Lambda(S, 1) \\ 2 & \Lambda(S, 1) \leq j < \Lambda(S, 2) \\ \vdots & \\ r & \Lambda(S, r-1) \leq j < \Lambda(S, r) \end{cases} \\ &= \left\{ q \quad \Lambda(S, q-1) \leq j < \Lambda(S, q), q \in \mathbb{N}_{[1, r]} \right\}. \end{aligned} \quad (4.17)$$

An example illustrating the action of functions  $\Lambda(\cdot, \cdot)$ ,  $\mu(\cdot, \cdot)$  and  $\sigma(\cdot, \cdot)$  can be found in Figure 4.1.

Finally, define the *shifted* blocking matrix of  $S \in \mathcal{A}(N, r)$ , where  $N > 1$  by the function

$$F(S) = \begin{cases} \begin{bmatrix} 0_{(N-1) \times 1} & I_{N-1} \end{bmatrix} S & \ell(S, 1) > 1 \\ \begin{bmatrix} 0_{(N-1) \times 1} & I_{N-1} \end{bmatrix} S \begin{bmatrix} 0_{1 \times (r-1)} \\ I_{r-1} \end{bmatrix} & \ell(S, 1) = 1. \end{cases} \quad (4.18)$$

**Remark 4.3.** Shifting essentially removes the first row of  $S$  and also the first column if  $\ell(S, 1) = 1$  to ensure admissibility of the resulting matrix. This is a modification of the operation used in [Cag+07] to suit the variable horizon.

0	1	2	3	4	5	6	7	8	$j$
$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	$v_7$	$v_8$	$v_9$	$v_{j+1}$
1		2				3			$\sigma(S, j)$
2		6				9			$\Lambda(S, \sigma(S, j))$
1	1	5	5	5	5	8	8	8	$\mu(S, j)$

**Figure 4.1:** Values of functions  $\Lambda(\cdot, \cdot)$  and  $\mu(\cdot, \cdot)$  for blocking matrix corresponding to  $N = 9$  and block lengths  $\{2, 4, 3\}$

Theorems 4.4 - 4.6 quantify the maximum number of blocking matrices and relaxations for a given number of elements and blocks.

**Theorem 4.4.** *Given  $N$  elements, the number of distinct blocking matrices comprising exactly  $r \leq N$  blocks is given by*

$$|\mathcal{A}(N, r)| = \binom{N-1}{r-1}. \quad (4.19)$$

*Proof.* Any blocking matrix  $S \in \mathcal{A}(N, r)$  maps bijectively onto a sequence of block lengths  $l_1, \dots, l_r$ . These lengths are required to satisfy

$$\sum_{q=1}^r l_q = N. \quad (4.20)$$

Thus, a particular blocking matrix maps bijectively onto a particular *composition* [HM09] of  $N$ . Using the notation

$$\{l_1, l_2, \dots, l_r\} = \{1 \square 1 \square 1 \square \dots \square 1\} \quad (N \text{ terms}), \quad (4.21)$$

it can be seen that a unique composition can be obtained by replacing each ' $\square$ ' by either an addition sign (+) or comma (.). This presents a binary choice, and since exactly  $r - 1$  commas must be placed to obtain  $r$  blocks, there are  $\binom{N-1}{r-1}$  possible ways of doing this. ■

**Theorem 4.5.** *The number of blocking matrices that subdivide  $N$  or fewer elements into  $r$  or fewer blocks is given by*

$$|\mathcal{S}(N, r)| = 2^{r-1} - 1 + \sum_{i=r}^N \sum_{q=1}^r \binom{i-1}{q-1}. \quad (4.22)$$

*Proof.* The result uses Theorem 4.4 and follows from expanding the summation

$$\sum_{i=1}^N \sum_{q=1}^{\min\{i,r\}} \binom{i-1}{q-1}. \quad (4.23)$$

Firstly consider counting the number of matrices for a fixed number of elements  $i$ , where  $r \leq i \leq N$ . Then, there are

$$|\mathcal{A}(i, r)| = \sum_{q=1}^r \binom{i-1}{q-1} \quad (4.24)$$

matrices having  $r$  blocks or fewer, given exactly  $i$  elements. From the fact that

$$\sum_{q=0}^i \binom{i}{q} = 2^i, \quad (4.25)$$

there are  $2^{i-1}$  matrices in the special case that  $r = i$ . To now sum over the number of elements  $i$ , two cases need to be considered, as there cannot be more blocks than elements. The number of matrices where  $r \leq i$  is given by the summation

$$\sum_{i=r}^N |\mathcal{A}(i, r)| = \sum_{i=r}^N \sum_{q=1}^r \binom{i-1}{q-1}. \quad (4.26)$$

When  $i < r$ , there can only be a maximum of  $i$  blocks, which means that the number of matrices is given by

$$\begin{aligned} \sum_{i=1}^{r-1} |\mathcal{A}(i, i)| &= \sum_{i=1}^{r-1} 2^{i-1} \\ &= 2^{r-1} - 1. \end{aligned} \quad (4.27)$$

Summing (4.26) and (4.27) gives the required result. ■

**Theorem 4.6.** Given  $S \in \mathcal{A}(N, r)$ ,  $|\mathcal{R}(S)| = 2^{N-r}$ .

*Proof.* As with the proof of Theorem 4.4, represent  $S$  as a sequence of block lengths  $l_1, \dots, l_r$ . Then, by noting that any relaxation of  $S$  may be re-blocked to give  $S$ , the problem reduces to finding the number of ways of replacing each ' $\square$ ' by either an addition sign (+) or comma (,) in equation (4.21), subject to  $r - 1$  commas already having been placed (corresponding to  $S$ ). This is again a binary decision problem, so the number of ways of placing exactly  $i$  commas in the remaining  $N - r$  squares is given by  $\binom{N-r}{i}$ . The result follows from applying (4.25). ■

Lemmas 4.7 - 4.11 are used in the development of a recursively-feasible move-blocked variable horizon predictive controller.

**Lemma 4.7.** For a blocking matrix  $S \in \mathcal{A}(N, r)$  and some relaxation  $S' \in \mathcal{R}(S, r')$ , where

$$\bigcup_{i=1}^N v_i = \left( \bigcup_{q=1}^r \omega_q \right) S^T, \quad (4.28)$$

there exists a sequence of block values  $\omega'_1, \dots, \omega'_{r'}$  such that

$$\bigcup_{i=1}^N v_i = \left( \bigcup_{q=1}^{r'} \omega'_q \right) S'^T. \quad (4.29)$$

*Proof.* Given  $S' \in \mathcal{R}(S, r')$ , there exists a blocking matrix  $M \in \mathcal{A}(r', r)$  such that  $S'M = S$ . Hence

$$\bigcup_{i=1}^N v_i = \left( \bigcup_{q=1}^r \omega_q \right) M^T S'^T. \quad (4.30)$$

Then, the result follows by choosing

$$\bigcup_{q=1}^{r'} \omega'_q = \left( \bigcup_{q=1}^r \omega_q \right) M^T. \quad (4.31)$$

■

**Lemma 4.8.** For any  $S \in \mathcal{A}(N, r)$  and mapping  $f : \mathbb{R} \mapsto \mathbb{R}^{n \times 1}$ ,

$$\bigcup_{j=0}^{N-1} f(\mu(S, j)) = \left( \bigcup_{q \in S} f(\Lambda(S, q) - 1) \right) S^T. \quad (4.32)$$

*Proof.* Firstly, the blocking matrix is decomposed into blocking vectors  $b_1, \dots, b_r$  as in (4.9). Then, it can be seen that

$$\left( \bigcup_{q \in S} f(\Lambda(S, q) - 1) \right) S^T = \bigcup_{q \in S} f(\Lambda(S, q) - 1) b_q^T. \quad (4.33)$$

As each  $b_q^T$  is a row vector of ones with length  $\ell(S, q)$ , the RHS of (4.33) has length  $N$ . From the definition of  $\mu(\cdot, \cdot)$  (4.16), expanding the  $q$ th term of the RHS gives

$$f(\Lambda(S, q) - 1) b_q^T = \bigcup_{j=\Lambda(S, q-1)}^{\Lambda(S, q)-1} f(\mu(S, j)). \quad (4.34)$$

Concatenating over all blocks gives the required result. ■

As a numerical example, consider blocking matrix  $S$  corresponding to block lengths  $\{3, 2, 2\}$ . For  $j = \mathbb{Z}_{[0,6]}$ , it is clear that  $\mu(S, j) = \{2, 2, 2, 4, 4, 6, 6\}$ . Also, the block indices are given by  $q = \{1, 2, 3\}$ , so  $\Lambda(S, q) - 1 = \{2, 4, 6\}$ . It is now easy to see that the result holds when concatenating the sequences into vectors.

**Lemma 4.9.** For any  $S \in \mathcal{A}(N, r)$  and  $S' \in \mathcal{R}(S)$ ,

$$\mu(S', j) \leq \mu(S, j), \forall j \in \mathbb{Z}_{[0, N-1]}. \quad (4.35)$$

*Proof.* Since any relaxation of a blocking matrix is itself re-blocked to give the original matrix, consider the  $q$ th block of  $S$ . In the relaxation, this block is either unchanged or itself divided into smaller blocks. This means that  $\ell(S', q)$  cannot be greater than  $\ell(S, q)$ , so  $\Lambda(S', q) \leq \Lambda(S, q)$ . The result then follows from (4.16). ■

**Lemma 4.10.** For any  $S \in \mathcal{A}(N, r)$

$$\Lambda(F(S), q) = \Lambda(S, q + \delta[\ell(S, 1) - 1]) - 1, \quad (4.36)$$

for all  $q \in \mathbb{Z}_{[0, r - \delta[\ell(S, 1) - 1]]}$ .

*Proof.* Shifting removes the first element of the first block, which reduces the cumulative block lengths by unity. In addition, when  $\ell(S, 1) = 1$ , the entire first block is removed, so  $\ell(F(S), q) = \ell(S, q + 1)$ . ■

**Lemma 4.11.** For any blocking matrix  $S \neq 1$ ,

$$\mu(S, j + 1) - 1 = \mu(F(S), j). \quad (4.37)$$

*Proof.* The proof follows from (4.16) and Lemma 4.10. It can be seen that

$$\begin{aligned} & \mu(S, j + 1) - 1 \\ &= \begin{cases} \Lambda(S, q) - 1 - 1, \\ \Lambda(S, q - 1) - 1 < j < \Lambda(S, q) - 1, q \in \mathbb{N}_{[1 + \delta[\ell(S, 1) - 1], r]} \end{cases} \\ &= \begin{cases} \Lambda(F(S), q) - 1, \\ \Lambda(F(S), q - 1) < j < \Lambda(F(S), q), q \in \mathbb{N}_{[1, r - \delta[\ell(S, 1) - 1]]} \end{cases} \\ &= \mu(F(S), j). \end{aligned} \quad \blacksquare$$

## 4.4 Blocked VH-MPC

A variable horizon predictive controller will now be formulated using move blocking for complexity reduction. The variable horizon is achieved by defining a number of fixed horizon optimisation problems with different horizon lengths and compatible distinct blocking matrices. Denote the maximum horizon length as  $\bar{N}$  and the maximum allowable number of input variables in the optimisation problem as  $\bar{r}$ . Then, there are  $|\mathcal{S}(\bar{N}, \bar{r})|$  possible optimisation problems corresponding to distinct blocking matrices. A nominal controller will be designed first using some subset of these, with constraint tightening applied subsequently for robustness to bounded disturbances.

#### 4.4.1 Nominal Controller

For a given blocking matrix  $S \in \mathcal{A}(N_S, r_S) \subseteq \mathcal{S}(\bar{N}, \bar{r})$  (that is, where  $N_S \leq \bar{N}$  and  $r_S \leq \bar{r}$ ), define the optimisation problem  $\mathcal{P}(S)$  as finding

$$J[\mathcal{P}(S)] = \min_{\{v(S, \cdot)\}} \sum_{j=0}^{N_S-1} (1 + \|\Theta(y(k+j|k) - y_r)\|), \quad (4.38)$$

subject to the dynamics constraints

$$x(k|k) = x(k) \quad (4.39a)$$

$$x(k+j+1|k) = Ax(k+j|k) + Bu(k+j|k) \quad (4.39b)$$

$$y(k+j|k) = Cx(k+j|k) + Du(k+j|k), \quad (4.39c)$$

the operating constraints

$$x(k+j|k) \in \mathcal{X} \quad (4.40a)$$

$$u(k+j|k) \in \mathcal{U} \quad (4.40b)$$

$$x(k+N_S|k) \in \mathcal{T}, \quad (4.40c)$$

and the move blocking constraint

$$\prod_{j=0}^{N_S-1} u(k+j|k) = \left( \prod_{q=1}^{r_S} v(S, q) \right) S^T. \quad (4.41)$$

The notation  $v(S, q)$  represents the value of the  $q$ th block given blocking matrix  $S$  and  $y_r$  is the output reference defined in Section 4.2.

**Remark 4.12.** Zero-based indexing is used in the LHS of (4.41) to be consistent with prediction model notation.

Define now a set of blocking matrices  $\mathcal{C} \subseteq \mathcal{S}(\bar{N}, \bar{r})$  and a relaxation function

$$\rho : \mathcal{S}(\bar{N}, \bar{r}) \mapsto \mathcal{S}(\bar{N}, \bar{r}), \quad (4.42)$$

where

$$\rho(S) \in \mathcal{R}(S, r'), r' \leq \bar{r}. \quad (4.43)$$

This function maps a given blocking matrix to precisely one of its relaxations having a maximum of  $\bar{r}$  blocks, and is a design choice. Algorithm 4.1 describes the implementation of the blocked variable horizon in terms of the set  $\mathcal{C}$  and the function  $\rho(\cdot)$ . At each time step, it essentially solves all problems in  $\mathcal{C}$  plus the shifted problem, choosing the blocking matrix that gives the best cost performance.

---

**Algorithm 4.1:** Nominal Blocked VH-MPC
 

---

```

1 while  $x(k) \notin \mathcal{T}$  do
2   if  $k = 0$  then
3      $\mathcal{C}_f \leftarrow \{S \in \mathcal{C} \mid \mathcal{P}(S) \text{ feasible}\};$ 
4     if  $\mathcal{C}_f = \emptyset$  then
5       No feasible solution exists for the given starting state and set of blocking matrices;
6       exit;
7     end
8   else
9      $\hat{S}(k) \leftarrow F(S^{*'}(k)), \text{ where } S^{*'}(k) = \rho(S^*(k));$ 
10     $\mathcal{C}_f \leftarrow \{S \in \mathcal{C} \mid \mathcal{P}(S) \text{ feasible}\} \cup \{\hat{S}(k)\};$ 
11  end
12   $J^*(x(k)) \leftarrow \min_{S \in \mathcal{C}_f} J[\mathcal{P}(S)]; S^*(k) \leftarrow \arg \min_{S \in \mathcal{C}_f} J[\mathcal{P}(S)];$ 
13   $u(k) \leftarrow v(S^*(k), 1) \implies x(k+1) = Ax(k) + Bv(S^*(k), 1);$ 
14   $k \leftarrow k + 1;$ 
15 end
    
```

---

**Remark 4.13.** The set of repeatedly shifted and relaxed blocking matrices

$$\mathcal{D} = \bigcup_{S \in \mathcal{C}} \{S, F(\rho(S)), F(\rho(F(\rho(S))))\}, \dots, 1\}, \quad (4.44)$$

can be calculated in advance if an a priori definition of the problem  $\mathcal{P}(\hat{S}(\cdot))$  is required for all time steps.

**Remark 4.14.**  $\mathcal{C}_f$  is defined as the set of blocking matrices in  $\mathcal{C}$  whose corresponding optimisation problems have a feasible solution, together with the shifted blocking matrix of the previously optimal problem. It is calculated by actually solving all of the optimisation problems corresponding to the matrices in  $\mathcal{C}$ , which implies the solution of multiple optimisation problems at each time step. However, these problems can be solved in parallel, meaning that the controller latency is that of the slowest single optimisation problem rather than the sum of the solution times. Note that either  $|\mathcal{C}|$  or  $|\mathcal{C}| + 1$  problems are solved at each step, depending on whether or not it is the first iteration and whether or not the shifted blocking matrix is already in  $\mathcal{C}$ .

**Theorem 4.15** (Nominal Recursive Feasibility). *Given the optimisation problem and associated state trajectory, input trajectory, output trajectory and horizon length that correspond to the optimal solution found by Algorithm 4.1 at time  $k$ , namely  $\mathcal{P}(S^*(k))$ ,  $\{x^*(k+j|k)\}$ ,  $\{u^*(k+j|k)\}$ ,  $\{y^*(k+j|k)\}$ , and  $N^*(k)$ , a feasible solution to the problem  $\mathcal{P}(\hat{S}(k))$  at time  $k+1$  is given by*

$$\hat{x}(k+j+1|k+1) = x^*(k+j+1|k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]} \quad (4.45a)$$

$$\hat{u}(k+j+1|k+1) = u^*(k+j+1|k), \forall j \in \mathbb{Z}_{[0, N^*(k)-1]} \quad (4.45b)$$

$$\hat{y}(k+j+1|k+1) = y^*(k+j+1|k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]} \quad (4.45c)$$

$$\hat{N}(k+1) = N^*(k) - 1. \quad (4.45d)$$

*Proof.* The candidate solution is evidently feasible with respect to the system dynamics and constraints, as it is simply a shifted version of the inputs and states at time  $k$ , albeit taking one fewer step to reach completion. To show that the blocking constraints are satisfied by the candidate inputs, observe that from Lemma 4.7,

$$\begin{aligned} \bigoplus_{j=0}^{N^*(k)-1} u^*(k+j|k) &= \left( \bigoplus_{q \in S^*(k)} v(S^*(k), q) \right) S^{*T}(k) \\ &= \left( \bigoplus_{q \in S^{*'}(k)} v'(S^{*'}(k), q) \right) S^{*'}(k), \end{aligned} \quad (4.46)$$

where  $S^{*'}(k) = \rho(S^*(k)) \in \mathcal{A}(N^*(k), r')$  and  $v'(\cdot, \cdot)$  are the block values corresponding to the relaxation. Then,

$$\bigoplus_{j=0}^{N^*(k)-2} u^*(k+j+1|k) = \left( \bigoplus_{q=s+1}^{r'} v'(S^{*'}(k), q) \right) F^T(S^{*'}(k)), \quad (4.47)$$

where  $s = \delta[\ell(S^{*'}(k), 1) - 1]$ . Hence, substituting the candidate solution gives

$$\bigoplus_{j=0}^{\hat{N}^{(k+1)}-1} \hat{u}(k+j+1|k+1) = \left( \bigoplus_{q \in \hat{S}(k)} \hat{v}(\hat{S}(k), q) \right) \hat{S}^T(k), \quad (4.48)$$

where  $\hat{v}(\hat{S}(k), q) = v'(S^{*'}(k), q + s)$ . This is the required result.  $\blacksquare$

By induction, therefore, the entire optimisation problem is feasible to completion from an initial feasible solution. Finite-time completion of the control manoeuvre can now be proven.

**Theorem 4.16** (Nominal Finite-Time Completion). *Algorithm 4.1 guarantees that  $\mathcal{T}$  will be reached in finite-time  $\lfloor J^*(x(0)) \rfloor$ , given  $\mathcal{C}_f$  is non-empty at the first iteration.*

*Proof.* It can be seen that the cost of the feasible trajectory at time  $k+1$ , denoted  $\hat{J}(\cdot)$  is related to the cost of the optimal trajectory at time  $k$  by the equation

$$\hat{J}(x(k+1)) = J^*(x(k)) - 1. \quad (4.49)$$

By optimality, this is an upper bound on the optimal cost at time  $k+1$ , therefore

$$J^*(x(k+1)) \leq J^*(x(k)) - 1. \quad (4.50)$$

As the cost function is non-negative by construction, this means that, given an initial feasible solution and recursive feasibility from Theorem 4.15, completion occurs in at most  $\lfloor J^*(x(0)) \rfloor$  steps from the starting state.  $\blacksquare$

#### 4.4.2 Robustness

The controller formulation is now extended to include robustness to bounded disturbances. A disturbance-feedback constraint tightening approach [KRH07] is used to achieve this robustness. A bounded disturbance is now introduced into the linear system model, giving the dynamic equation

$$x(k+1) = Ax(k) + Bu(k) + w(k), \quad (4.51)$$

where  $w(k) \in \mathcal{W} \subset \mathbb{R}^n$ , for some compact set  $\mathcal{W}$  containing the origin.

Define the tightened input constraints by the recursion

$$\mathcal{U}(0) = \mathcal{U} \quad (4.52a)$$

$$\mathcal{U}(j+1) = \mathcal{U}(j) \ominus P(j)\mathcal{W}. \quad (4.52b)$$

As in the previous chapter,  $\{P(j) \in \mathbb{R}^{m \times n}\}$  is a sequence of matrices representing a candidate control policy providing direct feedback on the disturbance. The policy will be used to ensure robustness to bounded disturbances, albeit in a modified manner to accommodate move blocking.

Now, in terms of the function  $\mu(\cdot, \cdot)$ , define the matrices

$$L(S, 0) = I_n \quad (4.53a)$$

$$L(S, j+1) = AL(S, j) + BP(\mu(F(\rho(S)), j)) \quad (4.53b)$$

and the corresponding sets

$$\mathcal{X}(S, 0) = \mathcal{X} \quad (4.54a)$$

$$\mathcal{T}(S, 0) = \mathcal{T} \quad (4.54b)$$

$$\mathcal{X}(S, j+1) = \mathcal{X}(F(\rho(S)), j) \ominus L(S, j)\mathcal{W} \quad (4.54c)$$

$$\mathcal{T}(S, j+1) = \mathcal{T}(F(\rho(S)), j) \ominus L(S, j)\mathcal{W}. \quad (4.54d)$$

**Remark 4.17.** The sets defined by (4.54c) and (4.54d) can be computed recursively by using the set  $\mathcal{P}$  (4.44), starting from the blocking matrix  $S$  relaxed and shifted to unity, and working backwards. This tightening scheme is substantially different to that used in [KRH07], as a different sequence of tightened input, state and terminal constraint sets is needed for each blocking matrix in the regime.

For a given blocking matrix  $S \in \mathcal{A}(N_S, r_S)$ , define the robust optimisation problem  $\tilde{\mathcal{P}}(S)$  as finding the minimum (4.38) subject to the tightened constraints

$$u(k+j|k) \in \mathcal{U}(\mu(S, j)) \quad (4.55a)$$

$$x(k+j|k) \in \mathcal{X}(S, j) \quad (4.55b)$$

$$x(k+N_S|k) \in \mathcal{T}(S, N_S), \quad (4.55c)$$

with the move-blocking constraint (4.41). Note how the function  $\mu(\cdot, \cdot)$  is used here to ensure that the input constraints applied to each block correspond to the tightest constraint for any input within the block. This is evident from the definition of  $\mu(\cdot, \cdot)$  as the block-end index.

By applying Algorithm 4.1 with the nominal problems  $\mathcal{P}(\cdot)$  replaced by the robust problems  $\tilde{\mathcal{P}}(\cdot)$ , recursive feasibility is guaranteed for any allowable sequence of disturbances. This result is presented in the following theorem.

**Theorem 4.18** (Robust Recursive Feasibility). *Given the optimisation problem and associated state trajectory, input trajectory, output trajectory and horizon length that correspond to the optimal solution found by Algorithm 4.1 at time  $k$ , namely  $\tilde{\mathcal{P}}(S^*(k))$ ,  $\{x^*(k+j|k)\}$ ,  $\{u^*(k+j|k)\}$ ,  $\{y^*(k+j|k)\}$  and  $N^*(k)$ , a feasible solution to the problem  $\tilde{\mathcal{P}}(\hat{S}(k))$  at time  $k+1$  is given by*

$$\hat{u}(k+j+1|k+1) = u^*(k+j+1|k) + P(\mu(F(S^{*'}(k)), j))w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]} \quad (4.56a)$$

$$\hat{x}(k+j+1|k+1) = x^*(k+j+1|k) + L(S^*(k), j)w(k), \forall j \in \mathbb{Z}_{[0, N^*(k)-2]} \quad (4.56b)$$

$$\hat{y}(k+j+1|k+1) = y^*(k+j+1|k) + (CL(S^*(k), j) + DP(\mu(F(S^{*'}(k)), j)))w(k), \quad (4.56c)$$

$$\forall j \in \mathbb{Z}_{[0, N^*(k)-2]}$$

$$\hat{N}(k+1) = N^*(k) - 1, \quad (4.56d)$$

where  $S^{*'}(k) = \rho(S^*(k)) \in \mathcal{A}(N^*(k), r')$ .

*Proof.* To demonstrate that the candidate solution is feasible, it must be shown to satisfy the initial constraints, dynamics constraints, move blocking constraints, state and input constraints and the terminal constraint enforced by problem  $\tilde{\mathcal{P}}(\hat{S}(k)) = \tilde{\mathcal{P}}(F(S^{*'}(k)))$ .

Firstly, to show that the initial constraint holds, observe that the optimal solution satisfies

$$x^*(k|k) = x(k) \quad (4.57)$$

from (4.39a). Using (4.39b), (4.41) and substituting the candidate solution (4.56), it can be seen that

$$x^*(k+1|k) = Ax(k) + Bu^*(k|k) \quad (4.58)$$

$$\begin{aligned} \implies \hat{x}(k+1|k+1) &= Ax(k) + Bv(S^*(k), 1) + w(k) \\ &= x(k+1) \end{aligned} \quad (4.59)$$

as required. For the dynamics constraints, the optimal solution satisfies

$$x^*(k+j+2|k) = Ax^*(k+j+1|k) + Bu^*(k+j+1|k). \quad (4.60)$$

Substituting the candidate solution gives

$$\begin{aligned} &\hat{x}(k+j+2|k+1) - L(S^*(k), j+1)w(k) \\ &= A(\hat{x}(k+j+1|k+1) - L(S^*(k), j)w(k)) + B(\hat{u}(k+j+1|k+1) - P(\mu(F(S^{*'}(k)), j))w(k)) \\ &= A\hat{x}(k+j+1|k+1) + B\hat{u}(k+j+1|k+1) - (AL(S^*(k), j) + BP(\mu(F(S^{*'}(k)), j)))w(k) \end{aligned}$$

$$= A\hat{x}(k+j+1|k+1) + B\hat{u}(k+j+1|k+1) - L(S^*(k), j+1)w(k),$$

hence,

$$\hat{x}(k+j+2|k+1) = A\hat{x}(k+j+1|k+1) + B\hat{u}(k+j+1|k+1) \quad (4.61)$$

as required. For the output dynamics, the optimal solution satisfies

$$y^*(k+j+1|k) = Cx^*(k+j+1|k) + Du^*(k+j+1|k). \quad (4.62)$$

Then, substituting the candidate solution shows that

$$\begin{aligned} & \hat{y}(k+j+1|k+1) - (CL(S^*(k), j) + DP(\mu(F(S^{*'}(k)), j)))w(k) \\ &= C(\hat{x}(k+j+1|k+1) - L(S^*(k), j)w(k)) + D(\hat{u}(k+j+1|k+1) - P(\mu(F(S^{*'}(k)), j))w(k)) \\ &= C\hat{x}(k+j+1|k+1) + D\hat{u}(k+j+1|k+1) - (CL(S^*(k), j) + DP(\mu(F(S^{*'}(k)), j)))w(k). \end{aligned} \quad (4.63)$$

Hence,

$$\hat{y}(k+j+1|k+1) = C\hat{x}(k+j+1|k+1) + D\hat{u}(k+j+1|k+1). \quad (4.64)$$

For the move blocking constraints, observe that

$$\mathcal{C}_{j=0}^{N^*(k)-2} \hat{u}(k+j+1|k+1) = \mathcal{C}_{j=0}^{N^*(k)-2} u^*(k+j+1|k) + \mathcal{C}_{j=0}^{N^*(k)-2} P(\mu(F(S^{*'}(k)), j))w(k). \quad (4.65)$$

Looking at each part of the RHS individually, it is evident that

$$\mathcal{C}_{j=0}^{N^*(k)-2} u^*(k+j+1|k) = \left( \mathcal{C}_{q \in \hat{S}(k)} \hat{v}(\hat{S}(k), q) \right) \hat{S}^T(k), \quad (4.66)$$

which is shown in the proof of Theorem 4.15. Also, from Lemmas 4.8 and 4.10,

$$\mathcal{C}_{j=0}^{N^*(k)-2} P(\mu(F(S^{*'}(k)), j))w(k) = \left( \mathcal{C}_{q=1}^{r'-s} P(\Lambda(F(S^{*'}(k)), q) - 1)w(k) \right) F^T(S^{*'}(k)) \quad (4.67)$$

$$= \left( \mathcal{C}_{q \in \hat{S}(k)} P(\Lambda(\hat{S}(k), q) - 1)w(k) \right) \hat{S}^T(k), \quad (4.68)$$

where  $s = \delta[\ell(S^{*'}(k), 1) - 1]$ . Combining (4.66) and (4.68) gives

$$\mathcal{C}_{j=0}^{\hat{N}(k+1)-1} \hat{u}(k+j+1|k+1) = \left( \mathcal{C}_{q \in \hat{S}(k)} \hat{v}(\hat{S}(k), q) + P(\Lambda(\hat{S}(k), q) - 1)w(k) \right) \hat{S}^T(k). \quad (4.69)$$

Hence, the blocking constraints on the inputs are respected.

For the input constraints, (4.52a)-(4.52b), together with Lemmas 4.9 and 4.11 show that

$$u^*(k+j+1|k) \in \mathcal{U}(\mu(S^*(k), j+1)) \quad (4.70)$$

$$\subseteq \mathcal{U}(\mu(S^{*'}(k), j+1)) \quad (4.71)$$

$$= \mathcal{U}(\mu(F(S^{*'}(k)), j) + 1) \quad (4.72)$$

$$= \mathcal{U}(\mu(F(S^{*'}(k)), j)) \ominus P(\mu(F(S^{*'}(k)), j))\mathcal{W}. \quad (4.73)$$

This means that, from (2.55),

$$u^*(k+j+1|k) + P(\mu(F(S^{*'}(k)), j))w(k) \in \mathcal{U}(\mu(F(S^{*'}(k)), j))$$

which implies that

$$\hat{u}(k+j+1|k+1) \in \mathcal{U}(\mu(F(S^{*'}(k)), j)) = \mathcal{U}(\mu(\hat{S}(k), j)). \quad (4.74)$$

For the state constraints

$$x^*(k+j+1|k) \in \mathcal{X}(S^*(k), j+1). \quad (4.75)$$

Then from (2.55) and (4.54c),

$$x^*(k+j+1|k) + L(S^*, j)w(k) \in \mathcal{X}(F(S^{*'}(k)), j) \implies \hat{x}(k+j+1|k+1) \in \mathcal{X}(\hat{S}(k), j). \quad (4.76)$$

Finally, for the terminal constraint

$$x^*(k+N^*(k)|k) \in \mathcal{T}(S^*(k), N^*(k)). \quad (4.77)$$

Then, from (2.55) and (4.54d),

$$\begin{aligned} x^*(k+N^*(k)|k) + L(S^*(k), N^*(k)-1)w(k) &\in \mathcal{T}(F(S^{*'}(k)), N^*(k)-1) \\ \implies \hat{x}(k+1+\hat{N}(k+1)|k+1) &\in \mathcal{T}(\hat{S}(k), \hat{N}(k+1)). \end{aligned} \quad (4.78)$$

■

**Theorem 4.19** (Robust Finite-Time Completion). *If the cost weightings on the states and inputs are chosen to ensure that  $\eta > 0$ , where*

$$\eta = 1 - \max_{\substack{w \in \mathcal{W} \\ S \in \mathcal{S}(\bar{N}, \bar{r})}} \left\{ \|\Theta\| \sum_{j=0}^{\bar{N}-2} \|(CL(S, j) + DP(\mu(F(S'), j))w)\| \right\} \quad (4.79)$$

and  $S' = \rho(S)$ , the robust controller guides the system to the target set in at most  $\lfloor J^*(x(0))/\eta \rfloor$  steps, from the initial state  $x(0)$ .

