



LUND UNIVERSITY

Gradient-Based Distributed Model Predictive Control

Giselsson, Pontus

2012

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Giselsson, P. (2012). *Gradient-Based Distributed Model Predictive Control*. [Doctoral Thesis (compilation), Department of Automatic Control]. Department of Automatic Control, Lund Institute of Technology, Lund University.

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Gradient-Based Distributed Model Predictive Control

Gradient-Based Distributed Model Predictive Control

Pontus Giselsson

Department of Automatic Control
Lund University
Lund, November 2012

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

ISSN 0280-5316
ISRN LUTFD2/TFRT--1094--SE

© 2012 by Pontus Giselsson. All rights reserved.
Printed in Sweden by Media-Tryck.
Lund 2012

To Michaela, Noomi, and Siri

Abstract

The thesis covers different topics related to model predictive control (MPC) and particularly distributed model predictive control (DMPC). One topic of the thesis is gradient-based optimization algorithms for solving the optimization problem arising in DMPC in a distributed manner. The underlying idea is to solve the optimization problem in distributed fashion using dual decomposition, which is a well-known method. Dual decomposition is traditionally used in conjunction with (sub)gradient methods which are known to have bad convergence rate properties, especially for ill-conditioned problem. In this thesis it is shown how to use accelerated gradient methods with dual decomposition, and how to choose the step size parameter optimally in the algorithm. A method to bound the number of iterations needed to guarantee a prespecified accuracy of the solution is also provided. Based on the iteration bound, it is shown how to precondition the problem data optimally to improve conditioning of the problem. These contributions significantly improve the performance of the distributed optimization algorithm compared to dual decomposition with a (sub)gradient method.

Another topic of the thesis is to guarantee feasibility and stability when using the developed distributed optimization algorithm in a DMPC context. Traditional methods of proving stability in MPC usually involve terminal cost functions and terminal constraints that are non-separable. These methods are not directly applicable in DMPC based on dual decomposition because of the non-separable terms. Further, dual decomposition does not provide feasible iterations but is guaranteed to be primal feasible only in the limit. These issues have been addressed in the thesis. The stability issue is addressed by showing that for problems without a terminal cost or terminal constraints and if a certain controllability assumption on the stage costs is satisfied, the optimal value function is decreasing in every time step by a prespecified amount. It is also shown how the controllability assumption can be verified by solving a mixed integer linear

Abstract

program. The feasibility issue is addressed by a novel adaptive constraint tightening approach. The adaptive constraint tightening guarantees that a primal feasible solution can be constructed with finite number of algorithm iterations without compromising the stability guarantee.

The developed distributed optimization algorithm is evaluated on a hydro power valley benchmark problem. The hydro power valley consists of several dams connected in series where each dam is equipped with a turbine to extract power from the water. The objective is to control the water flow between the dams such that the total power from the turbines matches a power reference while respecting constraints on water levels and water flows. The control problem is formulated as an optimization problem, which is solved in receding horizon fashion using the distributed optimization algorithm presented in the thesis. The performance of the proposed distributed controller is compared to the performance of a centralized controller.

Acknowledgments

First I would like to thank my adviser Anders Rantzer. I am very fortunate to have been given the opportunity to work with him. He has always given me the freedom to work on problems of my choosing and has been very perceptive to my desires of new research directions.

I would also like to thank Karl Mårtensson with whom I have shared office since my first day at the department. He always found time to discuss problems I encountered and has helped me several times on different mathematical issues. It has been a true privilege to share office with him.

My gratitude also goes to Minh Dang Doan with whom I had a great collaboration during his three-month stay at our department. I would also like to thank his advisers Tamás Keviczky and Bart De Schutter for their feedback on our joint work. Further, I would like to thank Anna Lindholm for our collaboration on plant-wide disturbance rejection.

Another person that deserves my gratitude is Karl Johan Åström who has been very supportive and encouraging and has at numerous occasions entered my office and asked how my work is progressing. This has been really appreciated.

I would also like to thank Rolf Braun, Johan Åkesson and Anders Robertsson for, in different ways, giving me the opportunity to work with the pendulum system. It was very rewarding to set the theoretical work aside to do some practical implementation.

Bo Bernhardsson and Andrey Ghulchak also deserve my gratitude for teaching some of the courses I have taken. I like the way you provide intuition for the studied material.

My gratitude also goes to the technical and administrative staff. The number of minutes my work has been stopped due to technical or administrative matters during my stay at the department is very low. It is a pleasure to work in such a hassle-free environment.

Finally, I would like to thank my wife Michaela and our wonderful children Noomi and Siri. You bring me so much joy and happiness every

Acknowledgments

day and I am truly fortunate to have you. Michaela, I greatly appreciate your positive attitude, your endless love for our family, as well as your total uninterest in technical matters. It gives me a lot of perspective to what I am doing and helps me think clearly about what is important in life.

Pontus

Financial Support

The Swedish Research Council through the Linnaeus Center, LCCC, is gratefully acknowledged for financial support.

Contents

Preface	15
Contributions of the Thesis	15
1. Background	21
1.1 Convex Optimization	21
1.2 Lagrange Duality Theory	24
1.3 Gradient-Based Optimization	31
1.4 Distributed Optimization Methods	35
1.5 Model Predictive Control	38
References	46
Paper I. Accelerated Gradient Methods and Dual Decomposition in Distributed Model Predictive Control	51
1. Introduction	53
2. Problem Setup	54
3. Dual Problem	56
4. Distributed Optimization Algorithm	58
5. Numerical Example	61
6. Conclusions	63
7. Acknowledgments	63
References	64
Paper II. Execution Time Certification for Gradient-Based Optimization in Model Predictive Control	67
1. Introduction	69
2. Problem Setup and Preliminaries	70
3. Lagrange Multiplier Norm Bounds	75
4. Algorithm Iteration Bounds	77
5. Preconditioning	80
6. Numerical Example	81
7. Conclusions and Future Work	83
References	84

Paper III. Optimal Preconditioning and Iteration Complexity Bounds for Gradient-Based Optimization in MPC . .	87
1. Introduction	89
2. Problem Setup and Preliminaries	90
3. Lagrange Multiplier Norm Bounds	94
4. Algorithm Iteration Bounds	96
5. Preconditioning	99
6. Numerical Example	103
7. Conclusions and Future Work	104
References	106
Paper IV. A Generalized Distributed Accelerated Gradient Method for DMPC with Iteration Complexity Bounds .	109
1. Introduction	111
2. Problem Setup	112
3. Distributed Algorithm	115
4. Lagrange Multiplier Norm Bounds	120
5. Iteration Bounds	125
6. Numerical Example	126
7. Conclusions and Future Work	127
References	128
Paper V. On Feasibility, Stability and Performance in Distributed Model Predictive Control	131
1. Introduction	133
2. Problem Setup and Preliminaries	136
3. Stopping Condition	140
4. Offline Controllability Verification	147
5. Numerical Example	152
6. Conclusions and Future Work	154
7. Acknowledgments	154
References	155
A. Appendix	159
Paper VI. Output Feedback Distributed Model Predictive Control with Inherent Robustness Properties	167
1. Introduction	169
2. Setup and Preliminaries	170
3. Robust State Feedback DMPC	175
4. Output Feedback DMPC	177
5. Numerical Example	181
6. Conclusions	183
References	184
Paper VII. A Distributed Accelerated Gradient Algorithm for DMPC of a Hydro Power Valley	187

1.	Introduction	189
2.	Problem Description	191
3.	Distributed Optimization Framework for MPC	195
4.	Control of HPV using Distributed MPC	198
5.	Simulation Results	206
6.	Conclusions and Future Work	210
7.	Acknowledgments	211
	References	212
A.	Appendix	216
Paper VIII. Gradient-Based Model Predictive Control in a		
	Pendulum System	219
1.	Introduction	221
2.	Problem Setup	222
3.	The Pendulum System	223
4.	Optimal Feedforward Trajectories	225
5.	Model Predictive Control	228
6.	Conclusions	238
	References	239
Supplement A. Specification of Randomly Generated Sys-		
	tems	241
A.1.	Random System in Paper III, Paper IV, and Paper V . . .	241
A.2.	Random System in Paper VI	243

Preface

Contributions of the Thesis

The thesis consists of one introductory chapter, eight papers and one supplement. This section describes the content of the introductory chapter, the contribution of each paper, and the content in the supplement.

Chapter 1 – Background

The background chapter consists of material relevant for the thesis. We cover convex optimization, Lagrange duality theory, gradient-based optimization methods, decomposition techniques, and model predictive control.

Paper I

Giselsson, P., M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer (2012) “Accelerated gradient methods and dual decomposition in distributed model predictive control.” To appear in *Automatica*.

In this paper it is shown how accelerated gradient methods can be used in conjunction with dual decomposition in a distributed model predictive control context. The optimal constant step size for the algorithm is provided. It is also shown how to, in a distributed manner, handle an additional non-smooth and non-separable 1-norm term in the objective. Another contribution is, besides a convergence rate for the dual function value, also a convergence rate for the primal variables.

The work was to a large extent carried out by P. Giselsson with help from M. D. Doan. Useful support and comments were given by T. Keviczky, B. De Schutter and A. Rantzer.

Paper II

Giselsson, P. (2012) “Execution time certification for gradient-based optimization in model predictive control.” To appear in *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, HI.

The paper considers execution time certification for a dual accelerated gradient method applied in a model predictive control context. A centralized model predictive control formulation with eliminated states is considered. To compute a bound on the number of iterations needed to achieve a prespecified accuracy of the dual function value, a bound on the norm of the optimal dual variables associated with the inequality constraints is needed. We show that by constructing a Slater vector for every feasible initial condition, the norm of the optimal dual variables can be bounded. This implies that the iteration bound for dual function accuracy can be computed. We also show how to precondition the inequality constraint matrices optimally, where optimally refers to the preconditioning that minimizes the obtained iteration bound.

Paper III

Giselsson, P. (2012) “Optimal preconditioning and iteration complexity bounds for gradient-based optimization in MPC.” Submitted to *2013 American Control Conference*, Washington, D.C.

This paper extends the result in Paper II to the case where the state variables are not eliminated. The resulting MPC optimization problem has both equality and inequality constraints and the presented results are hence applicable to distributed MPC as well as centralized MPC. To compute an iteration bound for the dual function accuracy, the norm of the optimal dual variables corresponding to the equality constraints and the inequality constraints needs to be bounded. We show how to compute a bound to the norm of the optimal dual variables, which is used to compute the iteration bound. We also show how to precondition the equality constraint and inequality constraint matrices optimally by solving a semidefinite program, where optimally refers to the preconditioning that minimizes the iteration bound.

Paper IV

Giselsson, P. (2012) “A generalized distributed accelerated gradient method for DMPC with iteration complexity bounds.” Submitted to *2013 American Control Conference*, Washington, D.C.

Gradient-based methods are known to have iterations of low complexity but might need a significant number of iterations to converge. In this

paper a distributed dual accelerated gradient method is proposed which significantly reduces the number of iterations needed to achieve a satisfactory accuracy of the dual function. This is done by, in a well-defined manner, incorporating hessian information to the algorithm. By approximating the hessian by a block-diagonal matrix, the algorithm can still be implemented in a distributed fashion. By offline computing the hessian approximation, the iteration complexity is greatly reduced at run-time for the DMPC controller. The paper also shows how to compute a bound on the number of iterations necessary to guarantee a prespecified dual accuracy.

Paper V

Giselsson, P. and A. Rantzer (2012) “On feasibility, stability and performance in distributed model predictive control.” Submitted to *IEEE Transactions on Automatic Control*.

This publication concerns closed loop properties for distributed model predictive control when the optimization problem is solved using dual decomposition methods. The traditional way of proving stability in centralized model predictive control is not directly applicable to distributed model predictive control. In this paper a method is presented that proves stability for distributed model predictive control where neither terminal constraints nor a terminal cost is used. The stability result is based on a controllability assumption of the stage costs. We show that this controllability assumption can be verified by solving a mixed integer linear program. The paper also proposes a novel adaptive constraint tightening approach that enables for early termination of the optimization algorithm while still guaranteeing closed loop properties such as feasibility, stability and a prespecified performance.

The basic ideas with dual decomposition and suboptimality bounds in distributed model predictive control are due to A. Rantzer. All details such as the method to verify the controllability assumption, the adaptive constraint tightening approach, and the possibility for early termination are due to P. Giselsson.

Paper VI

Giselsson, P. (2012) “Output feedback distributed model predictive control with inherent robustness properties.” Submitted to *2013 American Control Conference*, Washington, D.C.

This paper extends the results in Paper V to include the possibility of output feedback. A decentralized observer is created and the controller from Paper V is fed with state estimates from the observer. Stability results are provided by showing that the estimation error can be treated as

a bounded disturbance and that the controller from Paper V is inherently robust to small disturbances.

Paper VII

Doan, M. D., P. Giselsson, T. Keviczky, B. De Schutter, and A. Rantzer (2012) “A distributed accelerated gradient algorithm for DMPC of a hydro power valley.” Submitted to *Control Engineering Practice*.

In this paper distributed control of a hydro power valley (HPV) is considered. The objective of the control is to meet a time-varying power profile while respecting water flow and water level constraints. The model of the HPV has nonlinearities in the power production functions and boolean constraints. Further, the power production depends on states in all sub-systems, which complicates a distributed implementation. These issues are addressed and the distributed optimization algorithm from Paper I is used to solve the resulting optimization problem in a distributed fashion. The hydro power valley problem is a benchmark problem in the European Union FP7 STREP project HD-MPC.

Model reduction was performed by M. D. Doan and controller design, controller tuning, and simulations were performed by P. Giselsson. T. Keviczky, B. De Schutter, and A. Rantzer gave useful comments and A. Rantzer also suggested some useful ideas for the controller design.

Paper VIII

Giselsson, P. (2012) “Gradient-based model predictive control in a pendulum system.” Technical Report ISRN LUTFD2/TFRT--7624--SE. Department of Automatic Control, LTH, Lund University, Sweden.

In this paper optimal control of a pendulum system is considered. Optimal control trajectories are computed and used as feedforward control trajectories. A model predictive control (MPC) formulation is used as feedback to control the actual system trajectories towards the optimal trajectories. The MPC optimization problem is formulated as a quadratic program. A dual formulation to the MPC optimization problem is stated and an accelerated gradient method is applied to solve the dual problem. Experiments show that the optimization algorithm is efficient enough to be implemented in the pendulum application in real-time and that the MPC feedback gives good closed loop performance.

Related publications. The technical report is based on and extends the conference paper

Giselsson, P. (2011): “Model predictive control in a pendulum system.” In *Proceedings of the 31st IASTED Conference on Modelling, Identification and Control*. Innsbruck, Austria.

and is to some extent also based on the conference paper

Giselsson, P., J. Åkesson, and A. Robertsson (2009): “Optimization of a pendulum system using Optimica and Modelica.” In *Proceedings of the 7th International Modelica Conference 2009*, pp. 480–489. Como, Italy.

Supplement A – Specification of Randomly Generated Systems

The supplement specifies dynamics matrices, constraints, and cost functions in the randomly generated systems used in Paper III, Paper IV, Paper V, and Paper VI.

Additional Publications

The following publications were chosen not to be included in the thesis.

Torreblanca, P. M., P. Giselsson, and A. Rantzer (2010): “Distributed receding horizon Kalman filter.” In *Proceedings of the 49th IEEE Conference on Decision and Control*, pp. 5068–5073. Atlanta, GA.

Giselsson, P. and A. Rantzer (2010): “Distributed model predictive control with suboptimality and stability guarantees.” In *Proceedings of the 49th IEEE Conference on Decision and Control*, pp. 7272–7277. Atlanta, GA.

Giselsson, P. (2010): “Adaptive nonlinear model predictive control with suboptimality and stability guarantees.” In *Proceedings of the 49th IEEE Conference on Decision and Control*, pp. 3644–3649. Atlanta, GA.

Lindholm, A. and P. Giselsson (2012): “Formulating an optimization problem for minimization of losses due to utilities.” In *8th IFAC International Symposium on Advanced Control of Chemical Processes*. Singapore.

1

Background

In this chapter, background material relevant for the thesis is presented.

1.1 Convex Optimization

In this section we will describe convex optimization and introduce useful definitions and results.

Convex Sets

A set C is *convex* if for any points $x_1, x_2 \in C$ and any θ where $0 \leq \theta \leq 1$ we have

$$\theta x_1 + (1 - \theta)x_2 \in C.$$

The definition implies that between any two points in the set, there is a straight line that lies within the set.

Convex Functions

A convex function is a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ where for all $x, y \in \mathbb{R}^n$ and θ with $0 \leq \theta \leq 1$ we have

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y). \quad (1.1)$$

In this definition and hereafter we assume that $\text{dom} f = \mathbb{R}^n$ for convenience. The convexity definition implies that the line segment between any points $(x, f(x))$ and $(y, f(y))$ lies above (or on) the graph of f . A *subgradient* to a convex function f at x is any vector $\xi(x)$ such that

$$f(y) \geq f(x) + \xi(x)^T(y - x) \quad (1.2)$$

for all $y \in \mathbb{R}^n$. The set of vectors $\xi(x)$ that satisfy (1.2) at x is denoted by $\partial f(x)$ and is called the *subdifferential* of f at x . If the subdifferential for

Chapter 1. Background

every $x \in \mathbb{R}^n$ is a singleton, then f is differentiable. In that case $\partial f(x) = \{\nabla f(x)\}$ where $\nabla f(x)$ is called the *gradient* to f at x . For differentiable functions convexity holds if and only if for all $x, y \in \mathbb{R}^n$ the following holds

$$f(y) \geq f(x) + \nabla f(x)^T (y - x).$$

For differentiable functions we also define *strict convexity*. The function f is strictly convex if for any $x, y \in \mathbb{R}^n$ with $x \neq y$ we have

$$f(y) > f(x) + \nabla f(x)^T (y - x).$$

For differentiable functions, a *strongly convex* function f is a function that satisfies

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\sigma}{2} \|x - y\|^2$$

for every $x, y \in \mathbb{R}^n$ where $\sigma > 0$ is called the *convexity parameter*. The definitions imply that a strongly convex function is also strictly convex, but not vice versa. It is also possible to define strict and strong convexity for non-differentiable functions in accordance with (1.1). A differentiable convex function with Lipschitz continuous gradient satisfies

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$$

for all $x, y \in \mathbb{R}^n$ where $L > 0$ is the Lipschitz constant. This is equivalent to (cf. [Nesterov, 2003, Theorem 2.1.5])

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} \|x - y\|^2 \quad (1.3)$$

for all $x, y \in \mathbb{R}^n$, i.e., that a quadratic function with curvature L in all directions is an upper bound to f . Finally, a *concave* function g is a function $g : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $-g$ is convex.

Convex Optimization Problems

We consider the following general optimization problem

$$\begin{aligned} & \text{minimize } f_0(x) && (1.4) \\ & \text{subject to } f_i(x) \leq 0, && i = 1, \dots, q \\ & && h_i(x) = 0, \quad i = 1, \dots, r \end{aligned}$$

where $x \in \mathbb{R}^n$. We refer to $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ as the *objective function* or *cost function*. We call $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, q$, the *inequality constraint functions* and $f_i(x) \leq 0$, $i = 1, \dots, q$, are referred to as *inequality constraints*. Further, we call $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, r$, the *equality constraint*

functions and $h_i(x) = 0$, $i = 1, \dots, r$, the equality constraints. We use the shorthand vector notation

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_q(x) \end{bmatrix}, \quad h(x) = \begin{bmatrix} h_1(x) \\ \vdots \\ h_r(x) \end{bmatrix}.$$

The optimization problem (1.4) is called *feasible* if there exists an x such that $f(x) \leq 0$ and $h(x) = 0$, the problem is *strictly feasible* if there exists an x such that $f(x) < 0$ and $h(x) = 0$, and the problem is *infeasible* if it is not feasible. The optimal value of (1.4) is denoted by p^* . We use the convention of letting $p^* = \infty$ for infeasible problems and $p^* = -\infty$ for problems that are unbounded below.

We define convex optimization problems to be of the form

$$\begin{aligned} & \text{minimize } f_0(x) \\ & \text{subject to } f_i(x) \leq 0, \quad i = 1, \dots, q \\ & \quad \quad \quad a_i^T x = b_i, \quad i = 1, \dots, r \end{aligned} \tag{1.5}$$

where f_i , $i = 0, \dots, q$ are convex functions, $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$, $i = 1, \dots, r$. The differences between the general optimization problem (1.4) and the convex optimization problem (1.5) are that in (1.5) the objective function and inequality constraint functions are restricted to be convex, and the equality constraint functions in (1.5) are restricted to be affine. For future reference we introduce the matrix $A = [a_1, \dots, a_r]^T$ and the vector $b = [b_1, \dots, b_r]^T$ which implies that the equality constraints can be expressed as $Ax = b$.

The feasible set of (1.5) is a convex set since it is the intersection of level sets to convex functions (the inequality constraints) and hyperplanes (the equality constraints). This implies that in convex optimization, a convex cost is minimized over a convex set.

Existence of Optimal Solutions

Even if the optimal value p^* to (1.5) is finite, it is possible that no x exist that attains the minimum, i.e., no x exist such that $f_0(x) = p^*$, $f(x) \leq 0$ and $Ax = b$. An example is the unconstrained problem with $f_0(x) = e^x$ for which $p^* = 0$ but no x exist such that $e^x = 0$. Weierstrass extreme value theorem gives conditions for which a minimization problem is guaranteed to attain its minimum value. To state these conditions we define the feasible set

$$Z = \{x \in \mathbb{R}^n \mid f(x) \leq 0 \text{ and } Ax = b\}.$$

We state one version of Weierstrass extreme value theorem below.

Chapter 1. Background

PROPOSITION 1.1

Let $Z \subseteq \mathbb{R}^n$ be nonempty and let $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ be continuous. Assume that at least one of the following conditions hold

1. Z is compact.
2. Z is closed and f_0 is strongly convex.

Then a vector x exists such that $f_0(x) = \inf_{z \in Z} f_0(z)$. □

This is a slight modification of [Bertsekas, 1999, Proposition A.8]. We have exchanged the coercive property in [Bertsekas, 1999, Proposition A.8 (2)] to strongly convex since every strongly convex function is also coercive.

First-order Optimality Condition

Below we state the first-order optimality condition for differentiable cost functions f_0 (cf. [Boyd and Vandenberghe, 2004, §4.2.3]). We have that x^* is optimal if and only if $x^* \in Z$ and

$$\nabla f_0(x^*)^T (x - x^*) \geq 0 \quad \text{for all } x \in Z.$$

From the definition of strictly convex functions and the first-order optimality condition we get

$$f_0(x) - f_0(x^*) > \nabla f_0(x^*)^T (x - x^*) \geq 0$$

for all $x \in Z \setminus x^*$. This implies that the minimizing argument x^* to a convex optimization problem with a strictly convex objective function is unique, if it is attained.

1.2 Lagrange Duality Theory

In this section we describe Lagrange duality theory. The presentation in this section is influenced by [Boyd and Vandenberghe, 2004, Chapter 5]. For a more complete treatment of convex optimization and Lagrange duality theory the reader is referred to [Rockafellar, 1970, Bertsekas, 1999, Boyd and Vandenberghe, 2004, Hiriart-Urruty and Lemarechal, 1996].

We consider optimization problems as stated in (1.4) and refer to (1.4) as the *primal problem*. In Lagrange duality, the constraints are taken into account by augmenting the cost function by a weighted sum of the constraint functions. We introduce dual variables $\lambda \in \mathbb{R}^r$ and $\mu \in \mathbb{R}_{\geq 0}^q$ and form the Lagrangian associated with (1.4) as

$$\mathcal{L}(x, \lambda, \mu) = f_0(x) + \sum_{i=1}^r \lambda_i h_i(x) + \sum_{i=1}^q \mu_i f_i(x)$$

where λ_i is associated with equality constraint h_i and μ_i is associated with inequality constraint f_i . The Lagrange dual function is defined as

$$g(\lambda, \mu) = \inf_x \mathcal{L}(x, \lambda, \mu) = \inf_x \left(f_0(x) + \sum_{i=1}^r \lambda_i h_i(x) + \sum_{i=1}^q \mu_i f_i(x) \right)$$

which is a concave function even if the primal problem (1.4) is not convex [Boyd and Vandenberghe, 2004, §5.1.2]. For any $\mu \in \mathbb{R}_{\geq 0}^q$, any $\lambda \in \mathbb{R}^r$, and any feasible $\tilde{x} \in \mathbb{R}^n$, i.e., that satisfies $h_i(\tilde{x}) = 0$ and $g_i(\tilde{x}) \leq 0$, we have that

$$\sum_{i=1}^r \lambda_i h_i(\tilde{x}) + \sum_{i=1}^q \mu_i f_i(\tilde{x}) \leq 0. \quad (1.6)$$

This implies that

$$g(\lambda, \mu) = \inf \mathcal{L}(x, \lambda, \mu) \leq \mathcal{L}(\tilde{x}, \lambda, \mu) \leq f_0(\tilde{x}) \quad (1.7)$$

where the last inequality follows from the definition of \mathcal{L} and from (1.6). Since the relation holds for any feasible \tilde{x} , it holds for the minimizing argument to (1.4) which implies that

$$g(\lambda, \mu) \leq p^*. \quad (1.8)$$

Dual Function Differentiability Properties

Before we discuss differentiability properties of the dual function, we introduce the set of points that minimize the Lagrangian

$$X(\lambda, \mu) = \{x \in \mathbb{R}^n \mid x = \arg \min_x \mathcal{L}(x, \lambda, \mu)\}.$$

Using this definition, a subgradient to the concave dual function is any vector $\xi = [\xi_\lambda^T \ \xi_\mu^T]^T$ where ξ_λ^T and ξ_μ^T are any vectors that satisfy

$$\xi_\lambda = \begin{bmatrix} a_1^T x_{\lambda\mu} - b_1 \\ \vdots \\ a_r^T x_{\lambda\mu} - b_r \end{bmatrix}, \quad \xi_\mu = \begin{bmatrix} f_1(x_{\lambda\mu}) \\ \vdots \\ f_q(x_{\lambda\mu}) \end{bmatrix}$$

for some $x_{\lambda\mu} \in X(\lambda, \mu)$ (cf. [Bertsekas, 1999, §6.1]). Under some conditions the dual function is differentiable. These conditions are stated in the following proposition.

Chapter 1. Background

PROPOSITION 1.2

Assume that $f_i, i = 1, \dots, q$, are continuous and convex, that $h(x) = Ax - b$ where $A \in \mathbb{R}^{r \times n}$ and $b \in \mathbb{R}^r$, that f_0 is continuous and strongly convex, and that (1.4) is feasible. Then $X(\lambda, \mu)$ consists of a unique point $x_{\lambda\mu}$ and the dual function $g(\lambda, \mu)$ is differentiable with gradient $\nabla g(\lambda, \mu) = [\nabla_{\lambda} g(\lambda, \mu)^T \nabla_{\mu} g(\lambda, \mu)^T]^T$ where

$$\nabla_{\lambda} g(\lambda, \mu) = Ax_{\lambda\mu} - b, \quad \nabla_{\mu} g(\lambda, \mu) = f(x_{\lambda\mu}).$$

□

The proof follows almost immediately from Danskin's Theorem [Bertsekas, 1999, Proposition B.25]. A slight modification of the proof is however needed. The proof requires the infimum to be performed over a compact set. This is circumvented by letting f_0 be strongly convex, $f_i, i = 1, \dots, q$ convex and h_i affine. From these assumptions, uniqueness of $x_{\lambda\mu}$ is concluded and the necessary compact set can be constructed where needed in the proof.

By further restricting the problem data, we obtain the following property of the dual function.

PROPOSITION 1.3

Assume that $h(x) = Ax - b$, where $A \in \mathbb{R}^{r \times n}$ and $b \in \mathbb{R}^r$ and that $f(x) = Cx - d$ where $C \in \mathbb{R}^{q \times n}$ and $d \in \mathbb{R}^q$. Further assume that f_0 is continuous and strongly convex with convexity parameter σ , and that (1.4) is feasible. Then the gradient of the dual function $\nabla g(\lambda, \mu)$ has Lipschitz continuous gradient with Lipschitz constant

$$L = \frac{\|[A^T \ C^T]^T\|^2}{\sigma}.$$

□

A proof to this proposition is provided in [Nesterov, 2005, Theorem 1].

The Lagrange Dual Problem

We know from (1.7) that the dual function gives a lower bound to the optimal value p^* of the optimization problem (1.4) if $\mu \in \mathbb{R}_{\geq 0}^q$ and $\lambda \in \mathbb{R}^r$. To obtain the tightest lower bound, we maximize the dual function

$$\begin{aligned} & \text{maximize } g(\lambda, \mu) \\ & \text{subject to } \mu \geq 0. \end{aligned}$$

This problem is referred to as the *Lagrange dual problem* or the *dual problem*. The optimal value of the dual problem is denoted by d^* . From

(1.8) we have that $g(\lambda, \mu) \leq p^*$ holds for any $\lambda \in \mathbb{R}^r$ and $\mu \in \mathbb{R}_{\geq 0}^q$ which implies that

$$d^* \leq p^*. \quad (1.9)$$

This relation is referred to as *weak duality* and the difference $p^* - d^* \geq 0$ is referred to as the *duality gap*. For some optimization problems there is no duality gap, i.e.,

$$d^* = p^*. \quad (1.10)$$

This property is referred to as *strong duality*.

Alternative Characterization of Duality

The dual problem can by definition be written as

$$d^* = \sup_{\lambda, \mu \geq 0} \inf_x \mathcal{L}(x, \lambda, \mu).$$

We also have

$$\begin{aligned} \sup_{\lambda, \mu \geq 0} \mathcal{L}(x, \lambda, \mu) &= \sup_{\lambda, \mu \geq 0} \left(f_0(x) + \sum_{i=1}^r \lambda_i h_i(x) + \sum_{i=1}^q \mu_i f_i(x) \right) \\ &= \begin{cases} f_0(x) & f_i(x) \leq 0, \quad i = 1, \dots, q, \quad h_i(x) = 0 \\ \infty & \text{else} \end{cases} \end{aligned}$$

This states that if x is feasible, then $\sup_{\lambda, \mu \geq 0} \mathcal{L}(x, \lambda, \mu) = f_0(x)$, otherwise $\sup_{\lambda, \mu \geq 0} \mathcal{L}(x, \lambda, \mu) = \infty$. This implies that p^* can be expressed as

$$p^* = \inf_x \sup_{\lambda, \mu \geq 0} \mathcal{L}(x, \lambda, \mu).$$

By weak duality we get that

$$\sup_{\lambda, \mu \geq 0} \inf_x \mathcal{L}(x, \lambda, \mu) \leq \inf_x \sup_{\lambda, \mu \geq 0} \mathcal{L}(x, \lambda, \mu)$$

and if strong duality holds we get

$$\sup_{\lambda, \mu \geq 0} \inf_x \mathcal{L}(x, \lambda, \mu) = \inf_x \sup_{\lambda, \mu \geq 0} \mathcal{L}(x, \lambda, \mu).$$

We conclude that strong duality implies that the order of the minimization over x and the maximization over λ, μ can be interchanged without affecting the result.

Complementary Slackness

We assume that (λ^*, μ^*) are dual optimal, that x^* is primal optimal, and that strong duality holds. Under these assumptions we have that

$$\begin{aligned}
 f_0(x^*) &= g(\lambda^*, \mu^*) = \inf_x \mathcal{L}(x, \lambda^*, \mu^*) & (1.11) \\
 &= \inf_x \left(f_0(x) + \sum_{i=1}^r \lambda_i^* h_i(x) + \sum_{i=1}^q \mu_i^* f_i(x) \right) \\
 &\leq f_0(x^*) + \sum_{i=1}^r \lambda_i^* h_i(x^*) + \sum_{i=1}^q \mu_i^* f_i(x^*) \\
 &\leq f_0(x^*)
 \end{aligned}$$

where the first equality is due to strong duality, the second equality follows from the definition of the dual function, and the third equality is due to the definition of \mathcal{L} . The first inequality is due to the definition of inf and the final inequality holds since x^* is primal feasible, i.e., $h_i(x^*) = 0$ and $f_i(x^*) \leq 0$, and since $\mu^* \geq 0$. Since the inequalities in (1.11) can be replaced by equalities, we conclude that

$$\sum_{i=1}^q \mu_i^* f_i(x^*) = 0.$$

Each term in the sum is non-positive which implies that

$$\mu_i^* f_i(x^*) = 0, \quad i = 1, \dots, q.$$

This condition, which is referred to as *complementary slackness*, holds for any primal optimal solution x^* and dual optimal solution (λ^*, μ^*) when strong duality holds.

Obtaining Primal Optimal Solution From Dual Problem

From (1.11) we conclude that x^* is a minimizer to $\mathcal{L}(x, \lambda^*, \mu^*)$. However, $\mathcal{L}(x, \lambda^*, \mu^*)$ can have also other minimizers. To extract the primal problem from the dual further conditions are required. Suppose that a dual optimal pair (λ^*, μ^*) is known and that the minimizing argument to

$$\inf_x \left(f_0(x) + \sum_{i=1}^r \lambda_i^* h_i(x) + \sum_{i=1}^q \mu_i^* f_i(x) \right) \quad (1.12)$$

is unique. One example for which this occurs is if f_0 is strongly convex and that $f_i, i = 1, \dots, q$ are convex and $h_i, i = 1, \dots, r$ are affine. Then one

of two situations can happen. Either the minimizing argument to (1.12) is a feasible solution to (1.4), then it is also optimal due to (1.11), or the minimizing argument to (1.12) is infeasible for (1.4), then no optimal point to (1.4) can exist (cf. [Boyd and Vandenberghe, 2004, §5.5.5]).

Karush-Kuhn-Tucker Optimality Conditions

In this section we describe the Karush-Kuhn-Tucker (KKT) optimality conditions for optimization problem (1.4) when the functions f_0, \dots, f_q and h_1, \dots, h_r are differentiable.

Nonconvex problems. The following conditions (KKT) must be satisfied for any dual feasible optimum (λ^*, μ^*) and primal feasible optimum x^* for which strong duality holds (cf. [Boyd and Vandenberghe, 2004, §5.5.3])

$$f_i(x^*) \leq 0, \quad i = 1, \dots, q \quad (1.13)$$

$$h_i(x^*) = 0, \quad i = 1, \dots, r \quad (1.14)$$

$$\mu_i^* \geq 0, \quad i = 1, \dots, q \quad (1.15)$$

$$\mu_i^* f_i(x^*) = 0, \quad i = 1, \dots, q \quad (1.16)$$

$$\nabla f_0(x^*) + \sum_{i=1}^q \mu_i^* \nabla f_i(x^*) + \sum_{i=1}^r \lambda_i^* \nabla h_i(x^*) = 0. \quad (1.17)$$

The first two conditions imply that the primal problem is feasible, while the third condition ensures that the dual problem is feasible. The fourth condition is the complementary slackness and the final condition holds since x^* minimizes $\mathcal{L}(x, \lambda^*, \mu^*)$ over x which implies that the gradient w.r.t. x must vanish for optimal x^* .

Convex problems. For convex problems the KKT conditions are not only necessary (as in the nonconvex case above) but also sufficient for (λ^*, μ^*) and x^* to be primal and dual optimal. A condition for (1.4) to be convex is that f_i are convex and h_i are affine. This implies that any points (λ^*, μ^*) and x^* that satisfy the KKT conditions

$$f_i(x^*) \leq 0, \quad i = 1, \dots, q \quad (1.18)$$

$$h_i(x^*) = 0, \quad i = 1, \dots, r \quad (1.19)$$

$$\mu_i^* \geq 0, \quad i = 1, \dots, q \quad (1.20)$$

$$\mu_i^* f_i(x^*) = 0, \quad i = 1, \dots, q \quad (1.21)$$

$$\nabla f_0(x^*) + \sum_{i=1}^q \mu_i^* \nabla f_i(x^*) + \sum_{i=1}^r \lambda_i^* \nabla h_i(x^*) = 0 \quad (1.22)$$

are primal and dual optimal and $g(\lambda^*, \mu^*) = f_0(x^*)$, i.e., strong duality holds (cf. [Boyd and Vandenberghe, 2004, §5.5.3]).

Constraint Qualification

A number of different conditions can be stated for the optimization problem that ensure the KKT conditions to hold. Such conditions are called constraint qualifications. Below we describe two different constraint qualifications.

Slater constraint qualification. The most commonly used constraint qualification is Slater's constraint qualification which can be applied to convex optimization problems, i.e., for optimization problems of the form

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, q \\ & Ax = b. \end{aligned}$$

Slater's constraint qualification is presented in the following proposition which is proven, e.g., in [Boyd and Vandenberghe, 2004, §5.3.2].

PROPOSITION 1.4

Assume that f_i for $i = 0, \dots, q$ are continuously differentiable and convex and that there exists a point \tilde{x} such that

$$f_i(\tilde{x}) < 0, \quad i = 1, \dots, q, \quad A\tilde{x} = b$$

then there exist λ^*, μ^* and x^* such that the KKT conditions (1.18)-(1.22) hold. \square

Mangasarian-Fromovitz constraint qualification. A more general constraint qualification is the Mangasarian-Fromovitz constraint qualification (MFCQ). It applies to problems of the form (1.4). MFCQ is presented in the following proposition and proven, e.g., in [Bertsekas, 1999, Proposition 3.3.8].

PROPOSITION 1.5

Assume that $f_i, i = 0, \dots, q$ are continuously differentiable and that x^* is a local minimum of (1.4). Further assume that the gradients $\nabla h_i(x^*)$ for $i = 1, \dots, r$ are linearly independent and that there exists a vector v such that

$$\nabla h_i(x^*)^T v = 0, \quad \forall i = 1, \dots, r, \quad \nabla f_j(x^*)^T v < 0, \quad \forall j \in A^*(x^*)$$

where $A^*(x^*)$ is the set indices j for which the constraint $f_j(x^*) = 0$. Then there exist λ^*, μ^* such that the KKT conditions (1.13)-(1.17) hold. \square

If the matrix A defining the affine equality constraints $Ax = b$ has full row rank, then Slater's condition implies MFCQ [Bertsekas, 1999, Proposition 3.3.9].

An interesting property related to MFCQ is boundedness of the optimal dual variables as shown in [Gauvin, 1977]. Before we state this result, we introduce the set of optimal dual variables

$$Z(x) = \{\lambda^* \in \mathbb{R}^r, \mu^* \in \mathbb{R}_{\geq 0}^q \mid (1.13) - (1.17) \text{ holds with } x^* = x\}.$$

PROPOSITION 1.6

Let x^* be a local minimum of (1.4) where f_i for $i = 0, \dots, q$ and h_i for $i = 1, \dots, r$ are continuously differentiable. Then, boundedness and non-emptiness of $Z(x^*)$ is equivalent to that MFCQ holds. \square

1.3 Gradient-Based Optimization

In this section we describe subgradient, gradient and accelerated gradient methods. We start with a brief discussion on subgradient methods.

Subgradient Methods

We consider unconstrained convex optimization problems, i.e., problems of the form

$$\text{minimize } f_0(x) \tag{1.23}$$

where $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex and $x \in \mathbb{R}^n$. Such problems can be solved by the following subgradient iterations

$$x^{k+1} = x^k - t^k \frac{\xi(x^k)}{\|\xi(x^k)\|}$$

where $\xi(x^k)$ is a subgradient of f_0 at x^k , i.e., $\xi(x^k)$ satisfies (1.2), and t^k is the step size at iteration k . To show asymptotic convergence it is enough that the subgradients are bounded and to use step sizes that satisfy (cf. [Nesterov, 2003, Theorem 3.2.2])

$$t^k \geq 0, \quad \sum_{k=1}^{\infty} t^k = \infty, \quad \sum_{k=1}^{\infty} (t^k)^2 < \infty.$$

For more on subgradient methods the reader is referred to [Shor, 1985, Bertsekas, 1999, Nesterov, 2003, Polyak, 1987].

Gradient Methods

If the objective function f_0 in (1.23) is convex and differentiable, then gradient methods can be applied. We present a gradient algorithm below; choose x_0 , for $k \geq 1$ the iterations are defined by

$$x^{k+1} = x^k - t^k \nabla f_0(x^k) \quad (1.24)$$

where $t^k > 0$ is a step size parameter. The step size parameter can be chosen in many different ways, see [Bertsekas, 1999, Chapter 1] for an overview. For cost functions f_0 that have a Lipschitz continuous gradient convergence can be shown using a constant step size $t^k = t$. If we denote by L the Lipschitz constant to ∇f_0 then the optimal constant step size is $t = 1/L$ (cf. [Nesterov, 2003, Corollary 2.1.2]) and the convergence rate is (cf. [Beck and Teboulle, 2009, Theorem 3.1])

$$f_0(x^k) - p^* \leq \frac{L \|x^0 - x^*\|^2}{2k} \quad (1.25)$$

where x^* is any optimal point.

Proximal gradient methods. The gradient method has been generalized to handle problems of the form (see [Beck and Teboulle, 2009] and the references therein)

$$\text{minimize } f_0(x) + P(x) \quad (1.26)$$

where $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex and differentiable, ∇f_0 is Lipschitz continuous with Lipschitz constant L , and $P : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is lower semi-continuous and convex with $P(x) < \infty$ for at least one x and $P(x) > -\infty$ for every x . The proximal gradient method is defined by the following iteration

$$x^{k+1} = \arg \min_x \left\{ f_0(x^k) + \nabla f_0(x^k)^T (x - x^k) + \frac{L}{2} \|x - x^k\|^2 + P(x) \right\}. \quad (1.27)$$

Due to the quadratic upper bound property (1.3) of functions with a Lipschitz continuous gradient, the minimand in (1.27) is a quadratic upper bound to $f_0 + P$ which tangent the function $f_0 + P$ at x^k . Hence, a quadratic upper bound to $f_0 + P$ is minimized in each iteration of the proximal gradient algorithm.

The proximal gradient algorithm is shown to have the same convergence rate properties as the classical gradient method (1.25) (cf. [Beck

and Teboulle, 2009]). For $P \equiv 0$ we recover the classical gradient method with step size $t = 1/L$ since for $P \equiv 0$ we have

$$\begin{aligned}
 x^{k+1} &= \arg \min_x \left\{ f_0(x^k) + \nabla f_0(x^k)^T (x - x^k) + \frac{L}{2} \|x - x^k\|^2 + P(x) \right\} \\
 &= \arg \min_x \left\{ \nabla f_0(x^k)^T x + \frac{L}{2} \|x - x^k\|^2 \right\} \\
 &= \arg \min_x \left\{ (\nabla f_0(x^k) - Lx^k)^T x + \frac{L}{2} x^T x \right\} \\
 &= x^k - \frac{1}{L} \nabla f_0(x^k). \tag{1.28}
 \end{aligned}$$

Convex constrained optimization problems can also be solved using the proximal gradient method (1.27) by setting P to be the indicator function of the constraint set. This implies that a constrained quadratic optimization problem needs to be solved in each iteration of the algorithm.

For cases where the Lipschitz constant L is difficult to determine, the proximal gradient method can be used with backtracking line search with maintained convergence rate (cf. [Beck and Teboulle, 2009]).

The presented gradient methods are, however, not optimal w.r.t. what can be achieved by gradient methods. An exact lower bound for achievable performance for gradient methods applied to functions with Lipschitz continuous gradient is (cf. [Nemirovsky and Yudin, 1983, Nesterov, 2003])

$$f_0(x^k) - p^* \geq \frac{L \|x^0 - x^*\|^2}{8(k+1)^2}. \tag{1.29}$$

Accelerated Gradient Methods

The first gradient algorithm to achieve the convergence rate (1.29) up to a constant factor was presented in [Nesterov, 1983]. We refer to gradient algorithms that achieve this improved convergence rate by *accelerated gradient methods*. Variations of the accelerated gradient method in [Nesterov, 1983] have been presented in [Nesterov, 1988, Nesterov, 2003, Nesterov, 2005].

Accelerated proximal gradient methods. Some accelerated gradient methods have been presented that solve problems of the form (1.26). In [Beck and Teboulle, 2009] an accelerated gradient method that generalizes the method in [Nesterov, 1983] to solve problems of the form (1.26) was presented and in [Nesterov, 2007] the method in [Nesterov, 2005] was generalized to solve (1.26). A unified framework for accelerated proximal gradient methods was presented in [Tseng, 2008] where also a generalization to the algorithm in [Nesterov, 1988] was presented.

The algorithm from [Beck and Teboulle, 2009] as presented in [Tseng, 2008] is described by the following iterations for, $k \geq 0$:

$$y^k = x^k + \theta^k ((\theta^{k-1})^{-1} - 1)(x^k - x^{k-1}), \quad (1.30)$$

$$x^{k+1} = \arg \min_x \left\{ f_0(y^k) + \nabla f_0(y^k)^T (x - y^k) + \frac{L}{2} \|x - y^k\|^2 + P(x) \right\} \quad (1.31)$$

$$\theta^{k+1} = \frac{\sqrt{(\theta^k)^4 + 4(\theta^k)^2} - (\theta^k)^2}{2} \quad (1.32)$$

where $\theta^1 = \theta^0 = 1$, and the initial iterate $x^0 = x^{-1}$ needs to be chosen. In [Tseng, 2008] it was noted that the θ^k -sequence $\theta^k = 2/(k+2)$ can be used instead of (1.32). This choice of θ^k -sequence gives the following even simpler iterations for $k \geq 0$:

$$y^k = x^k + \frac{k-1}{k+2} (x^k - x^{k-1}) \quad (1.33)$$

$$x^{k+1} = \arg \min_x \left\{ f_0(y^k) + \nabla f_0(y^k)^T (x - y^k) + \frac{L}{2} \|x - y^k\|^2 + P(x) \right\}. \quad (1.34)$$

Both (1.30)-(1.32) and (1.33)-(1.34) share the following convergence rate property

$$f_0(x^k) - p^* \leq \frac{2L\|x^0 - x^*\|^2}{(k+1)^2} \quad (1.35)$$

which is proven in [Tseng, 2008, Corollary 2]. However, (1.30)-(1.32) performs slightly better in practice since that θ^k -sequence tends to zero somewhat faster within the allowed bounds. The convergence rate (1.35) for the accelerated gradient methods is up to a constant factor the same as the best achievable convergence rate (1.29). Also, the convergence rate for the accelerated gradient methods (1.35) is much better than the convergence rate for the non-accelerated proximal gradient method (1.25) despite that the complexity of the algorithms are nearly the same.

Generalized accelerated proximal gradient methods. The accelerated gradient method in [Beck and Teboulle, 2009] has been further generalized in [Zuo and Lin, 2011]. The method applies to problems of the form (1.26) where the requirement of ∇f_0 being Lipschitz continuous is replaced by the requirement that for every $x, y \in \mathbb{R}^n$ the following holds:

$$f_0(x) \leq f_0(y) + \nabla f_0(y)^T (x - y) + \frac{1}{2} \|x - y\|_M^2 \quad (1.36)$$

where $M \in \mathbb{R}^{n \times n}$ is a positive definite matrix. The algorithm with $\theta^k = 2/(k+2)$ is described below, for $k \geq 0$

$$y^k = x^k + \frac{k-1}{k+2}(x^k - x^{k-1})$$

$$x^{k+1} = \arg \min_x \left\{ f_0(y^k) + \nabla f_0(y^k)^T (x - y^k) + \frac{1}{2} \|x - y^k\|_M^2 + P(x) \right\}$$

which is very similar to (1.33)-(1.34). The requirement (1.36) on f_0 is very similar to the Lipschitz continuity requirement on ∇f_0 , which is equivalent to (1.3), for non-generalized accelerated gradient methods. The requirement (1.36) allows for quadratic upper bounds that need not have the same curvature in every direction. For appropriately chosen M (1.36) might give a significantly tighter upper bound to f_0 than (1.3). This can improve convergence for ill-conditioned problems for which gradient-based methods are known to have slow convergence. However, the choice of M is up to the user for different applications. The choice $M = LI$ where L is the Lipschitz constant to ∇f_0 gives the iterations (1.33)-(1.34). The convergence rate for the generalized accelerated gradient method is (cf. [Zuo and Lin, 2011])

$$f_0(x^k) - p^* \leq \frac{2\|x^0 - x^*\|_M^2}{(k+1)^2}. \quad (1.37)$$

1.4 Distributed Optimization Methods

We describe two different distributed optimization methods in this section, namely primal and dual decomposition. The objective for the decomposition is to distribute the computations when computing the solution to an optimization problem. The decomposition techniques can sometimes give improved performance compared to other solution techniques since parallel computational units can be utilized. Another advantage is that it provides increased flexibility for implementation of optimization algorithms on networked problems and that it enables for very large problems to be solved. See [Bertsekas, 1999, Chapter 6] or [Boyd *et al.*, 2008] for more on primal and dual decomposition.

To describe the decomposition techniques we consider the following

optimization problem

$$\begin{aligned}
 & \text{minimize } \sum_{i=1}^j f_i(x_i) && (1.38) \\
 & \text{subject to } x_i \in X_i, && i = 1, \dots, j \\
 & && \sum_{i=1}^j y_i = b, \quad A_i x_i = y_i, \quad i = 1, \dots, j.
 \end{aligned}$$

The vectors $x_i \in \mathbb{R}^{n_i}$ are local vectors, $A_i \in \mathbb{R}^{m \times n_i}$, $y_i \in \mathbb{R}^m$, and $b \in \mathbb{R}^m$. The sets X_i are feasible sets for the corresponding local vectors x_i and are assumed nonempty, closed, and convex. Further, the functions $f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ are assumed convex. The equality constraints are referred to as a *complicating constraints* since they involve not only local variables. The decomposition methods can handle also inequality constraints as complicating constraints but we restrict our presentation to equality constraints for brevity.

Primal Decomposition

To decompose (1.38) using primal decomposition we see by fixing y_i that (1.38) can be decomposed into the following subproblems

$$\begin{aligned}
 \phi_i(y_i) &= \min_{x_i} f_i(x_i) && (1.39) \\
 & \text{s.t. } x_i \in X_i, \quad A_i x_i = y_i
 \end{aligned}$$

for $i = 1, \dots, j$. These subproblems are convex and can be solved simultaneously in parallel. Using these subproblems, we construct the following problem that is equivalent to (1.38)

$$\begin{aligned}
 & \text{minimize } \phi(y_1, \dots, y_j) = \sum_{i=1}^j \phi_i(y_i) && (1.40) \\
 & \text{subject to } \sum_{i=1}^j y_i = b, \quad y_i \in \mathcal{Y}_i, \quad i = 1, \dots, j
 \end{aligned}$$

where \mathcal{Y}_i is the set of feasible vectors for which (1.39) has at least one feasible solution. The problem (1.40) is referred to as the *master problem*. The objective of the master problem is to distribute y_i to the different subsystems optimally. The objective function defining the master problem is often non-differentiable but can be solved using subgradient methods. By

letting $\lambda_i^*(y_i)$ be an optimal dual variable for constraint $A_i x_i = y_i$ in subproblem (1.39) we get that a subgradient to ϕ is given by (cf. [Bertsekas, 1999, §6.4.2])

$$-[\lambda_1^*(y_1), \dots, \lambda_j^*(y_j)]^T \in \partial\phi(y_1, \dots, y_j).$$

We denote by $x_i^*(y_i)$ the optimal primal variables for subproblem (1.39) and introduce the stacked vectors

$$y = [y_1^T, \dots, y_j^T]^T, \quad \lambda^*(y) = [\lambda_1^*(y_1)^T, \dots, \lambda_j^*(y_j)^T]^T.$$

Using this notation, the full algorithm when solving the master problem (1.40) using a subgradient method consists of solving subproblems (1.39) in parallel to find $x_i^*(y_i^k)$ and $\lambda_i^*(y_i^k)$, where k is the iteration number. Then the allocation variables y are updated according to

$$y^{k+1} = [y^k + t^k \lambda^*(y^k)]^+$$

where t^k is a positive step size and $[\cdot]^+$ denotes the Euclidean projection onto the constraint set

$$\left\{ y \mid \sum_{i=1}^j y_i = b, y_i \in \mathcal{Y}_i, i = 1, \dots, j \right\}.$$

A great benefit of primal decomposition is that the original problem is feasible in every iteration.

Dual Decomposition

Dual decomposition is an old technique that dates back to the early 1960s [Everett, 1963, Benders, 1962, Dantzig and Wolfe, 1960]. To solve (1.38), assuming that some constraint qualification holds, using dual decomposition, we introduce dual variables $\lambda \in \mathbb{R}^m$ and form the following dual problem

$$\sup_{\lambda} \inf_{x_i \in \mathcal{X}_i} \sum_{i=1}^j f_i(x_i) + \lambda^T \left(\sum_{i=1}^j A_i x_i - b \right). \quad (1.41)$$

For fixed $\lambda \in \mathbb{R}^m$ the inner minimization problem can be decomposed into the following subproblems

$$g_i(\lambda) = \inf_{x_i \in \mathcal{X}_i} f_i(x_i) + \lambda^T A_i x_i \quad (1.42)$$

for $i = 1, \dots, J$. These problems are convex and can be solved simultaneously in parallel. To guarantee that the minimum in (1.41) is attained we assume, besides the assumption on X_i being nonempty, closed, and convex, that f_i are strongly convex. These assumptions imply the existence of a solution by Weierstrass extreme value theorem, see Proposition 1.1. The master problem in dual decomposition is to find $\lambda \in \mathbb{R}^m$ that solves

$$\sup_{\lambda} g(\lambda) = \sum_{i=1}^J g_i(\lambda) \quad (1.43)$$

which is equivalent to (1.41). Since the objective function is assumed strongly convex, the dual function is differentiable and has gradient (cf. Proposition 1.2)

$$\nabla g(\lambda) = \sum_{i=1}^J A_i x_i^*(\lambda) - b$$

where $x_i^*(\lambda)$ is the optimal solution to (1.42) for given λ . Since the dual problem is differentiable, it can be solved using gradient methods. To solve the dual problem (1.43) using a gradient method, the subproblems (1.42) are solved in parallel to get $x_i^*(\lambda^k)$, where k is the iteration number. Then the dual variables are updated according to

$$\lambda^{k+1} = \lambda^k + t^k \left(\sum_{i=1}^J A_i x_i^*(\lambda^k) - b \right)$$

where t^k is a step size parameter.

1.5 Model Predictive Control

Model predictive control (MPC) is an optimization based control methodology that optimizes plant behavior based on state predictions from a plant model. In each sampling instant, a finite horizon optimal control problem is solved with the current state of the plant used as initial condition for the state predictions. The first control action from the optimal control trajectory is applied to the plant. This procedure is repeated in each sample, which introduces feedback into the methodology. MPC is becoming increasingly used in the process industry, see [Qin and Badgwell, 2003] for survey of MPC applications in the industry. For a thorough description of MPC, the reader is referred to [Maciejowski, 2002, Rawlings and Mayne, 2009] or the survey paper [Mayne *et al.*, 2000].

Optimal Control Problem

The plant to be controlled is usually described by a differential equation. However, since it is common to apply piece-wise constant control trajectories, the differential equation is often approximated by, or exactly reformulated as, a difference equation

$$x_{t+1} = f(x_t, u_t), \quad x_0 = \bar{x}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$ for $t \in \mathbb{N}_{\geq 0}$ and $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$. We assume that the plant has an equilibrium point in the origin, i.e., that $f(0, 0) = 0$ and that the system is controllable. A benefit of model predictive control over other control methodologies is its ability to handle state and control constraints, i.e., that the states and controls can be forced to satisfy $x \in \mathcal{X}$ and $u \in \mathcal{U}$ if possible. In this overview, we consider the regulation control problem which is to steer the system state to the origin while respecting the constraints. To achieve this, a cost function is used that penalizes deviations from the desired equilibrium point

$$J_N(\mathbf{x}, \mathbf{u}) = \ell_f(x_N) + \sum_{t=0}^{N-1} \ell(x_t, u_t)$$

where $\mathbf{x} = [x_0^T, x_1^T, \dots, x_N^T]^T$ and $\mathbf{u} = [u_0^T, u_1^T, \dots, u_{N-1}^T]^T$. We assume that the stage cost $\ell : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}_{\geq 0}$ satisfies $c_1(\|[x^T, u^T]^T\|) \leq \ell(x, u) \leq c_2(\|[x^T, u^T]^T\|)$ where $c_1(0) = c_2(0) = 0$ and c_1, c_2 are continuous, strictly increasing and unbounded. Often a terminal constraint set \mathcal{X}_f is used where $\mathcal{X}_f \subseteq \mathcal{X}$, i.e., we impose that $x_N \in \mathcal{X}_f$. This gives the following optimal control problem

$$\begin{aligned} V_N(\bar{x}) &= \min_{\mathbf{x}, \mathbf{u}} J_N(\mathbf{x}, \mathbf{u}) & (1.44) \\ \text{s.t. } x_{t+1} &= f(x_t, u_t), & t = 0, \dots, N-1 \\ x_0 &= \bar{x} \\ x &\in \mathcal{X}, u \in \mathcal{U}, & t = 0, \dots, N-1 \\ x_N &\in \mathcal{X}_f. \end{aligned}$$

We denote the optimal state and control sequences to (1.44) for initial state \bar{x} by $\{x_t^*(\bar{x})\}_{t=0}^N$ and $\{u_t^*(\bar{x})\}_{t=0}^{N-1}$ respectively and define the set of initial conditions such that (1.44) is feasible by \mathbb{X}_N . We hereafter assume that f , ℓ and ℓ_f are continuous, \mathcal{U} is compact and $\mathcal{X}, \mathcal{X}_f$ are closed. Under these assumptions and since N is finite, it can through Weierstrass extreme value theorem (see Proposition 1.1, [Bertsekas, 1999, Proposition A.8]) be

shown that the minimum in (1.44) is attained for every $\bar{x} \in \mathbb{X}_N$ (cf. [Mayne *et al.*, 2000]).

In MPC, the first control action $u_0^*(\bar{x})$ in the control sequence is applied to the system. In the following sample, the optimization problem (1.44) is solved with the new measured state as initial condition and the first control action from the obtained control sequence is applied to the system. This procedure is repeated in each sample which defines the static feedback control law $v_N(\bar{x}) = u_0^*(\bar{x})$.

Stability and Feasibility

There are numerous ways to prove stability and feasibility of the closed loop system using the MPC control law v_N . In [Mayne *et al.*, 2000] different methods to prove stability and feasibility presented in the literature was summarized. Most stability results use a terminal controller v_f , besides the terminal constraints \mathcal{X}_f and the terminal cost ℓ_f . We will see that if v_f , \mathcal{X}_f and ℓ_f are such that the following four assumptions hold (cf. [Mayne *et al.*, 2000]), stability and recursive feasibility can be guaranteed.

- A1: $\mathcal{X}_f \subseteq \mathcal{X}$, \mathcal{X}_f closed, $0 \in \mathcal{X}_f$ (state constraint satisfied in \mathcal{X}_f).
- A2: $v_f(\bar{x}) \in \mathcal{U}$, $\forall \bar{x} \in \mathcal{X}_f$ (control constraint satisfied in \mathcal{X}_f).
- A3: $f(\bar{x}, v_f(\bar{x})) \in \mathcal{X}_f$, $\forall \bar{x} \in \mathcal{X}_f$ (\mathcal{X}_f positively invariant under v_f).
- A4: $\ell_f(\bar{x}) \geq \ell_f(f(\bar{x}, v_f(\bar{x}))) + \ell(\bar{x}, v_f(\bar{x}))$, $\forall \bar{x} \in \mathcal{X}_f$ (ℓ_f is a local Lyapunov function).

Below we indicate how stability and recursive feasibility can be established under Assumptions A1-A4, see [Mayne *et al.*, 2000] for details. For each $\bar{x} \in \mathbb{X}_N$ we know by definition that (1.44) is feasible. We construct the following shifted control trajectory

$$\tilde{\mathbf{u}}(\bar{x}) = [u_1^*(\bar{x})^T, u_2^*(\bar{x})^T, \dots, u_{N-1}^*(\bar{x})^T, v_f(x_N^*(\bar{x}))^T]^T$$

and define the corresponding state trajectory

$$\tilde{\mathbf{x}}(\bar{x}) = [x_1^*(\bar{x})^T, x_2^*(\bar{x})^T, \dots, x_N^*(\bar{x})^T, f(x_N^*(\bar{x}), v_f(x_N^*(\bar{x})))^T]^T.$$

Since $\bar{x} \in \mathbb{X}_N$ we have that $x_N^*(\bar{x}) \in \mathcal{X}_f$ which by Assumption A2 implies that $v_f(x_N^*(\bar{x})) \in \mathcal{U}$ and by Assumptions A1, A3 implies that

$$f(x_N^*(\bar{x}), v_f(x_N^*(\bar{x}))) \in \mathcal{X}_f \subseteq \mathcal{X}.$$

Thus, Assumptions A1-A3 imply that $(\tilde{\mathbf{x}}(\bar{x}), \tilde{\mathbf{u}}(\bar{x}))$ is a feasible solution to (1.44) with initial state $f(\bar{x}, u_0^*(\bar{x}))$. Further

$$V_N(f(\bar{x}, u_0^*(\bar{x}))) \leq J_N(\tilde{\mathbf{x}}(\bar{x}), \tilde{\mathbf{u}}(\bar{x})) \leq V_N(\bar{x}) - \ell(\bar{x}, u_0^*(\bar{x}))$$

where Assumption A4 has been used to conclude the last inequality. Using the assumed properties of ℓ it can be shown that V_N is a Lyapunov function for the system. Further, since $f(\bar{x}, u_0^*(\bar{x})) \in \mathbb{X}_N$ the procedure can be repeated in the following time sample which gives recursive feasibility and asymptotic stability.

Many different methods to choose the terminal constraint set, the terminal cost, and the terminal controller that give the desired characteristics A1-A4 have been proposed. Below some of these are described starting with the case of systems with linear dynamics.

Linear Systems. One important and widely used special case in MPC is the problem with linear dynamics, i.e.,

$$x_{t+1} = f(x_t, u_t) = Ax_t + Bu_t, \quad x_0 = \bar{x}$$

where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, with polytopic constraint sets \mathcal{X} , \mathcal{U} , and with a quadratic cost

$$\ell(x, u) = \frac{1}{2} (x^T Q x + u^T R u)$$

where $Q \succeq 0$ and $R \succ 0$. A linear terminal control law $v_f(x) = K_f x$ is stabilizing if it satisfies $\rho(A + BK_f) < 1$, where $\rho(\cdot)$ denotes the spectral radius. The largest terminal set \mathcal{X}_f such that Assumptions A1-A3 hold is the maximal output admissible set (cf. [Gilbert and Tan, 1991]) for the dynamics $x_{t+1} = (A + BK_f)x_t$. Under certain assumptions [Gilbert and Tan, 1991, Theorem 4.1] the maximal output admissible set can be described by a finite number of linear inequalities. To satisfy also Assumption A4 the terminal cost ℓ_f must be chosen. A natural choice is to have a quadratic terminal cost $\ell_f(x) = x^T P_f x$ where $P_f \succ 0$. By setting $A_f := A + BK_f$ we get by insertion of the terminal cost into the condition in Assumption A4 that

$$x^T P_f x \geq x^T A_f^T P_f A_f x + x^T Q x + x^T K_f^T R K_f x$$

should hold for all $x \in \mathcal{X}_f$. This holds if $P_f \succ 0$ is chosen such that the following discrete-time Lyapunov matrix inequality in P_f holds

$$P_f \succeq A_f^T P_f A_f + Q + K_f^T R K_f. \quad (1.45)$$

The matrices K_f and P_f can in the linear case be chosen to achieve infinite horizon optimal performance for constrained linear systems as first proposed in [Sznaier and Damborg, 1987]. We choose the terminal control

law $v_f(x) = K_f x$ as the infinite horizon optimal feedback controller for the unconstrained case and P_f as the matrix defining the corresponding optimal value function. To determine such K_f and P_f is the well known LQ-problem and is found by solving the discrete time algebraic Riccati equation (see [Zhou *et al.*, 1996, Boyd and Barratt, 1991] for more on LQ). Insertion of the infinite horizon optimal matrices K_f and P_f gives equality in (1.45), which implies that Assumption A4 is satisfied. The terminal constraint set \mathcal{X}_f is chosen as the maximal output admissible set (cf. [Gilbert and Tan, 1991]) for the dynamics $x_{t+1} = (A + BK_f)x_t$. Inside the output admissible set no constraints are active and the system state will never leave this set once entered. The terminal constraint set \mathcal{X}_f is not used explicitly in the optimization routine, rather the control horizon is adapted until the final state in the trajectory is inside the terminal set. Stability guarantees for this scheme were presented in [Chmielewski and Manousiouthakis, 1996, Scokaert and Rawlings, 1998].

Nonlinear Systems. Nonlinear systems without state or control constraints has been treated in [Parisini and Zoppoli, 1995] for the discrete time case and in [Jadbabaie *et al.*, 2001] for the continuous time case. They use a stabilizing control law v_f and a terminal cost function ℓ_f that is a local Lyapunov function for the stabilized system. The terminal constraint set is chosen to be level sets of the local Lyapunov function ℓ_f and to be positively invariant for $x_{t+1} = f(x_t, v_f(x_t))$.

Nonlinear systems with constraints was treated in [De Nicolao *et al.*, 1996] for discrete time systems and in [Chen and Allgöwer, 1998] for continuous time systems. In both papers, v_f is chosen to stabilize the linearized system with linearization point in the origin. They differ in the choice of terminal cost ℓ_f which was chosen quadratic in [Chen and Allgöwer, 1998] and non-quadratic in [De Nicolao *et al.*, 1996]. The terminal constraint set \mathcal{X}_f is positively invariant for the nonlinear system and satisfies $\mathcal{X}_f \subseteq \mathcal{X}$ and $v_f(\mathcal{X}_f) \subseteq \mathcal{U}$. It was in [Mayne *et al.*, 2000] pointed out that all these choices of terminal controller, terminal constraint set and terminal cost are different ways of satisfying Assumptions A1-A4.

Stability without Terminal Cost or Terminal Constraints

Assumptions A1-A4 rely on a terminal controller, a terminal cost and a terminal constraint set to prove stability. Stability results for MPC without terminal cost, terminal controller or terminal constraint set, i.e., for

problems of the form

$$\begin{aligned}
 V_N(\bar{x}) = \min & \sum_{t=0}^{N-1} \ell(x_t, u_t) & (1.46) \\
 \text{s.t. } & x_{t+1} = f(x_t, u_t), & t = 0, \dots, N-2 \\
 & x_0 = \bar{x} \\
 & x \in \mathcal{X}, u \in \mathcal{U}, & t = 0, \dots, N-1
 \end{aligned}$$

has been presented in the literature. In [Grimm *et al.*, 2005] it was shown that closed loop stability holds when the control horizon is sufficiently long. In [Grüne and Rantzer, 2008] this was further elaborated on by showing stability based on an assumption on the relation between the optimal value function and the optimal stage-cost. The result in [Grüne and Rantzer, 2008] relies on relaxed dynamic programming which was presented in [Lincoln and Rantzer, 2006, Rantzer, 2006]. Relaxed dynamic programming applied to MPC states that if the following holds for all $x \in \mathcal{X}$

$$V_N(x) \geq V_N(f(x, v_N(x)) + \alpha \ell(x, v_N(x)) \quad (1.47)$$

where $\alpha \in (0, 1]$ is a suboptimality parameter, $v_N(x) = u_0^*(x)$, and u_0^* and V_N refer to the optimal solution to (1.46). Then the closed loop system $x_{t+1} = f(x_t, v_N(x_t))$ satisfies

$$\alpha \sum_{t=0}^{\infty} \ell(x_t, v_N(x_t)) \leq V_{\infty}(x_0).$$

Using appropriate assumptions on ℓ , (1.47) also implies asymptotically stability of the closed loop system. Further progress was reported in [Grüne, 2009], where it was shown how to compute the minimal control horizon that satisfies (1.47) by solving a linear program. This result is based on a quantification of a controllability assumption on the stage-costs.

Distributed Model Predictive Control

In distributed model predictive control (DMPC) the system to be controlled consists of several smaller subsystems that are coupled. The coupling might be due to a non-separable cost, due to dynamic interaction, or due to complicating constraints. In this section we focus on problems with interacting dynamics but separable cost and constraints. To introduce the system description we assign a unique label from the set $\{1, 2, \dots, \mathcal{J}\}$ to each subsystem where \mathcal{J} is the total number of subsystems. The subsystem interaction can be described by a directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ where the set $\mathcal{E} = \{1, 2, \dots, \mathcal{J}\}$ contains indices for the nodes in the graph and

the set \mathcal{V} contains ordered pairs (i, j) . The pair $(i, j) \in \mathcal{V}$ if and only if subsystem j is directly influenced by subsystem i through the dynamics. Using this, we construct the set of neighbors for each subsystem i as

$$\mathcal{N}_i = \{j \in \mathcal{E} \setminus i \mid (j, i) \in \mathcal{V}\}.$$

We also introduce the following vector which contains the states for all neighbors to subsystem i , $x^{-i} = (\dots, x^j, \dots)$, $x^j \in \mathcal{N}_i$ where $x^j \in \mathbb{R}^{n_j}$ are the states associated with subsystem j .

Two main directions have emerged in the distributed model predictive control literature. One direction is to pose centralized optimization problems and solve them in a distributed manner. This yields a centralized optimal solution. The other direction is to pose local optimization problems with constraints on neighboring interaction that can be used to prove stability. Both approaches apply to sparse systems where each local subsystem is described by

$$x_{t+1}^i = f_i(x_t^i, u_t^i, x_t^{-i}), \quad x_0^i = \bar{x}^i.$$

where for $t \in \mathbb{N}_0$, $x_t^i \in \mathbb{R}^{n_i}$, $u_t^i \in \mathbb{R}^{m_i}$, and $x_t^{-i} \in \mathbb{R}^{n_{-i}}$ where $n_{-i} = \sum_{j \in \mathcal{N}_i} n_j$. The controls and states are subject to local constraints, i.e., $u^i \in \mathcal{U}_i$ and $x^i \in \mathcal{X}_i$ where \mathcal{U}_i and \mathcal{X}_i are nonempty, closed and convex sets.

DMPC using distributed optimization. To get a centralized optimization problem that can be solved in distributed fashion, a separable cost is used

$$\sum_{t=0}^{N-1} \sum_{i=1}^J \ell_i(x_t^i, u_t^i).$$

The cost is often chosen strongly convex and quadratic, i.e.,

$$\ell_i(x^i, u^i) = (x^i)^T Q_i x^i + (u^i)^T R_i u^i$$

with $Q_i \in \mathbb{R}^{n_i \times n_i}$ and $R_i \in \mathbb{R}^{m_i \times m_i}$ that satisfy $Q_i \succ 0$ and $R_i \succ 0$ respectively. To get a convex optimization problem, the dynamic constraints need to be linear, i.e., of the form

$$x_{t+1}^i = A_{ii} x_t^i + B_{ii} u_t^i + \left(\sum_{j \in \mathcal{N}_i} A_{ij} x_t^j + B_{ij} u_t^j \right), \quad x_0^i = \bar{x}^i. \quad (1.48)$$

The MPC optimization problem becomes

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^j \left(\sum_{t=0}^{N-1} \ell_i(x_t^i, u_t^i) \right) & (1.49) \\ & \text{subject to} \quad x_t^i \in \mathcal{X}_i, \quad u_t^i \in \mathcal{U}_i, & t = 0, \dots, N-1, \\ & & (1.48), \quad t = 0, \dots, N-2. \end{aligned}$$

In [Venkat *et al.*, 2005, Venkat *et al.*, 2008] this problem was solved in distributed fashion by taking into account the system-wide behavior when optimizing local control action. This implies that full model knowledge is needed in every node. A benefit of the method is that stability and feasibility are guaranteed in every iteration. Another way to solve (1.49) in a distributed fashion is to use dual decomposition. In dual decomposition, small local subproblems are solved in each node and communication between subsystems i and j is needed only if $j \in \mathcal{N}_i$ or $i \in \mathcal{N}_j$. This approach has been used, e.g., in [Negenborn *et al.*, 2008, Wakasa *et al.*, 2008, Doan *et al.*, 2009] where (sub)gradient methods are used to solve the dual problem, and in [Necoara and Suykens, 2008, Necoara *et al.*, 2008] where the accelerated gradient method with smoothing originally presented in [Nesterov, 2005] was used. In the problem description (1.49) neither terminal cost nor terminal constraints are used. Both the terminal cost and terminal constraints used to prove stability in traditional MPC, i.e., Assumptions A1-A4, usually involve all states. This implies that dual decomposition cannot be used with centralized terminal constraints or cost without letting all subsystems communicate with each other.

DMPC with stability constraints. The other direction used in the DMPC literature is to create local optimization problems for each subsystem with stability constraints for the interaction with neighboring subsystems. This approach has been taken in [Jia and Krogh, 2001, Camponogara *et al.*, 2002, Richards and How, 2007] for linear systems where the local optimization problems are solved sequentially and each local solution is passed to neighboring subsystems. By solving sequentially, the subsystems further downstream can satisfy the stability constraint using optimized trajectories from subsystems upstream. In [Dunbar, 2007] a similar approach is used to show stability for nonlinear systems. In [R.M. Hermans, 2010] a stabilizing constraint is used that gives an explicit reduction of a control Lyapunov function. The only communication needed in that approach is to submit initial states between neighbors. This implies that an almost decentralized implementation is possible.

References

- Beck, A. and M. Teboulle (2009): “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” *SIAM J. Imaging Sciences*, **2:1**, pp. 183–202.
- Benders, J. F. (1962): “Partitioning procedures for solving mixed-variables programming problems.” *Numerische Mathematik*, **4:1**, pp. 238–252.
- Bertsekas, D. P. (1999): *Nonlinear Programming*, 2nd edition. Athena Scientific, Belmont, MA.
- Boyd, S. and C. Barratt (1991): *Linear Controller Design—Limits of Performance*. Prentice Hall, Englewood Cliffs, NJ.
- Boyd, S. and L. Vandenberghe (2004): *Convex Optimization*. Cambridge University Press, New York, NY.
- Boyd, S., L. Xiao, A. Mutapcic, and J. Mattingley (2008): “Notes on decomposition methods.” http://see.stanford.edu/materials/lsoctoe364b/08-decomposition_notes.pdf. Notes for EE364B, Stanford University.
- Camponogara, E., D. Jia, B. Krogh, and S. Talukdar (2002): “Distributed model predictive control.” *IEEE Control Systems Magazine*, **22**, pp. 44–52.
- Chen, H. and F. Allgöwer (1998): “A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability.” *Automatica*, **34:10**, pp. 1205–1217.
- Chmielewski, D. and V. Manousiouthakis (1996): “On constrained infinite-time linear quadratic optimal control.” *Systems and Control Letters*, **29:3**, pp. 121–129.
- Dantzig, G. B. and P. Wolfe (1960): “Decomposition principle for linear programs.” *Operations Research*, **8:1**, pp. 101–111.
- De Nicolao, G., L. Magni, and R. Scattolini (1996): “Stabilizing nonlinear receding horizon control via a nonquadratic penalty.” In *Proceedings of the IMACS multiconference CESA*, vol. 1, pp. 185–187. Lille, France.
- Doan, M. D., T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter (2009): “A distributed version of Han’s method for DMPC using local communications only.” *Control Engineering and Applied Informatics*, **11:3**, pp. 6–15.
- Dunbar, W. (2007): “Distributed receding horizon control of dynamically coupled nonlinear systems.” *IEEE Transactions on Automatic Control*, **52**, pp. 1249–1263.

- Everett, H. (1963): “Generalized Lagrange multiplier method for solving problems of optimum allocation of resources.” *Operations Research*, **11**, pp. 399–417.
- Gauvin, J. (1977): “A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming.” *Mathematical Programming*, **12**, pp. 136–138.
- Gilbert, E. and K. Tan (1991): “Linear systems with state and control constraints: the theory and application of maximal output admissible sets.” *IEEE Transactions on Automatic Control*, **36:9**, pp. 1008–1020.
- Grimm, G., M. J. Messina, S. E. Tuna, and A. R. Teel (2005): “Model predictive control: for want of a local control Lyapunov function, all is not lost.” *IEEE Transactions on Automatic Control*, **50:5**, pp. 546–558.
- Grüne, L. (2009): “Analysis and design of unconstrained nonlinear MPC schemes for finite and infinite dimensional systems.” *SIAM Journal on Control and Optimization*, **48**, pp. 1206–1228.
- Grüne, L. and A. Rantzer (2008): “On the infinite horizon performance of receding horizon controllers.” *IEEE Transactions on Automatic Control*, **53**, pp. 2100–2111.
- Hiriart-Urruty, J.-B. and C. Lemarechal (1996): *Convex Analysis and Minimization Algorithms*. Springer-Verlag, Berlin.
- Jadbabaie, A., J. Yu, and J. Hauser (2001): “Unconstrained receding-horizon control of nonlinear systems.” *IEEE Transactions on Automatic Control*, **46:5**, pp. 776–783.
- Jia, D. and B. Krogh (2001): “Distributed model predictive control.” In *Proceedings of the 2001 American Control Conference*, pp. 2767–2772. Arlington, VA.
- Lincoln, B. and A. Rantzer (2006): “Relaxing dynamic programming.” *IEEE Transactions on Automatic Control*, **51:8**, pp. 1249–1260.
- Maciejowski, J. M. (2002): *Predictive Control with Constraints*. Prentice Hall, Essex, England.
- Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert (2000): “Constrained model predictive control: Stability and optimality.” *Automatica*, **36:6**, pp. 789–814.
- Necoara, I., D. Doan, and J. A. K. Suykens (2008): “Application of the proximal center decomposition method to distributed model predictive control.” In *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 2900–2905. Mexico.

Chapter 1. Background

- Necoara, I. and J. Suykens (2008): “Application of a smoothing technique to decomposition in convex optimization.” *IEEE Transactions on Automatic Control*, **53:11**, pp. 2674–2679.
- Negenborn, R., B. De Schutter, and J. Hellendoorn (2008): “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes.” *Engineering Applications of Artificial Intelligence*, **21:3**, pp. 353–366.
- Nemirovsky, A. and D. Yudin (1983): *Informational Complexity and Efficient Methods for Solution of Convex Extremal Problems*. Wiley, New York, NY.
- Nesterov, Y. (1983): “A method of solving a convex programming problem with convergence rate $O(1/k^2)$.” *Soviet Mathematics Doklady*, **27:2**, pp. 372–376.
- Nesterov, Y. (1988): “On an approach to the construction of optimal methods of minimization of smooth convex functions.” *Ékonom. i. Mat. Metody*, **24**, pp. 509–517.
- Nesterov, Y. (2003): *Introductory Lectures on Convex Optimization: A Basic Course*, 1st edition. Springer Netherlands.
- Nesterov, Y. (2005): “Smooth minimization of non-smooth functions.” *Math. Program.*, **103:1**, pp. 127–152.
- Nesterov, Y. (2007): “Gradient methods for minimizing composite objective function.” CORE Discussion Papers 2007076. Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).
- Parisini, T. and R. Zoppoli (1995): “A receding-horizon regulator for nonlinear systems and a neural approximation.” *Automatica*, **31:10**, pp. 1443–1451.
- Polyak, B. (1987): *Introduction to Optimization*. Optimization Software, Inc.
- Qin, S. J. and T. A. Badgwell (2003): “A survey of industrial model predictive control technology.” *Control Engineering Practice*, **11:7**, pp. 733–764.
- Rantzer, A. (2006): “Relaxed dynamic programming in switching systems.” *IEE Proceedings - Control Theory and Applications*, **153:5**, pp. 567–574.
- Rawlings, J. and D. Mayne (2009): *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI.

- Richards, A. and J. How (2007): “Robust distributed model predictive control.” *International Journal of Control*, **80:9**, pp. 1517–1531.
- R.M. Hermans, M. Lazar, A. J. (2010): “Almost decentralized Lyapunov-based nonlinear model predictive control.” In *Proceedings of the 29th American Control Conference*. Baltimore.
- Rockafellar, K. (1970): *Convex Analysis*, vol. 28. Princeton University Press, Princeton, NJ.
- Scokaert, P. and J. Rawlings (1998): “Constrained linear quadratic regulation.” *IEEE Transactions on Automatic Control*, **43:8**, pp. 1163–1169.
- Shor, N. (1985): *Minimization Methods for Nondifferentiable Functions*. Springer, Berlin. Translated from Russian.
- Sznaier, M. and M. J. Damborg (1987): “Suboptimal control of linear systems with state and control inequality constraints.” In *Proceedings of the 26th IEEE conference on decision and control*, pp. 761–762. Los Angeles.
- Tseng, P. (2008): “On accelerated proximal gradient methods for convex-concave optimization.” Submitted to *SIAM J. Optim.* Available: <http://www.math.washington.edu/~tseng/papers/apgm.pdf>.
- Venkat, A., J. Rawlings, and S. Wright (2005): “Stability and optimality of distributed model predictive control.” In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, pp. 6680–6685. Spain.
- Venkat, A. N., I. A. Hiskens, J. B. Rawlings, and S. J. Wright (2008): “Distributed MPC strategies with application to power system automatic generation control.” *IEEE Transactions on Control Systems Technology*, **16:6**, pp. 1192–1206.
- Wakasa, Y., M. Arakawa, K. Tanaka, and T. Akashi (2008): “Decentralized model predictive control via dual decomposition.” In *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 381–386. Cancun, Mexico.
- Zhou, K., J. Doyle, and K. Glover (1996): *Robust and Optimal Control*. Prentice-Hall, Upper Saddle River, NJ.
- Zuo, W. and Z. Lin (2011): “A generalized accelerated proximal gradient approach for total-variation-based image restoration.” *IEEE Transactions on Image Processing*, **20:10**, pp. 2748–2759.

Paper I

Accelerated Gradient Methods and Dual Decomposition in Distributed Model Predictive Control

Pontus Giselsson, Minh Dang Doan, Tamás Keviczky,
Bart De Schutter, and Anders Rantzer

Abstract

We propose a distributed optimization algorithm for mixed $\mathcal{L}_1/\mathcal{L}_2$ -norm optimization based on accelerated gradient methods using dual decomposition. The algorithm achieves convergence rate $O(\frac{1}{k^2})$, where k is the iteration number, which significantly improves the convergence rates of existing duality-based distributed optimization algorithms that achieve $O(\frac{1}{k})$. The performance of the developed algorithm is evaluated on randomly generated optimization problems arising in distributed model predictive control (DMPC). The evaluation shows that, when the problem data is sparse and large-scale, our algorithm can outperform current state-of-the-art optimization software CPLEX and MOSEK.

1. Introduction

Gradient-based optimization methods are known for their simplicity and low complexity within each iteration. A limitation of classical gradient-based methods is the slow rate of convergence. It can be shown [Bertsekas, 1999, Nesterov, 2003] that for functions with a Lipschitz-continuous gradient, i.e., smooth functions, classical gradient-based methods converge at a rate of $O(\frac{1}{k})$, where k is the iteration number. In [Nemirovsky and Yudin, 1983] it was shown that a lower bound on the convergence rate for gradient-based methods is $O(\frac{1}{k^2})$. Nesterov showed in his work [Nesterov, 1983] that an accelerated gradient algorithm can be constructed such that this lower bound on the convergence rate is achieved when minimizing unconstrained smooth functions. This result has been extended and generalized in several publications to handle constrained smooth problems and smooth problems with an additional non-smooth term [Nesterov, 1988, Nesterov, 2005, Beck and Teboulle, 2009] and [Tseng, 2008]. Gradient-based methods are suitable for distributed optimization when they are used in combination with dual decomposition techniques.

Dual decomposition is a well-established concept since around 1960 when Uzawa's algorithm [Arrow *et al.*, 1958] was presented. Similar ideas were exploited in large-scale optimization [Danzig and Wolfe, 1961]. Over the next decades, methods for decomposition and coordination of dynamic systems were developed and refined [Findeisen, 1980, Mesarovic *et al.*, 1970, Singh and Titli, 1978] and used in large-scale applications [Carpentier and Cohen, 1993]. In [Tsitsiklis *et al.*, 1986] a distributed asynchronous method was studied. More recently dual decomposition has been applied in the distributed model predictive control literature in [Doan *et al.*, 2011, Doan *et al.*, 2009, Giselsson and Rantzer, 2010] and [Negenborn *et al.*, 2008] for problems with a strongly convex quadratic cost and arbitrary linear constraints. The above mentioned methods rely on gradient-based optimization, which suffers from slow convergence properties $O(\frac{1}{k})$. Also the step size parameter in the gradient scheme must be chosen appropriately to get good performance. Such information has not been provided or has been chosen conservatively in these publications.

In this work, we improve on the previously presented distributed optimization methods by using an accelerated gradient method to solve the dual problem instead of a classical gradient method. We also extend the class of problems considered by allowing an additional sparse but non-separable 1-norm penalty. Such 1-norm terms are used as regularization term or as penalty for soft constraints [Savorgnan *et al.*, 2011]. Further, we provide the optimal step size parameter for the algorithm, which is crucial for performance. The convergence rate for the dual function value using the accelerated gradient method is implicitly known from [Beck

and Teboulle, 2009, Tseng, 2008]. However, the convergence rate in the dual function value does not indicate the rate at which the primal iterate approaches the primal optimal solution. In this paper we also provide convergence rate results for the primal variables.

Related to our work is the method presented in [Necoara and Suykens, 2008] for systems with a (non-strongly) convex cost. It is based on the smoothing technique presented by Nesterov in [Nesterov, 2005]. Other relevant work is presented in [Kögel and Findeisen, 2011, Richter *et al.*, 2009] in which optimization problems arising in model predictive control (MPC) are solved in a centralized fashion using accelerated gradient methods. These methods are, however, restricted to handle only box-constraints on the control signals.

To evaluate the proposed distributed algorithm, we solve randomly generated large-scale and sparse optimization problems arising in distributed MPC and compare the execution times to state-of-the-art optimization software for large-scale optimization, in particular CPLEX and MOSEK. We also evaluate the performance loss obtained when suboptimal step lengths are used.

The paper is organized as follows. In Section 2, the problem setup is introduced. The dual problem to be solved is introduced in Section 3 and some properties of the dual function are presented. The distributed solution algorithm for the dual problem is presented in Section 4. In Section 5 a numerical example is provided, followed by conclusions drawn in Section 6.

2. Problem Setup

In this paper we present a distributed algorithm for optimization problems with cost functions of the form

$$J(x) = \frac{1}{2}x^T Hx + g^T x + \gamma \|Px - p\|_1. \quad (1)$$

The full decision vector, $x \in \mathbb{R}^n$, is composed of local decision vectors, $x_i \in \mathbb{R}^{n_i}$, according to $x = [x_1^T, \dots, x_M^T]^T$. The quadratic cost matrix $H \in \mathbb{R}^{n \times n}$ is assumed separable, i.e., $H = \text{blkdiag}(H_1, \dots, H_M)$ where $H_i \in \mathbb{R}^{n_i \times n_i}$. Further, H is assumed positive definite with $\underline{\sigma}(H)I \preceq H \preceq \bar{\sigma}(H)I$, where $0 < \underline{\sigma}(H) \leq \bar{\sigma}(H) < \infty$. The linear part $g \in \mathbb{R}^n$ consists of local parts, $g = [g_1^T, \dots, g_M^T]^T$ where $g_i \in \mathbb{R}^{n_i}$. Further, $P \in \mathbb{R}^{m \times n}$ is composed of $P = [P_1, \dots, P_m]^T$, where each $P_r = [P_{r1}^T, \dots, P_{rM}^T]^T \in \mathbb{R}^n$ and $P_{ri} \in \mathbb{R}^{n_i}$. We do not assume that the matrix P should be block-diagonal which means that the cost function J is not separable. However, we assume that the

2. Problem Setup

vectors P_r have sparse structure. Sparsity refers to the property that for each $r \in \{1, \dots, m\}$ there exist some $i \in \{1, \dots, M\}$ such that $P_{ri} = 0$. We also have $p = [p_1, \dots, p_m]^T$ and $\gamma > 0$. This gives the following equivalent formulation of (1)

$$J(x) = \sum_{i=1}^M \left[\frac{1}{2} x_i^T H_i x_i + g_i^T x_i \right] + \sum_{r=1}^m \left| \sum_{i=1}^M P_{ri}^T x_i - p_r \right|. \quad (2)$$

Minimization of (1) is subject to linear equality and inequality constraints

$$A_1 x = B_1, \quad A_2 x \leq B_2$$

where $A_1 \in \mathbb{R}^{q \times n}$ and $A_2 \in \mathbb{R}^{(s-q) \times n}$ contain $a_l \in \mathbb{R}^n$ as $A_1 = [a_1, \dots, a_q]^T$ and $A_2 = [a_{q+1}, \dots, a_s]^T$. Further each $a_l = [a_{l1}^T, \dots, a_{lM}^T]^T$ where $a_{li} \in \mathbb{R}^{n_i}$. Further we have $B_1 \in \mathbb{R}^q$ and $B_2 \in \mathbb{R}^{s-q}$ where $B_1 = [b_1, \dots, b_q]^T$ and $B_2 = [b_{q+1}, \dots, b_s]^T$. We assume that the matrices A_1 and A_2 are sparse. By introducing the auxiliary variables y and the constraint $Px - p = y$ we get the following optimization problem

$$\begin{aligned} \min_{x, y} \quad & \frac{1}{2} x^T H x + g^T x + \gamma \|y\|_1 \\ \text{s.t.} \quad & A_1 x = B_1 \\ & A_2 x \leq B_2 \\ & P x - p = y. \end{aligned} \quad (3)$$

The objective of the optimization routine is to solve (3) in a distributed fashion using several computational units, where each computational unit computes the optimal local variables, denoted x_i^* , only. Each computational unit is assigned a number of constraints in (3) for which it is responsible. We denote the set of equality constraints that unit i is responsible for by \mathcal{L}_i^1 , the set of inequality constraints by \mathcal{L}_i^2 and the set of constraints originating from the 1-norm by \mathcal{R}_i . This division is obviously not unique but all constraints should be assigned to one computational unit. Further for $l \in \mathcal{L}_i^1$ and $l \in \mathcal{L}_i^2$ we require that $a_{li} \neq 0$ and for $r \in \mathcal{R}_i$ that $P_{ri} \neq 0$. Now we are ready to define two sets of neighbors to computational unit i

$$\begin{aligned} \mathcal{N}_i &= \{j \in \{1, \dots, M\} \mid \exists l \in \mathcal{L}_i^1 \text{ s.t. } a_{lj} \neq 0 \\ &\quad \text{or } \exists l \in \mathcal{L}_i^2 \text{ s.t. } a_{lj} \neq 0 \\ &\quad \text{or } \exists r \in \mathcal{R}_i \text{ s.t. } P_{rj} \neq 0\}, \\ \mathcal{M}_i &= \{j \in \{1, \dots, M\} \mid \exists l \in \mathcal{L}_j^1 \text{ s.t. } a_{li} \neq 0 \\ &\quad \text{or } \exists l \in \mathcal{L}_j^2 \text{ s.t. } a_{li} \neq 0 \\ &\quad \text{or } \exists r \in \mathcal{R}_j \text{ s.t. } P_{ri} \neq 0\}. \end{aligned}$$

Through the introduction of these sets, the constraints that are assigned to unit i can equivalently be written as

$$a_i^T x = b_l \iff \sum_{j \in \mathcal{N}_i} a_{lj}^T x_j = b_l, \quad l \in \mathcal{L}_i^1 \quad (4)$$

$$a_i^T x \leq b_l \iff \sum_{j \in \mathcal{N}_i} a_{lj}^T x_j \leq b_l, \quad l \in \mathcal{L}_i^2 \quad (5)$$

and the 1-norm term can equivalently be written as

$$|P_r^T x - p_r| = \left| \sum_{j \in \mathcal{N}_i} P_{rj}^T x_j - p_r \right|, \quad r \in \mathcal{R}_i. \quad (6)$$

In the following section, the dual function to be maximized is introduced. First, we state an assumption that will be useful in the continuation of the paper.

ASSUMPTION 1

We assume that there exists a vector \bar{x} such that $A_1 \bar{x} = b_1$ and $A_2 \bar{x} < b_2$. Further, we assume that $a_l, l = 1, \dots, q$ and $P_r, r = 1, \dots, m$ are linearly independent. \square

REMARK 1

Assumption 1 is known as the Mangasarian-Fromovitz constraint qualification (MFCQ). In [Gauvin, 1977] it was shown that MFCQ is equivalent to the set of optimal dual variables being bounded. For convex problems, MFCQ is equivalent to Slater's constraint qualification with the additional requirement that the vectors defining the equality constraints should be linearly independent. \square

3. Dual Problem

In this section we introduce a dual problem to (3) from which the primal solution can be obtained. We show that this dual problem has the properties required to apply accelerated gradient methods.

3.1 Formulation of the Dual Problem

We introduce Lagrange multipliers, $\lambda \in \mathbb{R}^q, \mu \in \mathbb{R}_{\geq 0}^{s-q}, \nu \in \mathbb{R}^m$ for the constraints in (3). Under Assumption 1 it is well known (cf. [Boyd and

Vandenberghe, 2004, §5.2.3]) that there is no duality gap and we get the following dual problem:

$$\sup_{\lambda, \mu \geq 0, \nu} \inf_{x, y} \left\{ \frac{1}{2} x^T H x + g^T x + \gamma \|y\|_1 + \lambda^T (A_1 x - B_1) + \mu^T (A_2 x - B_2) + \nu^T (P x - p - y) \right\}. \quad (7)$$

After rearranging the terms we get

$$\sup_{\lambda, \mu \geq 0, \nu} \left\{ \inf_x \left[(A_1^T \lambda + A_2^T \mu + P^T \nu + g)^T x + \frac{1}{2} x^T H x \right] - \lambda^T B_1 - \mu^T B_2 - \nu^T p + \inf_y [\gamma \|y\|_1 - \nu^T y] \right\}. \quad (8)$$

The infimum over y can be solved explicitly:

$$\begin{aligned} \inf_y \{ \gamma \|y\|_1 - \nu^T y \} &= \inf_y \left\{ \sum_i (\gamma |y_i| - [\nu]_i y_i) \right\} \\ &= \sum_i \left\{ \inf_{|y_i|} (\gamma |y_i| - [\nu]_i y_i) \right\} = \begin{cases} 0 & \text{if } \|\nu\|_\infty \leq \gamma \\ -\infty & \text{else} \end{cases} \end{aligned}$$

where $[\cdot]_i$ denotes the i -th element in the vector. The infimum over y becomes a box-constraint for the dual variables ν . This is a crucial observation for distribution reasons.

Before we explicitly solve the minimization over x in (8) the following notation is introduced:

$$\mathcal{A} = [A_1^T \ A_2^T \ P^T]^T, \quad \mathcal{B} = [B_1^T \ B_2^T \ p^T]^T, \quad z = [\lambda^T \ \mu^T \ \nu^T]^T$$

where $\mathcal{A} \in \mathbb{R}^{(s+m) \times n}$, $\mathcal{B} \in \mathbb{R}^{s+m}$ and $z \in \mathbb{R}^{s+m}$. We also introduce the set of feasible dual variables:

$$Z = \left\{ z \in \mathbb{R}^{s+m} \left| \begin{array}{ll} z_l \in \mathbb{R} & l \in \{1, \dots, q\} \\ z_l \geq 0 & l \in \{q+1, \dots, s\} \\ |z_l| \leq \gamma & l \in \{s+1, \dots, s+m\} \end{array} \right. \right\}. \quad (9)$$

The minimization over x in (8) can be solved explicitly:

$$\inf_x \left[(\mathcal{A}^T z + g)^T x + \frac{1}{2} x^T H x \right] = -\frac{1}{2} (\mathcal{A}^T z + g)^T H^{-1} (\mathcal{A}^T z + g)$$

and we get the following dual problem

$$\sup_{z \in Z} \left\{ -\frac{1}{2}(\mathcal{A}^T z + g)^T H^{-1}(\mathcal{A}^T z + g) - \mathcal{B}^T z \right\}. \quad (10)$$

We introduce the following definition of the negative dual function

$$f(z) := \frac{1}{2}(\mathcal{A}^T z + g)^T H^{-1}(\mathcal{A}^T z + g) + \mathcal{B}^T z.$$

Since f consists of a quadratic term with positive semidefinite hessian and a linear term, f is differentiable and has the following gradient

$$\nabla f(z) = \mathcal{A}H^{-1}(\mathcal{A}^T z + g) + \mathcal{B}. \quad (11)$$

Further, from the min-max theorem we have that the smallest Lipschitz constant, L , to ∇f is $L = \|\mathcal{A}H^{-1}\mathcal{A}^T\|_2$.

4. Distributed Optimization Algorithm

In this section we show how the accelerated gradient method can be used to distributively solve (3) by minimizing the negative dual function f . The accelerated proximal gradient method for problem (10) is defined by the following iteration as presented in [Tseng, 2008, Algorithm 2] and [Beck and Teboulle, 2009, Eq. 4.1-4.3]

$$v^k = z^k + \frac{k-1}{k+2}(z^k - z^{k-1}) \quad (12)$$

$$z^{k+1} = \mathcal{P}_Z \left(v^k - \frac{1}{L} \nabla f(v^k) \right) \quad (13)$$

where \mathcal{P}_Z is the Euclidean projection onto the set Z . Thus, the new iterate, z^{k+1} , is the previous iterate plus a step in the negative gradient direction projected onto the feasible set.

We define the primal iteration $x^k = H^{-1}(-\mathcal{A}^T z^k - g)$. Using this definition, straightforward insertion of v^k into (11) gives

$$\nabla f(v^k) = -\mathcal{A} \left(x^k + \frac{k-1}{k+2}(x^k - x^{k-1}) \right) + \mathcal{B}$$

By defining $\tilde{x}^k = x^k + \frac{k-1}{k+2}(x^k - x^{k-1})$ and recalling the partition $z = [\lambda^T \mu^T \nu^T]^T$ and the definition (9) of the set Z , we find that (12)-(13) can

be parallelized:

$$x^k = H^{-1}(-\mathcal{A}^T z^k - g) \quad (14)$$

$$\bar{x}^k = x^k + \frac{k-1}{k+2}(x^k - x^{k-1}) \quad (15)$$

$$\lambda_i^{k+1} = \lambda_i^k + \frac{k-1}{k+2}(\lambda_i^k - \lambda_i^{k-1}) + \frac{1}{L}(a_i^T \bar{x}^k - b_i) \quad (16)$$

$$\mu_i^{k+1} = \max \left\{ 0, \mu_i^k + \frac{k-1}{k+2}(\mu_i^k - \mu_i^{k-1}) + \frac{1}{L}(a_i^T \bar{x}^k - b_i) \right\} \quad (17)$$

$$v_r^{k+1} = \min \left\{ \gamma, \max \left[-\gamma, v_r^k + \frac{k-1}{k+2}(v_r^k - v_r^{k-1}) + \frac{1}{L}(P_r^T \bar{x}^k - p_r) \right] \right\}. \quad (18)$$

From these iterations it is not obvious that the algorithm is distributed. By partitioning the constraint matrix as

$$\mathcal{A} = [\mathcal{A}_1, \dots, \mathcal{A}_M]$$

where each $\mathcal{A}_i = [a_{1i}, \dots, a_{si}, P_{1i}, \dots, P_{mi}]^T \in \mathbb{R}^{(s+m) \times n_i}$, and noting that H is block-diagonal, the local primal variables are updated according to

$$\begin{aligned} x_i^k &= H_i^{-1}(-\mathcal{A}_i^T z^k - g_i) \\ &= -H_i^{-1} \left[g_i + \sum_{j \in \mathcal{M}_i} \left[\sum_{l \in \mathcal{L}_j^1} a_{li} \lambda_l^k + \sum_{l \in \mathcal{L}_j^2} a_{li} \mu_l^k + \sum_{r \in \mathcal{R}_j} P_{ri} v_r^k \right] \right]. \end{aligned} \quad (19)$$

Thus, each local primal update, x_i^k , can be computed after communication with neighbors $j \in \mathcal{M}_i$. Through (4)-(6) we note that the dual variable iterations can be updated after communication with neighbors $i \in \mathcal{N}_j$. We get the following distributed algorithm.

ALGORITHM 1—DISTRIBUTED ACCELERATED PROXIMAL GRADIENT ALGORITHM

Initialize $\lambda^0 = \lambda^{-1}, \mu^0 = \mu^{-1}, v^0 = v^{-1}$ and $x^0 = x^{-1}$

In every node, i , the following computations are performed:

For $k \geq 0$

1. Compute x_i^k according to (19) and set

$$\bar{x}_i^k = x_i^k + \frac{k-1}{k+2}(x_i^k - x_i^{k-1})$$

2. Send \bar{x}_i^k to each $j \in \mathcal{M}_i$, receive \bar{x}_j^k from each $j \in \mathcal{N}_i$

3. Compute λ_l^{k+1} according to (16), (4) for $l \in \mathcal{L}_i^1$
 Compute μ_l^{k+1} according to (17), (5) for $l \in \mathcal{L}_i^2$
 Compute v_r^{k+1} according to (18), (6) for $l \in \mathcal{R}_i$
4. Send $\{\lambda_l^{k+1}\}_{l \in \mathcal{L}_i^1}, \{\mu_l^{k+1}\}_{l \in \mathcal{L}_i^2}, \{v_r^{k+1}\}_{r \in \mathcal{R}_i}$ to each $j \in \mathcal{N}_i$,
 receive $\{\lambda_l^{k+1}\}_{l \in \mathcal{L}_j^1}, \{\mu_l^{k+1}\}_{l \in \mathcal{L}_j^2}$ and $\{v_r^{k+1}\}_{r \in \mathcal{R}_j}$ from each $j \in \mathcal{M}_i$

End □

The convergence rates for the dual function f and the primal variables when running Algorithm 1 are stated in the following theorem.

THEOREM 1

Algorithm 1 has the following convergence rate properties:

1. Denote an optimizer of the dual problem (10) as z^* . The convergence rate is:

$$f(z^k) - f(z^*) \leq \frac{2L\|z^0 - z^*\|_2^2}{(k+1)^2}, \forall k \geq 1 \quad (20)$$

2. Denote the unique optimizer of the primal problem as x^* . The rate of convergence for the primal variable is

$$\|x^k - x^*\|_2^2 \leq \frac{4L\|z^0 - z^*\|_2^2}{\underline{\sigma}(H)(k+1)^2}, \forall k \geq 1 \quad (21)$$

□

PROOF

Algorithm 1 is a distributed implementation of [Tseng, 2008, Algorithm 2] and [Beck and Teboulle, 2009, Eq. 4.1-4.3] applied to minimize f . The convergence rate in argument 1 follows from [Tseng, 2008, Proposition 2] and [Beck and Teboulle, 2009, Theorem 4.4].

For argument 2 we get that the necessary and sufficient KKT conditions [Boyd and Vandenberghe, 2004, p. 244] implies $x^* = H^{-1}(-\mathcal{A}^T z^* - g)$

since H is invertible. This leads to

$$\begin{aligned}
\|x^k - x^*\|_2^2 &= \|H^{-1}(\mathcal{A}^T z^k - \mathcal{A}^T z^*)\|_2^2 \\
&\leq \|H^{-1}\| \|\mathcal{A}^T z^k - \mathcal{A}^T z^*\|_{H^{-1}}^2 \\
&= \frac{1}{\underline{\sigma}(H)} (z^k - z^*)^T \mathcal{A} H^{-1} \mathcal{A}^T (z^k - z^*) \\
&= \frac{1}{\underline{\sigma}(H)} \left((z^k)^T \mathcal{A} H^{-1} \mathcal{A}^T z^k - (z^*)^T \mathcal{A} H^{-1} \mathcal{A}^T z^* - \right. \\
&\quad \left. - 2(\mathcal{A} H^{-1} \mathcal{A}^T z^*)^T (z^k - z^*) + \right. \\
&\quad \left. + 2(\mathcal{B} + \mathcal{A} H^{-1} g)^T (z^k - z^k + z^* - z^*) \right) \\
&= \frac{2}{\underline{\sigma}(H)} \left(f(z^k) - f(z^*) - (\mathcal{A} H^{-1} (\mathcal{A}^T z^* + g) + \mathcal{B})^T (z^k - z^*) \right) \\
&= \frac{2}{\underline{\sigma}(H)} (f(z^k) - f(z^*) - \nabla f(z^*)^T (z^k - z^*)) \\
&\leq \frac{2}{\underline{\sigma}(H)} (f(z^k) - f(z^*)) \leq \frac{4L \|z^0 - z^*\|_2^2}{\underline{\sigma}(H)(k+1)^2}
\end{aligned}$$

where the first inequality comes from the min-max theorem, the equalities are algebra with addition of some zero-terms, the first inequality in the final row is from the first-order optimality condition [Nesterov, 2003, Theorem 2.2.5], and the final inequality is due to (20). This completes the proof. \square

5. Numerical Example

In this section we evaluate the performance of Algorithm 1. We compare the presented algorithm to state-of-the-art centralized optimization software for large-scale optimization implemented in C, namely CPLEX and MOSEK. We also evaluate the performance loss when using suboptimal step sizes. Our algorithm is implemented on a single processor to be able to compare execution times.

The comparison is made on 100 random optimization problems arising in distributed MPC. A batch of random stable controllable dynamical systems with random structure and random initial conditions are created. The sparsity fraction, i.e., the fraction of non-zero elements in the dynamics matrix and the input matrix, is chosen to be 0.1. We have random inequality constraints that are generated to guarantee a feasible solution

Table 1. Algorithm comparison with 1-norm cost term and random state and input constraints. Algorithm 1 is implemented in MATLAB while CPLEX and MOSEK are implemented in C.

Alg.	vars./constr.	tol.	# iters		exec (ms)	
			mean	max	mean	max
1 (L)	4320/3231	0.005	69.8	160	253	609
1 (L_1)	4320/3231	0.005	160	420	594	1532
1 (L_F)	4320/3231	0.005	248	640	934	2444
MOSEK	4320/3231	-	-	-	1945	2674
CPLEX	4320/3231	0.005	-	-	1663	2832
1 (L)	2160/1647	0.005	63.8	100	94	200
1 (L_1)	2160/1647	0.005	75.8	180	115	368
1 (L_F)	2160/1647	0.005	121	320	185	488
MOSEK	2160/1647	-	-	-	334	399
CPLEX	2160/1647	0.005	-	-	282	522

and a 1-norm cost where the P -matrix and p -vector are randomly chosen. The quadratic cost matrices are chosen $Q = I$ and $R = I$. Table 1 shows the numerical results obtained running MATLAB on a Linux PC with a 3 GHz Intel Core i7 processor and 4 GB memory. The optimization software used is CPLEX V12.2 and MOSEK 6.0.0.114 that are accessed via the provided MATLAB interfaces.

The first column specifies the algorithm used where Algorithm 1 is supplemented with the step size used. L is the optimal step size $L = \|\mathcal{A}H^{-1}\mathcal{A}^T\|_2$, $L_F = \|\mathcal{A}H^{-1}\mathcal{A}^T\|_F$ and $L_1 = \sqrt{\|\mathcal{A}H^{-1}\mathcal{A}^T\|_1\|\mathcal{A}H^{-1}\mathcal{A}^T\|_\infty}$. We compare to the suboptimal step sizes L_1 and L_F since they can be computed in distributed fashion. The step sizes satisfy $L \leq L_1$ and $L \leq L_F$. The second column specifies the number of variables and constraints in the optimization problems. In the third column we have information about the duality gap tolerance that is used as stopping condition in the algorithms (if possible to set). The two final columns present the results in terms of number of iterations and execution time. The difference between the upper and lower halves of the table is the size of the problems that are solved.

Table 1 reveals that Algorithm 1 performs better than CPLEX and MOSEK on these large-scale sparse problems despite the fact that CPLEX and MOSEK are implemented in C and Algorithm 1 is implemented in MATLAB. We also conclude that the choice of step size in Algorithm 1 is important for performance reasons.

6. Conclusions

We have presented a distributed optimization algorithm for strongly convex optimization problems with sparse problem data. The algorithm is based on an accelerated gradient method that is applied to the dual problem. The algorithm was applied to large-scale sparse optimization problems originating from a distributed model predictive control formulation. Our algorithm performed better than state-of-the-art optimization software for large-scale sparse optimization, namely CPLEX and MOSEK, on these problems.

7. Acknowledgments

The second author would like to thank Quoc Tran Dinh for helpful discussions on the topic of this paper.

The second, third and fourth authors were supported by the European Union Seventh Framework STREP project “Hierarchical and distributed model predictive control (HD-MPC)”, contract number INFSO-ICT-223854, and the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 257462 HYCON2 Network of Excellence.

The first and last authors were supported by the Swedish Research Council through the Linnaeus center LCCC.

References

- Arrow, K., L. Hurwicz, and H. Uzawa (1958): *Studies in Linear and Nonlinear Programming*. Stanford University Press.
- Beck, A. and M. Teboulle (2009): “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” *SIAM J. Imaging Sciences*, **2:1**, pp. 183–202.
- Bertsekas, D. P. (1999): *Nonlinear Programming*, 2nd edition. Athena Scientific, Belmont, MA.
- Boyd, S. and L. Vandenberghe (2004): *Convex Optimization*. Cambridge University Press, New York, NY.
- Carpentier, P. and G. Cohen (1993): “Applied mathematics in water supply network management.” *Automatica*, **29:5**, pp. 1215–1250.
- Danzig, G. and P. Wolfe (1961): “The decomposition algorithm for linear programming.” *Econometrica*, **4**, pp. 767–778.
- Doan, M. D., T. Keviczky, and B. De Schutter (2011): “An iterative scheme for distributed model predictive control using Fenchel’s duality.” *Journal of Process Control*, **21:5**, pp. 746–755. Special Issue on Hierarchical and Distributed Model Predictive Control.
- Doan, M. D., T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter (2009): “A distributed version of Han’s method for DMPC using local communications only.” *Control Engineering and Applied Informatics*, **11:3**, pp. 6–15.
- Findeisen, W. (1980): *Control and Coordination in Hierarchical Systems*. International series on applied systems analysis. Wiley.
- Gauvin, J. (1977): “A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming.” *Mathematical Programming*, **12**, pp. 136–138.
- Giselsson, P. and A. Rantzer (2010): “Distributed model predictive control with suboptimality and stability guarantees.” In *Proceedings of the 49th Conference on Decision and Control*, pp. 7272–7277. Atlanta, GA.
- Kögel, M. and R. Findeisen (2011): “A fast gradient method for embedded linear predictive control.” In *Proceedings of the 18th IFAC World Congress*, pp. 1362–1367. Milan, Italy.
- Mesarovic, M. D., D. Macko, and Y. Takahara (1970): *Theory of Hierarchical Multilevel Systems*. Academic Press, New York.

- Necoara, I. and J. Suykens (2008): “Application of a smoothing technique to decomposition in convex optimization.” *IEEE Transactions on Automatic Control*, **53:11**, pp. 2674–2679.
- Negenborn, R., B. De Schutter, and J. Hellendoorn (2008): “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes.” *Engineering Applications of Artificial Intelligence*, **21:3**, pp. 353–366.
- Nemirovsky, A. and D. Yudin (1983): *Informational Complexity and Efficient Methods for Solution of Convex Extremal Problems*. Wiley, New York, NY.
- Nesterov, Y. (1983): “A method of solving a convex programming problem with convergence rate $O(1/k^2)$.” *Soviet Mathematics Doklady*, **27:2**, pp. 372–376.
- Nesterov, Y. (1988): “On an approach to the construction of optimal methods of minimization of smooth convex functions.” *Ékonom. i. Mat. Metody*, **24**, pp. 509–517.
- Nesterov, Y. (2003): *Introductory Lectures on Convex Optimization: A Basic Course*, 1st edition. Springer Netherlands.
- Nesterov, Y. (2005): “Smooth minimization of non-smooth functions.” *Math. Program.*, **103:1**, pp. 127–152.
- Richter, S., C. Jones, and M. Morari (2009): “Real-time input-constrained MPC using fast gradient methods.” In *Proceedings of the 48th Conference on Decision and Control*, pp. 7387–7393. Shanghai, China.
- Savorgnan, C., C. Romani, A. Kozma, and M. Diehl (2011): “Multiple shooting for distributed systems with applications in hydro electricity production.” *Journal of Process Control*, **21:5**, pp. 738–745.
- Singh, M. G. and A. Titli (1978): *Systems: Decomposition, Optimisation, and Control*. Pergamon.
- Tseng, P. (2008): “On accelerated proximal gradient methods for convex-concave optimization.” Submitted to SIAM J. Optim. Available: <http://www.math.washington.edu/~tseng/papers/apgm.pdf>.
- Tsitsiklis, J., D. Bertsekas, and M. Athans (1986): “Distributed asynchronous deterministic and stochastic gradient optimization algorithms.” *IEEE Transactions on Automatic Control*, **31:9**, pp. 803–812.

Paper II

Execution Time Certification for Gradient-Based Optimization in Model Predictive Control

Pontus Giselsson

Abstract

We consider model predictive control (MPC) problems with linear dynamics, polytopic constraints, and quadratic objective. The resulting optimization problem is solved by applying an accelerated gradient method to the dual problem. The focus of this paper is to provide bounds on the number of iterations needed in the algorithm to guarantee a prespecified accuracy of the dual function value and the primal variables as well as guaranteeing a prespecified maximal constraint violation. The provided numerical example shows that the iteration bounds are tight enough to be useful in an inverted pendulum application.

1. Introduction

Model predictive control (MPC) is an optimization based control methodology that can handle state and control constraints (see [Maciejowski, 2002, Mayne *et al.*, 2000] for thorough descriptions of MPC). In the optimization problem a cost function is minimized based on predicted future state and control trajectories and subject to constraints. Optimal control and state trajectories are obtained and the first element in the input trajectory is applied to the system. This procedure is repeated every sampling instant which sets requirements on the execution time of the optimization problem. The topic of this paper is to provide certificates for the execution time of the optimization algorithm such that for every feasible initial condition the optimization algorithm provides a solution within the sampling time. We consider linear time-invariant systems with polytopic constraints and quadratic cost and a dual accelerated gradient method [Giselsson *et al.*, 2012] is used to solve the resulting optimization problem.

For accelerated gradient methods there are convergence rate results [Nesterov, 2003, Beck and Teboulle, 2009, Tseng, 2008, Giselsson *et al.*, 2012] that depend explicitly on the norm of the difference between the optimal solution and the initial iterate. If this norm can be bounded, a bound on the number of iterations to achieve a prespecified accuracy of the function value can be computed. This was done in [Richter *et al.*, 2009] where input constrained MPC was considered. The condensed problem, i.e., the problem with all state variables eliminated, was solved using a fast gradient method. An iteration bound was obtained since the norm of the difference between the optimal solution and the initial iterate is bounded by the size of the input constraint set. Accelerated gradient methods can also be applied to the dual problem [Richter *et al.*, 2011, Giselsson *et al.*, 2012]. To compute a bound on the number of iterations to achieve a prespecified accuracy, a bound on the norm of the difference between the optimal dual variables and initial dual iterate is needed. This is more involved in the dual space than in a constrained primal space since dual variables are not chosen from a compact set. This is addressed in [Richter *et al.*, 2011] where the equality constraints are dualized and a bound on the norm of the optimal dual variables is obtained using a recent result in [Devolder *et al.*, 2011]. The obtained bounds turn out to be quite conservative. Another method to provide computation time certificates in MPC is to bound the search time in the look-up table in explicit MPC [Bemporad *et al.*, 2002, Alessio and Bemporad, 2009]. Practically this method is limited to small or medium-sized problems. For interior point methods, iteration bounds are available [McGovern, 2000], these are, however, reported to be quite conservative [Richter *et al.*, 2011, McGovern, 2000].

In this paper we consider the dual to the condensed problem, i.e., the dual to the problem where the state variables are eliminated. The resulting optimization problem has only inequality constraints and we apply the accelerated gradient method to the dual problem. To compute an iteration bound, we need a bound on the norm of the optimal dual variables. Using a result in [Nedic and Ozdaglar, 2009] a bound to this norm can be computed if a Slater vector to the optimization problem is known. Computation of the norm bound requires that the distance from equality in the inequality constraints for the Slater vector is known, as well as the primal cost for the Slater vector. We will see that such a Slater vector can be constructed for almost all feasible initial conditions in the MPC case. The provided numerical example shows that the presented bounds are tight enough to give useful bounds in an inverted pendulum application.