*Proof.* The proof utilises the value function as a Lyapunov-like function. Applying the triangle inequality and the definition of the induced p-norm, it can be seen that the cost of (4.56),

denoted  $\hat{J}(x(k+1))$  is related to the optimal cost at time  $k$ , namely  $J^*(x(k))$ , by

$$\hat{J}(x(k+1)) - J^*(x(k)) \leq -1 + \sum_{j=0}^{N^*(k)-2} (\|\Theta(CL(S^*, j) + DP(\mu(F(S^{*'}), j)))w(k)\|) \quad (4.80)$$

$$\leq \max_{w \in \mathcal{W}} \left\{ \sum_{j=0}^{\bar{N}-2} \|\Theta(CL(S^*, j) + DP(\mu(F(S^{*'}), j)))w\| \right\} - 1 \quad (4.81)$$

$$\leq \max_{\substack{w \in \mathcal{W} \\ S \in \mathcal{S}(\bar{N}, \bar{r})}} \left\{ \|\Theta\| \sum_{j=0}^{\bar{N}-2} \|(CL(S, j) + DP(\mu(F(S'), j)))w\| \right\} - 1 \quad (4.82)$$

$$= -\eta. \quad (4.83)$$

From optimality,  $\hat{J}(x(k+1))$  is an upper bound on the optimal cost at time  $k+1$ , hence

$$J^*(x(k)) - J^*(x(k+1)) \geq J^*(x(k)) - \hat{J}(x(k+1)) \geq \eta. \quad (4.84)$$

This means that the cost at successive time steps must reduce by at least  $\eta$ . Given that the cost must be non-negative by construction and  $\eta > 0$ , completion must be reached in at most  $\lfloor J^*(x(0))/\eta \rfloor$  steps, from initial state  $x(0)$ .  $\blacksquare$

## 4.5 Implementation

The formulation presented in the previous section uses equality constraints on the inputs to enforce move blocking. This is the most suitable form for analysis, but not necessarily in implementation. Explicitly implementing the equality constraints in an optimisation program will not necessarily provide a reduction in computation time. Whether or not a reduction occurs will depend on how the optimisation algorithm treats equality constraints. In order to achieve a guaranteed reduction, the prediction model can be condensed to a form that depends directly on the blocked inputs, removing the equality constraints for move blocking. A further reduction may be achieved by condensation of the state prediction variables. The condensed problems defined in the following subsections can be used when implementing Algorithm 4.1.

### 4.5.1 Input Condensed Prediction Model

Define the input condensed robust optimisation problem  $\tilde{\mathcal{P}}_{C_1}(S)$ , for  $S \in \mathcal{A}(N_S, r_s)$  finding the minimum

$$J[\tilde{\mathcal{P}}_{C_1}(S)] = \min_{\{v(S, \cdot)\}} \sum_{j=0}^{N_S-1} (1 + \|\Theta(y(k+j|k) - y_r)\|), \quad (4.85)$$

subject to the dynamics constraints

$$x(k|k) = x(k) \quad (4.86a)$$

$$x(k+j+1|k) = Ax(k+j|k) + Bv(S, \sigma(S, j)) \quad (4.86b)$$

$$y(k+j|k) = Cx(k+j|k) + Dv(S, \sigma(S, j)), \quad (4.86c)$$

and the tightened state and input constraints

$$v(S, q) \in \mathcal{U}(\Lambda(S, q) - 1) \quad (4.87a)$$

$$x(k+j|k) \in \mathcal{X}(S, j) \quad (4.87b)$$

$$x(k+N_S|k) \in \mathcal{T}(S, N_S), \quad (4.87c)$$

where the function  $\sigma(\cdot, \cdot)$  (4.17) is used to return the block number corresponding to  $j$ .

## 4.5.2 State & Input Condensed Prediction Model

In order to condense the state variables, define parameter-varying matrices by the recursion

$$\bar{A}(0) = I_n, \quad \bar{A}(i+1) = A\bar{A}(i) \quad (4.88)$$

$$\bar{B}(0) = 0_{n \times m}, \quad \bar{B}(i+1) = A\bar{B}(i) + B. \quad (4.89)$$

Define also the condensed state prediction variable for block  $q$  by  $\phi(S, q)$ . Then, define the input and state condensed robust optimisation problem  $\tilde{\mathcal{P}}_{C_2}(S)$ , for  $S \in \mathcal{A}(N_S, r_s)$  as finding the minimum

$$J[\tilde{\mathcal{P}}_{C_2}(S)] = \min_{\{v(S, \cdot)\}} \sum_{j=0}^{N_S-1} (1 + \|\Theta(y(k+j|k) - y_r)\|), \quad (4.90)$$

subject to the dynamics constraints

$$\phi(S, 1) = x(k) \quad (4.91a)$$

$$\phi(S, q+1) = \bar{A}(\ell(S, q))\phi(S, q) + \bar{B}(\ell(S, q))v(S, q) \quad (4.91b)$$

$$y(k+j|k) = C(\bar{A}(j-t)\phi(S, \sigma(S, j)) + \bar{B}(j-t)v(S, \sigma(S, j))) + Dv(S, \sigma(S, j)), \quad (4.91c)$$

and the tightened state and input constraints

$$v(S, q) \in \mathcal{U}(\Lambda(S, q) - 1) \quad (4.92a)$$

$$\bar{A}(i)\phi(S, q) + \bar{B}(i)v(S, q) \in \mathcal{X}(S, \Lambda(S, q-1) + i), \forall i \in \mathbb{Z}_{[0, \ell(S, q)-1]} \quad (4.92b)$$

$$\phi(S, r_s+1) \in \mathcal{T}(S, N_S), \quad (4.92c)$$

where  $q \in \mathbb{N}_{[1, r_s]}$  and  $t = \mu(S, j) - \ell(S, \sigma(S, j)) + 1$  is the block-initial index.

**Remark 4.20.** It is possible to condense the entire problem, removing all equality constraints. The resulting problem will then depend only on the initial state and the block values (a so-called dense MPC formulation). The degree of condensation that gives the greatest computational benefit will depend on the nature of the system and the solver used, as some solvers can exploit the sparsity of a problem in order to quicken computation.

## 4.6 Example

From the general controller specification described in the previous sections, there are many possible ways of choosing the subset of  $\mathcal{S}(\bar{N}, \bar{r})$ . The simplest method is to subdivide the entire horizon into equal-length blocks and have the regime consist of matrices corresponding to multiples of blocks. For some applications, however, it may be beneficial to utilise the flexibility of the formulation to apply input blocking in only certain parts of the horizon. A particular method is presented in this section, as may be applicable to vehicle manoeuvring problems.

### 4.6.1 The Partitioned Horizon

The partitioned horizon regime involves blocking matrices where no blocking is applied at the start and end of the manoeuvre, but only in the middle. This is equivalent to partitioning the prediction horizon into three sections that will be referred to as the initial, medial and terminal sub-horizons. By applying move blocking only in the medial sub-horizon, a reduction in complexity is achieved whilst maintaining fine-grained control action at the start and end of a control manoeuvre, as is pertinent to an application like spacecraft rendezvous. This formulation is inspired by the *rubber horizon* concept presented in [Har10]. It differs from the latter by explicitly ensuring pointwise-in-time constraint satisfaction within the medial sub-horizon.

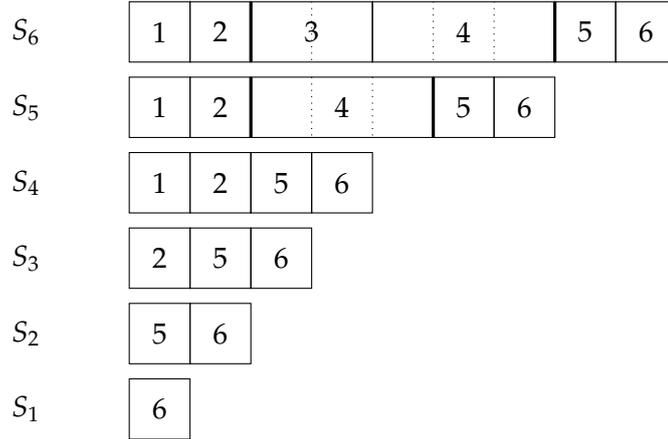
### 4.6.2 Formulation

Define  $\Omega_1, \Omega_3 : \Omega_1 + \Omega_3 < \bar{N}$  as the lengths of the initial and final sub-horizons respectively. Also define  $\Omega_2 = \bar{N} - \Omega_1 - \Omega_3$ . Assume that a block length of  $\lambda$  is desired in the medial sub-horizon. The partitioned horizon regime comprises blocking matrices corresponding to all horizon lengths unblocked up to  $\Omega_1 + \Omega_3$  and subsequently adds blocks of length  $\lambda$  to the medial sub-horizon until the final blocking matrix corresponds to  $\bar{N}$  variables. The final block added may be shorter than  $\lambda$  if  $(\Omega_2 \bmod \lambda) \neq 0$ . The sequence of blocking matrices  $S_i$  in the regime can be expressed mathematically as

$$S_i = \begin{cases} I_i & 1 \leq i \leq \Omega_1 + \Omega_3 \\ I_{\Omega_1} \oplus (I_{i-\Omega_1-\Omega_3} \otimes \mathbf{1}_\lambda) \oplus I_{\Omega_3} & \Omega_1 + \Omega_3 < i < \bar{i} \\ I_{\Omega_1} \oplus \mathbf{1}_\xi \oplus (I_{i-\Omega_1-\Omega_3-1} \otimes \mathbf{1}_\lambda) \oplus I_{\Omega_3} & i = \bar{i}, \end{cases} \quad (4.93)$$

where  $\bar{i} = \Omega_1 + \Omega_3 + \lceil \Omega_2 / \lambda \rceil$  is the total number of blocking matrices in the sequence and  $\xi = ((\Omega_2 - 1) \bmod \lambda) + 1$  is the length of the first block in the medial sub-horizon when  $i = i_{max}$ . Due to the construction of the regime,  $\bar{i}$  also corresponds to the maximum number of input variable blocks in any single optimisation problem when the regime is applied.

Then, Algorithm 4.1 is applied to the set of optimisation problems defined by the blocking matrices  $S_i$ . The relaxation applied before shifting is defined as follows. When move blocking



**Figure 4.2:** Illustration of the partitioned horizon regime

is applied, the blocking matrix  $S \in \mathcal{A}(N, r)$  can be written in the form

$$S = I_{\Omega_1} \oplus \Delta \oplus I_{\Omega_3}, \quad (4.94)$$

where  $\Delta \in \mathcal{A}(N - \Omega_1 - \Omega_3, r - \Omega_1 - \Omega_3)$  is the blocking matrix corresponding to the move-blocked part of  $S$ . Then, define the relaxation function as

$$\rho(S) = \begin{cases} I_{\Omega_1} \oplus 1 \oplus F(\Delta) \oplus I_{\Omega_3} & \ell(\Delta, 1) > 1 \\ S & \text{otherwise.} \end{cases} \quad (4.95)$$

The relaxation unblocks the first element of the medial horizon, which has the effect of only shifting the medial horizon when  $F(\rho(S))$  is calculated. By using this blocking and relaxation scheme, Theorem 4.19 ensures that length of the medial sub-horizon in the matrix  $S^*$  reduces with time, ultimately vanishing. This means that the controller behaves like conventional VH-MPC formulation after a finite number of steps. Figure 4.2 illustrates the regime for a simple example, where  $\bar{N} = 9$ ,  $\Omega_1 = 2$  and  $\Omega_3 = 2$ . The structure of  $i$ th blocking matrix  $S_i$  is shown graphically by the length of the blocks. Note that the medial sub-horizon lies between the thick lines.

**Remark 4.21.** As the initial sub-horizon is unblocked, by choosing a constraint-tightening policy with at most degree- $\Omega_1$  nilpotency, the tightened sets corresponding to the states in the medial and terminal sub-horizons will all be the same. This means that the calculations described by (4.54c) and (4.54d) simplify significantly, allowing standard constraint tightening calculations to be used.

### 4.6.3 Simulation

To illustrate the action of the partitioned horizon, consider a point mass moving in two dimensions, specified by the 1 Hz discretised dynamics

$$x(k+1) = Ax(k) + Bu(k) + w(k), \quad (4.96)$$

where

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (4.97)$$

and the disturbance satisfies  $\|w(k)\|_\infty \leq 0.45$ . The control problem is to reach the target set

$$\left\{ x \mid \begin{bmatrix} 200 & 60 & -2 & -2 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 202 & 62 & 2 & 2 \end{bmatrix}^T \right\} \quad (4.98)$$

in finite time  $N$ , starting from the origin, whilst ideally minimising the cost function

$$\sum_{k=0}^{N-1} 1 + 0.3 \|u(k)\|_1, \quad (4.99)$$

subject to  $\|u(k)\|_\infty \leq 2$  and  $\|[0 \ I_2]x(k)\|_\infty \leq 5$  at all time instants up to completion.

Choosing  $\bar{N} = 50$  and  $\Omega_1 = \Omega_3 = 5$ , and using a state and input condensed formulation, the system is simulated<sup>1</sup> for different values of  $\lambda$  over 100 randomly generated admissible disturbance realisations. A two-step nilpotent policy is used for tightening constraints, which minimises the number of Pontryagin set difference calculations that are required. The MPC optimisation problems are solved with a dual-simplex algorithm.

Figure 4.3 shows the effect of different block lengths in the medial sub-horizon on the mean closed-loop cost. The error bars on the plot indicate one standard deviation. Figure 4.4 illustrates the maximum computation times at each solution time step, for serial and parallel evaluations of the optimisation problems. The computation times have been normalised to the maximum serial solution time per time step of the unblocked problem (0.1644 s). Figures 4.5-4.7 show the closed loop trajectories for selected values of  $\lambda$ , whilst Figures 4.8-4.10 show the input components.

#### 4.6.4 Analysis

It can be seen that the cost is not monotonic with respect to the number of decision variables, indicating the importance of the blocking structure itself on the closed-loop performance under disturbances. The fact that the mean cost reduces with reduced complexity of the problem may at first seem counter-intuitive. However, a possible explanation is the action of the move blocking, which seems to be affecting the aggressiveness of the control action in responding to the disturbance. From the control input plots, it is clear that the lower cost is associated with less aggressive control action being applied. However, after the block lengths are increased beyond a certain point, the action becomes more aggressive. The maximum latency in both the serial and parallel cases reduces approximately linearly with a reduction in the number of

---

<sup>1</sup>Simulations were performed on an 8-core Mac Pro 3.1 running OS X 10.6.8.

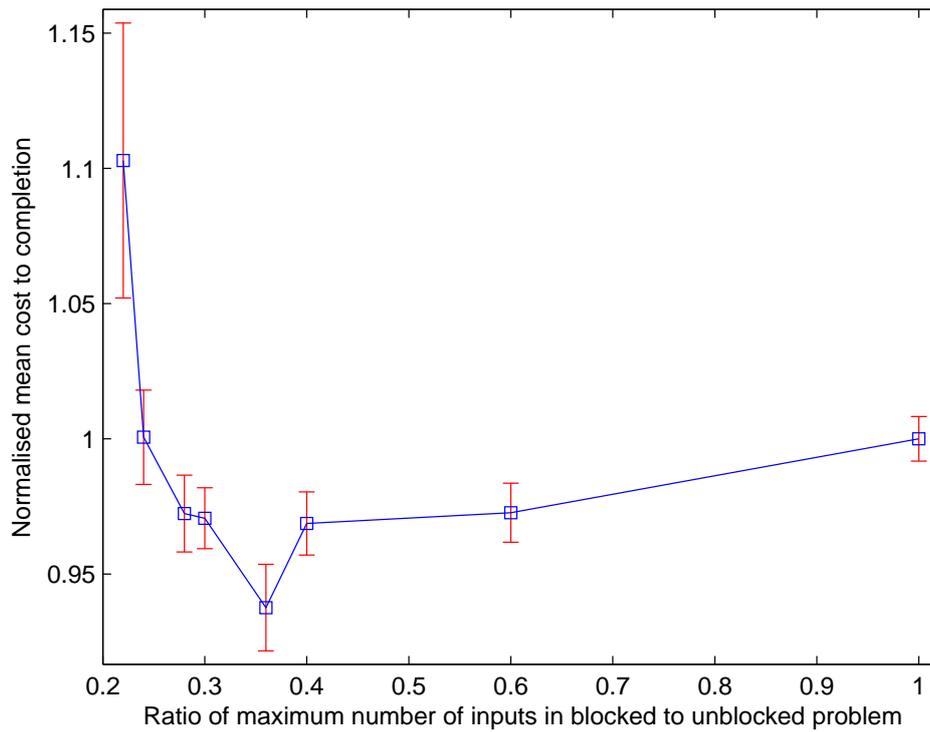


Figure 4.3: Cost comparison of different block lengths in the partitioned horizon

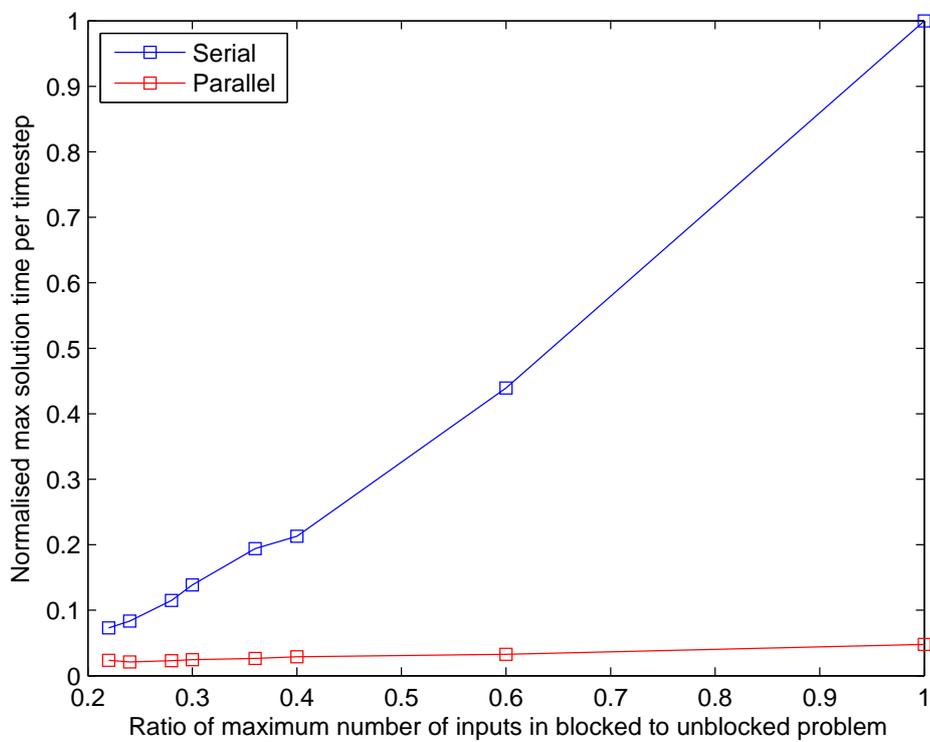
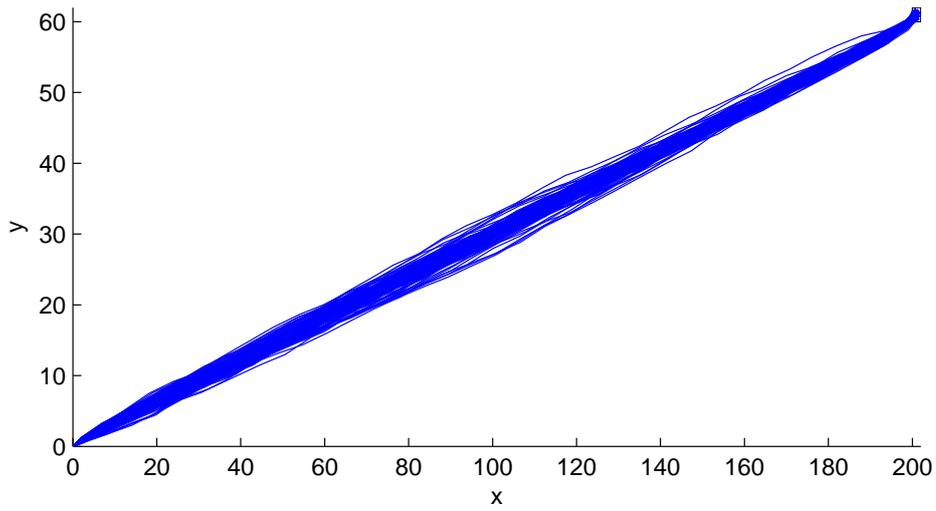
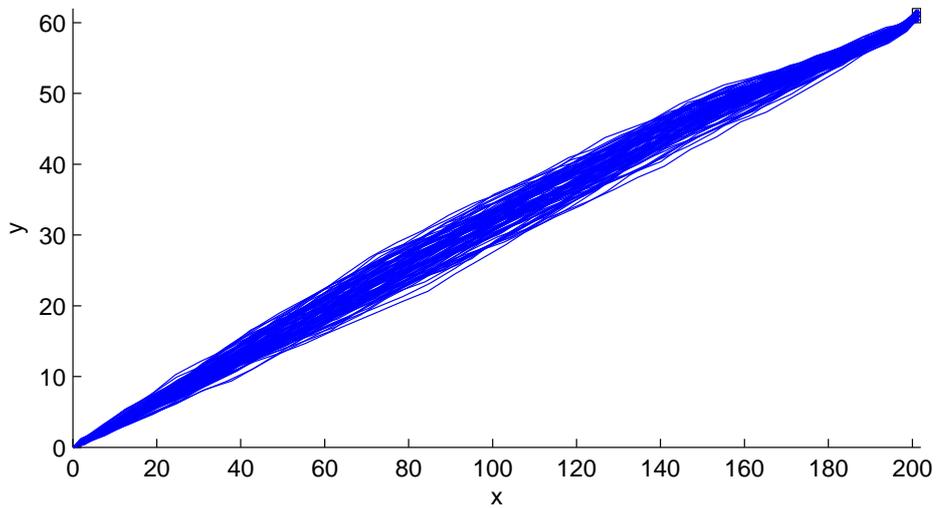


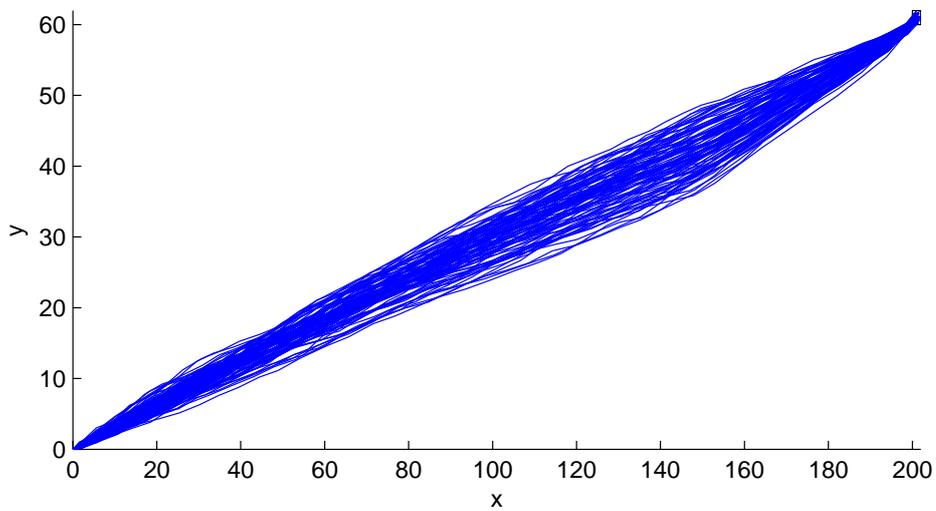
Figure 4.4: Computation time comparison different block lengths in the partitioned horizon



**Figure 4.5:** Closed-loop trajectories for  $\lambda = 1$



**Figure 4.6:** Closed-loop trajectories for  $\lambda = 4$



**Figure 4.7:** Closed-loop trajectories for  $\lambda = 40$

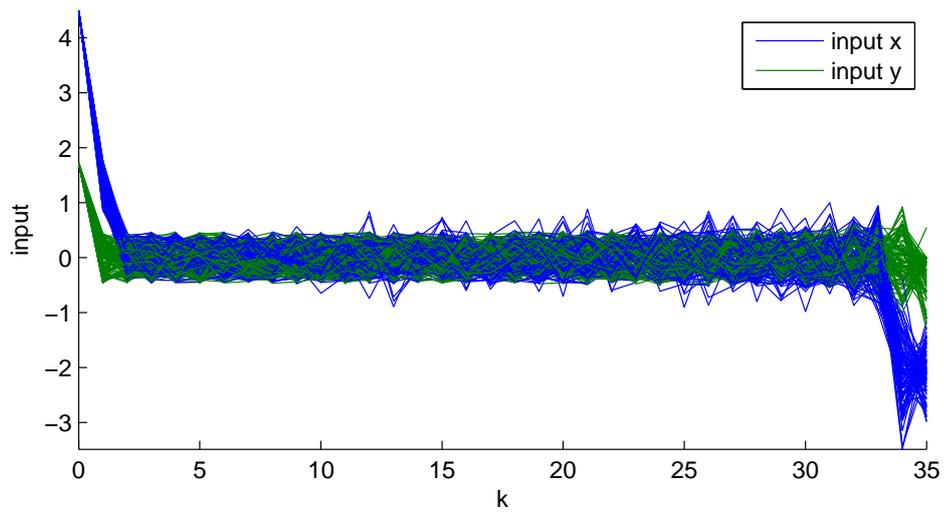


Figure 4.8: Control inputs for  $\lambda = 1$

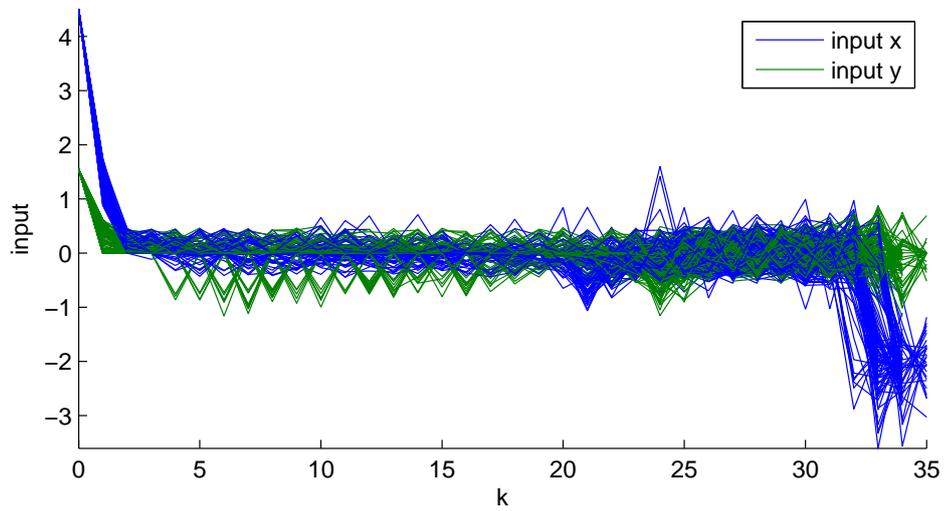


Figure 4.9: Control inputs for  $\lambda = 4$

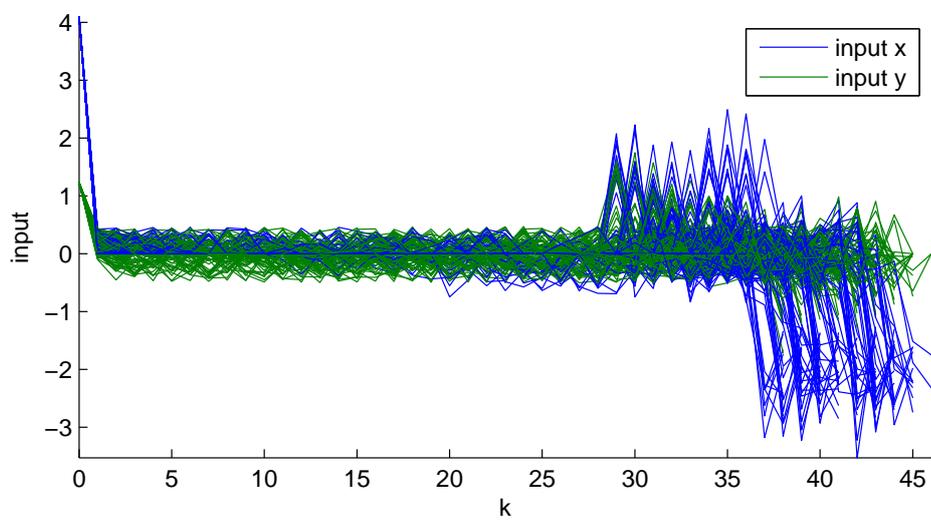


Figure 4.10: Control inputs for  $\lambda = 40$

input variables. These latencies give an indication of what the maximum sampling rate of the system can be when considering real-time implementation.

### 4.7 Conclusion

This chapter has developed a new robust formulation of VH-MPC using move blocking to reduce computational complexity. The formulation has been shown to preserve the guarantees of recursive feasibility and finite-time completion in the presence of bounded disturbances. Using a particular choice of blocking regime, namely the partitioned horizon, the formulation has been demonstrated on an example system.

### 4.8 Acknowledgements

The work in this chapter has been published in the journal *Systems & Control Letters* [SM12]. The simulations in this chapter were carried out using Matlab R2009b, YALMIP [Löf04], GLPK [Glp] and the Invariant Set Toolbox [Ker00].