2. Problem Setup and Preliminaries

We consider the problem of controlling a linear dynamical system to the origin subject to polytopic constraints. To achieve this we use MPC in which the following finite horizon optimization problem is solved at the current state $\bar{x} \in \mathbb{R}^n$:

$$\begin{aligned}
 V_N(\bar{x}) := \min_{\mathbf{x}, \mathbf{u}} \quad & \frac{1}{2} \sum_{t=0}^{N-1} (x_t^T Q x_t + u_t^T R u_t) + \frac{1}{2} x_N^T Q_N x_N \quad (1) \\
 \text{s.t.} \quad & (x_t, u_t) \in \mathcal{X} \times \mathcal{U}, \quad t = 0, \dots, N-1 \\
 & x_{t+1} = A x_t + B u_t, \quad t = 0, \dots, N-1 \\
 & x_N \in \mathcal{X}_f, x_0 = \bar{x}
 \end{aligned}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, $\mathbf{x} = [x_1^T, \dots, x_N^T]^T$, and $\mathbf{u} = [u_0^T, \dots, u_{N-1}^T]^T$. We use the standard assumptions that $Q \succeq 0$, $Q_N \succeq 0$, and $R \succ 0$. The constraint sets are assumed to be polytopes

$$\begin{aligned}
 \mathcal{X} &= \{x \in \mathbb{R}^n \mid C_x x \leq d_x\}, & \mathcal{X}_f &= \{x \in \mathbb{R}^n \mid C_f x \leq d_f\}, \\
 \mathcal{U} &= \{u \in \mathbb{R}^m \mid C_u u \leq d_u\}.
 \end{aligned}$$

Throughout this paper we assume that the sets \mathcal{X} , \mathcal{X}_f , and \mathcal{U} are non-empty and compact and that $0 \in \text{int } \mathcal{X}$, $0 \in \text{int } \mathcal{X}_f$, and $0 \in \text{int } \mathcal{U}$ which

2. Problem Setup and Preliminaries

implies that $d_x, d_f, d_u > 0$. By introducing the following matrices

$$\mathbf{A} = \begin{pmatrix} A \\ \vdots \\ A^N \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} B & 0 & \cdots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A^{N-1}B & \cdots & AB & B \end{pmatrix}$$

the predicted future state variables can be described in the current state \bar{x} and control variables \mathbf{u} as

$$\mathbf{x} = \mathbf{A}\bar{x} + \mathbf{B}\mathbf{u}.$$

We further define

$$\begin{aligned} \mathbf{Q} &:= \text{blkdiag}(Q, \dots, Q, Q_N), & \mathbf{R} &:= \text{blkdiag}(R, \dots, R), \\ \mathbf{C}_x &:= \text{blkdiag}(C_x, \dots, C_x, C_f), & \mathbf{d}_x &:= [d_x^T, \dots, d_x^T, d_f^T]^T, \\ \mathbf{C}_u &:= \text{blkdiag}(C_u, \dots, C_u), & \mathbf{d}_u &:= [d_u^T, \dots, d_u^T]^T. \end{aligned}$$

The optimization problem (1) can, using these matrices, equivalently be written as

$$\begin{aligned} V_N(\bar{x}) = \min_{\mathbf{u}} \quad & J_N(\bar{x}, \mathbf{u}) := \frac{1}{2}\mathbf{u}^T \mathbf{H}\mathbf{u} + \bar{x}^T \mathbf{G}\mathbf{u} + \frac{1}{2}\bar{x}^T \mathbf{F}\bar{x} \\ \text{s.t.} \quad & g(\bar{x}, \mathbf{u}) \leq 0 \end{aligned} \quad (2)$$

where $H = \mathbf{B}^T \mathbf{Q} \mathbf{B} + \mathbf{R}$, $G = \mathbf{A}^T \mathbf{Q} \mathbf{B}$, $F = \mathbf{A}^T \mathbf{Q} \mathbf{A} + Q$, $g(\bar{x}, \mathbf{u}) = \mathbf{C}\mathbf{u} - d(\bar{x})$ and

$$C = \begin{pmatrix} \mathbf{C}_u \\ \mathbf{C}_x \mathbf{B} \end{pmatrix}, \quad d(\bar{x}) = \begin{pmatrix} \mathbf{d}_u \\ \mathbf{d}_x - \mathbf{C}_x \mathbf{A} \bar{x} \end{pmatrix}.$$

To solve (2) we introduce dual variables $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^p$ for the inequality constraints. The first $p_u \leq p$ dual variables in the dual variable vector $\boldsymbol{\mu}$ correspond to the input constraints and the last $p - p_u$ dual variables correspond to the state constraints. If Slater's condition holds, we get the following dual problem (cf. [Boyd and Vandenberghe, 2004])

$$V_N(\bar{x}) = \max_{\boldsymbol{\mu} \geq 0} \min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T \mathbf{H}\mathbf{u} + \bar{x}^T \mathbf{G}\mathbf{u} + \boldsymbol{\mu}^T (\mathbf{C}\mathbf{u} - d(\bar{x})).$$

As shown in [Giselsson *et al.*, 2012] the dual problem becomes

$$\max_{\boldsymbol{\mu} \geq 0} -\frac{1}{2}(\mathbf{C}^T \boldsymbol{\mu} + \mathbf{G}^T \bar{x})^T \mathbf{H}^{-1}(\mathbf{C}^T \boldsymbol{\mu} + \mathbf{G}^T \bar{x}) - \boldsymbol{\mu}^T d(\bar{x}). \quad (3)$$

We define the dual function

$$D_N(\bar{x}, \boldsymbol{\mu}) = -\frac{1}{2}(C^T \boldsymbol{\mu} + G^T \bar{x})^T H^{-1}(C^T \boldsymbol{\mu} + G^T \bar{x}) - \boldsymbol{\mu}^T d(\bar{x})$$

which satisfies the following properties (cf. [Giselsson *et al.*, 2012]).

PROPOSITION 1

The dual function has Lipschitz continuous gradient with Lipschitz constant $L = \|CH^{-1}C^T\|$ and the gradient is given by

$$\nabla D_N(\bar{x}, \boldsymbol{\mu}) = -CH^{-1}(C^T \boldsymbol{\mu} + G^T \bar{x}) - d(\bar{x}).$$

□

This implies that the dual function can be maximized using an accelerated gradient method [Nesterov, 2003, Beck and Teboulle, 2009, Tseng, 2008, Giselsson *et al.*, 2012]. The algorithm presented in [Giselsson *et al.*, 2012] with a cold-starting strategy, i.e., $\boldsymbol{\mu}^0 = \mathbf{0}$ is presented below.

ALGORITHM 1—ACCELERATED GRADIENT ALGORITHM

Initialize $\boldsymbol{\mu}^0 = \boldsymbol{\mu}^{-1} = \mathbf{0}$ and $\mathbf{u}^{-1} = -H^{-1}G^T \bar{x}$.

For $k \geq 0$

$$\begin{aligned} \mathbf{u}^k &= -H^{-1}(C^T \boldsymbol{\mu}^k + G^T \bar{x}) \\ \tilde{\mathbf{u}}^k &= \mathbf{u}^k + \frac{k-1}{k+2}(\mathbf{u}^k - \mathbf{u}^{k-1}) \\ \boldsymbol{\mu}^{k+1} &= \max \left\{ 0, \boldsymbol{\mu}^k + \frac{k-1}{k+2}(\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}) + \frac{1}{L}(C\tilde{\mathbf{u}}^k - d(\bar{x})) \right\} \end{aligned}$$

End

□

Before we state the convergence rate properties of the algorithm, we introduce the set of optimal dual variables

$$M^*(\bar{x}) = \{\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^p \mid D_N(\bar{x}, \boldsymbol{\mu}) \geq V_N(\bar{x})\}.$$

We also introduce \mathbb{X}_N which is the steerable set defined as

$$\mathbb{X}_N = \{\bar{x} \in \mathbb{R}^n \mid \text{there exist } \mathbf{u} \text{ s.t. } C\mathbf{u} \leq d(\bar{x})\}.$$

REMARK 1

From [Rawlings *et al.*, 2008], we know that the steerable set \mathbb{X}_N is convex and that $\mathbf{0} \in \mathbb{X}_N$. □

We also denote by $\mathbf{u}^*(\bar{x})$ the optimal solution to (2) with initial condition \bar{x} and $\underline{\sigma}(H)$ the smallest eigenvalue to H .

PROPOSITION 2

Suppose that $\bar{x} \in \mathbb{X}_N$. For any $\boldsymbol{\mu}^* \in M^*(\bar{x})$ Algorithm 1 has the following convergence rate properties:

1. The dual function converges as

$$D_N(\bar{x}, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\mu}^k) \leq \frac{2L\|\boldsymbol{\mu}^*\|^2}{(k+1)^2}, \forall k \geq 1. \quad (4)$$

2. The primal variable rate of convergence is

$$\|\mathbf{u}^k - \mathbf{u}^*(\bar{x})\|^2 \leq \frac{4L\|\boldsymbol{\mu}^*\|^2}{\underline{\sigma}(H)(k+1)^2}, \forall k \geq 1. \quad (5)$$

3. The constraint violation is bounded by

$$\|g(\bar{x}, \mathbf{u}^k) - g(\bar{x}, \mathbf{u}^*(\bar{x}))\|^2 \leq \frac{4L^2\|\boldsymbol{\mu}^*\|^2}{(k+1)^2}, \forall k \geq 1.$$

□

PROOF

Argument 1 is proven in [Beck and Teboulle, 2009, Tseng, 2008, Giselsson *et al.*, 2012] and argument 2 is proven in [Giselsson *et al.*, 2012]. To prove the third argument we have

$$\begin{aligned} \|g(\bar{x}, \mathbf{u}^k) - g(\bar{x}, \mathbf{u}^*(\bar{x}))\|^2 &= \|C\mathbf{u}^k - d(\bar{x}) - (C\mathbf{u}^*(\bar{x}) - d(\bar{x}))\|^2 \\ &= \|\nabla D_N(\bar{x}, \boldsymbol{\mu}^k) - \nabla D_N(\bar{x}, \boldsymbol{\mu}^*)\|^2 \\ &\leq 2L \left(-\langle -\nabla D_N(\bar{x}, \boldsymbol{\mu}^*), \boldsymbol{\mu}^k - \boldsymbol{\mu}^* \rangle + \right. \\ &\quad \left. + D_N(\bar{x}, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\mu}^k) \right) \\ &\leq 2L(D_N(\bar{x}, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\mu}^k)). \end{aligned}$$

The first inequality comes from [Nesterov, 2003, Theorem 2.1.5] since $-D_N$ is convex. The second inequality is due to first order optimality condition [Nesterov, 2003, Theorem 2.2.5] for the convex function $-D_N$. It is left to apply Argument 1 to prove the result. □

The objective of the paper is to, a priori, compute bounds on the number of iterations needed to achieve a prespecified dual function, primal variable, and constraint satisfaction tolerance when initializing the algorithm with $\boldsymbol{\mu}^0 = 0$. These bounds should ideally hold for any initial state $\bar{x} \in \mathbb{X}_N$. In

this paper we will show how to compute bounds that hold for any $\bar{x} \in \beta\mathbb{X}_N$ where $\beta \in (0, 1)$ and $\beta\mathbb{X}_N$ is defined as

$$\beta\mathbb{X}_N := \{\bar{x} \in \mathbb{R}^n \mid \frac{1}{\beta}\bar{x} \in \mathbb{X}_N\}.$$

From the definition and Remark 1 we conclude that $\beta\mathbb{X}_N \subseteq \mathbb{X}_N$ and that $0 \in \beta\mathbb{X}_N$. Before we proceed with the presentation we introduce the following definition.

DEFINITION 1

We define $\kappa \geq 1$ as the smallest scalar such that for every $\bar{x} \in \mathbb{X}_N$ the following holds

$$V_N(\bar{x}) \leq \kappa \min_{\mathbf{u}} J_N(\bar{x}, \mathbf{u}).$$

□

REMARK 2

The optimal solution to $\min_{\mathbf{u}} J_N(\bar{x}, \mathbf{u})$ is $\mathbf{u}_{uc}^*(\bar{x}) = -H^{-1}G^T\bar{x}$. The corresponding cost becomes

$$\begin{aligned} \min_{\mathbf{u}} J_N(\bar{x}, \mathbf{u}) &= \frac{1}{2}\bar{x}^T GH^{-1}G^T\bar{x} - \bar{x}^T GH^{-1}G^T\bar{x} + \frac{1}{2}\bar{x}^T F\bar{x} \\ &= \frac{1}{2}\bar{x}^T (F - GH^{-1}G^T)\bar{x}. \end{aligned}$$

By defining $P := F - GH^{-1}G^T$ where $P \succ 0$ we get

$$V_N(\bar{x}) \leq \kappa \min_{\mathbf{u}} J_N(\bar{x}, \mathbf{u}) = \frac{\kappa}{2}\bar{x}^T P\bar{x}.$$

Also, note that we have

$$V_N(\bar{x}) \geq \min_{\mathbf{u}} J_N(\bar{x}, \mathbf{u}) = \frac{1}{2}\bar{x}^T P\bar{x}.$$

□

2.1 Notation

We denote by \mathbb{R} the real numbers and by $\mathbb{R}_{\geq 0}$ non-negative real numbers. The norm $\|\cdot\|$ refers to the Euclidean norm or the induced Euclidean norm unless otherwise is specified and $\langle x, y \rangle = x^T y$. Further $\bar{\sigma}(H)$ denotes the largest singular value of H and $\underline{\sigma}(H)$ denotes the smallest singular value of H . Further $[\cdot]_i$ denotes the i :th element in the vector.

3. Lagrange Multiplier Norm Bounds

All quantities in the bounds in Proposition 2 are known except for $\|\boldsymbol{\mu}^*\|$ where $\boldsymbol{\mu}^* \in M^*(\bar{x})$. This section is devoted to bounding the norm of the optimal dual variables in (3) for any $\bar{x} \in \beta\mathbb{X}_N$ where $\beta \in (0, 1)$. The following result is used to achieve this.

LEMMA 1

Assume that $\bar{\mathbf{u}}(\bar{x})$ is a Slater vector, i.e., that $\bar{\mathbf{u}}(\bar{x})$ satisfies $C\bar{\mathbf{u}}(\bar{x}) < d(\bar{x})$. Then

$$\max_{\boldsymbol{\mu} \in M^*(\bar{x})} \|\boldsymbol{\mu}\| \leq \frac{1}{\gamma(\bar{x}, \bar{\mathbf{u}}(\bar{x}))} (J_N(\bar{x}, \bar{\mathbf{u}}(\bar{x})) - V_N(\bar{x}))$$

where $\gamma(\bar{x}, \bar{\mathbf{u}}(\bar{x})) := \min_{1 \leq j \leq p} [-g(\bar{x}, \bar{\mathbf{u}}(\bar{x}))]_j$. \square

PROOF

A proof is provided in [Nedic and Ozdaglar, 2009]. \square

Thus, if we can find a Slater vector for any initial condition $\bar{x} \in \beta\mathbb{X}_N$ we can bound the norm of the optimal Lagrange multipliers, $\boldsymbol{\mu}^*$. In the following lemma we show how to construct a Slater vector to (2) for any initial state $\bar{x} \in \beta\mathbb{X}_N$. Before we present the lemma the following notation is introduced; $\mathbf{d} := [\mathbf{d}_u^T, \mathbf{d}_x^T]^T$ and $d_{\min} := \min_j [\mathbf{d}]_j$ which implies that $d_{\min} > 0$.

LEMMA 2

For every $\bar{x} \in \beta\mathbb{X}_N$ with $\beta \in (0, 1)$, $\bar{\mathbf{u}}(\bar{x}) = \beta\mathbf{u}^*(\bar{x}/\beta)$ is a Slater vector to the optimization problem (2). The Slater vector satisfies $\gamma(\bar{x}, \bar{\mathbf{u}}(\bar{x})) \geq (1 - \beta)d_{\min}$. \square

PROOF

Since $\bar{x} \in \beta\mathbb{X}_N$ we have by definition that $\bar{x}/\beta \in \mathbb{X}_N$. The optimal control trajectory at \bar{x}/β is $\mathbf{u}^*(\bar{x}/\beta)$. Since $\bar{x}/\beta \in \mathbb{X}_N$ the optimal control trajectory is feasible, i.e., the following holds

$$g\left(\frac{\bar{x}}{\beta}, \mathbf{u}^*\left(\frac{\bar{x}}{\beta}\right)\right) = \begin{pmatrix} \mathbf{C}_u \mathbf{u}^*(\bar{x}/\beta) - \mathbf{d}_u \\ \mathbf{C}_x(\mathbf{A}\bar{x}/\beta + \mathbf{B}\mathbf{u}^*(\bar{x}/\beta)) - \mathbf{d}_x \end{pmatrix} \leq 0.$$

For any $\bar{x} \in \beta\mathbb{X}_N$ we have for the chosen Slater vector $\bar{\mathbf{u}}(\bar{x}) = \beta\mathbf{u}^*(\bar{x}/\beta)$

that

$$\begin{aligned}
 g(\bar{x}, \beta \mathbf{u}^*(\bar{x}/\beta)) &= \begin{pmatrix} \mathbf{C}_u \beta \mathbf{u}^*(\bar{x}/\beta) - \mathbf{d}_u \\ \mathbf{C}_x (\mathbf{A}\bar{x} + \mathbf{B}\beta \mathbf{u}^*(\bar{x}/\beta)) - \mathbf{d}_x \end{pmatrix} \\
 &= \begin{pmatrix} \beta (\mathbf{C}_u \mathbf{u}^*(\bar{x}/\beta) - \mathbf{d}_u) + (\beta - 1) \mathbf{d}_u \\ \beta (\mathbf{C}_x (\mathbf{A}\bar{x}/\beta + \mathbf{B}\mathbf{u}^*(\bar{x}/\beta)) - \mathbf{d}_x) + (\beta - 1) \mathbf{d}_x \end{pmatrix} \\
 &= \beta \begin{pmatrix} \mathbf{C}_u \mathbf{u}^*(\bar{x}/\beta) - \mathbf{d}_u \\ \mathbf{C}_x (\mathbf{A}\bar{x}/\beta + \mathbf{B}\mathbf{u}^*(\bar{x}/\beta)) - \mathbf{d}_x \end{pmatrix} - \begin{pmatrix} (1 - \beta) \mathbf{d}_u \\ (1 - \beta) \mathbf{d}_x \end{pmatrix} \\
 &\leq - \begin{pmatrix} (1 - \beta) \mathbf{d}_u \\ (1 - \beta) \mathbf{d}_x \end{pmatrix}.
 \end{aligned}$$

This gives

$$\gamma(\bar{x}, \bar{\mathbf{u}}(\bar{x})) = \min_{1 \leq j \leq p} [-g(\bar{x}, \beta \mathbf{u}^*(\bar{x}/\beta))]_j \geq (1 - \beta) \min([\mathbf{d}]_j) = (1 - \beta) d_{\min}.$$

This completes the proof. \square

By limiting the set of initial states, a Slater vector can be constructed with a certain distance to equality in the inequality constraints. Using this result the following theorem provides a bound on the norm of the optimal dual variables.

THEOREM 1

For every $\bar{x} \in \beta \mathbb{X}_N$ we have that

$$\max_{\boldsymbol{\mu} \in M^*(\bar{x})} \|\boldsymbol{\mu}\| \leq \frac{\kappa - 1}{2(1 - \beta) d_{\min}} \bar{x}^T P \bar{x}. \quad (6)$$

\square

PROOF

We will show that Lemma 1 gives (6) using the Slater vector $\bar{\mathbf{u}}(\bar{x}) = \beta \mathbf{u}^*(\bar{x}/\beta)$. We have

$$\begin{aligned}
 J_N(\bar{x}, \beta \mathbf{u}^*(\bar{x}/\beta)) &= \frac{1}{2} (\beta \mathbf{u}^*(\bar{x}/\beta))^T H \beta \mathbf{u}^*(\bar{x}/\beta) + \bar{x}^T G \beta \mathbf{u}^*(\bar{x}/\beta) + \frac{1}{2} \bar{x}^T F \bar{x} \\
 &= \frac{\beta^2}{2} \left((\mathbf{u}^*(\bar{x}/\beta))^T H \mathbf{u}^*(\bar{x}/\beta) + 2 \left[\frac{\bar{x}}{\beta} \right]^T G \mathbf{u}^*(\bar{x}/\beta) + \left[\frac{\bar{x}}{\beta} \right]^T F \left[\frac{\bar{x}}{\beta} \right] \right) \\
 &= \beta^2 V_N(\bar{x}/\beta) \leq \frac{\kappa}{2} \beta^2 \left[\frac{\bar{x}}{\beta} \right]^T P \left[\frac{\bar{x}}{\beta} \right] = \frac{\kappa}{2} \bar{x}^T P \bar{x}
 \end{aligned}$$

where the inequality comes from Remark 2. From Lemma 1 and Lemma 2 we have

$$\begin{aligned} \max_{\boldsymbol{\mu} \in M^*(\bar{x})} \|\boldsymbol{\mu}\| &\leq \frac{J_N(\bar{x}, \beta \mathbf{u}^*(\bar{x}/\beta)) - V_N(\bar{x})}{\gamma(\bar{x}, \beta \mathbf{u}^*(\bar{x}/\beta))} \leq \frac{1}{(1-\beta)d_{\min}} \left(\frac{\kappa}{2} \bar{x}^T P \bar{x} - V_N(\bar{x}) \right) \\ &\leq \frac{\kappa - 1}{2(1-\beta)d_{\min}} \bar{x}^T P \bar{x} \end{aligned}$$

where the last inequality is due to Remark 2. This completes the proof. \square

REMARK 3

If Definition 1 is changed such that κ_{β_1} is the smallest scalar such that for all $\bar{x} \in \beta_1 \mathbb{X}_N$ and for some $\beta_1 \in (0, 1)$ we have an upper bound $V_N(\bar{x}) \leq \frac{\kappa_{\beta_1}}{2} \bar{x}^T P \bar{x}$. Then for every $\bar{x} \in \beta_2 \mathbb{X}_N$ where $\beta_2 \in (0, \beta_1)$ it is straightforward to verify that

$$\max_{\boldsymbol{\mu} \in M^*(\bar{x})} \|\boldsymbol{\mu}\| \leq \frac{\kappa_{\beta_1} - 1}{2(1 - \beta_2/\beta_1)d_{\min}} \bar{x}^T P \bar{x}.$$

If $\frac{\kappa_{\beta_1} - 1}{1 - \beta_2/\beta_1} < \frac{\kappa - 1}{1 - \beta_2}$ we get an improved bound on the norm of the dual variables compared to Theorem 1. \square

The provided bound on the norm of optimal dual variables can, together with Proposition 2, be used to bound the number of iterations to get a pre-specified accuracy in the function value, primal variables and constraint violation. This is the topic of the following section.

4. Algorithm Iteration Bounds

In this section we provide bounds on the number of iterations within which a dual ϵ_d -solution, ϵ_c constraint violation and ϵ_p norm-distance to the primal optimal solution are guaranteed. The bounds are developed for the cold starting case, i.e., when the initial iterate is $\boldsymbol{\mu}^0 = 0$.

4.1 Iteration Bound to Guarantee Dual ϵ -solution

The first bound is on the number of iterations within which a dual ϵ -solution is guaranteed. To avoid that scaling the Q and R -matrices give different bounds we use a relative tolerance.

THEOREM 2

Suppose that Algorithm 1 is initialized with $\boldsymbol{\mu}^0 = 0$. Then for every $\bar{x} \in \beta \mathbb{X}_N$ with $\beta \in (0, 1)$ we have

$$V_N(\bar{x}) - D_N(\bar{x}, \boldsymbol{\mu}^k) \leq \epsilon_d V_N(\bar{x}) \quad (7)$$

if

$$k \geq k_d(\bar{x}) := \sqrt{\frac{L\bar{x}^T P\bar{x}}{\epsilon_d} \frac{\kappa - 1}{(1 - \beta)d_{\min}}} - 1.$$

□

PROOF

Inequality (7) is equivalent to

$$D_N(\bar{x}, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\mu}^k) \leq \epsilon_d D_N(\bar{x}, \boldsymbol{\mu}^*)$$

for any $\boldsymbol{\mu}^* \in M^*(\bar{x})$. From Proposition 2 and Theorem 1 we have that

$$D_N(\bar{x}, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\mu}^k) \leq \frac{2L\|\boldsymbol{\mu}^*\|^2}{(k+1)^2} \leq \frac{2L(\kappa-1)^2}{4(1-\beta)^2 d_{\min}^2 (k+1)^2} (\bar{x}^T P\bar{x})^2.$$

Since $\frac{1}{2}\bar{x}^T P\bar{x} \leq V_N(\bar{x})$, we have that

$$\frac{2L(\kappa-1)^2}{4(1-\beta)^2 d_{\min}^2 (k+1)^2} (\bar{x}^T P\bar{x})^2 \leq \epsilon_d \frac{1}{2}\bar{x}^T P\bar{x} \quad (8)$$

implies (7). Rearranging the terms gives the result. □

REMARK 4

By scaling the penalty-matrices $Q_a = aQ$, $R_a = aR$ we get $H_a = aH$ which implies $L_a = \|CH_a^{-1}C^T\| = \frac{1}{a}\|CH^{-1}C^T\| = \frac{1}{a}L$ and $P_a = aP$. Thus, using a relative tolerance the same bound is obtained for every scaling factor $a > 0$. □

REMARK 5

To get a bound that holds for all $\bar{x} \in \beta\mathbb{X}_N$, $k_d(\bar{x})$ should be maximized subject to $\bar{x} \in \beta\mathbb{X}_N$. An over-estimator is to maximize $k_d(\bar{x})$ subject to $\bar{x} \in \beta\mathcal{X}$ which is readily available. The resulting maximization problem depends affinely on $\sqrt{\bar{x}^T P\bar{x}}$, hence the maximizing argument can be found by maximizing $\bar{x}^T P\bar{x}$ on $\beta\mathcal{X}$. This is a quadratic maximization problem that can be rewritten as a mixed integer linear program (MILP) as shown in [Jones and Morari, 2009]. MILP software produce upper and lower bounds to the objective in each iteration and since an upper bound to the objective is enough to compute an iteration bound, the optimization can be stopped when sufficient accuracy is achieved. □

4.2 Iteration Bound for Constraint Violation

In this section we bound the number of iterations within which a pre-specified constraint violation is guaranteed. We use the following relative tolerance $g(\bar{x}, \mathbf{u}^k) \leq \epsilon \mathbf{d}$.

THEOREM 3

Suppose that Algorithm 1 is initialized with $\boldsymbol{\mu}^0 = 0$. Then, $g(\bar{x}, \mathbf{u}^k) \leq \epsilon_c \mathbf{d}$ holds for every $\bar{x} \in \beta \mathbb{X}_N$ if

$$k \geq k_c(\bar{x}) := \frac{L(\kappa - 1)\bar{x}^T P \bar{x}}{(1 - \beta)d_{\min}^2 \epsilon_c} - 1.$$

□

PROOF

First note that if $\|g(\bar{x}, \mathbf{u}^k) - g(\bar{x}, \mathbf{u}^*)\| \leq \epsilon_c d_{\min}$ then $g(\bar{x}, \mathbf{u}^k) \leq \epsilon_c \mathbf{d}$ since $g(\bar{x}, \mathbf{u}^*) \leq 0$. From Proposition 2 and Theorem 1 we get

$$\|g(\bar{x}, \mathbf{u}^k) - g(\bar{x}, \mathbf{u}^*(\bar{x}))\| \leq \frac{2L\|\boldsymbol{\mu}^*\|}{k + 1} \leq \frac{L(\kappa - 1)\bar{x}^T P \bar{x}}{(1 - \beta)d_{\min}(k + 1)}.$$

Setting this $\leq \epsilon_c d_{\min}$ and rearranging the terms gives the bound. □

REMARK 6

This result can be used in a constraint tightening approach to guarantee a feasible solution w.r.t. to the original constraint sets within $k_c(\bar{x})$ iterations. □

4.3 Primal Variable Iteration Bound

Using the same techniques it is also possible to bound the number of iterations needed to guarantee a primal solution that is within a prespecified distance to the optimal solution.

THEOREM 4

Suppose that Algorithm 1 is initialized with $\boldsymbol{\mu}^0 = 0$. Then, for every $\bar{x} \in \beta \mathbb{X}_N$ we have

$$\|\mathbf{u}^k - \mathbf{u}^*(\bar{x})\| \leq \epsilon_p$$

if

$$k \geq k_p(\bar{x}) := \sqrt{\frac{L}{\underline{\sigma}(H)} \frac{(\kappa - 1)\bar{x}^T P \bar{x}}{\epsilon_p(1 - \beta)d_{\min}}} - 1.$$

□

PROOF

From Proposition 2 and Theorem 1 we have

$$\|\mathbf{u}^k - \mathbf{u}^*(\bar{x})\| \leq \sqrt{\frac{L}{\underline{\sigma}(H)} \frac{2\|\boldsymbol{\mu}^*\|}{(k+1)}} \leq \sqrt{\frac{L}{\underline{\sigma}(H)} \frac{(\kappa-1)x^T P x}{(1-\beta)d_{\min}(k+1)}}.$$

Setting this $\leq \epsilon_p$ and rearranging gives the result. \square

5. Preconditioning

There are two different ways of preconditioning the problem data to possibly achieve smaller iteration bounds. One is to do a variable change in the primal variables and another is to scale the matrices defining the inequality constraints. We start by considering scaling the matrices defining the inequality constraints.

5.1 Scaling Inequality Constraints

All iteration bounds k_d, k_c, k_p depend on \sqrt{L}/d_{\min} or L/d_{\min}^2 . By introducing $D = \text{diag}(\mathbf{d})$ and recalling the definition of L we get $L/d_{\min}^2 = \|CH^{-1}C^T\|/\lambda_{\min}^2(D)$. By scaling the inequality constraints, this ratio can be minimized to get less conservative bounds without affecting the solutions of the optimization problem. We introduce the scaling matrix $S = \text{blkdiag}(S_u, S_x)$ where $S_u = \text{diag}(s_1, \dots, s_{p_u})$, $S_x = \text{diag}(s_{p_u+1}, \dots, s_p)$ where $p_u \leq p$ and all elements $s_i > 0, i = 1, \dots, p$. We get the following scaling

$$SC\mathbf{u} \leq Sd(\bar{x}).$$

From the definition of C and $d(\bar{x})$ we see that this is equivalent to

$$S_u \mathbf{C}_u \leq S_u \mathbf{d}_u \qquad S_x \mathbf{C}_x (\mathbf{A}\bar{x} + \mathbf{B}\mathbf{u}) \leq S_x \mathbf{d}_x.$$

The scaling of constraints will give as small bounds as possible if the scaling is chosen according to the following minimization

$$\min_S \frac{\|SCH^{-1}C^T S\|}{\lambda_{\min}^2(SD)}.$$

We introduce $\bar{S} = SD$ which is a diagonal matrix with strictly positive elements since it is a product of two diagonal matrices with strictly positive elements. This gives the equivalent minimization problem

$$\min_{\bar{S}} \frac{\|\bar{S}D^{-1}CH^{-1}C^T D^{-1}\bar{S}\|}{\lambda_{\min}^2(\bar{S})}. \tag{9}$$

It was shown in [Richter *et al.*, 2011, Lemma 1] that for invertible \bar{S} , an optimal solution is $\bar{S} = I$. Since diagonal matrices with positive elements are a subset of all invertible matrices, we get that $\bar{S} = I$ minimizes (9). The optimal scaling becomes $S = \bar{S}D^{-1} = D^{-1}$.

5.2 Preconditioning of Primal Variables

When performing a linear change of variables in primal variables, i.e., set $q = T^{-1}\mathbf{u}$ where T is an invertible matrix, H , G and C must be changed accordingly to not affect the primal optimal solution. We get $H_q = T^T H T$, $G_q = G T$ and $C_q = C T$. The Lipschitz constant does not change since

$$L_q = \|C_q H_q^{-1} C_q^T\| = \|C T T^{-1} H^{-1} T^{-T} T^T C^T\| = \|C H^{-1} C^T\| = L.$$

Straightforward verification of the algorithm when initialized with $\boldsymbol{\mu}^0 = 0$ gives that the $\boldsymbol{\mu}^k$ -sequence is identical whether using the new variables q or the original variables \mathbf{u} . It is also straightforward to verify that the relation between the iterates in the new variable q^k and the iterates in the original variable \mathbf{u}^k is $q^k = T^{-1}\mathbf{u}^k$. Thus, we do not get better (or worse) convergence properties by preconditioning the primal variables.

6. Numerical Example

We evaluate the conservatism of the iteration bounds by applying them to a double integrator system and a double integrator with a pendulum attached. We consider the pendulum in [Giselsson, 2011] with pendulum length $l = 0.4\text{m}$. The cart has inner control loops that make it behave as a double integrator. We choose sample time $h = 0.02\text{s}$ as in [Giselsson, 2011]. We get the following discrete time dynamics for the pendulum system when the pendulum is in its inverted position (cf. [Giselsson, 2011])

$$x(t+1) = \begin{pmatrix} 1 & 0.02 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1.0049 & 0.0200 \\ 0 & 0 & 0.4913 & 1.0049 \end{pmatrix} x(t) + \begin{pmatrix} 0.0002 \\ 0.02 \\ -0.0005 \\ -0.0501 \end{pmatrix} u(t).$$

The state variables are $x = [p \ \dot{p} \ \theta \ \dot{\theta}]^T$ where p is cart position, \dot{p} is cart velocity, θ is pendulum angle and $\dot{\theta}$ is pendulum angular velocity. The double integrator system is the system consisting of only the first two states, $[p \ \dot{p}]$. We have the following constraints

$$\begin{aligned} -0.5 \leq p \leq 0.5 & & -1 \leq \dot{p} \leq 1 & & -5 \leq u \leq 5 \\ -0.2 \leq \theta \leq 0.2 & & -0.5 \leq \dot{\theta} \leq 0.5 & & \end{aligned}$$

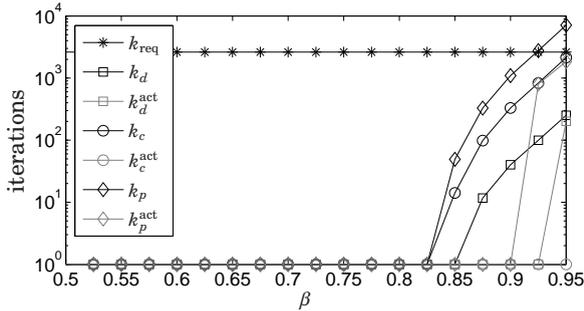


Figure 1. The figure shows the iteration bounds (k_d, k_c, k_p), the actual number of iterations ($k_d^{\text{act}}, k_c^{\text{act}}, k_p^{\text{act}}$), and the maximal number of iterations needed to certify execution time within $h=0.02\text{s}$ for the double integrator system (k_{req}).

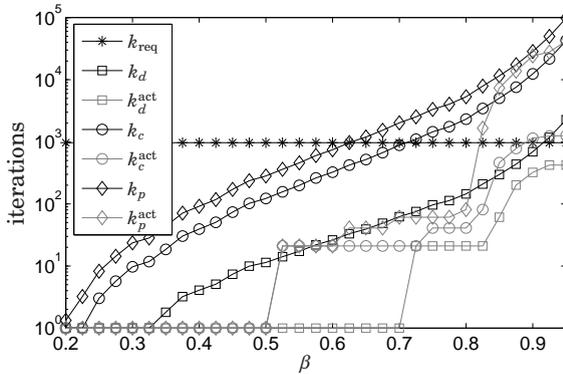


Figure 2. The figure shows the iteration bounds (k_d, k_c, k_p), the actual number of iterations ($k_d^{\text{act}}, k_c^{\text{act}}, k_p^{\text{act}}$), and the maximal number of iterations needed to certify execution time within $h=0.02\text{s}$ for the pendulum system (k_{req}).

The objective is to minimize

$$\sum_{t=0}^{N-1} (x_t^T Q x_t + u_t^T R u_t) + x_N^T P x_N$$

where $Q = \text{diag}(1, 0.3, 0.3, 0.1)$, $R = 0.1$ and P is the infinite horizon cost for the unconstrained LQ-problem with weighting matrices Q and R . Further we choose the terminal set $\mathcal{X}_f = \mathcal{X}$.

In Figures 1 and 2 we compare the iteration bounds with the worst case actual number of iterations and the maximum number of iterations, k_{req} , to guarantee an execution time less than $h = 0.02\text{s}$ on a machine with 1 Gflops/s computing power. If implemented wisely, the number of

flops per iteration in Algorithm 1 is $2(pN)^2 + 7pN$ and we get

$$k_{\text{req}} = \frac{10^9 h}{2(pN)^2 + 7pN}.$$

In all examples, we use control horizon $N = 10$, accuracy requirements $\epsilon_d = 0.01$ and $\epsilon_c = \epsilon_p = 0.05$. In Figure 1, the results for the double integrator are presented. On the x-axis β in $\beta\mathbb{X}_N$ is plotted and on the y-axis the iterations bounds and the actual number of iterations are plotted. We are able to certify that the optimization algorithm will terminate with a close to optimal solution for all $\bar{x} \in 0.925\mathbb{X}_N$ within the sampling time, $h = 0.02\text{s}$. We also see that for $\bar{x} \in 0.825\mathbb{X}_N$ we can guarantee that the optimal solution is found in one iteration, i.e., that no constraints are active.

In Figure 2, the results for the inverted pendulum system are presented. Also here we have β in $\beta\mathbb{X}_N$ on the x-axis and the iteration bounds and the actual number of iterations on the y-axis. We are able to certify that for $\bar{x} \in 0.6\mathbb{X}_N$ that the required accuracy is achieved within the sampling time, $h = 0.02\text{s}$. We can also certify that a dual ϵ_d -solution is found within the sampling time for any $\bar{x} \in 0.9\mathbb{X}_N$. We see that for large parts of the steerable set, \mathbb{X}_N , the iteration bounds give meaningful results that can be used to certify the MPC-controller with respect to execution time.

7. Conclusions and Future Work

We solve the optimization problems arising in MPC with linear dynamics, polytopic constraints, and a quadratic cost using a dual accelerated gradient method [Giselsson *et al.*, 2012]. By constructing Slater vectors to the optimization problems, we are able to bound the norm of the optimal dual variables. This is used to compute iteration bounds on the number of iterations within which a certain accuracy of the dual function value, constraint violation, and primal variables is guaranteed. The provided numerical example shows that the bounds are tight enough to be useful in a pendulum application. A future work direction is to search for tighter iteration bounds when using warm-starting strategies.

References

- Alessio, A. and A. Bemporad (2009): “A survey on explicit model predictive control.” In *Nonlinear Model Predictive Control*, pp. 345–369. Springer Berlin / Heidelberg.
- Beck, A. and M. Teboulle (2009): “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” *SIAM J. Imaging Sciences*, **2:1**, pp. 183–202.
- Bemporad, A., M. Morari, V. Dua, and E. Pistikopoulos (2002): “The explicit linear quadratic regulator for constrained systems.” *Automatica*, **38:1**, pp. 3–20.
- Boyd, S. and L. Vandenberghe (2004): *Convex Optimization*. Cambridge University Press, New York, NY.
- Devolder, O., F. Glineur, and Y. Nesterov (2011): “Double smoothing technique for convex optimization problems in Hilbert spaces with applications to optimal control.” Technical Report. Center for Operations Research and Econometrics (CORE), Universit catholique de Louvain (UCL), Belgium.
- Giselsson, P. (2011): “Model predictive control in a pendulum system.” In *Proceedings of the 31st IASTED Conference on Modelling, Identification and Control*. Innsbruck, Austria.
- Giselsson, P., M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer (2012): “Accelerated gradient methods and dual decomposition in distributed model predictive control.” *Automatica*. To appear. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Jones, C. and M. Morari (2009): “Approximate explicit MPC using bilevel optimization.” In *European Control Conference*. Budapest, Hungary.
- Maciejowski, J. M. (2002): *Predictive Control with Constraints*. Prentice Hall, Essex, England.
- Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert (2000): “Constrained model predictive control: Stability and optimality.” *Automatica*, **36:6**, pp. 789–814.
- McGovern, L. K. (2000): *Computational Analysis of Real-time Convex Optimization for Control Systems*. PhD thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics.
- Nedic, A. and A. E. Ozdaglar (2009): “Approximate primal solutions and rate analysis for dual subgradient methods.” *SIAM Journal on Optimization*, **19:4**, pp. 1757–1780.

- Nesterov, Y. (2003): *Introductory Lectures on Convex Optimization: A Basic Course*, 1st edition. Springer Netherlands.
- Rawlings, J., D. Bonne, J. Jorgensen, A. Venkat, and S. Jorgensen (2008): “Unreachable setpoints in model predictive control.” *IEEE Transactions on Automatic Control*, **53:9**, pp. 2209–2215.
- Richter, S., C. Jones, and M. Morari (2009): “Real-time input-constrained MPC using fast gradient methods.” In *Proceedings of the 48th Conference on Decision and Control*, pp. 7387–7393. Shanghai, China.
- Richter, S., M. Morari, and C. Jones (2011): “Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method.” In *Proceedings of the 50th IEEE Conference on Decision and Control*, pp. 5223–5229. Orlando, FL.
- Tseng, P. (2008): “On accelerated proximal gradient methods for convex-concave optimization.” Submitted to *SIAM J. Optim.* Available: <http://www.math.washington.edu/~tseng/papers/apgm.pdf>.

Paper III

Optimal Preconditioning and Iteration Complexity Bounds for Gradient-Based Optimization in MPC

Pontus Giselsson

Abstract

Model predictive control (MPC) formulations for systems with linear dynamics, polytopic constraints, and a quadratic cost are considered. The resulting quadratic program is solved using the accelerated gradient method presented in [Giselsson *et al.*, 2012]. Bounds on the number of iterations needed to ensure a prespecified tolerance of the dual function value and primal variables are provided. Further, we present an optimal preconditioning for the matrices describing the equality and inequality constraints, where optimal refers to the preconditioning that minimizes the iteration bound for dual function accuracy. A numerical example is provided which shows that the optimal preconditioning gives significantly less iterations in the algorithm, especially for ill-conditioned problem.

Submitted to *2013 American Control Conference*, Washington, D.C., 2013.

1. Introduction

Model predictive control (MPC) is an optimization based control methodology that can handle state and control constraints (for more on MPC, see [Maciejowski, 2002, Rawlings and Mayne, 2009]). In each time step a cost function is minimized based on predicted future state and control trajectories and subject to constraints. The first element in the obtained optimal input trajectory is applied to the system. This procedure is repeated in every sampling instant, which defines a feedback control law. A bound on the allowed execution time of the optimization algorithm is set by the sampling time. In this paper, methods to bound the execution time for the accelerated gradient method presented in [Giselsson *et al.*, 2012] when applied to MPC optimization problems are developed. The execution time bounds can be computed by bounding the number of iterations necessary to achieve a satisfactory solution and to evaluate the complexity within each iteration.

Convergence rate results for accelerated gradient methods are presented in [Nesterov, 2003, Beck and Teboulle, 2009, Tseng, 2008, Giselsson *et al.*, 2012]. The convergence rate depends on the norm of the difference between the optimal solution and the initial iterate. By bounding this norm, a bound on the number of iterations to achieve a prespecified accuracy of the function value can be obtained. In [Richter *et al.*, 2009] an accelerated gradient method was applied to the primal problem for input constrained MPC. An iteration bound was obtained by bounding the difference between the initial and optimal control trajectories. In [Richter *et al.*, 2011, Giselsson *et al.*, 2012] accelerated gradient methods was applied to the dual problem. To guarantee a prespecified accuracy of the dual function value, an upper bound to the norm of the difference between the optimal dual variables and initial dual iterate is needed. This is complicated by the fact that dual variables are not constrained to be in a compact set. In [Richter *et al.*, 2011] a recent result in [Devolder *et al.*, 2011] is used to bound the dual variables associated with the equality constraints. These bounds are used to compute a lower iteration bound to achieve a prespecified dual function accuracy. The bounds are reported to be quite conservative [Richter *et al.*, 2011]. Another method to provide computation time certificates in MPC is to bound the search time in the look-up table in explicit MPC [Bemporad *et al.*, 2002, Alessio and Bemporad, 2009]. Practically this method is limited to small or medium-sized problems. For interior point methods, iteration bounds are available [McGovern, 2000], these are, however, reported to be quite conservative [Richter *et al.*, 2011, McGovern, 2000].

In this paper the accelerated gradient method presented in [Giselsson *et al.*, 2012] is applied to an MPC problem with linear dynamics, polytopic

constraints and quadratic cost. The objective is to bound the number of iterations necessary to guarantee a prespecified accuracy of the dual function value and the primal variables. From the convergence rate results in [Giselsson *et al.*, 2012] it is clear that a bound on the norm of the optimal dual variables is needed to compute the iteration bounds. By extending and generalizing results from [Nedic and Ozdaglar, 2009] and [Giselsson, 2012] we show how to compute a bound to the norm of the optimal dual variables. These bounds are then used to compute the desired iteration bounds for dual function and primal variable accuracy. We also show how to precondition the matrices describing the inequality and equality constraints optimally, where optimally refers to the preconditioning that minimizes the iteration bound for dual function accuracy. A numerical example is provided that shows that the preconditioning decreases significantly the number of iteration needed to achieve a prespecified accuracy of the solution, especially for ill-conditioned problems.

The results in this paper are related to the results in [Giselsson, 2012]. They differ in that in this paper the problem with equality and inequality constraints is considered, while in [Giselsson, 2012] the condensed problem, i.e., the problem where the state variables are eliminated leaving only inequality constraints, is treated. This difference poses different technical challenges for the dual variable bounds and for optimal preconditioning.

The paper is organized as follows. In Section 2 the MPC optimization problem is stated and the solution algorithm from [Giselsson *et al.*, 2012] is presented. In Section 3 we present bounds on the norm of the optimal dual variables. Based on these, we present algorithm iteration bounds in Section 4. We show how to precondition the optimization data optimally in Section 5. In Section 6 a numerical example is provided and the paper is concluded in Section 7.

2. Problem Setup and Preliminaries

The following MPC optimization problem with initial condition $\bar{x} \in \mathbb{R}^n$ is considered

$$\begin{aligned}
 V_N(\bar{x}) := \min_{\mathbf{x}, \mathbf{u}} \quad & \frac{1}{2} \sum_{t=0}^{N-1} (x_t^T Q x_t + u_t^T R u_t) + \frac{1}{2} x_N^T Q_N x_N & (1) \\
 \text{s.t.} \quad & (x_t, u_t) \in \mathcal{X} \times \mathcal{U}, \quad t = 0, \dots, N-1 \\
 & x_{t+1} = A x_t + B u_t, \quad t = 0, \dots, N-1 \\
 & x_N \in \mathcal{X}_f, x_0 = \bar{x}
 \end{aligned}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, $\mathbf{x} = [x_1^T, \dots, x_N^T]^T$ and $\mathbf{u} = [u_0^T, \dots, u_{N-1}^T]^T$. The

2. Problem Setup and Preliminaries

cost matrices are assumed to satisfy $\mathbf{Q} \succ 0$, $\mathbf{Q}_N \succ 0$ and $\mathbf{R} \succ 0$ and the constraint sets are assumed to be polytopes defined by

$$\begin{aligned} \mathcal{X} &= \{x \in \mathbb{R}^n \mid C_x x \leq d_x\}, & \mathcal{X}_f &= \{x \in \mathbb{R}^n \mid C_f x \leq d_f\}, \\ \mathcal{U} &= \{u \in \mathbb{R}^m \mid C_u u \leq d_u\}, \end{aligned}$$

where $C_x \in \mathbb{R}^{n_x \times n}$, $C_u \in \mathbb{R}^{n_u \times m}$ and $C_f \in \mathbb{R}^{n_f \times n}$. We also assume that \mathcal{X} , \mathcal{X}_f and \mathcal{U} contain zero in their respective interiors which implies that $d_x, d_f, d_u > 0$. We also assume that the sets \mathcal{X} , \mathcal{X}_f and \mathcal{U} are compact. By stacking all decision variables into one vector, $\mathbf{y} = [x_0, \dots, x_N, u_0, \dots, u_{N-1}]$ and introducing the cost

$$J_N(\mathbf{y}) = \frac{1}{2} \mathbf{y}^T \mathbf{H} \mathbf{y}$$

where $\mathbf{H} \in \mathbb{R}^{(n+m)N+n \times (n+m)N+n}$ is chosen accordingly, the optimization problem (1) can more compactly be written as

$$\begin{aligned} V_N(\bar{x}) := \min_{\mathbf{y}} \quad & J_N(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{A} \mathbf{y} = \mathbf{b} \bar{x} \\ & \mathbf{C} \mathbf{y} \leq \mathbf{d} \end{aligned} \tag{2}$$

where matrices $\mathbf{A} \in \mathbb{R}^{nN \times (n+m)N+n}$, $\mathbf{b} \in \mathbb{R}^{nN \times n}$, $\mathbf{C} \in \mathbb{R}^{(n_x+n_u)N+n_f \times (n+m)N+n}$ and $\mathbf{d} \in \mathbb{R}^{(n_x+n_u)N+n_f}$ are built according to the introduced vector \mathbf{y} . Dual variables $\boldsymbol{\lambda} \in \mathbb{R}^{nN}$ for the equality constraints and $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{(n_x+n_u)N+n_f}$ for the inequality constraints are introduced. Under the assumption that Slater's condition holds, the following dual problem is obtained (cf. [Boyd and Vandenberghe, 2004])

$$V_N(\bar{x}) = \max_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} \min_{\mathbf{y}} \frac{1}{2} \mathbf{y}^T \mathbf{H} \mathbf{y} + \boldsymbol{\lambda}^T (\mathbf{A} \mathbf{y} - \mathbf{b} \bar{x}) + \boldsymbol{\mu}^T (\mathbf{C} \mathbf{y} - \mathbf{d}).$$

The dual problem can be rewritten as (cf. [Giselsson *et al.*, 2012])

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} -\frac{1}{2} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu})^T \mathbf{H}^{-1} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu}) - \boldsymbol{\lambda}^T \mathbf{b} \bar{x} - \boldsymbol{\mu}^T \mathbf{d}. \tag{3}$$

The dual function is defined as the maximand in the dual problem, i.e.,

$$D_N(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = -\frac{1}{2} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu})^T \mathbf{H}^{-1} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu}) - \boldsymbol{\lambda}^T \mathbf{b} \bar{x} - \boldsymbol{\mu}^T \mathbf{d}$$

and satisfies the properties stated in the following proposition (cf. [Giselsson *et al.*, 2012]).

PROPOSITION 1

The gradient of the dual function ∇D_N is Lipschitz continuous with constant $L = \|\mathbf{A}^T \mathbf{C}^T\|^T \mathbf{H}^{-1} \mathbf{A}^T \mathbf{C}^T\|$. The gradient w.r.t. $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are given by

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}} D_N(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= -\mathbf{A} \mathbf{H}^{-1} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu}) - \mathbf{b} \bar{x}, \\ \nabla_{\boldsymbol{\mu}} D_N(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= -\mathbf{C} \mathbf{H}^{-1} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu}) - \mathbf{d}\end{aligned}$$

respectively. □

These properties are such that an accelerated gradient method [Nesterov, 2003, Beck and Teboulle, 2009, Tseng, 2008, Giselsson *et al.*, 2012] can be used to solve the dual problem. Below, a cold-starting variant, i.e., with $\boldsymbol{\lambda}^0 = 0$ and $\boldsymbol{\mu}^0 = 0$, of the algorithm in [Giselsson *et al.*, 2012] is presented.

ALGORITHM 1—ACCELERATED GRADIENT ALGORITHM

Initialize $\boldsymbol{\lambda}^0 = \boldsymbol{\lambda}^{-1} = 0$, $\boldsymbol{\mu}^0 = \boldsymbol{\mu}^{-1} = 0$ and $\mathbf{y}^{-1} = 0$.

For $k \geq 0$

$$\begin{aligned}\mathbf{y}^k &= -\mathbf{H}^{-1} (\mathbf{A}^T \boldsymbol{\lambda}^k + \mathbf{C}^T \boldsymbol{\mu}^k) \\ \tilde{\mathbf{y}}^k &= \mathbf{y}^k + \frac{k-1}{k+2} (\mathbf{y}^k - \mathbf{y}^{k-1}) \\ \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \frac{k-1}{k+2} (\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) + \frac{1}{L} (\mathbf{A} \tilde{\mathbf{y}}^k - \mathbf{b} \bar{x}) \\ \boldsymbol{\mu}^{k+1} &= \max \left\{ 0, \boldsymbol{\mu}^k + \frac{k-1}{k+2} (\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}) + \frac{1}{L} (\mathbf{C} \tilde{\mathbf{y}}^k - \mathbf{d}) \right\}\end{aligned}$$

End □

The set of optimal dual variables is denoted by

$$M^*(\bar{x}) = \left\{ \boldsymbol{\lambda} \in \mathbb{R}^{nN}, \boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{(n_x+n_u)N+n_f} \mid D_N(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \geq V_N(\bar{x}) \right\}.$$

The set of initial conditions for which (2) is feasible is denoted by \mathbb{X}_N . The optimal solution to (2) with initial condition $\bar{x} \in \mathbb{X}_N$ is denoted by $\mathbf{y}^*(\bar{x})$. Next, we state the convergence rate properties of Algorithm 1.

PROPOSITION 2

Suppose that $\bar{x} \in \mathbb{X}_N$. For every $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$, Algorithm 1 has the following convergence rate properties:

1. For all $k \geq 1$ the dual function converges as

$$D_N(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) - D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \leq \frac{2L}{(k+1)^2} \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\|^2. \quad (4)$$

2. The primal variable rate of convergence is

$$\|\mathbf{y}^k - \mathbf{y}^*(\bar{x})\|^2 \leq \frac{4L}{\underline{\sigma}(\mathbf{H})(k+1)^2} \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\|^2, \forall k \geq 1 \quad (5)$$

where $\underline{\sigma}(\mathbf{H})$ denotes the smallest eigenvalue to \mathbf{H} .

□

PROOF

Argument 1 is proven in [Beck and Teboulle, 2009, Tseng, 2008, Giselsson *et al.*, 2012] and argument 2 is proven in [Giselsson *et al.*, 2012]. □

To compute lower iteration bounds for the cold starting case, i.e., with $\boldsymbol{\lambda}^0 = 0$ and $\boldsymbol{\mu}^0 = 0$, to guarantee a prespecified dual function value or primal variable accuracy is the objective of this paper. We will show how to compute bounds that hold for every $\bar{x} \in \beta\mathbb{X}_N$ with $\beta \in (0, 1)$ and where $\beta\mathbb{X}_N$ is defined as

$$\beta\mathbb{X}_N := \{\bar{x} \in \mathbb{R}^n \mid \frac{1}{\beta}\bar{x} \in \mathbb{X}_N\}.$$

Since the set \mathbb{X}_N is convex and $0 \in \mathbb{X}_N$ (cf. [Rawlings *et al.*, 2008]) we have that $\beta\mathbb{X}_N \subseteq \mathbb{X}_N$ and that $0 \in \beta\mathbb{X}_N$. Before we proceed with the presentation we introduce

$$P := \mathbf{b}^T (\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)^{-1} \mathbf{b}$$

which characterizes the optimal solution with only equality constraints and satisfies

$$\frac{1}{2}\bar{x}^T P \bar{x} = \max_{\boldsymbol{\lambda}} D_N(\bar{x}, \boldsymbol{\lambda}, 0) \leq V_N(\bar{x}) \quad (6)$$

We also make the following definition.

DEFINITION 1

The scalar $\kappa \geq 1$ is defined as the smallest scalar such that for every $\bar{x} \in \mathbb{X}_N$ we have

$$V_N(\bar{x}) \leq \frac{\kappa}{2}\bar{x}^T P \bar{x}.$$

□

Finally, we make the following assumption.

ASSUMPTION 1

We assume that \mathbf{A} has full row rank.

□

2.1 Notation

The real line is denoted by \mathbb{R} and the non-negative real numbers are denoted by $\mathbb{R}_{\geq 0}$. The Euclidean norm and the induced Euclidean norm are denoted by $\|\cdot\|$ and $\langle x, y \rangle = x^T y$. Further, $\bar{\sigma}(H)$ and $\underline{\sigma}(H)$ are the largest and smallest singular value of H respectively. Finally, the i :th element in a vector is denoted by $[\cdot]_i$.

3. Lagrange Multiplier Norm Bounds

The only unknown quantity in the bounds in Proposition 2 is the norm of the optimal dual variables. The topic of this section is to show how such norms can be computed for any initial condition $\bar{x} \in \beta \mathbb{X}_N$ with $\beta \in (0, 1)$. The following result is a straightforward generalization of the result in [Nedic and Ozdaglar, 2009, Lemma 1].

LEMMA 1

Assume that there exists a vector $\bar{\mathbf{y}}(\bar{x})$ such that $\mathbf{C}\bar{\mathbf{y}}(\bar{x}) < \mathbf{d}$ and $\mathbf{A}\bar{\mathbf{y}}(\bar{x}) = \mathbf{b}\bar{x}$. Then for every $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$ we have that $\boldsymbol{\mu}^*$ satisfies

$$\|\boldsymbol{\mu}^*\| \leq \frac{1}{\gamma(\bar{\mathbf{y}}(\bar{x}))} (J_N(\bar{\mathbf{y}}(\bar{x})) - V_N(\bar{x}))$$

where $\gamma(\bar{\mathbf{y}}(\bar{x})) := \min_{1 \leq j \leq (n_x + n_u)N + n_f} -[\mathbf{C}\bar{\mathbf{y}}(\bar{x}) - \mathbf{d}]_j$. □

PROOF

For every $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$ we have

$$\begin{aligned} V_N(\bar{x}) &= \inf_{\mathbf{y}} J_N(\mathbf{y}) + (\boldsymbol{\lambda}^*)^T (\mathbf{A}\mathbf{y} - \mathbf{b}\bar{x}) + (\boldsymbol{\mu}^*)^T (\mathbf{C}\mathbf{y} - \mathbf{d}) \\ &\leq J_N(\bar{\mathbf{y}}(\bar{x})) + (\boldsymbol{\lambda}^*)^T (\mathbf{A}\bar{\mathbf{y}}(\bar{x}) - \mathbf{b}\bar{x}) + (\boldsymbol{\mu}^*)^T (\mathbf{C}\bar{\mathbf{y}}(\bar{x}) - \mathbf{d}) \\ &\leq J_N(\bar{\mathbf{y}}(\bar{x})) - \gamma(\bar{\mathbf{y}}(\bar{x})) (\boldsymbol{\mu}^*)^T \mathbf{1} \\ &= J_N(\bar{\mathbf{y}}(\bar{x})) - \gamma(\bar{\mathbf{y}}(\bar{x})) \|\boldsymbol{\mu}^*\|_1 \\ &\leq J_N(\bar{\mathbf{y}}(\bar{x})) - \gamma(\bar{\mathbf{y}}(\bar{x})) \|\boldsymbol{\mu}^*\| \end{aligned}$$

Rearranging the terms gives the result. □

By constructing a strictly feasible vector, referred to as a Slater vector, a bound on the norm of the optimal Lagrange multipliers associated with the inequality constraints can be computed. Next, a straightforward generalization to [Giselsson, 2012, Lemma 2] is presented where it was shown how a Slater vector to (2) for every initial state $\bar{x} \in \beta \mathbb{X}_N$ can be constructed. Before the lemma is presented we introduce $d_{\min} := \min_j [\mathbf{d}]_j > 0$.

3. Lagrange Multiplier Norm Bounds

LEMMA 2

For every $\bar{x} \in \beta\mathbb{X}_N$ with $\beta \in (0, 1)$, a Slater vector to the optimization problem (2) is given by $\bar{\mathbf{y}}(\bar{x}) = \beta\mathbf{y}^*(\bar{x}/\beta)$. Further, $\gamma(\bar{\mathbf{y}}(\bar{x})) \geq (1 - \beta)d_{\min}$. \square

PROOF

We first note that

$$\mathbf{A}\bar{\mathbf{y}}(\bar{x}) = \beta\mathbf{A}\mathbf{y}^*\left(\frac{\bar{x}}{\beta}\right) = \beta\mathbf{b}\frac{\bar{x}}{\beta} = \mathbf{b}\bar{x}$$

which implies that the equality constraints are satisfied. Further

$$\mathbf{C}\bar{\mathbf{y}}(\bar{x}) = \beta\mathbf{C}\mathbf{y}^*\left(\frac{\bar{x}}{\beta}\right) \leq \beta\mathbf{d} = \mathbf{d} - (1 - \beta)\mathbf{d}$$

Hence $-(\mathbf{C}\bar{\mathbf{y}}(\bar{x}) - \mathbf{d}) \geq (1 - \beta)\mathbf{d}$ which by definition of the function γ and d_{\min} gives the result. \square

Next, we present a theorem that, using Lemma 1 and Lemma 2, shows how a bound on the norm of the optimal dual variables can be computed. Before we present the theorem, we introduce the matrices

$$\Phi := \mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T, \quad \Psi := (\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{H}^{-1}\mathbf{C}^T. \quad (7)$$

By Assumption 1 \mathbf{A} has full row rank and \mathbf{H} is positive definite, hence $\Phi = \mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T$ is invertible and Ψ exists.

THEOREM 1

For every $\bar{x} \in \beta\mathbb{X}_N$ we have for every $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$ that

$$\left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} \Psi \\ I \end{bmatrix} \right\| \frac{\kappa - 1}{2(1 - \beta)d_{\min}} \bar{x}^T P \bar{x} + \|\Phi^{-1}\mathbf{b}\bar{x}\|. \quad (8)$$

\square

PROOF

Using the Slater vector $\bar{\mathbf{y}}(\bar{x}) = \beta\mathbf{y}^*(\bar{x}/\beta)$ we get

$$\begin{aligned} J_N(\bar{\mathbf{y}}(\bar{x})) &= J_N(\beta\mathbf{y}^*\left(\frac{\bar{x}}{\beta}\right)) = \beta^2 \frac{1}{2} \mathbf{y}^*\left(\frac{\bar{x}}{\beta}\right)^T \mathbf{H} \mathbf{y}^*\left(\frac{\bar{x}}{\beta}\right) \\ &= \beta^2 V_N\left(\frac{\bar{x}}{\beta}\right) \leq \beta^2 \frac{\kappa}{2} \left[\frac{\bar{x}}{\beta} \right]^T P \left[\frac{\bar{x}}{\beta} \right] = \frac{\kappa}{2} \bar{x}^T P \bar{x} \end{aligned}$$

where the inequality comes from Definition 1. Further, KKT conditions to (2) and Proposition 1 gives that for every $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$ we have

$$-\mathbf{A}\mathbf{H}^{-1}(\mathbf{A}^T \boldsymbol{\lambda}^* + \mathbf{C}^T \boldsymbol{\mu}^*) = \mathbf{b}\bar{x}.$$

This implies that

$$\boldsymbol{\lambda}^* = -(\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)^{-1}(\mathbf{A}\mathbf{H}^{-1}\mathbf{C}^T\boldsymbol{\mu}^* + \mathbf{b}\bar{x}) = -\Psi\boldsymbol{\mu}^* - \Phi^{-1}\mathbf{b}\bar{x}$$

where the last equality comes from the definitions of Φ and Ψ in (7). This gives

$$\begin{aligned} \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\| &= \left\| \begin{bmatrix} -\Psi \\ I \end{bmatrix} \boldsymbol{\mu}^* + \begin{bmatrix} -\Phi^{-1}\mathbf{b}\bar{x} \\ 0 \end{bmatrix} \right\| \\ &\leq \left\| \begin{bmatrix} \Psi \\ I \end{bmatrix} \right\| \|\boldsymbol{\mu}^*\| + \|\Phi^{-1}\mathbf{b}\bar{x}\| \\ &\leq \left\| \begin{bmatrix} \Psi \\ I \end{bmatrix} \right\| \frac{1}{\gamma(\bar{\mathbf{y}}(\bar{x}))} (J_N(\bar{\mathbf{y}}(\bar{x})) - V_N(\bar{x})) + \|\Phi^{-1}\mathbf{b}\bar{x}\| \\ &\leq \left\| \begin{bmatrix} \Psi \\ I \end{bmatrix} \right\| \frac{\kappa - 1}{2(1 - \beta)d_{\min}} \bar{x}^T P \bar{x} + \|\Phi^{-1}\mathbf{b}\bar{x}\| \end{aligned}$$

where the second inequality comes from Lemma 1 and the final inequality from Lemma 2, Definition 1 and (6). This completes the proof. \square

In the following section, the bound on the optimal dual variables is used, together with the convergence rate results in Proposition 2, to compute lower iteration bounds to achieve a prespecified dual function value and primal variable accuracy.

4. Algorithm Iteration Bounds

Lower iteration bounds to achieve prespecified dual function value and primal variable tolerances are presented in this section. We consider bounds for the cold starting case, i.e., when $\boldsymbol{\lambda}^0 = 0$ and $\boldsymbol{\mu}^0 = 0$.

4.1 Iteration Bound to Guarantee Dual ϵ -solution

First, a lower iteration bound to achieve a prespecified dual function value accuracy is presented. As in [Giselsson, 2012] a relative tolerance is used to avoid that a scaling of the Q and R matrices affects the bound.

THEOREM 2

Suppose that Algorithm 1 is initialized with $\boldsymbol{\lambda}^0 = 0$ and $\boldsymbol{\mu}^0 = 0$. Then for every $\bar{x} \in \beta\mathbb{X}_N$ with $\beta \in (0, 1)$ we have

$$V_N(\bar{x}) - D(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \leq \epsilon_d V_N(\bar{x}) \quad (9)$$

for every $k \geq k_d(\bar{x})$ where

$$k_d(\bar{x}) = 2\sqrt{\frac{L}{\epsilon_d}} \left(\frac{(\kappa - 1)\sqrt{\bar{x}^T P \bar{x}}}{2(1 - \beta)} \nu + \rho \right) - 1 \quad (10)$$

and $\rho = \|\Phi^{-1} \mathbf{b} P^{-1/2}\|$ and $\nu = \|[\Psi^T \ I^T]^T\|/d_{\min}$. \square

PROOF

Inequality (9) is equivalent to

$$D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \leq \epsilon_d D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$$

for any $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$. From Proposition 2 and (6) we conclude that (9) holds if

$$\frac{2L}{(k+1)^2} \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\|^2 \leq \epsilon_d \frac{1}{2} \bar{x}^T P \bar{x}. \quad (11)$$

Insertion of the bound in Theorem 1 into (11) and rearranging the terms gives

$$k_d(\bar{x}) = 2\sqrt{\frac{L}{\epsilon_d}} \left(\frac{(\kappa - 1)\sqrt{\bar{x}^T P \bar{x}}}{2(1 - \beta)} \nu + \frac{\|\Phi^{-1} \mathbf{b} \bar{x}\|}{\sqrt{\bar{x}^T P \bar{x}}} \right) - 1.$$

We have

$$\frac{\|\Phi^{-1} \mathbf{b} \bar{x}\|}{\sqrt{\bar{x}^T P \bar{x}}} \leq \rho \iff \frac{\|\Phi^{-1} \mathbf{b} \bar{x}\|^2}{\|P^{1/2} \bar{x}\|^2} \leq \rho^2 \iff \bar{x}^T \mathbf{b}^T \Phi^{-2} \mathbf{b} \bar{x} \leq \rho^2 \bar{x}^T P \bar{x}.$$

Since $0 \in \text{int}(\beta \mathbb{X}_N)$ this holds for every $\bar{x} \in \beta \mathbb{X}_N$ if and only if ρ is such that

$$\mathbf{b}^T \Phi^{-2} \mathbf{b} \leq \rho^2 P \iff P^{-1/2} \mathbf{b}^T \Phi^{-2} \mathbf{b} P^{-1/2} \leq \rho^2 I \iff \|\Phi^{-1} \mathbf{b} P^{-1/2}\| \leq \rho. \quad (12)$$

Choosing ρ such that the last step in (12) holds with equality completes the proof. \square

REMARK 1

As in [Giselsson, 2012] the lower iteration bound is not affected by scaling the cost matrices by a factor $a > 0$. This is true since for cost matrices $Q_a = aQ$ and $R_a = aR$ we get $L_a = \frac{1}{a}L$, $P_a = aP$, $\rho_a = \sqrt{a}\rho$, and $\nu_a = \nu$. By insertion into (10) the factor a is cancelled. \square

REMARK 2

It is desirable to compute a lower iteration bound for all $\bar{x} \in \beta\mathbb{X}_N$. By maximizing $k_d(\bar{x})$ subject to $\bar{x} \in \beta\mathbb{X}_N$ this can be obtained. Since it is often difficult to describe the set $\bar{x} \in \beta\mathbb{X}_N$ an over estimator to the lower iteration bound is found by maximizing $k_d(\bar{x})$ subject to $\bar{x} \in \beta\mathcal{X}$. The only \bar{x} -dependency in the iteration bound is $\sqrt{\bar{x}^T P \bar{x}}$ which affects the bound affinely. By maximizing $\bar{x}^T P \bar{x}$ over $\beta\mathcal{X}$, which is a quadratic convex maximization problem, the maximizing \bar{x} can be found. Such problems are NP-complete but can be rewritten as mixed integer linear programs (MILP) as shown in [Jones and Morari, 2009, Lemma 1]. There are efficient MILP-solvers that in every iteration produce upper and lower bounds to the optimal value. An upper bound to the optimal value is enough to compute an iteration bound, hence the MILP solver can be stopped when sufficient accuracy has been reached. \square

4.2 Primal Variable Iteration Bound

In this section, a bound on the number of iterations needed to guarantee a prespecified distance between the algorithm primal variables and the optimal primal variables is presented.

THEOREM 3

Suppose that Algorithm 1 is initialized with $\lambda^0 = 0$ and $\mu^0 = 0$. Then for every $\bar{x} \in \beta\mathbb{X}_N$ we have

$$\|\mathbf{y}^k - \mathbf{y}^*(\bar{x})\| \leq \epsilon_p \tag{13}$$

for every $k \geq k_p(\bar{x})$ where

$$k_p(\bar{x}) = \frac{2}{\epsilon_p} \sqrt{\frac{L}{\underline{\sigma}(\mathbf{H})}} \left(\left\| \begin{bmatrix} \Psi \\ I \end{bmatrix} \right\| \left\| \frac{(\kappa - 1)\bar{x}^T P \bar{x}}{2(1 - \beta)d_{\min}} + \|\Phi^{-1}\mathbf{b}\bar{x}\| \right\| \right) - 1.$$

\square

PROOF

From Proposition 2 we have that (13) holds if

$$\frac{2}{k + 1} \sqrt{\frac{L}{\underline{\sigma}(\mathbf{H})}} \left\| \begin{bmatrix} \lambda^* \\ \mu^* \end{bmatrix} \right\| \leq \epsilon_p$$

Insertion of the bound in Theorem 1 and rearranging gives the result. \square

5. Preconditioning

In this section we focus on how to precondition the problem data to get improved iteration bounds. We precondition the equality constraints with an invertible matrix E such that $E\mathbf{A}\mathbf{y} = E\mathbf{b}\bar{x}$ and the inequality constraints with a diagonal matrix F with positive diagonal elements such that $FC\mathbf{y} \leq F\mathbf{d}$. To keep the sparse structure of the equality constraints, E should satisfy $E \in \mathcal{E}$ where \mathcal{E} defines the sparsity structure. We assume that \mathcal{E} is such at least all diagonal elements may be non-zero. We note that the only terms in (10) that are affected by the preconditioning are L , ν , and ρ . In the following lemma we show that the optimal preconditioner for the inequality constraint has the form $F = tD^{-1}$ where $t > 0$ and $D := \text{diag}(\mathbf{d})$.

LEMMA 3

Let the Lipschitz constant to ∇D_N be bounded by L . Then the optimal preconditioner for the inequality constraints satisfies $F = tD^{-1}$ for some $t > 0$ where optimal refers to the preconditioners that minimize the iteration bound in Theorem 2. \square

PROOF

Since F and D are diagonal matrices with positive elements, F can be represented as $F = GD^{-1}$ where G is a diagonal matrix with positive elements. The variables in the iteration bound in Theorem 2 that are affected by the preconditioning are ρ , ν and L . For preconditioners E and $F = GD^{-1}$, ρ satisfies

$$\begin{aligned} \rho &= \|(EA\mathbf{H}^{-1}\mathbf{A}^T E^T)^{-1} E\mathbf{b}P^{-1/2}\| = \|E^{-T}(\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)^{-1} E^{-1} E\mathbf{b}P^{-1/2}\| \\ &= \|E^{-T}\Phi^{-1}\mathbf{b}P^{-1/2}\| \end{aligned} \quad (14)$$

and ν satisfies

$$\begin{aligned} \nu &= \left\| \begin{bmatrix} (EA\mathbf{H}^{-1}\mathbf{A}^T E^T)^{-1} EA\mathbf{H}^{-1}\mathbf{C}^T D^{-T} G^T \\ I \end{bmatrix} \right\| / d_{\min} \\ &= \left\| \begin{bmatrix} E^{-T}(\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)^{-1} E^{-1} EA\mathbf{H}^{-1}\mathbf{C}^T D^{-T} G^T \\ I \end{bmatrix} \right\| / \min_j [GD^{-1}\mathbf{d}]_j \\ &= \left\| \begin{bmatrix} E^{-T}\Psi D^{-T} G^T \\ I \end{bmatrix} \right\| / \lambda_{\min}(G) \end{aligned} \quad (15)$$

where Φ and Ψ are defined in (7). Further, since the Lipschitz constant to ∇D_N should be bounded by L for all feasible E and $F = GD^{-1}$, they

must satisfy (see Proposition 1)

$$L \geq \left\| \left[\begin{array}{c} \mathbf{EA} \\ \mathbf{GD}^{-1}\mathbf{C} \end{array} \right] \mathbf{H}^{-1} \left[\begin{array}{c} \mathbf{EA} \\ \mathbf{GD}^{-1}\mathbf{C} \end{array} \right]^T \right\|. \quad (16)$$

Next, we show that the optimal F satisfies $F = tD^{-1}$ where $t \geq 0$, i.e., that the optimal $G = tI$. We represent G as $G = tI + \tilde{G}$ where \tilde{G} is diagonal and $\tilde{G} \succeq 0$. This implies that (15) is equivalent to ν being the smallest scalar such that

$$\mathbf{GD}^{-1}\Psi^T \mathbf{E}^{-1} \mathbf{E}^{-T} \Psi \mathbf{D}^{-T} \mathbf{G}^T + I \preceq \nu^2 t^2 I$$

which in turn is equivalent to ν being the smallest scalar such that

$$\begin{aligned} (\nu^2 t^2 - 1)I &\succeq \mathbf{E}^{-T} \Psi \mathbf{D}^{-T} \mathbf{G}^T \mathbf{GD}^{-1} \Psi^T \mathbf{E}^{-1} \\ &= \mathbf{E}^{-T} \Psi \mathbf{D}^{-T} (t^2 I + 2\tilde{G}t + \tilde{G}^T \tilde{G}) \mathbf{D}^{-1} \Psi^T \mathbf{E}^{-1}. \end{aligned}$$

Hence, for given $t > 0$, $\tilde{G} = 0$ gives the smallest ν independent of E . Further, the Lipschitz constant constraint (16) is equivalent to

$$LI \succeq \mathbf{H}^{-1/2} [\mathbf{A}^T \mathbf{E}^T, \mathbf{C}^T \mathbf{D}^{-T} \mathbf{G}^T] [\mathbf{A}^T \mathbf{E}^T, \mathbf{C}^T \mathbf{D}^{-T} \mathbf{G}^T]^T \mathbf{H}^{-1/2}$$

which in turn is equivalent to

$$\begin{aligned} \mathbf{LH} &\succeq \mathbf{A}^T \mathbf{E}^T \mathbf{EA} + \mathbf{C}^T \mathbf{D}^{-T} \mathbf{G}^T \mathbf{GD}^{-1} \mathbf{C} \\ &= \mathbf{A}^T \mathbf{E}^T \mathbf{EA} + \mathbf{C}^T \mathbf{D}^{-T} (t^2 I + 2\tilde{G}t + \tilde{G}^T \tilde{G}) \mathbf{D}^{-1} \mathbf{C}. \end{aligned}$$

We introduce $X(\tilde{G}) = \mathbf{LH} - \mathbf{C}^T \mathbf{D}^{-T} (t^2 I + 2\tilde{G}t + \tilde{G}^T \tilde{G}) \mathbf{D}^{-1} \mathbf{C}$ which satisfies $X(\tilde{G}_1) \prec X(\tilde{G}_2)$ if $\tilde{G}_1 \succ \tilde{G}_2 \succeq 0$. We also define the set of feasible preconditioners for the equality constraints

$$\Sigma(\tilde{G}) = \{E \in \mathcal{E} \mid \mathbf{A}^T \mathbf{E}^T \mathbf{EA} \preceq X(\tilde{G})\}.$$

For $E \in \Sigma(0)$ we have $\mathbf{A}^T \mathbf{E}^T \mathbf{EA} \preceq X(0)$ and for $E \in \Sigma(\tilde{G})$ with $\tilde{G} \succ 0$ we have $\mathbf{A}^T \mathbf{E}^T \mathbf{EA} \preceq X(\tilde{G}) \prec X(0)$, which implies $\Sigma(\tilde{G}) \subset \Sigma(0)$ for every $\tilde{G} \succ 0$. Hence, by setting $\tilde{G} = 0$ the variable ν is as small as possible and the set of feasible E is as large as possible. This implies that if $\tilde{G} = 0$ then E gets maximal freedom in minimizing (14) and (15) while respecting the constraint (16). This concludes the proof. \square

Before we state the theorem about how to compute the optimal preconditioner we define $P_x := \max_{x \in \mathbb{X}} \sqrt{x^T P x}$. This implies $\max_{x \in \beta \mathbb{X}} \sqrt{x^T P x} = \beta P_x$.

THEOREM 4

Let the Lipschitz constant to ∇D_N be bounded by L . Then the preconditioners E and F that minimize the iteration bound in Theorem 2 for fixed $\beta \in (0, 1)$ are found by solving the following semidefinite program

$$\min \frac{(\kappa - 1)\beta P_x}{2(1 - \beta)} v + \rho$$

$$\text{s.t. } \begin{bmatrix} Z & \theta \Phi^{-1} \mathbf{b} \\ \theta \mathbf{b}^T \Phi^{-T} & P \end{bmatrix} \succeq 0 \quad (17)$$

$$\begin{bmatrix} Z & \phi \Psi D^{-T} \\ \phi D^{-1} \Psi^T & (1 - s)I \end{bmatrix} \succeq 0 \quad (18)$$

$$\begin{bmatrix} sI & \phi D^{-1} \mathbf{C} \\ \phi \mathbf{C}^T D^{-T} & L\mathbf{H} - \mathbf{A}^T \mathbf{Z} \mathbf{A} \end{bmatrix} \succeq 0 \quad (19)$$

$$\begin{bmatrix} v & 1 \\ 1 & \phi \end{bmatrix} \succeq 0 \quad (20)$$

$$\begin{bmatrix} \rho & 1 \\ 1 & \theta \end{bmatrix} \succeq 0 \quad (21)$$

$$Z \succ 0, Z \in \mathcal{E}$$

$$\rho > 0, \theta > 0, \phi > 0, v > 0, s > 0$$

where $Z = E^T E$, $s = \frac{\phi^2}{t^2}$ and $F = tD^{-1}$. Further, the optimal value, i.e., the lower iteration bound, is the same for any choice of $L > 0$. \square

PROOF

The variables in the iteration bound in Theorem 2 that are affected by the preconditioning are ρ , v and L . Variables ρ and v are given by (14) and (15) respectively when preconditioning is used, and the constraint imposed by L is given in (16) in the preconditioning case. We will show that the posed semidefinite program implies (14), (15), and (16) and chooses the preconditioners that minimize the iteration bound in Theorem 2.

Schur complement of (21) gives $\rho \geq 1/\theta$. Further, Schur complement gives that (17) and $Z \succ 0$ implies

$$\mathbf{b}^T \Phi^{-T} Z^{-1} \Phi^{-1} \mathbf{b} \leq \frac{1}{\theta^2} P \leq \rho^2 P.$$

Hence

$$\|E^{-T} \Phi^{-1} \mathbf{b} P^{-1/2}\| \leq \rho \quad (22)$$

and (14) is implied by choosing the smallest ρ such that (22) holds. Schur complement of (20) gives $v \geq 1/\phi$ and Schur complement of (18) gives

$$D^{-1} \Psi^T Z^{-1} \Psi D^{-T} \leq \frac{1}{\phi^2} (1 - s)I \leq (v^2 - \frac{1}{t^2})I.$$

This is equivalent to

$$tD^{-1}\Psi^T Z^{-1}\Psi D^{-T}t + I \preceq v^2 t^2 I = v^2 d_{\min}^2 I$$

which in turn is equivalent to

$$\left\| \begin{bmatrix} E^{-T}\Psi D^{-T}t \\ I \end{bmatrix} \right\| \frac{1}{d_{\min}} \preceq v. \quad (23)$$

By choosing the smallest v such that (23) holds, (15) is satisfied since by Lemma 3, $G = tI$ is optimal and since $\lambda_{\min}(G) = d_{\min}$. Finally, since $s > 0$ Schur complement of (19) gives

$$LH \succeq \mathbf{A}^T Z \mathbf{A} + \mathbf{C}^T D^{-T} \frac{\phi^2}{s} D^{-1} \mathbf{C} = \mathbf{A}^T Z \mathbf{A} + \mathbf{C}^T D^{-T} t^2 D^{-1} \mathbf{C}$$

which is equivalent to

$$L \geq \|\mathbf{H}^{-1/2}[\mathbf{A}^T E^T, \mathbf{C}^T D^{-T}t][\mathbf{A}^T E^T, \mathbf{C}^T D^{-T}t]^T \mathbf{H}^{-1/2}\|$$

which in turn is equivalent to

$$L \geq \left\| \begin{bmatrix} \mathbf{E} \mathbf{A} \\ tD^{-1}\mathbf{C} \end{bmatrix} \mathbf{H}^{-1} \begin{bmatrix} \mathbf{E} \mathbf{A} \\ tD^{-1}\mathbf{C} \end{bmatrix}^T \right\|. \quad (24)$$

This implies that the Lipschitz constant constraint (16) holds. Next we show that the cost

$$\min \frac{(\kappa - 1)\beta P_x}{2(1 - \beta)} v + \rho \quad (25)$$

implies that (24) holds with equality which implies that the iteration bound (10) is minimized. Since $t = d_{\min}$ the bound (23) is equivalent to

$$\left\| \begin{bmatrix} E^{-T}\Psi D^{-T} \\ 1/t \end{bmatrix} \right\| \preceq v.$$

Further, (22) depends on E^{-1} . This implies that ρ and v are decreasing when E and t are increasing, while the r.h.s. of (24) is increasing with E and t . Hence, the optimal preconditioners must have equality in (24). Further, since (24) holds with equality, the Lipschitz constant L in (10) is fixed and (25) minimizes the iteration bound (10).

It remains to show that the iteration bound (10) is independent of the choice of $L > 0$. The iteration bound (10) depends on $\sqrt{L}\rho$ and $\sqrt{L}v$ and the only hard constraints are (24) and the constraints on positive

definiteness. We introduce the set of feasible preconditioners for fixed L as follows

$$\Theta(L) = \{t > 0, E \in \mathcal{E} \mid E \succ 0 \text{ and (24) holds}\}.$$

Since \mathcal{E} is only a sparsity constraint we get for any $L_1 > 0$ and $L_2 > 0$ that $\sqrt{L_2}\Theta(L_1) = \sqrt{L_1}\Theta(L_2)$. Hence every pair $(t_2, E_2) \in \Theta(L_2)$ can be described as $(t_2, E_2) = \sqrt{L_2}(t_1, E_1)$ where $(t_1, E_1) \in \Theta(L_1)$ and $L_1 = 1$. We denote by ρ_1 and v_1 the bounds (22) and (23) using t_1 and E_1 and by ρ_2 and v_2 the bound (22) and (23) using t_2 and E_2 . We see from (22) that using $E_2 = \sqrt{L_2}E_1$ and $t_2 = \sqrt{L_2}t_1$ gives $\rho_2 = \rho_1/\sqrt{L_2}$ and from (23) we conclude that $v_2 = v_1/\sqrt{L_2}$. Since the iteration bound depends on $\sqrt{L}\rho$ and $\sqrt{L}v$ we get $\sqrt{L_2}\rho_2 = \sqrt{L_2}\rho_1/\sqrt{L_2} = \rho_1 = \sqrt{L_1}\rho_1$ and $\sqrt{L_2}v_2 = \sqrt{L_2}v_1/\sqrt{L_2} = v_1 = \sqrt{L_1}v_1$. Hence, the choice of L does not influence the iteration bound. This completes the proof. \square

REMARK 3

Since Z is symmetric and positive definite it can be decomposed as $Z = U\Sigma U^T$ where U is unitary and Σ is diagonal with positive diagonal elements. Setting $E = U\Sigma^{1/2}U^T$ gives that if $Z \in \mathcal{E}$ then also $E \in \mathcal{E}$ and $E^T E = U\Sigma^{1/2}U^T U\Sigma^{1/2}U^T = U\Sigma U^T = Z$. The preconditioning matrix F is readily computed by setting $F = \frac{\phi}{\sqrt{s}}D^{-1}$. \square

6. Numerical Example

The efficiency of the preconditioning and the conservatism of the iteration bound are evaluated by applying the optimization algorithm on a DMPC problem where the dynamics matrix is randomly generated and has sparse structure. The system is unstable since the largest eigenvalue of the dynamics matrix is 1.1. The system has 3 sub-systems with 5 states and 1 input each, i.e., in total 15 states and 3 inputs. The state and input variables are upper and lower bounded by random numbers in the intervals $[0.5 \ 1.5]$ and $[-0.15 \ -0.05]$ respectively. We evaluate the preconditioning and the iteration bound on two different choices of cost-matrices. The first choice is $Q_1 = I$, $R_1 = I$ and in the second choice, the cost matrices Q_2 and R_2 are diagonal and each diagonal element is randomly chosen from the interval $[1 \ 100]$. The control horizon is $N = 6$. All simulations are performed in MATLAB and the semidefinite program for the preconditioning is solved through YALMIP [Löfberg, 2004] using SeDuMi [Sturm, 1999].

In Table 1 we compare the number of iterations needed to achieve a certain dual accuracy using for the randomly generated DMPC problem

Table 1. Experimental results for Algorithm 1 with and without preconditioning. The number of algorithm iterations and iteration complexity bounds are presented.

Cost	ϵ_v	β	precond	# iters		iter bound
				avg.	max.	
Q_1, R_1	0.005	0.25	y	7.44	19	220
Q_1, R_1	0.005	0.25	n	36.41	57	1014
Q_2, R_2	0.005	0.25	y	9.57	68	484
Q_2, R_2	0.005	0.25	n	191.15	343	5087
Q_1, R_1	0.005	0.5	y	10.65	28	496
Q_1, R_1	0.005	0.5	n	40.26	82	2449
Q_2, R_2	0.005	0.5	y	23.26	123	1333
Q_2, R_2	0.005	0.5	n	201.41	505	12463
Q_1, R_1	0.005	0.75	y	13.22	39	1326
Q_1, R_1	0.005	0.75	n	46.12	128	6752
Q_2, R_2	0.005	0.75	y	30.17	155	3878
Q_2, R_2	0.005	0.75	n	214.52	624	34585

with and without preconditioning for the two choices of cost matrices. We also compare the actual number of iterations with the iteration bounds to evaluate the conservatism of the bounds. The first column specifies which cost matrices that are used. The second column specifies the duality tolerance and the third column specifies the set from which the initial conditions are chosen where β is the scaling factor, i.e., initial conditions are chosen from $\beta\mathbb{X}_N$. The fourth column specifies if preconditioning is used or not. The fifth and sixth columns present average and max number of iterations while the seventh column presents the iteration bound. The data in Table 1 is obtained by solving the DMPC optimization problem for 10000 randomly generated initial conditions.

In Table 1 we see that the preconditioning reduces significantly the number of iterations needed to achieve a relative dual accuracy of 0.005. This holds especially when using cost-matrices Q_2, R_2 which result in more ill-conditioned problems than using cost-matrices Q_1, R_1 . Further, the conservatism of the iteration bound is about one to two orders of magnitude.

7. Conclusions and Future Work

We solve the optimization problems arising in MPC with linear dynamics, polytopic constraints and quadratic cost using a dual accelerated gradient

method [Giselsson *et al.*, 2012]. We have presented iteration bounds that guarantee a prespecified dual function value and primal variable accuracy. The iteration bounds are used to create an optimal preconditioning where optimal refers to the preconditioning that minimizes the iteration bound for the dual function value. The provided numerical example shows that the resulting preconditioning can reduce significantly the number of iterations needed to achieve the desired accuracy of the dual function, especially for ill-conditioned problems.

References

- Alessio, A. and A. Bemporad (2009): “A survey on explicit model predictive control.” In *Nonlinear Model Predictive Control*, pp. 345–369. Springer Berlin / Heidelberg.
- Beck, A. and M. Teboulle (2009): “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” *SIAM J. Imaging Sciences*, **2:1**, pp. 183–202.
- Bemporad, A., M. Morari, V. Dua, and E. Pistikopoulos (2002): “The explicit linear quadratic regulator for constrained systems.” *Automatica*, **38:1**, pp. 3–20.
- Boyd, S. and L. Vandenberghe (2004): *Convex Optimization*. Cambridge University Press, New York, NY.
- Devolder, O., F. Glineur, and Y. Nesterov (2011): “Double smoothing technique for convex optimization problems in Hilbert spaces with applications to optimal control.” Technical Report. Center for Operations Research and Econometrics (CORE), Universit catholique de Louvain (UCL), Belgium.
- Giselsson, P. (2012): “Execution time certification for gradient-based optimization in model predictive control.” In *Proceedings of the 51st IEEE Conference on Decision and Control*. Maui, HI. To appear.
- Giselsson, P., M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer (2012): “Accelerated gradient methods and dual decomposition in distributed model predictive control.” *Automatica*. To appear. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Jones, C. and M. Morari (2009): “Approximate explicit MPC using bilevel optimization.” In *European Control Conference*. Budapest, Hungary.
- Löfberg, J. (2004): “YALMIP : A toolbox for modeling and optimization in MATLAB.” In *Proceedings of the CACSD Conference*. Taipei, Taiwan.
- Maciejowski, J. M. (2002): *Predictive Control with Constraints*. Prentice Hall, Essex, England.
- McGovern, L. K. (2000): *Computational Analysis of Real-time Convex Optimization for Control Systems*. PhD thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics.
- Nedic, A. and A. E. Ozdaglar (2009): “Approximate primal solutions and rate analysis for dual subgradient methods.” *SIAM Journal on Optimization*, **19:4**, pp. 1757–1780.

- Nesterov, Y. (2003): *Introductory Lectures on Convex Optimization: A Basic Course*, 1st edition. Springer Netherlands.
- Rawlings, J., D. Bonne, J. Jorgensen, A. Venkat, and S. Jorgensen (2008): “Unreachable setpoints in model predictive control.” *IEEE Transactions on Automatic Control*, **53:9**, pp. 2209–2215.
- Rawlings, J. and D. Mayne (2009): *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI.
- Richter, S., C. Jones, and M. Morari (2009): “Real-time input-constrained MPC using fast gradient methods.” In *Proceedings of the 48th Conference on Decision and Control*, pp. 7387–7393. Shanghai, China.
- Richter, S., M. Morari, and C. Jones (2011): “Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method.” In *Proceedings of the 50th IEEE Conference on Decision and Control*, pp. 5223–5229. Orlando, FL.
- Sturm, J. F. (1999): “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones.” *Optimization Methods and Software*, **11–12**, pp. 625–653.
- Tseng, P. (2008): “On accelerated proximal gradient methods for convex-concave optimization.” Submitted to *SIAM J. Optim.* Available: <http://www.math.washington.edu/~tseng/papers/apgm.pdf>.

Paper IV

A Generalized Distributed Accelerated Gradient Method for DMPC with Iteration Complexity Bounds

Pontus Giselsson

Abstract

Most distributed optimization methods used for distributed model predictive control (DMPC) are gradient based. Gradient based optimization algorithms are known to have iterations of low complexity. However, the number of iterations needed to achieve satisfactory accuracy might be significant. This is not a desirable characteristic for distributed optimization in distributed model predictive control. Rather, the number of iterations should be kept low to reduce communication requirements, while the complexity within an iteration can be significant. By incorporating hessian information in a distributed accelerated gradient method in a well-defined manner, we are able to significantly reduce the number of iterations needed to achieve satisfactory accuracy in the solutions, compared to distributed methods that are strictly gradient-based. Further, we provide convergence rate results and iteration complexity bounds for the developed algorithm.

Submitted to *2013 American Control Conference*, Washington, D.C., 2013.

1. Introduction

Many distributed optimization algorithms are based on gradient methods, see [Boyd *et al.*, 2011] and the references therein. Gradient-based optimization methods have low computational complexity within each iteration. However, a limitation of gradient-based methods is the slow convergence rate. For functions with a Lipschitz continuous gradient, i.e., smooth functions, classical gradient-based methods converge at a rate of $O(\frac{1}{k})$ as shown in [Bertsekas, 1999, Nesterov, 2003], where k is the iteration number. This convergence rate is not optimal for gradient methods. It was in [Nemirovsky and Yudin, 1983] shown that a lower bound on the convergence rate for gradient-based methods is $O(\frac{1}{k^2})$. The first method that achieves this accelerated convergence rate was presented by Nesterov in [Nesterov, 1983] for unconstrained problems. This result has been extended and generalized in several publications to handle constrained smooth problems and smooth problems with an additional non-smooth term [Nesterov, 1988, Nesterov, 2005, Beck and Teboulle, 2009, Tseng, 2008]. Recently the accelerated gradient methods has been generalized in [Zuo and Lin, 2011] to allow for a step matrix instead of a scalar step length with preserved convergence rate guarantees.

In the DMPC literature some distributed optimization methods have been used to control sparsely interacting dynamical systems. These include [Negenborn *et al.*, 2008, Wakasa *et al.*, 2008, Doan *et al.*, 2009] in which different reformulations of the classical gradient method with sub-optimal step sizes are used to solve the dual problem. In [Giselsson *et al.*, 2012] an accelerated gradient method is used to solve the DMPC problem and the optimal step size is provided. Further, in [Giselsson, 2012] iteration bounds for the method presented in [Giselsson *et al.*, 2012] are given. In [Trnka *et al.*, 2011] a quasi-Newton method is used to solve the DMPC problem in a water distribution network. The subproblems are solved in parallel, but a central coordinator is needed for this approach.

In this paper we extend the results in [Giselsson *et al.*, 2012] and [Giselsson, 2012] using the generalized accelerated gradient algorithm presented in [Zuo and Lin, 2011]. We present a distributed optimization algorithm applicable to DMPC that use not only gradient information, as is common in distributed optimization, but also hessian information in each iteration. This significantly improves convergence rate compared to previous gradient-based distributed optimization methods for DMPC as is demonstrated by a numerical example. We also provide a bound on the number of iterations needed to guarantee a prespecified dual accuracy and indicate how an iteration bound for the primal variables can be computed. The latter bound is left out for space considerations.

2. Problem Setup

The problem of controlling a linear dynamical system in distributed fashion to the origin is considered. We assume polytopic constraints and apply a distributed MPC controller in which the following optimal control problem with initial condition $\bar{x} \in \mathbb{R}^n$ is solved iteratively

$$\begin{aligned} V_N(\bar{x}) := \min_{\mathbf{x}, \mathbf{u}} \quad & \frac{1}{2} \sum_{t=0}^{N-1} (x_t^T Q x_t + u_t^T R u_t) \\ \text{s.t.} \quad & (x_t, u_t) \in \mathcal{X} \times \mathcal{U}, \quad t = 0, \dots, N-1 \\ & x_{t+1} = A x_t + B u_t, \quad t = 0, \dots, N-2 \\ & x_0 = \bar{x}. \end{aligned} \quad (1)$$

Note that no terminal constraint set or terminal cost is present in the problem formulation. Stability and feasibility results for distributed MPC without terminal constraint set and terminal cost is presented in [Giselson and Rantzer, 2012]. We introduce the following state and control variable partitions

$$x_t = [(x_t^1)^T, \dots, (x_t^M)^T]^T, \quad u_t = [(u_t^1)^T, \dots, (u_t^M)^T]^T$$

where $x_t^i \in \mathbb{R}^{n_i}$ and $u_t^i \in \mathbb{R}^{m_i}$ are referred to as local variables and $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$ are referred to as global variables. The dynamics matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are partitioned accordingly

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \cdots & A_{MM} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & \cdots & B_{1M} \\ \vdots & \ddots & \vdots \\ B_{M1} & \cdots & B_{MM} \end{pmatrix}$$

where $A_{ij} \in \mathbb{R}^{n_i \times n_j}$ and $B_{ij} \in \mathbb{R}^{n_i \times m_j}$. These matrices are assumed to have a sparse structure, i.e., that some $A_{ij} = 0$ and $B_{ij} = 0$. The neighboring interaction is defined by the following sets

$$\begin{aligned} \mathcal{N}_i &= \{j \in \{1, \dots, M\} \mid A_{ij} \neq 0 \text{ or } B_{ij} \neq 0\}, \\ \mathcal{M}_i &= \{j \in \{1, \dots, M\} \mid A_{ji} \neq 0 \text{ or } B_{ji} \neq 0\} \end{aligned}$$

which gives the local dynamics

$$x_{t+1}^i = \sum_{j \in \mathcal{N}_i} (A_{ij} x_t^j + B_{ij} u_t^j), \quad x_0^i = \bar{x}_i$$

2. Problem Setup

for $i = 1, \dots, M$. The global constraint sets are assumed to be products of local sets, i.e.,

$$\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_M, \quad \mathcal{U} = \mathcal{U}_1 \times \dots \times \mathcal{U}_M$$

where the local constraint sets \mathcal{X}_i and \mathcal{U}_i for $i = 1, \dots, M$ are bounded polytopes containing zero in their respective interiors. The local constraint sets can be represented as

$$\mathcal{X}_i = \{x^i \in \mathbb{R}^{n_i} \mid C_x^i x^i \leq d_x^i\}, \quad \mathcal{U}_i = \{u^i \in \mathbb{R}^{m_i} \mid C_u^i u^i \leq d_u^i\}$$

where $C_x^i \in \mathbb{R}^{n_{c_x^i} \times n_i}$, $C_u^i \in \mathbb{R}^{n_{c_u^i} \times m_i}$, $d_x^i \in \mathbb{R}_{>0}^{n_{c_x^i}}$ and $d_u^i \in \mathbb{R}_{>0}^{n_{c_u^i}}$. We define the total number of inequalities in \mathcal{X} and \mathcal{U} by $n_c = \sum_i (n_{c_x^i} + n_{c_u^i})$. The quadratic cost function in (1) is assumed separable, i.e., $\mathbf{Q} = \text{blkdiag}(Q_1, \dots, Q_M)$ and $R = \text{blkdiag}(R_1, \dots, R_M)$ where $Q_i \in \mathbb{S}_{++}^{n_i}$ and $R_i \in \mathbb{S}_{++}^{m_i}$ for $i = 1, \dots, M$ and \mathbb{S}_{++}^n denotes the set of symmetric positive definite matrices in $\mathbb{R}^{n \times n}$. We create the stacked vectors

$$\mathbf{y}_i = [(x_1^i)^T, \dots, (x_{N-1}^i)^T, (u_0^i)^T, \dots, (u_{N-1}^i)^T]^T$$

for $i = 1, \dots, M$ and $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_M^T]^T$. This implies that the optimization (1) problem can more compactly be written as

$$\begin{aligned} V_N(\bar{x}) := \min_{\mathbf{y}} \quad & \frac{1}{2} \mathbf{y}^T \mathbf{H} \mathbf{y} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{y} = \mathbf{b} \bar{x} \\ & \mathbf{C} \mathbf{y} \leq \mathbf{d} \end{aligned} \tag{2}$$

where

$$\begin{aligned} \mathbf{H} &= \text{blkdiag}(\mathbf{H}_1, \dots, \mathbf{H}_M), & \bar{x} &= [\bar{x}_1^T, \dots, \bar{x}_M^T]^T, \\ \mathbf{A} &= [\mathbf{A}_1^T, \dots, \mathbf{A}_M^T]^T, & \mathbf{b} &= [\mathbf{b}_1^T, \dots, \mathbf{b}_M^T]^T, \\ \mathbf{C} &= \text{blkdiag}(\mathbf{C}_1, \dots, \mathbf{C}_M), & \mathbf{d} &= [\mathbf{d}_1^T, \dots, \mathbf{d}_M^T]^T \end{aligned}$$

2.1 Assumptions and Definitions

We define by \mathbb{X}_N the set of initial conditions for which (2) is feasible. We also define

$$P := \mathbf{b}^T (\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)^{-1}\mathbf{b}$$

which characterizes the optimal solution without inequality constraints since

$$\frac{1}{2}\bar{x}^T P \bar{x} = \max_{\lambda} D_N(\bar{x}, \lambda, 0) \leq V_N(\bar{x}). \quad (7)$$

We also introduce the following definition.

DEFINITION 1

We define $\kappa \geq 1$ as the smallest scalar such that for every $\bar{x} \in \mathbb{X}_N$ the following holds

$$V_N(\bar{x}) \leq \frac{\kappa}{2}\bar{x}^T P \bar{x}.$$

□

ASSUMPTION 1

We assume that \mathbf{A} has full row rank and that $\mathbf{A}^T\mathbf{A} + \mathbf{C}^T\mathbf{C}$ is invertible.

□

2.2 Notation

We denote by \mathbb{R} real numbers and $\mathbb{R}_{\geq 0}$ non-negative real numbers. We use the following norm notation $\|x\|_L = \sqrt{x^T L x}$ and $\|x\| = \sqrt{x^T x}$ and inner product $\langle x, y \rangle = x^T y$. Also, $[\cdot]_i$ denotes the i :th element in the vector.

3. Distributed Algorithm

In this section we show how the generalized accelerated gradient method presented in [Zuo and Lin, 2011] can be used in distributed model predictive control. The generalized accelerated gradient method can be applied to problems of the form

$$\min_{x \in \mathcal{X}} f(x)$$

where \mathcal{X} is a closed, convex, and non-empty set and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex and differentiable. Further, f should satisfy

$$f(x_1) \leq f(x_2) + \langle \nabla f(x_2), x_1 - x_2 \rangle + \frac{1}{2}\|x_1 - x_2\|_L^2 \quad (8)$$

for every $x_1, x_2 \in \mathbb{R}^n$ where L is a positive definite matrix. The generalized accelerated gradient algorithm is defined by the iterations

$$\begin{aligned} v^k &= x^k + \frac{k-1}{k+2}(x^k - x^{k-1}) \\ x^{k+1} &= \arg \min_{x \in \mathcal{X}} \left[f(v^k) + \langle \nabla f(v^k), x - v^k \rangle + \frac{1}{2} \|x - v^k\|_L^2 \right] \end{aligned}$$

where k is the iteration number. Straightforward verification gives that these iterations can equivalently be written as

$$v^k = x^k + \frac{k-1}{k+2}(x^k - x^{k-1}) \quad (9)$$

$$x^{k+1} = \arg \min_{x \in \mathcal{X}} \left(\|x - v^k + L^{-1} \nabla f(v^k)\|_L^2 \right). \quad (10)$$

We see that L^{-1} serves as a step matrix for the gradient. The algorithm is a generalization of the algorithm in [Beck and Teboulle, 2009] with the difference is that in [Beck and Teboulle, 2009], L is restricted to being a multiple of the identity matrix.

REMARK 1

The convergence of the algorithm depends on the quadratic upper bound (8) to f . The tighter this upper bound, the fewer iterations can be expected. For L being a multiple of the identity matrix, the quadratic part of the upper bound has the same curvature in every direction, which typically leads to bad convergence rate for ill-conditioned problems. For an appropriately chosen L -matrix the quadratic upper bound to f becomes tighter and a better convergence rate is expected. \square

In the following proposition we show how L should be chosen to satisfy (8) for $f = -D_N$. Before the result is stated, we introduce the matrix

$$T := [\mathbf{A}^T \mathbf{C}^T]^T \mathbf{H}^{-1} [\mathbf{A}^T \mathbf{C}^T]. \quad (11)$$

PROPOSITION 1

Every positive definite matrix L that satisfies $L \succeq T$ satisfies (8) for $f = -D_N$ where D_N is defined in (5). \square

PROOF

We introduce $g = [(\mathbf{b}\bar{x})^T \mathbf{d}]^T$, $z = [\boldsymbol{\lambda}^T \boldsymbol{\mu}^T]^T$ and $\tilde{D}_N(\bar{x}, z) = -z^T T z - g^T z$.

For every $z_1, z_2 \in \mathbb{R}^{N(n+n_c)-n}$ the following holds

$$\begin{aligned}
 \frac{1}{2} \|z_1 - z_2\|_L^2 &\geq \frac{1}{2} \|z_1 - z_2\|_T^2 \\
 &= \frac{1}{2} z_1^T T z_1 + \frac{1}{2} z_2^T T z_2 - z_1^T T z_2 \\
 &= -\tilde{D}_N(\bar{x}, z_1) - g^T z_1 - \frac{1}{2} z_2^T T z_2 - \langle T z_2, z_1 - z_2 \rangle \\
 &= -\tilde{D}_N(\bar{x}, z_1) - g^T z_1 + \tilde{D}_N(\bar{x}, z_2) + g^T z_2 + \\
 &\quad + \langle \nabla \tilde{D}_N(\bar{x}, z_2), z_1 - z_2 \rangle + g^T (z_1 - z_2) \\
 &= -\tilde{D}_N(\bar{x}, z_1) + D_N(\bar{x}, z_2) + \langle \nabla D_N(\bar{x}, z_2), z_1 - z_2 \rangle.
 \end{aligned}$$

Since $\tilde{D}_N(\bar{x}, z) = D_N(\bar{x}, \lambda, \mu)$ if $z = [\lambda^T \mu^T]^T$ and since D_N is concave we have that $f = -D_N$ is convex and satisfies (8). This concludes the proof. \square

We have shown that the generalized accelerated gradient method can be applied to solve the dual problem provided that the matrix L satisfies $L \succeq T = [\mathbf{A}^T \mathbf{C}^T]^T \mathbf{H}^{-1} [\mathbf{A}^T \mathbf{C}^T]$. The following semidefinite program can be used to compute such an L -matrix:

$$\begin{aligned}
 \min_{L \in \mathcal{L}} \text{tr}(L) & \tag{12} \\
 \text{s.t. } L &\succeq [\mathbf{A}^T \mathbf{C}^T]^T \mathbf{H}^{-1} [\mathbf{A}^T \mathbf{C}^T] \\
 L &\succ 0
 \end{aligned}$$

where \mathcal{L} defines some structural constraint on the L -matrix.

To apply the generalized accelerated gradient method, defined by iterations (9)-(10), to solve the dual problem (4) we introduce the dual variable iterations λ^k and μ^k where k is the iteration number and $\bar{\lambda}^k = \lambda^k + \frac{k-1}{k+2}(\lambda^k - \lambda^{k-1})$ and $\bar{\mu}^k = \mu^k + \frac{k-1}{k+2}(\mu^k - \mu^{k-1})$. We also define primal variable iterations as $\mathbf{y}^k = -\mathbf{H}^{-1}(\mathbf{A}^T \lambda^k + \mathbf{C}^T \mu^k)$ and $\bar{\mathbf{y}}^k = \mathbf{y}^k + \frac{k-1}{k+2}(\mathbf{y}^k - \mathbf{y}^{k-1})$. By insertion into (6), the dual function gradient becomes

$$\nabla D_N(\bar{x}, \bar{\lambda}^k, \bar{\mu}^k) = \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \bar{\mathbf{y}}^k - \begin{bmatrix} \mathbf{b}\bar{x} \\ \mathbf{d} \end{bmatrix}.$$

By restricting the set of L matrices to be of the form $L = \text{blkdiag}(L_\lambda, L_\mu)$ it can be verified that the iterations (9)-(10) when applied to the dual

problem (4) becomes

$$\mathbf{y}^k = -\mathbf{H}^{-1}(\mathbf{A}^T \boldsymbol{\lambda}^k + \mathbf{C}^T \boldsymbol{\mu}^k) \quad (13)$$

$$\bar{\mathbf{y}}^k = \mathbf{y}^k + \frac{k-1}{k+2}(\mathbf{y}^k - \mathbf{y}^{k-1}) \quad (14)$$

$$\bar{\boldsymbol{\lambda}}^k = \boldsymbol{\lambda}^k + \frac{k-1}{k+2}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) \quad (15)$$

$$\boldsymbol{\lambda}^{k+1} = \bar{\boldsymbol{\lambda}}^k + L_\lambda^{-1}(\mathbf{A}\bar{\mathbf{y}}^k - \mathbf{b}\bar{x}) \quad (16)$$

$$\bar{\boldsymbol{\mu}}^k = \boldsymbol{\mu}^k + \frac{k-1}{k+2}(\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}) \quad (17)$$

$$\boldsymbol{\mu}^{k+1} = \arg \min_{\boldsymbol{\mu} \geq 0} \left(\|\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}^k - L_\mu^{-1}(\mathbf{C}\bar{\mathbf{y}}^k - \mathbf{d})\|_{L_\mu}^2 \right) \quad (18)$$

REMARK 2

For diagonal L_μ the projection operation in (18) becomes very cheap, namely a max-operation for each element in $\boldsymbol{\mu}$. However, the number of iterations to achieve satisfactory accuracy might be significant. For, non-diagonal L_μ the projection operation is more computationally expensive but for appropriately chosen L a reduced number of iterations is expected. This is desirable in DMPC where the number of iterations, i.e., the amount of communication, should be kept as low as possible. \square

We introduce dual variable partitions

$$\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^T, \dots, \boldsymbol{\lambda}_M^T]^T, \quad \boldsymbol{\mu} = [\boldsymbol{\mu}_1^T, \dots, \boldsymbol{\mu}_M^T]^T$$

according to the division of the equality and inequality constraint matrices \mathbf{A} and \mathbf{C} . By restricting the set of possible step matrices L_λ to $L_\lambda = \text{blkdiag}(L_\lambda^1, \dots, L_\lambda^M)$ and L_μ to $L_\mu = \text{blkdiag}(L_\mu^1, \dots, L_\mu^M)$, where the partitioning corresponds to the partitioning of \mathbf{A} and \mathbf{C} , and by noting that

$$\mathbf{y}_i^k = -\mathbf{H}_i^{-1} \left(\left(\sum_{j \in \mathcal{M}_i} \mathbf{A}_{ji}^T \boldsymbol{\lambda}_j^k \right) + \mathbf{C}_i^T \boldsymbol{\mu}_i^k \right)$$

we get the following distributed algorithm.

ALGORITHM 1—DISTRIBUTED ALGORITHM

Initialize $\boldsymbol{\lambda}_i^0 = \boldsymbol{\lambda}_i^{-1}$, $\boldsymbol{\mu}_i^0 = \boldsymbol{\mu}_i^{-1}$ and $\mathbf{y}_i^0 = \mathbf{y}_i^{-1}$

In every node, i , the following computations are performed

For $k \geq 0$

1. Update primal variables according to

$$\begin{aligned}\mathbf{y}_i^k &= -\mathbf{H}_i^{-1} \left(\left(\sum_{j \in \mathcal{M}_i} \mathbf{A}_{ji}^T \boldsymbol{\lambda}_j^k \right) + \mathbf{C}_i^T \boldsymbol{\mu}_i^k \right) \\ \bar{\mathbf{y}}_i^k &= \mathbf{y}_i^k + \frac{k-1}{k+2} (\mathbf{y}_i^k - \mathbf{y}_i^{k-1})\end{aligned}$$

2. Send $\bar{\mathbf{y}}_i^k$ to each $j \in \mathcal{M}_i$, receive $\bar{\mathbf{y}}_j^k$ from each $j \in \mathcal{N}_i$

3. Update dual variables according to

$$\begin{aligned}\bar{\boldsymbol{\lambda}}_i^k &= \boldsymbol{\lambda}_i^k + \frac{k-1}{k+2} (\boldsymbol{\lambda}_i^k - \boldsymbol{\lambda}_i^{k-1}) \\ \boldsymbol{\lambda}_i^{k+1} &= \bar{\boldsymbol{\lambda}}_i^k + (L_\lambda^i)^{-1} \left(\sum_{j \in \mathcal{N}_i} (\mathbf{A}_{ij} \bar{\mathbf{y}}_j^k - \mathbf{b}_{ij} \bar{x}_j) \right) \\ \bar{\boldsymbol{\mu}}_i^k &= \boldsymbol{\mu}_i^k + \frac{k-1}{k+2} (\boldsymbol{\mu}_i^k - \boldsymbol{\mu}_i^{k-1}) \\ \boldsymbol{\mu}_i^{k+1} &= \arg \min_{\boldsymbol{\mu} \geq 0} \left(\|\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}_i^k - (L_\mu^i)^{-1} (\mathbf{C}_i \bar{\mathbf{y}}_i^k - \mathbf{d}_i)\|_{L_\mu}^2 \right)\end{aligned}$$

4. Send $\boldsymbol{\lambda}_i^{k+1}$ to each $j \in \mathcal{N}_i$, receive $\boldsymbol{\lambda}_j^{k+1}$ from each $j \in \mathcal{M}_i$

End □

We introduce the set of optimal dual variables

$$M^*(\bar{x}) = \left\{ \boldsymbol{\lambda} \in \mathbb{R}^{n(N-1)}, \boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{Nn_c} \mid D_N(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \geq V_N(\bar{x}) \right\}.$$

The convergence rates for the dual function D_N and the primal variables when running Algorithm 1 are stated in the following theorem.

THEOREM 1

Suppose that $\bar{x} \in \mathbb{X}_N$ and let $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$. Then Algorithm 1 has the following convergence rate properties:

1. For $k \geq 1$ the convergence rate for the dual function is

$$D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \leq \frac{2 \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} - \begin{bmatrix} \boldsymbol{\lambda}^0 \\ \boldsymbol{\mu}^0 \end{bmatrix} \right\|_L^2}{(k+1)^2}. \quad (19)$$

2. Let $\mathbf{y}^*(\bar{x})$ be the unique optimal solution to (2) with initial condition \bar{x} . For $k \geq 1$ the convergence rate is

$$\|\mathbf{y}^k - \mathbf{y}^*(\bar{x})\|_2^2 \leq \frac{4 \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} - \begin{bmatrix} \boldsymbol{\lambda}^0 \\ \boldsymbol{\mu}^0 \end{bmatrix} \right\|_L^2}{\sigma_{\min}(\mathbf{H})(k+1)^2} \quad (20)$$

where $\sigma_{\min}(\mathbf{H})$ is the smallest eigenvalue to \mathbf{H} .

□

PROOF

Argument 1 is proven in [Zuo and Lin, 2011] while argument 2 is a straightforward generalization of [Giselsson *et al.*, 2012, Theorem 1(2)].

□

4. Lagrange Multiplier Norm Bounds

From Theorem 1 we conclude that a bound on the norm of the optimal dual variables is needed to bound the number of iterations necessary to achieve a prespecified dual accuracy. First, we state a result from [Giselsson, 2012] in which a bound on the optimal dual variables is presented. Before the result is presented we define $d_{\min} := \min_i [\mathbf{d}]_i$, and

$$\Phi := \mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T, \quad \Psi := (\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{H}^{-1}\mathbf{C}^T.$$

The matrix Φ is invertible since \mathbf{H}^{-1} has full rank and \mathbf{A} has full row rank due to Assumption 1.

LEMMA 1

For every $\bar{x} \in \beta \mathbb{X}_N$ where $\beta \in (0, 1)$ we have that

$$\max_{(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})} \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\| \leq h_\beta(\bar{x}) \quad (21)$$

where

$$h_\beta(\bar{x}) := \left\| \begin{bmatrix} \Psi \\ I \end{bmatrix} \right\| \frac{\kappa - 1}{2(1 - \beta)d_{\min}} \bar{x}^T P \bar{x} + \|\Phi^{-1} \mathbf{b} \bar{x}\|$$

and κ is defined in Definition 1.