## Chapter 5

# Optimal Constraint Tightening Policies

### 5.1 Introduction

As constraint tightening achieves robustness by restricting the size of constraint sets at future steps into the prediction horizon, this will result in a reduction in the size of the controller's Region of Attraction (ROA), which is the set of states for which an initial feasible solution exists to the MPC optimisation problem. In the formulations of constraint tightening presented in [RH03], [RH06a], no restriction is placed on the policy for the guarantees of robust recursive feasibility and finite-time completion to hold in the closed loop. Of course, it is necessary that the chosen policy will not cause any of the tightened sets to become empty, which would result in the controller's ROA being empty. For convenience in calculating tightened sets, nilpotent policies have been proposed; for this reason, such policies have been used in the preceding chapters. However, these policies may not be the best choice if a large size ROA is desired; the degree of freedom in choosing the CT policy suggests that optimisation is possible.

Kuwata, Richards and How [KRH07] demonstrate that a disturbance feedback parameterisation allows the constraint tightening policy to form part of a convex optimisation problem. The problem aims to find the constraint tightening policy able to tolerate strong disturbances, for a fixed prediction horizon. Essentially it maximises a scaling factor on the disturbance set, whilst still ensuring non-emptiness of the tightened constraint sets, as well as the existence of a non-empty robust control-invariant terminal set. A numerical example shows how the resulting policy attenuates the size of the region of attraction less severely than policies selected in other ways, as the disturbance set is dilated.

This chapter builds upon the work of [KRH07], but considers a fundamentally different criterion. It demonstrates how, given a fixed polytopic disturbance set, constraint-tightening policies parameterised in terms of disturbance feedback can be optimised directly to increase the volume of the controller's ROA. Furthermore, this chapter analyses the variable horizon case, so there is no requirement on terminal set invariance. The variable horizon does, however, require approximations to be made, given the non-convexity of the problem. An inner ellipsoidal approximation to a fixed horizon ROA is maximised, which is an inner approximation to the

non-convex (in general) union of convex sets that defines the full ROA over all horizon lengths (denoted the Variable Horizon ROA or VH-ROA). This inner approximation is shown to be effective in the numerical examples that are used to demonstrate the technique.

The examples show that, compared to using nilpotent policies or other heuristics, the policy resulting from the optimisation method presented in this chapter produces the largest domain of attraction volume. Furthermore, it is shown that this policy can drastically reduce the horizon length required for the ROA to contain a given subset of the state space. The effect of the policy on closed-loop cost performance is also investigated.

### 5.1.1 Nomenclature

For any matrix  $A$  (including vectors), the notation  $A_i^T$  represents its  $i$ th row. The hypervolume of a given polytope  $\mathcal{C}$ , or its Lebesgue measure, is denoted  $\text{vol}(\mathcal{C})$ . The projection of a polyhedron  $\mathcal{A} \subset \mathbb{R}^n$  onto the subspace  $\mathbb{R}^m$ , where  $m < n$  is denoted by

$$\mathbb{P}_m(\mathcal{A}) = \{x \in \mathbb{R}^m \mid \exists u \in \mathbb{R}^{(n-m)} : (x, u) \in \mathcal{A}\}. \quad (5.1)$$

A  $p$ -norm ball in  $n$  dimensions with radius  $\epsilon$  is defined by the set

$$\mathbb{B}_p^n(\epsilon) = \{x \in \mathbb{R}^n \mid \|x\|_p \leq \epsilon\}. \quad (5.2)$$

## 5.2 Problem Formulation

As in the last two chapters, consider a discrete-time linear system with states  $x(k) \in \mathbb{R}^n$ , inputs  $u(k) \in \mathbb{R}^m$  and outputs  $y(k) \in \mathbb{R}^p$  specified by the difference equation

$$x(k+1) = Ax(k) + Bu(k) + w(k) \quad (5.3)$$

$$y(k) = Cx(k) + Du(k), \quad (5.4)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$  and  $D \in \mathbb{R}^{p \times m}$ . It is assumed that the pair  $(A, B)$  is controllable. The system is subject to pointwise-in-time output constraints of the form

$$y(k) \in \mathcal{Y} = \{y \mid Ey \leq f, E \in \mathbb{R}^{q \times p}, f \in \mathbb{R}^q\}, \quad (5.5)$$

where  $\mathcal{Y}$  is a polytope defined by  $q$  individual half-space constraints. In terms of this set, define the output admissible set

$$\mathcal{O} = \mathbb{P}_n(\{(x, u) : E(Cx + Du) \leq f\}). \quad (5.6)$$

The control objective is to steer the state to the target set

$$\mathcal{T} = \{x \mid Gx \leq h, G \in \mathbb{R}^{r \times n}, h \in \mathbb{R}^r\}, \quad (5.7)$$

which is a polytope comprising  $r$  individual half-space constraints. This set is to be reached in  $N < \infty$  steps, expressed as the constraint

$$x(N) \in \mathcal{T}. \quad (5.8)$$

The polytopically-bounded state disturbance satisfies

$$w(k) \in \mathcal{W} = \{w \mid \zeta w \leq \theta\}, \quad (5.9)$$

for a given matrix and vector  $\zeta \in \mathbb{R}^{a \times n}$  and  $\theta \in \mathbb{R}^a$  respectively. This set is assumed to contain the origin. The control problem is to steer the system state to  $\mathcal{T}$  in finite time, whilst minimising a given cost function.

**Remark 5.1.** In this chapter, the output constraints are assumed to be polytopic to ensure that the ROA is compact. If this assumption were not made, certain system dynamics (e.g. open-loop nilpotent systems) could result in an unbounded ROA.

A disturbance-feedback constraint-tightened predictive controller is designed to address the control problem. Recursively define the matrices

$$L(0) = I_n \quad (5.10a)$$

$$L(j+1) = AL(j) + BP(j) \quad (5.10b)$$

$$Q(j) = CL(j) + DP(j) \quad (5.10c)$$

and the corresponding sets

$$\mathcal{Y}(0) = \mathcal{Y} \quad (5.11a)$$

$$\mathcal{T}(0) = \mathcal{T} \quad (5.11b)$$

$$\mathcal{Y}(j+1) = \mathcal{Y}(j) \ominus Q(j)\mathcal{W} \quad (5.11c)$$

$$\mathcal{T}(j+1) = \mathcal{T}(j) \ominus L(j)\mathcal{W}. \quad (5.11d)$$

As in previous chapters, the sequence of matrices

$$\mathcal{P} = \{P(j) \in \mathbb{R}^{n \times m}\}, \quad (5.12)$$

is the constraint tightening policy providing direct feedback on the disturbance. The prediction model is then defined by the dynamics

$$x(k+j+1|k) = Ax(k+j|k) + Bu(k+j|k) \quad (5.13a)$$

$$y(k+j|k) = Cx(k+j|k) + Du(k+j|k), \quad (5.13b)$$

subject to the constraints

$$y(k+j|k) \in \mathcal{Y}(j) \quad (5.14a)$$

$$x(k + N(k)|k) \in \mathcal{T}(N(k)). \quad (5.14b)$$

### 5.3 Affine Parameterisation of Tightened Sets

In order to optimise over the disturbance feedback matrices, the Pontryagin difference operations (5.11c)-(5.11d) need to be parameterised as affine functions of the policy matrices  $P(\cdot)$ . In order to accomplish this, define the sequences of vectors  $\{s(j) \in \mathbb{R}^q\}_{j=0}^{N-1}$  and  $\{\sigma(j) \in \mathbb{R}^r\}_{j=0}^N$ , as well as the sequences of matrix variables  $\{Z(j) \in \mathbb{R}^{q \times q}\}_{j=0}^{N-1}$  and  $\{\bar{Z}(j) \in \mathbb{R}^{r \times r}\}_{j=0}^N$ , for some fixed horizon length  $N$ .

**Theorem 5.2.** *For any output  $y$  and state  $x$ , for all  $j \in \mathbb{Z}_{[0, N-1]}$ ,*

$$y \in \mathcal{Y}(j) \iff \exists \{Z(l) \geq 0\}_{l=0}^{j-1} : Ey \leq f - s(j), \quad (5.15)$$

and for all  $j \in \mathbb{Z}_{[0, N]}$

$$x \in \mathcal{T}(j) \iff \exists \{\bar{Z}(l) \geq 0\}_{l=0}^{j-1} : Gx \leq h - \sigma(j), \quad (5.16)$$

where

$$s(0) = 0_{q \times 1} \quad (5.17a)$$

$$s(j+1) = s(j) + Z^T(j)\theta \quad (5.17b)$$

$$EQ(j) = Z^T(j)\zeta \quad (5.17c)$$

and

$$\sigma(0) = 0_{r \times 1} \quad (5.18a)$$

$$\sigma(j+1) = \sigma(j) + \bar{Z}^T(j)\theta \quad (5.18b)$$

$$GL(j) = \bar{Z}^T(j)\zeta. \quad (5.18c)$$

*Proof.* The theorem is proved by induction on  $j$  for the tightened output sets. The proof for the terminal sets is completely analogous. Then, for the base case, it is clear that

$$y \in \mathcal{Y}(0) \iff Ey \leq f - s(0) = f, \quad (5.19)$$

as  $s(0) = 0_{q \times 1}$  and there is no dependence on  $Z(\cdot)$ . Now, assume that for any  $j$ ,

$$y \in \mathcal{Y}(j) \iff \exists \{Z(l) \geq 0\}_{l=0}^{j-1} : Ey \leq f - s(j), \quad (5.20)$$

where the constraints (5.17) are satisfied. Then, using (5.11c) and the definition of the Pontryagin difference (2.55),

$$y \in \mathcal{Y}(j+1) \iff y + b \in \mathcal{Y}(j), \forall b \in Q(j)\mathcal{W} \quad (5.21)$$

which, using the induction assumption, means that there exists  $Z(0), \dots, Z(j-1) \geq 0$  satisfying (5.17) such that

$$E(y + b) \leq f - s(j), \forall b \in Q(j)\mathcal{W} \quad (5.22)$$

$$Ey + EQ(j)w \leq f - s(j), \forall w \in \mathcal{W}. \quad (5.23)$$

The universal quantifier can then be eliminated to give

$$Ey + \max_{w \in \mathcal{W}} EQ(j)w \leq f - s(j) \quad (5.24)$$

where the maximisation is taken row-wise. Given that each row-wise maximisation defines a linear program, its (asymmetric) dual problem can be exploited as in [GKM06]. Hence, for the  $i$ th row,

$$\max_{w \in \mathcal{W}} E_i^T Q(j)w = \min_{z_i} z_i^T \theta \quad (5.25)$$

subject to

$$E_i^T Q(j) = z_i^T \zeta \quad (5.26)$$

$$z_i \geq 0, \quad (5.27)$$

for dual variable  $z_i \in \mathbb{R}^{1 \times a}$ . The minimisation can then be removed by observing that, for the  $i$ th row of (5.24),

$$E_i^T y + \min_{z_i} z_i^T \theta \leq f_i - s_i(j) \iff \exists z_i \geq 0 : E_i^T y + z_i^T \theta \leq f_i - s_i(j). \quad (5.28)$$

Applying the same reasoning to all  $i \in \mathbb{N}_{[1,q]}$ , the dual variables can be combined to give

$$Ey + \max_{w \in \mathcal{W}} EQ(j)w \leq f - s(j) \iff \exists Z(j) \geq 0 : Ey + Z^T(j)\theta \leq f - s(j), \quad (5.29)$$

where

$$Z(j) = \bigoplus_{i=1}^q z_i \quad (5.30)$$

$$EQ(j) = Z^T(j)\zeta. \quad (5.31)$$

Hence, there exists  $Z(0), \dots, Z(j) \geq 0$  satisfying (5.17) such that

$$Ey \leq f - (s(j) + Z^T(j)\theta) \quad (5.32)$$

$$\leq f - s(j+1). \quad (5.33)$$

which demonstrates that

$$y \in \mathcal{Y}(j+1) \iff \exists \{Z(l)\}_{l=0}^j : Ey \leq f - s(j+1), \quad (5.34)$$

if the conditions (5.17) are satisfied. The same method can be used to show the result for the tightened terminal set constraints, so the proof is omitted for brevity. ■

This parameterisation goes further than [KRH07], where the tightened output set  $\mathcal{Y}(N-1)$  is expressed in terms of a series of affine inequalities based on the vertices of the disturbance set, which can become computationally burdensome for higher-dimensional polytopes. Furthermore, the assumption  $0 \in \mathcal{Y}$  is made in [KRH07], whereas no such assumption is required in Theorem 5.2.

Corollary 5.3 shows that the RHS of the polytopes defining the output and terminal constraints in Theorem 5.2 can never increase, which means that the constraint sets can never grow. This is expected, given the action of the repeated P-difference operations. The result also verifies the fact that the sets remain compact under tightening.

**Corollary 5.3.** *In the affine parameterisation presented in Theorem 5.2, the variables  $s(\cdot)$  and  $\sigma(\cdot)$  are always monotonically increasing in  $j$ .*

*Proof.* This result arises from the fact that the disturbance set must contain the origin and the matrices  $Z(\cdot)$  and  $\bar{Z}(\cdot)$  are element-wise non-negative. The former condition implies that  $\theta \geq 0$ , which, combined with the latter condition ensures that  $Z^T(j)\theta \geq 0$  and  $\bar{Z}^T(j)\theta \geq 0$ , for all  $j$ . These conditions can then be combined with recurrence relationships for  $s(\cdot)$  and  $\sigma(\cdot)$  defined in (5.17) - (5.18) to show that  $s(\cdot)$  and  $\sigma(\cdot)$  increase monotonically. ■

## 5.4 Maximising the Region of Attraction

Having defined the tightened sets as affine functions of the constraint tightening policy, the regions of attraction for fixed horizon lengths can also be parameterised in terms of this policy. This then allows a measure of the volume of this region to be optimised.

### 5.4.1 Parameterisation

Defining the block matrix

$$\Phi = \begin{bmatrix} EC & ED & 0 & \cdots & 0 & 0 \\ ECA & ECB & ED & \cdots & 0 & 0 \\ ECA^2 & ECAB & ECB & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ ECA^{N-1} & ECA^{N-2}B & ECA^{N-3}B & \cdots & ECB & ED \\ GA^N & GA^{N-1}B & GA^{N-2}B & \cdots & GAB & GB \end{bmatrix}, \quad (5.35)$$

and vectors

$$v = \begin{bmatrix} x(0) \\ \mathbf{u} \end{bmatrix}, \mathbf{u} = \begin{bmatrix} u(0|0) \\ u(1|0) \\ \vdots \\ u(N-1|0) \end{bmatrix}, \boldsymbol{\psi} = \begin{bmatrix} f \\ f \\ \vdots \\ f \\ h \end{bmatrix}, \mathbf{s} = \begin{bmatrix} s(0) \\ s(1) \\ \vdots \\ s(N-1) \\ \sigma(N) \end{bmatrix}, \quad (5.36)$$

the following lemma shows how the tightened operating constraints can be condensed into half-space constraints on the initial states and sequence of inputs.

**Lemma 5.4.** *For a given fixed horizon length  $N$ , the constraints (5.14) on the initial MPC optimisation problem can be expressed in the form*

$$\Phi v \leq \boldsymbol{\psi} - \mathbf{s}, \quad (5.37)$$

$$\begin{aligned} \exists \{Z(j) \geq 0_{a \times q}\}, \{\bar{Z}(j) \geq 0_{a \times r}\} : \\ \mathbf{s} \text{ satisfies (5.17) and (5.18)} \end{aligned} \quad (5.38)$$

*Proof.* This result is obtained by condensing the states in the dynamics equations (5.13) at time step  $k = 0$  and enforcing the affine constraints derived in Theorem 5.2 on the resulting outputs. Specifically,

$$\begin{aligned} y(0|0) \in \mathcal{Y}(0) &\iff E(Cx(0) + Du(0|0)) \leq f - s(0) \\ y(1|0) \in \mathcal{Y}(1) &\iff E(C(Ax(0) + Bu(0|0)) + Du(1|0)) \leq f - s(1) \\ y(2|0) \in \mathcal{Y}(2) &\iff E(C(A^2x(0) + ABu(0|0) + Bu(1|0)) + Du(2|0)) \leq f - s(2) \\ &\vdots \\ y(N-1|0) \in \mathcal{Y}(N-1) &\iff E(C(A^{N-1}x(0) + A^{N-2}Bu(0|0) + \dots + Bu(N-2|0)) \\ &\quad + Du(N-1|0)) \leq f - s(N-1) \\ x(N|0) \in \mathcal{T}(N) &\iff G(A^Nx(0) + A^{N-1}Bu(0|0) + \dots + Bu(N-1|0)) \leq h - \sigma(N), \end{aligned}$$

where  $\mathbf{s}$  satisfies (5.17) and (5.18). ■

In terms of this constraint set, Theorem 5.5 quantifies the fixed horizon region of attraction for a given horizon length  $N$ .

**Theorem 5.5.** *The set of states  $\mathcal{R}(N, \mathcal{P})$  for which an initial  $N$ -step feasible solution exists, for CT policy  $\mathcal{P}$  is given by*

$$\mathcal{R}(N, \mathcal{P}) = \left\{ x \left| \begin{array}{l} \hat{\Phi}x \leq \hat{\boldsymbol{\psi}} - \Gamma \mathbf{s} \\ \exists \{Z(j) \geq 0_{a \times q}\}, \{\bar{Z}(j) \geq 0_{a \times r}\} : \\ \mathbf{s} \text{ satisfies (5.17) and (5.18)} \end{array} \right. \right\}, \quad (5.39)$$

where the matrices  $\Gamma \in \mathbb{R}^{t \times (d-n)}$  and  $\hat{\Phi} \in \mathbb{R}^{t \times n}$ , as well as vector  $\hat{\boldsymbol{\psi}} \in \mathbb{R}$  define the projection of the

polyhedron

$$\mathcal{A} = \left\{ (s, v) \mid \begin{bmatrix} I & \Phi \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} \leq \psi \right\} \quad (5.40)$$

onto the subspace  $\mathbb{R}^d$ , where

$$d = (N - 1)q + r + n. \quad (5.41)$$

That is,

$$\left\{ (s, x(0)) \in \mathbb{R}^d \mid \begin{bmatrix} \Gamma & \hat{\Phi} \end{bmatrix} \begin{bmatrix} s \\ x(0) \end{bmatrix} \leq \hat{\psi} \right\} \triangleq \mathbb{P}_d(\mathcal{A}). \quad (5.42)$$

*Proof.* The constraint (5.37) derived in Lemma 5.4 can be rewritten in the form

$$\Phi \begin{bmatrix} x(0) \\ u \end{bmatrix} + s \leq \psi \quad (5.43)$$

$$\implies \begin{bmatrix} I & \Phi \end{bmatrix} \begin{bmatrix} s \\ \begin{bmatrix} x(0) \\ u \end{bmatrix} \end{bmatrix} \leq \psi \quad (5.44)$$

which defines the polyhedron  $\mathcal{A}$ . By taking its projection onto  $\mathbb{R}^d$ , what remains is a polyhedron defined by inequalities depending only on the initial state  $x(0)$  and the vector  $s$ . This polyhedron can be written in the form

$$\begin{bmatrix} \Gamma & \hat{\Phi} \end{bmatrix} \begin{bmatrix} s \\ x(0) \end{bmatrix} \leq \hat{\psi} \quad (5.45)$$

$$\implies \hat{\Phi}x(0) \leq \hat{\psi} - \Gamma s. \quad (5.46)$$

Combining these constraints with the requirements on the dual variables gives the desired result. ■

**Remark 5.6.** It is assumed that  $|\mathcal{P}| \geq N - 1$ . Any extra disturbance feedback gains are simply unused in the calculation of  $s(\cdot)$  and  $\sigma(\cdot)$ .

The projection operation in Theorem 5.5 can be implemented with the conceptually simple Fourier-Motzkin elimination technique [DE73], or alternatively, a less computationally intensive, but conceptually complex algorithm such as Equality Set Projection (ESP) [JKM04]. Using ESP will require an artificial bounding of the polyhedron  $\mathcal{A}$  to convert it into a polytope. This can be achieved by adding extra constraints of the form  $s < M\mathbf{1}_{(d-u)}$ , as these are the only unbounded variables in the absence of the dual constraints.  $M$  needs to be chosen large enough that it does not interfere with the other constraints on the dual variables. A sufficient condition is to choose  $M \geq \max\{\text{diam}(\mathcal{Y}), \text{diam}(\mathcal{T})\}$ . Regardless of the projection method used, the computational burden is only incurred offline. Note that it is possible to project away the dual variables as well, however, in the interests of reducing computational complexity, they are retained to be defined explicitly in subsequent optimisation problems.

### 5.4.2 Optimisation

The volume of a general polytope has no closed-form expression, so instead, optimisation focuses on maximising the volume of an inner ellipsoidal approximation to  $\mathcal{R}(N, \mathcal{P})$ . Consider an ellipsoid specified by

$$\mathcal{E}(c, \Omega) = \{x \in \mathbb{R}^n \mid (x - c)^T (\Omega^T \Omega)^{-1} (x - c) \leq 1\}, \quad (5.47)$$

with centre  $c \in \mathbb{R}^n$  and shape matrix  $\Omega \in \mathbb{R}^{n \times n}$ . A common assumption is that  $\Omega \succ 0$  [ZG03]. Then, Theorem 5.7 states the conditions under which this ellipsoid is contained within the region of attraction.

**Theorem 5.7.**  $\mathcal{E}(c, \Omega) \subset \mathcal{R}(N, \mathcal{P})$  iff

$$\hat{\Phi}c + \gamma(\mathcal{E}) \leq \hat{\psi} - \Gamma s \quad (5.48a)$$

$$\begin{aligned} \exists \{Z(j) \geq 0_{a \times q}\}, \{\bar{Z}(j) \geq 0_{a \times r}\} : \\ s \text{ satisfies (5.17) and (5.18)} \end{aligned} \quad (5.48b)$$

where

$$\gamma(\mathcal{E}) = \left( \begin{array}{c} t \\ \mathcal{E} \\ \| \Omega \hat{\Phi}_i \|_2 \end{array} \right)^T \in \mathbb{R}^t. \quad (5.49)$$

*Proof.* The proof uses the same method as in [ZG03], utilising the parameterised polytope defining the N-step ROA  $\mathcal{R}(N, \mathcal{P})$ . Firstly,  $\mathcal{E}(\cdot, \cdot)$  is re-expressed as an affine transformation on a unit hypersphere centred at the origin, giving the expression

$$\mathcal{E}(c, \Omega) = \{x \in \mathbb{R}^n \mid x = \Omega \zeta + c : \zeta \in \mathbb{R}^n, \|\zeta\|_2 \leq 1\}. \quad (5.50)$$

Then, it can be seen that all points within the ellipsoid satisfy the half-space constraint defined by the  $i$ th row of (5.46) iff

$$\hat{\Phi}_i^T (c + \Omega \zeta) \leq \hat{\psi} - \Gamma s, \forall \zeta : \|\zeta\|_2 = 1 \quad (5.51)$$

As in (5.24), the universal quantifier can be replaced by

$$\hat{\Phi}_i^T c + \sup_{\|\zeta\|_2=1} \hat{\Phi}_i^T \Omega \zeta \leq \hat{\psi}_i - \Gamma_i s. \quad (5.52)$$

Given that the magnitude of  $\zeta$  is fixed, the supremum is achieved when vectors  $\Omega \hat{\Phi}_i$  and  $\zeta$  are parallel. Hence

$$\hat{\Phi}_i^T c + \|\Omega \hat{\Phi}_i\|_2 \leq \hat{\psi}_i - \Gamma_i s. \quad (5.53)$$

Defining  $\gamma(\cdot)$  as in (5.49), the inequalities for all rows can be expressed in the form (5.48a). The result then follows by enforcing the conditions on the dual variables.  $\blacksquare$

**Theorem 5.8.** For a given  $N$ , the volume of an inner ellipsoidal approximation to  $\mathcal{R}(N, \mathcal{P})$  can be

maximised by solving the convex constrained optimisation problem

$$\mathcal{P}(N) = \begin{cases} \min_{c, \Omega, \mathcal{P}} -\log \det \Omega \\ \text{subject to (5.48) and } \Omega \succeq 0 \end{cases} \quad (5.54)$$

*Proof.* Given that the volume of an ellipsoid is proportional to the product of the lengths of its semi-axes, which correspond to the eigenvalues of  $\Omega$ , the volume is proportional to  $\det(\Omega)$ . The problem is made convex by taking logarithms; it is easy to show that the volume of the ellipsoid increases monotonically with  $\log \det(\Omega)$ . Constrained maximisation of this objective is a Semidefinite Program (SDP), specifically, a convex generalised eigenvalue problem, where all of the constraints (5.48) can be expressed in LMI form [Boy+94; BV04]. ■

**Remark 5.9.** If it is desirable to maximise the volume of the ROA around some given region  $\mathcal{G} \subseteq \mathcal{O}$ , then the constraint

$$c \in \mathcal{G} \quad (5.55)$$

can be enforced. In addition, if a nilpotent policy is desired for computational reasons, then constraints of the form

$$\begin{aligned} L(\eta) &= 0_{n \times n} \\ P(\eta + j) &= 0_{m \times n}, \forall j \geq 0 \end{aligned} \quad (5.56)$$

can be enforced, for degree- $\eta$  nilpotence. These constraints also reduce the complexity of the projection, as  $s(\eta + j) = s(\eta)$  and  $\sigma(\eta + j) = \sigma(\eta)$ , for all  $j > 0$ , allowing the dimension of  $s$  to be reduced. This will require the identity matrix in (5.44) to be replaced appropriately. It should be noted that enforcing either (5.55) or (5.56) can make it harder to find a feasible solution to the problem.

The constraint tightening policy that minimises (5.54), denoted  $\mathcal{P}^*(N)$  defines the policy that maximises an approximation to the volume of the  $N$ -step region of attraction. To use this in a variable horizon framework, where the maximum horizon length is  $\bar{N}$ , it remains to solve all problems  $\mathcal{P}(1), \dots, \mathcal{P}(\bar{N})$  and choose the policy with horizon length given by

$$N^* = \arg \min_{N \in \mathbb{N}_{[1, \bar{N}]}} \mathcal{P}(N) \quad (5.57)$$

where, with an abuse of notation,  $\min \mathcal{P}(\cdot)$  denotes the minimum value of the problem's objective function. Then, the policy  $\mathcal{P}_{\max\text{-vol}} = \mathcal{P}^*(N^*)$  will give the maximum ellipsoidal volume. Once the optimal policy has been found, the tightened sets do not need to be recalculated by repeated application of the P-difference operation. This is due to the fact that the values of  $s(\cdot)$  and  $\sigma(\cdot)$  that correspond to the optimal policy are already known after solving (5.54) and hence, can be subtracted from  $f$  and  $h$  respectively to immediately give the half-space representation of the tightened sets.

For any horizon length, the resulting policy ensures non-emptiness of the tightened sets, as shown in Corollary 5.10.

**Corollary 5.10.** *The optimal policy  $\mathcal{P}^*(N)$ , if it exists, ensures that the tightened sets*

$$\{\mathcal{Y}(j)\}_{j=0}^{N-1}, \quad (5.58)$$

*as well as the terminal set  $\mathcal{T}(N)$ , are non-empty.*

*Proof.* This can be easily seen by observing that for every  $x(0) \in \mathcal{R}(N, \mathcal{P}^*(N))$ , there exists a sequence of inputs  $\mathbf{u}$  such that, for any  $j \in \mathbb{N}_{[1, N-1]}$ ,

$$y(j|0) = Cx(j|0) + Du(j|0) \in \mathcal{Y}(j). \quad (5.59)$$

This input sequence also guarantees the existence of  $x(N|0)$  that satisfies

$$x(N|0) \in \mathcal{T}(N). \quad (5.60)$$

Hence, these tightened sets must be non-empty by construction. ■

**Remark 5.11.** Corollary 5.10 shows that, unlike in [KRH07], additional constraints do not need to be applied to ensure the non-emptiness of the state constraint sets and the terminal set when the policy is chosen according to (5.54). This formulation does not, however, guarantee that the sets are of nonzero measure. If such a guarantee is required for improving numerical conditioning, a sufficient condition is to ensure that there exist  $\epsilon_x, \epsilon_y > 0$  such that  $\mathbb{B}_p^p(\epsilon_x) \subset \mathcal{Y}(N-1)$  and  $\mathbb{B}_p^n(\epsilon_y) \subset \mathcal{T}(N)$ , for some  $p$ -norm balls. Specifically, the region of attraction (5.39) is restricted with the additional constraints

$$\exists y, x : \quad (5.61a)$$

$$Ey + \epsilon_y \left( \sum_{i=1}^q \|E_i\| \right)^T \leq f - s(N-1) \quad (5.61b)$$

$$Gx + \epsilon_x \left( \sum_{i=1}^q \|G_i\| \right)^T \leq h - \sigma(N).$$

The derivation of these constraints is similar to (5.53).

## 5.5 Examples

This section presents example scenarios that demonstrate the efficacy of designing a constraint-tightening policy in the manner detailed in the previous sections. A two-state system is considered first, which allows the region of attraction to be easily visualised and its exact volume to be readily computed. This is used to show the superiority of the max-vol policy by comparing actual ROA volumes over different policies. In addition, the effect of the policy on

closed-loop costs is also analysed. A final scenario presents a three-state system which further demonstrates the benefit of using max-vol as opposed to using a simple heuristic.

### 5.5.1 Symmetric Target Region Placement

Consider the unstable discrete-time linear system defined by state-space matrices

$$A = \begin{bmatrix} 1.2 & 1.2 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (5.62)$$

The outputs of the system comprise the states and the inputs fed through, that is,  $y = (x, u)$ . Given a maximum prediction horizon length of  $\bar{N} = 4$  steps, the control objective is to steer the system to the target set

$$\mathcal{T}_1 = \left\{ x \mid \begin{bmatrix} -2 & -2 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 2 & 2 \end{bmatrix}^T \right\}, \quad (5.63)$$

whilst satisfying the output constraints

$$\mathcal{Y} = \left\{ y \mid \begin{bmatrix} -15 & -5 & -3 \end{bmatrix}^T \leq y \leq \begin{bmatrix} 15 & 5 & 3 \end{bmatrix}^T \right\} \quad (5.64)$$

and being robust to the bounded disturbance

$$\mathcal{W} = \left\{ w \mid \begin{bmatrix} -0.3 & -1 \end{bmatrix}^T \leq w \leq \begin{bmatrix} 0.3 & 1 \end{bmatrix}^T \right\}. \quad (5.65)$$

This is a problem with symmetric constraints and a target set that is placed centrally within the output-admissible set. Solving (5.57) shows that the maximum ellipsoid volume results from choosing  $N = 4$ , corresponding to the maximum horizon length. Figure 5.1 illustrates the nominal (untightened) region of attraction as well as the 4-step ROA corresponding to  $\mathcal{P}_{\max\text{-vol}}$ . The terminal set and inner ellipsoidal approximation are also shown. Figure 5.2 shows the regions of attraction for all horizon lengths using the same policy. Note that the 1-step ROA is policy-independent, since CT over a prediction horizon of one tightens the terminal set directly by  $\mathcal{W}$ , with no input tightening applied. It is evident that the ROA corresponding to the maximum horizon length contains all of the others (apart from  $N = 1$  which is policy-independent), suggesting that the approximation is reasonable for this system.

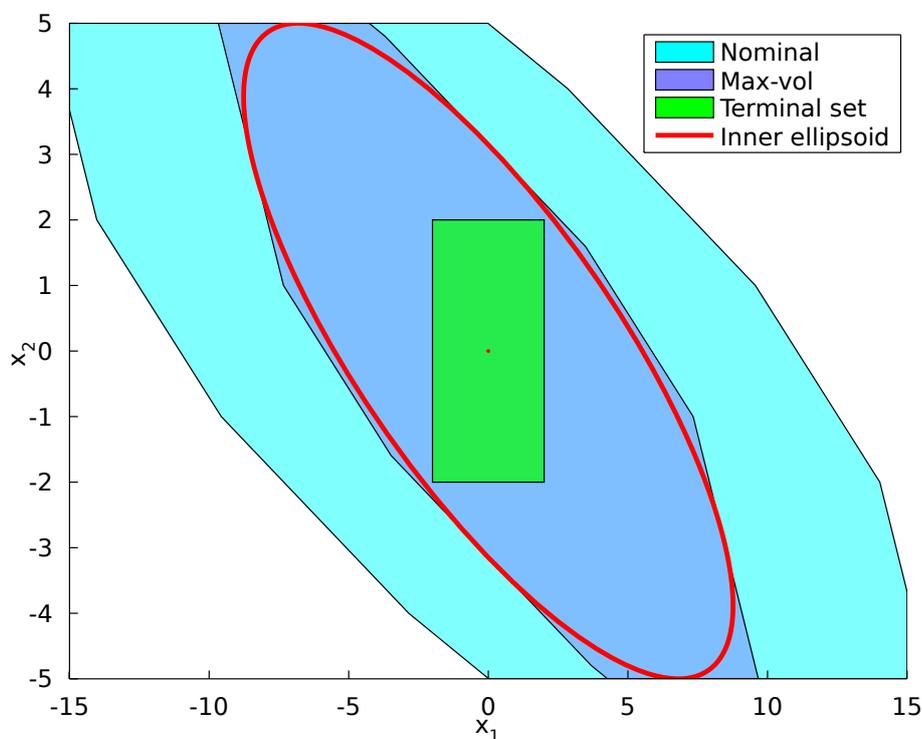
Table 5.1 compares the exact volumes for the 4-step ROA, as well as the total volume over all horizon lengths, using different tightening policies. These ROAs are illustrated in Figure 5.3. Exact volume calculations are achieved using Delaunay triangulation [Del34] for the convex sets and in the case of the non-convex union over horizon lengths, the inclusion-exclusion principle is applied [Com74]. The volumes for the entry marked “nominal” correspond to those for an arbitrary policy when  $\mathcal{W} = \emptyset$  i.e. the untightened non-robust problem. In addition, the

policy labelled “max-dist” is an implementation of the policy designed in [KRH07], adapted to the variable horizon problem. The derivation of this policy can be found in Appendix A. Plots of the full VH-ROAs for all policies can also be found there.

These results demonstrate the superiority of the max-vol policy for this application, as far as the size of the ROA is concerned. The policy provides a 36.6% larger 4-step ROA and a 35.6% larger VH-ROA as compared to max-dist policy. This is not surprising however, as the max-dist strategy poses a fundamentally different optimisation problem, as discussed in Section 5.1. The nilpotent policies are seen to drastically reduce the region of attraction, suggesting the importance of optimising the constraint tightening policy if a large ROA is desired. Subsection 5.5.2 vividly illustrates this point.

$\mathcal{P}$	$\text{vol}(\mathcal{R}(4, \mathcal{P}))$	$\text{vol}\left(\bigcup_{i=1}^4 \mathcal{R}(i, \mathcal{P})\right)$
nominal	199.2115	199.2115
max-vol	93.8338	94.0955
max-dist	59.4968	60.6070
3-step nilpotent	39.4018	40.8979
2-step nilpotent	23.2753	28.0190

**Table 5.1:** ROA volumes over different policies for  $\mathcal{T}_1$



**Figure 5.1:** Four-step ROA for  $\mathcal{T}_1$  using  $\mathcal{P}_{\text{max-vol}}$

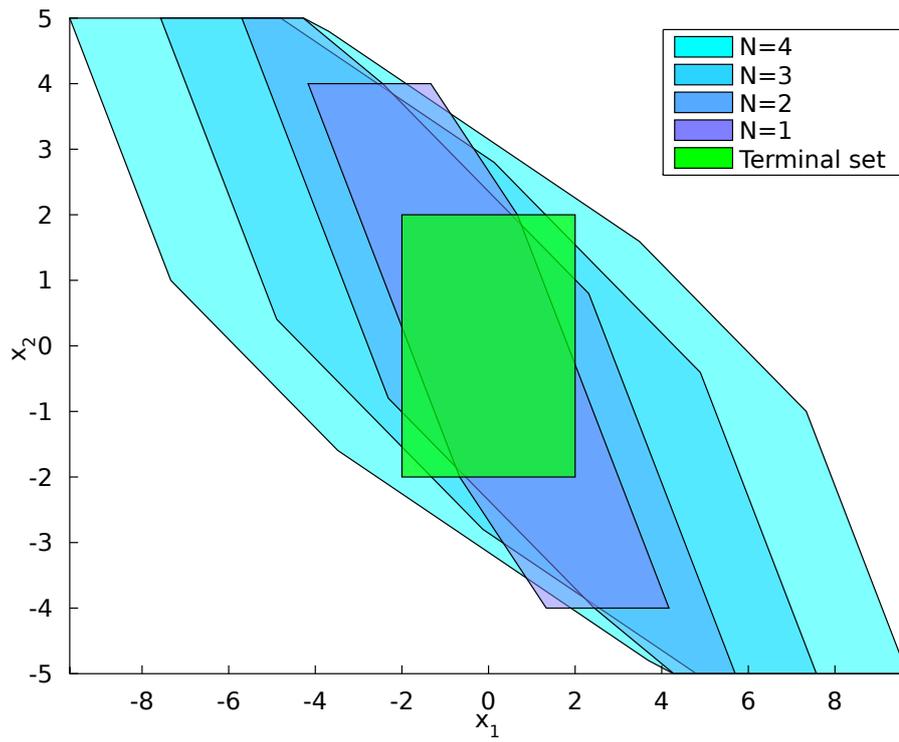


Figure 5.2: Total variable horizon ROA for  $\mathcal{T}_1$  using  $\mathcal{P}_{\max\text{-vol}}$

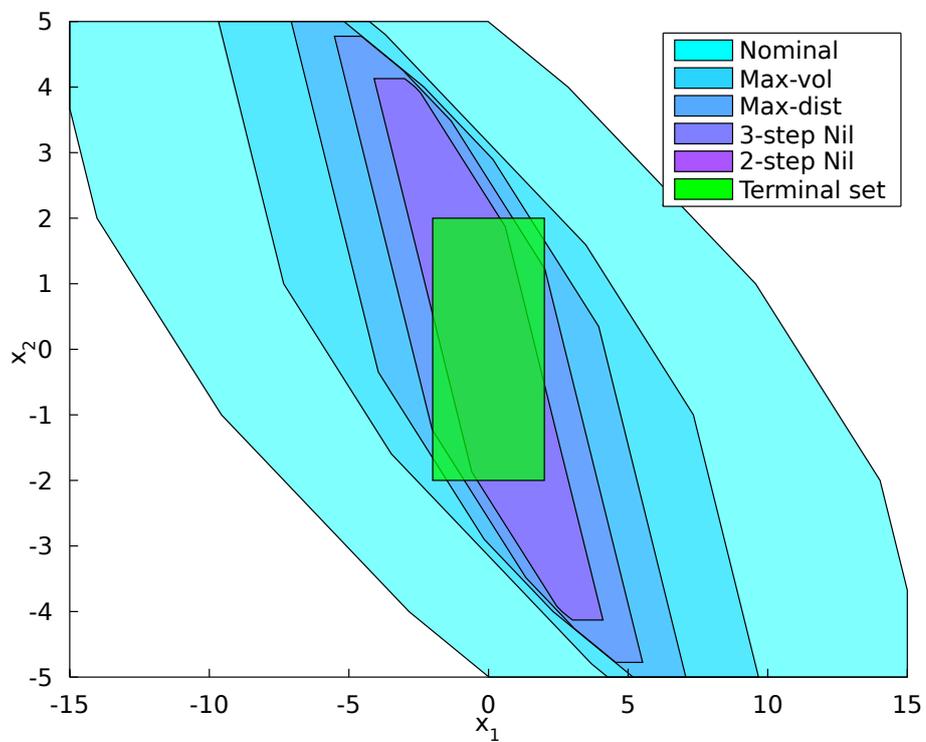
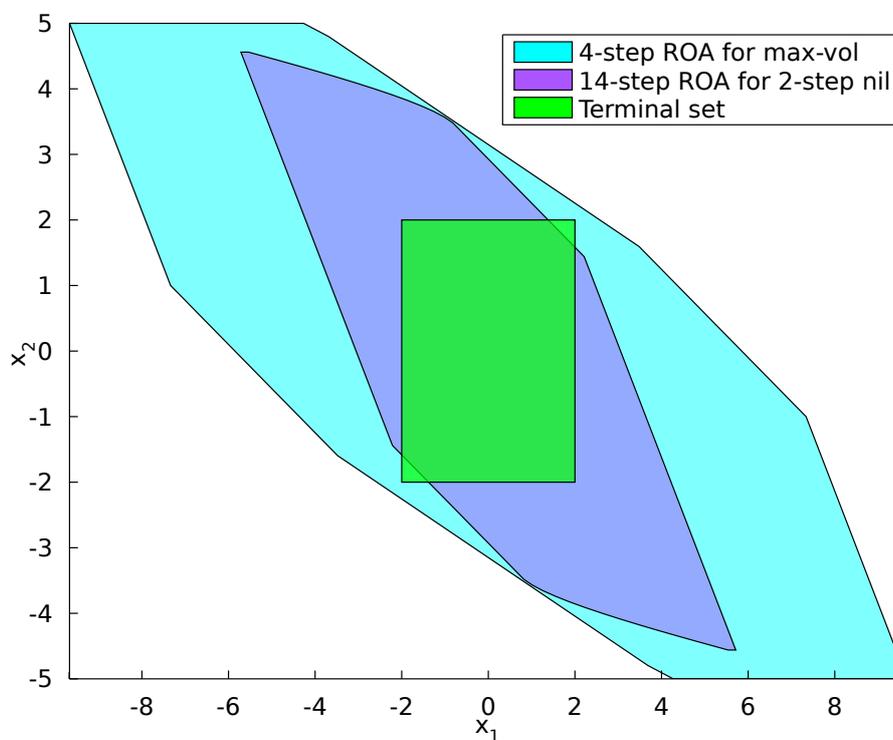


Figure 5.3: Four-step ROA comparison over different policies for  $\mathcal{T}_1$

### 5.5.2 Horizon Length Reduction

Choosing the max-vol policy can provide a drastic reduction in the required maximum horizon length, directly decreasing online computational complexity. Considering once again the control problem posed in Subsection 5.5.1, Figure 5.4 shows the 4-step ROA for  $\mathcal{P}_{\text{max-vol}}$  compared to the 14-step ROA for a two-step nilpotent tightening policy. For both policies, the full variable horizon ROA, apart from the 1-step ROA which is policy-independent, is contained within the 4-step ROA. It is evident that even with a horizon length over three times as long, the nilpotent policy is unable to give the region of attraction volume achievable by using  $\mathcal{P}_{\text{max-vol}}$ . This suggests that simply choosing a 2-step nilpotent policy for ease of computing the P-difference as in [RH06a] can significantly increase the required horizon length if the ROA is required to contain a given subset of  $\mathcal{O}$  that is far from the target set.



**Figure 5.4:** Comparison of 4-step ROA for  $\mathcal{T}_1$  using  $\mathcal{P}_{\text{max-vol}}$  with 14-step ROA for  $\mathcal{T}_1$  using  $\mathcal{P}_{2\text{-step-nil}}$

### 5.5.3 Asymmetric Target Region Placement

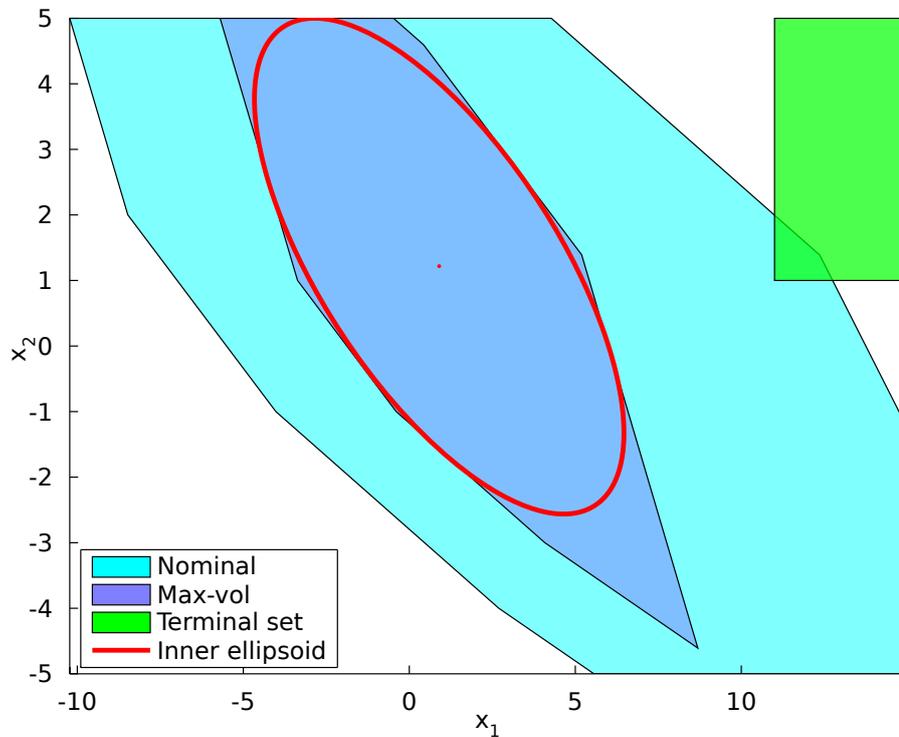
The effect of allowing a variable centre for the ellipsoid in (5.54) can be illustrated by shifting the terminal set. Using the same system dynamics, the new target region is given by

$$\mathcal{T}_2 = \left\{ x \mid \begin{bmatrix} 11 & 1 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 15 & 5 \end{bmatrix}^T \right\}, \quad (5.66)$$

which is the upper-right corner of the output-admissible set. It can be seen in Figure 5.5 that the ellipsoid centre has moved to accommodate the asymmetry. The effect of this new policy is especially evident when looking at the total variable horizon ROA for the max-vol policy, as illustrated in Figure 5.6. The 4-step ROA no longer contains all of the other ROAs for smaller horizon lengths. Table 5.2 compares the volumes as before, where it is seen that max-vol still provides the largest volume of all of the policies. However, the increase in VH-ROA volume over the max-dist policy is more modest (14%). As in the previous subsection, the 4-step ROAs for all policies are illustrated in Figure 5.7.

$\mathcal{P}$	$\text{vol}(\mathcal{R}(4, \mathcal{P}))$	$\text{vol}\left(\bigcup_{i=1}^4 \mathcal{R}(i, \mathcal{P})\right)$
nominal	167.0490	174.0646
max-vol	58.5689	99.0855
max-dist	34.4009	86.7941
3-step nilpotent	24.5657	80.5391
2-step nilpotent	17.3852	74.1435

**Table 5.2:** ROA volumes over different policies for  $\mathcal{T}_2$



**Figure 5.5:** Four-step ROA for  $\mathcal{T}_2$  using  $\mathcal{P}_{\text{max-vol}}$  compared with nominal ROA

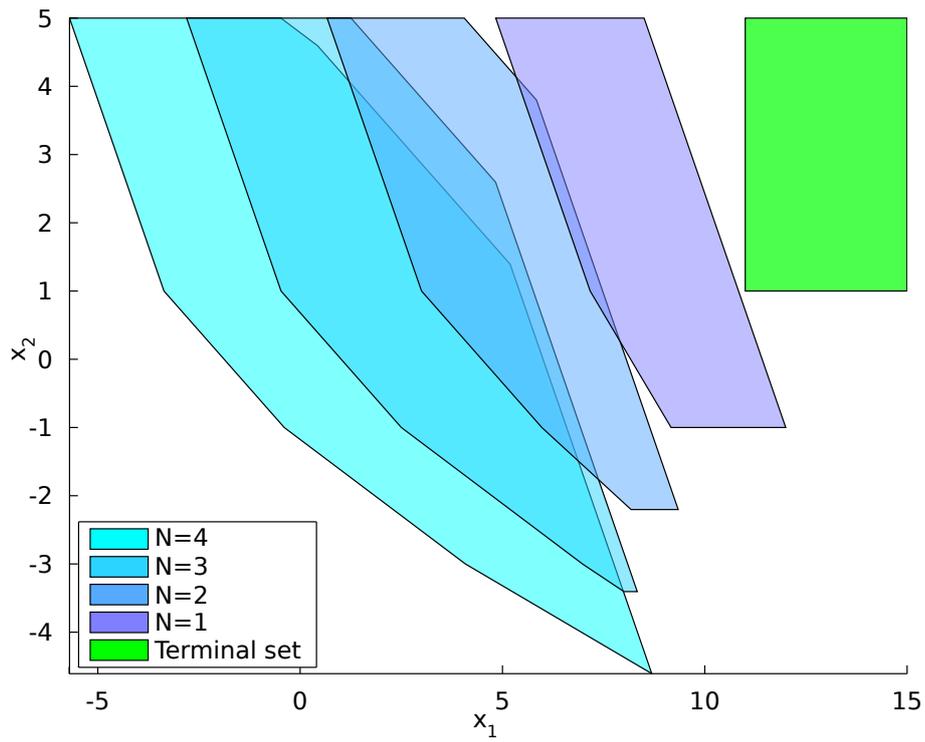


Figure 5.6: Total variable horizon ROA for  $\mathcal{T}_2$  using  $\mathcal{P}_{\max\text{-vol}}$

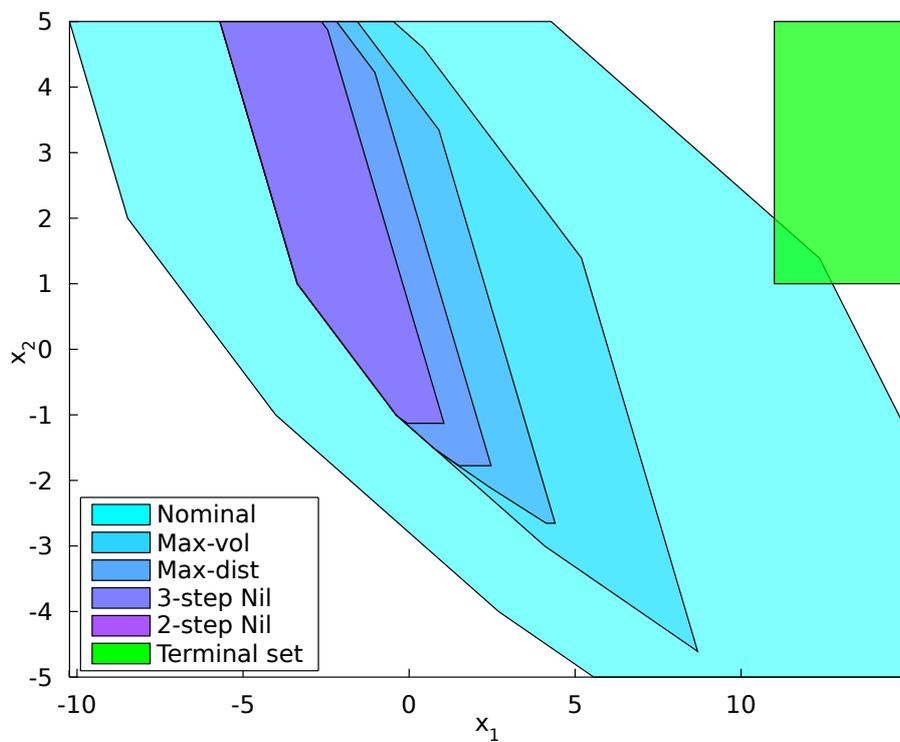


Figure 5.7: Four-step ROA comparison over different policies for  $\mathcal{T}_2$

### 5.5.4 Effect on Cost

It is possible to evaluate the cost performance of different tightening policies by choosing initial states randomly from a starting region that is contained within the 4-step regions of attraction of all of the policies considered. Considering once again the problem of steering the system state to  $\mathcal{T}_2$ , a suitable starting region is given by

$$\mathcal{X}_0 = \left\{ x \mid \begin{bmatrix} -3 & 2 \end{bmatrix}^T \leq x \leq \begin{bmatrix} -2 & 3 \end{bmatrix}^T \right\}. \quad (5.67)$$

The cost function

$$\sum_{j=0}^{N-1} 1 + \|\Theta(y(k+j|k) - y_r)\|_1, \quad (5.68)$$

where

$$\Theta = \text{diag} \left\{ \begin{bmatrix} 0.1 & 0.1 & 1 \end{bmatrix} \right\}, y_r = \begin{bmatrix} 0 & 13 & 3 \end{bmatrix}, \quad (5.69)$$

penalises a weighted sum of time to completion, input effort, and distance from the centre of  $\mathcal{T}_2$ . Table 5.3 shows the mean cost over 100 random starting states within  $\mathcal{X}_0$  and random admissible disturbance sequences, for different tightening policies. It is evident that optimising the ROA volume does not directly correspond to cost optimality, as the max-dist policy gives the best cost performance. It should be noted, however, that the differences in cost are minimal in comparison to the differences in volume of the ROA.

$\mathcal{P}$	mean cost
max-vol	10.4253
max-dist	10.1801
3-step nilpotent	10.3566
2-step nilpotent	10.3485

**Table 5.3:** Mean costs for Monte Carlo simulation using different policies

### 5.5.5 Heuristic Optimisation

Instead of solving directly for an inner approximation to the volume, a computationally simpler problem can be solved that avoids the projection operation entirely. Given that the vector  $s$  represents the amount of tightening applied to the nominal sets, minimising its two-norm may, in some way, be expected to reduce the amount by which the ROA is shrunk by the constraint tightening policy. Specifically, the optimisation problem is given by

$$\mathcal{P}_{\text{heur}}(N) = \begin{cases} \min_{\mathcal{P}} \|s\|_2 \\ \text{subject to:} \\ \exists \{Z(j) \geq 0_{a \times q}\}, \{\bar{Z}(j) \geq 0_{a \times r}\} : \\ s \text{ satisfies (5.17) and (5.18).} \end{cases} \quad (5.70)$$

Applying this policy on different two-state systems shows that it gives the same region of attraction as applying max-vol, suggesting that it may provide a simpler way of calculating a volume-optimal policy. However, the heuristic produces a smaller volume for higher state dimensions. Consider the three-state system

$$A = \begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0.167 \\ 0.5 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (5.71)$$

Given a maximum prediction horizon length of  $\bar{N} = 4$  steps, the control objective is to steer the system to the target set

$$\mathcal{T}_1 = \left\{ x \mid \begin{bmatrix} -1 & -1 & -2 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 1 & 1 & 2 \end{bmatrix}^T \right\}, \quad (5.72)$$

whilst satisfying the output constraints

$$\mathcal{Y} = \left\{ y \mid \begin{bmatrix} -20 & -5 & -3 & -3 \end{bmatrix}^T \leq y \leq \begin{bmatrix} 20 & 5 & 3 & 3 \end{bmatrix}^T \right\} \quad (5.73)$$

and being robust to the bounded disturbance contained within the set

$$\mathcal{W} = \left\{ w \mid \begin{bmatrix} -0.1 & -0.2 & -0.2 \end{bmatrix}^T \leq w \leq \begin{bmatrix} 0.1 & 0.2 & 0.2 \end{bmatrix}^T \right\}. \quad (5.74)$$

Repeating the same analysis as in Subsection 5.5.1, Table 5.4 shows the comparison of ROA volumes. In this case, the VH-ROA volumes are identical to the 4-step volumes, so only one set of data is listed. Whilst the heuristic outperforms the other policies considered in this scenario, it still results in a smaller ROA volume than max-vol. However, this means that it can provide a reasonable approximation to max-vol whilst being computationally tractable in high dimensions. Using a 1-norm or  $\infty$ -norm on  $s$  in (5.70) gives only marginally smaller volumes than the 2-norm, so using these norms can quicken computation even more, given that an LP rather than an SOCP needs to be solved. This technique is used in Chapter 6 for a 7-state system.

$\mathcal{P}$	$\text{vol} \left( \bigcup_{i=1}^4 \mathcal{R}(i, \mathcal{P}) \right)$
nominal	911.8229
max-vol	598.4078
heuristic	549.0186
max-dist	545.6656
3-step nilpotent	488.8198

**Table 5.4:** ROA volumes over different policies for 3-state system

## 5.6 Discussion

The examples in the preceding sections showed that max-vol performs well for two and three-state systems. However, it is prudent to consider what may happen in higher dimensions. There are both theoretical and computational issues which may affect performance and viability, and these issues are addressed in this section. Potential solutions to these issues are also discussed.

### 5.6.1 Curse of Dimensionality

A potential theoretical problem arises with the inner-ellipsoidal approximation in higher dimensions. As discussed in [Hay11], the ratio of the volume of a  $n$ -dimensional hypersphere to its circumscribed  $n$  dimensional hypercube is given by

$$\frac{\pi^{\frac{n}{2}}}{2^n \Gamma(\frac{n}{2} + 1)}, \quad (5.75)$$

where  $\Gamma(\cdot)$  represents the gamma function. As  $n$  increases, this ratio becomes extremely small. It can be reasonably expected that a similar phenomenon will be observed with polytopes of certain shapes, so the inner ellipsoidal approximation will occupy only a small fraction of the overall polytope volume, which tends to be concentrated near the vertices. One potential way of addressing this issue is to use inner superellipsoids instead, with the same norm parameter on each dimension. These are convex sets of the form,

$$\tilde{\mathcal{E}}(c, \Omega, p) = \{x \mid x = c + \Omega \xi, \forall \|\xi\|_p \leq 1\}, \quad (5.76)$$

for centre  $c$ , shape matrix  $\Omega \succ 0$  and norm parameter  $p$ . For computational tractability, the restriction  $p \in \{1, 2, \infty\}$  is imposed. Then, for  $p \in \{1, \infty\}$ , the resulting shapes are parallelotopes. In order to use this approximation, the definition of the matrix norm allows the constraints (5.53) to be modified to

$$\hat{\Phi}_i^T c + \|\Omega \hat{\Phi}_i\|_q \leq \hat{\psi}_i - \Gamma_i s, \quad (5.77)$$

where  $q = \infty$  if  $p = 1$  and  $q = 1$  if  $p = \infty$ . Then, for certain polytope geometries, values of  $p$  other than 2 may result in a better approximation to the true volume in higher dimensions. Experimentation with the examples covered in the previous section for  $p \in \{1, \infty\}$  gives the same ROA as using the ellipsoid.

### 5.6.2 Computational Complexity

For higher dimensional problems, both the projection operation described in Theorem 5.5 and the optimisation problem (5.54) can pose a very high computational burden. For the projection operation, an algorithm such as ESP whose complexity grows linearly with facet count becomes necessary. In terms of reducing the complexity of the optimisation problem,

the heuristic optimisation approach proposed in Subsection 5.5.5 results in a computationally simpler problem whilst still improving the region of attraction.

Another strategy to simplify the optimisation problem is to restrict the degrees of freedom in which the ellipsoid can vary, by a priori specifying the directions in which the semiaxes will point. This is possible due to the fact that  $\Omega$  is positive definite, so it can be diagonalised to give

$$\Omega = T\Lambda T^T, \quad (5.78)$$

where  $T$  is an orthogonal matrix and  $\Lambda$  is a diagonal matrix with the eigenvalues, which correspond to the semiaxis lengths, on the diagonal. These eigenvalues will all be real and positive. By specifying the matrix  $T$ , the search space is reduced. By also specifying a ratio between the semiaxis lengths, the problem will collapse into a linear program. Using a superellipsoid with  $p = 1$  or  $p = \infty$  with a fixed semiaxes ratio and directions is similar to the approach taken in [BFT04], where inner-box approximations to polytopes are specified. Note that specifying the orientation and aspect ratio of the ellipsoid a priori will require knowledge of the typical ROA shape over varying tightening policies for a given system, which can be difficult in higher dimensions.

### 5.6.3 Volume Calculation

An alternative to a convex problem that may be effective at higher dimensions is directly applying nonlinear optimisation. By creating a function that returns the exact volume of a polytope for a given tightening policy, nonlinear optimisation could be applied to find a locally-optimal policy to maximise a fixed horizon ROA. To test this idea, projection and Delaunay triangulation are used to define a volume function for the two dimensional problem defined in Subsection 5.5.1. Applying nonlinear optimisation with different initial guesses does not seem to find a better volume than that given by max-vol. However, not all starting guesses result in the optimisation ultimately converging to the volume corresponding to max-vol demonstrating the existence of local maxima. The issues with extending this approach to higher dimensions are the projection operation and the exact volume computation for the polytope, which can become computationally intensive, given that Delaunay triangulation requires vertex enumeration as a preliminary step.

An alternative to exact volume measurement is estimation. Monte Carlo methods could be used, which would allow direct estimation of the VH-ROA by counting the number of initial states for which a feasible solution exists. This method still suffers from the curse of dimensionality as well, as a very large number of random initial states will be needed for adequate coverage in higher dimensions. Other methods include box-approximation algorithms [BFT04] and entropy methods detailed in [BH10]. These noisy approximations could be used in conjunction with gradient descent algorithms such as simultaneous-perturbation stochastic approximation [Spa92] to locally optimise the policy for a fixed horizon ROA.

## 5.7 Conclusion

This chapter has developed a technique for choosing constraint-tightening policies that increase the size of the region of attraction. By expressing the tightened sets as affine functions of the constraint tightening policy and dual variables, a convex representation of the region of attraction, parameterised by the tightening policy, is formed. A convex optimisation problem is then posed to find the constraint tightening policy that maximises the volume of an inner-ellipsoidal approximation to the ROA over fixed horizon lengths. Simulations with example systems show this technique to enlarge even the full region of attraction over the variable horizon, as well as permit vastly smaller prediction horizon lengths to be used when compared to using nilpotent policies.