□

For the cold starting case, i.e., with $\boldsymbol{\lambda}^0 = 0$ and $\boldsymbol{\mu}^0 = 0$, the convergence rates (19) and (20) depend on $\|[(\boldsymbol{\lambda}^*)^T, (\boldsymbol{\mu}^*)^T]^T\|_L$. A bound on this can

4. Lagrange Multiplier Norm Bounds

be obtained by noting that $\|[(\boldsymbol{\lambda}^*)^T, (\boldsymbol{\mu}^*)^T]^T\|_L \leq \|L\| \|[(\boldsymbol{\lambda}^*)^T, (\boldsymbol{\mu}^*)^T]^T\|$ and using Lemma 1. However, this bound becomes quite conservative and a tighter bound can be computed. To achieve this, we introduce the following decomposition of the dual variables, $\boldsymbol{\lambda} = \boldsymbol{\lambda}_p + \boldsymbol{\lambda}_n$ and $\boldsymbol{\mu} = \boldsymbol{\mu}_p + \boldsymbol{\mu}_n$, where

$$\begin{bmatrix} \boldsymbol{\lambda}_p \\ \boldsymbol{\mu}_p \end{bmatrix} \perp \mathcal{N}([\mathbf{A}^T \ \mathbf{C}^T]) , \quad \begin{bmatrix} \boldsymbol{\lambda}_n \\ \boldsymbol{\mu}_n \end{bmatrix} \in \mathcal{N}([\mathbf{A}^T \ \mathbf{C}^T]) \quad (22)$$

and \mathcal{N} denotes the null-space. We denote by Z an orthonormal basis to the null-space of $[\mathbf{A}^T \ \mathbf{C}^T]$, i.e., $[\mathbf{A}^T \ \mathbf{C}^T]Z = 0$ and $Z^T Z = I$. Since the null-space to $[\mathbf{A}^T \ \mathbf{C}^T]$ is perpendicular to the range of $[\mathbf{A}^T \ \mathbf{C}^T]^T$ the decomposed dual variables can be represented as

$$\begin{bmatrix} \boldsymbol{\lambda}_p \\ \boldsymbol{\mu}_p \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \bar{z}_p , \quad \begin{bmatrix} \boldsymbol{\lambda}_n \\ \boldsymbol{\mu}_n \end{bmatrix} = Z \bar{z}_n \quad (23)$$

where \bar{z}_p and \bar{z}_n are new variables of smaller dimension. The KKT conditions for the dual problem described by the decomposed dual variables are presented next.

PROPOSITION 2

The KKT conditions to (4) are

$$-\mathbf{A}\mathbf{H}^{-1}(\mathbf{A}^T \boldsymbol{\lambda}_p^* + \mathbf{C}^T \boldsymbol{\mu}_p^*) = \mathbf{b}\bar{x} \quad (24)$$

$$-\mathbf{C}\mathbf{H}^{-1}(\mathbf{A}^T \boldsymbol{\lambda}_p^* + \mathbf{C}^T \boldsymbol{\mu}_p^*) = \mathbf{d} + s \quad (25)$$

$$s \leq 0 , \quad \boldsymbol{\mu}_p^* + \boldsymbol{\mu}_n^* \geq 0 \quad (26)$$

$$[(\boldsymbol{\mu}_p^*) + (\boldsymbol{\mu}_n^*)]_i [s]_i = 0 \quad (27)$$

where $\boldsymbol{\lambda}_p^*$, $\boldsymbol{\lambda}_n^*$, $\boldsymbol{\mu}_p^*$ and $\boldsymbol{\mu}_n^*$ satisfy (22) and the optimal dual variables $\boldsymbol{\lambda}^*$, $\boldsymbol{\mu}^*$ satisfy $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_p^* + \boldsymbol{\lambda}_n^*$ and $\boldsymbol{\mu}^* = \boldsymbol{\mu}_p^* + \boldsymbol{\mu}_n^*$. \square

PROOF

The result is immediate from the KKT conditions [Boyd and Vandenberghe, 2004, §5.5.3], the dual variable decomposition $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_p^* + \boldsymbol{\lambda}_n^*$, $\boldsymbol{\mu}^* = \boldsymbol{\mu}_p^* + \boldsymbol{\mu}_n^*$, and due to (22) which implies that $\mathbf{A}^T \boldsymbol{\lambda}_n^* + \mathbf{C}^T \boldsymbol{\mu}_n^* = 0$. \square

REMARK 3

The variables $\boldsymbol{\lambda}_p^*$ and $\boldsymbol{\mu}_p^*$ satisfy the stationarity conditions while $\boldsymbol{\lambda}_n^*$ and $\boldsymbol{\mu}_n^*$ do not affect the stationarity conditions but instead ensure dual feasibility and complementarity. \square

Before we present bounds on the decomposed dual variables we define ζ as the smallest positive scalar such that

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix}^T L \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \preceq \zeta \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix}^T T \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \quad (28)$$

where T is defined in (11). Such finite ζ exists since by Assumption 1 $\mathbf{A}^T \mathbf{A} + \mathbf{C}^T \mathbf{C}$ is invertible and

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix}^T T \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} = (\mathbf{A}^T \mathbf{A} + \mathbf{C}^T \mathbf{C}) \mathbf{H}^{-1} (\mathbf{A}^T \mathbf{A} + \mathbf{C}^T \mathbf{C})$$

which is positive definite since \mathbf{H}^{-1} is positive definite. In the following lemma, bounds for the decomposed optimal dual variables are presented.

LEMMA 2

Suppose that $\bar{x} \in \beta \mathbb{X}_N$ and $\beta \in (0, 1)$. Then

$$\begin{bmatrix} \lambda_p^* \\ \mu_p^* \end{bmatrix}^T L \begin{bmatrix} \lambda_p^* \\ \mu_p^* \end{bmatrix} \leq \kappa \zeta \bar{x}^T P \bar{x} \quad (29)$$

and

$$\begin{bmatrix} \lambda_n^* \\ \mu_n^* \end{bmatrix}^T L \begin{bmatrix} \lambda_n^* \\ \mu_n^* \end{bmatrix} \leq \|Z^T LZ\| \left((h_\beta(\bar{x}))^2 - \frac{\bar{x}^T P \bar{x}}{\|L\|} \right) \quad (30)$$

hold for every λ_p^* , λ_n^* , μ_p^* and μ_n^* that satisfies (22) and the KKT conditions (24)-(27). \square

PROOF

To show (29) we have

$$\begin{aligned} \begin{bmatrix} \lambda_p^* \\ \mu_p^* \end{bmatrix}^T L \begin{bmatrix} \lambda_p^* \\ \mu_p^* \end{bmatrix} &= \bar{z}_p^T \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix}^T L \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \bar{z}_p \leq \zeta \bar{z}_p^T \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix}^T T \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \bar{z}_p \\ &= \zeta \begin{bmatrix} \lambda_p^* \\ \mu_p^* \end{bmatrix}^T T \begin{bmatrix} \lambda_p^* \\ \mu_p^* \end{bmatrix} \\ &= \zeta \begin{bmatrix} \lambda^* - \lambda_n^* \\ \mu^* - \mu_n^* \end{bmatrix}^T \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \mathbf{H}^{-1} \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix}^T \begin{bmatrix} \lambda^* - \lambda_n^* \\ \mu^* - \mu_n^* \end{bmatrix} \\ &= \zeta \begin{bmatrix} \lambda^* \\ \mu^* \end{bmatrix}^T T \begin{bmatrix} \lambda^* \\ \mu^* \end{bmatrix} \end{aligned} \quad (31)$$

where the first equality comes from (23), the first inequality from (28), the second equality from (23) the third equality holds since $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_p^* + \boldsymbol{\lambda}_n^*$ and $\boldsymbol{\mu}^* = \boldsymbol{\mu}_p^* + \boldsymbol{\mu}_n^*$ and due to (11) and the last equality is due to (22) which implies $\mathbf{A}^T \boldsymbol{\lambda}_n^* + \mathbf{C}^T \boldsymbol{\mu}_n^* = 0$.

Further, the KKT conditions for the dual problem (24)-(25) give that

$$0 = T \begin{bmatrix} \boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^* \end{bmatrix} + \begin{bmatrix} \mathbf{b}\bar{x} \\ \mathbf{d} + s \end{bmatrix} = T \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} + \begin{bmatrix} \mathbf{b}\bar{x} \\ \mathbf{d} + s \end{bmatrix}.$$

This implies that

$$\begin{aligned} 0 &= [(\boldsymbol{\lambda}^*)^T (\boldsymbol{\mu}^*)^T] \left(T \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} + \begin{bmatrix} \mathbf{b}\bar{x} \\ \mathbf{d} + s \end{bmatrix} \right) \\ &= [(\boldsymbol{\lambda}^*)^T (\boldsymbol{\mu}^*)^T] T \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} + \bar{x}^T \mathbf{b}^T \boldsymbol{\lambda}^* + (s + \mathbf{d})^T \boldsymbol{\mu}^* \\ &= -V_N(\bar{x}) + \frac{1}{2} [(\boldsymbol{\lambda}^*)^T (\boldsymbol{\mu}^*)^T] T \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \end{aligned} \quad (32)$$

where $s^T \boldsymbol{\mu}^* = 0$ from (27) is used in the final equality. Using (31) and (32) we get

$$\begin{bmatrix} \boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^* \end{bmatrix}^T L \begin{bmatrix} \boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^* \end{bmatrix} \leq \zeta \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix}^T T \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} = \zeta 2V_N(\bar{x}) \leq \zeta \kappa \bar{x}^T P \bar{x}$$

where Definition 1 is used in the last inequality. This proves (29).

Next we show that (30) holds. From (22) we have that

$$[(\boldsymbol{\lambda}_p^*)^T (\boldsymbol{\mu}_p^*)^T][(\boldsymbol{\lambda}_n^*)^T (\boldsymbol{\mu}_n^*)^T]^T = 0,$$

hence Pythagoras' theorem implies that

$$\left\| \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\|^2 - \left\| \begin{bmatrix} \boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^* \end{bmatrix} \right\|^2. \quad (33)$$

Further,

$$\|L\| \left\| \begin{bmatrix} \boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^* \end{bmatrix} \right\|^2 \geq \left\| \begin{bmatrix} \boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^* \end{bmatrix} \right\|_L^2 \geq \left\| \begin{bmatrix} \boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^* \end{bmatrix} \right\|_T^2 = 2V_N(\bar{x}) \geq \bar{x}^T P \bar{x} \quad (34)$$

where the equality comes from (32) and the final inequality comes from (7). By applying Lemma 1 and (34) to (33), we get

$$\left\| \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\|^2 - \left\| \begin{bmatrix} \boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^* \end{bmatrix} \right\|^2 \leq (h_\beta(\bar{x}))^2 - \frac{\bar{x}^T P \bar{x}}{\|L\|}. \quad (35)$$

Further, from (23) we have

$$\begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix} = Z\bar{z}_n = Z(Z^T Z)^{-1} Z^T \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix} = ZZ^T \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix}$$

since $Z^T Z = I$. This implies

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix}^T L \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix}^T ZZ^T LZZ^T \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix} \leq \|ZZ^T LZZ^T\| \left\| \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix} \right\|^2 \\ &\leq \|Z^T LZ\| \left((h_\beta(\bar{x}))^2 - \frac{\bar{x}^T P \bar{x}}{\|L\|} \right) \end{aligned}$$

where the last equality holds since $Z^T Z = I$ and due to (35). This concludes the proof. \square

Using Lemma 2, we are now ready to state the following theorem on dual variable bounds.

THEOREM 2

Suppose that $\bar{x} \in \beta \mathbb{X}_N$ and $\beta \in (0, 1)$. Then for every $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$ we have

$$\begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix}^T L \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \leq \left(\sqrt{\|Z^T LZ\| \left[(h_\beta(\bar{x}))^2 - \frac{\bar{x}^T P \bar{x}}{\|L\|} \right]} + \sqrt{\kappa \zeta \bar{x}^T P \bar{x}} \right)^2.$$

\square

PROOF

Using the triangle inequality we get

$$\begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix}^T L \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} = \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\|_L^2 \leq \left(\left\| \begin{bmatrix} \boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^* \end{bmatrix} \right\|_L + \left\| \begin{bmatrix} \boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^* \end{bmatrix} \right\|_L \right)^2$$

Insertion of the corresponding bounds in Lemma 2 gives the result. \square

Most conservatism in the dual variable bound comes from the function h_β , which originates from the estimate of the dual variable bound in Lemma 1. In Theorem 2 the function h_β is multiplied by $\|Z^T LZ\|$. If L approximates T well, it is anticipated that $\|Z^T LZ\|$ becomes small which gives improved bounds compared to using $\|L\|h_\beta(\bar{x})$.

5. Iteration Bounds

The dual variable bounds presented in the previous section can be used to bound the number of iterations necessary to guarantee a prespecified accuracy of the dual function value and the primal variables. However, for space considerations we omit the primal variable iteration bound result, which is derived similarly to the dual function iteration bound. In the following theorem we present an iteration bound for the cold starting case. We have used a relative accuracy of the optimization problem to avoid that a scaling of the cost-matrices affects the iteration bound.

THEOREM 3

Suppose that $\bar{x} \in \beta\mathbb{X}_N$ and $\beta \in (0, 1)$ and that Algorithm 1 is cold-started, i.e., initialized with $\boldsymbol{\lambda}^0 = 0$, $\boldsymbol{\mu}^0 = 0$, and $\mathbf{y}^0 = 0$. Then the dual function satisfies

$$D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \leq \epsilon_v D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \quad (36)$$

for every $k \geq k_v(\bar{x})$ where

$$k_v(\bar{x}) = \frac{2}{\sqrt{\epsilon_v}} \left(\sqrt{\|Z^T L Z\| \left[\frac{(h_\beta(\bar{x}))^2}{\bar{x}^T P \bar{x}} - \frac{1}{\|L\|} \right]} + \sqrt{\kappa \zeta} \right) - 1. \quad (37)$$

□

PROOF

For the cold starting case we have $\boldsymbol{\lambda}^0 = 0$ and $\boldsymbol{\mu}^0 = 0$. Due to Theorem 1 and since $\frac{1}{2}\bar{x}^T P \bar{x} \leq D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ we conclude that if k is such that

$$\frac{2}{(k+1)^2} \left\| \begin{bmatrix} \boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^* \end{bmatrix} \right\|_L^2 \leq \epsilon_v \frac{1}{2} \bar{x}^T P \bar{x} \quad (38)$$

then (36) holds. Insertion of the bound in Theorem 2 into (38) and rearranging the terms gives the result. □

To compute a bound that holds for all $\bar{x} \in \beta\mathbb{X}_N$ (37) is maximized subject to $\bar{x} \in \beta\mathbb{X}_N$. A more conservative bound is obtained by removing $1/\|L\|$ from (37) which gives the following maximization problem

$$\max_{\bar{x} \in \beta\mathbb{X}_N} \frac{2}{\sqrt{\epsilon_v}} \left(\sqrt{\|Z^T L Z\|} \left(\rho \sqrt{\bar{x}^T P \bar{x}} + \gamma \right) + \sqrt{\kappa \zeta} \right) \quad (39)$$

where

$$\gamma = \|\Phi^{-1} \mathbf{b} P^{-1/2}\|, \quad \rho = \left\| \begin{bmatrix} \Psi \\ I \end{bmatrix} \right\| \frac{\kappa - 1}{2(1 - \beta) \bar{d}_{\min}}.$$

An over-estimator to (39) can be computed by optimizing over $\beta\mathcal{X}$, which satisfies $\beta\mathbb{X}_N \subseteq \beta\mathcal{X}$. This is beneficial since \mathbb{X}_N might be difficult to express explicitly and \mathcal{X} is of lower complexity. The resulting optimization problem depends affinely on $\sqrt{\bar{x}^T P \bar{x}}$. Hence, the maximizing \bar{x} can be computed by maximizing $\bar{x}^T P \bar{x}$ over $\beta\mathcal{X}$ which is a quadratic maximization problem over a polytopic set. Such maximization problems are known to be NP-complete, but can be rewritten as a mixed integer linear program (MILP) as shown in [Jones and Morari, 2009, Lemma 2] for which efficient solvers exist. In every iteration, MILP-software produce upper and lower bounds to the optimal value. To compute an iteration bound, an upper bound to the objective is enough. This implies that the MILP optimization can be stopped when sufficient accuracy has been achieved.

6. Numerical Example

We evaluate the efficiency of the proposed distributed optimization algorithm and the conservatism of the iteration bound by applying it to a dynamical system with sparse structure that is randomly generated. The largest eigenvalue of the dynamics matrix is 1.1, i.e., the system is unstable. The system has 3 subsystems with 5 states and 1 input each, i.e., 15 states and 3 inputs in all. The state and input variables are bounded from above and below by random numbers in the interval $[0.5 \ 1.5]$ and $[-0.15 \ -0.05]$ respectively. The cost matrices Q and R are diagonal and each diagonal element is randomly chosen from the interval $[1 \ 100]$ and the control horizon is chosen to $N = 6$. We use two different L -matrices, one block-diagonal denoted L_d with three blocks corresponding to each of the three subsystems and computed using (12). The other L -matrix is $L_I = \|T\|I$ which is the optimal L -matrix when restricted to being a multiple of the identity matrix as in standard distributed gradient-based optimization (cf. [Giselsson *et al.*, 2012]). The problem data is preconditioned using the technique presented in [Giselsson, 2012]. All simulations are performed in MATLAB and the semidefinite programs are solved through YALMIP [Löfberg, 2004] using SeDuMi [Sturm, 1999] for the preconditioning and SDPNAL [Zhao *et al.*, 2010] (which is more memory-efficient than SeDuMi) for the L -matrix.

In Table 1 the number of iterations needed to achieve a prespecified dual accuracy using the proposed method with L -matrices L_d and L_I are compared. We also compare the iteration complexity bounds presented in Theorem 3 and the one presented [Giselsson, 2012] which was developed for the case where L is a multiple of the identity matrix. The first column in Table 1 specifies the L -matrix used. The second column specifies the duality tolerance and the third column specifies the set from which

Table 1. Experimental results for Algorithm 1 with step matrix L_d and L_I . The number of algorithm iterations and iteration complexity bounds from Theorem 3 and [Giselsson, 2012] are presented.

L-mat.	ϵ_v	β	# iters		iter bound	
			avg.	max.	Thm 3	[Giselsson, 2012]
L_d	0.005	0.25	3.69	10	158	504
L_I	0.005	0.25	9.59	69	-	484
L_d	0.005	0.5	5.62	11	231	1391
L_I	0.005	0.5	23.93	126	-	1333
L_d	0.005	0.75	6.59	12	448	4047
L_I	0.005	0.75	31.18	182	-	3878

the initial conditions are chosen where β is the scaling factor, i.e., initial conditions are chosen from $\beta \mathbb{X}_N$. The fourth and fifth columns present average and max number of iterations while the sixth and seventh columns specify the iteration bounds. The comparison is obtained by solving the optimization problem for 10000 randomly generated initial conditions.

From Table 1 we conclude two things. The first is that by allowing for block-diagonal L -matrices, the presented algorithm reduces significantly the number of iterations needed to achieve a prespecified dual accuracy compared to if L_I is used. This is because second order information is incorporated into the algorithm. The second conclusion is that the iteration bound presented in Theorem 3 is conservative with about one to two orders of magnitude. Further, the bound presented in Theorem 3 grows slower with β and is much less conservative than the one in [Giselsson, 2012] when applied to the algorithm with block-diagonal L -matrix, L_d .

7. Conclusions and Future Work

We have presented a distributed optimization algorithm for distributed MPC that reduces significantly the number of iterations compared to distributed optimization algorithms where only gradient information is used. The reason for this improved iteration complexity is that we have shown how to incorporate hessian information into the distributed algorithm. Further, we have presented an iteration complexity bound for the proposed algorithm that is conservative with about one to two orders of magnitude for the presented numerical example.

A future work direction is to precondition the problem data optimally, where optimally refers to the preconditioning that minimizes the provided iteration bound.

References

- Beck, A. and M. Teboulle (2009): “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” *SIAM J. Imaging Sciences*, **2:1**, pp. 183–202.
- Bertsekas, D. P. (1999): *Nonlinear Programming*, 2nd edition. Athena Scientific, Belmont, MA.
- Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein (2011): “Distributed optimization and statistical learning via the alternating direction method of multipliers.” *Foundations and Trends in Machine Learning*, **3:1**, pp. 1–122.
- Boyd, S. and L. Vandenberghe (2004): *Convex Optimization*. Cambridge University Press, New York, NY.
- Doan, M. D., T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter (2009): “A distributed version of Han’s method for DMPC using local communications only.” *Control Engineering and Applied Informatics*, **11:3**, pp. 6–15.
- Giselsson, P. (2012): “Optimal preconditioning and iteration complexity bounds for gradient-based optimization in model predictive control.” Submitted to 2013 American Control Conference. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Giselsson, P., M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer (2012): “Accelerated gradient methods and dual decomposition in distributed model predictive control.” *Automatica*. To appear. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Giselsson, P. and A. Rantzer (2012): “On feasibility, stability and performance in distributed model predictive control.” Submitted to IEEE Transactions on Automatic Control. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Jones, C. and M. Morari (2009): “Approximate explicit MPC using bilevel optimization.” In *European Control Conference*. Budapest, Hungary.
- Löfberg, J. (2004): “YALMIP : A toolbox for modeling and optimization in MATLAB.” In *Proceedings of the CACSD Conference*. Taipei, Taiwan.
- Negenborn, R., B. De Schutter, and J. Hellendoorn (2008): “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes.” *Engineering Applications of Artificial Intelligence*, **21:3**, pp. 353–366.

- Nemirovsky, A. and D. Yudin (1983): *Informational Complexity and Efficient Methods for Solution of Convex Extremal Problems*. Wiley, New York, NY.
- Nesterov, Y. (1983): “A method of solving a convex programming problem with convergence rate $O(1/k^2)$.” *Soviet Mathematics Doklady*, **27:2**, pp. 372–376.
- Nesterov, Y. (1988): “On an approach to the construction of optimal methods of minimization of smooth convex functions.” *Ékonom. i. Mat. Metody*, **24**, pp. 509–517.
- Nesterov, Y. (2003): *Introductory Lectures on Convex Optimization: A Basic Course*, 1st edition. Springer Netherlands.
- Nesterov, Y. (2005): “Smooth minimization of non-smooth functions.” *Math. Program.*, **103:1**, pp. 127–152.
- Sturm, J. F. (1999): “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones.” *Optimization Methods and Software*, **11–12**, pp. 625–653.
- Trnka, P., J. Pekar, and V. Havlena (2011): “Application of distributed MPC to Barcelona water distribution network.” In *Proceedings of the 18th IFAC World Congress*. Milan, Italy.
- Tseng, P. (2008): “On accelerated proximal gradient methods for convex-concave optimization.” Submitted to *SIAM J. Optim.* Available: <http://www.math.washington.edu/~tseng/papers/apgm.pdf>.
- Wakasa, Y., M. Arakawa, K. Tanaka, and T. Akashi (2008): “Decentralized model predictive control via dual decomposition.” In *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 381–386. Cancun, Mexico.
- Zhao, X.-Y., D. Sun, and K.-C. Toh (2010): “A newton-cg augmented lagrangian method for semidefinite programming.” *SIAM J. on Optimization*, **20:4**, pp. 1737–1765.
- Zuo, W. and Z. Lin (2011): “A generalized accelerated proximal gradient approach for total-variation-based image restoration.” *IEEE Transactions on Image Processing*, **20:10**, pp. 2748–2759.

Paper V

On Feasibility, Stability and Performance in Distributed Model Predictive Control

Pontus Giselsson and Anders Rantzer

Abstract

We present a stopping condition to the duality based distributed optimization algorithm presented in [Giselsson *et al.*, 2012] when used in a distributed model predictive control (DMPC) context. To enable distributed implementation, the optimization problem has neither terminal constraints nor terminal cost that has become standard in model predictive control (MPC). The developed stopping condition guarantees a prespecified performance, stability, and feasibility with finite number of algorithm iterations. Feasibility is guaranteed using a novel adaptive constraint tightening approach that gives the same feasible set as when no constraint tightening is used. Stability and performance of the proposed DMPC controller without terminal cost or terminal constraints is shown based on a controllability parameter for the stage costs. To enable quantification of the control horizon necessary to ensure stability and the prespecified performance, we show how the controllability parameter can be computed by solving a mixed integer linear program (MILP).

Submitted to *IEEE Transactions on Automatic Control*.

1. Introduction

Model predictive control (MPC) is an optimization based control technology for input and state constrained systems. The idea behind MPC is to, in every time step, minimize some cost function based on predictions of future states while respecting state and control constraints. A control trajectory is obtained from the optimization and the first control action from this trajectory is applied to the plant. In the following samples the procedure is repeated with the latest state measurement as initial condition to the state predictions. For a thorough description of MPC, see [Maciejowski, 2002, Rawlings and Mayne, 2009]. There exist a variety of methods to prove stability for system controlled by MPC, see [Mayne *et al.*, 2000] for a survey of such methods and [Rawlings and Mayne, 2009] for further material. As pointed out in [Mayne *et al.*, 2000], common 'ingredients' in these stability proofs are the use of a terminal cost and/or a terminal constraint set in the optimization problem together with a terminal controller that controls the system to the origin once the terminal constraint set is reached. These 'ingredients' are then used to, in various ways, prove that the optimal value function to the optimization problem is a Lyapunov function for the system.

The methods to prove stability in standard MPC [Mayne *et al.*, 2000] are not directly applicable in DMPC formulations where a centralized optimization problem is solved in distributed fashion. Such distributed optimization algorithms often require the cost function to be separable and the constraints to be sparse. This is not the case for the terminal cost or the terminal constraints in standard MPC [Mayne *et al.*, 2000]. Further, the terminal controller that is commonly used to show stability in standard MPC [Mayne *et al.*, 2000] needs to be decentralized or distributed in the context of DMPC. Such stabilizing controllers do not exist for all constrained linear systems [Sandell *et al.*, 1978]. One approach to overcome the aforementioned problems to prove stability in DMPC is to solve local optimization problems sequentially that take neighboring interaction and solutions into account. This is done in [Richards and How, 2007] for linear systems and in [Dunbar, 2007] for nonlinear systems. In [R.M. Hermans, 2010] a DMPC scheme is presented in which stability is proven by adding a constraint to the optimization problem that requires a reduction of an explicit control Lyapunov function. In [Jia and Krogh, 2001, Camponogara *et al.*, 2002] stability is guaranteed for systems satisfying a certain matching condition and if the coupling interaction is small enough. None of the above methods solves a centralized MPC problem and worse global performance is expected than if using an appropriate centralized MPC controller.

To achieve the same performance in DMPC as in centralized MPC,

a centralized problem formulation needs to be considered and solved in distributed fashion. In [Venkat *et al.*, 2008] a centralized MPC problem is solved in distributed fashion and stability is guaranteed in every algorithm iteration. A drawback to this method is that full model knowledge is assumed in each node. Some methods in the DMPC literature rely on duality theory to solve a centralized MPC problem in distributed fashion. In [Negenborn *et al.*, 2008, Wakasa *et al.*, 2008, Doan *et al.*, 2009, Doan *et al.*, 2010] a (sub)gradient algorithm is used to solve the dual problem while the algorithm in [Necoara and Suykens, 2008, Necoara *et al.*, 2008] is based on the smoothing technique presented in [Nesterov, 2005]. The only stability proof is given in [Doan *et al.*, 2009, Doan *et al.*, 2010] where the terminal constraint is set to the origin which is very restrictive and requires long control horizons. Other distributed MPC formulations have been presented in [Maestre *et al.*, 2011] where the DMPC controller is based on a cooperative game and [Dunbar and Murray, 2006, Keviczky *et al.*, 2006] for dynamically decoupled systems. See also [Scattolini, 2009] for a recent survey of distributed and hierarchical MPC methods.

In this paper, a centralized optimization problem is solved in distributed fashion using the distributed accelerated gradient method presented in [Giselsson *et al.*, 2012]. We present a stopping condition for this optimization algorithm that guarantees feasibility, stability, and pre-specified performance of the closed loop system. However, the stopping conditions are not restricted to the optimization algorithm in [Giselsson *et al.*, 2012] but any (distributed) optimization algorithm that produce dual feasible points can be used. The stated optimization problem has neither terminal cost nor terminal constraint set. Stability for MPC without terminal constraints and terminal cost has previously been treated in, e.g., [Grimm *et al.*, 2005]. Further results were reported in [Grüne, 2009] where it was shown how to compute the minimal control horizon necessary to achieve stability and a prespecified performance. The results in [Grüne, 2009] rely on relaxed dynamic programming that was originally presented in [Lincoln and Rantzer, 2006] and extended to MPC in [Grüne and Rantzer, 2008], and on a controllability assumption on the stage costs. The parameters in the controllability assumption in [Grüne, 2009] may be very difficult to compute for a given system. In this paper we take a similar approach as in [Grüne, 2009], but we specify a different controllability parameter than in [Grüne, 2009]. We show, through an explicit expression, how the introduced controllability parameter relates to the performance of the closed loop system. We also show how, for systems with linear dynamics and linear constraints, the controllability parameter can be computed by solving a mixed integer linear program (MILP). This makes the stabilizing control horizon practically computable. We will see that one benefit of not using terminal constraints is that the region of

attraction can be increased significantly compared to standard MPC.

Previous work on MPC where a suboptimal solution to the optimization problem is enough to prove stability has been reported in [Chisci *et al.*, 1996, Scokaert *et al.*, 1999, Diehl *et al.*, 2005]. These rely on that the terminal constraint set can be reached also for suboptimal solutions, which can be used to show closed loop stability. For MPC without a terminal constraint set, stability was shown in [Grüne and Pannek, 2010] for incomplete optimization. The optimization algorithm is terminated early when a certain decrease in the cost has been obtained. However, they do not provide any guarantees that this decrease is achievable in each step. In this paper we use a different decrease condition than in [Grüne and Pannek, 2010] which enables a priori guarantees that the condition will hold with finite number of algorithm iterations in every time step.

An issue associated with duality-based optimization is that primal feasibility cannot be guaranteed before convergence of the optimization algorithm. Such feasibility problems have previously been addressed in [Doan *et al.*, 2011] using a constraint tightening approach. Constraint tightening can be used to generate feasible solutions but complicates stability analysis. The reason is that the optimal value function without constraint tightening is used to show stability, while the actual optimization is performed with constraint tightening. This problem is overcome in [Doan *et al.*, 2011] by assuming that the difference between the optimal value functions with and without constraint tightening is bounded by a constant. However, to actually compute such a constant may be difficult. In this paper we instead use a novel adaptive constraint tightening approach that ensures feasibility w.r.t. the original constraint set with a finite number of algorithm iterations. We introduce a condition for the adaptation that bounds the difference between the optimal value functions with and without constraint tightening. This makes it possible to prove stability without stating additional assumptions.

The paper is organized as follows. In Section 2 we introduce the problem and present the distributed optimization algorithm in [Giselsson *et al.*, 2012]. In Section 3 the stopping condition is presented and feasibility, stability, and performance is analyzed. Section 4 is devoted to computation of the controllability parameter. A numerical example that shows the efficiency of the proposed stopping condition is presented in Section 5. Finally, in Section 6 we conclude the paper.

2. Problem Setup and Preliminaries

We consider linear dynamical systems of the form

$$x_{t+1} = Ax_t + Bu_t, \quad x_0 = \bar{x} \quad (1)$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ denote the state and control vectors at time t and the pair (A, B) is assumed controllable. We introduce the following state and control variable partitions

$$x_t = [(x_t^1)^T, (x_t^2)^T, \dots, (x_t^M)^T]^T, \quad u_t = [(u_t^1)^T, (u_t^2)^T, \dots, (u_t^M)^T]^T \quad (2)$$

where the local variables $x_t^i \in \mathbb{R}^{n_i}$ and $u_t^i \in \mathbb{R}^{m_i}$. The A and B matrices are partitioned accordingly

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \cdots & A_{MM} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & \cdots & B_{1M} \\ \vdots & \ddots & \vdots \\ B_{M1} & \cdots & B_{MM} \end{pmatrix}.$$

These matrices are assumed to have a sparse structure, i.e., some $A_{ij} = 0$ and $B_{ij} = 0$ and the neighboring interaction is defined by the following sets

$$\mathcal{N}_i = \{j \in \{1, \dots, M\} \mid \text{if } A_{ij} \neq 0 \text{ or } B_{ij} \neq 0\}$$

for $i = 1, \dots, M$. This gives the following local dynamics

$$x_{t+1}^i = \sum_{j \in \mathcal{N}_i} (A_{ij}x_t^j + B_{ij}u_t^j), \quad x_0^i = \bar{x}_i$$

for $i = 1, \dots, M$. The local control and state variables are constrained, i.e., $u^i \in \mathcal{U}_i$ and $x^i \in \mathcal{X}_i$. The constraint sets, $\mathcal{X}_i, \mathcal{U}_i$ are assumed to be bounded polytopes containing zero in their respective interiors and can hence be represented as

$$\mathcal{X}_i = \{x^i \in \mathbb{R}^{n_i} \mid C_x^i x^i \leq d_x^i\}, \quad \mathcal{U}_i = \{u^i \in \mathbb{R}^{m_i} \mid C_u^i u^i \leq d_u^i\} \quad (3)$$

where $C_x^i \in \mathbb{R}^{n_{c_x^i} \times n_i}$, $C_u^i \in \mathbb{R}^{n_{c_u^i} \times m_i}$, $d_x^i \in \mathbb{R}_{>0}^{n_{c_x^i}}$ and $d_u^i \in \mathbb{R}_{>0}^{n_{c_u^i}}$. We also denote the total number of linear inequalities describing all constraint sets by $n_c := \sum_{i=1}^M (n_{c_x^i} + n_{c_u^i})$. The global constraint sets are defined from the local ones through

$$\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_M, \quad \mathcal{U} = \mathcal{U}_1 \times \dots \times \mathcal{U}_M.$$

2. Problem Setup and Preliminaries

We use a separable quadratic stage cost

$$\ell(x, u) = \sum_{i=1}^M \ell_i(x^i, u^i) = \frac{1}{2} \left(\sum_{i=1}^M (x^i)^T Q_i x^i + (u^i)^T R_i u^i \right) \quad (4)$$

where $Q_i \in \mathbb{S}_{++}^{n_i}$ and $R_i \in \mathbb{S}_{++}^{m_i}$ for $i = 1, \dots, M$ and \mathbb{S}_{++}^n denotes the set of symmetric positive definite matrices in $\mathbb{R}^{n \times n}$. The optimal infinite horizon cost from initial state $\bar{x} \in \mathcal{X}$ is defined by

$$\begin{aligned} V_\infty(\bar{x}) := \min_{x, u} & \sum_{t=0}^{\infty} \ell(x_t, u_t) \\ \text{s.t.} & x_t \in \mathcal{X}, \quad u_t \in \mathcal{U} \\ & x_{t+1} = Ax_t + Bu_t \\ & x_0 = \bar{x}. \end{aligned} \quad (5)$$

Such infinite horizon optimization problems are in general intractable to solve exactly. A common approach is to solve the problem approximately in receding horizon fashion. To this end we introduce the predicted state and control sequences $\{z_\tau\}_{\tau=0}^{N-1}$ and $\{v_\tau\}_{\tau=0}^{N-1}$ and the corresponding stacked vectors

$$\mathbf{z} = [z_0^T, \dots, z_{N-1}^T]^T, \quad \mathbf{v} = [v_0^T, \dots, v_{N-1}^T]^T \quad (6)$$

where z_τ and v_τ are predicted states and controls τ time steps ahead. The predicted state and control variables z_τ, v_τ are partitioned into local variables as in (2). We also introduce the following stacked local vectors

$$\mathbf{z}_i = [(z_0^i)^T, \dots, (z_{N-1}^i)^T]^T, \quad \mathbf{v}_i = [(v_0^i)^T, \dots, (v_{N-1}^i)^T]^T. \quad (7)$$

Further, we introduce the tightened state and control constraint sets

$$(1 - \delta)\mathcal{X}_i = \{x^i \in \mathbb{R}^{n_i} \mid C_x^i x^i \leq (1 - \delta)d_x^i\}, \quad (8)$$

$$(1 - \delta)\mathcal{U}_i = \{u^i \in \mathbb{R}^{m_i} \mid C_u^i u^i \leq (1 - \delta)d_u^i\} \quad (9)$$

where $\delta \in (0, 1)$ decides the amount of relative constraint tightening. The following optimization problem is solved in the DMPC controller for the current state $\bar{x} \in \mathbb{R}^n$

$$\begin{aligned} V_N^\delta(\bar{x}) := \min_{z, v} & \sum_{\tau=0}^{N-1} \ell(z_\tau, v_\tau) \\ \text{s.t.} & z_\tau \in (1 - \delta)\mathcal{X}, \quad \tau = 0, \dots, N - 1 \\ & v_\tau \in (1 - \delta)\mathcal{U}, \quad \tau = 0, \dots, N - 1 \\ & z_{\tau+1} = Az_\tau + Bv_\tau, \quad \tau = 0, \dots, N - 2 \\ & z_0 = \bar{x}. \end{aligned} \quad (10)$$

Such optimization problems can be solved in distributed fashion using, i.e., the alternating direction of multipliers method [Boyd *et al.*, 2011] or dual ascent [Boyd *et al.*, 2011]. In this work we have chosen to use the recently developed distributed method presented in [Giselsson *et al.*, 2012] which is an accelerated dual ascent method which has superior convergence properties $O(1/k^2)$ compared to the classical dual ascent method which achieves $O(1/k)$. For distribution purposes, we have neither a terminal cost nor a terminal constraint set in the optimization problem (10).

Next, we present the distributed optimization algorithm in [Giselsson *et al.*, 2012]. We stack all decision variables into one vector

$$\mathbf{y} = [z_0^T, \dots, z_{N-1}^T, v_0^T, \dots, v_{N-1}^T]^T \in \mathbb{R}^{(n+m)N}. \quad (11)$$

The optimization problem (10) can more compactly be written as

$$\begin{aligned} V_N^\delta(\bar{x}) := \min_{\mathbf{y}} \quad & \frac{1}{2} \mathbf{y}^T \mathbf{H} \mathbf{y} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{y} = \mathbf{b} \bar{x} \\ & \mathbf{C} \mathbf{y} \leq (1 - \delta) \mathbf{d} \end{aligned} \quad (12)$$

where $\mathbf{H} \in \mathbb{S}_{++}^{(n+m)N}$, $\mathbf{A} \in \mathbb{R}^{n(N-1) \times (n+m)N}$, $\mathbf{b} \in \mathbb{R}^{n(N-1) \times n}$, $\mathbf{C} \in \mathbb{R}^{n_c N \times (n+m)N}$ and $\mathbf{d} \in \mathbb{R}_{>0}^{N n_c}$ are built accordingly. The separable structure of the cost function (4) and constraint sets (3) gives block diagonal \mathbf{H} and \mathbf{C} -matrices. Further, the matrix \mathbf{A} is sparse since it is composed of sparse matrices A , B and I that define the linear dynamic constraints (1). The dual problem to (12) is created by introducing dual variables $\boldsymbol{\lambda} \in \mathbb{R}^{n(N-1)}$ for the equality constraints and dual variables $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{N n_c}$ for the inequality constraints. The dual problem becomes

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} \min_{\mathbf{y}} \frac{1}{2} \mathbf{y}^T \mathbf{H} \mathbf{y} + \boldsymbol{\lambda}^T (\mathbf{A} \mathbf{y} - \mathbf{b} \bar{x}) + \boldsymbol{\mu}^T (\mathbf{C} \mathbf{y} - (1 - \delta) \mathbf{d}) \quad (13)$$

which, as shown in [Giselsson *et al.*, 2012], can explicitly be written as

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} -\frac{1}{2} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu})^T \mathbf{H}^{-1} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu}) - \boldsymbol{\lambda}^T \mathbf{b} \bar{x} - \boldsymbol{\mu}^T \mathbf{d} (1 - \delta). \quad (14)$$

We define the dual function for initial condition $\bar{x} \in \mathbb{R}^n$ as

$$D_N^\delta(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := -\frac{1}{2} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu})^T \mathbf{H}^{-1} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu}) - \boldsymbol{\lambda}^T \mathbf{b} \bar{x} - \boldsymbol{\mu}^T \mathbf{d} (1 - \delta). \quad (15)$$

It was in [Giselsson *et al.*, 2012] shown that the smallest Lipschitz constant to ∇D_N^δ is $L = \|[\mathbf{A}^T, \mathbf{C}^T]^T \mathbf{H}^{-1} [\mathbf{A}^T, \mathbf{C}^T]\|$ and that (12) can be solved by the following accelerated dual gradient method

$$\mathbf{y}^k = -\mathbf{H}^{-1}(\mathbf{A}^T \boldsymbol{\lambda}^k + \mathbf{C}^T \boldsymbol{\mu}^k) \quad (16)$$

$$\bar{\mathbf{y}}^k = \mathbf{y}^k + \frac{k-1}{k+2}(\mathbf{y}^k - \mathbf{y}^{k-1}) \quad (17)$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \frac{k-1}{k+2}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) + \frac{1}{L}(\mathbf{A}\bar{\mathbf{y}}^k - \mathbf{b}\bar{x}) \quad (18)$$

$$\boldsymbol{\mu}^{k+1} = \max\left(0, \boldsymbol{\mu}^k + \frac{k-1}{k+2}(\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}) + \frac{1}{L}(\mathbf{C}\bar{\mathbf{y}}^k - \mathbf{d}(1-\delta))\right). \quad (19)$$

Due to the structure of the matrices $\mathbf{H}, \mathbf{C}, \mathbf{A}$ this algorithm can be implemented in distributed fashion where communication between subsystems i and j takes place if $j \in \mathcal{N}_i$ or $i \in \mathcal{N}_j$, see [Giselsson *et al.*, 2012] for details.

In the following section we present a stopping condition to algorithm (16)-(19) when solving (10) that guarantees feasibility, stability, and a prespecified performance of the DMPC scheme. However, the stopping condition is not developed exclusively for the presented algorithm. It is directly applicable to any (distributed) optimization algorithm that produces dual feasible iterations that converge to the optimal dual variables.

2.1 Notation

We denote by \mathbb{R} the set of real numbers, $\mathbb{R}_{\geq c}$ the set of real numbers $d \geq c$ and $\mathbb{R}_{> c}$ the set of real numbers $d > c$. We denote by $\mathbb{S}_{++}^n \subset \mathbb{R}^{n \times n}$ the set of real symmetric positive definite matrices. Further $\mathbb{N}_{\geq T}$ is the set of natural numbers $t \geq T$. The norm $\|\cdot\|$ refers to the Euclidean norm or the induced Euclidean norm unless otherwise is specified and $\langle \cdot, \cdot \rangle$ refers to the inner product in Euclidean space. The norm $\|x\|_T = \sqrt{x^T T x}$. The interior of a set \mathcal{X} is denoted $\text{int}(\mathcal{X})$. The optimal value function with original constraint set, i.e. $V_N^0(\bar{x})$, is denoted $V_N(\bar{x})$. The optimal state and control sequences to (10) for initial value x and constraint tightening δ are denoted $\{z_\tau^*(x, \delta)\}_{\tau=0}^{N-1}$ and $\{v_\tau^*(x, \delta)\}_{\tau=0}^{N-1}$ respectively and the optimal solution to the equivalent problem (12) by $\mathbf{y}^*(x, \delta)$. The state and control sequences for iteration k in (16)-(19) are denoted $\{z_\tau^k(x, \delta)\}_{\tau=0}^{N-1}$ and $\{v_\tau^k(x, \delta)\}_{\tau=0}^{N-1}$ respectively. We drop the initial state and constraint tightening arguments (x, δ) when no ambiguities can arise.

2.2 Definitions and Assumptions

We adopt the convention that $V_N^\delta(\bar{x}) = \infty$ for states $\bar{x} \in \mathbb{R}^n$ that result in (12) being infeasible. We define by \mathbb{X}_∞ the set for which (5) is feasible.

We also define the minimum of the stage-cost ℓ for fixed x

$$\ell^*(x) := \min_{u \in \mathcal{U}} \ell(x, u) = \frac{1}{2} x^T Q x.$$

Further, κ is the smallest scalar such that $\kappa Q - A^T Q A \geq 0$. The state sequence resulting from applying $\{v_\tau\}_{\tau=0}^{N-1}$ to (1) is denoted by $\{\xi_\tau\}_{\tau=0}^{N-1}$, i.e.,

$$\xi_{\tau+1} = A\xi_\tau + Bv_\tau, \quad \xi_0 = \bar{x}. \quad (20)$$

We introduce $\xi = [(\xi_0)^T, \dots, (\xi_{N-1})^T]^T$ and define the primal cost

$$P_N(\bar{x}, \mathbf{v}) := \begin{cases} \sum_{\tau=0}^{N-1} \ell(\xi_\tau, v_\tau) & \text{if } \xi \in \mathcal{X}^N, \mathbf{v} \in \mathcal{U}^N, \text{ and (20) holds} \\ \infty & \text{else} \end{cases} \quad (21)$$

where \mathcal{X}^N and \mathcal{U}^N are the state and control constraints for the full horizon. We also introduce the shifted control sequence

$$\mathbf{v}_s = [(v_1)^T, \dots, (v_{N-1})^T, 0^T]^T.$$

We have $P_N(\bar{x}, \mathbf{v}^k) \geq V_N(\bar{x})$ and $P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) \geq V_N(A\bar{x} + Bv_0^k)$ for every algorithm iteration k . We denote by $\{\xi_\tau^k\}_{\tau=0}^{N-1}$ the state sequence that satisfies (20) using controls $\{v_\tau^k\}_{\tau=0}^{N-1}$. The definition of the cost (21) implies

$$P_N(\bar{x}, \mathbf{v}^k) = P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + \ell(\bar{x}, v_0^k) - \ell^*(A\xi_{N-1}^k) \quad (22)$$

if $v_0^k \in \mathcal{U}$, $\bar{x} \in \mathcal{X}$ and $A\xi_{N-1}^k \in \mathcal{X}$.

3. Stopping Condition

Rather than finding the optimal solution in each time step in the MPC controller, the most important task is to find a control action that gives desirable closed loop properties such as stability, feasibility, and a desired performance. Such properties can sometimes be ensured well before convergence to the optimal solution. To benefit from this observation, a stopping condition is developed that allows the iterations to stop when the desired performance, stability, and feasibility can be guaranteed. Before the stopping condition is introduced, we briefly go through the main ideas below.

3.1 Main Ideas

The distributed nature of the optimization algorithm makes it unsuitable for centralized terminal costs and terminal constraints. Thus, stability and performance need to be ensured without these constructions. We define the following infinite horizon performance for feedback control law v

$$V_\infty^v(\bar{x}) = \sum_{t=0}^{\infty} \ell(x_t, v(x_t)) \quad (23)$$

where $x_{t+1} = Ax_t + Bv(x_t)$ and $x_0 = \bar{x}$. For a given performance parameter $\alpha \in (0, 1]$ and control law v it is known (c.f. [Lincoln and Rantzer, 2006, Grüne and Rantzer, 2008, Grüne, 2009]) that the following decrease in the optimal value function

$$V_N^0(x_t) \geq V_N^0(Ax_t + Bv(x_t)) + \alpha \ell(x_t, v(x_t)) \quad (24)$$

for every $t \in \mathbb{N}_{\geq 0}$ gives stability and closed loop performance according to

$$\alpha V_\infty^v(\bar{x}) \leq V_\infty(\bar{x}). \quad (25)$$

Analysis of the control horizon N needed for an MPC control law without terminal cost and terminal constraints such that (24) holds, is performed in [Grüne and Rantzer, 2008, Grüne, 2009] and also in this paper. Once a control horizon N is known such that (24) is guaranteed, the performance result (25) relies on computation of the optimal solution to the MPC optimization problem in every time step. An exact optimal solution cannot be computed and the idea behind this paper is to develop stopping conditions that enable early termination of the optimization algorithm with maintained feasibility, stability, and performance guarantees. The idea behind our stopping condition is to compute a lower bound to $V_N^0(x)$ through the dual function $D_N^0(x, \lambda^k, \mu^k)$ and an upper bound to the next step value function $V_N^0(Ax + Bv_0^k)$ through a feasible solution $P_N(Ax + Bv_0^k, \mathbf{v}_s^k)$. If at iteration k the following test is satisfied

$$D_N^0(\bar{x}, \lambda^k, \mu^k) \geq P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + \alpha \ell(\bar{x}, v_0^k) \quad (26)$$

the performance condition (24) holds since

$$\begin{aligned} V_N^0(\bar{x}) &\geq D_N^0(\bar{x}, \lambda^k, \mu^k) \geq P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + \alpha \ell(\bar{x}, v_0^k) \\ &\geq V_N^0(A\bar{x} + Bv_0^k) + \alpha \ell(\bar{x}, v_0^k). \end{aligned}$$

This implies that stability and the performance result (25) can be guaranteed with finite algorithm iterations k by using control action v_0^k .

The test (26) includes computation of $P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k)$ which is a feasible solution to the optimization problem in the following step. A feasible solution cannot be expected with finite number of iterations k for duality-based methods since primal feasibility is only guaranteed in the limit of iterations. Therefore we introduce tightened state and control constraint sets $(1 - \delta)\mathcal{X}$, $(1 - \delta)\mathcal{U}$ with $\delta \in (0, 1)$ and use these in the optimization problem. By generating a state trajectory $\{\xi_\tau^k\}_{\tau=0}^{N-1}$ from the control trajectory $\{v_\tau^k\}_{\tau=0}^{N-1}$ that satisfies the equality constraints (20), we will see that $\{\xi_\tau^k\}_{\tau=0}^{N-1}$ satisfies the original inequality constraints with finite number of iterations. Thus, a primal feasible solution $P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k)$ can be generated after a finite number of algorithm iterations k . However, since the optimization now is performed over a tightened constraint set, the dual function value $D_N^\delta(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ is not a lower bound to $V_N^0(\bar{x})$ and cannot be used directly in the test (26) to ensure stability and the performance specified by (25). In the following lemma we show a relation between the dual function value when using the tightened constraint sets and the optimal value function when using the original constraint sets.

LEMMA 1

For every $\bar{x} \in \mathbb{R}^n$, $\boldsymbol{\lambda} \in \mathbb{R}^{n(N-1)}$ and $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{Nn_c}$ we have that

$$V_N^0(\bar{x}) \geq D_N^\delta(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) - \delta \boldsymbol{\mu}^T \mathbf{d}. \quad (27)$$

□

PROOF

From the definition of the dual function (15) we get that

$$D_N^\delta(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = D_N^0(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) + \delta \mathbf{d}^T \boldsymbol{\mu}.$$

By weak duality we get

$$V_N^0(\bar{x}) \geq D_N^0(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = D_N^\delta(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) - \delta \mathbf{d}^T \boldsymbol{\mu}. \quad (28)$$

This completes the proof. □

The presented lemma enables computation of a lower bound to $V_N^0(\bar{x})$ at algorithm iteration k that depends on $\delta \boldsymbol{\mu}^T \mathbf{d}$. By adapting the amount of constraint tightening δ to satisfy

$$\delta (\boldsymbol{\mu}^k)^T \mathbf{d} \leq \epsilon \ell^*(\bar{x}) \quad (29)$$

for some $\epsilon > 0$ and use this together with the following test

$$D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \geq P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + \alpha \ell(\bar{x}, v_0^k) \quad (30)$$

we get from Lemma 1 and if (29) and (30) holds that

$$\begin{aligned} V_N^0(\bar{x}) &\geq D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) - \delta(\boldsymbol{\mu}^k)^T \mathbf{d} \geq P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + \alpha\ell(\bar{x}, v_0^k) - \epsilon\ell^*(\bar{x}) \\ &\geq V_N^0(A\bar{x} + Bv_0^k) + (\alpha - \epsilon)\ell(\bar{x}, v_0^k). \end{aligned}$$

This is condition (24), which guarantees stability and performance specified by (25) if $\alpha > \epsilon$.

3.2 Stopping Conditions

From the discussion in the previous section we conclude that two parameters need to be specified in the stopping condition. The first is the performance parameter $\alpha \in (0, 1]$ which guarantees closed loop performance as specified by (25). The larger α , the better performance is guaranteed but a longer control horizon N will be needed to guarantee the specified performance. The second is an initial constraint tightening parameter, which we denote by $\delta_{\text{init}} \in (0, 1]$, from which the constraint tightening parameter δ will be adapted (reduced), to satisfy (29). A generic value of δ_{init} is $\delta_{\text{init}} = 0.2$, i.e., 20% initial constraint tightening. Also a third parameter needs to be set. It is the relative optimality tolerance $\epsilon > 0$ where $\epsilon < \alpha$. The effect of this parameter on the algorithm is smaller than the effect of the other parameters and it is generically chosen to satisfy $\epsilon \in [0.01, 0.001]$.

ALGORITHM 1—STOPPING CONDITION

Input: \bar{x}

Set: $k = 0, l = 0, \delta = \delta_{\text{init}}$

Initialize algorithm (16)-(19) with:

$\boldsymbol{\lambda}^0 = \boldsymbol{\lambda}^{-1} = 0, \boldsymbol{\mu}^0 = \boldsymbol{\mu}^{-1} = 0$ and $\mathbf{y}^0 = \mathbf{y}^{-1} = 0$.

Do

If $D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \geq P_N(\bar{x}, \mathbf{v}^k) - \frac{\epsilon}{l+1}\ell^*(\bar{x})$

or $\delta \mathbf{d}^T \boldsymbol{\mu}^k > \epsilon\ell^*(\bar{x})$

Set $\delta \leftarrow \delta/2$ // reduce constraint tightening

Set $l \leftarrow l + 1$

Set $k = 0$ // reset step size and iteration counter

End

Run Δk iterations of (16)-(19)

Set $k \leftarrow k + \Delta k$

Until $D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \geq P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + \alpha\ell(\bar{x}, v_0^k)$ **and**

$\delta \mathbf{d}^T \boldsymbol{\mu}^k \leq \epsilon\ell^*(\bar{x})$

Output: v_0^k □

Except for the initial condition \bar{x} , Algorithm 1 is always identically initialized and follows a deterministic scheme. Thus, for fixed initial condition

the same control action is always computed. This implies that Algorithm 1 defines a static feedback control law, which we denote by v_N . We get the following closed loop dynamics

$$x_{t+1} = Ax_t + Bv_N(x_t), \quad x_0 = \bar{x}. \quad (31)$$

The objective of this section is to present a theorem stating that the feedback control law v_N is well defined on $\text{int}(\mathbb{X}_N^0)$ where

$$\mathbb{X}_N^\delta := \{\bar{x} \in \mathbb{R}^n \mid V_N^\delta(\bar{x}) < \infty \text{ and } Az_{N-1}^*(\bar{x}, 0) \in \text{int}(X)\} \quad (32)$$

which satisfies $\mathbb{X}_N^{\delta_1} \subseteq \mathbb{X}_N^{\delta_2}$ for $\delta_1 > \delta_2$. First, however we state the following definition.

DEFINITION 1

The constant Φ_N is the smallest constant such that the optimal solution $\{z_\tau^*(\bar{x}, 0)\}_{\tau=0}^{N-1}$, $\{v_\tau^*(\bar{x}, 0)\}_{\tau=0}^{N-1}$ to (10) for every $\bar{x} \in \mathbb{X}_N^0$ satisfies

$$\ell^*(z_{N-1}^*(\bar{x}, 0)) \leq \Phi_N \ell(\bar{x}, v_0^*(\bar{x}, 0)) \quad (33)$$

for the chosen control horizon N . □

In Section 4 a method to compute Φ_N is presented.

REMARK 1

In [Grimm *et al.*, 2005, Grüne, 2009] an exponential controllability on the stage costs is assumed, i.e., that for $C \geq 1$ and $\sigma \in (0, 1)$ the following holds for $\tau = 0, \dots, N-1$

$$\ell^*(z_\tau^*(\bar{x}, 0), v_\tau^*(\bar{x}, 0)) \leq C\sigma^\tau \ell(\bar{x}, v_0^*(\bar{x}, 0)). \quad (34)$$

This implies $\Phi_N \leq C\sigma^{N-1}$. □

We also need the following lemmas that are proven in Appendix A.1, Appendix A.2 and Appendix A.3 respectively to prove the upcoming theorem.

LEMMA 2

Suppose that $\epsilon > 0$ and $\delta \in (0, 1]$. For every $\bar{x} \in \mathbb{X}_N^\delta$ we have for some finite k that

$$D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \geq P_N(\bar{x}, \mathbf{v}^k) - \epsilon \ell^*(\bar{x}). \quad (35)$$

□

LEMMA 3

Suppose that $\epsilon > 0$ and $\delta \in (0, 1]$. For every $\bar{x} \in \mathbb{X}_N^\delta$ and algorithm iteration k such that (35) holds we have for $\tau = 0, \dots, N-1$ that

$$\frac{1}{2} \left\| \begin{bmatrix} \xi_\tau^k(\bar{x}, \delta) \\ u_\tau^k(\bar{x}, \delta) \end{bmatrix} - \begin{bmatrix} z_\tau^*(\bar{x}, 0) \\ u_\tau^*(\bar{x}, 0) \end{bmatrix} \right\|_H^2 \leq \epsilon \ell^*(\bar{x}) + \delta (\boldsymbol{\mu}^k)^T \mathbf{d}$$

where $H = \text{blkdiag}(Q, R)$. □

LEMMA 4

Suppose that $\epsilon > 0$ and $\delta \in (0, 1]$. For $\bar{x} \in \mathbb{X}_N^0$ but $\bar{x} \notin \mathbb{X}_N^\delta$ we have that $\delta (\boldsymbol{\mu}^k)^T \mathbf{d} > \epsilon \ell^*(\bar{x})$ with finite k . □

We are now ready to state the following theorem, which is proven in Appendix A.4.

THEOREM 1

Assume that $\epsilon > 0$, $\delta_{\text{init}} \in (0, 1]$ and

$$\alpha \leq 1 - \epsilon - \kappa(\sqrt{2\epsilon} + \sqrt{\Phi_N})^2(\sqrt{2\epsilon} + 1)^2. \quad (36)$$

Then the feedback control law v_N , defined by Algorithm 1, is well defined for every $\bar{x} \in \text{int}(\mathbb{X}_N^0)$. Further

$$V_N^0(\bar{x}) \geq V_N^0(A\bar{x} + Bv_N(\bar{x})) + (\alpha - \epsilon)\ell(\bar{x}, v_N(\bar{x})). \quad (37)$$

holds for every $\bar{x} \in \text{int}(\mathbb{X}_N^0)$. □

COROLLARY 1

Suppose that $\alpha \leq 1 - \kappa\Phi_N$ and that $v_N^*(\bar{x}) = v_0^*(\bar{x}, 0)$. Then

$$V_N^0(\bar{x}) \geq V_N^0(A\bar{x} + Bv_N^*(\bar{x})) + \alpha\ell(\bar{x}, v_N^*(\bar{x})).$$

holds for every $\bar{x} \in \mathbb{X}_N^0$. □

PROOF

For every $\bar{x} \in \mathbb{X}_N^0$ we have

$$\begin{aligned} V_N^0(\bar{x}) &= \sum_{\tau=0}^{N-1} \ell(z_\tau^*, u_\tau^*) + \ell(Az_{N-1}^*, 0) - \ell(Az_{N-1}^*, 0) \\ &\geq V_N^0(A\bar{x} + Bv_N^*(\bar{x})) + \ell(\bar{x}, v_0^*) - \ell(Az_{N-1}^*, 0) \\ &\geq V_N^0(A\bar{x} + Bv_N^*(\bar{x})) + \ell(\bar{x}, v_0^*) - \kappa\ell(z_{N-1}^*, 0) \\ &\geq V_N^0(A\bar{x} + Bv_N^*(\bar{x})) + (1 - \kappa\Phi_N)\ell(\bar{x}, v_0^*) \end{aligned}$$

where the first inequality holds since $Az_{N-1}^* \in \mathcal{X}$ by construction of \mathbb{X}_N^0 , the second due to the definition of κ and the third due to the definition of Φ_N . \square

REMARK 2

By setting $\epsilon = 0$ in Theorem 1 we get $\alpha \leq 1 - \kappa\Phi_N$ as in Corollary 1. \square

We have proven that the feedback control law is well defined on $\text{int}(\mathbb{X}_N^0)$. The topic of the following section is to analyze feasibility, stability, and performance of the proposed feedback controller.