Future work will look at ways of formulating the optimisation problem for an inner approximation of the VH-ROA as a quasi-convex optimisation problem. It will further look at addressing some of the issues that occur in higher state dimensions. Another avenue of investigation will be the design of cost optimal policies. These extensions will potentially allow the optimisation of a policy that provides a trade-off between ROA volume and mean cost over a given set of starting states.

## 5.8 Acknowledgements

The work in this chapter has been submitted for review to the 50th IEEE Conference on Decision and Control. The simulations in this chapter were carried out using MATLAB R2011b, YALMIP [Löf04], the Multi-parametric Toolbox [KGB04], CPLEX [CPL09], SDPT3 [TTT99] and the Invariant Set Toolbox [Ker00].

## Chapter 6

# Case Study - Surface Excavation

### 6.1 Introduction

Surface mining operations involve the recovery of valuable mineral ores from the earth through removal of the covering material, known as overburden. In most cases, this requires the use of high explosives in order to blast the overlying earth and ore to facilitate their removal. The recovered ore and waste are separated and crushed for further processing.

The majority of surface mines utilise large-scale machinery for removing the blasted ore and waste. These machines are presently controlled by skilled operators who require many years of training in order to achieve proficiency. In particular, difficult digging conditions require operators to have an intuitive understanding of material and bucket interactions. Control strategies to be used in such conditions are only learnt through years of experience. This experience results in the ability to achieve consistent bucket payloads, which is highly desirable when considering the entire supply chain. These factors make the digging cycle of the machine, where it engages material, an ideal candidate for automation.

MPC is a means through which such automation can be potentially achieved. Its ability to handle constraints and optimise an objective function for performance suggest that it may be useful for automating a surface mining machine. In this application, there is no notion of stability once the bucket engages the material: the overall objective is to fill the bucket to a target payload as quickly as possible and then exit the material. VH-MPC is therefore lends itself to this control problem.

In order to demonstrate the applicability of VH-MPC to digging, the rope shovel is chosen as a candidate machine for analysis. It is widely used in surface mining operations. As shown in Figure 6.1, rope shovels have two degrees of freedom: the translational movement of the crowd arm and the length of the hoist rope. The crowd arm pivots about the crowd pin. A combination of hoist and crowd movements generate digging trajectories through material, as shown in Figure 6.2

This chapter develops a simplified dynamic model of a representative rope-shovel with a 60-100 tonne nominal bucket capacity. A material model based on Mohr-Coulomb theory is

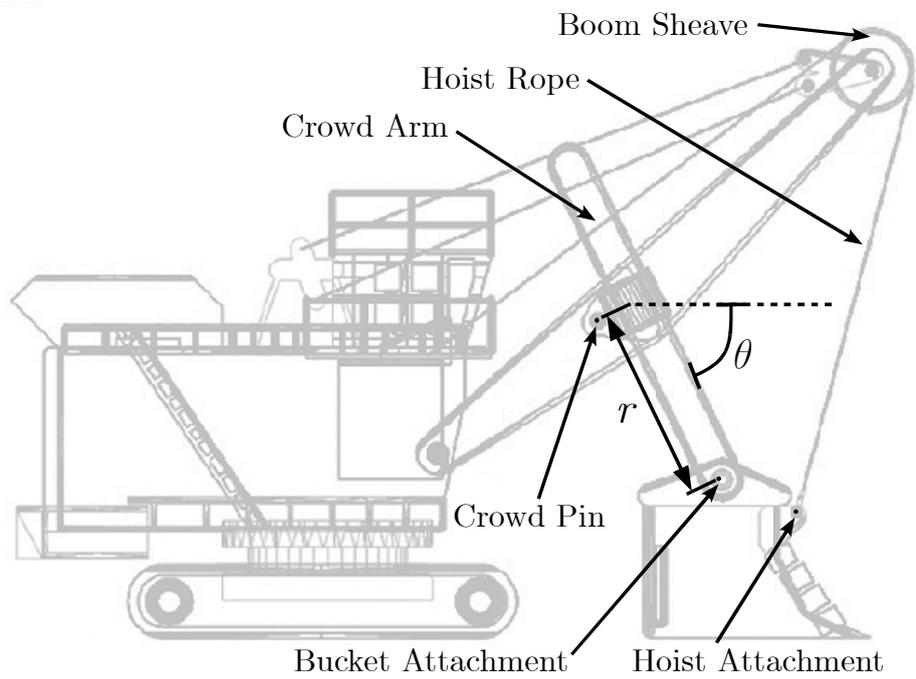


Figure 6.1: The rope shovel

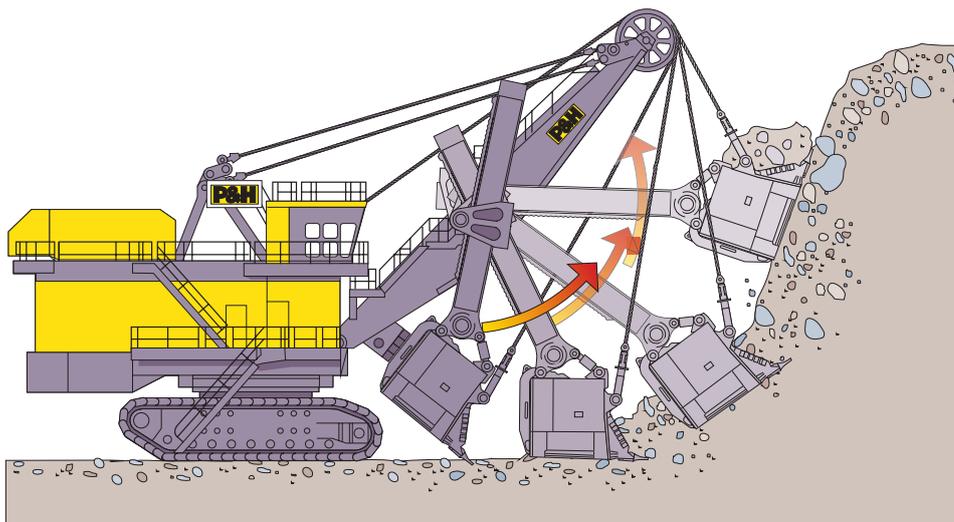


Figure 6.2: Rope shovel excavation trajectory [P&03]

also developed, simulating excavation in oil sand. The combined model of the shovel and material goes further than [FHA05] by including the inertial effects of a varying bucket mass and allowing the material force to influence the bucket's trajectory. Based on the overall nonlinear model of the machine and material, an inner pre-stabilising controller is designed, following which, a linear discrete-time model is identified. This model is used for designing a VH-MPC controller for trajectory generation. Subsequently, simulations are performed, which incorporate the theory on feasible contingencies, move blocking and optimal constraint tightening developed in the previous chapters. These simulations show the generation of realistic trajectories, motivating new applications of VH-MPC for autonomy that go beyond the realm of vehicle manoeuvring.

### 6.1.1 Nomenclature

Vectors denoting actual physical quantities such as positions, velocities, forces and moments will be typeset in boldface. When the symbol appears without boldface, it denotes the magnitude of the vector in question. Boldface will not be used for vectors in generalised coordinates (where the components may have different physical units). Standard robotics notation will be used to denote vectors in a given coordinate frame, so the leading subscript in  ${}^n\mathbf{z}_m$  indicates the vector  $\mathbf{z}_m$  in coordinate frame  $n$ . Position vectors will also be expressed in polar and Cartesian form using the notation

$${}^n\mathbf{z}_m = r_m \begin{bmatrix} \cos({}^n\theta_m) \\ \sin({}^n\theta_m) \\ 0 \end{bmatrix} = \begin{bmatrix} {}^nx_m \\ {}^ny_m \\ 0 \end{bmatrix}. \quad (6.1)$$

Note that a different font is used to indicate cartesian components to distinguish them from state and output variables. Although only two-dimensional motion is considered in this model, the third vector component is required for the calculation of moments using cross-products. The standard Cartesian unit vectors  $i$ ,  $j$  and  $k$  will be used as needed.

## 6.2 Mechanism Model

A simplified rigid-body model of the shovel mechanism combined with bucket-material interaction is developed for simulation purposes, based on reasonably realistic parameter values from [FL07]. The kinematic and dynamic parameters are representative of a typical medium capacity rope shovel. Two coordinates are selected as shown in Figure 6.1: crowd arm translation ( $r$ ) and crowd arm rotation ( $\theta$ ). These coordinates are kinematically related to the hoist rope length. The various lengths and angles are shown in Figure 6.3, and the corresponding parameter values are listed in Table 6.1. Also pictured are two systems of coordinates used for specifying vectors: the base frame ( ${}^0x, {}^0y$ ) fixed to the machine body and ( ${}^1x, {}^1y$ ) rotating with the crowd arm. The base frame is positioned 6.6 m above ground level.

Figure 6.4 illustrates the simplified mechanism model as well as the coordinates  $r$  and  $\theta$ . The

crowd arm and bucket are treated as a rigid body (combined mass  $M$ , inertia about combined centre of mass  $J$ ), with a sliding contact at the crowd pin approximating the rack-and-pinion on an actual shovel. In addition, the weight of the material in the bucket,  $m_p$ , is treated as a point mass located at the bucket's centre of mass. Estimated dynamic parameters, masses, centres of mass, inertias and coefficients of friction are listed in Table 6.2.

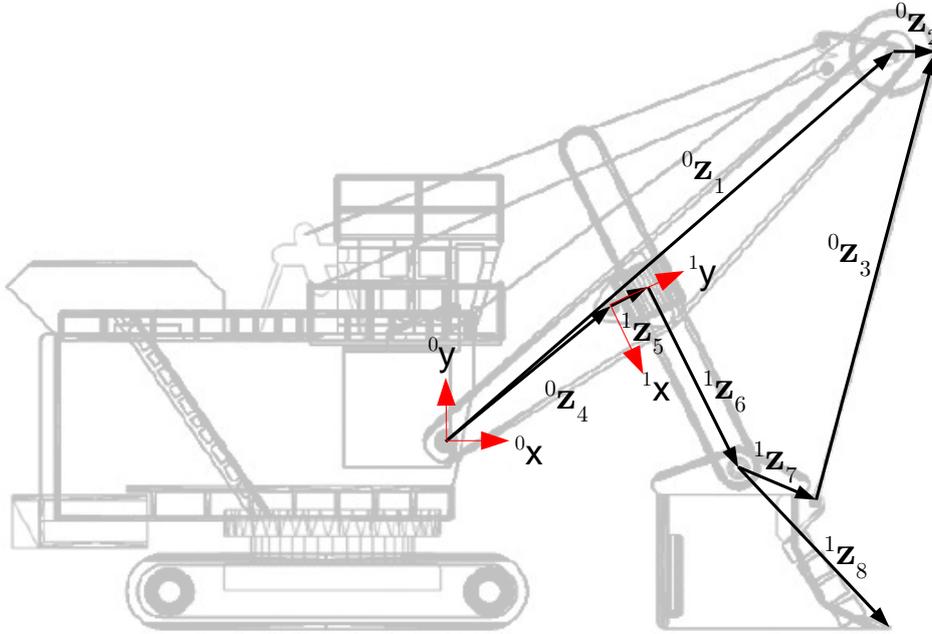


Figure 6.3: Kinematic vectors

### 6.2.1 Kinematic Equations

For convenience, vectors in the base coordinate frame will have their leading superscripts omitted. In terms of the crowd length  $r$ , pictured in 6.4, the vector  ${}^1z_6$  is defined as

$${}^1z_6 = \begin{bmatrix} r & 0 & 0 \end{bmatrix}^T. \quad (6.2)$$

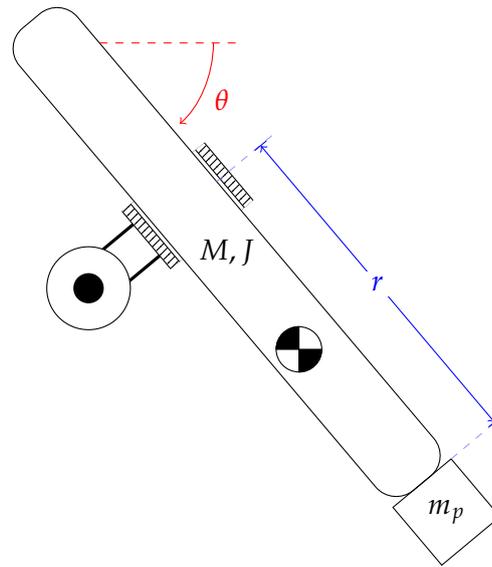
Using the rotation matrix

$$R_\theta = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6.3)$$

which is a function of the crowd angle, this vector can be transformed into the base frame to give

$$z_6 = R_\theta {}^1z_6 = \begin{bmatrix} r \cos \theta & r \sin \theta & 0 \end{bmatrix}^T, \quad (6.4)$$

The two degrees of freedom will be used as generalised coordinates for deriving equations of motion in Subsection 6.2.2. Before doing this, however, kinematic equations are required



**Figure 6.4:** Mechanism model showing generalised coordinates

Parameter	Value	Description
$L_c$	13.19 m	Crowd arm length
$r_1$	20.55 m	Machine origin to boom sheave centre
$r_2$	1.53 m	Boom sheave radius
$r_4$	7.39 m	Machine origin to crowd pin
$r_5$	1.47 m	Crowd pivot radius about pin
$r_7$	2.87 m	Bucket attachment to hoist attachment
$r_8$	7.64 m	Bucket attachment to bucket teeth
$r_{ccm}$	6.7570 m	Crowd pin to crowd's centre of mass when $r = L_c$
$r_{kcm}$	3.2000 m	Bucket pin to bucket's centre of mass
${}^0\theta_1$	$41.74^\circ$	Angle of vector $z_1$ in base frame
${}^0\theta_2$	$-22.50^\circ$	Angle of vector $z_2$ in base frame
${}^0\theta_4$	$39.92^\circ$	Angle of vector $z_4$ in base frame
${}^1\theta_5$	$90.00^\circ$	Angle of vector $z_5$ in crowd pin frame
${}^1\theta_6$	$0.00^\circ$	Angle of crowd arm in crowd pin frame
${}^1\theta_7$	$39.42^\circ$	Angle of vector $z_7$ in crowd pin frame
${}^1\theta_8$	$16.20^\circ$	Angle of vector $z_8$ in crowd pin frame
${}^1\theta_{ccm}$	$12.57^\circ$	Angle of vector from crowd pin to crowd centre of mass in crowd pin frame
${}^1\theta_{kcm}$	$-26.57^\circ$	Angle of vector from bucket attachment to bucket centre or mass in crowd pin frame

**Table 6.1:** Kinematic parameters

Parameter	Value	Description
$m_c$	65800 kg	Mass of crowd arm
$m_k$	24500 kg	Mass of bucket
$J_c$	774900 kg m <sup>2</sup>	Crowd arm inertia about its mass centre
$J_k$	64580 kg m <sup>2</sup>	Bucket inertia about its mass centre
$c_r$	$1 \times 10^5$ N s m <sup>-1</sup>	Viscous friction coefficient for crowd translation
$c_\theta$	$1 \times 10^6$ N m s	Viscous friction coefficient for crowd rotation
$g$	9.81 m s <sup>-2</sup>	Acceleration due to gravity

**Table 6.2:** Dynamic parameters

to relate the hoist length and velocity to the generalised coordinates. This is due to the fact that the crowd angle is in reality a function of both the crowd length and hoist rope length. In addition, equations relating the generalised coordinates to the bucket tip position and velocity are needed for trajectory information.

Closing the vector loop in Figure 6.3 gives the hoist vector relationship

$$z_3 = z_1 + z_2 - z_4 - R_\theta \left( {}^1z_5 + {}^1z_6 + {}^1z_7 \right). \quad (6.5)$$

The right hand side of (6.5) is a function of  $r$  and  $\theta$ . The hoist length is then given by  $z_3$ . From the expression for  $z_3$ , it is possible to relate the hoist rope velocity to  $\dot{r}$  and  $\dot{\theta}$ . By defining the vector of generalised coordinates

$$q = \begin{bmatrix} r & \theta \end{bmatrix}^T, \quad (6.6)$$

the time derivative of the hoist rope vector is calculated as

$$\begin{aligned} \dot{z}_3 &= \nabla_q z_3 \dot{q} \\ &= \begin{bmatrix} -\cos \theta & r \sin \theta + r_5 \sin (\theta + {}^1\theta_5) + r_7 \sin (\theta + {}^1\theta_7) \\ -\sin \theta & -r \cos \theta - r_5 \cos (\theta + {}^1\theta_5) - r_7 \cos (\theta + {}^1\theta_7) \\ 0 & 0 \end{bmatrix} \dot{q}. \end{aligned} \quad (6.7)$$

In this expression,  $\nabla_q z_3$  is the vector gradient, or Jacobian, of the vector  $z_3$  with respect to the generalised coordinates. Given that  $z_3 = \sqrt{z_3 \cdot z_3}$ , the chain rule can be applied to calculate the rate of hoist extension or retraction

$$\begin{aligned} \frac{dz_3}{dt} &= \frac{z_3}{z_3} \cdot \dot{z}_3 \\ &= \hat{z}_3 \cdot \dot{z}_3. \end{aligned} \quad (6.8)$$

This is simply the component of  $\dot{z}_3$  in the direction of the hoist unit vector  $\hat{z}_3$ , which is expected intuitively.

The position vector of the combined centre of mass  ${}^1z_{cm}$  of the crowd arm and bucket, in the

crowd pin frame is given by

$${}^1\mathbf{z}_{\text{cm}} = \frac{1}{M} \left( m_c {}^1\mathbf{z}_{\text{ccm}} + \begin{bmatrix} r - L_c & 0 & 0 \end{bmatrix}^T + m_k {}^1\mathbf{z}_{\text{kcm}} + {}^1\mathbf{z}_5 + {}^1\mathbf{z}_6 \right), \quad (6.9)$$

Note that this centre of mass moves depending on the crowd extension or retraction length. The  $r - L_c$  term accounts for the fact that the position of the crowd centre of mass is defined with the crowd fully extended. In terms of the parameters quantified in Table 6.1, the position of the centre of mass in the base frame is given by the expression

$$\bar{\mathbf{z}} = \mathbf{z}_4 + R_\theta {}^1\mathbf{z}_{\text{cm}}. \quad (6.10)$$

Its velocity is then given by

$$\dot{\bar{\mathbf{z}}} = \nabla_q \bar{\mathbf{z}} \dot{q}, \quad (6.11)$$

where  $\nabla_q \bar{\mathbf{z}}$  is the Jacobian of the centre of mass with respect to the generalised coordinate vector. Additionally, the velocity of the payload, assumed to be located at the bucket's centre of mass, is required. The position of the payload centre of mass in the base frame is then

$$\bar{\mathbf{z}}_p = \mathbf{z}_4 + R_\theta \left( {}^1\mathbf{z}_5 + {}^1\mathbf{z}_6 + {}^1\mathbf{z}_{\text{kcm}} \right). \quad (6.12)$$

As with the equivalent centre of mass, the velocity of the payload is given by

$$\dot{\bar{\mathbf{z}}}_p = \nabla_q \bar{\mathbf{z}}_p \dot{q}. \quad (6.13)$$

### 6.2.2 Equations of Motion

As a first step towards deriving the equations of motion for the mechanism, it is necessary to find the combined inertia for the rigidly connected bucket and crowd-arm for in-plane rotation. As this is independent of the coordinate frame, assume that the crowd arm is fully extended. With this assumption, denote the resulting centre of mass position in the crowd pin frame as  ${}^1\mathbf{z}'_{\text{cm}}$ . This vector is constant in this frame. Then, the parallel axis theorem gives the equivalent inertia of the combined crowd pin and bucket system about this centre of mass as

$$J = J_c + m_c \left( \left( {}^1x'_{\text{cm}} - {}^1x_{\text{ccm}} \right)^2 + \left( {}^1y'_{\text{cm}} - {}^1y_{\text{ccm}} \right)^2 \right) + J_k + m_k \left( \left( {}^1x'_{\text{cm}} - L_c - {}^1x_{\text{kcm}} \right)^2 + \left( {}^1y'_{\text{cm}} - r_5 - {}^1y_{\text{kcm}} \right)^2 \right). \quad (6.14)$$

Using the kinematic and dynamic parameters, the total kinetic energy of the system, denoted  $T$  is calculated as

$$T = \frac{1}{2} M \dot{\bar{\mathbf{z}}} \cdot \dot{\bar{\mathbf{z}}} + \frac{1}{2} m_p \dot{\bar{\mathbf{z}}}_p \cdot \dot{\bar{\mathbf{z}}}_p + \frac{1}{2} J \dot{\theta}^2. \quad (6.15)$$

The potential energy, denoted  $V$ , depends only on the y components of the mechanism and payload centres of mass in the base frame, so

$$V = M g j \cdot \bar{\mathbf{z}} + m_p g j \cdot \bar{\mathbf{z}}_p. \quad (6.16)$$

From the definition of the Lagrangian

$$L = T - V, \quad (6.17)$$

the Euler-Lagrange differential equation is applied to each generalised coordinate to give

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{r}} \right) - \frac{\partial L}{\partial r} = F_r - F_{r,f} \quad (6.18)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \tau_\theta - \tau_{\theta,f}. \quad (6.19)$$

The generalised force and moment  $F_r$  and  $\tau_\theta$ , as well as the dissipative force terms  $F_{r,f}$  and  $\tau_{\theta,f}$  are quantified in terms of applied forces in Subsection 6.2.3, allowing the equations of motion to be written in the form [SHV05]

$$D(q, m_p)\ddot{q} + C(q, \dot{q}, m_p, \dot{m}_p) + G(q, m_p) + \Lambda\dot{q} = \Phi, \quad (6.20)$$

where  $D(\cdot, \cdot)$  is the inertia matrix,  $C(\cdot, \cdot, \cdot, \cdot)$  is a vector of Coriolis terms and  $G(\cdot, \cdot)$  is a vector of gravity terms. The dissipative forces are contained in the vector

$$\Lambda\dot{q} = \begin{bmatrix} F_{r,f} & \tau_{\theta,f} \end{bmatrix}^T \quad (6.21)$$

and the applied external forces are denoted by the vector

$$\Phi = \begin{bmatrix} F_r & \tau_\theta \end{bmatrix}^T. \quad (6.22)$$

These applied forces are transformations of the hoist rope and crowd arm actuator forces, as well as the material resistance force  $F_{\text{tip}}$ , which is assumed to act at the bucket tip. As the dynamic equations (6.20) depend on the varying payload  $m_p$ , expressions for the matrices in terms of the kinematic and dynamic parameters are rather long. Hence, they are presented in Appendix B, having been derived using MATLAB's symbolic toolbox. The expressions for the external forces and frictional forces are derived in the following subsection.

### 6.2.3 External & Dissipative Forces

The equations of motion derived above require external forces to be expressed as generalised forces acting on the coordinates  $r$  and  $\theta$ . As frictional forces have not yet been quantified, they will also be introduced here. Define the base-frame unit vector in the crowd extension direction

$$\hat{r} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \end{bmatrix}^T \quad (6.23)$$

Then, the crowd extension force  $F_c$ , hoist pull force  $F_h$ , and tip force  $F_{\text{tip}}$  give the generalised crowd force

$$F_r = F_c + \hat{r} \cdot (F_h \hat{z}_3 + F_{\text{tip}}). \quad (6.24)$$

Varignon's theorem [BF08] is used to calculate the generalised moment about the crowd pin, which is given by

$$\tau_\theta = [(z_5 + z_6 + z_8) \times \mathbf{F}_{\text{tip}} + (z_5 + z_6 + z_7) \times F_h \hat{\mathbf{z}}_3 + \boldsymbol{\tau}_{\text{tip}}] \cdot \mathbf{k}, \quad (6.25)$$

where  $\boldsymbol{\tau}_{\text{tip}}$  is any moment applied to the bucket teeth. In the material model presented in the next section, this is assumed to be zero, as the line of action of the force is assumed to pass through the bucket teeth. The viscous friction term acting against crowd motion is given by

$$F_{r,f} = c_r \dot{r}. \quad (6.26)$$

To quantify the dissipative moment it is necessary to sum both the viscous moment on the crowd pin itself and the moment induced by the viscous friction acting on the crowd arm. The resulting moment is given by

$$\tau_{\theta,f} = c_\theta \dot{\theta} + c_r r_5 \dot{r}. \quad (6.27)$$

Hence, the dissipative force matrix is given by

$$\Lambda = \begin{bmatrix} c_r & 0 \\ r_5 c_r & c_\theta \end{bmatrix} \quad (6.28)$$

### 6.3 Material Interaction Model

In general, analytically modelling bucket interaction behaviour is not possible. However, homogeneous materials like oil sand can be approximated through Mohr-Coulomb theory, based on static material failure. The Fundamental Equation of Earthmoving Mechanics (FEE) [Ree64; LSC98; CS99] uses this theory to estimate the force required for a planar tool to breach the shear failure threshold of the material, as a function of tool and terrain orientation. Figure 6.5 illustrates the required parameters. Numerical values for the material parameters can be found in Table 6.3 [Cla57; Col07]. The remainder are as follows:  $L_t$  is the length of the tool extending under the terrain surface,  $d$  is the depth of the tool tip below the surface,  $L_f$  is the length of the failure plane below the surface,  $Q$  is the surcharge, or displaced soil that rests on the wedge,  $\phi$  is the material angle of repose,  $R$  is the resistance force of the material exerted on the wedge. The material in the shaded wedge amounts to the payload i.e. the current bucket fill. Its volume is denoted  $V_s$ .

In terms of the material and geometrical parameters, the force required to induce shear failure is given by

$$\mathbf{R}_s = (d^2 w \gamma g N_w + c w d N_c + V_s \gamma g N_q) \hat{\mathbf{h}}, \quad (6.29)$$

where  $w$  is the width of the tool and  $g$  is the acceleration due to gravity.  $N_w$ ,  $N_c$  and  $N_q$  are dimensionless parameters defined by the equations

$$N_w = \frac{(\cot \beta - \tan \alpha)(\cos \alpha + \sin \alpha \cot(\beta + \phi))}{2[\cos(\rho + \delta) + \sin(\rho + \delta) \cot(\beta + \phi)]} \quad (6.30)$$

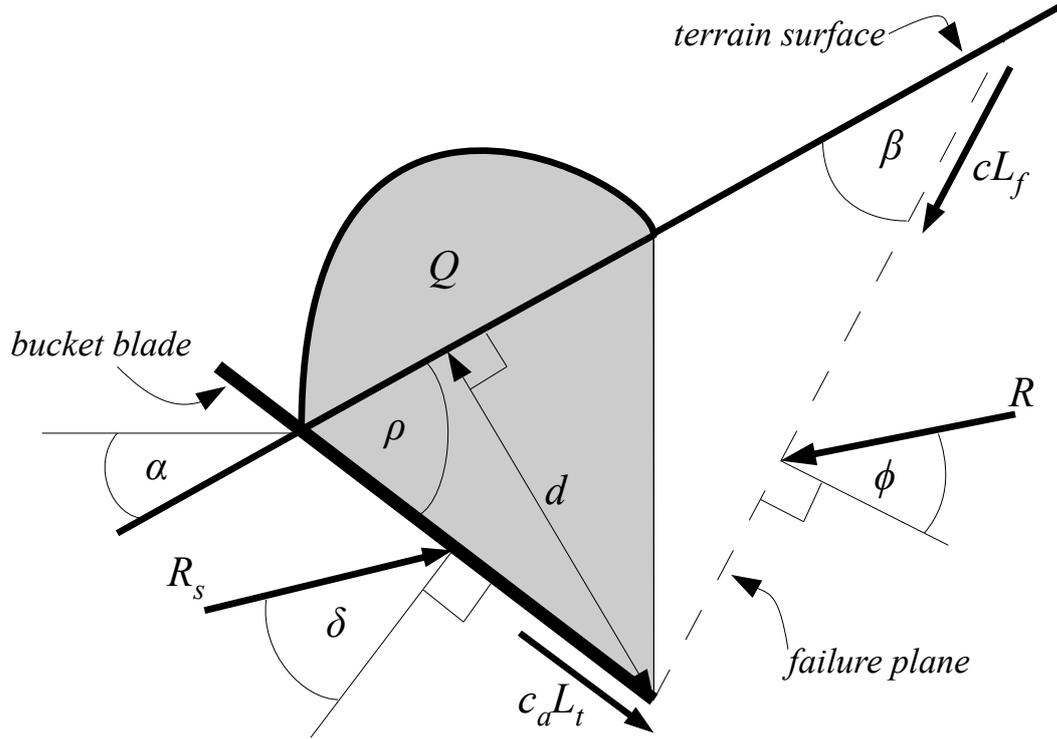


Figure 6.5: Static balance of material failure forces [CS99]

$$N_c = \frac{1 + \cot \beta \cot(\beta + \phi)}{\cos(\rho + \delta) + \sin(\rho + \delta) \cot(\beta + \phi)} \quad (6.31)$$

$$N_q = \frac{\cos \alpha + \sin \alpha \cot(\beta + \phi)}{\cos(\rho + \delta) + \sin(\rho + \delta) \cot(\beta + \phi)}. \quad (6.32)$$

The direction of  $R_s$  is given by the unit vector

$$\hat{h} = \begin{bmatrix} \sin(\rho - \alpha + \delta) & \cos(\rho - \alpha + \delta) & 0 \end{bmatrix}^T, \quad (6.33)$$

and its line of action is, for the purposes of this model, assumed to pass through the bucket teeth. Certain geometric configurations of the bucket can result in  $R_s$  becoming negative, implying a material force pulling the bucket deeper horizontally, which is not observed in reality. Hence, under such conditions,  $R_s$  is taken to be zero. Also note that for simplicity, a linear terrain face profile defined by the equation

$$\begin{bmatrix} 1.66 & -1 \end{bmatrix} z_{\text{tip}} = 27.018 \text{ m} \quad (6.34)$$

is used for simulation, where  $z_{\text{tip}} \in \mathbb{R}^2$  defines the  $(x, y)$  bucket tip position in frame 0. The fill volume  $V_s$  is estimated from the bucket width and area subtended by the trajectory and face, assuming the bucket were to pull out of the face vertically from its current pose as in Figure 6.5.

Given that the FEE only describes static failure conditions, it can be considered as a threshold on the force that the material is able to resist until failure. Assuming that the sum of the forces

Parameter	Value	Description
$\alpha$	59°	Terrain surface (face) angle
$\beta$	35°	Failure surface angle (assumed)
$\delta$	37°	Material-tool friction angle
$\phi$	50°	Angle of repose
$\gamma$	2000 kg m <sup>-3</sup>	Bulk density
$c$	0 Pa	Cohesion
$c_a$	0 Pa	Material-tool adhesion

**Table 6.3:** Material properties

being exerted by the actuators and gravity at the bucket teeth is given by the vector  $F_t$ , the resulting force being exerted by the soil at the bucket teeth is given by

$$F_{\text{tip}} = -\min(F_t \cdot \hat{h}, R_s) \hat{h}, \quad (6.35)$$

where the force is exerted opposite to the failure direction. This transforms through (6.24) and (6.25) to give a complete model of the machine digging through the material.

### 6.3.1 Implementation

In order to implement the dynamic model described by (6.20), it is necessary to define five state variables, namely  $r$ ,  $\theta$ ,  $\dot{r}$ ,  $\dot{\theta}$  and  $m_p$ . Then, define the state vector  $\chi$  as

$$\chi(t) \equiv \begin{bmatrix} r(t) & \theta(t) & \dot{r}(t) & \dot{\theta}(t) & m_p(t) \end{bmatrix}^T. \quad (6.36)$$

Equation (6.20) can therefore be rewritten as a nonlinear state space equation of the form

$$\dot{\chi}(t) = f(\chi(t), \Phi(t)), \quad (6.37)$$

where  $f(\cdot, \cdot)$  is a nonlinear vector function. This is now in a suitable form for simulation with standard ODE solvers.

## 6.4 System Identification

As a first step towards identifying a linear model for the shovel, a pre-stabilising linear controller with integral action is designed for trajectory tracking, to help mitigate some of the effects of the nonlinear dynamics. This linear controller is designed by sampling data from the open-loop nonlinear system to identify a suitable linearisation, then applying LQR-like methods. A linear model is subsequently identified for the closed loop nonlinear trajectory-tracking system, which will be used for MPC.

### 6.4.1 Pre-Stabilising Controller

Given that there are no open-loop equilibrium states that are of interest for digging control, a forced equilibrium is calculated, for an initial machine pose at rest given by  $r_0 = 6$  m and  $\theta_0 = -60^\circ$ , with the bucket tip positioned just within the material face. Perturbations are made to the force and moment keeping the crowd arm in equilibrium, causing the machine to move through the material. During the resulting motion, data for four states, namely  $r$ ,  $\theta$ ,  $\dot{r}$  and  $\dot{\theta}$ , are collected with a sampling frequency 100 Hz. This allows a discrete-time linearisation to be found using least-squares techniques, with the state and input vectors given by

$$x(k) = \begin{bmatrix} \tilde{r}(k) & \tilde{\theta}(k) & \dot{r}(k) & \dot{\theta}(k) \end{bmatrix}^T \quad (6.38)$$

$$u(k) = \begin{bmatrix} \tilde{F}_c(k), \tilde{F}_h(k) \end{bmatrix}^T, \quad (6.39)$$

where  $t = kT_s$ . The tilde on the states and inputs indicates a trim value from equilibrium. As the machine is at rest in the trim condition, the crowd velocity and angular velocity correspond to the actual states.

**Remark 6.1.** The trim value for the crowd force has an important role to play in correcting an inadequacy of the material model. Given that the model does not account for the supporting force which would act on the base of the bucket, the crowd is assumed to be acted upon by the equilibrium force at all times. This simplification results in more realistic crowd forces than ignoring the supporting force entirely.

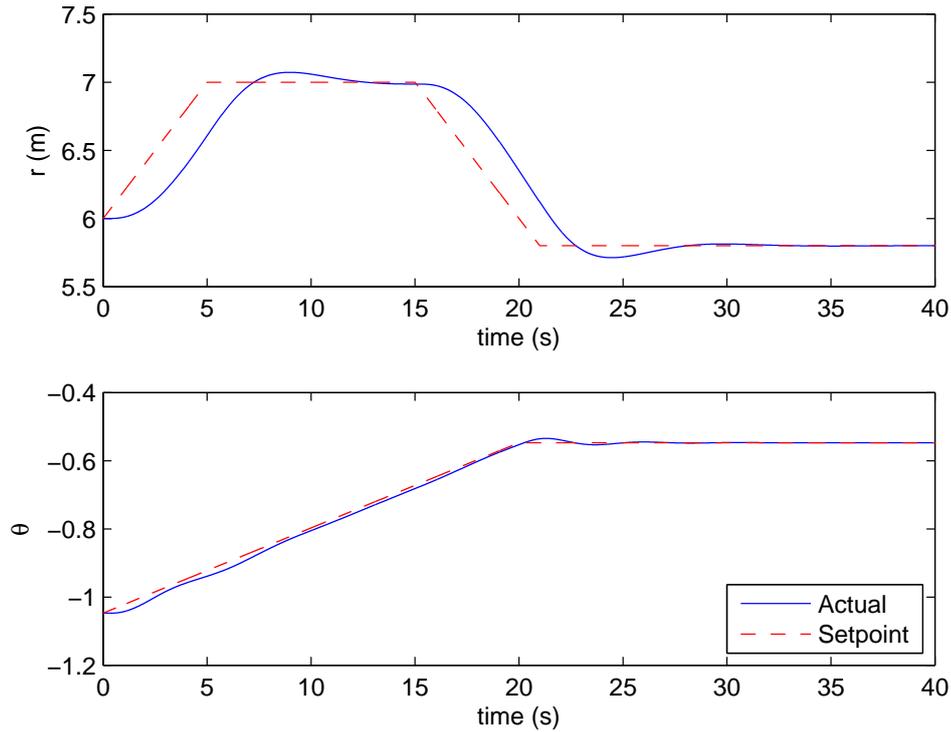
To achieve offset-free tracking, this linear model is augmented with two discrete-time integral states. In terms of the hoist and crowd position estimates from equilibrium, the integrator has dynamics

$$\eta(k+1) = \eta(k) + \begin{bmatrix} \tilde{r}(k) & \tilde{\theta}(k) \end{bmatrix}^T - \begin{bmatrix} \tilde{r}_d(k) & \tilde{\theta}_d(k) \end{bmatrix}^T, \quad (6.40)$$

where  $\eta(\cdot) \in \mathbb{R}^{2 \times 1}$  is a vector of integral states,  $\tilde{r}_d(\cdot)$  is the desired trimmed crowd length and  $\tilde{\theta}_d(\cdot)$  is the desired trimmed crowd angle. A state-feedback controller is now designed using an LQR-like approach, with some extra manual tuning of the integral gains. Assuming that the desired (reference) trajectory inputs are zero, a stabilising gain matrix  $K$  is found such that the feedback law

$$u(k) = -K \begin{bmatrix} x(k) \\ \eta(k) \end{bmatrix} \quad (6.41)$$

provides adequate tracking performance with sufficient damping on the crowd velocity and angular velocity. The numerical value of the gain can be found in Appendix B. Figure 6.6 illustrates the controller's performance when tracking a simple piecewise-linear reference. Tuning to achieve a non-oscillatory response results in the  $r$  reference tracking delay. It is inconsequential, however, as the next stage of system identification allows it to be predicted.



**Figure 6.6:** Performance of trajectory tracking controller

### 6.4.2 Identification of Trajectory Tracking System

Having stabilised the nonlinear system, a linear model is now identified for the closed-loop system, taking the trajectory reference as its input. Using the MATLAB system identification toolbox, a subspace method [VODM96] with prediction error minimisation is employed to identify the model, where it is assumed that there is no input feedforward. This time, the payload and integral states are also included, producing a linear model with seven states. A better linearisation is obtained by taking the square root of the payload trajectory data before identification, as this state varies quadratically with linear changes in  $r$ .

The outputs are chosen to feed forward all of the states, in addition to the Cartesian bucket teeth position deviations from the equilibrium pose. Note that the payload and position of the bucket teeth depend nonlinearly on  $r(\cdot)$  and  $\theta(\cdot)$ , but predictions are required for enforcing constraints later on with MPC, so linear estimates are desirable. The purpose of including the integral states in the linear model is to allow constraining of actuator forces, since they can be inferred from  $-Kx$ .

The linearised closed-loop model adequately predicts most of the states for unseen inputs when compared to the original system. In order to utilise it for trajectory generation, it is downsampled to 1 Hz using a zero-order hold, giving a new discrete time system linear system of the form

$$\hat{x}(k+1) = A\hat{x}(k) + B\hat{u}(k) \quad (6.42)$$

$$\hat{y}(k) = C\hat{x}(k) + D\hat{u}(k), \quad (6.43)$$

where  $A \in \mathbb{R}^{7 \times 7}$ ,  $B \in \mathbb{R}^{7 \times 2}$ ,  $C \in \mathbb{R}^{9 \times 7}$  and  $D = 0_{9 \times 2}$ . Index  $k$  is redefined accordingly to account for the new sampling frequency. The input, state and output vectors are defined by

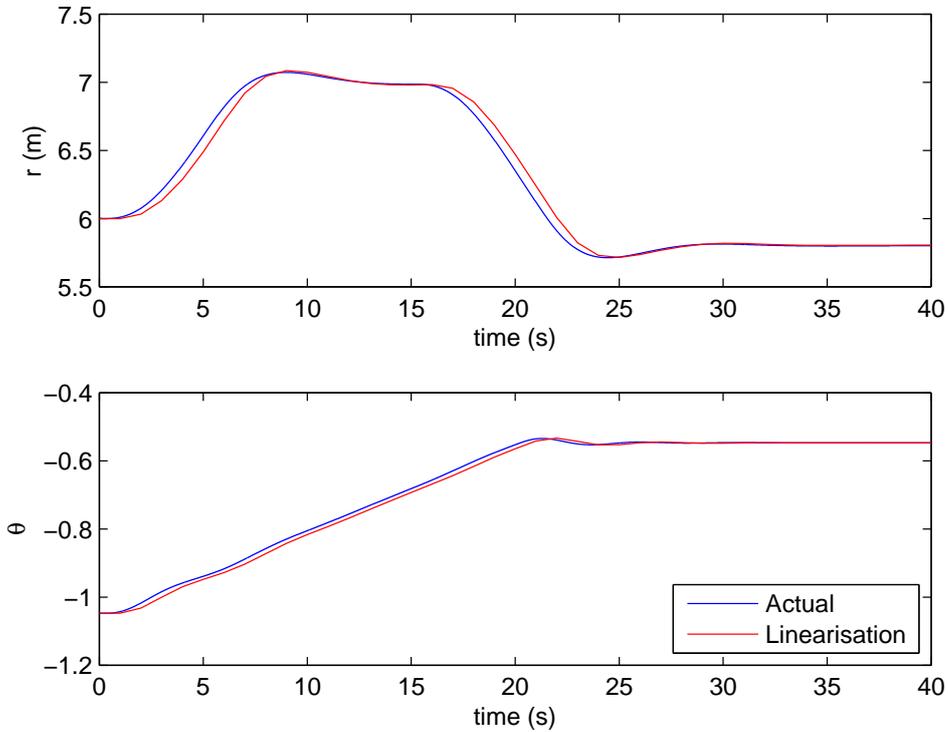
$$\hat{u}(k) = \begin{bmatrix} \tilde{r}_d(k) & \tilde{\theta}_d(k) \end{bmatrix}^T \quad (6.44)$$

$$\hat{x}(k) = \begin{bmatrix} \tilde{r}(k) & \tilde{\theta}(k) & \dot{r}(k) & \dot{\theta}(k) & \sqrt{m_p(k)} & \eta^T(k) \end{bmatrix}^T \quad (6.45)$$

$$\hat{y}(k) = \begin{bmatrix} \hat{x}^T(k) & \tilde{z}_{\text{tip}}^T(k) \end{bmatrix}^T. \quad (6.46)$$

Note that the output vector contains all of the states along with an estimate of the bucket tip position from the trim condition, namely  $\tilde{z}_{\text{tip}}(k)$ .

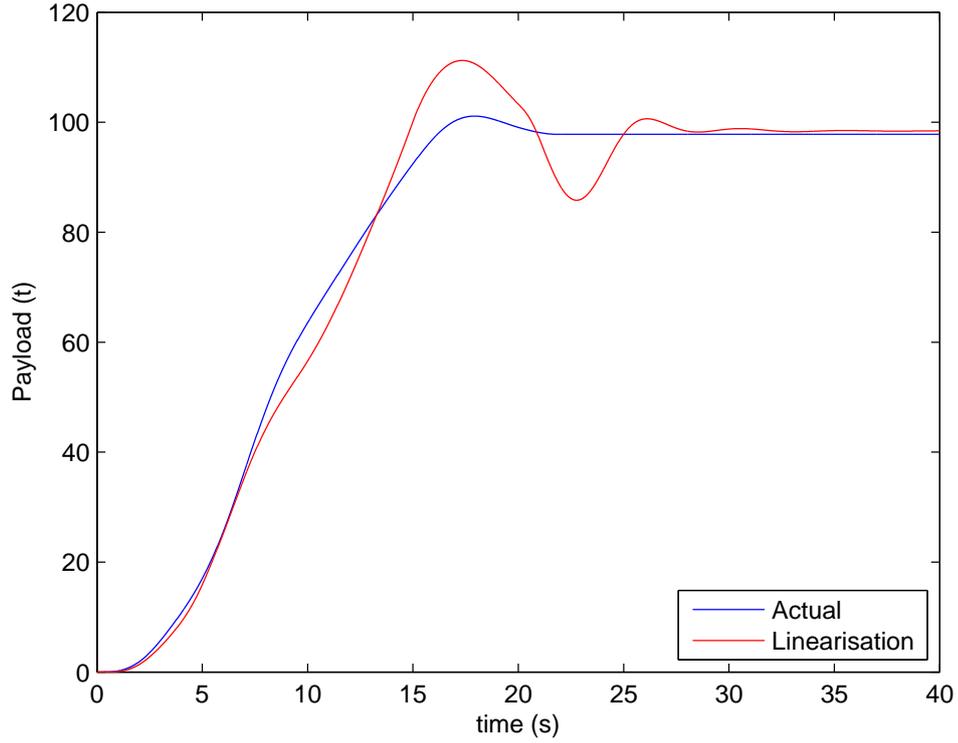
The performance of the downsampled model is illustrated in Figure 6.7, for the crowd length and angle. The performance in tracking all other signals and outputs is similarly close, apart from the payload state, which is pictured in Figure 6.8.



**Figure 6.7:** Performance of identified linear model for crowd length and angle

## 6.5 VH-MPC Controller Design

The result of the preceding identification is a linear model that estimates the true state and bucket tip trajectory of the nonlinear system, given the desired hoist and crowd position signals as inputs. However, costing these inputs directly is not useful. Instead, the model must be modified to allow costing of the change in  $u(\cdot)$  from its previous value, by augmenting



**Figure 6.8:** Performance of identified linear model for payload

additional states [Mac02] to store  $u(k-1)$ . This gives the prediction model

$$\begin{bmatrix} x(k+j+1|k) \\ u(k+j) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k+j|k) \\ u(k+j-1|k) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(k+j|k) \quad (6.47)$$

$$y(k+j|k) = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x(k+j|k) \\ u(k+j-1|k) \end{bmatrix} + D\Delta u(k+j|k), \quad (6.48)$$

where  $\Delta u(k) = u(k) - u(k-1)$ . A maximum prediction horizon length  $\bar{N} = 15$  steps is chosen, as a typical digging cycle time for this material and machine is approximately 10-15 seconds.

### 6.5.1 Constraints

Constraints are now defined on the various system parameters. The critical constraints are given by

$$|\tilde{r}| \leq 5 \text{ m} \quad (6.49)$$

$$|\tilde{r}_d| \leq 4 \text{ m} \quad (6.50)$$

$$|\Delta \tilde{r}_d| \leq 0.2 \text{ m} \quad (6.51)$$

$$0 \leq \Delta \theta_d \leq 0.05 \quad (6.52)$$

$$0 \leq m_p \leq 100 \text{ t} \quad (6.53)$$

$$|\tilde{F}_c| \leq 5 \times 10^5 \text{ N} \quad (6.54)$$

$$|\tilde{F}_h| \leq 1.4 \times 10^6 \text{ N} \quad (6.55)$$

$$\begin{bmatrix} -1 \text{ m} & 6 \text{ m} \end{bmatrix}^T \leq \tilde{z}_{\text{tip}} \leq \begin{bmatrix} -2 \text{ m} & 15 \text{ m} \end{bmatrix}^T. \quad (6.56)$$

The constraint on the crowd position reference  $r_d$  is tighter than the actual constraint on the crowd position  $r$ , to add an extra level of protection against violating this constraint. Constraints on the change in crowd position and angle references ensure smooth trajectories. The change in angle reference is constrained to be greater than zero, as during a standard digging cycle, the hoist should only retract. The payload is limited to a maximum value of 100 t to avoid overloading the machine, and can never be negative. The crowd and hoist force constraints ensure that these forces never reach the stall thresholds for those motors. Finally, the tip position of the bucket is constrained to be within a reasonable region of the state space. This final constraint is the reason that no bound constraints are needed on the angle of the crowd arm. Note that, as with the states, the tilde on the tip position vector indicates a trimmed value.

For dig completion, there are two terminal constraints imposed, in addition to all of the state constraints in (6.49) - (6.56) being satisfied, namely

$$-30^\circ \leq \theta \leq -25^\circ \quad (6.57)$$

$$\begin{bmatrix} 1.66 & -1 \end{bmatrix} \tilde{z}_{\text{tip}} \leq -0.32 \text{ m}. \quad (6.58)$$

The former constraint implies that the crowd arm is at a suitably high angle for dig completion and the latter ensures that the bucket tip has emerged from the face. Given that the linear payload estimate is not highly accurate, no terminal payload constraint is imposed. However, soft constraints are imposed through the cost function, as explained below.

As with the inverted pendulum example in Chapter 5, some constraint tightening is applied to help mitigate the effect of linearisation error. Assuming appropriate units on each channel, a disturbance vector of the form

$$\mathcal{W} = \left\{ w \mid |w| \leq 1 \times 10^{-4} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \times 10^4 & 1 & 1 & 1 & 1 \end{bmatrix}^T \right\}, \quad (6.59)$$

is introduced, which applies a large amount of tightening to the payload and a smaller margin of tightening to the remaining states. The payload disturbance is made as large as possible whilst still ensuring non-empty tightened sets. This includes the tightened sets for feasible contingencies and move blocking, which will be presented in Subsections 6.6.2 and 6.6.3. Given the high state dimension and long prediction horizon, the heuristic optimisation approach presented in Section 5.5, using a 1-norm cost, is used to calculate the policy. The region of attraction is not critical for this scenario, as a fixed starting state has been assumed.

### 6.5.2 Cost Function

The cost function is designed to weight the completion time with commanded input changes and state-based operating costs, which are effectively soft constraints. The effect is to “shape” the digging trajectory to encourage sufficient penetration depth of the material and adequate payload acquisition. The cost function has the form

$$\begin{aligned} \sum_{j=0}^{N(k)-1} & 1 + 0.01 \|u(k+j|k)\|_1 + 0.8 \|\dot{r}(k+j|k)\|_1 + 0.3 \|\dot{\theta}(k+j|k)\|_1 \\ & + 0.1 \left\| \begin{bmatrix} 1.66 & -1 \end{bmatrix} \tilde{z}_{\text{tip}}(k+j|k) - 1.29 \right\|_1 \\ & + 0.01 \left\| \sqrt{m_p(k+j|k)} - \sqrt{65 \times 10^3} \right\|_1, \quad (6.60) \end{aligned}$$

which can be written in the form (3.40) by appropriately choosing the matrix  $\Theta$  and output reference  $y_r$ . Notice the square root of the payload appearing in the last term of the cost function, which is a result of having identified a system based on this quantity rather than the payload itself. The cost function has the following effects:

- penalise time to completion
- penalise the value of the change in crowd position and angle reference
- penalise high crowd translational and rotational velocities to encourage smooth trajectories
- encourage the face to be penetrated to a depth of 0.67 m.
- encourage a payload close to 65 t.

## 6.6 Simulation

Using the controller designed in the previous section, the system is simulated, starting with all states of the prediction model initialised to zero (i.e. at their trim values for the actual model). The first set of simulations consider a normal digging scenario. The next two sets consider the implementation of a feasible contingency and move blocking respectively.

### 6.6.1 Normal Operation

Figure 6.9 shows the predicted tip trajectory from the linear model plotted with the actual trajectory obtained when using the VH-MPC controller on the stabilised nonlinear model. It can be seen that a realistic trajectory is generated. The trajectories for the crowd position and angle are shown in Figure 6.10, together with the reference signal generated by the VH-MPC controller. The payload acquisition, shown in Figure 6.11 is slightly less than the target value,

but this is partly due to the error in estimating this state in the linear model. As shown in Figure 6.12, the actuator forces respect the constraints imposed on them. The components of the cutting force are displayed in Figure 6.13. For the actuator forces and the cutting force, the values are shown only for the time whilst the bucket is in the material.

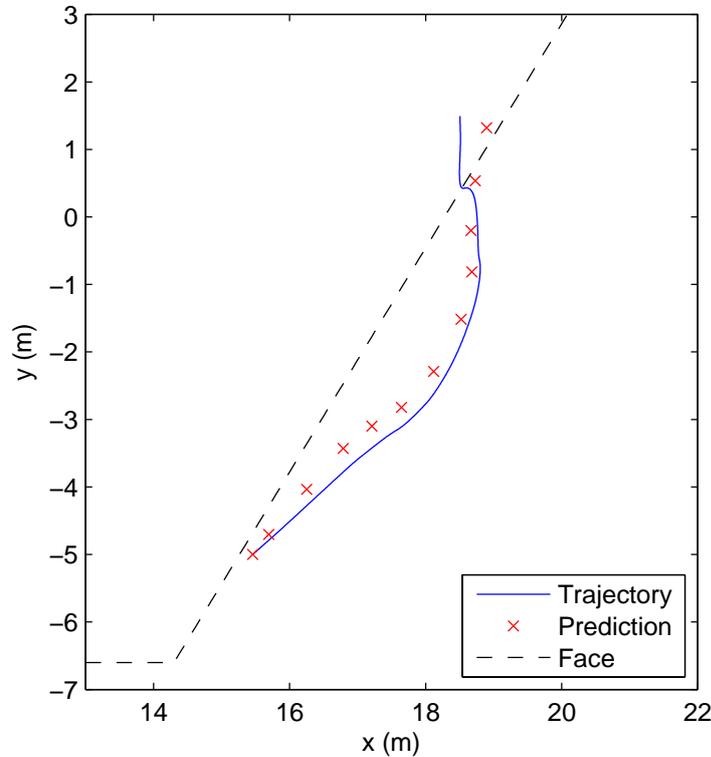
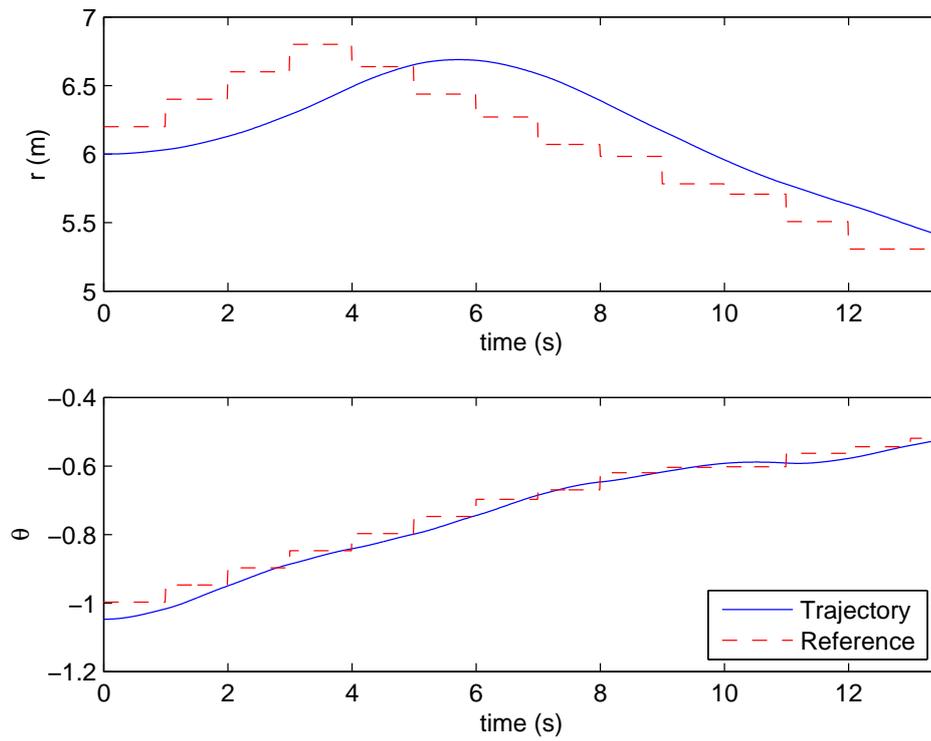


Figure 6.9: Actual trajectory and prediction from linear model

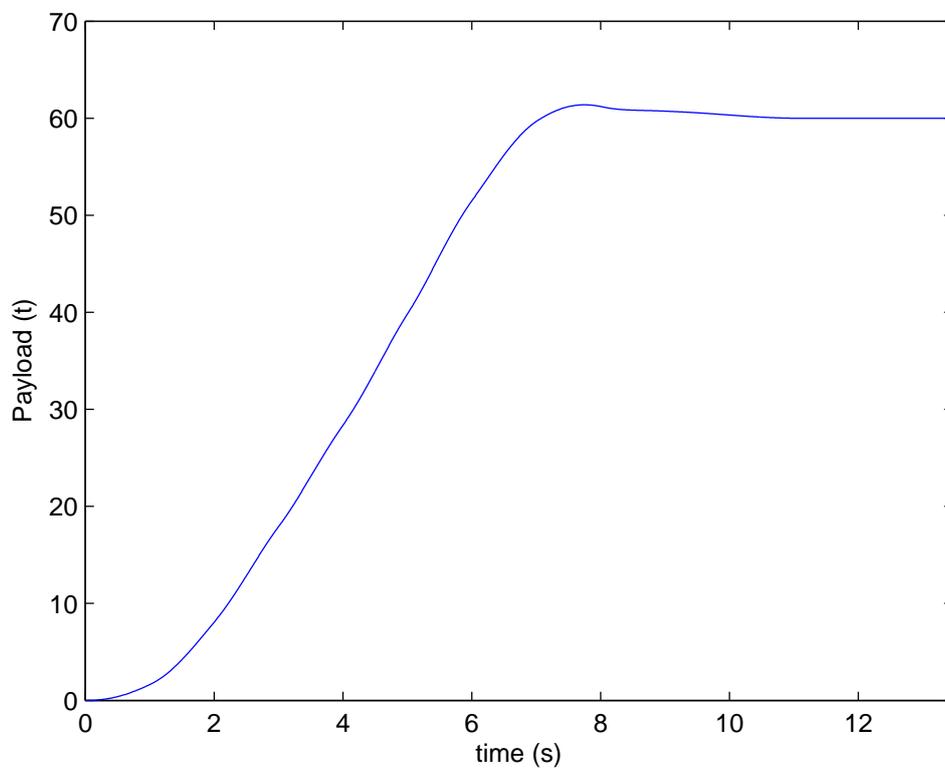
### 6.6.2 Stall Contingency

A contingency scenario is now implemented on the shovel, to handle a crowd stalling condition. This could be caused by a large particle within the material, or a number of small particles becoming lodged between the bucket teeth. Under such conditions, the crowd is unable to extend and can only retract. Such stalls are typically handled by retracting the crowd whilst still retracting the hoist. In the case of a severe stall, the bucket is to be withdrawn from the face. The contingency therefore constrains the crowd reference to be non-negative and ensures that a pull-back trajectory of at most 3 s exists to the face of the material.

Using the theory developed in Chapter 3, the same disturbance set (6.59) and optimised CT policy is used for the primary and contingency constraints, which now allow a slightly higher maximum magnitude for the change in crowd and hoist reference (0.3 m and 0.7 respectively). Figure 6.14 illustrates the effect on the resulting trajectory under normal operation. As expected, the shovel now digs closer to the face to ensure the existence of the pullout trajectory. The actual pullout trajectory is illustrated in green, assuming a crowd stall at 6 s. Figure 6.15 shows that the bucket fill is slightly lower when the contingency is required to be available.