3.3 Feasibility, Stability and Performance

The following proposition shows one-step feasibility when using the feedback control law v_N .

PROPOSITION 1

Suppose that α satisfies (36). For every $x_t \in \text{int}(\mathbb{X}_N^0)$ we have that $x_{t+1} = Ax_t + Bv_N(x_t) \in \mathcal{X}$. \square

PROOF

From Theorem 1 we have that $v_N(x_t)$ is well defined and from Algorithm 1 we have that $P_N(x_{t+1}, \mathbf{v}_s^k) < \infty$ which, by definition, implies that $x_{t+1} \in \mathcal{X}$. \square

The proposition shows that x_{t+1} is feasible if the control law $v_N(x_t)$ is well defined. We define the recursively feasible set as the maximal set such that

$$\mathbb{X}_{\text{rf}} = \{x \in \mathcal{X} \mid Ax + Bv_N(x) \in \mathbb{X}_{\text{rf}}\} \quad (38)$$

In the following theorem we show that \mathbb{X}_{rf} is the region of attraction and that the control law v_N achieves a prespecified performance as specified by (23).

THEOREM 2

Suppose that $\alpha > \epsilon$ satisfies (36). Then for every initial condition $\bar{x} \in \mathbb{X}_{\text{rf}}$ we have that $\|x_t\| \rightarrow 0$ as $t \rightarrow \infty$ and that the closed loop performance satisfies

$$(\alpha - \epsilon)V_\infty^{v_N}(\bar{x}) \leq V_\infty(\bar{x}). \quad (39)$$

Further, \mathbb{X}_{rf} is the region of attraction. \square

PROOF

From the definition of \mathbb{X}_{rf} we know that $\bar{x} = x_0 \in \mathbb{X}_{\text{rf}}$ implies $x_t \in \mathbb{X}_{\text{rf}}$ for all $t \in \mathbb{N}_{\geq 0}$. This implies that $v_N(x_t)$ is well defined and that (37) holds for all x_t , $t \in \mathbb{N}_{\geq 0}$. In [Grüne and Rantzer, 2008, Proposition 2.2] it was shown using telescope summation that (37) implies (39). Further,

4. Offline Controllability Verification

since the stage cost ℓ satisfies [Grüne, 2009, Assumption 5.1] we get from [Grüne, 2009, Theorem 5.2] that $\|x_t\| \rightarrow 0$ as $t \rightarrow \infty$.

What is left to show is that \mathbb{X}_{rf} is the region of attraction. Denote by \mathbb{X}_{roa} the region of attraction using v_N . We have above shown that $\mathbb{X}_{\text{rf}} \subseteq \mathbb{X}_{\text{roa}}$. We next show that $\mathbb{X}_{\text{roa}} \subseteq \mathbb{X}_{\text{rf}}$ by a contradiction argument to conclude that $\mathbb{X}_{\text{rf}} = \mathbb{X}_{\text{roa}}$. Assume that there exist $\bar{x} \in \mathbb{X}_{\text{roa}}$ such that $\bar{x} \notin \mathbb{X}_{\text{rf}}$. If $\bar{x} \in \mathbb{X}_{\text{roa}}$ the closed loop state sequence $\{x_t\}_{t=0}^{\infty}$ is feasible in every step (and converges to the origin) and consequently $\{Ax_t + Bv_N(x_t)\}_{t=0}^{\infty}$ is feasible in every step. This is exactly the requirement to have $\bar{x} \in \mathbb{X}_{\text{rf}}$, which is a contradiction. Thus $\mathbb{X}_{\text{rf}} \subseteq \mathbb{X}_{\text{roa}} \subseteq \mathbb{X}_{\text{rf}}$ which implies that $\mathbb{X}_{\text{rf}} = \mathbb{X}_{\text{roa}}$. \square

This completes the proof. \square

REMARK 3

The lack of terminal constraint sets implies that recursive feasibility cannot be guaranteed. However, to actually guarantee recursive feasibility in presence of disturbances, robust MPC formulations need to be considered. These can be fairly restrictive and have a rather small region of attraction. In the examples we will see that the region of attraction can be significantly enlarged by not using terminal constraints. \square

To guarantee a priori that the control law v_N achieves the performance (39) specified by α , we need to find a control horizon N such that the corresponding controllability parameter Φ_N satisfies (36). This requires the computation of controllability parameter Φ_N which is the topic of the next section.

4. Offline Controllability Verification

The stability and performance results in Theorem 2 rely on Definition 1. For the results to be practically meaningful it must be possible to compute Φ_N in Definition 1. In this section we will show that this can be done by solving a mixed integer linear program (MILP). For desired performance specified by α , we get a requirement on the controllability parameter through (36) for Theorem 1 and Theorem 2 to hold. We denote by Φ_α the largest controllability parameter such that Theorem 1 and Theorem 2 holds for the specified α . This parameter is the one that gives equality in (36), i.e., satisfies

$$\alpha = 1 - \epsilon - \kappa(\sqrt{2\epsilon} + \sqrt{\Phi_\alpha})^2(\sqrt{2\epsilon} + 1)^2 \quad (40)$$

for the desired performance α and optimality tolerance ϵ . The parameters α and ϵ must be chosen such that $\Phi_\alpha > 0$. The objective is to find a

control horizon N such that the corresponding controllability parameter Φ_N satisfies $\Phi_N \leq \Phi_\alpha$. First, we show that for long enough control horizon N there exist a $\Phi_N \leq \Phi_\alpha$.

LEMMA 5

Assume that α and ϵ are chosen such that $\Phi_\alpha > 0$ where Φ_α is implicitly defined in (40). Then there exists control horizon N and corresponding controllability parameter $\Phi_N \leq \Phi_\alpha$. \square

PROOF

Since \mathbb{X}_{rf} is the region of attraction we have $\mathbb{X}_{\text{rf}} \subseteq \mathbb{X}_\infty$. This in turn implies that (12) is feasible for every control horizon $N \in \mathbb{N}_{\geq 1}$ due to the absence of terminal constraints. We have

$$V_N(\bar{x}) = \sum_{\tau=0}^{N-2} \ell(z_\tau^*, v_\tau^*) + \ell(z_{N-1}^*, v_{N-1}^*) \geq V_{N-1}(\bar{x}) + \ell(z_{N-1}^*, v_{N-1}^*).$$

Since the pair (A, B) is assumed controllable and since (12) has neither terminal constraints nor terminal cost we have for some finite M that $M \geq V_\infty(\bar{x}) \geq V_N(\bar{x}) \geq V_{N-1}(\bar{x})$. Thus, the sequence $\{V_N(\bar{x})\}_{N=0}^\infty$ is a bounded monotonic increasing sequence which is well known to be convergent. Thus, for $N \geq \bar{N}$ where \bar{N} is large enough the difference $V_N(\bar{x}) - V_{N-1}(\bar{x})$ is arbitrarily small. Especially $\ell(z_{N-1}^*, v_{N-1}^*) = \ell^*(z_{N-1}^*) \leq V_N(\bar{x}) - V_{N-1}(\bar{x}) \leq \Phi_\alpha \ell(\bar{x}, v_0^*)$ since $\Phi_\alpha > 0$. That is, for long enough control horizon $N \geq \bar{N}$, $\Phi_N \leq \Phi_\alpha$. This completes the proof. \square

The preceding Lemma shows that there exists a control horizon N such that $\Phi_N \leq \Phi_\alpha$ if $\Phi_\alpha > 0$ for the chosen performance α and tolerance ϵ . The choice of performance parameter α gives requirements on how ϵ can be chosen to give $\Phi_\alpha > 0$. Larger ϵ requires smaller Φ_α to satisfy (40) which in turn requires longer control horizons N since Φ_N must satisfy $\Phi_N \leq \Phi_\alpha$. In the following section we address the problem of how to compute the control horizon N and corresponding Φ_N such that the desired performance specified by α can be guaranteed.

4.1 Exact Verification of Controllability Parameter

In the following proposition we introduce an optimization problem that tests if the controllability parameter Φ_N corresponding to control horizon N satisfies $\Phi_N \leq \Phi_\alpha$ for the desired performance specified by α . Before we state the proposition, the following matrices are introduced

$$\begin{aligned} T &= \text{blkdiag}(0, \dots, 0, -Q, \Phi_\alpha R, 0, \dots, 0, -R) \\ S &= \text{blkdiag}(0, \dots, 0, I, 0, \dots, 0) \end{aligned}$$

4. Offline Controllability Verification

where Q and R are the cost matrices for states and inputs and Φ_α is the required controllability parameter for the chosen α . Recalling the partitioning (11) of \mathbf{y} implies that

$$\begin{aligned}\mathbf{y}^T T \mathbf{y} &= v_0^T R v_0 - z_{N-1}^T Q z_{N-1} - v_{N-1}^T R v_{N-1} \\ S \mathbf{y} &= z_{N-1}\end{aligned}$$

PROPOSITION 2

Assume that $\Phi_\alpha > 0$ satisfies (40) for the chosen performance parameter α and optimality tolerance ϵ . Further assume that the control horizon N is such that

$$\begin{aligned}0 &= \min_{\bar{x}} \frac{1}{2} (\Phi_\alpha \bar{x}^T Q \bar{x} + \mathbf{y}^T T \mathbf{y}) & (41) \\ \text{s.t. } \bar{x} &\in \mathbb{X}_N^0 \\ \mathbf{y} &= \arg \min V_N^0(\bar{x})\end{aligned}$$

then $\Phi_N \leq \Phi_\alpha$. □

PROOF

First we note that $\bar{x} = 0$ gives $\mathbf{y} = 0$ and $\Phi_\alpha \bar{x}^T Q \bar{x} + \mathbf{y}^T T \mathbf{y} = 0$, i.e., we have that 0 is always a feasible solution. Further, (41) implies for every $\bar{x} \in \mathbb{X}_N^0$ that

$$0 \leq \Phi_\alpha \bar{x}^T Q \bar{x} + \mathbf{y}^T T \mathbf{y} = \Phi_\alpha \ell(\bar{x}, v_0^*) - \ell(z_{N-1}^*, v_{N-1}^*) = \Phi_\alpha \ell(\bar{x}, v_0^*) - \ell^*(z_{N-1}^*)$$

since $v_{N-1}^* = 0$. This is exactly the condition in Definition 1. Since Φ_N is the smallest such constant, we have $\Phi_N \leq \Phi_\alpha$ for the chosen control horizon N and desired performance α and optimality tolerance ϵ . □

The optimization problem (41) is a bilevel optimization problem with indefinite quadratic cost (see [Colson *et al.*, 2005] for a survey on bilevel optimization). Such problems are in general NP-hard to solve. The problem can, however, be rewritten as an equivalent MILP as shown in the following proposition which is a straightforward application of [Jones and Morari, 2009, Theorem 2].

PROPOSITION 3

Assume that $\Phi_\alpha > 0$ satisfies (40) for the chosen performance parameter α and optimality tolerance ϵ . If the control horizon N is such that the

following holds

$$0 = \min -\frac{1}{2} (d_x^T \mu^{U1} + d_x^T \mu^{U2} + \mathbf{d}^T \mu^{UL1}) \quad (42)$$

$$\text{s.t. } \beta_i^L \in \{0, 1\}, \beta_i^{U1} \in \{0, 1\}, \beta_i^{U2} \in \{0, 1\}$$

Upper level

Primal and dual feasibility

$$C_x \bar{x} - d_x - s^x = 0$$

$$s^x \leq 0, \mu^{U1} \geq 0$$

$$C_x \mathbf{A} \mathbf{S} \mathbf{y} - d_x - s^z = 0$$

$$s^z \leq 0, \mu^{U2} \geq 0$$

Stationarity

$$\Phi_\alpha \mathbf{Q} \bar{x} + (C_x)^T \mu^{U1} - \mathbf{b}^T \lambda^{UL2} = 0$$

$$\mathbf{T} \mathbf{y} + \mathbf{H}^T \lambda^{UL1} + \mathbf{A}^T \lambda^{UL2} + \mathbf{C}^T \mu^{UL1} + (C_x \mathbf{A} \mathbf{S})^T \mu^{U2} = 0$$

$$\mathbf{A} \lambda^{UL1} = 0$$

$$\mathbf{C} \lambda^{UL1} - \mu^{UL2} = 0$$

Complementarity

$$\beta_i^L = 1 \Rightarrow \mu_i^{UL2} = 0, \beta_i^L = 0 \Rightarrow \mu_i^{UL1} = 0$$

$$\beta_i^{U1} = 1 \Rightarrow s_i^x = 0, \beta_i^{U1} = 0 \Rightarrow \mu_i^{U1} = 0$$

$$\beta_i^{U2} = 1 \Rightarrow s_i^z = 0, \beta_i^{U2} = 0 \Rightarrow \mu_i^{U2} = 0$$

Lower level

Primal and dual feasibility

$$\mathbf{A} \mathbf{y} - \mathbf{b} \bar{x} = 0$$

$$\mathbf{C} \mathbf{y} - \mathbf{d} - s = 0$$

$$s \leq 0, \mu^L \geq 0$$

Stationarity

$$\mathbf{H} \mathbf{y} + \mathbf{A}^T \lambda^L + \mathbf{C}^T \mu^L = 0$$

Complementarity

$$\beta_i^L = 1 \Rightarrow s_i = 0, \beta_i^L = 0 \Rightarrow \mu_i^L = 0$$

then $\Phi_\alpha \geq \Phi_N$. □

PROOF

The set \mathbb{X}_N^0 can equivalently be written as

$$\mathbb{X}_N^0 = \{x \in \mathbb{R}^n \mid \mathbf{A} \mathbf{y}^*(x, 0) = \mathbf{b} x, \mathbf{C} \mathbf{y}^*(x, 0) \leq \mathbf{d}, \\ C_x \mathbf{A} \mathbf{S} \mathbf{y}^*(x, 0) \leq d_x, C_x x \leq d_x\}. \quad (43)$$

4. Offline Controllability Verification

We express the set \mathbb{X}_N^0 in (41) using (43). The equivalence between the optimization problems (42) and (41) is established in [Jones and Morari, 2009, Theorem 2]. The remaining parts of the proposition follow by applying Proposition 2. \square

The transformation from (41) to (42) is done by expressing the lower level optimization problem in (41) by its sufficient and necessary KKT conditions to get a single level indefinite quadratic program with complementarity constraints. The resulting indefinite quadratic program with complementarity constraints can in turn be cast as a MILP to get (42).

REMARK 4

Although MILP problems are NP-hard, there are efficient solvers available such as CPLEX and GUROBI. There are also solvers available for solving the bilevel optimization problem (41) directly, e.g., the function *solvebilevel* in YALMIP, [Löfberg, 2004]. \square

If the chosen control horizon N is not long enough for $\Phi_N \leq \Phi_\alpha$, different heuristics can be used to choose a new longer horizon to be verified. One heuristic is to assume exponential controllability as in Remark 1, i.e., that there exist constants $C \geq 1$ and $\sigma \in (0, 1)$ such that

$$C\sigma^\tau \ell(\bar{x}, v_0^k) \geq \ell(z_\tau^k, v_\tau^k) \quad (44)$$

for all $\tau = 0, \dots, N - 1$. The C and σ -parameters should be determined using the optimal solution \mathbf{y} to (12) for the x that minimized (42) in the previous test. Under the assumption that (44) holds as N increases, a new guess on the control horizon N can be computed by finding the smallest N such that $C\sigma^{N-1} \leq \Phi_\alpha$.

4.2 Controllability Parameter Estimation

The test in Proposition 3 verifies if the control horizon N is long enough for the controllability assumption to hold for the required controllability parameter Φ_α . Thus, an initial guess on the control horizon is needed. A guaranteed lower bound can easily be computed by solving (12) for a variety of initial conditions \bar{x} and compute the worst controllability parameter, denoted by $\hat{\Phi}_N$, for these sample points. If the estimated controllability parameter $\hat{\Phi}_N \geq \Phi_\alpha$, we know that the control horizon need to be increased for (42) to hold. If instead $\hat{\Phi}_N \leq \Phi_\alpha$ the control horizon N might serve as a good initial guess to be verified by (42).

REMARK 5

For large systems (42) may be too complex to verify the desired performance. In such cases the heuristic method mentioned above can be used

in conjunction with an adaptive horizon scheme. The adaptive scheme keeps the horizon fixed for all time-steps until the controllability assumption does not hold. Then, the control horizon is increased to satisfy the assumption and kept at the new level until the controllability assumption does not hold again. Eventually the control horizon will be large enough for $\Phi_N \leq \Phi_\alpha$ and the horizon need not be increased again. \square

5. Numerical Example

We evaluate the efficiency of the proposed distributed feedback control law v_N by applying it to a randomly generated dynamical system with sparsity structure. The random dynamics matrix is scaled such that the magnitude of the largest eigenvalue is 1.1, i.e., the system is unstable. The system has 3 subsystems with 5 states and 1 input each, i.e., 15 states and 3 inputs in all. All state and input variables are upper and lower bounded by random numbers in the intervals $[0.5, 1.5]$ and $[-0.15 - 0.05]$ respectively. The stage cost is chosen to be

$$\ell_i(x_i, u_i) = x_i^T x_i + u_i^T u_i$$

for $i = 1, 2, 3$. We have chosen two different suboptimality parameters $\alpha_1 = 0.01$ and $\alpha_2 = 0.5$. We need to find control horizon $N(\alpha_i)$ such that the controllability parameter $\Phi_{N(\alpha_i)} \leq \Phi_{\alpha_i}$ for $i = 1, 2$. To compute Φ_{α_i} the optimality tolerance ϵ need to be chosen and κ need to be computed where κ is the smallest constant such that $\kappa Q \succeq A^T Q A$. We have chosen $\epsilon = 0.005$ and we have found that $\kappa = 1.22$. Using (40) we get $\Phi_{\alpha_1} = 0.51$ and $\Phi_{\alpha_2} = 0.22$. This implies that we need to find a control horizon $N(0.01)$ such that $\Phi_{N(0.01)} \leq 0.51$ and $N(0.5)$ such that $\Phi_{N(0.5)} \leq 0.22$. Verification by solving the MILP in (42) gives that $N(0.01) = 6$ and $N(0.5) = 9$.

The efficiency of the optimization algorithm (16)-(19) is investigated in [Giselsson *et al.*, 2012]. The focus of this section is to evaluate the efficiency of the proposed adaptive constraint tightening approach in Algorithm 1. Further, we analyze the region of attraction for Algorithm 1, which is based on an optimization problem without terminal constraint. We compare the region of attraction to the region of attraction in standard MPC where a terminal constraint set is used. The terminal constraint set is computed as the maximal positive invariant set (see [Gilbert and Tan, 1991]) which in our example is a polytope defined by 288 linear inequality constraints.

Table 1 presents the results obtained when the algorithm is running with different suboptimality parameters, $\alpha_1 = 0.01$ and $\alpha_2 = 0.5$. The first

Table 1. Experimental results for different performance requirements α and different initial constraint tightenings δ_{init} in the DMPC-controller. Also, the region of attraction (R.o.A.) for the DMPC-controller is compared to the region of attraction in centralized MPC with terminal constraint set.

Algorithm comparison, $\alpha = 0.01$, $N = 6$

	ϵ	δ_{init}	avg. # iters	avg. δ	R.o.A.
Alg. 1	0.005	0.0001	278.2	0.0001	82.4 %
Alg. 1	0.005	0.001	155.6	0.001	82.4 %
Alg. 1	0.005	0.01	66.6	0.01	82.4 %
Alg. 1	0.005	0.05	36.9	0.047	82.4 %
Alg. 1	0.005	0.1	35.6	0.056	82.4 %
Alg. 1	0.005	0.2	35.3	0.064	82.4 %
Alg. 1	0.005	0.5	35.3	0.080	82.4 %
CMPC	-	-	-	-	0.9 %

Algorithm comparison, $\alpha = 0.5$, $N = 9$

	ϵ	δ_{init}	avg. # iters	avg. δ	R.o.A.
Alg. 1	0.005	0.0001	403.2	0.0001	92.2 %
Alg. 1	0.005	0.001	199.0	0.001	92.2 %
Alg. 1	0.005	0.01	82.5	0.01	92.2 %
Alg. 1	0.005	0.05	61.3	0.026	92.2 %
Alg. 1	0.005	0.1	60.6	0.030	92.2 %
Alg. 1	0.005	0.2	60.1	0.035	92.2 %
Alg. 1	0.005	0.5	59.8	0.042	92.2 %
CMPC	-	-	-	-	9.7 %

column specifies the stopping condition used, Alg. 1 refers to Algorithm 1 and CMPC refers to a centralized MPC-formulation with terminal constraints which is solved by a centralized solver. The second column specifies the tolerance ϵ and the third column specifies the initial constraint tightening δ_{init} .

Columns four, five and six contain the simulation results. The results are obtained by simulating the system with 10000 randomly chosen initial conditions that are drawn from a uniform distribution on \mathcal{X} . Column four contains the mean number of iterations needed and column five presents the average constraint tightening δ used at termination of Algorithm 1. The final column shows the fraction (in %) of initial conditions that were steered to the origin using the different methods, i.e., an estimate of the region of attraction.

We see that the adaptive constraint tightening approach gives considerably less iterations for a larger initial tightening. However, for more

than 10% initial constraint tightening ($\delta_{\text{init}} = 0.1$), the number of iterations is not significantly affected. It is remarkable to note that 50% initial constraint tightening ($\delta_{\text{init}} = 0.5$) is as efficient as, e.g., 5% ($\delta_{\text{init}} = 0.05$) considering that more reductions in the constraint tightening need to be performed. This indicates early detection of infeasibility. In the final column we have estimated the region of attraction, \mathbb{X}_{rf} . We see that, for the considered example, there is a huge improvement in the region of attraction using our method without terminal constraints compared to classical MPC (CMPC) with terminal constraints.

6. Conclusions and Future Work

We have equipped the duality-based distributed optimization algorithm in [Giselsson *et al.*, 2012], when used in a DMPC context, with a stopping condition that guarantees a prespecified performance, stability and feasibility. We have used an optimization problem without terminal constraints and have shown how to verify stability and a prespecified performance. Further, we have developed an adaptive constraint tightening approach that enables us to generate a feasible solution w.r.t. the original constraint set with finite number of iterations. The numerical example shows that the region of attraction can be significantly enlarged when no terminal constraint set is used compared to when using (the maximal positive invariant) terminal constraint set as in standard MPC. Further, the numerical example shows that the adaptive constraint tightening approach can significantly reduce the number of iterations needed to guarantee feasibility, stability, and the prespecified performance.

7. Acknowledgments

The authors were supported by the Swedish Research Council through the Linnaeus center LCCC.

References

- Beck, A. and M. Teboulle (2009): “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” *SIAM J. Imaging Sciences*, **2:1**, pp. 183–202.
- Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein (2011): “Distributed optimization and statistical learning via the alternating direction method of multipliers.” *Foundations and Trends in Machine Learning*, **3:1**, pp. 1–122.
- Boyd, S. and L. Vandenberghe (2004): *Convex Optimization*. Cambridge University Press, New York, NY.
- Camponogara, E., D. Jia, B. Krogh, and S. Talukdar (2002): “Distributed model predictive control.” *IEEE Control Systems Magazine*, **22**, pp. 44–52.
- Chisci, L., A. Lombardi, and E. Mosca (1996): “Dual receding horizon control of constrained discrete-time systems.” *European Journal of Control*, **2**, pp. 278–285.
- Colson, B., P. Marcotte, and G. Savard (2005): “Bilevel programming: A survey.” *4OR: A Quarterly Journal of Operations Research*, **3:2**, pp. 87–107.
- Diehl, M., R. Findeisen, F. Allgower, H. Bock, and J. Schlöder (2005): “Nominal stability of real-time iteration scheme for nonlinear model predictive control.” *IEE Proceedings of Control Theory and Applications*, **152:3**, pp. 296–308.
- Doan, M., T. Keviczky, and B. De Schutter (2011): “A dual decomposition-based optimization method with guaranteed primal feasibility for hierarchical MPC problems.” In *Proceedings of the 18th IFAC World Congress*, pp. 392–397. Milan, Italy.
- Doan, M. D., T. Keviczky, and B. De Schutter (2010): “An improved distributed version of Han’s method for distributed MPC of canal systems.” In *Proceedings of the 12th symposium on Large Scale Systems: Theory and Applications*. Villeneuve d’Ascq, France.
- Doan, M. D., T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter (2009): “A distributed version of Han’s method for DMPC using local communications only.” *Control Engineering and Applied Informatics*, **11:3**, pp. 6–15.
- Dunbar, W. (2007): “Distributed receding horizon control of dynamically coupled nonlinear systems.” *IEEE Transactions on Automatic Control*, **52**, pp. 1249–1263.

- Dunbar, W. and R. Murray (2006): “Distributed receding horizon control for multi-vehicle formation stabilization.” *Automatica*, **42:4**, pp. 549–558.
- Gilbert, E. and K. Tan (1991): “Linear systems with state and control constraints: the theory and application of maximal output admissible sets.” *IEEE Transactions on Automatic Control*, **36:9**, pp. 1008–1020.
- Giselsson, P., M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer (2012): “Accelerated gradient methods and dual decomposition in distributed model predictive control.” *Automatica*. To appear. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Grimm, G., M. J. Messina, S. E. Tuna, and A. R. Teel (2005): “Model predictive control: for want of a local control Lyapunov function, all is not lost.” *IEEE Transactions on Automatic Control*, **50:5**, pp. 546–558.
- Grüne, L. (2009): “Analysis and design of unconstrained nonlinear MPC schemes for finite and infinite dimensional systems.” *SIAM Journal on Control and Optimization*, **48**, pp. 1206–1228.
- Grüne, L. and J. Pannek (2010): “Analysis of unconstrained NMPC schemes with incomplete optimization.” In *Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems – NOLCOS 2010*, pp. 238–243.
- Grüne, L. and A. Rantzer (2008): “On the infinite horizon performance of receding horizon controllers.” *IEEE Transactions on Automatic Control*, **53**, pp. 2100–2111.
- Jia, D. and B. Krogh (2001): “Distributed model predictive control.” In *Proceedings of the 2001 American Control Conference*, pp. 2767–2772. Arlington, VA.
- Jones, C. and M. Morari (2009): “Approximate explicit MPC using bilevel optimization.” In *European Control Conference*. Budapest, Hungary.
- Keviczky, T., F. Borrelli, and G. Balas (2006): “Decentralized receding horizon control for large scale dynamically decoupled systems.” *Automatica*, **42:12**, pp. 2105–2115.
- Lincoln, B. and A. Rantzer (2006): “Relaxing dynamic programming.” *IEEE Transactions on Automatic Control*, **51:8**, pp. 1249–1260.
- Löfberg, J. (2004): “YALMIP : A toolbox for modeling and optimization in MATLAB.” In *Proceedings of the CACSD Conference*. Taipei, Taiwan.
- Maciejowski, J. M. (2002): *Predictive Control with Constraints*. Prentice Hall, Essex, England.

- Maestre, J. M., D. Muñoz de la Peña, and E. F. Camacho (2011): “Distributed model predictive control based on a cooperative game.” *Optimal Control Applications and Methods*, **32:2**, pp. 153–176.
- Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert (2000): “Constrained model predictive control: Stability and optimality.” *Automatica*, **36:6**, pp. 789–814.
- Necoara, I., D. Doan, and J. A. K. Suykens (2008): “Application of the proximal center decomposition method to distributed model predictive control.” In *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 2900–2905. Mexico.
- Necoara, I. and J. Suykens (2008): “Application of a smoothing technique to decomposition in convex optimization.” *IEEE Transactions on Automatic Control*, **53:11**, pp. 2674–2679.
- Nedić, A. and A. Ozdaglar (2009): “Approximate primal solutions and rate analysis for dual subgradient methods.” *SIAM J. on Optimization*, **19:4**, pp. 1757–1780.
- Negenborn, R., B. De Schutter, and J. Hellendoorn (2008): “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes.” *Engineering Applications of Artificial Intelligence*, **21:3**, pp. 353–366.
- Nesterov, Y. (2003): *Introductory Lectures on Convex Optimization: A Basic Course*, 1st edition. Springer Netherlands.
- Nesterov, Y. (2005): “Smooth minimization of non-smooth functions.” *Math. Program.*, **103:1**, pp. 127–152.
- Rawlings, J. and D. Mayne (2009): *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI.
- Richards, A. and J. How (2007): “Robust distributed model predictive control.” *International Journal of Control*, **80:9**, pp. 1517–1531.
- R.M. Hermans, M. Lazar, A. J. (2010): “Almost decentralized Lyapunov-based nonlinear model predictive control.” In *Proceedings of the 29th American Control Conference*. Baltimore.
- Sandell, N., P. Varaiya, M. Athans, and M. Safonov (1978): “Survey of decentralized control methods for large scale systems.” *IEEE Transactions on Automatic Control*, **23:2**, pp. 108–128.
- Scattolini, R. (2009): “Architectures for distributed and hierarchical model predictive control - A review.” *Journal of Process Control*, **19:5**, pp. 723–731.

- Scokaert, P. O. M., D. Q. Mayne, and J. B. Rawlings (1999): “Suboptimal model predictive control (feasibility implies stability).” *IEEE Transactions on Automatic Control*, **44:3**, pp. 648–654.
- Venkat, A. N., I. A. Hiskens, J. B. Rawlings, and S. J. Wright (2008): “Distributed MPC strategies with application to power system automatic generation control.” *IEEE Transactions on Control Systems Technology*, **16:6**, pp. 1192–1206.
- Wakasa, Y., M. Arakawa, K. Tanaka, and T. Akashi (2008): “Decentralized model predictive control via dual decomposition.” In *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 381–386. Cancun, Mexico.

A. Appendix

A.1 Proof for Lemma 2

We divide the proof into two parts, the first for $\bar{x} = 0$ and the second for $\bar{x} \neq 0$. For $\bar{x} = 0$ we have at iteration $k = 0$ that $\mathbf{y}^0 = 0$ which is the optimal solution. Hence (35) holds for $k = 0$ since all terms are 0 and $0 = A\xi_{N-1}^0 \in \mathcal{X}$.

Next, we show the result for $\bar{x} \neq 0$. Whenever (12) is feasible we have convergence in primal variables [Giselsson *et al.*, 2012, Theorem 1]. This together with the linear relation through which ξ is defined (20) gives $\xi_\tau^k \rightarrow z_\tau^*$ for $\tau = 0, \dots, N-1$ as $k \rightarrow \infty$. We have $z_\tau^* \in (1-\delta)\mathcal{X}$ and since $(1-\delta)\mathcal{X} \subset \mathcal{X}$ for every $\delta \in (0, 1]$ this implies that there exists finite k_0^x such that $\xi_\tau^k \in \mathcal{X}$ for all $k \geq k_0^x$. Equivalent convergence reasoning holds for v_τ^k . Together this implies that there exists finite k_0^p such that $P_N(\bar{x}, \mathbf{v}^k) < \infty$ and that $P_N(\bar{x}, \mathbf{v}^k) \rightarrow V_N^\delta(\bar{x})$ for all $k \geq k_0^p$. Together with convergence in dual function value [Giselsson *et al.*, 2012, Theorem 1] gives that

$$D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \geq P_N(\bar{x}, \mathbf{v}^k) - \epsilon \ell^*(\bar{x})$$

holds with finite k since $\ell^*(\bar{x}) > 0$ and $\epsilon > 0$. This concludes the proof. \square

A.2 Proof for Lemma 3

We introduce $\mathbf{y}^k = [(\boldsymbol{\xi}^k(\bar{x}, \delta))^T (\mathbf{v}^k(\bar{x}, \delta))^T]^T$, where $\boldsymbol{\xi}^k(\bar{x}, \delta)$ and $\mathbf{v}^k(\bar{x}, \delta)$ satisfies the dynamic equations (20). Whenever (35) holds we have that $\xi_\tau^k(\bar{x}, \delta) \in \mathcal{X}$ and $v_\tau^k(\bar{x}, \delta) \in \mathcal{U}$ for $\tau = 0, \dots, N-1$. We also introduce $\mathbf{y}^* = [(\mathbf{z}^*(\bar{x}, 0))^T (\mathbf{v}^*(\bar{x}, 0))^T]^T$. This implies

$$\begin{aligned} \frac{1}{2}(\mathbf{y}^k - \mathbf{y}^*)^T \mathbf{H}(\mathbf{y}^k - \mathbf{y}^*) &= \frac{1}{2}(\mathbf{y}^k)^T \mathbf{H} \mathbf{y}^k - \frac{1}{2}(\mathbf{y}^*)^T \mathbf{H} \mathbf{y}^* - \langle \mathbf{H} \mathbf{y}^*, \mathbf{y}^k - \mathbf{y}^* \rangle \\ &\leq P_N(\bar{x}, \mathbf{v}^k) - V_N^0(\bar{x}) \\ &\leq D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) + \epsilon \ell^*(\bar{x}) - V_N^0(\bar{x}) \\ &\leq \delta (\boldsymbol{\mu}^k)^T \mathbf{d} + \epsilon \ell^*(\bar{x}) \end{aligned}$$

where the first inequality comes from the first order optimality condition [Nesterov, 2003, Theorem 2.2.5] and by definition of V_N^0 and P_N . The second inequality is due to (35) and the last inequality follows from Lemma 1. Further, since $\mathbf{H} = \text{blkdiag}(\mathbf{Q}, \dots, \mathbf{Q}, \mathbf{R}, \dots, \mathbf{R})$ we have for $\tau = 0, \dots, N-1$ that

$$\begin{aligned} \frac{1}{2} \left\| \begin{bmatrix} \xi_\tau^k(\bar{x}, \delta) \\ v_\tau^k(\bar{x}, \delta) \end{bmatrix} - \begin{bmatrix} z_\tau^*(\bar{x}, 0) \\ v_\tau^*(\bar{x}, 0) \end{bmatrix} \right\|_H^2 &\leq \frac{1}{2} (\mathbf{y}^k - \mathbf{y}^*)^T \mathbf{H} (\mathbf{y}^k - \mathbf{y}^*) \\ &\leq \delta (\boldsymbol{\mu}^k)^T \mathbf{d} + \epsilon \ell^*(\bar{x}) \end{aligned}$$

where $H = \text{blkdiag}(Q, R)$, whenever (35) holds. This completes the proof. \square

A.3 Proof for Lemma 4

Since $x \in \mathbb{X}_N^0$ but $x \notin \mathbb{X}_N^\delta$ we have that $V_N^0(\bar{x}) < \infty$ and $V_N^\delta(\bar{x}) = \infty$. Further, from the strong theorem of alternatives [Boyd and Vandenberghe, 2004, Section 5.8.2] we know that since $V_N^\delta(\bar{x}) = \infty$ for the current constraint tightening δ the dual problem is unbounded. Hence there exist λ_f, μ_f such that

$$\delta \mu_f^T \mathbf{d} \geq D_N^\delta(\bar{x}, \lambda_f, \mu_f) - V_N^0(\bar{x}) \geq 2\epsilon \ell^*(\bar{x}) \quad (45)$$

where Lemma 1 is used in the first inequality. Further, the convergence rate in [Beck and Teboulle, 2009, Theorem 4.4] for algorithm (16)-(19) is

$$D_N^\delta(\bar{x}, \lambda^*, \mu^*) - D_N^\delta(\bar{x}, \lambda^k, \mu^k) \leq \frac{2L}{(k+1)^2} \left\| \begin{bmatrix} \lambda^* \\ \mu^* \end{bmatrix} - \begin{bmatrix} \lambda^0 \\ \mu^0 \end{bmatrix} \right\|^2.$$

By inspecting the proof to [Beck and Teboulle, 2009, Theorem 4.4] (and [Beck and Teboulle, 2009, Lemma 2.3, Lemma 4.1]) it is concluded that the optimal point λ^*, μ^* can be changed to any feasible point λ_f, μ_f and the convergence result still holds, i.e.,

$$D_N^\delta(\bar{x}, \lambda_f, \mu_f) - D_N^\delta(\bar{x}, \lambda^k, \mu^k) \leq \frac{2L}{(k+1)^2} \left\| \begin{bmatrix} \lambda_f \\ \mu_f \end{bmatrix} - \begin{bmatrix} \lambda^0 \\ \mu^0 \end{bmatrix} \right\|^2.$$

That is, there exists a feasible pair (λ_f, μ_f) such that with finite k we have

$$D_N^\delta(\bar{x}, \lambda^k, \mu^k) > D_N^\delta(\bar{x}, \lambda_f, \mu_f) - \epsilon \ell^*(\bar{x}). \quad (46)$$

This implies

$$\delta \mathbf{d}^T \mu^k \geq D_N^\delta(\bar{x}, \lambda^k, \mu^k) - V_N^0(\bar{x}) > D_N^\delta(\bar{x}, \lambda_f, \mu_f) - V_N^0(\bar{x}) - \epsilon \ell^*(\bar{x}) \geq \epsilon \ell^*(\bar{x})$$

where Lemma 1 is used in the first inequality, (46) in the second inequality and (45) in the final inequality. This completes the proof. \square

A.4 Proof for Theorem 1

To prove the assertion we need to show that the do loop will exit for every $\bar{x} \in \text{int}(\mathbb{X}_N^0)$. For every point $\bar{x} \in \text{int}(\mathbb{X}_N^0)$ there exists $\bar{\delta} \in (0, 1)$ such that $\frac{\bar{x}}{1-\bar{\delta}} \in \text{int}(\mathbb{X}_N^0)$. Since $\text{int}(\mathbb{X}_N^0) \subseteq \mathbb{X}_N^0$, we have that $V_N^0(\frac{\bar{x}}{1-\bar{\delta}}) < \infty$ and the

optimal solution $\mathbf{y}(\frac{\bar{x}}{1-\delta}, 0)$ satisfies $\mathbf{A}\mathbf{y}^*(\frac{\bar{x}}{1-\delta}, 0) = \mathbf{b}\frac{\bar{x}}{1-\delta}$ and $\mathbf{C}\mathbf{y}^*(\frac{\bar{x}}{1-\delta}, 0) \leq \mathbf{d}$. We create the following vector

$$\bar{\mathbf{y}}(\bar{x}) := (1 - \bar{\delta})\mathbf{y}^*\left(\frac{\bar{x}}{1 - \bar{\delta}}, 0\right) \quad (47)$$

which satisfies

$$\mathbf{A}\bar{\mathbf{y}}(\bar{x}) = \mathbf{A}\mathbf{y}^*\left(\frac{\bar{x}}{1 - \bar{\delta}}, 0\right)(1 - \bar{\delta}) = \mathbf{b}\bar{x}\frac{1 - \bar{\delta}}{1 - \bar{\delta}} = \mathbf{b}\bar{x} \quad (48)$$

$$\mathbf{C}\bar{\mathbf{y}}(\bar{x}) = \mathbf{C}\mathbf{y}^*\left(\frac{\bar{x}}{1 - \bar{\delta}}, 0\right)(1 - \bar{\delta}) \leq \mathbf{d}(1 - \bar{\delta}). \quad (49)$$

Hence, by definition (32) of \mathbb{X}_N^δ we conclude that for every $\bar{x} \in \text{int}(\mathbb{X}_N^0)$ there exist $\bar{\delta} \in (0, 1)$ such that $\bar{x} \in \mathbb{X}_N^{\bar{\delta}}$. This implies that for every $\bar{x} \in \text{int}(\mathbb{X}_N^0)$ we have that either $\bar{x} \in \mathbb{X}_N^{\bar{\delta}}$ for the current constraint tightening $\bar{\delta} \in (0, 1)$ or $\bar{x} \notin \mathbb{X}_N^{\bar{\delta}}$ but $\bar{x} \in \mathbb{X}_N^0$. Thus, from Lemma 2 and Lemma 4 we conclude that either the do loop is terminated or $\bar{\delta}$ is reduced and l is increased for every $\bar{x} \in \text{int}(\mathbb{X}_N^0)$ with finite number of algorithm iterations k .

To guarantee that the do loop will terminate for every $\bar{x} \in \text{int}(\mathbb{X}_N^0)$, we need to show that the conditions in the do loop will hold for small enough $\bar{\delta}$ and with finite k . That is, we need to show that the following two conditions will hold.

1. For small enough $\bar{\delta}$, i.e., large enough l , we have that

$$\delta(\boldsymbol{\mu}^k)^T \mathbf{d} \leq \epsilon \ell^*(\bar{x}) \quad (50)$$

where $\delta = 2^{-l} \delta_{\text{init}}$ holds for every algorithm iteration k .

2. For small enough $\bar{\delta}$, i.e., large enough l , the condition

$$D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \geq P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + \alpha \ell(\bar{x}, v_0^k) \quad (51)$$

with α satisfying (36) holds with finite k whenever

$$D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \geq P_N(\bar{x}, \mathbf{v}^k) + \frac{\epsilon}{l+1} \ell(\bar{x}, v_0^k) \quad (52)$$

holds.

We start by showing argument 1. From the convergence rate of the algorithm [Giselsson *et al.*, 2012] it follows that there exists $\underline{D} > -\infty$ such that $D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \geq \underline{D}$ for every algorithm iteration $k \geq 0$. This is used below where we extend the result from [Nedić and Ozdaglar, 2009,

Lemma 1] to handle the presence of equality constraints. For algorithm iteration $k \geq 0$, $\bar{x} \in \text{int}(\mathbb{X}_N^0)$ and $\delta \leq \bar{\delta}/2$ we have

$$\begin{aligned}
 \underline{D} &\leq D_N^\delta(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \\
 &= \inf_{\mathbf{y}} \frac{1}{2} \mathbf{y}^T \mathbf{H} \mathbf{y} + (\boldsymbol{\lambda}^k)^T (\mathbf{A} \mathbf{y} - \mathbf{b} \bar{x}) + (\boldsymbol{\mu}^k)^T (\mathbf{C} \mathbf{y} - (1 - \delta) \mathbf{d}) \\
 &\leq \frac{1}{2} (\bar{\mathbf{y}}(\bar{x}))^T \mathbf{H} \bar{\mathbf{y}}(\bar{x}) + (\boldsymbol{\lambda}^k)^T (\mathbf{A} \bar{\mathbf{y}}(\bar{x}) - \mathbf{b} \bar{x}) + (\boldsymbol{\mu}^k)^T (\mathbf{C} \bar{\mathbf{y}}(\bar{x}) - (1 - \delta) \mathbf{d}) \\
 &\leq (1 - \bar{\delta})^2 V_N^0\left(\frac{\bar{x}}{1 - \bar{\delta}}\right) + (\boldsymbol{\mu}^k)^T (\mathbf{C} \bar{\mathbf{y}}(\bar{x}) - (1 - \bar{\delta}) \mathbf{d}) + (\boldsymbol{\mu}^k)^T \mathbf{d} (\delta - \bar{\delta}) \\
 &\leq V_N^0\left(\frac{\bar{x}}{1 - \bar{\delta}}\right) + (\boldsymbol{\mu}^k)^T \mathbf{d} (\delta - \bar{\delta}) \\
 &\leq V_N^0\left(\frac{\bar{x}}{1 - \bar{\delta}}\right) - \frac{1}{2} (\boldsymbol{\mu}^k)^T \mathbf{d} \bar{\delta}
 \end{aligned}$$

where the equality is by definition, the second inequality holds since any vector $\bar{\mathbf{y}}(\bar{x})$ is gives larger value than the infimum, the third and fourth inequalities are due to (47), (48) and (49) and since $(1 - \bar{\delta}) \in (0, 1)$ and the final inequality holds since $\delta \leq \bar{\delta}/2$. This implies that

$$(\boldsymbol{\mu}^k)^T \mathbf{d} \leq \frac{2(V_N^0(\frac{\bar{x}}{1 - \bar{\delta}}) - \underline{D})}{\bar{\delta}}$$

which is finite. We denote by l_d the smallest l such that $\bar{\delta} \geq 2^{-l_d} \delta_{\text{init}}$. Since $\delta = 2^{-l} \delta_{\text{init}}$ this implies that

$$\begin{aligned}
 \delta (\boldsymbol{\mu}^k)^T \mathbf{d} &\leq \delta \frac{2(V_N^0(\frac{\bar{x}}{1 - \bar{\delta}}) - \underline{D})}{\bar{\delta}} \leq 2^{-l} \delta_{\text{init}} \frac{2(V_N^0(\frac{\bar{x}}{1 - \bar{\delta}}) - \underline{D})}{2^{-l_d} \delta_{\text{init}}} \\
 &\leq 2^{-l+l_d+1} (V_N^0(\frac{\bar{x}}{1 - \bar{\delta}}) - \underline{D}) \rightarrow 0
 \end{aligned} \tag{53}$$

as $l \rightarrow \infty$. Especially, with finite l we have that (50) holds for every algorithm iteration k . This proves argument 1.

Next we prove argument 2. We start by showing for large enough but finite l that $P^N(A\bar{x} + B\mathbf{v}_N(\bar{x}), \mathbf{v}_s^k)$ is finite whenever (52) holds. From the definition of P_N and \mathbf{v}_s^k we have that $P^N(A\bar{x} + B\mathbf{v}_N(\bar{x}), \mathbf{v}_s^k)$ is finite whenever $P_N(\bar{x}, \mathbf{v}_s^k)$ is finite and if $A_{\zeta_{N-1}^k}(\bar{x}, \delta) \in \mathcal{X}$. For algorithm iteration k

such that (52) holds we have

$$\begin{aligned}
& \|A(\xi_{N-1}^k(\bar{x}, \delta) - z_{N-1}^*(\bar{x}, 0))\|^2 \leq \\
& \leq \frac{\|A\|^2}{\lambda_{\min}(H)} \|\xi_{N-1}^k(\bar{x}, \delta) - z_{N-1}^*(\bar{x}, 0)\|_H^2 \\
& \leq \frac{2\|A\|^2}{\lambda_{\min}(H)} (\delta(\boldsymbol{\mu}^k)^T \mathbf{d} + \frac{\epsilon}{l+1} \ell^*(\bar{x})) \\
& \leq \frac{2\|A\|^2}{\lambda_{\min}(H)} \left(2^{-l+l_d+1} (V_N^0(\frac{\bar{x}}{1-\delta}) - \underline{D}) + \frac{\epsilon}{l+1} \ell^*(\bar{x}) \right) \rightarrow 0 \quad (54)
\end{aligned}$$

as $l \rightarrow \infty$, where $H = \text{blkdiag}(Q, R)$ and, since H is positive definite, the smallest eigenvalue $\lambda_{\min}(H) > 0$. The first inequality follows from Cauchy-Schwarz inequality and Courant-Fischer-Weyl min-max principle, the second inequality comes from Lemma 3 and the third comes from (53). By definition of \mathbb{X}_N^δ we have $Az_{N-1}^*(\bar{x}, 0) \in \text{int}(\mathcal{X})$ which through (54) implies that $A\xi_{N-1}^k(\bar{x}, \delta) \in \mathcal{X}$ for some large enough by finite l , i.e., small enough δ , and for algorithm iteration k such that (52) holds.

What is left to show argument 2 is that (51) holds for every $\alpha \leq 1 - \epsilon - \kappa(\sqrt{2\epsilon} + \sqrt{\Phi_N})^2(\sqrt{2\epsilon} + 1)^2$ for large enough but finite l whenever (52) holds. From Lemma 3 and (53) we know for large enough l and any algorithm iteration k such that (52) holds that

$$\begin{aligned}
\frac{1}{2} \left\| \begin{bmatrix} \xi_\tau^k \\ v_\tau^k \end{bmatrix} - \begin{bmatrix} z_\tau^* \\ v_\tau^* \end{bmatrix} \right\|_H^2 & \leq \delta(\boldsymbol{\mu}^k)^T \mathbf{d} + \frac{\epsilon}{l+1} \ell^*(\bar{x}) \\
& = 2^{-l} \delta_{\text{init}}(\boldsymbol{\mu}^k)^T \mathbf{d} + \frac{\epsilon}{l+1} \ell^*(\bar{x}) \leq \epsilon \ell^*(\bar{x})
\end{aligned}$$

for any $\tau = 0, \dots, N-1$, where $H = \text{blkdiag}(Q, R)$. Taking the square-root and applying the reversed triangle inequality gives

$$\left| \left\| \begin{bmatrix} \xi_\tau^k \\ v_\tau^k \end{bmatrix} \right\|_H - \left\| \begin{bmatrix} z_\tau^* \\ v_\tau^* \end{bmatrix} \right\|_H \right| \leq \left\| \begin{bmatrix} \xi_\tau^k \\ v_\tau^k \end{bmatrix} - \begin{bmatrix} z_\tau^* \\ v_\tau^* \end{bmatrix} \right\|_H \leq 2\sqrt{\epsilon \ell^*(\bar{x})}. \quad (55)$$

This implies that

$$\begin{aligned}
 \left\| \begin{bmatrix} \xi_{N-1}^k \\ v_{N-1}^k \end{bmatrix} \right\|_H &\leq \left\| \begin{bmatrix} z_{N-1}^* \\ v_{N-1}^* \end{bmatrix} \right\|_H + 2\sqrt{\epsilon \ell^*(\bar{x})} = \sqrt{2} \sqrt{\ell(z_{N-1}^*, v_{N-1}^*)} + 2\sqrt{\epsilon \ell^*(\bar{x})} \\
 &\leq \sqrt{2\Phi_N} \sqrt{\ell(z_0^*, v_0^*)} + 2\sqrt{\epsilon \ell^*(\bar{x})} \leq (\sqrt{2\Phi_N} + 2\sqrt{\epsilon}) \sqrt{\ell(z_0^*, v_0^*)} \\
 &= (\sqrt{\Phi_N} + \sqrt{2\epsilon}) \left\| \begin{bmatrix} z_0^* \\ v_0^* \end{bmatrix} \right\|_H \\
 &\leq (\sqrt{\Phi_N} + \sqrt{2\epsilon}) \left(\left\| \begin{bmatrix} \xi_0^k \\ v_0^k \end{bmatrix} \right\|_H + 2\sqrt{\epsilon \ell^*(\bar{x})} \right) \\
 &\leq (\sqrt{\Phi_N} + \sqrt{2\epsilon})(1 + \sqrt{2\epsilon}) \left\| \begin{bmatrix} \xi_0^k \\ v_0^k \end{bmatrix} \right\|_H
 \end{aligned}$$

where we have used (55), $z_0^* = \xi_0^k = \bar{x}$, $\|z^T v^T\|_H = \sqrt{z^T Q z + v^T R v} = \sqrt{2\ell(z, v)}$ and Definition 1. Squaring both sides gives through the definition of κ that

$$\begin{aligned}
 \frac{1}{\kappa} \ell^*(A \xi_{N-1}^k) &\leq \ell^*(\xi_{N-1}^k) = \ell(\xi_{N-1}^k, v_{N-1}^k) \\
 &\leq (\sqrt{\Phi_N} + \sqrt{2\epsilon})^2 (1 + \sqrt{2\epsilon})^2 \ell(\xi_0^k, v_0^k). \tag{56}
 \end{aligned}$$

We get for large enough l and for k such that (52) holds that

$$\begin{aligned}
 D_N^\delta(\bar{x}, \lambda^k, \mu^k) &\geq P_N(\bar{x}, \mathbf{v}^k) - \frac{\epsilon}{l+1} \ell^*(\bar{x}) \geq P_N(\bar{x}, \mathbf{v}^k) - \epsilon \ell^*(\bar{x}) \\
 &= P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + (1-\epsilon)\ell(\xi_0^k, v_0^k) - \ell^*(A\xi_{N-1}^k) \\
 &\geq P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + \\
 &\quad + \left(1 - \epsilon - \kappa(\sqrt{\Phi_N} + \sqrt{2\epsilon})^2(1 + \sqrt{2\epsilon})^2\right) \ell(\bar{x}, v_0^k) \\
 &\geq P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) + \alpha \ell(\bar{x}, v_0^k) \tag{57}
 \end{aligned}$$

where the first inequality comes from (52), the second since $l \geq 0$, the equality is due to (22), the third inequality comes from (56), and the final inequality comes from (36). This concludes the proof for argument 2. Thus, the do loop will terminate with finite l and k which implies that $v_N(\bar{x})$ is well defined for every $\bar{x} \in \text{int}(\mathbb{X}_N^0)$.

Finally, to show (37) we have that

$$\begin{aligned}
 V_N^0(\bar{x}) &\geq D_N^\delta(\bar{x}, \lambda^k, \mu^k) - \delta \mathbf{d}^T \mu^k \\
 &\geq P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) - \epsilon \ell^*(\bar{x}) + \alpha \ell(\bar{x}, v_0^k) \\
 &\geq V_N^0(A\bar{x} + Bv_0^k) + (\alpha - \epsilon) \ell(\bar{x}, v_0^k)
 \end{aligned}$$

where the first inequality comes from Lemma 1, the second from (50) and (51), and the third holds since $P_N(A\bar{x} + Bv_0^k, \mathbf{v}_s^k) \geq V_N(A\bar{x} + Bv_0^k)$ and by definition of ℓ^* . This concludes the proof. \square

Paper VI

Output Feedback Distributed Model Predictive Control with Inherent Robustness Properties

Pontus Giselsson

Abstract

We consider robust output feedback distributed model predictive control (DMPC). The proposed controller is based on the results in [Giselsson and Rantzer, 2012] in which nominal stability and feasibility was proven for a DMPC formulation without terminal constraint set or terminal cost in the optimization. We extend these results to show robust stability under state feedback as well as output feedback when dynamics and measurements are affected by bounded noise. The provided numerical example suggests that the region of attraction without terminal constraint set may be significantly larger than if a terminal constraint set is used.

Submitted to *2013 American Control Conference*, Washington, D.C., 2013.

1. Introduction

In the model predictive control (MPC) literature nominal stability of the closed loop system is a well studied subject and is usually proven using a terminal constraint set, a terminal cost and a terminal controller, see [Mayne *et al.*, 2000] for a survey of such methods. Also robustness properties in MPC has received increased attention. In [Mayne *et al.*, 2000] different approaches from the literature to achieve robustness are presented. The survey shows three main approaches to guarantee robustness in MPC: to exploit the inherent robustness in nominal MPC, to design the MPC controller to deal with any possible realization of the disturbance, or to introduce feedback in the design that compensates for the disturbances. Within the first category, it was shown in [Grimm *et al.*, 2004] that linear systems with convex constraints are inherently robust to small disturbances. This is due to the fact that the value function of the optimization problem is continuous [Bemporad *et al.*, 2002, Grimm *et al.*, 2004]. To address both robust feasibility and robust stability, a tube-based model predictive controller for linear systems was presented in [Mayne *et al.*, 2005]. This was extended to tube-based output feedback model predictive control in [Mayne *et al.*, 2006]. These tube-based MPC controllers also rely on a terminal cost and terminal constraints to show stability.

It was pointed out in [Giselsson and Rantzer, 2012] that terminal costs, terminal constraint sets, and terminal controllers usually involve all decision variables and are therefore not directly applicable for distributed model predictive control formulations where a centralized optimization problem is solved in distributed fashion. This is circumvented in [Doan *et al.*, 2009] where stability is proven by setting a terminal point constraint in the origin, which is not desirable for performance and region of attraction reasons. In [Giselsson and Rantzer, 2012] a DMPC controller based on an optimization problem without terminal constraint set or terminal cost is proposed. Nominal stability for this is shown based on a controllability assumption on the optimal stage costs. Another formulation that solves a centralized MPC problem in distributed fashion can be found in [Negenborn *et al.*, 2008] but no stability guarantees are given. In the DMPC literature some formulations do not solve a centralized problem but local optimization problems that take neighboring interaction into account, [Dunbar, 2007, Richards and How, 2007, R.M. Hermans, 2010]. In [Dunbar, 2007, Richards and How, 2007] stability (and robustness in the latter case) is guaranteed by letting the subsystems solve local optimization problems sequentially and pass the local solutions downstream to be used in the remaining local optimizations. In [R.M. Hermans, 2010] stability is shown by setting explicit stabilizing constraints in the optimization. In the case of output feedback, there are quite few contributions in the

DMPC literature. One exception is [Venkat *et al.*, 2006] in which nominal stability is proven using a decentralized estimator and local optimizations with full model data.

In this paper we extend the DMPC formulation presented in [Giselsson and Rantzer, 2012] to guarantee robustness to small disturbances using a constraint tightening approach and the inherent robustness of linear MPC. In [Giselsson and Rantzer, 2012] stability is shown without the use of a terminal constraint set which in many applications increases the region of attraction since there are no constraints on the end point. Using ideas from [Mayne *et al.*, 2006] we also propose an output feedback DMPC controller that is shown to be robustly stable and robustly feasible for small disturbances. Stability is shown by containing the estimation error within a positively robust invariant set and view the estimation error as a (bounded) disturbance. The inherent robustness of linear MPC is then used to show robust stability. To cope with the output feedback case, we restrict our treatment to systems with input couplings only since this allows for decentralized observer design. Such system descriptions arise, for instance, when flow between subsystems is controlled. The flow might be power in an electric network [Almassalkhi and Hiskens, 2011], water in hydro power valley [Petrone, 2010] or intermediate products in a supply chain [Dunbar and Desa, 2005].

The paper is organized as follows. In Section 2 we formulate the problem and present useful results from [Giselsson and Rantzer, 2012]. In Section 3 we show robust stability and robust feasibility in the state feedback case. These results are used in Section 4 to show robust stability and feasibility in the output feedback case. A numerical example is provided in Section 5 and the paper is concluded in Section 6.