**Figure 6.10:** Crowd position and angle compared to reference



**Figure 6.11:** Payload acquisition

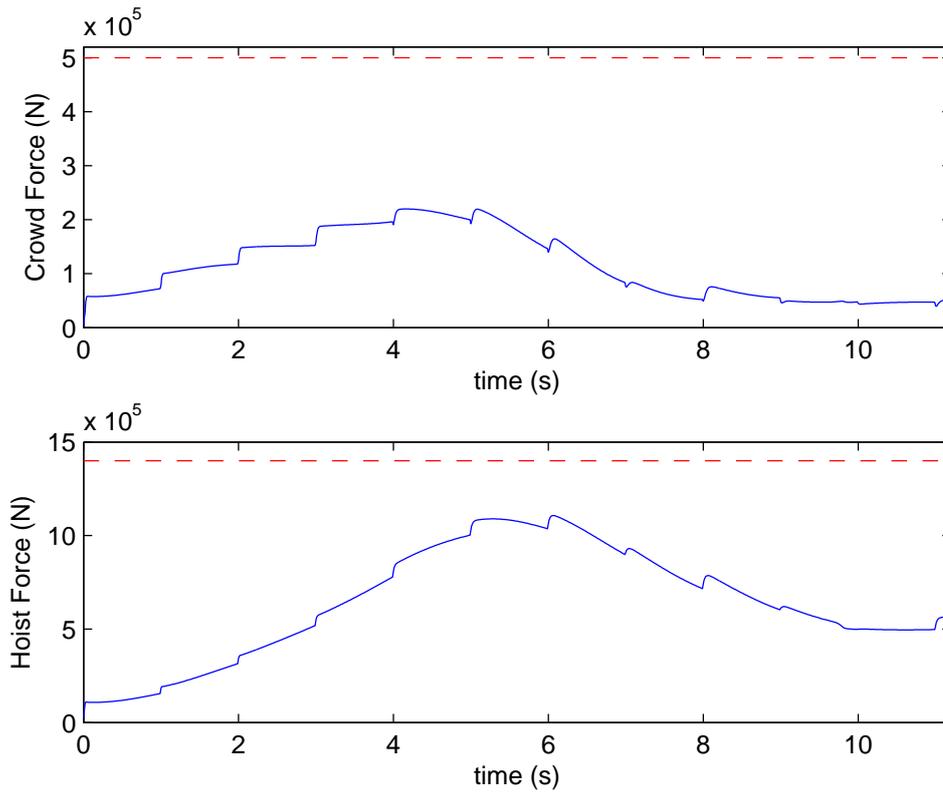


Figure 6.12: Hoist and crowd forces from the trim point

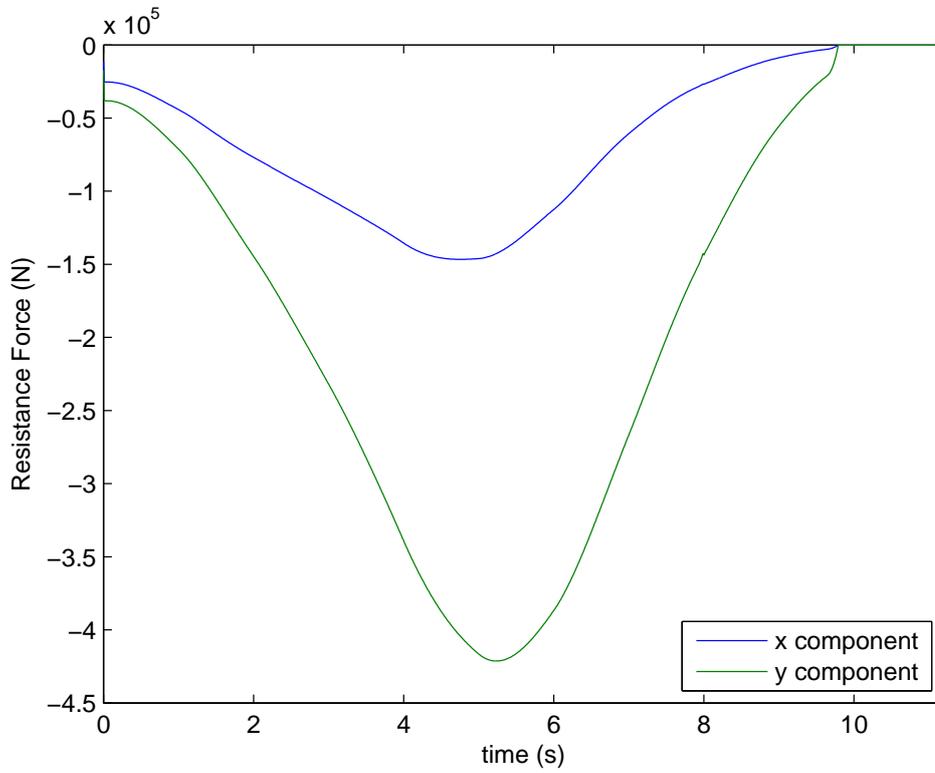


Figure 6.13: Cutting resistance force

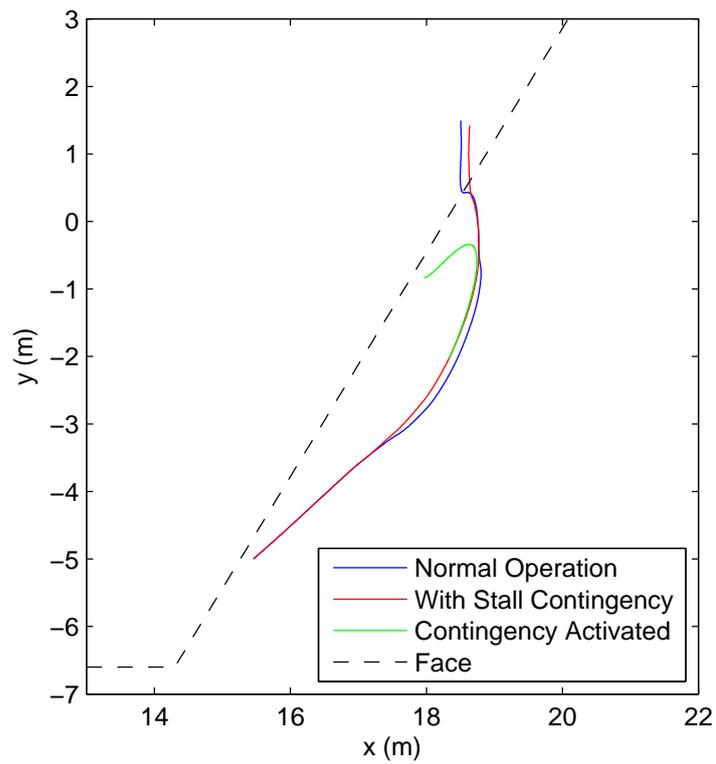


Figure 6.14: Trajectory comparison with crowd stall contingency

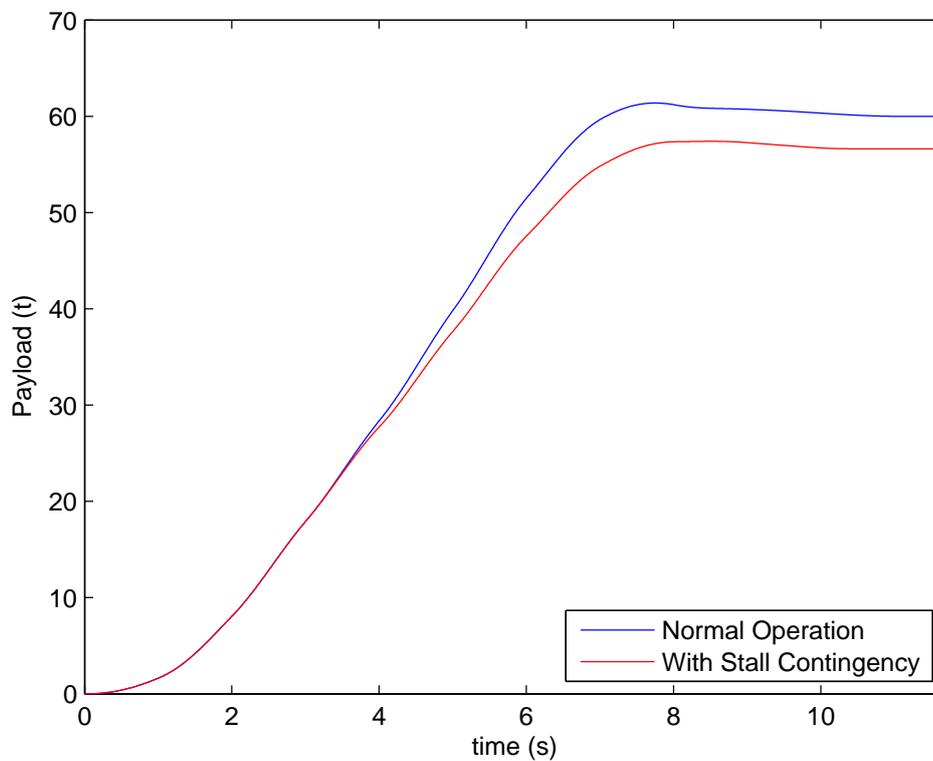


Figure 6.15: Payload comparison with crowd stall contingency

### 6.6.3 Move Blocking

It is interesting to investigate the effect of move blocking on the closed loop trajectories. Using the input and state condensed partitioned horizon strategy formulated in Chapter 4, with initial sub-horizon length 5, terminal sub-horizon length 1 and medial sub-horizon block length 9, the normal operation scenario is considered for simulation. The resulting trajectories are shown in Figure 6.16 and the payloads are compared in Figure 6.17. In this case, a reduction in the number of input variables by over 50% produces only a marginal reduction in the resulting payload, but a longer completion time. The maximum solution time per timestep with move blocking is 0.1814 s (serial computation), whereas without blocking it is 0.4437 s; this is a reduction of almost 60% with blocking.

## 6.7 Conclusion

This chapter has demonstrated how VH-MPC can be applied to trajectory generation for surface excavation using a rope shovel. It has also shown the application of the theory developed in the previous chapters, namely feasible contingencies, move blocking and optimal constraint tightening, to a higher-dimensional model with a longer prediction horizon. Reasonably realistic trajectories have been obtained, confirming that VH-MPC has applicability beyond vehicle path-planning problems.

## 6.8 Acknowledgements

The work in this chapter is based upon a paper presented by the author at the 49th IEEE Conference on Decision and Control [SM10]. The simulations in this chapter were carried out using MATLAB R2011b, YALMIP [Löf04], the Multi-parametric Toolbox [KGB04], CPLEX [CPL09] and the Invariant Set Toolbox [Ker00].

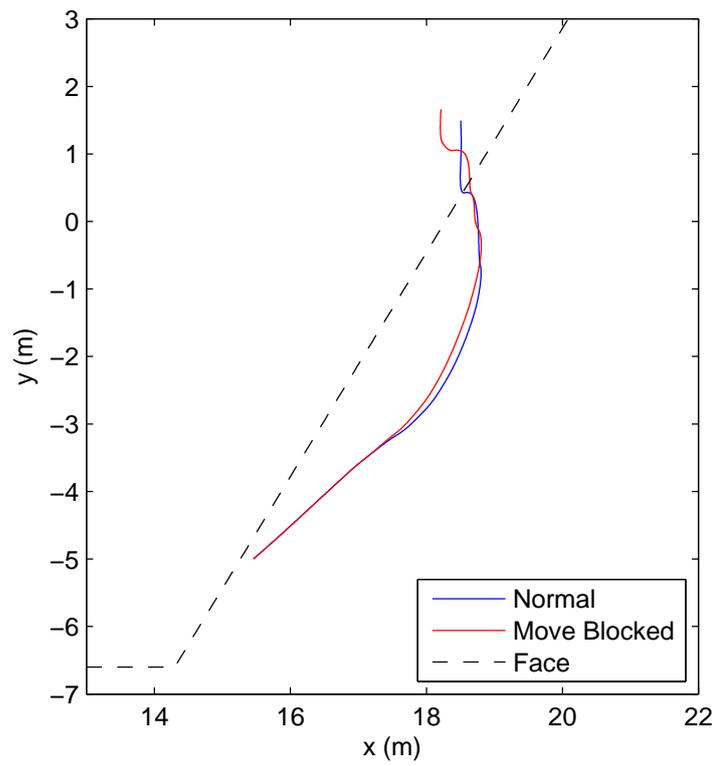


Figure 6.16: Trajectory comparison with move blocking

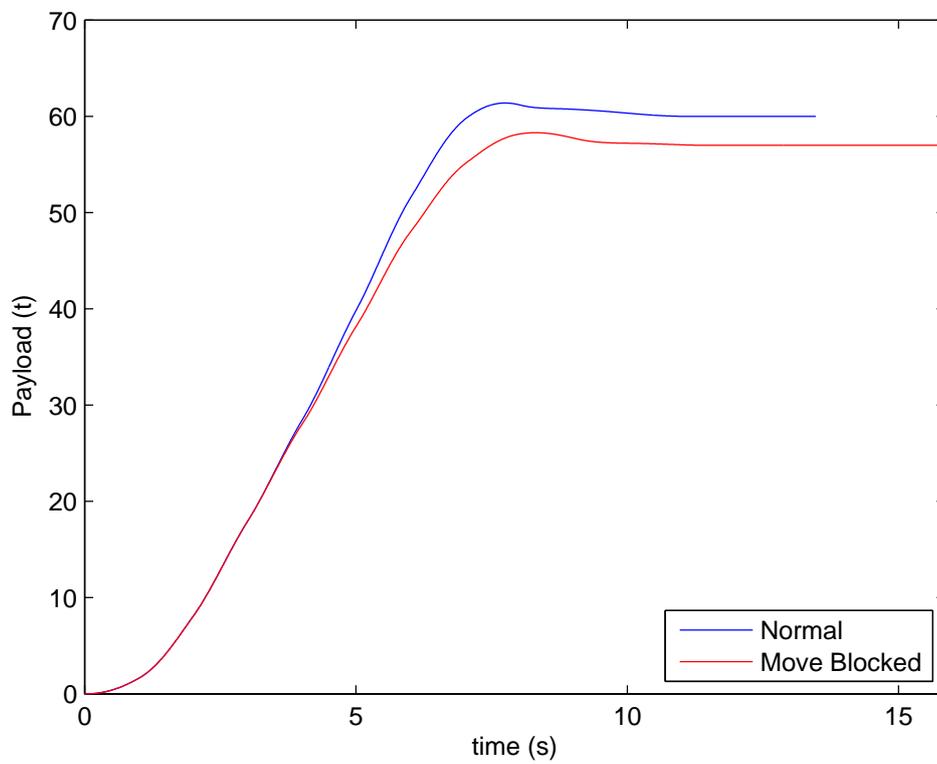


Figure 6.17: Payload comparison with move blocking



# Chapter 7

## Conclusion

This thesis has developed new techniques and algorithms for robustness and optimality in variable horizon model predictive control. These include: the development of control algorithms robust to changes in system dynamics, constraints and objectives; an algorithm to reduce computational complexity whilst maintaining robustness guarantees; and a technique to optimise a controller's region of attraction when robustness guarantees are required. This chapter summarises these key contributions and motivates directions for future work.

### 7.1 Contributions

This thesis makes four main contributions to the body of academic knowledge. These contributions are summarised in the subsections below.

#### 7.1.1 Feasible Contingencies

To address control scenarios where system dynamics, constraints and control objectives can change, the notion of a controller with feasible contingencies has been introduced for completion problems. The key feature of this work is that, although the effect of the contingency is known, the activation time is not known a priori. Criteria for nominal and robust contingency availability at all times have been proposed. Sufficient conditions have then been used to design a VH-MPC controller to fulfil these criteria, whilst maintaining robust recursive feasibility and finite time completion guarantees. A double constraint tightening approach has been formulated to ensure robustness before and after contingency activation.

Various contingency scenarios have then been presented, to encompass a wide gamut of potential applications. These scenarios include multiple, costed, prioritised and state-dependent contingencies. Two systems have then been used to showcase the application of the developed theory. A simple vehicle example has been used to illustrate the feasible contingency idea on a fault-tolerant control problem. In addition, a cart pendulum model has shown the applicability of feasible contingencies to regulation problems.

### 7.1.2 Complexity-Reduced VH-MPC with Move Blocking

Given the computational complexity of VH-MPC optimisation problems over long time horizons, new theory has been developed to apply input move blocking to completion problems, to reduce computational complexity. A new notation has been proposed to describe the action of move blocking; this has been subsequently applied to the design of a robust move-blocked variable horizon controller. A modified technique for tightening constraints has then been formulated, which preserves robust recursive feasibility and finite-time completion guarantees, subject to certain conditions on the cost function.

As the new framework provides flexibility in the choice of blocking regime, a particular form known as the partitioned horizon has been described, which enables fine-grained control action at the beginning and end of a control manoeuvre, whilst implementing move blocking in the middle for complexity reduction. A simple example has then been used to illustrate the effect of different block lengths when using this regime. From this example, it has been observed that a reduction in decision variables, whilst reducing the computational burden as expected, does not necessarily increase the cost.

### 7.1.3 Optimal Constraint Tightening Policies

A new algorithm for designing disturbance-feedback constraint tightening policies has been presented, which improves the region of attraction for VH-MPC controllers. The algorithm has been designed around the solution of a convex optimisation problem, affinely parameterised in terms of the constraint tightening policy. By maximising the volume of an inner ellipsoidal approximation to the true ROA, improvements in region of attraction size have been demonstrated, in comparison to other popular approaches to designing CT policies. It has also been shown how the algorithm proposed in this thesis can reduce the maximum prediction horizon length. Methods have been proposed for dealing with high state-dimensions, where the ellipsoid maximisation method may struggle computationally.

### 7.1.4 Case Study on Surface Excavation

A reasonably realistic model of a rope shovel used in surface excavation has been developed to illustrate the efficacy of VH-MPC and the theoretical techniques developed in this thesis. The model has applied VH-MPC to generate trajectories which are then tracked by an inner pre-stabilising controller. The resulting realistic trajectories have motivated new applications of VH-MPC for autonomy that go beyond the realm of vehicle path planning.

## 7.2 Future Research Directions

This section investigates future avenues of research stemming from the results presented in this thesis.

### 7.2.1 Uncertain Contingencies

The work in this thesis assumes that the system dynamics after contingency activation are well defined, apart from some bounded state disturbance. It is interesting to investigate to what degree contingency availability can be guaranteed for uncertain systems, where the state space matrices describing the dynamics after contingency activation are contained within some polytope. This would be especially useful in fault-tolerant control scenarios, where the mode of failure might not be known exactly.

### 7.2.2 Optimised & Multiplexed Move Blocking

The partitioned horizon has been proposed as an example blocking regime, but it has been shown that the closed loop cost performance when applying the regime is not monotonically related to the number of decision variables. This poses the problem of determining the closed-loop cost-optimal blocking regime under disturbances. It may be difficult to determine this in general, as the optimal regime is likely to be system and problem dependent. In this case, some heuristics or rules of thumb for selecting blocking regimes for certain classes of problems will be desirable.

Another avenue of research related to move blocking is the application of different regimes on each input channel. More research is required to design a new constraint tightening algorithm to maintain recursive feasibility and finite-time completion guarantees with this scheme. Finding the optimal blocking regime for each channel also presents another research opportunity.

### 7.2.3 Cost Optimal Constraint Tightening Policies

The algorithms presented for optimal constraint tightening in this thesis have focused on improving the region of attraction. In some applications however, the expected closed loop cost may be more important. A worthwhile avenue of investigation is policy optimisation for some weighted sum of ROA size and expected closed loop cost. If the set of initial states is small, then it would also be desirable to find the cost optimal policy where the ROA is just large enough to contain this set.

### 7.2.4 Combinations of Strategies

Whilst the key contributions of feasible contingencies, move blocking and optimal constraint tightening have been presented separately, further research will allow them to be combined. Move blocking on the primary and contingency trajectories would reduce the complexity of the variable horizon feasible contingency algorithm. Also, the optimal constraint tightening algorithm could be modified to account for move blocking constraints, which would allow the ROA volume to be maximised under blocking. The problem of finding the optimal combination

of blocking and constraint tightening is a future research direction that follows naturally from this.

### 7.2.5 Shovel & Material Model Generalisation

In Chapter 6, a single linearisation has been obtained for the shovel and material system, which assumed a particular face profile. To generalise this to arbitrary profiles, an algorithm is needed to identify an appropriate linearisation online, profiling the current face prior to commencing the digging cycle. The algorithm would also have to appropriately tune the cost function of the VH-MPC controller, depending on the identified model.

To further generalise this algorithm to different materials, other modelling approaches will be required. Whilst Mohr-Coulomb theory is suitable for homogeneous materials, it does not adequately model blasted materials which can have particles on several different length scales. To capture the behaviour of such materials, Discrete Element Modelling (DEM) can be used. First proposed by [Cun71] to model the failure of brittle blocks of rock, DEM is an adaptation of methods used widely in physics and chemistry to model liquid and gas behaviours [HF06]. Using a suitable software package, the dynamic model of the shovel could be interfaced to the material model, and a linearisation identified for the combined system before applying MPC.

# Bibliography

- [BF08] A. Bedford and W. L. Fowler. *Engineering Mechanics : Statics & Dynamics*. 5th. London: Pearson Education, 2008.
- [BFT04] A. Bemporad, C. Filippi and F. D. Torrisi. “Inner and outer approximations of polytopes using boxes”. In: *Computational Geometry* 27.2 (2004), pp. 151–178.
- [BH07] L. Breger and J. P. How. “Powered safe abort for autonomous rendezvous of spacecraft”. In: *AIAA Guidance, Navigation and Control Conference*. 2007.
- [BH08] L. Breger and J. How. “Safe trajectories for autonomous rendezvous of spacecraft”. In: *Journal of Guidance, Control and Dynamics* 31.5 (2008), pp. 1478–1496.
- [BH10] A. Barvinok and J. Hartigan. “Maximum entropy gaussian approximations for the number of integer points and volumes of polytopes”. In: *Advances in Applied Mathematics* 45.2 (2010), pp. 252–289.
- [BM99] A. Bemporad and M. Morari. “Control of systems integrating logic, dynamics, and constraints”. In: *Automatica* 35.3 (1999), pp. 407–427.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [Boy+94] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: Society for Industrial and Applied Mathematics, 1994.
- [CLM96] L. Chisci, A. Lombardi and E. Mosca. “Dual receding horizon control of constrained discrete-time systems”. In: *European Journal of Control* 2 (1996), pp. 278–285.
- [CPL09] CPLEX. *IBM ILOG CPLEX V12.1 User’s Manual*. 2009.
- [CRZ01] L. Chisci, J. A. Rossiter and G. Zappa. “Systems with persistent disturbances: predictive control with restricted constraints”. In: *Automatica* 37.7 (July 2001), pp. 1019–1028.
- [CS99] H. Cannon and S. Singh. “Models for automated earthmoving”. In: *Experimental Robotics VI, Lecture Notes in Control and Information Sciences*. Springer Verlag, 1999, pp. 183–192.
- [Cag+07] R. Cagienard, P. Grieder, E. Kerrigan and M. Morari. “Move blocking strategies in receding horizon control”. In: *Journal of Process Control* 17.6 (2007), pp. 563–570.

- [Cam09] C Campbell. "Model Predictive Control for Stability Augmentation and Guidance of a Flexible Micro Air Vehicle." MSc Thesis. University of Cambridge, 2009.
- [Car+08] J. M. Carson, B. Açikmeşe, R. M. Murray and D. G. MacMynowski. "A robust model predictive control algorithm with a reactive safety mode". In: *Proceedings of the 17th IFAC World Congress*. 2008.
- [Cla57] K. Clark. *Bulk Densities, Porosities and Liquid Saturations of Good Grade Athabasca Oil Sands*. Tech. rep. Research Council of Alberta, 1957.
- [Col07] P. M. Collins. "Geomechanical effects on the SAGD process". In: *SPE Reservoir Evaluation & Engineering* 10.4 (2007), pp. 367–375.
- [Com74] L Comtet. *Advanced Combinatorics: The Art of Finite and Infinite Expansions, rev. enl. ed.* Dordrecht, Netherlands: Reidel, 1974, pp. 176–177.
- [Cun71] P. Cundall. "Distinct element models of rock and soil structure". In: *Analytical and Computational Methods in Engineering Rock Mechanics*. Ed. by E. Brown. London: Unwin Publishers, 1971, pp. 129–163.
- [DE73] G. B. Dantzig and B. C. Eaves. "Fourier-motzkin elimination and its dual". In: *Journal of Combinatorial Theory, Series A* 14.3 (1973), pp. 288 –297.
- [Del34] B. N. Delaunay. "Sur la sphère vide". In: *Bulletin of Academy of Sciences of the USSR* 6 (1934), pp. 793–800.
- [FHAO05] S. Frimpong, Y. Hu and K. Awuah-Offei. "Mechanics of cable shovel-formation interactions in surface mining excavations". In: *Journal of Terramechanics* 42.1 (2005), pp. 15 –33.
- [FL07] S. Frimpong and Y. Li. "Stress loading of the cable shovel boom under in-situ digging conditions". In: *Engineering Failure Analysis* 14.4 (2007), pp. 702 –715.
- [GI07] R. Gondhalekar and J. Imura. "Recursive feasibility guarantees in move-blocking MPC". In: *Proceedings of the 46th IEEE Conference on Decision and Control*. 2007, pp. 1374–1379.
- [GIK09] R. Gondhalekar, J. Imura and K. Kashima. "Controlled invariant feasibility – a general approach to enforcing strong feasibility in MPC applied to move-blocking". In: *Automatica* 45.12 (2009), pp. 2869 –2875.
- [GKM06] P. J. Goulart, E. C. Kerrigan and J. M. Maciejowski. "Optimization over state feedback policies for robust control with constraints". In: *Automatica* 42.4 (2006), pp. 523 –533.
- [GKR97] J. R. Gossner, B. Kouvaritakis and J. A. Rossiter. "Stable generalized predictive control with constraints and bounded disturbances". In: *Automatica* 33.4 (Apr. 1997), pp. 551–568.
- [Glp] *GNU Linear Programming Toolkit v4.47*. 2010.

- [HF06] S. Hardy and E. Finch. “Discrete element modelling of the influence of cover strength on basement-involved fault-propagation folding”. In: *Tectonophysics* 415.1-4 (2006), pp. 225–238.
- [HM09] S. Heubach and T. Mansour. *Combinatorics of Compositions and Words*. CRC Press, 2009.
- [Har10] E. N. Hartley. “Model Predictive Control for Spacecraft Rendezvous”. PhD thesis. University of Cambridge, 2010.
- [Hay11] B. Hayes. “An adventure in the nth dimension”. In: *American Scientist* 99 (2011), pp. 442–446.
- [JKM04] C. N. Jones, E. C. Kerrigan and J. M. Maciejowski. *Equality Set Projection: A new algorithm for the projection of polytopes in halfspace representation*. Tech. rep. CUED/F-INFENG/TR.463. Department of Engineering, University of Cambridge, 2004.
- [KBM96] M. V. Kothare, V. Balakrishnan and M. Morari. “Robust constrained model predictive control using linear matrix inequalities”. In: *Automatica* 32.10 (1996), pp. 1361–1379.
- [KG88] S. S. Keerthi and E. G. Gilbert. “Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations”. In: *Journal of Optimization Theory and Applications* 57.2 (May 1988), pp. 265–293.
- [KG95] I. Kolmanovsky and E. Gilbert. “Maximal output admissible sets for discrete-time systems with disturbance inputs”. In: *Proceedings of the American Control Conference*. Vol. 3. 1995, pp. 1995–1999.
- [KGB04] M. Kvasnica, P. Grieder and M. Baotić. *Multi-Parametric Toolbox (MPT)*. 2004.
- [KM04] E. C. Kerrigan and J. M. Maciejowski. “Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution”. In: *International Journal of Robust and Nonlinear Control* 14.4 (2004), pp. 395–413.
- [KRH07] Y. Kuwata, A. Richards and J. How. “Robust receding horizon control using generalized constraint tightening”. In: *Proceedings of the 26th American Control Conference*. 2007, pp. 4482–4487.
- [Ker00] E. C. Kerrigan. “Robust Constraint Satisfaction: Invariant Sets and Predictive Control”. PhD thesis. University of Cambridge, 2000.
- [LSC98] O. Luengo, S. Singh and H. Cannon. “Modeling and identification of soil-tool interaction in automated excavation”. In: *Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 3. 1998, pp. 1900–1906.
- [Lan+04] W. Langson, I. Chrysochoos, S. V. Rakovic and D. Q. Mayne. “Robust model predictive control using tubes”. In: *Automatica* 40.1 (Jan. 2004), pp. 125–133.
- [Löf04] J. Löfberg. “YALMIP : a toolbox for modeling and optimization in MATLAB”. In: *Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004, pp. 284–289.

- [ML01] D. Mayne and W. Langson. "Robustifying model predictive control of constrained linear systems". In: *Electronics Letters* 37.23 (2001), pp. 1422–1423.
- [MM93] H. Michalska and D. Mayne. "Robust receding horizon control of constrained non-linear systems". In: *IEEE Transactions on Automatic Control* 38.11 (1993), pp. 1623–1633.
- [Mac02] J. M. Maciejowski. *Predictive control with constraints*. Essex, England: Prentice Hall, 2002.
- [May+00] D. Mayne, J. Rawlings, C. Rao and P. Scokaert. "Constrained model predictive control: stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.
- [May+11] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk and P. Falugi. "Tube-based robust non-linear model predictive control". In: *International Journal of Robust and Nonlinear Control* 21.11 (2011), pp. 1341–1353.
- [Old+09] F. Oldewurtel, R. Gondhalekar, C. Jones and M. Morari. "Blocking parameterizations for improving the computational tractability of affine disturbance feedback MPC problems". In: *Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference*. 2009, pp. 7381–7386.
- [RH03] A. Richards and J. How. "Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility". In: *Proceedings of the American Control Conference*. Vol. 5. 2003, pp. 4034–4040.
- [RH06a] A. Richards and J. P. How. "Robust variable horizon model predictive control for vehicle maneuvering". In: *International Journal of Robust & Nonlinear Control* 16.7 (2006), pp. 333–351.
- [RH06b] A. Richards and J. How. "Robust stable model predictive control with constraint tightening". In: *Proceedings of the American Control Conference*. 2006, pp. 1557–1562.
- [RKC00] J. Rossiter, B. Kouvaritakis and M. Cannon. "Triple mode control in mpc". In: *Proceedings of the American Control Conference*. Vol. 6. 2000, pp. 3753–3757.
- [RM09] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Madison, WI.: Nob Hill Publishing, 2009.
- [Rak+06] S. Rakovic, E. Kerrigan, D. Mayne and J. Lygeros. "Reachability analysis of discrete-time systems with disturbances". In: *IEEE Transactions on Automatic Control* 51.4 (2006), pp. 546–561.
- [Ree64] A. Reece. "The fundamental equation of earthmoving mechanics". In: *Proceedings of the Institution of Mechanical Engineers*. Vol. 179. 3F. 1964, pp. 16–22.
- [Ric05a] A. Richards. "Robust model predictive control for time-varying systems". In: *44th IEEE Conference on Decision and Control and 2005 European Control Conference*. 2005, pp. 3747–3752.

- [Ric05b] A. G. Richards. "Robust Constrained Model Predictive Control". PhD thesis. Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics., 2005.
- [SHF04] T. Schouwenaars, J. How and E. Feron. "Receding horizon path planning with implicit safety guarantees". In: *Proceedings of the 2004 American Control Conference*. Vol. 6. 2004, pp. 5576–5581.
- [SHV05] M. W. Spong, S. Hutchinson and M. Vidyasagar. *Robot Modelling and Control*. Wiley, 2005.
- [SM10] R. C. Shekhar and J. M. Maciejowski. "Surface excavation with model predictive control". In: *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*. 2010, pp. 5239–5244.
- [SM11] R. C. Shekhar and J. M. Maciejowski. "Robust predictive control with feasible contingencies for fault tolerance". In: *Proceedings of the 18th IFAC World Congress*. 2011, pp. 4666–4671.
- [SM12] R. C. Shekhar and J. M. Maciejowski. "Robust variable horizon MPC with move blocking". In: *Systems & Control Letters* 61.4 (2012), pp. 587–594.
- [SM98] P. Scokaert and D. Mayne. "Min-max feedback model predictive control for constrained linear systems". In: *IEEE Transactions on Automatic Control* 43.8 (1998), pp. 1136–1142.
- [SMR99] P. Scokaert, D. Mayne and J. Rawlings. "Suboptimal model predictive control (feasibility implies stability)". In: *IEEE Transactions on Automatic Control* 44.3 (1999), pp. 648–654.
- [Sch+01] T. Schouwenaars, E. Feron, B. de Moor and J. How. "Mixed integer programming for multi-vehicle path planning". In: *Proceedings of the European Control Conference*. 2001.
- [Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. Chichester, UK: Wiley-Interscience, 1986.
- [Spa92] J. Spall. "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation". In: *IEEE Transactions on Automatic Control* 37.3 (1992), pp. 332–341.
- [TTT99] K. C. Toh, M. Todd and R. H. Tütüncü. "SDPT3 - a MATLAB software package for semidefinite programming". In: *Optimization Methods and Software* 11 (1999), pp. 545–581.
- [Tro09] P. A. Trodden. "Robust Distributed Control of Constrained Linear Systems". PhD thesis. University of Bristol, 2009.
- [VODM96] P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996.

- [WB10] Y. Wang and S. Boyd. "Fast model predictive control using online optimization". In: *IEEE Transactions on Control Systems Technology* 18.2 (2010), pp. 267–278.
- [ZG03] Y. Zhang and L. Gao. "On numerical solution of the maximum volume ellipsoid problem". In: *SIAM Journal on Optimization* 14.1 (2003), pp. 53–76.
- [P&03] P&H MinePro Services. *Peak Performance Practices: Excavator Selection*. 2003.