2. Setup and Preliminaries

We consider linear dynamical systems, where each subsystem $i = 1, \dots, M$ is described by

$$\begin{aligned} x_{t+1}^i &= A_{ii}x_t^i + \sum_{j \in \mathcal{N}_i} B_{ij}u_t^j + w_t^i & x_0^i &= \bar{x}^i \\ y_t^i &= C_i x_t^i + \xi_t^i \end{aligned}$$

where $x_t^i \in \mathbb{R}^{n_i}$, $u_t^i \in \mathbb{R}^{m_i}$, $w_t^i \in \mathbb{R}^{n_i}$, $y_t^i \in \mathbb{R}^{p_i}$, $\xi_t^i \in \mathbb{R}^{p_i}$, and \mathcal{N}_i is the neighboring interaction defined by

$$\mathcal{N}_i = \{j \in \{1, \dots, M\} \mid B_{ij} \neq 0\}.$$

2. Setup and Preliminaries

We assume that the system has some sparsity structure, i.e., that some $B_{ij} = 0$. We introduce the global variables

$$\begin{aligned} x &= [(x^1)^T, \dots, (x^M)^T]^T, & u &= [(u^1)^T, \dots, (u^M)^T]^T, \\ w &= [(w^1)^T, \dots, (w^M)^T]^T, & y &= [(y^1)^T, \dots, (y^M)^T]^T, \\ \xi &= [(\xi^1)^T, \dots, (\xi^M)^T]^T \end{aligned}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $w \in \mathbb{R}^n$, $y \in \mathbb{R}^p$ and $\xi \in \mathbb{R}^p$. This gives the following global system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_t & x_0 &= \bar{x} & (1) \\ y_t &= Cx_t + \xi_t & & & (2) \end{aligned}$$

where the matrices A and C are block-diagonal and B is sparse. We assume hereafter that the pair (A, B) is stabilizable and the pair (A, C) is detectable. The local control and state variables as well as the disturbances are constrained, i.e., $u^i \in \mathcal{U}_i$, $x^i \in \mathcal{X}_i$, $w^i \in \mathcal{W}_i$ and $\xi^i \in \mathcal{E}_i$ where

$$\begin{aligned} \mathcal{X}_i &= \{x^i \in \mathbb{R}^{n_i} \mid F_i^x x^i \leq g_i^x\}, & \mathcal{U}_i &= \{u^i \in \mathbb{R}^{m_i} \mid F_i^u u^i \leq g_i^u\}, \\ \mathcal{W}_i &= \{w^i \in \mathbb{R}^{n_i} \mid F_i^w w^i \leq g_i^w\}, & \mathcal{E}_i &= \{\xi^i \in \mathbb{R}^{p_i} \mid F_i^\xi \xi^i \leq g_i^\xi\} \end{aligned}$$

where $F_i^x \in \mathbb{R}^{n_{f_x^i} \times n_i}$, $g_i^x \in \mathbb{R}^{n_{f_x^i}}$, $F_i^u \in \mathbb{R}^{n_{f_u^i} \times m_i}$, $g_i^u \in \mathbb{R}^{n_{f_u^i}}$, $F_i^w \in \mathbb{R}^{n_{f_w^i} \times n_i}$, $g_i^w \in \mathbb{R}^{n_{f_w^i}}$, $F_i^\xi \in \mathbb{R}^{n_{f_\xi^i} \times p_i}$ and $g_i^\xi \in \mathbb{R}^{n_{f_\xi^i}}$. We denote the total number of inequalities in \mathcal{X}_i and \mathcal{U}_i for all $i = 1, \dots, M$ by q , i.e., $q = \sum_i (n_{f_x^i} + n_{f_u^i})$. The global constraint sets $\mathcal{X}, \mathcal{U}, \mathcal{W}$ and \mathcal{E} are defined as the set product of their respective local constraint sets. By introducing the predicted state and control vectors

$$\mathbf{z} = [z_0^T, \dots, z_{N-1}^T]^T \quad \mathbf{v} = [v_0^T, \dots, v_{N-1}^T]^T \quad (3)$$

we formulate the following optimization problem which was used in the DMPC formulation in [Giselsson and Rantzer, 2012]

$$\begin{aligned} V_N(x) &:= \min_{\mathbf{z}, \mathbf{v}} J_N(\mathbf{z}, \mathbf{v}) & (4) \\ \text{s.t.} & \quad z_\tau \in \mathcal{X}, & \tau &= 0, \dots, N-1, \\ & \quad v_\tau \in \mathcal{U}, & \tau &= 0, \dots, N-1, \\ & \quad z_{\tau+1} = Az_\tau + Bv_\tau, & \tau &= 0, \dots, N-2, \\ & \quad z_0 = x. \end{aligned}$$

We denote the optimal state and control at time step τ for (4) by $z_\tau^*(x)$ and $v_\tau^*(x)$ respectively. The cost in (4) is assumed quadratic and separable

$$\begin{aligned} J_N(\mathbf{z}, \mathbf{v}) &:= \sum_{\tau=0}^{N-1} \ell(z_\tau, v_\tau) = \sum_{\tau=0}^{N-1} \sum_{i=1}^M \ell_i(z_\tau^i, v_\tau^i) \\ &= \sum_{\tau=0}^{N-1} \sum_{i=1}^M \left(\frac{1}{2} (z_\tau^i)^T \mathbf{Q}_i z_\tau^i + \frac{1}{2} (v_\tau^i)^T \mathbf{R}_i v_\tau^i \right) \end{aligned}$$

where $\mathbf{Q}_i \succ 0$ and $\mathbf{R}_i \succ 0$. Problem (4) can be solved efficiently in distributed fashion using the method developed in [Giselsson *et al.*, 2012] which was also used in [Giselsson and Rantzer, 2012]. A short description of the optimization algorithm is given below. By introducing the vector $\boldsymbol{\chi} = [\mathbf{z}^T, \mathbf{v}^T]^T$ the optimization problem (4) can more compactly be written as

$$\begin{aligned} V_N(\bar{x}) &:= \min_{\boldsymbol{\chi}} \quad \frac{1}{2} \boldsymbol{\chi}^T H \boldsymbol{\chi} \\ &\text{s.t.} \quad \mathbf{A} \boldsymbol{\chi} = \mathbf{b} x \\ &\quad \mathbf{F} \boldsymbol{\chi} \leq \mathbf{g} \end{aligned}$$

where H and \mathbf{F} are block-diagonal and \mathbf{A} has the same structure as B in (1). We introduce dual variables $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{Ng}$ for the inequality constraints and $\boldsymbol{\lambda} \in \mathbb{R}^{n(N-1)}$ for the equality constraints. As shown in [Giselsson *et al.*, 2012] the dual problem can be written as

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} -\frac{1}{2} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{F}^T \boldsymbol{\mu})^T H^{-1} (\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{F}^T \boldsymbol{\mu}) - \boldsymbol{\lambda}^T \mathbf{b} x - \boldsymbol{\mu}^T \mathbf{g}. \quad (5)$$

The dual function was in [Giselsson *et al.*, 2012] shown to have Lipschitz continuous gradient with Lipschitz constant $L = \|[\mathbf{A}^T \ \mathbf{F}^T]^T H^{-1} [\mathbf{A}^T \ \mathbf{F}^T]\|$ and can hence be maximized using accelerated gradient methods. The algorithm from [Giselsson *et al.*, 2012] is presented here

$$\boldsymbol{\chi}^k = -H^{-1} \left(\mathbf{F}^T \boldsymbol{\mu}^k + \mathbf{A}^T \boldsymbol{\lambda}^k \right) \quad (6)$$

$$\bar{\boldsymbol{\chi}}^k = \boldsymbol{\chi}^k + \frac{k-1}{k+2} (\boldsymbol{\chi}^k - \boldsymbol{\chi}^{k-1}) \quad (7)$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \frac{k-1}{k+2} (\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) + \frac{1}{L} (\mathbf{A} \bar{\boldsymbol{\chi}}^k - \mathbf{b} x) \quad (8)$$

$$\boldsymbol{\mu}^{k+1} = \max \left[0, \boldsymbol{\mu}^k + \frac{k-1}{k+2} (\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}) + \frac{1}{L} (\mathbf{F} \bar{\boldsymbol{\chi}}^k - \mathbf{g}) \right] \quad (9)$$

where k denotes the iteration number. Due to the structure of the matrices \mathbf{A} , \mathbf{F} and H the algorithm can be implemented in distributed fashion

where communication between subsystems i and j takes place if $j \in \mathcal{N}_i$ or $i \in \mathcal{N}_j$, see [Giselsson *et al.*, 2012] for details. Further results from [Giselsson *et al.*, 2012] shows that the algorithm converges as $O(\frac{1}{k^2})$ in dual function value. This is a significant enhancement compared to if the classical gradient method was used which converges as $O(\frac{1}{k})$.

In [Giselsson and Rantzer, 2012] feasibility, stability and performance of the closed loop system when solving (4), which has neither terminal cost nor terminal constraints, using (6)-(9) was established. Since (6)-(9) gives a primal feasible solution only in the limit of iterations, an adaptive constraint tightening approach was used to ensure feasibility, stability, and performance with finite number of algorithm iterations. However, in this paper we state all results as if the optimal solution to (6)-(9) is found in each iteration. The generalization to allow for early termination using the stopping condition in [Giselsson and Rantzer, 2012] is straightforward but requires quite some notation to be introduced. We introduce

$$\mathbb{X}_N := \{x \in \mathbb{R}^n \mid V_N(x) < \infty \text{ and } Az_{N-1}^*(x) \in \mathcal{X}\}.$$

We also define the infinite horizon steerable set

$$\mathbb{X}_\infty := \{x \in \mathbb{R}^n \mid V_\infty(x) < \infty\}$$

and the following definition.

DEFINITION 1

The constant Φ_N is the smallest constant such that the optimal solution $\{z_\tau^*(x)\}_{\tau=0}^{N-1}$, $\{v_\tau^*(x)\}_{\tau=0}^{N-1}$ to (4) for given N and every $x \in \mathbb{X}_N$ satisfies

$$\ell^*(z_{N-1}^*(x)) \leq \Phi_N \ell(x, v_0^*(x)). \quad (10)$$

□

We introduce the optimal feedback control law $v_N(x) := v_0^*(x)$ and define the nominal and actual next states

$$\begin{aligned} \bar{x}_{t+1} &:= Ax_t + Bv_N(x_t) \\ x_{t+1} &:= Ax_t + Bv_N(x_t) + w_t \end{aligned}$$

where $w_t \in \mathcal{W}$. We define $\kappa = \|Q^{-1/2}A^TQAQ^{-1/2}\|$ and state the following result from [Giselsson and Rantzer, 2012, Corollary 1].

THEOREM 1

Suppose that $\alpha \leq 1 - \kappa\Phi_N$. Then

$$V_N(x) \geq V_N(Ax + Bv_N(x)) + \alpha\ell(x, v_N(x))$$

holds for every $x \in \mathbb{X}_N$. □

Throughout the remainder of the paper we assume that $\alpha > 0$ and N are chosen in accordance with Theorem 1.

ASSUMPTION 1

We assume that the disturbance sets \mathcal{W}, \mathcal{E} are bounded and that $0 \in \text{int}\mathcal{W}$, $0 \in \text{int}\mathcal{E}$. Further we assume that $\mathcal{B}_\infty^{n_i}(r_x) \subset \mathcal{X}_i$, $\mathcal{B}_\infty^{m_i}(r_u) \subset \mathcal{U}_i$ for some $r_x, r_u > 0$ where $\mathcal{B}_\infty^n(r)$ is defined in (11). □

2.1 Notation

We denote by \mathbb{R} the real numbers and by $\mathbb{R}_{\geq 0}$ non-negative real numbers. The norm $\|\cdot\|$ refers to the Euclidean norm or the induced Euclidean norm unless otherwise is specified. The norm ball is defined as

$$\mathcal{B}_l^n(r) := \{x \in \mathbb{R}^n \mid \|x\|_l \leq r\}. \quad (11)$$

The \oplus denotes the Minkowski sum defined by $\mathcal{X}_1 \oplus \mathcal{X}_2 \triangleq \{x_1 + x_2 \mid x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2\}$ and \ominus denotes the Pontryagin difference defined by

$$\mathcal{X}_1 \ominus \mathcal{X}_2 \triangleq \{x \mid \{x\} \oplus \mathcal{X}_2 \subseteq \mathcal{X}_1\}. \quad (12)$$

Finally $h_{\mathcal{X}}(\theta)$ is the support function which is defined as

$$h_{\mathcal{X}}(\theta) \triangleq \sup_{x \in \mathcal{X}} \theta^T x.$$

REMARK 1

For polytopic sets $\mathcal{X}_1 = \{x \in \mathbb{R}^n \mid \mathbf{X}_1 x \leq y_1\}$, $\mathcal{X}_2 = \{x \in \mathbb{R}^n \mid \mathbf{X}_2 x \leq y_2\}$ we have from [Kolmanovsky and Gilbert, 1998, Theorem 2.3] that

$$\mathcal{X}_1 \ominus \mathcal{X}_2 = \{x \in \mathbb{R}^n \mid [X_1]_j x \leq [y_1]_j - h_{\mathcal{X}_2}([X_1]_j^T), \quad j = 1, \dots, p\}$$

where X_1 has p rows, $[X_1]_j$ is the j :th row of X_1 and $[y_1]_j$ is the j :th element of y_1 . Thus, $\mathcal{X}_1 \ominus \mathcal{X}_2$ and \mathcal{X}_1 can be described using the same number of linear inequalities. □

3. Robust State Feedback DMPC

In this section we consider the state feedback problem, i.e., with $C = I$ and $\xi = 0$ in (2). We will see that by tightening the constraints in the optimization problem we can guarantee robust stability and robust feasibility. We start by investigating robust feasibility.

3.1 One-step Robust Feasibility

To guarantee that the system is one-step robustly feasible, a constraint tightening approach is used. We introduce the sets $\mathcal{X}_i \ominus \mathcal{W}_i$ for $i = 1, \dots, M$ which can be computed as in Remark 1. Since the number of constraints that describes $\mathcal{X}_i \ominus \mathcal{W}_i$ is the same as the number of constraints that describes \mathcal{X}_i , these tightened constraint sets can be used in the optimization without increasing the complexity. Defining the corresponding global constraint set $\mathcal{X} \ominus \mathcal{W}$ as the set product of the local sets, we get the following optimization problem with tightened constraints

$$\begin{aligned}
 V_N(x) := \min_{\mathbf{z}, \mathbf{v}} \quad & J_N(\mathbf{z}, \mathbf{v}) & (13) \\
 \text{s.t.} \quad & z_\tau \in \mathcal{X} \ominus \mathcal{W}, & \tau = 0, \dots, N-1, \\
 & v_\tau \in \mathcal{U}, & \tau = 0, \dots, N-1, \\
 & z_{\tau+1} = Az_\tau + Bv_\tau, & \tau = 0, \dots, N-2, \\
 & z(0) = x.
 \end{aligned}$$

The state constraint set is changed in (13) compared to in (4). Thus, we get a different control law v_N , infinite horizon steerable set \mathbb{X}_∞ , set \mathbb{X}_N , and value function V_N . To avoid introducing new notation we use the same notation but the quantities are in this section based on optimization problem (13) instead of (4). The following proposition shows one-step robust feasibility.

PROPOSITION 1

For any $x_t \in \mathbb{X}_N$ we have that $x_{t+1} \in \mathcal{X}$ for any disturbance $w_t \in \mathcal{W}$. \square

PROOF

From the problem formulation we have that $\bar{x}_{t+1} \in \mathcal{X} \ominus \mathcal{W}$. From [Kolmanovsky and Gilbert, 1998, Theorem 2.1] we know that $(\mathcal{X} \ominus \mathcal{W}) \oplus \mathcal{W} \subseteq \mathcal{X}$. Further, $x_{t+1} = \bar{x}_{t+1} + w_t \in (\mathcal{X} \ominus \mathcal{W}) \oplus \mathcal{W} \subseteq \mathcal{X}$. This concludes the proof. \square

This shows that if the optimization problem is feasible, we get one-step robust feasibility.

3.2 Robust Stability

For systems with linear dynamics, quadratic cost and polytopic constraints we know that the value function is continuous [Grimm *et al.*, 2004, Bemporad *et al.*, 2002]. Thus, for every $x \in \mathbb{X}_N \ominus \mathcal{W}$ we have for some finite $\beta_w \geq 0$ that

$$\max_{w \in \mathcal{W}} V_N(x+w) - V_N(x) \leq \beta_w \quad (14)$$

since $x+w \in \mathbb{X}_N$ for any $x \in \mathbb{X}_N \ominus \mathcal{W}$ and $w \in \mathcal{W}$. This observation is used to prove inherent robustness of the closed loop system to small disturbance sets \mathcal{W} . To show robust stability we need to introduce some sets. The first is the following ellipsoid

$$\mathcal{E}(\gamma) := \{x \in \mathbb{R}^n \mid (\alpha - \epsilon)\ell^*(x) \leq \gamma\} \quad (15)$$

where $\ell^*(x) = \frac{1}{2}x^T Qx$, $\epsilon > 0$ is small and $\alpha > \epsilon$ is from Theorem 1. The second is the value function level sets

$$\Omega(c) := \{x \in \mathbb{R}^n \mid V_N(x) \leq c\}.$$

We also introduce the following recursive definition of the maximal positively robust invariant set

$$\mathbb{X}_{\text{rf}} = \{x \in \mathbb{X}_N \mid \{Ax + Bv_N(x)\} \oplus \mathcal{W} \subseteq \mathbb{X}_{\text{rf}}\}.$$

Before we state the theorem about asymptotic convergence, we need the following assumption.

ASSUMPTION 2

We assume that the disturbance set \mathcal{W} is small enough to guarantee $\Omega(\delta) \subset \mathbb{X}_{\text{rf}}$ where $\delta = 2 \max_{x \in \mathcal{E}(\beta_w)} V_N(x)$. \square

THEOREM 2

Suppose that Assumption 2 holds. Then for any initial condition $x_0 \in \mathbb{X}_{\text{rf}}$, the closed loop system is asymptotically converging to $\Omega(\delta)$, where $\delta = 2 \max_{x \in \mathcal{E}(\beta_w)} V_N(x)$. Further, $x_t \in \mathcal{X}$ for all $t \geq 0$. \square

PROOF

For any $x_t \in \mathbb{X}_{\text{rf}} \setminus \mathcal{E}(\beta_w)$ we have

$$\begin{aligned} V_N(x_t) &\geq V_N(\bar{x}_{t+1}) + \alpha \ell(x_t, v_N(x_t)) + \\ &\quad + \max_{w \in \mathcal{W}} V_N(\bar{x}_{t+1} + w_t) - \max_{w \in \mathcal{W}} V_N(\bar{x}_{t+1} + w_t) \\ &\geq \max_{w \in \mathcal{W}} V_N(\bar{x}_{t+1} + w_t) + \alpha \ell^*(x_t) - \beta_w \\ &\geq V_N(x_{t+1}) + \epsilon \ell^*(x_t) \geq V_N(x_{t+1}) + \frac{\epsilon \beta_w}{\alpha - \epsilon} \end{aligned}$$

where the first inequality comes from Theorem 1 since $x_t \in \mathbb{X}_{\text{rf}} \setminus \mathcal{E}(\beta_w) \subseteq \mathbb{X}_N$. The second inequality is by definition of ℓ^* and from (14) since by definition of \mathbb{X}_{rf} and of \ominus we have $\bar{x}_{t+1} \in \mathbb{X}_{\text{rf}} \ominus \mathcal{W} \subseteq \mathbb{X}_N \ominus \mathcal{W}$. The third and fourth inequalities are from (15) since $x_t \notin \mathcal{E}(\beta_w)$. By definition of δ we have $\mathcal{E}(\beta_w) \subseteq \Omega(\delta/2)$ which implies $\mathbb{X}_{\text{rf}} \setminus \Omega(\delta/2) \subseteq \mathbb{X}_{\text{rf}} \setminus \mathcal{E}(\beta_w)$. This implies that for any $x_t \in \mathbb{X}_{\text{rf}} \setminus \Omega(\delta/2)$ we have

$$V_N(x_t) \geq V_N(x_{t+1}) + \frac{\epsilon \beta_w}{\alpha - \epsilon}. \quad (16)$$

By definition of \mathbb{X}_{rf} we have $x_{t+1} \in \mathbb{X}_{\text{rf}}$ which implies that the preceding argument can be applied recursively. Thus, for any initial state $x_0 \in \mathbb{X}_{\text{rf}} \setminus \Omega(\delta/2)$ there is a finite time $t = t_0$ such that $x_{t_0} \in \Omega(\delta/2)$. Note that if $x_0 \in \Omega(\delta/2)$ we get $t_0 = 0$.

The system state can leave $\Omega(\delta/2)$ ones entered. However, the departure from this set is bounded. We have that

$$\frac{\delta}{2} = \max_{x \in \Omega(\delta/2)} V_N(x) \geq \max_{x \in \Omega(\delta/2)} \ell^*(x) \geq \max_{x \in \mathcal{E}(\beta_w)} \ell^*(x) = \frac{\beta_w}{\alpha - \epsilon}.$$

This gives that for every $x_t \in \Omega(\delta/2)$ we have

$$\begin{aligned} \max_{w \in \mathcal{W}} V_N(\bar{x}_{t+1} + w_t) &\leq V_N(x_t) - \alpha \ell(x_t, v_N(x_t)) + \beta_w \leq V_N(x_t) + \beta_w \\ &\leq \frac{\delta}{2} + \beta_w \leq \frac{\delta}{2} (1 + \alpha - \epsilon) \leq \delta. \end{aligned}$$

Thus, for $x_t \in \Omega(\delta/2)$ we have $x_{t+1} \in \Omega(\delta)$ for any $w \in \mathcal{W}$. Since by Assumption 2 we have $\Omega(\delta) \subset \mathbb{X}_{\text{rf}}$ get from (16) that system never leaves $\Omega(\delta)$.

To show that $x_t \in \mathcal{X}$ for all $t \geq 0$ we note due to the definition of \mathbb{X}_{rf} that $\bar{x}_{t+1} \in \mathbb{X}_{\text{rf}} \ominus \mathcal{W}$ for any $t \geq 1$. This implies that $x_{t+1} = \bar{x}_{t+1} + w_t \in (\mathbb{X}_{\text{rf}} \ominus \mathcal{W}) \oplus \mathcal{W} \subseteq \mathbb{X}_{\text{rf}} \subseteq \mathcal{X}$ for any $t \geq 1$.

This completes the proof. \square

In the following section we will see that the result presented in this section can be used to prove robust stability and robust feasibility for output feedback DMPC.

4. Output Feedback DMPC

We will use the result presented in the previous section to prove feasibility and stability properties in the output feedback setting. We start by designing the observer.

4.1 Observer Design

A crucial part for keeping the resulting output feedback controller simple is that the observer design can be performed in decentralized fashion. With the assumed structure on the dynamics, i.e., block-diagonal A -matrix, we can design local observers for each subsystem. In each subsystem the following observer is used

$$\hat{x}_{t+1}^i = A_{ii}\hat{x}_t^i + \sum_{j \in \mathcal{N}_i} \left(B_{ij}u_t^j \right) + K_i(y_t^i - C_i\hat{x}_t^i).$$

The information, besides the local information, needed to update the local estimates are the control action from neighboring nodes. This information is available in node i from the optimization algorithm communications. The local observers together form the following global observer

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t + K(y_t - C\hat{x}_t) \quad (17)$$

where $K = \text{blkdiag}(K_1, \dots, K_M)$. The error dynamics for the observer is purely local. We introduce the local error variables as $\tilde{x}^i = x^i - \hat{x}^i$ and get the following local error dynamics

$$\begin{aligned} \tilde{x}_{t+1}^i &= A_{ii}\tilde{x}_t^i + \sum_{j \in \mathcal{N}_i} \left(B_{ij}u_t^j \right) + w_t^i - A_{ii}\hat{x}_t^i - \sum_{j \in \mathcal{N}_i} \left(B_{ij}u_t^j \right) - K_i(y_t^i - C_i\hat{x}_t^i) \\ &= (A_{ii} - K_iC_i)\tilde{x}_t^i - K_i\xi_t^i + w_t^i. \end{aligned}$$

This shows that the poles of the observer dynamics can be placed arbitrarily using a block-diagonal observer gain K . For given K_i such that $\rho(A_{ii} - K_iC_i) < 1$ there exists a robust invariant set for the estimation error [Kolmanovsky and Gilbert, 1998]. In [Raković *et al.*, 2005] it was shown how an invariant outer approximation of the minimal robust invariant set can be computed. The minimal robust invariant set is (cf. [Raković *et al.*, 2005])

$$\mathcal{R}_i = \bigoplus_{j=0}^{\infty} \mathcal{F}_i^j$$

where $\mathcal{F}_i^j := (A_{ii} - K_iC_i)^j[-K_i\xi_i \oplus \mathcal{W}_i]$. In the approximation only a finite number of terms in the Minkowski sum is used and the resulting set sum is scaled to guarantee a certain accuracy of the approximation. The approximation is

$$\mathcal{R}_i^{c_e} = \frac{1}{1 - \kappa_i} \bigoplus_{j=0}^{s_i} \mathcal{F}_i^j$$

where s_i and κ_i can, for given accuracy ϵ_e , be computed without performing the Minkowski summation (cf. [Raković *et al.*, 2005]). The approximation is also robust invariant and satisfies (cf. [Raković *et al.*, 2005])

$$\mathcal{R}_i \subseteq \mathcal{R}_i^{\epsilon_e} \subseteq \mathcal{R}_i \oplus \mathcal{B}_\infty^{n_i}(\epsilon_e).$$

From the definition of a robust invariant set we get that if $\tilde{x}_0^i \in \mathcal{R}_i^{\epsilon_e}$ we have $\tilde{x}_t^i \in \mathcal{R}_i^{\epsilon_e}$ for all $t \geq 0$ and any disturbance sequences $\{\xi_t^i\}_{t=0}^\infty, \{w_t^i\}_{t=0}^\infty$. We define the global robust invariant set as $\mathcal{R} = \mathcal{R}_1 \times \dots \times \mathcal{R}_M$ and the approximation \mathcal{R}^{ϵ_e} accordingly. We get

$$\mathcal{R} \subseteq \mathcal{R}^{\epsilon_e} \subseteq \mathcal{R} \oplus \mathcal{B}_\infty^n(\epsilon_e)$$

since $\mathcal{B}_\infty^n(\epsilon_e) = \mathcal{B}_\infty^{n_1}(\epsilon_e) \times \dots \times \mathcal{B}_\infty^{n_M}(\epsilon_e)$.

4.2 One-step Robust Feasibility

The feedback in the output feedback case is based on the estimated current state \hat{x}_t . The objective of this section is to show how the original constraints need to be tightened to guarantee feasibility of the next state x_{t+1} and the estimated next state \hat{x}_{t+1} for any disturbances $w \in \mathcal{W}, \xi \in \mathcal{E}$. We rewrite the observer dynamics (17) as

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t + \hat{w}_t, \quad \hat{w}_t = K(C\tilde{x}_t + \xi_t) \quad (18)$$

and introduce the following set $\mathcal{W}_o = KC\mathcal{R}^{\epsilon_e} \oplus K\mathcal{E}$ and the corresponding local sets $\mathcal{W}_{o,i} = K_i C_i \mathcal{R}_i^{\epsilon_e} \oplus K_i \mathcal{E}_i$. We will see that the following optimization problem gives one-step robust feasibility when the initial condition is the estimated state:

$$\begin{aligned} V_N(\hat{x}) := \min_{\mathbf{z}, \mathbf{v}} \quad & J_N(\mathbf{z}, \mathbf{v}) & (19) \\ \text{s.t.} \quad & z_\tau \in \mathcal{X} \ominus \mathcal{W}_o \ominus \mathcal{R}^{\epsilon_e}, \quad \tau = 0, \dots, N-1, \\ & v_\tau \in \mathcal{U}, \quad \tau = 0, \dots, N-1, \\ & z_{\tau+1} = Az_\tau + Bv_\tau, \quad \tau = 0, \dots, N-2, \\ & z_0 = \hat{x}. \end{aligned}$$

REMARK 2

The tightened state constraint set $\mathcal{X} \ominus \mathcal{W}_o \ominus \mathcal{R}^{\epsilon_e}$ is the product of the corresponding tightened local constraint sets $\mathcal{X}_i \ominus \mathcal{W}_{o,i} \ominus \mathcal{R}_i^{\epsilon_e}$ which can be

computed efficiently by noting that

$$\begin{aligned}
 X_i^\epsilon \ominus \mathcal{W}_{o,i} \ominus \mathcal{R}_i^{\epsilon_e} &= X_i \ominus K_i C_i \mathcal{R}_i^{\epsilon_e} \ominus K_i \Xi_i \ominus \mathcal{R}_i^{\epsilon_e} \\
 &= X_i^\epsilon \left(\bigoplus_{j=0}^{s_i} \left(\frac{K_i C_i}{1 - \kappa_i} \mathcal{F}_i^j \right) \ominus K_i \Xi_i \bigoplus_{j=0}^{s_i} \left(\frac{1}{1 - \kappa_i} \mathcal{F}_i^j \right) \right) \\
 &= \left(X_i^\epsilon \ominus \frac{K_i C_i}{1 - \kappa_i} \mathcal{F}_i^0 \right) \left(\bigoplus_{j=1}^{s_i} \left(\frac{K_i C_i}{1 - \kappa_i} \mathcal{F}_i^j \right) \ominus K_i \Xi_i \ominus \mathcal{R}_i^{\epsilon_e} \right)
 \end{aligned}$$

where [Kolmanovsky and Gilbert, 1998, Theorem 2.1] is used in all steps. This implies that the local tightened constraint set can be computed by taking the Pontryagin difference \ominus recursively set by set. The number of inequalities that describes the final tightened constraint set is the same as in X_i due to Remark 1. This way, an explicit description of $\mathcal{R}_i^{\epsilon_e}$, which can be very expensive to compute, is avoided. \square

The new optimization problem with tightened constraints gives a new feedback control law v_N , infinite horizon steerable set \mathbb{X}_∞ , set \mathbb{X}_N and value function V_N . The notation is kept from previous sections, but the respective definitions refer in this section to optimization problem (19). Also, the definition of the recursively feasible set is different, we define

$$\mathbb{X}_{\text{rf}} = \{\hat{x} \in \mathbb{X}_N \mid \{A\hat{x} + Bv_N(\hat{x})\} \oplus \mathcal{R}^{\epsilon_e} \oplus \mathcal{W}_o \subseteq \mathbb{X}^{\text{rf}}\}.$$

We also define the one-step nominal prediction

$$\bar{x}_{t+1} := A\hat{x}_t + Bv_N(\hat{x}_t).$$

The following proposition shows that when using optimization problem (19) one-step robust feasibility in plant state x and estimated state \hat{x} is achieved regardless of disturbances $w \in \mathcal{W}$, $\xi \in \mathcal{E}$.

PROPOSITION 2

Suppose that $\tilde{x}_t \in \mathcal{R}^{\epsilon_e}$ and $\hat{x}_t \in \mathbb{X}_N$. Then $\hat{x}_{t+1} \in X \ominus \mathcal{R}^{\epsilon_e}$ and $x_{t+1} \in X$. \square

PROOF

From the problem formulation we have that $\bar{x}_{t+1} \in X \ominus \mathcal{R}^{\epsilon_e} \ominus \mathcal{W}_o$. Further $\hat{x}_{t+1} = \bar{x}_{t+1} + \hat{w}_t \in (X \ominus \mathcal{R}^{\epsilon_e} \ominus \mathcal{W}_o) \oplus \mathcal{W}_o \subseteq X \ominus \mathcal{R}^{\epsilon_e}$. Since $\tilde{x}_t \in \mathcal{R}^{\epsilon_e}$ we have $\tilde{x}_{t+1} \in \mathcal{R}^{\epsilon_e}$ and $x_{t+1} = \hat{x}_{t+1} + \tilde{x}_{t+1} \in X \ominus \mathcal{R}^{\epsilon_e} \oplus \mathcal{R}^{\epsilon_e} \subseteq X$. This concludes the proof. \square

4.3 Robust Stability

The estimation is affected by additive noise \hat{w}_t which satisfies $\hat{w}_t \in \mathcal{W}_o$ for all $t \geq 0$ if the estimation error $\tilde{x}_t \in \mathcal{R}^e$ for all $t \geq 0$. From the discussion in Section 3 we conclude that for every $x \in \mathbb{X}_N \ominus \mathcal{W}_o$ we have with finite $\beta_{w_o} \geq 0$ that

$$\max_{\hat{w} \in \mathcal{W}_o} V_N(x + \hat{w}) - V_N(x) \leq \beta_{w_o}.$$

In the following theorem we show that the estimated state \hat{x}_t and plant state x_t converges to robust invariant sets. Before we state the theorem, the following assumption is needed.

ASSUMPTION 3

We assume that the disturbance sets \mathcal{W} and \mathcal{E} are small enough to guarantee $\Omega(\delta) \subset \mathbb{X}_{\text{rf}}$ where $\delta = 2 \max_{x \in \mathcal{E}(\beta_{w_o})} V_N(x)$. \square

THEOREM 3

Suppose that Assumption 3 holds and that $\tilde{x}_0 = x_0 - \hat{x}_0 \in \mathcal{R}^e$. Then for any $\hat{x}_0 \in \mathbb{X}_{\text{rf}}$ the state estimation \hat{x}_t converges to $\Omega(\delta)$ where $\delta = 2 \max_{x \in \mathcal{E}(\beta_{w_o})} V_N(x)$ and the plant state x_t converges to $\Omega(\delta) \oplus \mathcal{R}^e$. Further $x_t \in \mathcal{X}$ for all $t \geq 1$. \square

PROOF

Since $\tilde{x}_0 \in \mathcal{R}^e$ we have $\tilde{x}_t \in \mathcal{R}^e$ for all $t \geq 0$. This implies that the disturbance to the estimated state (18) satisfies $\hat{w}_t \in \mathcal{W}_o$ for all $t \geq 0$. Convergence of the estimated state \hat{x}_t to $\Omega(\delta)$ is then given by Theorem 2 since the situation for \hat{x}_t is analogous to the situation for x_t in Theorem 2. Further, Theorem 2 also gives together with the definition of \mathbb{X}_{rf} that $\hat{x}_t \in \mathbb{X}_{\text{rf}} \ominus \mathcal{R}^e$ for all $t \geq 1$.

Convergence of the plant state $x_t = \hat{x}_t + \tilde{x}_t$ to $\Omega(\delta) \oplus \mathcal{R}^e$ follows directly from the estimated state \hat{x}_t convergence to $\Omega(\delta)$ and since $\tilde{x}_t \in \mathcal{R}^e$ for all $t \geq 0$. That $x_t \in \mathcal{X}$ for all $t \geq 1$ follows directly from $x_t = \hat{x}_t + \tilde{x}_t \in (\mathbb{X}_{\text{rf}} \ominus \mathcal{R}^e) \oplus \mathcal{R}^e \subseteq \mathbb{X}_{\text{rf}} \subseteq \mathcal{X}$.

This concludes the proof. \square

5. Numerical Example

We evaluate the efficiency of the proposed output feedback controller by applying it to a randomly generated system. The system is composed of six subsystems with five states, one control signal, and one output each. The measurement and system noise are bounded and within the following

sets

$$\mathcal{E}_i = \{\xi^i \in \mathbb{R} \mid |\xi^i| \leq 0.01\}, \quad \mathcal{W}_i = \{w^i \in \mathbb{R}^5 \mid \|w^i\|_\infty \leq 0.01\}$$

and the state and control constraint sets are

$$\begin{aligned} \mathcal{U}_i &= \{u^i \in \mathbb{R} \mid |u^i| \leq 0.1\}, \\ \mathcal{X}_i &= \{x^i \in \mathbb{R}^5 \mid -0.11 \leq [x^i]_j \leq 2, j = 1, \dots, 5\}. \end{aligned}$$

The observer gain is chosen as Kalman gain computed using unit noise variances. The tightened constraint set $\mathcal{X}_i^\epsilon \ominus K_i C_i \mathcal{R}_i^{\epsilon_e} \ominus K_i \mathcal{E}_i \ominus \mathcal{R}_i^{\epsilon_e}$ is computed using accuracy $\epsilon_e = 0.0001$ in $\mathcal{R}_i^{\epsilon_e}$. The resulting set has upper bounds on all state variables in the range $[1.895, 1.959]$ and lower bounds on the state variables in the range $[-0.069, -0.005]$. The nominal next state must satisfy these constraints to ensure that the estimated and true states satisfy the original constraints defined by \mathcal{X}_i . State and control costs are chosen, $Q = I$, $R = I$.

Numerical simulations suggest that for $\alpha = 0.5$ we get $N = 15$ in Theorem 1 and for $\alpha = 0.2$ we get $N = 6$. In Figure 1 the largest and smallest state values for each time step are plotted. The initial state vector comes from a uniform distribution and is scaled such that the largest element in the vector equals the original upper bound, i.e., 2 and the smallest element in the vector equals the original lower bound, i.e., -0.11.

We also analyze the size of the region of attraction and compare it to standard MPC where the terminal set is chosen as the maximal positive invariant set for the LQ-feedback computed using $Q = I$, $R = I$ (see [Gilbert and Tan, 1991]). The system is initialized with 40000 different initial conditions and each element in the initial state vector is chosen from a uniform distribution in the interval $[-0.11, 2]$, i.e., in the original constraint set. We have made two comparisons, the first is with $\alpha = 0.2$ which gives $N = 6$. Using $N = 6$ our controller managed to steer 98.9% of the initial conditions to the origin while respecting all constraints. The corresponding number in standard MPC with terminal constraint set and $N = 6$, was that 21.9% of the initial conditions were controlled to the origin. In the case for $\alpha = 0.5$ which gives $N = 15$ our controller managed to steer 98.9% of the initial conditions to the origin. For standard MPC with $N = 15$ the corresponding number was 52.8%. Note that the same set of initial conditions was used for all controllers. This shows that by not using a terminal constraint set, the region of attraction can be increased significantly while the computational burden is reduced.

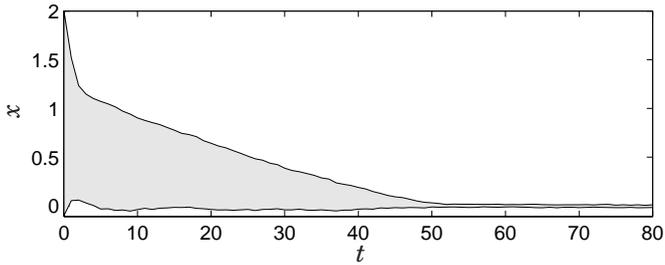


Figure 1. Region within which all state trajectories are confined for $N = 15$ in the output feedback case. Guaranteed upper and lower bounds for the all state variables are $-0.11 \leq x \leq 2$.

6. Conclusions

A robust distributed output feedback DMPC controller is proposed where the nominal behavior is optimized and the optimization problem has no terminal constraint set or terminal cost. Nominal stability for such DMPC formulations was proven in [Giselsson and Rantzer, 2012]. The results in [Giselsson and Rantzer, 2012] are in this paper extended to show robust stability in the state feedback case as well as the output feedback case. The provided numerical example also suggests that the lack of terminal constraint set can increase the region of attraction significantly.

References

- Almassalkhi, M. and I. Hiskens (2011): “Cascade mitigation in energy hub networks.” In *Proceedings of the 50th IEEE Control and Decision Conference*, pp. 2181–2188. Orlando, FL.
- Bemporad, A., M. Morari, V. Dua, and E. Pistikopoulos (2002): “The explicit linear quadratic regulator for constrained systems.” *Automatica*, **38:1**, pp. 3–20.
- Doan, M. D., T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter (2009): “A distributed version of Han’s method for DMPC using local communications only.” *Control Engineering and Applied Informatics*, **11:3**, pp. 6–15.
- Dunbar, W. (2007): “Distributed receding horizon control of dynamically coupled nonlinear systems.” *IEEE Transactions on Automatic Control*, **52**, pp. 1249–1263.
- Dunbar, W. B. and S. Desa (2005): “Distributed model predictive control for dynamic supply chain management.” In *Proceedings of the International Workshop on Assessment and Future Directions of NMPC*. Freudenstadt-Lauterbad, Germany.
- Gilbert, E. and K. Tan (1991): “Linear systems with state and control constraints: the theory and application of maximal output admissible sets.” *IEEE Transactions on Automatic Control*, **36:9**, pp. 1008–1020.
- Giselsson, P., M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer (2012): “Accelerated gradient methods and dual decomposition in distributed model predictive control.” *Automatica*. To appear. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Giselsson, P. and A. Rantzer (2012): “On feasibility, stability and performance in distributed model predictive control.” Submitted to *IEEE Transactions on Automatic Control*. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Grimm, G., M. J. Messina, S. E. Tuna, and A. R. Teel (2004): “Examples when nonlinear model predictive control is nonrobust.” *Automatica*, **40:10**, pp. 1729–1738.
- Kolmanovsky, I. and E. G. Gilbert (1998): “Theory and computation of disturbance invariant sets for discrete-time linear systems.” *Mathematical Problems in Engineering*, **4:4**, pp. 317–367.
- Mayne, D., S. Raković, R. Findeisen, and F. Allgöwer (2006): “Robust output feedback model predictive control of constrained linear systems.” *Automatica*, **42:7**, pp. 1217–1222.

- Mayne, D., M. Seron, and S. Raković (2005): “Robust model predictive control of constrained linear systems with bounded disturbances.” *Automatica*, **41:2**, pp. 219–224.
- Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert (2000): “Constrained model predictive control: Stability and optimality.” *Automatica*, **36:6**, pp. 789–814.
- Negenborn, R., B. De Schutter, and J. Hellendoorn (2008): “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes.” *Engineering Applications of Artificial Intelligence*, **21:3**, pp. 353–366.
- Petrone, F. (2010): “Model predictive control of a hydro power valley.”. Master’s thesis, Politecnico di Milano, Italy.
- Raković, S. V., E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne (2005): “Invariant approximations of the minimal robust positively invariant set.” *Automatic Control, IEEE Transactions on*, **50:3**, pp. 406–410.
- Richards, A. and J. How (2007): “Robust distributed model predictive control.” *International Journal of Control*, **80:9**, pp. 1517–1531.
- R.M. Hermans, M. Lazar, A. J. (2010): “Almost decentralized Lyapunov-based nonlinear model predictive control.” In *Proceedings of the 29th American Control Conference*. Baltimore.
- Venkat, A., I. Hiskens, J. Rawlings, and S. Wright (2006): “Distributed output feedback MPC for power system control.” In *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 4038–4045.

Paper VII

A Distributed Accelerated Gradient Algorithm for DMPC of a Hydro Power Valley

**Minh Dang Doan, Pontus Giselsson, Tamás Keviczky,
Bart De Schutter, and Anders Rantzer**

Abstract

A distributed model predictive control (DMPC) approach based on distributed optimization is applied to the power reference tracking problem of a hydro power valley (HPV) system. The applied optimization algorithm is based on accelerated gradient methods and achieves a convergence rate of $O\left(\frac{1}{k^2}\right)$, where k is the iteration number. Major challenges in the control of the HPV include a nonlinear and large-scale model, nonsmoothness in the power-production functions, and a globally coupled cost function that prevents distributed schemes to be applied directly. We propose a linearization and approximation approach that accommodates the proposed DMPC framework and provides very similar performance compared to a centralized solution in simulations. The provided numerical studies also suggest that for the sparsely interconnected system at hand, the distributed algorithm we propose is faster than a centralized state-of-the-art solver, namely CPLEX.

Submitted to *Control Engineering Practice*.

1. Introduction

Hydro power plants generate electricity from potential energy and kinetic energy of natural water, and often a number of power plants are placed along a long river or a water body system to generate the power at different stages. Currently, hydro power is one of the most important means of renewable power generation in the world. In order to meet the world's electricity demand, hydro power production should continue to grow due to the increasing cost of fossil fuels. However, hydro electricity, like any renewable energy, depends on the availability of a primary resource, in this case: water. Most natural locations where power-generating infrastructure can be built economically have already been utilized [PEW Center on Global Climate Change, 2011]. The expected trend for future use of hydro power is to build small-scale plants that can generate electricity for a single community. Thus, an increasingly important objective of hydro power plants is to manage the available water resources efficiently, while following an optimal production profile with respect to changes in the electricity market, to maximize the long-term benefit of the plant. This water resource management must be compatible with ship navigation and irrigation, and it must respect environmental and safety constraints on levels and flow rates in the lakes and the rivers. This is why real-time control of the water flows in a hydro power valley (HPV) becomes important and can increase significantly the power efficiency of these systems.

An HPV may contain several rivers and lakes, spanning a wide geographical area and exhibiting complex dynamics. In order to tackle the plant-wide control of such a complex system, an HPV is often treated as a large-scale system consisting of interacting subsystems. Large-scale system control has been an active research area that has resulted in a variety of control techniques, which can be classified in three main categories: decentralized control, distributed control, and centralized control. Application of these approaches can be found in a rich literature on control of water canals for irrigation and hydro systems [Mareels *et al.*, 2005, Litrico and Fromion, 2009]. We are interested in applying model predictive control (MPC), a control method that has been successfully used in industry [Qin and Badgwell, 2003], thanks to its capability of handling hard constraints and the simple way of incorporating an economical objective by means of an optimization problem. For the control problem of open water systems, centralized MPC has been studied in numerical examples using nonlinear MPC approaches in combination with model smoothing and/or model reduction techniques [Igreja and Lemos, 2009, Nederkoorn *et al.*, 2011], and in real implementations with linear MPC of low-dimensional systems [van Overloop, 2006, van Overloop *et al.*, 2010]. However, centralized MPC has a drawback when controlling large-scale systems due to limitations

in communications and the computational burden. These issues fostered the studies of decentralized MPC and distributed MPC for large-scale water systems. Early decentralized MPC methods for irrigation canals used the decomposition-coordination approach to obtain decentralized versions of LQ control [Fawal *et al.*, 1998]. Several decentralized MPC simulations applied to irrigation canals and rivers were presented in [Georges, 1994, Sawadogo *et al.*, 1998, Gomez *et al.*, 2002, Sahin and Morari, 2010]. Distributed MPC approaches based on coordination and cooperation for water delivery canals were presented in [Georges, 1994, Negenborn *et al.*, 2009, Igreja *et al.*, 2011, Anand *et al.*, 2011]. The typical control objective in these studies is to regulate water levels and to deliver the required amount of water to the right place at some time in the future, i.e., the cost function does not have any special term except the quadratic penalties on the states and the inputs. On the other hand, in hydro power control, there are output penalty terms in the cost function that represent the objective of manipulating power production. Recent literature taking into account this cost function includes centralized nonlinear MPC with a parallel version of the multiple-shooting method for the optimal control problem using continuous nonlinear dynamics [Savorgnan *et al.*, 2011], and a software framework that formulates a discrete-time linear MPC controller with the possibility to integrate a nonlinear prediction model and to use commercial solvers to solve the optimization problem [Petroni, 2010]. The hydro power control problem considered in the current paper is similar to the setup in [Savorgnan *et al.*, 2011, Petroni, 2010]. However, it distinguishes itself by using a distributed control structure that aims to avoid global communications and that divides the computational tasks into local sub-tasks that are handled by subsystems, making the approach more suitable for scaling up to even more complicated hydro power plants.

The distributed MPC design approach proposed in this paper is enabled by a distributed optimization algorithm that has recently been developed by the authors in [Giselsson *et al.*, 2012]. This optimization algorithm is designed for a class of strongly convex problems with mixed 1-norm and 2-norm terms in the cost function, which perfectly suits the power reference tracking objective in the HPV control benchmark. The underlying optimization algorithm in [Giselsson *et al.*, 2012], although being implemented in a distributed way, is proved to achieve the global optimum with an $O(\frac{1}{k^2})$ convergence rate, where k is the iteration number. This is a significant improvement compared to the distributed MPC methods presented in [Doan *et al.*, 2011, Doan *et al.*, 2009, Giselsson and Rantzer, 2010, Negenborn *et al.*, 2008], which achieve an $O(\frac{1}{k})$ convergence rate. There are three main challenges in applying distributed MPC using the algorithm from [Giselsson *et al.*, 2012] to the HPV benchmark problem. The first one is that the nonlinear continuous-time model yields a

relatively large linear model after spatial and temporal discretizations. We present a decentralized model order reduction method that significantly reduces the model complexity while maintaining prominent dynamics. The second challenge is that the power production functions are nonsmooth, which prevents gradient-based methods to be applied directly. A method to overcome this difficulty and to enable optimal control using the algorithm from [Giselsson *et al.*, 2012] is also presented. The third challenge is that the whole system should follow a centralized power reference which, if the algorithm from [Giselsson *et al.*, 2012] is applied directly, requires centralized communication. We propose a dynamic power division approach that allows to track this centralized power reference with only distributed communications. By means of numerical examples, we will demonstrate the fast convergence property of the distributed algorithm which, when implemented on a single core, can outperform a state-of-the-art centralized solver (CPLEX) when solving the same optimization problem.

The remaining parts of the paper are organized as follows. In Section 2, we describe the HPV system and the power reference tracking problem that were formulated in the HPV benchmark problem [Savorgnan and Diehl, 2011]. Section 3 provides a summary of the distributed optimization framework that the authors have developed in [Giselsson *et al.*, 2012]. In Section 4, we present our approach for modeling and model reduction of the HPV system, followed by a reformulation of the MPC optimization problem, and developing a distributed estimator so that the closed loop distributed MPC scheme can be implemented using neighbor-to-neighbor communications only. The simulation results are presented in Section 5, which also features a comparison with centralized MPC and decentralized MPC. Through the various aspects of the comparison including performance, computational efficiency, and communication requirements, the advantages of the distributed MPC algorithm will be highlighted. Section 6 concludes the paper and outlines future work.

2. Problem Description

In this section, we provide a summary of the hydro power valley benchmark [Savorgnan and Diehl, 2011] and we present the linearized model that serves as the starting point of our controller design.

2.1 Hydro Power Valley System

We consider a hydro power plant composed of several interconnected subsystems, as illustrated in Figure 1. The plant can be divided into 8 subsystems, of which subsystem S_1 is composed of the lakes L_1, L_2 , the duct U_1

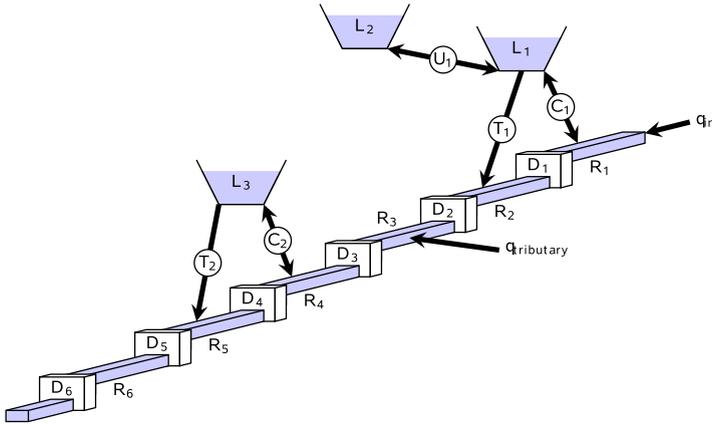


Figure 1. Overview of the HD-MPC hydro power valley system [Savorgnan and Diehl, 2011]

connecting them, and the ducts C_1, T_1 that connect L_1 with the reaches¹ R_1, R_2 , respectively. Subsystem S_2 is composed of the lake L_3 and the ducts C_2, T_2 that connect L_3 to the reaches R_4, R_5 , respectively. There are 6 other subsystems, each of which consists of a reach and a dam at the end of the reach. These six reaches R_1, R_2, R_3, R_4, R_5 , and R_6 are connected in series, separated by the dams D_1, D_2, D_3, D_4 , and D_5 . The large lake that follows the dam D_6 is assumed to have a fixed water level, which will absorb all the discharge. The outside water flows enter the system at the upstream end of reach R_1 and at the middle of reach R_3 .

There are structures placed in the ducts and at the dams to control the flows. These are the turbines placed in the ducts T_1, T_2 and at each dam for power production. In the ducts C_1, C_2 there are composite structures that can either function as pumps (for transporting water to the lakes) or as turbines (when water is drained from the lakes).

The whole system has 10 manipulated variables, which are composed of six dam flows ($q_{D1}, q_{D2}, q_{D3}, q_{D4}, q_{D5}, q_{D6}$), two turbine flows (q_{T1}, q_{T2}) and two pump/turbine flows (q_{C1}, q_{C2}). Further, the system has 9 measured variables, the water levels in the three lakes (h_{L1}, h_{L2}, h_{L3}) and the water levels at the end of each reach ($h_{R1}, h_{R2}, h_{R3}, h_{R4}, h_{R5}, h_{R6}$).

2.2 Power Reference Tracking Problem

One of the control problems specified in [Savorgnan and Diehl, 2011] is the

¹A reach is a river segment between two dams.

power reference tracking problem. We introduce state variables x , which consist of water levels in the lakes and reaches and water flows within the reaches, and control variables q , which are the manipulated water flows. The problem is to track a power production profile, $p^{\text{ref}}(t)$, on a daily basis using the following cost function:

$$\begin{aligned} J \triangleq & \int_0^T \gamma \left| p^{\text{ref}}(t) - \sum_{i=1}^8 p_i(x(t), q(t)) \right| dt + \\ & + \sum_{i=1}^8 \int_0^T (x_i(t) - x_i^{\text{ss}})^T Q_i (x_i(t) - x_i^{\text{ss}}) dt + \\ & + \sum_{i=1}^8 \int_0^T (q_i(t) - q_i^{\text{ss}})^T R_i (q_i(t) - q_i^{\text{ss}}) dt \end{aligned} \quad (1)$$

subject to the nonlinear dynamics and linear constraints on outputs and inputs as specified in [Savorgnan and Diehl, 2011]. The weights $Q_i, R_i, i = 1, \dots, 8, \gamma$, and the testing period T are parameters of the benchmark.

The quadratic term in the cost function represents the penalties on the state deviation from the steady state x^{ss} and the energy used for manipulating the inputs away from the steady state flows q^{ss} . The 1-norm term represents the power reference tracking mismatch, in which the function p^{ref} is the power reference and the function p_i represents the locally produced/consumed power by a subsystem $i \in \{1, \dots, 8\}$. For $i = 1, 2$ the produced/consumed power is (cf. [Savorgnan and Diehl, 2011])

$$p_i(x(t), q(t)) = k_{C_i}(q_{C_i}(t))q_{C_i}(t)\Delta x_{C_i}(t) + k_{T_i}q_{T_i}(t)\Delta x_{T_i}(t) \quad (2)$$

where q_{C_i} and q_{T_i} are the flows through ducts C_i and T_i , Δx_{C_i} and Δx_{T_i} are the relative differences in water levels before and after ducts C_i and T_i respectively, k_{T_i} is the power coefficient of the turbine T_i , and

$$k_{C_i}(q_{C_i}(t)) = \begin{cases} k_{T_{C_i}}, & q_{C_i}(t) \geq 0 \\ k_{P_{C_i}}, & q_{C_i}(t) < 0 \end{cases} \quad (3)$$

is a discontinuous power coefficient that depends on whether the duct C_i acts as a turbine ($q_{C_i}(t) > 0$) or as a pump ($q_{C_i}(t) < 0$). For $i = 3, \dots, 8$ we have

$$p_i(x(t), q(t)) = k_{D_{i-2}}q_{D_{i-2}}(t)\Delta x_{D_{i-2}}(t) \quad (4)$$

which is the power produced by the turbine located at dam D_{i-2} . The produced/consumed power functions given in (2) and (4) are nonlinear

and even nonsmooth for subsystems 1 and 2 due to (3), thus complicating a direct application of a standard MPC scheme.

Still, the complexity of the system and control objective suggests an optimization based control strategy, such as MPC. Further, the distributed nature of the system makes it possible to consider distributed MPC techniques. However, the stated optimization problem (1) is a nonlinear continuous time dynamic optimization problem, which in general is very hard to solve. In the next sections we will discuss the modeling of the hydro power valley that leads to a linearized model.

2.3 Nonlinear Hydro Power Valley Model

The model of the reaches is based on the one-dimensional Saint Venant partial differential equation, representing the mass and momentum balance (see [Savorgnan and Diehl, 2011] for details):

$$\begin{cases} \frac{\partial q(t, z)}{\partial z} + \frac{\partial s(t, z)}{\partial t} = 0 \\ \frac{1}{g} \frac{\partial}{\partial t} \left(\frac{q(t, z)}{s(t, z)} \right) + \frac{1}{2g} \frac{\partial}{\partial z} \left(\frac{q^2(t, z)}{s^2(t, z)} \right) + \frac{\partial h(t, z)}{\partial z} + I_f(t, z) - I_0(z) = 0 \end{cases} \quad (5)$$

with z the spatial variable, t the time variable, q the river flow (or discharge), s the cross-section surface of the river, h the water level w.r.t. the river bed, I_f the friction slope, $I_0(z)$ the river bed slope, and g the gravitational acceleration constant.

The partial differential equation (5) is converted into a system of ordinary differential equations by using spatial discretization. To achieve this, each reach is divided into 20 cells, yielding 20 additional states, which are the water levels at the beginning of the cells. For details of the spatial discretization and the equations for the resulting nonlinear dynamical system the reader is referred to [Savorgnan and Diehl, 2011, Section 2.1.1]. The resulting nonlinear dynamical system has in total 249 states, 10 inputs, and 9 outputs.

2.4 Model Linearization and Discretization

As mentioned in Section 2.3 a set of nonlinear ordinary differential equations that describe the hydro power valley dynamics is presented in [Savorgnan and Diehl, 2011, Section 2.1.1]. A linear continuous-time model which is linearized around the steady state operating point (x^{ss}, q^{ss}) is also provided in the HPV benchmark package [Savorgnan and Diehl, 2011]. Discretizing this model using zero-order-hold gives a discrete-time linear system with 249 states and 10 inputs. The coupling of the subsystems is through the inputs only. This implies that discretization using zero-order-hold of the continuous-time system keeps the structure of the original

3. Distributed Optimization Framework for MPC

system description. Thus, the resulting discrete time system has a block-diagonal dynamics matrix, a block-diagonal output matrix, and a sparse input matrix, and each subsystem $i = 1, \dots, 8$ can be expressed in the following form:

$$\begin{aligned}x_i^d(k+1) &= A_{ii}x_i^d(k) + \sum_{j=1}^8 B_{ij}q_j^d(k) \\y_i^d(k) &= C_i x_i^d(k)\end{aligned}\tag{6}$$

in which the variables x^d, q^d , and y^d stand for the deviation from the steady-state values, and the subscripts i, j stand for the subsystem indices. As mentioned the subsystems are coupled through the inputs only and at least for some $j \in \{1, \dots, 8\}$ we have $B_{ij} = 0$ for every $i = 1, \dots, 8$.

The use of a discrete-time linearized model enables controller design with some specific approaches, which include our proposed distributed optimization technique presented in [Giselsson *et al.*, 2012]. Before describing our main contributions, we now provide a summary of this distributed optimization framework in the next section.