# Appendix A

## Constraint Tightening Addenda

### A.1 Derivation of the Max-Dist Policy

The aim of this policy is to replace the disturbance set with one of the form

$$\mathcal{W} = \{w \mid \zeta w \leq \beta\theta\}, \quad (\text{A.1})$$

for some variable scaling factor  $\beta > 0$ . Given the effect of the scaling factor  $\beta$  on  $s(\cdot)$  and  $\sigma(\cdot)$  in (5.17) and (5.18), the non-emptiness constraints (5.61) need to be reformulated as

$$\exists y, x : \quad (\text{A.2})$$

$$Ey + \epsilon_y \left( \sum_{i=1}^q \|E_i\| \right)^T \leq f - \beta s(N-1) \quad (\text{A.3})$$

$$Gx + \epsilon_x \left( \sum_{i=1}^q \|G_i\| \right)^T \leq h - \beta\sigma(N).$$

Then, this implies that

$$\exists y, x : \quad (\text{A.4})$$

$$\frac{1}{\beta}Ey + \frac{1}{\beta}\epsilon_y \left( \sum_{i=1}^q \|E_i\| \right)^T \leq \frac{1}{\beta}f - s(N-1) \quad (\text{A.5})$$

$$\frac{1}{\beta}Gx + \frac{1}{\beta}\epsilon_x \left( \sum_{i=1}^q \|G_i\| \right)^T \leq \frac{1}{\beta}h - \sigma(N).$$

To retain convexity, define the variables  $\delta = 1/\beta$ ,  $\tilde{y} = y/\beta$  and  $\tilde{x} = x/\beta$  to give the equivalent constraints

$$\exists \tilde{y}, \tilde{x} : \quad (\text{A.6})$$

$$E\tilde{y} \leq \delta \left( f - \epsilon_y \left( \sum_{i=1}^q \|E_i\| \right)^T \right) - s(N-1) \quad (\text{A.7})$$

$$G\tilde{x} \leq \delta \left( h - \epsilon_x \left( \sum_{i=1}^q \|G_i\| \right)^T \right) - \sigma(N).$$

It is evident that maximising  $\beta$  and therefore dilating the disturbance set is equivalent to minimising  $\delta$ . Hence, the policy max-dist is found by solving the convex optimisation problem

$$\mathcal{P}_{\text{max-dist}}(N) = \begin{cases} \min_{\mathcal{P}, \{Z(\cdot)\}, \{\bar{Z}(\cdot)\}} \delta \\ \text{subject to:} \\ \text{(A.6), (A.7)} \\ \exists \{Z(j) \geq 0_{a \times q}\}, \{\bar{Z}(j) \geq 0_{a \times r}\} : \\ \text{s satisfies (5.17) and (5.18)} \end{cases} \quad (\text{A.8})$$

## A.2 VH-ROA Plots for Different Policies

Presented below are plots for the full VH-ROA corresponding to the CT policies compared in Section 5.5.

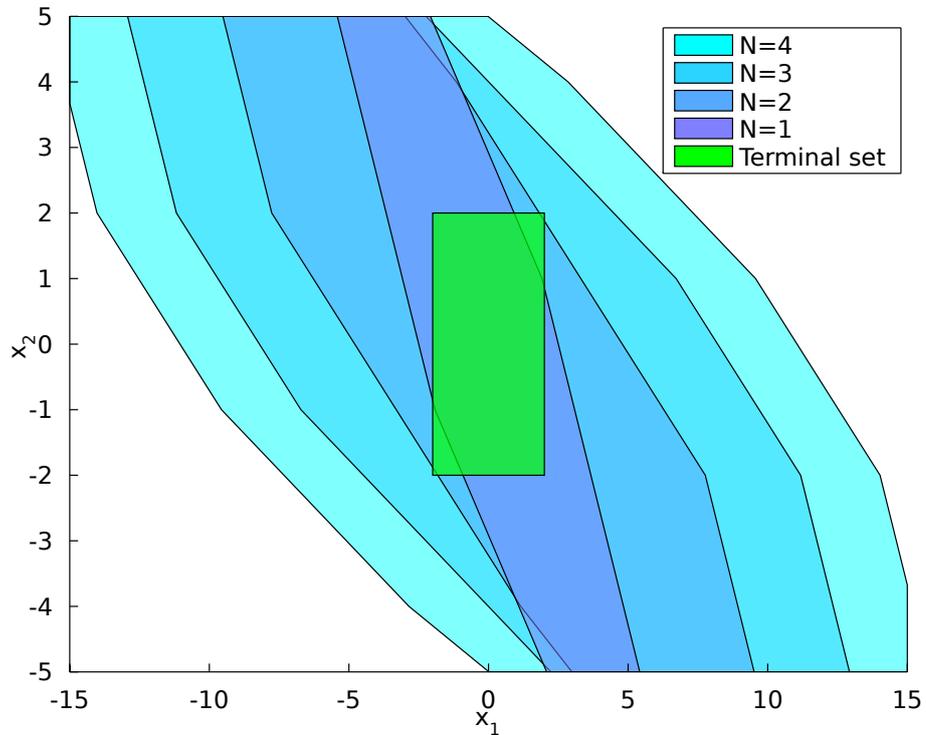


Figure A.1: Nominal VH-ROA for  $\mathcal{T}_1$

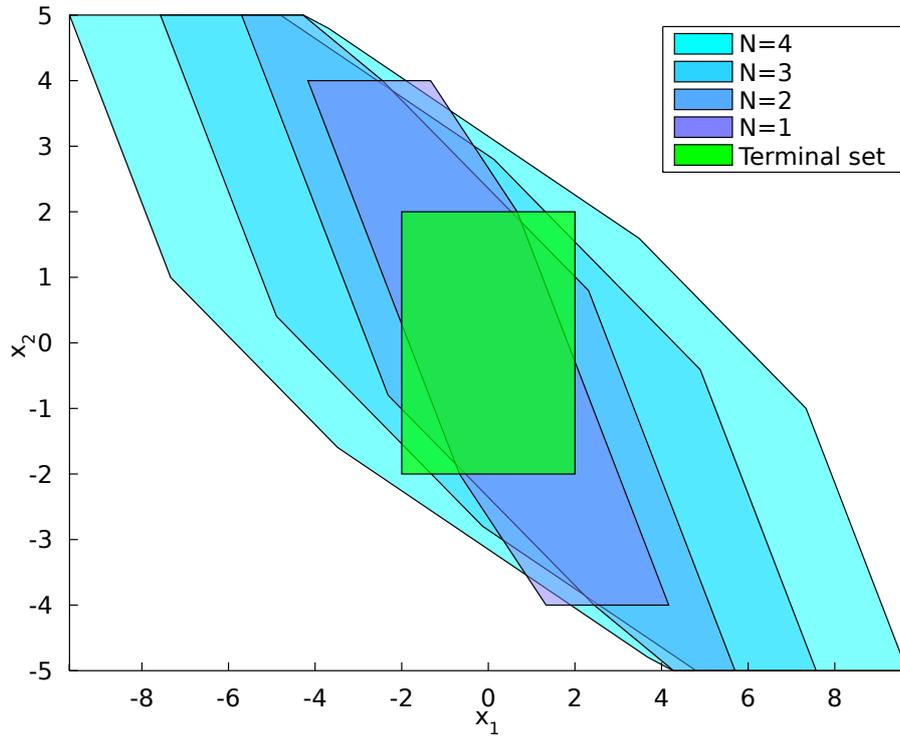


Figure A.2: VH-ROA for  $\mathcal{T}_1$  using max-vol

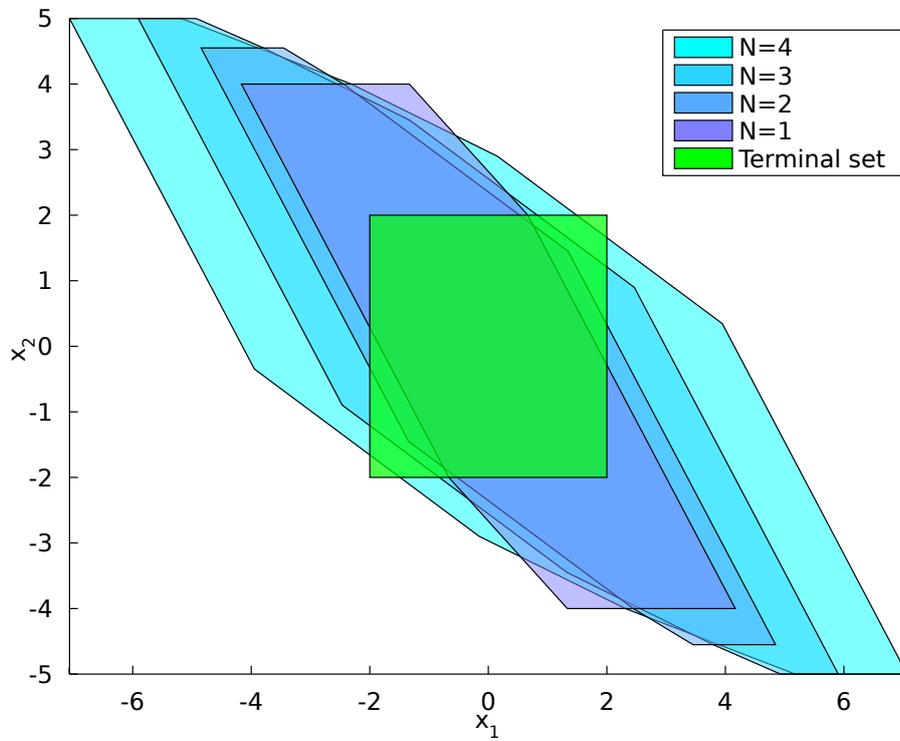


Figure A.3: VH-ROA for  $\mathcal{T}_1$  using max-dist

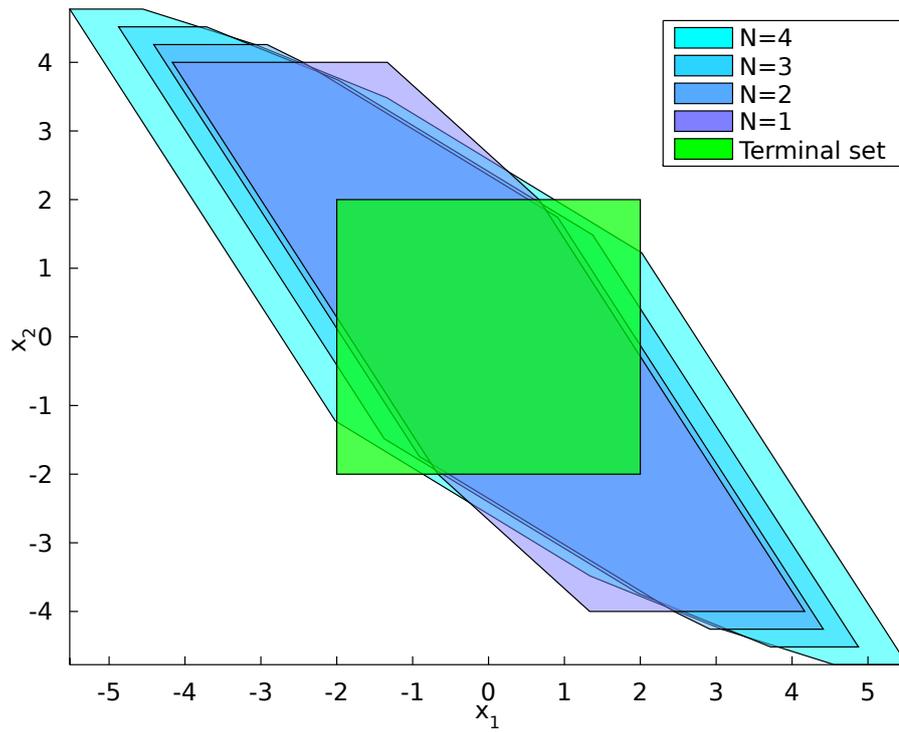


Figure A.4: VH-ROA for  $\mathcal{T}_1$  using 3-step nil

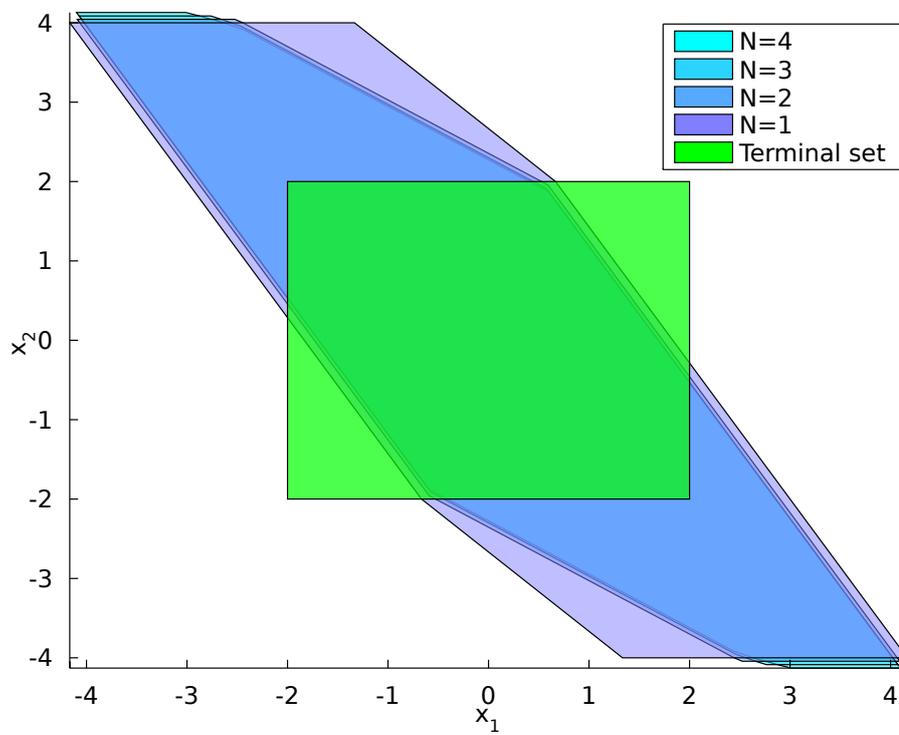


Figure A.5: VH-ROA for  $\mathcal{T}_1$  using 2-step nil

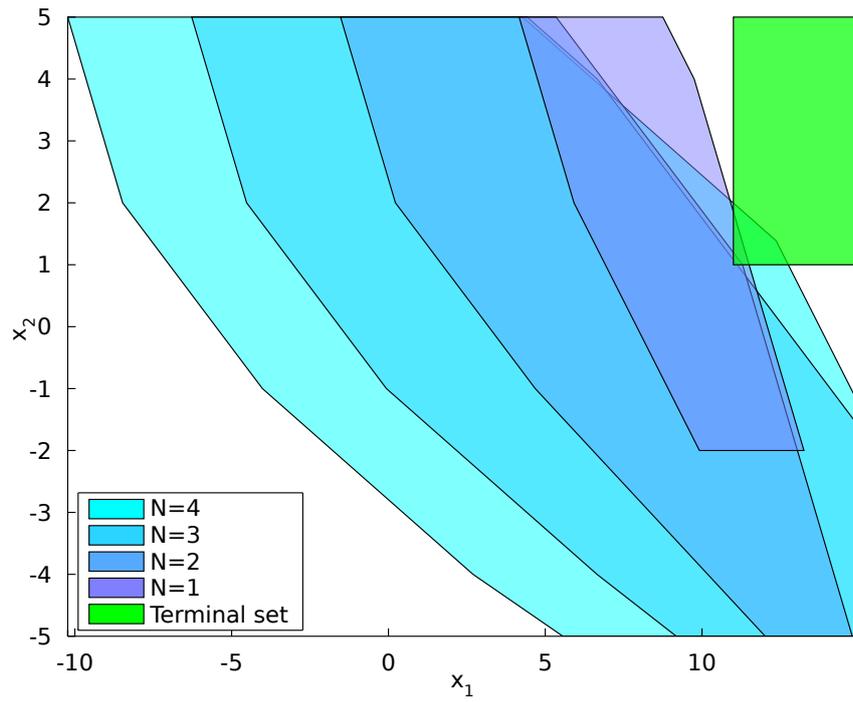


Figure A.6: Nominal VH-ROA for  $\mathcal{T}_2$

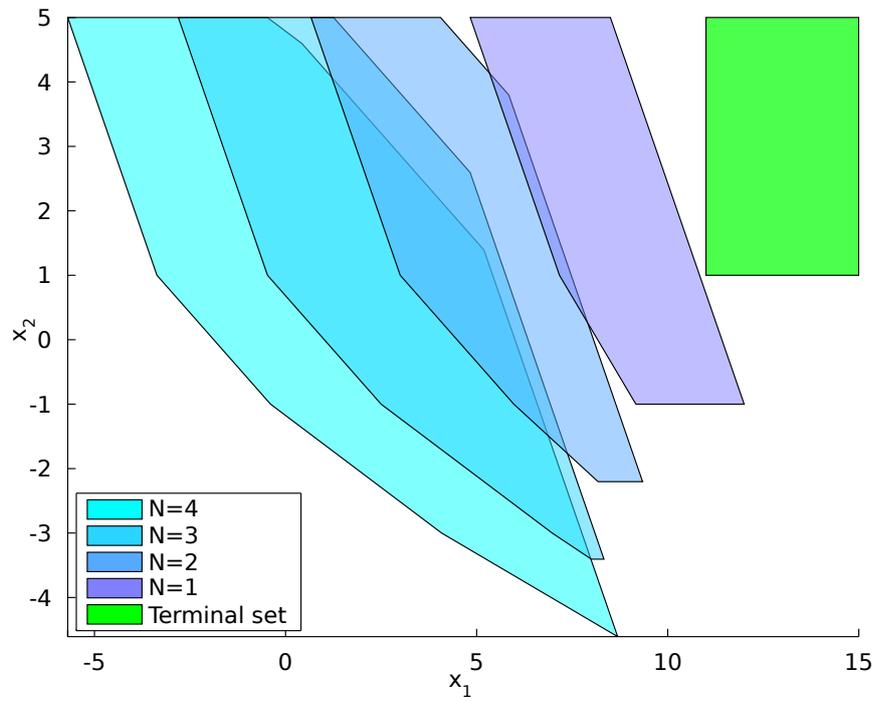


Figure A.7: VH-ROA for  $\mathcal{T}_2$  using max-vol

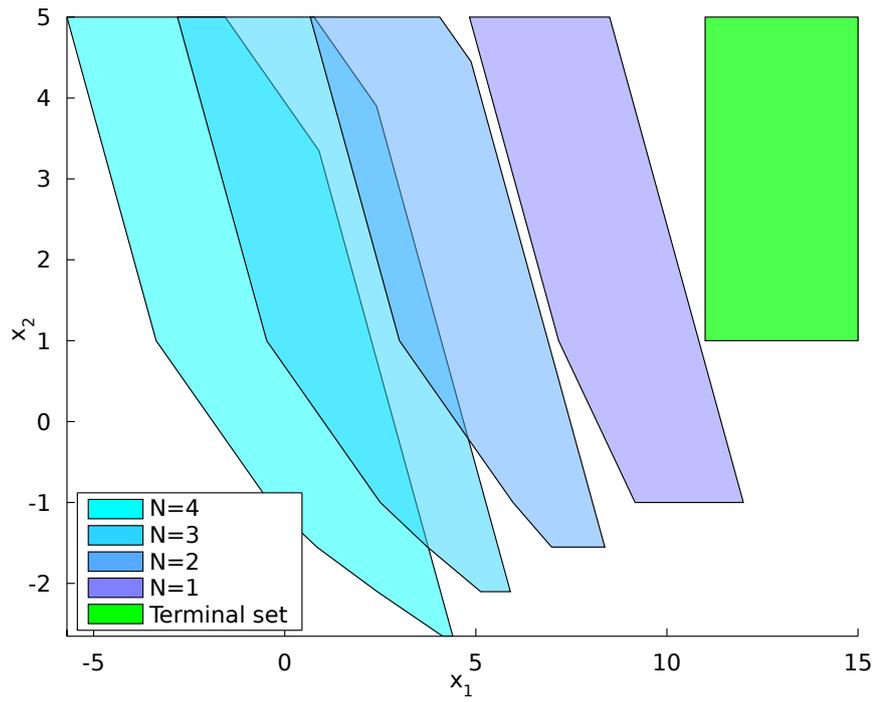


Figure A.8: VH-ROA for  $\mathcal{T}_2$  using max-dist

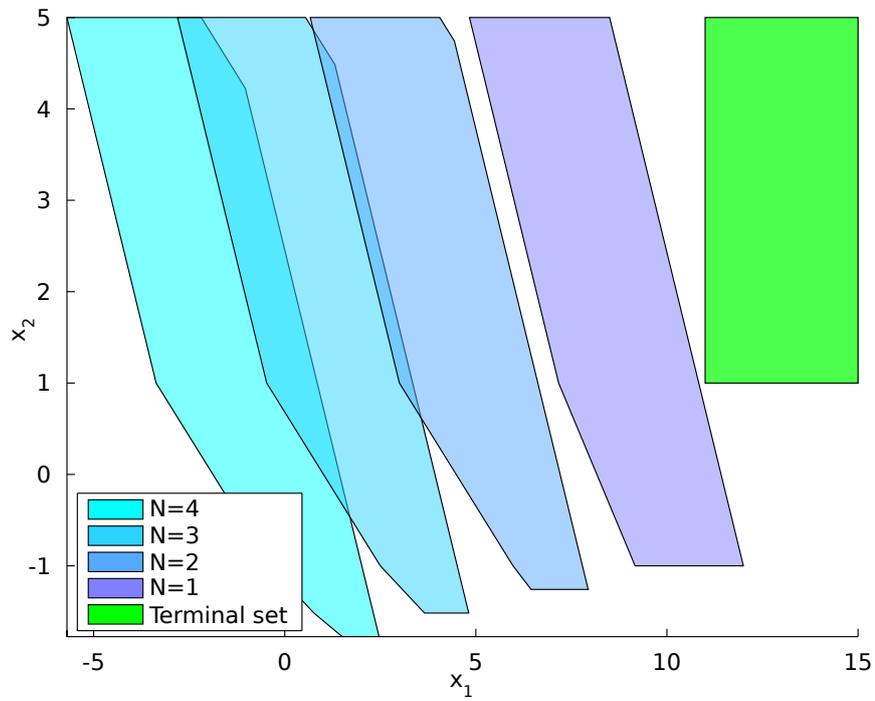


Figure A.9: VH-ROA for  $\mathcal{T}_2$  using 3-step-nil

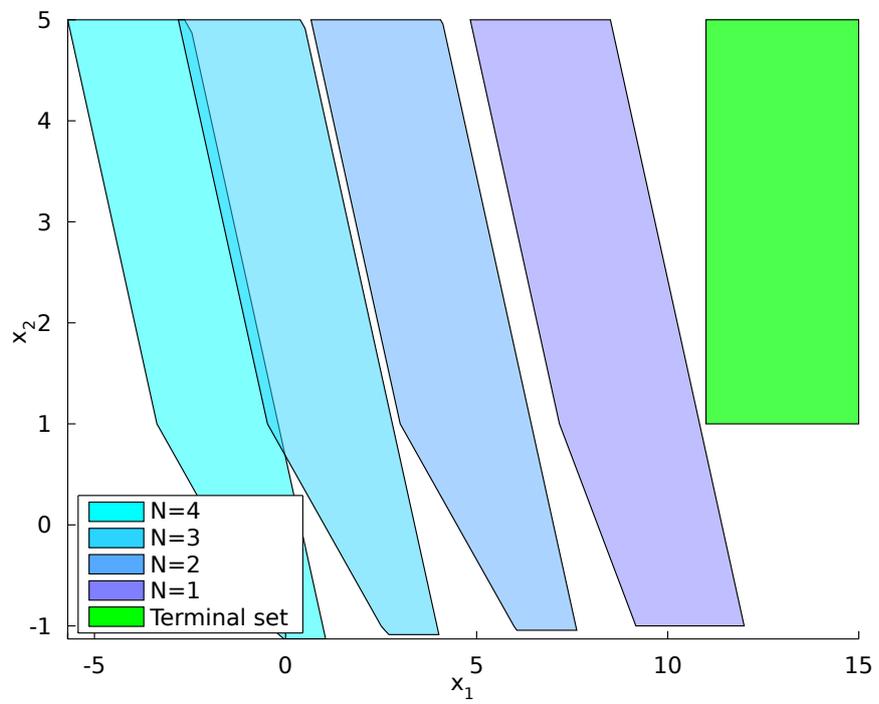


Figure A.10: VH-ROA for  $\mathcal{T}_2$  using 2-step-nil



## Appendix B

# Shovel Model & Controller Details

### B.1 Equations of Motion

This section contains the symbolic equations of motion in as derived in Subsection 6.2.2. Recall that they can be expressed in the compact form

$$D(q, m_p)\ddot{q} + C(q, \dot{q}, m_p, \dot{m}_p) + G(q, m_p) + \Lambda\dot{q} = \Phi. \quad (\text{B.1})$$

This equation can be written explicitly in terms of components as

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \ddot{r} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} + \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix} \begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} F_r \\ \tau_\theta \end{bmatrix}. \quad (\text{B.2})$$

In terms of the kinematic and dynamic parameters used in Section 6.2, the components of the dynamic matrices are given by

$$D_{11} = M + m_p \quad (\text{B.3})$$

$$D_{12} = \left( -r_5 \sin({}^1\theta_5) - {}^1y_{kcm} \right) m_p - M {}^1y'_{cm} \quad (\text{B.4})$$

$$D_{21} = D_{12} \quad (\text{B.5})$$

$$D_{22} = \left( r^2 + {}^1y_{kcm}^2 + r_5^2 + 2r_5 \sin({}^1\theta_5) {}^1y_{kcm} + 2r {}^1x_{kcm} + {}^1x_{kcm}^2 + 2r_5 \cos({}^1\theta_5) r \right. \quad (\text{B.6})$$

$$\left. + 2r_5 \cos({}^1\theta_5) {}^1x_{kcm} \right) m_p + \left( {}^1y_{cm}^2 + r^2 + 2r({}^1x'_{cm} - L_c) + ({}^1x'_{cm} - L_c)^2 \right) M + J$$

$$C_1 = \left( \dot{r} - \dot{\theta} r_5 \sin({}^1\theta_5) - \dot{\theta} {}^1y_{kcm} \right) \dot{m}_p - \left( r_5 \cos({}^1\theta_5) \dot{\theta}^2 + r \dot{\theta}^2 + {}^1x_{kcm} \dot{\theta}^2 \right) m_p \quad (\text{B.7})$$

$$- \left( r \dot{\theta}^2 + ({}^1x'_{cm} - L_c) \dot{\theta}^2 \right) M$$

$$C_2 = \left( \dot{\theta} {}^1y_{kcm}^2 - \dot{r} {}^1y_{kcm} + \dot{\theta} r^2 + \dot{\theta} r_5^2 + 2\dot{\theta} r_5 \sin({}^1\theta_5) {}^1y_{kcm} - \dot{r} r_5 \sin({}^1\theta_5) + 2\dot{\theta} r {}^1x_{kcm} \right. \quad (\text{B.8})$$

$$\left. + 2\dot{\theta} r_5 \cos({}^1\theta_5) {}^1x_{kcm} + 2\dot{\theta} r_5 \cos({}^1\theta_5) r + \dot{\theta} {}^1x_{kcm}^2 \right) \dot{m}_p$$

$$+ \left( 2\dot{r} \dot{\theta} {}^1x_{kcm} + 2\dot{r} \dot{\theta} r_5 \cos({}^1\theta_5) + 2\dot{r} \dot{\theta} r \right) m_p + \left( 2\dot{r} \dot{\theta} ({}^1x'_{cm} - L_c) + 2\dot{r} \dot{\theta} r \right) M$$

$$G_1 = g \left( M \sin(\theta) + m_p \sin(\theta) \right) \quad (\text{B.9})$$

$$G_2 = g \left( \left( r_5 \cos(\theta + {}^1\theta_5) - \sin(\theta) {}^1y_{kcm} + \cos(\theta) {}^1x_{kcm} + \cos(\theta)r \right) m_p \right. \\ \left. + \left( -\sin(\theta) {}^1y'_{cm} + \cos(\theta)r + \cos(\theta)({}^1x'_{cm} - L_c) \right) M \right). \quad (\text{B.10})$$

The components of the frictional matrix  $\Lambda$  are given in (6.28).

## B.2 Controller Parameters

The pre-stabilising controller gain is given by

$$K = 1 \times 10^7 \begin{bmatrix} 0.5113 & 0.1593 & 0.7160 & -0.1016 & 0.0034 & -0.0004 \\ 0.0052 & 2.0962 & 0.0368 & 2.8854 & -0.0001 & 0.0556 \end{bmatrix}. \quad (\text{B.11})$$

The identified, downsampled linear system is given by the state space matrices

$$A = \begin{bmatrix} -3.5326 & 1.6879 & -5.5772 & 6.9162 & 0.0054 & -0.0186 & 0.0271 \\ -0.1732 & 0.5527 & -0.2599 & 1.0306 & 0.0004 & -0.0007 & 0.0019 \\ 6.5103 & -5.5942 & 10.1230 & -16.0435 & -0.0103 & 0.0293 & -0.0586 \\ -1.8555 & 0.1872 & -2.6759 & 3.9571 & 0.0030 & -0.0079 & 0.0153 \\ 381.4447 & -1942.8321 & 580.8633 & -2955.5991 & -0.5684 & 1.9160 & -10.7236 \\ -1170.9576 & 1168.1313 & -1822.8751 & 3206.4945 & 1.9817 & -4.6376 & 11.3804 \\ 488.6781 & 138.6933 & 700.0382 & -773.7160 & -0.7584 & 2.0613 & -2.8468 \end{bmatrix} \quad (\text{B.12})$$

$$B = \begin{bmatrix} 0.1562 & 0.0208 \\ 0.0007 & 0.5822 \\ 0.3149 & 0.0172 \\ -0.0137 & 1.0105 \\ 45.1662 & 450.6227 \\ -141.9427 & -76.3676 \\ -4.3556 & -438.5990 \end{bmatrix} \quad (\text{B.13})$$

$$C = \begin{bmatrix} & & & I_7 & & & \\ \left[ \begin{array}{cccccc} -7.8467 & 15.5957 & -12.5835 & 27.0824 & 0.0211 & -0.0371 & 0.1016 \\ -10.7536 & 20.2900 & -14.8304 & 23.6959 & 0.0145 & -0.0428 & 0.0864 \end{array} \right] & & & & & & \end{bmatrix} \quad (\text{B.14})$$

$$D = 0_{9 \times 2} \quad (\text{B.15})$$