3. Distributed Optimization Framework for MPC

In this section, we describe the distributed optimization algorithm developed in [Giselsson *et al.*, 2012] which is based on an accelerated gradient method. The first accelerated gradient method was developed in [Nesterov, 1983] and further elaborated and extended in [Beck and Teboulle, 2009, Nesterov, 1988, Nesterov, 2005, Toh and Yun, 2010, Tseng, 2008]. The main idea of the algorithm presented in [Giselsson *et al.*, 2012] is to exploit the problem structure of the dual problem such that accelerated gradient computations can be distributed to subsystems. Hence, the distributed algorithm effectively solves the centralized optimization problem. Dual decomposition has been used in the past to tackle the complexity of large-scale optimization problems arising in water supply networks [Carpentier and Cohen, 1993]. In our work however, in addition to simplifying the local computations, we apply this decomposition philosophy in order to distribute the decision-making process.

The algorithm in [Giselsson *et al.*, 2012] is developed to handle opti-

mization problems of the form

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{x}_a} \quad & \frac{1}{2} \mathbf{x}^T H \mathbf{x} + g^T \mathbf{x} + \gamma \|\mathbf{x}_a\|_1 & (7) \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} = \mathbf{b} \\ & \mathbf{C} \mathbf{x} \leq \mathbf{d} \\ & \mathbf{x}_a = P \mathbf{x} - p \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{x}_a \in \mathbb{R}^m$ are vectors of decision variables, and \mathbf{x} is partitioned according to:

$$\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_M^T]^T, \quad (8)$$

and $\mathbf{x}_i \in \mathbb{R}^{n_i}$. Further, the matrix $H \in \mathbb{R}^{n \times n}$ is positive definite and block-diagonal, the matrices $\mathbf{A} \in \mathbb{R}^{q \times n}$, $\mathbf{C} \in \mathbb{R}^{r \times n}$, and $P \in \mathbb{R}^{m \times n}$ have sparse structures, and $g \in \mathbb{R}^n$, $p \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^q$, $\mathbf{d} \in \mathbb{R}^r$. We introduce the partitions $g = [g_1^T, \dots, g_M^T]^T$, $p = [p_1^T, \dots, p_M^T]^T$, $\mathbf{b} = [\mathbf{b}_1^T, \dots, \mathbf{b}_M^T]^T$, $\mathbf{d} = [\mathbf{d}_1^T, \dots, \mathbf{d}_M^T]^T$,

$$\begin{aligned} H &= \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_M \end{bmatrix}, & \mathbf{A} &= \begin{bmatrix} \mathbf{A}_{11} & \dots & \mathbf{A}_{1M} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{M1} & \dots & \mathbf{A}_{MM} \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} \mathbf{C}_{11} & \dots & \mathbf{C}_{1M} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{M1} & \dots & \mathbf{C}_{MM} \end{bmatrix}, & P &= \begin{bmatrix} P_{11} & \dots & P_{1M} \\ \vdots & \ddots & \vdots \\ P_{M1} & \dots & P_{MM} \end{bmatrix} \end{aligned}$$

where the partitions are introduced in accordance with (8) and $g_i \in \mathbb{R}^{n_i}$, $p_i \in \mathbb{R}^{m_i}$, $\mathbf{b}_i \in \mathbb{R}^{q_i}$, $\mathbf{d}_i \in \mathbb{R}^{r_i}$, $H_i \in \mathbb{R}^{n_i \times n_i}$, $\mathbf{A}_{ij} \in \mathbb{R}^{q_i \times n_j}$, $\mathbf{C}_{ij} \in \mathbb{R}^{r_i \times n_j}$ and $P_{ij} \in \mathbb{R}^{m_i \times n_j}$. The assumption on sparsity of \mathbf{A} , \mathbf{C} and P is that $\mathbf{A}_{ij} = 0$, $\mathbf{C}_{ij} = 0$, and $P_{ij} = 0$ for some i, j and we assume that the constraint matrices are built such that $\mathbf{A}_{ii} \neq 0$, $\mathbf{C}_{ii} \neq 0$, and $P_{ii} \neq 0$ for all $i = 1, \dots, M$. Based on the coupling, we define for each subsystem a neighborhood set, denoted by \mathcal{N}_i , as follows:

$$\begin{aligned} \mathcal{N}_i &= \{j \in \{1, \dots, M\} \mid \mathbf{A}_{ij} \neq 0 \text{ or } \mathbf{A}_{ji} \neq 0 \text{ or } \mathbf{C}_{ij} \neq 0 \text{ or } \mathbf{C}_{ji} \neq 0 \text{ or} \\ &\quad \text{or } P_{ij} \neq 0 \text{ or } P_{ji} \neq 0\}. \end{aligned} \quad (9)$$

We introduce dual variables $\lambda \in \mathbb{R}^q$, $\mu \in \mathbb{R}^r$, $\nu \in \mathbb{R}^m$ for the equality constraints, inequality constraints, and equality constraints originating from the 1-norm cost in (7) respectively. We also introduce the dual variable

3. Distributed Optimization Framework for MPC

partitions $\lambda = [\lambda_1^T, \dots, \lambda_M^T]^T$, $\mu = [\mu_1^T, \dots, \mu_M^T]^T$, and $\nu = [\nu_1^T, \dots, \nu_M^T]^T$ where $\lambda_i \in \mathbb{R}^{q_i}$, $\mu_i \in \mathbb{R}^{r_i}$, and $\nu_i \in \mathbb{R}^{m_i}$. Based on [Giselsson *et al.*, 2012], the dual problem of (7) can be cast as minimization of the negative dual function

$$f(\lambda, \mu, \nu) = \frac{1}{2}(\mathbf{A}_1^T \lambda + \mathbf{A}_2^T \mu + \mathbf{P}^T \nu)^T \mathbf{H}^{-1}(\mathbf{A}_1^T \lambda + \mathbf{A}_2^T \mu + \mathbf{P}^T \nu) + B_1^T \lambda + B_2^T \mu + p^T \nu \quad (10)$$

where the dual variables are constrained to satisfy

$$\lambda \in \mathbb{R}^q, \quad \mu \in \mathbb{R}_+^r, \quad \nu \in [-\gamma, \gamma]^m \quad (11)$$

and \mathbb{R}_+ denotes the non-negative real orthant. The negative dual function (10) has a Lipschitz continuous gradient with constant (cf. [Giselsson *et al.*, 2012])

$$L = \|[\mathbf{A}^T \ \mathbf{C}^T \ \mathbf{P}^T]^T \mathbf{H}^{-1} [\mathbf{A}^T \ \mathbf{C}^T \ \mathbf{P}^T]\|_2 \quad (12)$$

and can hence be minimized using accelerated gradient methods. The distributed accelerated gradient method as presented in [Giselsson *et al.*, 2012] is summarized below in a slightly different form.

ALGORITHM 1—DISTRIBUTED ACCELERATED GRADIENT ALGORITHM

Initialize $\lambda^0 = \lambda^{-1}$, $\mu^0 = \mu^{-1}$, $\nu^0 = \nu^{-1}$ and \mathbf{x}^{-1} with the last values from previous sampling time. For the first sampling time, these variables are initialized by zeros.

In every node, i , the following computations are performed:

For $k = 0, 1, 2, \dots$

1. Compute

$$\begin{aligned} \mathbf{x}_i^k &= -\mathbf{H}_i^{-1} \left(\sum_{j \in \mathcal{N}_i} (\mathbf{A}_{ji}^T \lambda_j^k + \mathbf{C}_{ji}^T \mu_j^k + \mathbf{P}_{ji}^T \nu_j^k) \right) \\ \bar{\mathbf{x}}_i^k &= \mathbf{x}_i^k + \frac{k-1}{k+2} (\mathbf{x}_i^k - \mathbf{x}_i^{k-1}) \end{aligned}$$

2. Send $\bar{\mathbf{x}}_i^k$ to each $j \in \mathcal{N}_i$, receive $\bar{\mathbf{x}}_j^k$ from each $j \in \mathcal{N}_i$

3. Compute

$$\begin{aligned}\lambda_i^{k+1} &= \lambda_i^k + \frac{k-1}{k+2}(\lambda_i^k - \lambda_i^{k-1}) + \frac{1}{L} \left(\sum_{j \in \mathcal{N}_i} \mathbf{A}_{ij} \bar{\mathbf{x}}_j^k - \mathbf{b}_i \right) \\ \mu_i^{k+1} &= \max \left\{ 0, \mu_i^k + \frac{k-1}{k+2}(\mu_i^k - \mu_i^{k-1}) + \frac{1}{L} \left(\sum_{j \in \mathcal{N}_i} \mathbf{C}_{ij} \bar{\mathbf{x}}_j^k - \mathbf{d}_i \right) \right\} \\ v_i^{k+1} &= \min \left\{ \gamma, \max \left[-\gamma, v_i^k + \frac{k-1}{k+2}(v_i^k - v_i^{k-1}) + \right. \right. \\ &\quad \left. \left. + \frac{1}{L} \left(\sum_{j \in \mathcal{N}_i} P_{ij} \bar{\mathbf{x}}_j^k - p_i \right) \right] \right\}\end{aligned}$$

4. Send λ_i^{k+1} , μ_i^{k+1} , v_i^{k+1} to each $j \in \mathcal{N}_i$, receive λ_j^{k+1} , μ_j^{k+1} , v_j^{k+1} from each $j \in \mathcal{N}_i$.

□

The Lipschitz constant L of ∇f is used in the algorithm. For MPC purposes, we only need to compute L once in a centralized way and use it through all MPC problem instances.

Besides the suitability for distributed implementation, another merit of Algorithm 1 is its fast convergence rate. The main convergence results of Algorithm 1 are given in [Giselsson *et al.*, 2012], stating that both the dual function value and the primal variables converge towards their respective optima with the rate of $O\left(\frac{1}{k^2}\right)$ where k is the iteration index. This convergence rate is much better than the convergence rate of classical gradient-based optimization algorithms, which is $O\left(\frac{1}{k}\right)$.

4. Control of HPV using Distributed MPC

We have so far described the linear discrete-time model of the HPV in Section 2 and the fast distributed optimization method, Algorithm 1, that serves as a basis for designing a distributed model predictive controller to be applied to the HPV. However, there are three major challenges for this application. First, the linear discrete-time model cannot be directly used in an MPC context due to the existence of a number of unobservable and uncontrollable modes. These unobservable/uncontrollable modes are a result of the discretization in space and time. Second, the power functions associated with the ducts C_1 and C_2 are nonsmooth (cf. (2) and (3)). The

nonsmoothness is caused by the fact that the flow through C_1 and C_2 is bidirectional and the powers consumed/produced do not have equivalent coefficients. The third major challenge is the global coupling in the cost function due to the fact that we have to track a central power reference function that specifies the desired sum of locally generated power outputs. This global coupling prevents a distributed implementation of Algorithm 1 since the sparsity in the constraints is lost. These issues are addressed in the following sections.

4.1 Modification of the Linear Model

In this section we show how to create a model of the HPV that is suitable for the DMPC framework presented in [Giselsson *et al.*, 2012]. First we present a model reduction technique that keeps the system structure, then the nonsmooth power function is treated.

Decentralized Model Order Reduction

The block-diagonal structure of the discrete-time dynamical system (6) makes it possible to perform model reduction on each subsystem individually. We use balanced truncation [Gugercin and Antoulas, 2004] to reduce the order of each local model (6).

Let us introduce $B_i = [B_{i1} \dots B_{i8}]$ and $q = [(q_1^d)^T \dots (q_8^d)^T]^T$ to get the following discrete-time linear model of each subsystem:

$$\begin{aligned} x_i^d(k+1) &= A_{ii}x_i^d(k) + B_iq^d(k) \\ y_i^d(k) &= C_ix_i^d(k). \end{aligned} \quad (13)$$

Applying the balanced truncation technique yields transformation matrices denoted by T_i^r and $T_i^{r,inv}$ for each subsystem, where $T_i^r T_i^{r,inv} = I$. By denoting the new state variables, $x_i^r = T_i^r x_i^d$, and the control variable $q^r = q$, we represent the reduced order model as:

$$x_i^r(k+1) = A_{ii}^r x_i^r(k) + B_i^r q^r(k) \quad (14)$$

$$y_i^r(k) = C_i^r x_i^r(k) \quad (15)$$

where $A_{ii}^r = T_i^r A_{ii} T_i^{r,inv}$, $B_i^r = T_i^r B_i$ and $C_i^r = C_i T_i^{r,inv}$. It should be noted that the block-sparsity structure of B_i^r is the same as in the non-reduced input matrix B_i , since the model reduction is performed for each local model separately.

The model reduction gives a 32-state reduced model that approximately represents the dynamics of the full linear model with 249 states.

Treatment of Nonlinear and Nonsmooth Power Function

One of the difficulties in applying a linear MPC approach to the hydro power valley is the nonsmoothness of the power functions associated with the ducts C_1 and C_2 . The nonsmoothness is caused by the fact that the flow through C_1 and C_2 is bidirectional and the power generated or consumed have different coefficients. The consumed/produced power in ducts C_1 and C_2 is included in the expression for power generation (2) in subsystem 1 and subsystem 2, respectively. In order to handle this nonsmoothness, we use a double-flow technique, which means introducing two nonnegative variables to express the flow in C_i , $i = 1, 2$ at a sampling step k :

- $q_{C_{iP}}(k)$: virtual flow such that C_i functions as a pump
- $q_{C_{iT}}(k)$: virtual flow such that C_i functions as a turbine

The introduction of virtual flows requires the input-matrices B_i^r to be augmented with two extra columns identical to the ones multiplying q_{C_i} , $i = 1, 2$ with the opposite sign to capture that pump action is also introduced with a positive flow. The resulting reduced order model has 12 inputs instead of the original 10. Using the introduced flows $q_{C_{iP}}$ and $q_{C_{iT}}$, the power function (2) for subsystems 1 and 2 can be rewritten as

$$p_i(x(k), q(k)) = \left(k_{T_{C_i}} q_{C_{iT}}(k) - k_{P_{C_i}} q_{C_{iP}}(k) \right) \Delta x_{C_i}(k) + k_{T_i} q_{T_i}(k) \Delta x_{T_i}(t) \quad (16)$$

with the additional constraints that

$$q_{C_{iT}}(k) \geq 0, \quad q_{C_{iP}}(k) \geq 0, \quad q_{C_{iT}}(k) q_{C_{iP}}(k) = 0.$$

The last constraint expresses the fact that water flows in only one direction at a time, i.e., that either the pump or the turbine is active. The resulting nonlinear expression (16) can in turn be linearized around the steady-state solution (x^{ss}, q^{ss}) . Since $q_{C_i}^{ss} = 0$ for $i = 1, 2$ we get the following linear local power production/consumption approximation for subsystems $i = 1, 2$:

$$\begin{aligned} \hat{p}_i(x(k), q(k)) = & \Delta x_{C_i}^{ss} \begin{bmatrix} k_{T_{C_i}} & -k_{P_{C_i}} \end{bmatrix} \begin{bmatrix} q_{C_{iT}}(k) \\ q_{C_{iP}}(k) \end{bmatrix} + \\ & + k_{T_i} q_{T_i}^{ss} (\Delta x_{T_i}(k) - \Delta x_{T_i}^{ss}) + k_{T_i} \Delta x_{T_i}^{ss} (q_{T_i}(k) - q_{T_i}^{ss}) + \\ & + k_{T_i} q_{T_i}^{ss} \Delta h_{T_i}^{ss}. \end{aligned}$$

This reformulation results in a linear expression with a nonlinear constraint, that is $q_{C_{iT}}(k) q_{C_{iP}}(k) = 0$, that approximates the original nonsmooth nonlinear power production/consumption expression (2). We propose a method to handle the nonlinear constraint in Section 4.2.

For subsystems $i = 3, \dots, 8$ we have smooth power production expressions (4) that can be directly linearized without introducing virtual flows:

$$\hat{p}_i(x(k), q(k)) = k_{D_i} q_{D_i}^{\text{ss}} \Delta x_{D_i}^{\text{ss}} + k_{D_i} q_{D_i}^{\text{ss}} (\Delta x_{D_i}(k) - \Delta x_{D_i}^{\text{ss}}) + k_{D_i} \Delta x_{D_i}^{\text{ss}} (q_{D_i}(k) - q_{D_i}^{\text{ss}}).$$

4.2 HPV Optimization Problem Formulation

In this section we will formulate an optimization problem of the form (7) that can be used for power reference tracking in the HPV benchmark using MPC. We have obtained a linear discrete-time dynamical system (14)-(15) for the HPV with state variables x^r and control variables q^r . The constraints are upper and lower bounds on the outputs and inputs and their values can be found in [Savorgnan and Diehl, 2011]. Using the transformations matrices T_i^r and $T_i^{\text{r,inv}}$, these constraints can readily be recast as linear constraints for the reduced order model variables x^r, q^r . The power reference problem formulation (1) specifies a quadratic cost on states and control variables and a 1-norm penalty on deviations from the provided power reference, p^{ref} . For control horizon, N , this optimization problem can be written as

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{x}_a} \quad & \sum_{t=0}^{N-1} \left\{ \sum_{i=1}^8 [x_i^r(k)^T Q_i x_i^r(k) + q_i^r(k)^T R_i q_i^r(k)] + \gamma \|x_a(k)\|_1 \right\} \quad (17) \\ \text{s.t.} \quad & (14), (15) \quad k = 0, \dots, N-1 \quad i = 1, \dots, 8 \\ & C_i^r x_i^r(k) \in \mathcal{Y}_i \quad k = 0, \dots, N-1 \quad i = 1, \dots, 8 \\ & q_i(k) \in Q_i \quad k = 0, \dots, N-1 \quad i = 1, \dots, 8 \\ & x_a(k) = p^{\text{ref}}(k) - \sum_{i=1}^8 \hat{p}_i(x^r(k), q^r(k)) \quad k = 0, \dots, N-1 \\ & q_{C_{\text{IT}}}(k) q_{C_{\text{IP}}}(k) = 0 \quad k = 0, \dots, N-1 \quad i = 1, \dots, 2 \end{aligned}$$

where \mathcal{Y}_i and Q_i are sets representing the local output and input constraints, the additional variable x_a captures the power reference tracking mismatch, and the notation \mathbf{x} represents the stack of variables $x_i^r(k)$ and $q_i^r(k)$ for all i and k , while \mathbf{x}_a is the stacked variable of $x_a(k)$ for all k . Note that we can write $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_8^T]^T$ where each $\mathbf{x}_i, i = 1, \dots, 8$ includes all the variables that belong to subsystem i .

Power Reference Division

Since the original cost function contains a non-separable 1-norm term, the power reference constraints in the optimization problem (17) are coupled between all subsystems. This implies that Algorithm 1 requires some global communication even though the only information that is sent to the

global coordinator is $\hat{p}_i(x^r(k), q^r(k))$ for $k = 0, \dots, N - 1$ from each subsystem $i = 1, \dots, 8$.

In order to obtain a suitable dual problem, we first need to reformulate the cost function in a separable form. For the sake of brevity, we focus on one sampling step and drop the time index k . Thus for now our simplified objective is to decompose the following problem:

$$\min_{\{\mathbf{x}_i\}_{i=1,\dots,8}} \left| p^{\text{ref}} - \sum_{i=1}^8 P_i \mathbf{x} \right| \quad (18)$$

with $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_8^T]^T$, and P_i the matrix coefficient such that the power function produced or consumed by each subsystem $\hat{p}_i(x^r(k), q^r(k))$ is linearized as $P_i \mathbf{x}(k)$.

In this section we present two different ways that avoid global communication when solving this problem. In the first approach, we divide and distribute the global power reference to the subsystems in a static fashion. In the second approach, we show how the subsystems can trade power references between neighbors to achieve a satisfactory centralized reference tracking.

Static local power references. The idea here is straightforward. We divide the global power reference into local ones, i.e., p^{ref} is divided into local parts p_i^{ref} , $i = 1, \dots, 8$. We have chosen to compute p_i^{ref} such that it satisfies

$$\frac{p_i^{\text{ref}}(k)}{\sum_{i=1}^8 p_i^{\text{ref}}(k)} = \frac{p_i(x^{\text{ss}}, q^{\text{ss}})}{\sum_{i=1}^8 p_i(x^{\text{ss}}, q^{\text{ss}})}, \quad \text{for } k = 0, \dots, N - 1 \quad (19)$$

with $p_i(x^{\text{ss}}, q^{\text{ss}})$ the power produced by subsystem i in the steady-state condition.

This means that the fraction of the total power reference given to subsystem i is constant. The optimization problem is changed accordingly, i.e., the following cost function can be used instead of (18):

$$\min_{\{\mathbf{x}_i\}_{i=1,\dots,8}} \sum_{i=1}^8 \left| p_i^{\text{ref}} - P_i \mathbf{x} \right| \quad (20)$$

with $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_8^T]^T$. This allows for a distributed implementation since the matrix P_i introduces only local couplings, i.e., subsystem i needs only neighboring and local water levels and local water flows to compute the corresponding power output. The disadvantage of the static power reference division is that the global power reference tracking is not very accurate, as will be shown in the simulations section.

Dynamic local power references. The static power division essentially means that each subsystem always tracks a fraction of power reference that is equal to the proportion it produces in the steady-state condition. When the total power reference deviates significantly from the steady-state power, this idea may not work well since the proportional change of the local power reference can lead to sub-optimal performance. Inspired by an idea in [Madjidian *et al.*, 2011], we now introduce the dynamic power division, in which the subsystems have more flexibility in choosing the appropriate local power reference to be tracked. The main idea is that each subsystem will exchange power references with its direct neighbors.

Let us define for each pair (i, j) with $j \in \mathcal{N}_i$ a node that is in charge of determining the power exchange variable between subsystems i and j , denoted by δ_{ij} if node i is in charge and by δ_{ji} if node j is in charge². Then, for each subsystem we form the set³:

$$\Delta_i = \{j \mid j \in \mathcal{N}_i, i \text{ is in charge of } \delta_{ij}\}. \quad (21)$$

Now we replace (18) by the following cost function:

$$\min_{\{\mathbf{x}_i, \delta_i\}_{i=1, \dots, 8}} \sum_{i=1}^8 \left| p_i^{\text{ref}} + \sum_{j \in \Delta_i} \delta_{ij} - \sum_{j \in \mathcal{N}_i \setminus \Delta_i} \delta_{ji} - P_i \mathbf{x} \right| \quad (22)$$

with δ_i the vector containing all $\delta_{ij}, j \in \Delta_i$, and p_i^{ref} the nominal power reference for subsystem i . In words, the local power reference for each subsystem i deviates from the nominal value by adding the exchange amounts of the links that i manages and subtracting the exchange amounts of the links that affect i but that are decided upon by its neighbors. Note that problem (22) has a sparse structure that complies with the existing sparse structure of the HPV system, i.e., this method does not expand the neighborhood set of each subsystem.

The advantage of this dynamic power division is that it makes use of the existing network topology to form a sparse cost function, and the total power reference is preserved even if the local power references can deviate from the nominal values, i.e., we always have:

$$\sum_{i=1}^8 \left\{ p_i^{\text{ref}} + \sum_{j \in \Delta_i} \delta_{ij} - \sum_{j \in \mathcal{N}_i \setminus \Delta_i} \delta_{ji} \right\} = p^{\text{ref}}. \quad (23)$$

²Note that here we discuss the power division for each sampling step, i.e., there are $\delta_{ij}(k)$ or $\delta_{ji}(k)$ with $k = 0, \dots, N - 1$.

³A simple way is to let the subsystem with smaller index lead the exchange, i.e., $\Delta_i = \{j \mid j \in \mathcal{N}_i, j > i\}$.

Table 1. Neighborhoods of subsystems (\mathcal{N}_i)

Subsystem	GLOBAL-REF	LOC-REF-DYN	LOC-REF-STAT
1	{1, ..., 8}	{1, 3, 4}	{1, 3, 4}
2	{1, ..., 8}	{2, 6, 7}	{2, 6, 7}
3	{1, ..., 8}	{3, 1, 4}	{3, 1, 4}
4	{1, ..., 8}	{4, 1, 3, 5}	{4, 1, 3, 5}
5	{1, ..., 8}	{5, 4, 6}	{5, 4, 6}
6	{1, ..., 8}	{6, 2, 7, 5}	{6, 2, 7, 5}
7	{1, ..., 8}	{7, 2, 6, 8}	{7, 2, 6, 8}
8	{1, ..., 8}	{8, 7}	{8, 7}

Now that we have a separable cost function by using either a static or a dynamic power division technique, we can cast the approximate optimization problem in the form (7) that has a separable dual problem, and apply Algorithm 1 at every sampling step. However, due to the requirement of positive definiteness of the quadratic term in the objective function, the introduced power exchange variables δ_{ij} must be penalized with a positive definite quadratic term. This implies that power reference exchange has an associated cost.

Communication structures. In the preceding sections we have presented three different ways to handle the power reference term. The first is the one with centralized power reference term which we hereby denote by GLOBAL-REF. The second is the one with static local power references which we denote by LOC-REF-STAT. The third is the dynamic local power reference which from here on is denoted by LOC-REF-DYN. In Table 1 we provide an overview of the neighborhood sets \mathcal{N}_i for the different power reference tracking schemes. We can see that all subsystems have the same neighborhood sets for the dynamic local reference tracking and the static local reference tracking.

Relaxation of Nonlinear Constraint

The second issue that hinders the optimization problem (17) from being solved using Algorithm 1 are the nonlinear constraints $q_{C_{iT}}(k)q_{C_{iP}}(k) = 0$ with $i = 1, 2$. In this section we present a way to relax these constraints.

Assuming a diagonal cost, we have the following penalty on the pump and turbine action in ducts C_i , $i = 1, 2$

$$R_{C_i} = \begin{bmatrix} R_{C_{iT}} & 0 \\ 0 & R_{C_{iP}} \end{bmatrix}. \quad (24)$$

We also have the constraints that

$$q_{C_{iP}}(k) \geq 0, \quad q_{C_{iT}}(k) \geq 0, \quad q_{C_{iT}}(k)q_{C_{iP}}(k) = 0.$$

We relax this by removing the nonlinear constraint and adding a cross-penalty $\alpha \sqrt{R_{C_{iP}}R_{C_{iT}}}$ for some $\alpha \in (0, 1)$ in the cost function, i.e., we set

$$R_{C_i} = \begin{bmatrix} R_{C_{iT}} & \alpha \sqrt{R_{C_{iP}}R_{C_{iT}}} \\ \alpha \sqrt{R_{C_{iP}}R_{C_{iT}}} & R_{C_{iP}} \end{bmatrix}. \quad (25)$$

This relaxation is implementable using the proposed algorithm since the nonlinear constraint is removed and replaced by a cross-penalty. The cross-penalty gives an additional cost if both $q_{C_{iT}}$ and $q_{C_{iP}}$ are non-zero. The closer α is to 1, the larger the penalty. For $\alpha \geq 1$ it is easily verified that we lose strong convexity on the quadratic cost function, i.e., R_{C_i} loses positive definiteness and such choices for α are therefore prohibited.

The relaxation is not equivalent to the original nonlinear constraint and thus cannot guarantee that the nonlinear constraint is respected using this relaxation. However, it turns out that the optimal solution using the cross-penalty in the cost (25) in most simulated cases coincides with the optimal solution when the nonlinear constraint $q_{C_{iT}}(k)q_{C_{iP}}(k) = 0$ is enforced and the original diagonal cost (24) is used. In some cases, however, the optimal solution using the relaxation does not respect the nonlinear constraint. To address this, a two-phase optimization strategy is developed and presented next.

Two-phase Optimization

We propose a two-phase optimization strategy as an ad-hoc branch and bound optimization routine that uses two consecutive optimizations. In the first optimization, the relaxed optimization problem is solved. If the nonlinear constraints are respected, i.e., we get a solution that satisfies $q_{C_{iT}}(k)q_{C_{iP}}(k) = 0$, the global optimal solution for the non-relaxed problem is found. If some of the nonlinear constraints do not hold, the optimization routine is restarted with the smallest flow, $q_{C_{iT}}(k)$ or $q_{C_{iP}}(k)$ for $i = 1, 2$, $k = 0, \dots, N - 1$, set to zero. The resulting algorithm is summarized below.

ALGORITHM 2—DISTRIBUTED BRANCH AND BOUND ALGORITHM

1. Solve the relaxed problem using Algorithm 1
2. **If** $q_{C_{iT}}(k) > q_{C_{iP}}(k)$, $i = 1, 2, k = 0, \dots, N - 1$
 Add constraint: $q_{C_{iP}}(k) = 0$
 Else
 Add constraint: $q_{C_{iT}}(k) = 0$
 End

3. Solve relaxed problem using Algorithm 1 with the additional flow constraints

□

This ad-hoc branch and bound technique does not theoretically guarantee that the optimal flow directions are chosen. However, we can guarantee that the nonlinear constraints are always satisfied. Further, for the distributed MPC formulation we will see in the simulations section that the global optimal solution for the non-relaxed problem is found at every time step using this branch and bound algorithm.

4.3 Distributed Estimation

From Section 2 we know that not all states can be measured, which implies that an observer needs to be used to feed an initial condition to the optimizer. The reduced-order linear model (14)-(15) has local dynamics and outputs only, which implies that an observer can be designed in decentralized fashion. We introduce the local estimate \hat{x}_i^r and the local observer-gain K_i and the following local observer dynamics

$$\hat{x}_i^r(k+1) = A_{ii}^r \hat{x}_i^r(k) + B_i^r q^r(k) + K_i(y_i^r(k) - C_i^r \hat{x}_i^r(k)).$$

Because of the sparse structure of B_i^r this observer can be implemented in a distributed fashion where only the inflows to subsystem i need to be communicated. The estimation error $\tilde{x}_i^r = x_i^r - \hat{x}_i^r$ has local error dynamics

$$\tilde{x}_i^r(k+1) = (A_{ii}^r - K_i C_i^r) \tilde{x}_i^r(k).$$

Thus, the observer can be designed in a decentralized fashion and be implemented in a distributed fashion.

5. Simulation Results

We perform distributed MPC simulations of the hydro power valley using 3 different ways of handling the power reference: GLOBAL-REF, LOC-REF-DYN, and LOC-REF-STAT, using the proposed Algorithm 2. We also solve the problem (17) using a state-of-the-art MIQP-solver, namely CPLEX. In CPLEX the nonlinear constraints given in (17) can be addressed by introducing binary variables. More specifically, for each duct $C_i, i = 1, 2$, we define two virtual flows, $q_{C_{iP}}$ and $q_{C_{iT}}$, and require that

both values are nonnegative. Each virtual flow has a maximum capacity, hence the constraints for these flows are:

$$\begin{aligned} 0 &\leq q_{C_{iP}} \leq q_{C_{iP}}^{\max} \\ 0 &\leq q_{C_{iT}} \leq q_{C_{iT}}^{\max} \end{aligned} \quad (26)$$

We introduce binary variables $b_i \in \{0, 1\}$ and impose the following constraints:

$$\begin{aligned} q_{C_{iT}} &\leq q_{C_{iT}}^{\max} b_i \\ q_{C_{iP}} &\leq q_{C_{iP}}^{\max} (1 - b_i) \end{aligned} \quad (27)$$

The constraints (26) and (27) ensure that either $q_{C_{iP}} = 0, q_{C_{iT}} \geq 0$ (if $b_i = 1$) or $q_{C_{iT}} = 0, q_{C_{iP}} \geq 0$ (if $b_i = 0$).

This formulation results in an MIQP for which there are efficient Branch-and-Bound algorithms implemented in CPLEX. To make the 1-norm term in (17) fit the MIQP-formulation used in CPLEX we introduce auxiliary variables v and use the following equivalent reformulation

$$\begin{aligned} \min_x \|Px - p\|_1 &\iff \min_{x,v} 1^T v \\ &\text{s.t. } -v \leq Px - p \leq v \end{aligned}$$

We also compare the proposed distributed MPC method to a decentralized MPC approach in which each subsystem solves its own local MPC problem without any communication, in order to show the advantage of DMPC w.r.t. decentralized MPC.

5.1 Simulation Details

We use the original nonlinear continuous model presented in [Savorgnan and Diehl, 2011] as simulation model. The ode-solver *ode15s* in MATLAB is used to perform the simulations. A MATLAB function that computes the derivatives needed by *ode15s* is provided in the benchmark package [Savorgnan and Diehl, 2011]. The control system consists of the distributed observer from Section 4.3 which feeds Algorithm 2, with estimates of the current state.

Besides the mismatch between the model used for control and the model used for simulation, we have also added bounded process noise to capture mismatch between the simulation model and the real plant. The magnitude of the worst case process noise was chosen to be 1% of the steady-state level x^{ss} . We also use bounded additive measurement noise where the measured water levels are within ± 3 cm from the actual water levels.

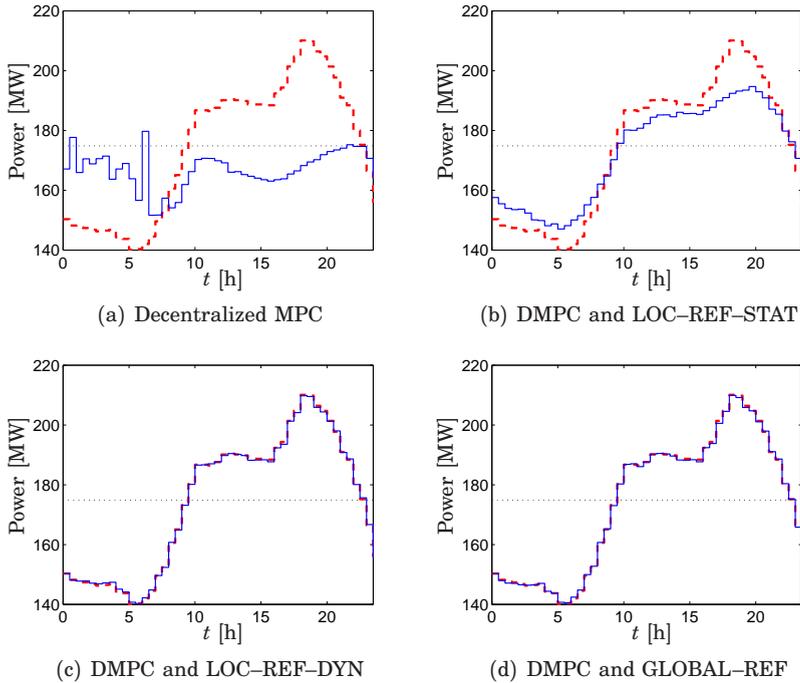


Figure 2. Comparison of power reference tracking performance using DMPC and decentralized MPC approaches. Solid lines: produced power, dashed lines: reference power, dotted lines: steady state power.

We use a sampling time of 30 minutes in all simulations and the control horizon is $N = 10$, i.e., 5 hours. The simulations are performed over a 24 hour period since the power reference trajectories are periodic with this interval.

All simulations and optimizations were implemented on a PC running MATLAB on Linux with an Intel(R) Core(TM) i7 CPU running at 3 GHz and with 4 GB RAM.

5.2 Control Performance Comparison

The power reference tracking results are plotted in Figures 2(a)–2(d) where the full power reference and the sum of the local power productions are plotted. The scheme GLOBAL-REF achieves very good tracking performance, while the scheme LOC-REF-STAT shows a significant deterioration in tracking performance. However, the introduction of the possibility to exchange power references in LOC-REF-DYN between sub-

Table 2. Comparison of computation time between Algorithm 2 and CPLEX for 48 instances of the same problem

	Algorithm 2	CPLEX for MIQP	CPLEX for QP
min t (s)	0.023	0.087	0.049
max t (s)	0.086	0.121	0.089
average t (s)	0.054	0.098	0.063
std dev t (s)	0.017	0.009	0.009

systems restores the very good tracking performance while keeping the computations distributed. The tracking performance of the decentralized MPC approach is very poor, due to the lack of communications. Hence, it is recommended not to use a decentralized MPC approach, unless communication is prohibited due to the lack of communication facilities or due to the policy of different authorities.

In Appendices A.1 and A.2 there are figures that show the input and output evolutions and the corresponding constraints with the scheme LOC-REF-DYN. We can observe that all constraints are satisfied despite disturbances, model mismatch, and the use of an observer. For the schemes GLOBAL-REF and LOC-REF-STAT all the constraints on the inputs and outputs are also satisfied.

5.3 Computational Efficiency

In Table 2 we provide a comparison of the execution times of the centralized MPC problem (17). We compare the distributed Algorithm 2 to the solver CPLEX when solving (17), i.e., with power-division GLOBAL-REF. To solve this problem using CPLEX, an MIQP formulation is used, and to solve the problem using Algorithm 2, the relaxed problem is solved twice. We also compare the above execution times to the case when we solve the first relaxed problem in Algorithm 2, which is a QP, using CPLEX. At each sampling step, the same problem is solved with the different solvers, and the execution times t are measured. Although in this example the solvers easily solve the problem within the time frame of the sampling time, we can see that the computation time for our MATLAB-implemented algorithm is lower than the C-implemented CPLEX for both the MIQP and QP cases. As previously discussed, Algorithm 2 cannot guarantee that the global optimum for (17) is found. However, in the DMPC simulations presented in this section the global optimum of (17) is found at every sampling step using Algorithm 2.

Table 3. Number of iterations to solve the MPC optimization in one step

	Alg. 1 with GLOBAL-REF	Alg. 1 with LOC-REF-DYN	Alg. 1 with LOC-REF-STAT
average n_{iter}	311.3	579.1	942.5
max n_{iter}	498	1054	2751
std dev n_{iter}	93.8	210.9	440.8

5.4 Communication Requirements

The sizes of the optimization problems using power reference division GLOBAL-REF, LOC-REF-DYN or LOC-REF-STAT are almost equal. Comparing GLOBAL-REF to LOC-REF-STAT we get some additional constraints due to the power reference division and comparing LOC-REF-DYN to LOC-REF-STAT we get some additional decision variables δ_{ij} to enable distributed power reference re-assignment.

In Table 3 the number of iterations n_{iter} needed to obtain the solution is presented. The mean and max values of n_{iter} and the standard deviation are computed using 48 simulation steps, i.e., 24 hours. We can notice that different DMPC schemes converge with different average numbers of iterations. The reason is that for LOC-REF-STAT it is more difficult to satisfy the different 1-norm terms with equality, i.e., to follow the local power references. This implies that the corresponding dual variable ν becomes large (close or equal to γ) and it takes more iterations to achieve convergence. As a result, the scheme LOC-REF-STAT with a simpler communication structure might require more communication resources than e.g., GLOBAL-REF, which has a more complicated communication structure but needs fewer iterations.

The scheme LOC-REF-DYN performs very well in terms of communication, computation, as well as performance aspects and is therefore the chosen candidate for distributed implementation for the given case study.

6. Conclusions and Future Work

The proposed distributed MPC approach has been applied to the power reference tracking problem of the HD-MPC hydro power valley benchmark. Two distributed schemes have been compared to centralized and decentralized MPC methods. We have provided relaxations and approximations for the original nonlinear nonsmooth problem formulation as well as proposed a way to follow a centralized power reference in a distributed fashion. Furthermore, we have presented a practical branch-and-bound

algorithm that solves all optimization problems encountered in the simulations and achieves as good performance as the centralized MPC that is known to have global optimum. The simulation results show that the introduced approximations and relaxations capture the behavior of the system well and that very good control performance is achieved. Finally, a comparison to state-of-the-art optimization software (CPLEX) shows that the proposed algorithm has better execution times in general.

As the next step before implementation in real plants, the proposed distributed MPC approach should be tested against different hydraulic scenarios and other HPV setups. To cope with varying water flows entering the system, these should be estimated and compensated for. Furthermore, a weather model could be included that estimates the future inflows to the system.

7. Acknowledgments

The authors were supported by the European Union Seventh Framework STREP project “Hierarchical and distributed model predictive control (HD-MPC)” with contract number INFSO-ICT-223854, the European Union Seventh Framework Programme under grant agreement no. 257462 HYCON2 Network of Excellence, the BSIK project “Next Generation Infrastructures (NGI)”, and the Swedish Research Council through the Linnaeus center LCCC.

References

- Anand, A., G. Joshua, S. Sundaramoorthy, and L. Samavedham (2011): “Coordinating multiple model predictive controllers for multi-reservoir management.” In *Proceedings of the 2011 IEEE International Conference on Networking, Sensing and Control*, pp. 1–6. Delft, The Netherlands.
- Beck, A. and M. Teboulle (2009): “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” *SIAM Journal on Imaging Sciences*, **2:1**, pp. 183–202.
- Carpentier, P. and G. Cohen (1993): “Applied mathematics in water supply network management.” *Automatica*, **29:5**, pp. 1215 – 1250.
- Doan, D., T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter (2009): “A distributed version of Han’s method for DMPC using local communications only.” *Control Engineering and Applied Informatics*, **11:3**, pp. 6–15.
- Doan, M. D., T. Keviczky, and B. De Schutter (2011): “An iterative scheme for distributed model predictive control using Fenchel’s duality.” *Journal of Process Control*, **21:5**, pp. 746–755. Special Issue on Hierarchical and Distributed Model Predictive Control.
- Fawal, H., D. Georges, and G. Bornard (1998): “Optimal control of complex irrigation systems via decomposition-coordination and the use of augmented Lagrangian.” In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 3874–3879. San Diego, CA.
- Georges, D. (1994): “Decentralized adaptive control for a water distribution system.” In *Proceedings of the Third IEEE Conference on Control Applications*, pp. 1411–1416 vol.2. Glasgow, UK.
- Giselsson, P., M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer (2012): “Accelerated gradient methods and dual decomposition in distributed model predictive control.” *Automatica*. To appear. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Giselsson, P. and A. Rantzer (2010): “Distributed model predictive control with suboptimality and stability guarantees.” In *Proceedings of the 49th Conference on Decision and Control*, pp. 7272–7277. Atlanta, GA.
- Gomez, M., J. Rodellar, and J. A. Mantecon (2002): “Predictive control method for decentralized operation of irrigation canals.” *Applied Mathematical Modelling*, **26:11**, pp. 1039–1056.

- Gugercin, S. and A. C. Antoulas (2004): “A survey of model reduction by balanced truncation and some new results.” *International Journal of Control*, **77:8**, pp. 748–766.
- Igreja, J., F. Cadete, and J. Lemos (2011): “Application of distributed model predictive control to a water delivery canal.” In *Proceedings of the 19th Mediterranean Conference on Control & Automation*, pp. 682–687. Corfu, Greece.
- Igreja, J. and J. Lemos (2009): “Nonlinear model predictive control of a water distribution canal pool.” In Magni *et al.*, Eds., *Nonlinear Model Predictive Control*, vol. 384 of *Lecture Notes in Control and Information Sciences*, pp. 521–529. Springer, Berlin / Heidelberg, Germany.
- Litrico, X. and V. Fromion (2009): *Modeling and Control of Hydrosystems*. Springer, London, UK.
- Madjidian, D., K. Mårtensson, and A. Rantzer (2011): “A distributed power coordination scheme for fatigue load reduction in wind farms.” In *Proceedings of 2011 American Control Conference*, pp. 5219–5224. San Francisco, CA.
- Mareels, I., E. Weyer, S. K. Ooi, M. Cantoni, Y. Li, and G. Nair (2005): “Systems engineering for irrigation systems: Successes and challenges.” *Annual Reviews in Control*, **29:2**, pp. 191–204.
- Nederkoorn, E., J. Schuurmans, and W. Schuurmans (2011): “Continuous nonlinear model predictive control of a hybrid water system - Application of DNPC to a Dutch polder.” In *Proceedings of the 2011 IEEE International Conference on Networking, Sensing and Control*, pp. 209–214. Delft, The Netherlands.
- Negenborn, R., B. De Schutter, and J. Hellendoorn (2008): “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes.” *Engineering Applications of Artificial Intelligence*, **21:3**, pp. 353–366.
- Negenborn, R. R., P. J. van Overloop, T. Keviczky, and B. De Schutter (2009): “Distributed model predictive control for irrigation canals.” *Networks and Heterogeneous Media*, **4:2**, pp. 359–380.
- Nesterov, Y. (1983): “A method of solving a convex programming problem with convergence rate $O(1/k^2)$.” *Soviet Mathematics Doklady*, **27:2**, pp. 372–376.
- Nesterov, Y. (1988): “On an approach to the construction of optimal methods of minimization of smooth convex functions.” *Ekonomika i Matematicheskie Metody*, **24**, pp. 509–517. In Russian.

- Nesterov, Y. (2005): “Smooth minimization of non-smooth functions.” *Mathematical Programming*, **103:1**, pp. 127–152.
- Petrone, F. (2010): “Model predictive control of a hydro power valley.”. Master’s thesis, Politecnico di Milano, Italy.
- PEW Center on Global Climate Change (2011): “Hydropower - Climate TechBook.” Available: http://www.c2es.org/docUploads/Hydropower_0.pdf.
- Qin, S. J. and T. A. Badgwell (2003): “A survey of industrial model predictive control technology.” *Control Engineering Practice*, **11:7**, pp. 733–764.
- Sahin, A. and M. Morari (2010): “Decentralized model predictive control for a cascade of river power plants.” In Negenborn *et al.*, Eds., *Intelligent Infrastructures*, vol. 42 of *Intelligent Systems, Control and Automation: Science and Engineering*, pp. 463–485. Springer Netherlands.
- Savorgnan, C. and M. Diehl (2011): “Control benchmark of a hydropower plant.” Technical Report. Hierarchical and Distributed Model Predictive Control for Large-Scale Systems (HD-MPC). Available at http://www.ict-hd-mpc.eu/benchmarks/HD-MPC_hydropower_valley_public_benchmark.zip.
- Savorgnan, C., C. Romani, A. Kozma, and M. Diehl (2011): “Multiple shooting for distributed systems with applications in hydro electricity production.” *Journal of Process Control*, **21:5**, pp. 738 – 745. Special Issue on Hierarchical and Distributed Model Predictive Control.
- Sawadogo, S., R. Faye, P. Malaterre, and F. Mora-Camino (1998): “Decentralized predictive controller for delivery canals.” In *Proceedings of 1998 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3880–3884 vol.4. San Diego, CA.
- Toh, K.-C. and S. Yun (2010): “An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems.” *Pacific Journal of Optimization*, **6**, pp. 615–640.
- Tseng, P. (2008): “On accelerated proximal gradient methods for convex-concave optimization.” Submitted to *SIAM Journal on Optimization*. Available: <http://www.math.washington.edu/~tseng/papers/apgm.pdf>.
- van Overloop, P.-J. (2006): *Model Predictive Control on Open Water Systems*. PhD thesis, Delft University of Technology, The Netherlands.

- van Overloop, P.-J., A. J. Clemmens, R. J. Strand, R. M. J. Wagemaker, and E. Bautista (2010): “Real-time implementation of model predictive control on Maricopa-Stanfield irrigation and drainage district’s WM canal.” *Journal of Irrigation and Drainage Engineering*, **136:11**, pp. 747–756.

A. Appendix

A.1

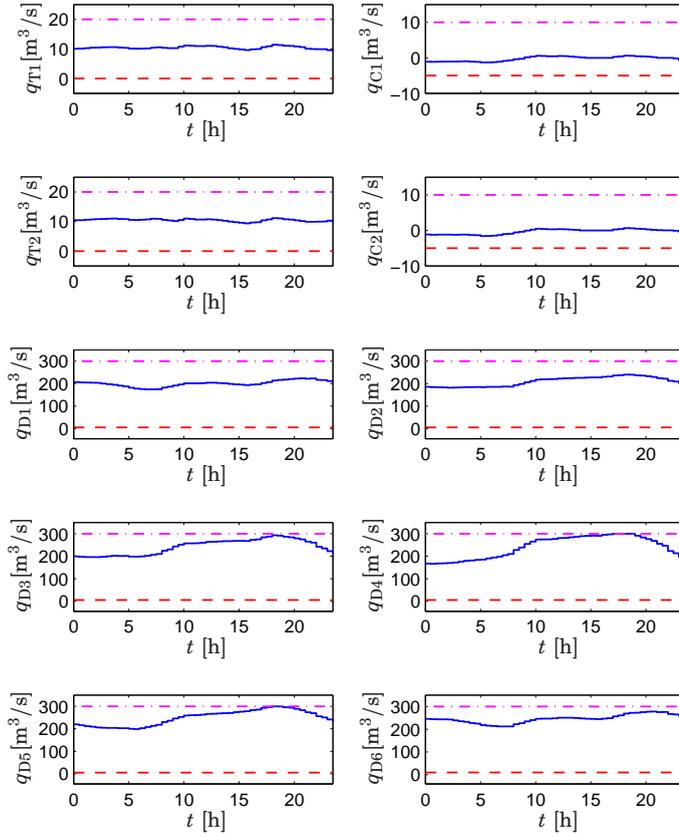


Figure 3. Input constraint satisfaction using Algorithm 2 and power division LOC-REF-DYN. Dash-dotted lines: upper bounds, dashed lines: lower bounds.

A.2

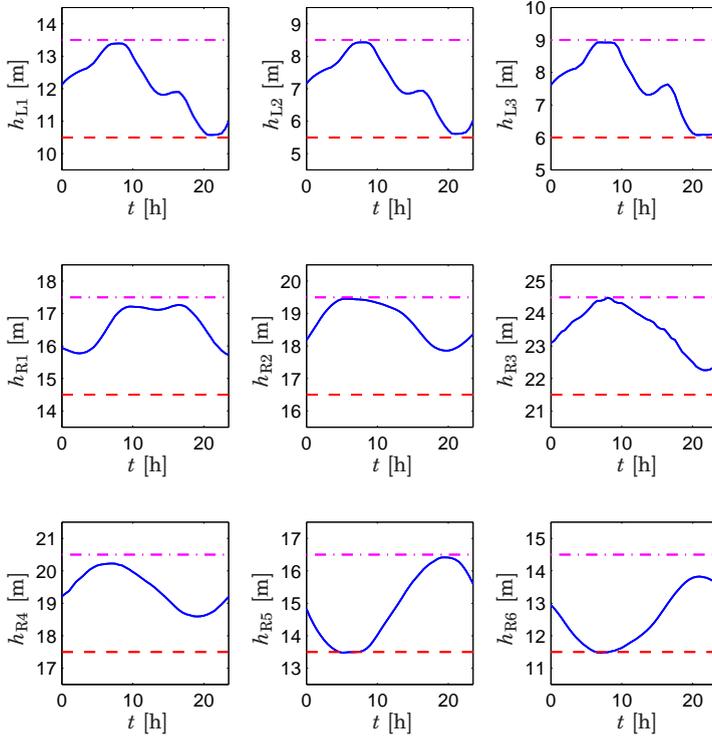


Figure 4. Output constraint satisfaction using Algorithm 2 and power division LOC-REF-DYN. Dash-dotted lines: upper bounds, dashed lines: lower bounds.

Paper VIII

Gradient-Based Model Predictive Control in a Pendulum System

Pontus Giselsson

Abstract

Model predictive control (MPC) is applied to a physical pendulum system consisting of a pendulum and a cart. The objective of the MPC controller is to steer the system towards precomputed, time-optimal feedforward trajectories that move the system from one stationary point to another. The sample time of the controller sets hard limitations on the execution time of the optimization algorithm in the MPC controller. The MPC optimization problem is stated as a quadratic program, which is solved using the algorithm presented in [Giselsson, 2012a]. The algorithm in [Giselsson, 2012a] is an accelerated gradient method that is applied to solve a dual formulation of the MPC optimization problem. Experiments show that the optimization algorithm is efficient enough to be implemented in a real-time pendulum application.

Technical report based on and extending the conference paper

Giselsson, P. (2011): “Model predictive control in a pendulum system.” In *Proceedings of the 31st IASTED Conference on Modelling, Identification and Control*. Innsbruck, Austria.

and with some material from the conference paper

Giselsson, P., J. Åkesson, and A. Robertsson (2009): “Optimization of a pendulum system using Optimica and Modelica.” In *Proceedings of the 7th International Modelica Conference 2009*, pp 480–489. Como, Italy.

1. Introduction

Model predictive control (MPC) is a widely recognized control methodology for control of complex systems with state and control constraints. The idea of model predictive control is to determine a control trajectory by minimizing a cost function based on predictions of future states over a finite time interval, with the current state of the system as initial condition. The first control action in the obtained trajectory is applied to the system. When new state measurements become available, the optimization procedure is repeated with the new measurements as initial values to the state predictions. There are hard timing constraints on the optimization routine before the control action must be applied. Solving an optimization problem can be a time consuming task, which is why MPC has traditionally been considered a control methodology for systems with relatively slow dynamics. Over the past decade, faster computers and increasingly efficient algorithms have been developed. This development has enabled for systems with faster dynamics to be controlled using MPC. If the system dimensions are small, explicit MPC can be used, [Bemporad *et al.*, 2002, Bemporad and Filippi, 2001] for linear systems and [Johansen, 2004] for systems with nonlinear dynamics, to speed up online execution times. In [Wang and Boyd, 2010] the structure and sparsity inherent in MPC optimization problems are exploited to reduce the execution time when using an interior point algorithm to solve the online optimization problem. For systems with input constraints only, accelerated gradient methods are used to solve the resulting MPC optimization problem in [Richter *et al.*, 2009]. For more on MPC see [Maciejowski, 2002, Morari and Lee, 1999, Rawlings and Mayne, 2009], and for examples of industrial processes that have successfully been controlled using MPC see [Camacho and Bordons, 1997].

In this paper optimal control and model predictive control of a pendulum system is considered. The pendulum system consists of a cart, which is mounted on a track, and has a pendulum attached to it. The track length sets limitations on the cart movement. Two minimum time optimization problems for the pendulum system are posed: one swing-up problem and one translation problem with constraints on the location of the pendulum end-point. The solutions to the minimum time optimization problems are precomputed and used as feedforward trajectories. We introduce feedback by designing an MPC controller with the objective to steer the system towards the optimal feedforward trajectories. The model used in the MPC optimization problem is a time dependent linear system that is obtained by linearizing the nonlinear pendulum dynamics around the precomputed feedforward trajectories. We use a quadratic cost and linear constraints in the MPC optimization problem. This gives a quadratic pro-

gram to be solved in the MPC controller. To solve the quadratic program the algorithm presented in [Giselsson, 2012a] is used. The algorithm is an accelerated gradient method applied to a dual formulation of the optimization problem.

This report is based on the material in [Giselsson, 2011] and to some extent on the material in [Giselsson *et al.*, 2009]. The paper is organized as follows. In Section 2, the problem formulated is stated. Section 3 describes the pendulum system. In Section 4 the minimum-time optimization problems are stated and the optimal trajectories are plotted. The model predictive controller is described in Section 5 and experimental results are presented. Finally, in Section 6 the paper is concluded.

2. Problem Setup

The problem considered in this paper is to achieve time optimal transitions through the nonlinear dynamics of the pendulum system. We use the following minimum time optimization formulation

$$\begin{aligned}
 & \text{minimize} && t_f \\
 & \text{subject to} && \dot{x} = f(x) + g(x)u \\
 & && (x, u) \in \mathcal{X} \\
 & && x(0) = x_0 \\
 & && x(t_f) = x_{t_f}
 \end{aligned} \tag{1}$$

where $f(x)$ and $g(x)$ describes the nonlinear dynamics of the pendulum system. The optimization objective is to minimize the transition time, t_f , between the initial state, $x(0)$, and the terminal state, $x(t_f)$, while satisfying state and control constraints defined by the set \mathcal{X} . We consider two different minimum time optimization problems. The first problem concerns swing-up of the pendulum. The second problem is to move the cart from one side of the track to the other with the pendulum starting and stopping in the downward position, while the end-point of the pendulum must avoid a prespecified fixed obstacle.

The resulting optimal control trajectories are applied to the pendulum system as feedforward control trajectories. The problem considered in this paper is to design an MPC controller that controls the system towards the precomputed optimal feedforward trajectories. The resulting optimization problem is solved using the method presented in [Giselsson, 2012a] in which an accelerated gradient method is applied to a dual formulation to the optimization problem. The dynamics of the system are relatively

fast, which sets requirements on the execution time of the optimization algorithm.

3. The Pendulum System

The pendulum system consists of a cart that is mounted on a track with a pendulum freely hanging from the cart. The length of the pendulum is $l = 0.4$ m. The cart is driven by a Faulhaber DC-motor and a rack and pinion to convert the rotating motion of the motor to linear motion of the cart. Further, the system is equipped with a Hall effect sensor to measure pendulum angle, a current sensor to measure motor current, and a magnetic motor encoder to extract position measurements of the cart. There are also two programmable Atmel ATmega 16 microprocessors mounted on the cart for control purposes. The first microprocessor can output motor voltage to the motor drive unit and receive current measurements. The second microprocessor receives the motor encoder signals and the angle measurement signal. The two microprocessors can communicate with each other and the second microprocessor communicates with Matlab/Simulink on a PC via the serial interface.

3.1 Cart Control

The motion of the cart is controlled in a cascaded control structure. See Figure 1 for a schematic view of the cascaded control structure. The innermost loop controls the current through the DC-motor. P_1 represents the current dynamics that is modeled as a first order dynamical system with a time constant of 0.17 ms. C_1 represents the current controller, which is a PI controller that controls the actual motor current, i , to its reference, i_r . This current controller runs at a sampling rate of 28.8 kHz on the first microprocessor. The current reference, i_r , is set by the outer control loop that controls the cart velocity. The current dynamics are fast in comparison to the velocity dynamics, which makes $i_r \approx i$ a good approximation. The transfer function from i to v , i.e., P_2 , is modeled as an integrator with a gain. The velocity dynamics are controlled with another

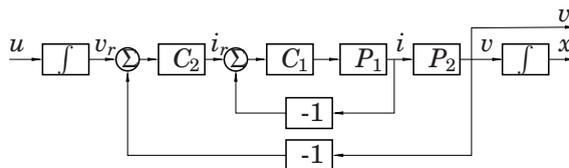


Figure 1. Cascaded control structure for the cart control.

PI-controller, C_2 . This controller runs on the second microprocessor at a sampling rate of 1 kHz. There are no velocity measurements available. A velocity estimate is obtained by applying a derivative filter to the position measurement in the micro processor at a frequency of 1 kHz. The reference to the velocity control loop, v_r , is received from Matlab/Simulink that runs on a PC. The velocity reference sent from the PC is updated at a frequency of about 50 Hz. We denote the corresponding sampling time by h . The settling time for the velocity controller is faster than the update frequency of the velocity reference. To avoid nonsmooth behavior of the cart, the velocity reference is updated in a first-order-hold manner in the microprocessor. That is, a piece-wise constant acceleration reference u is sent to the microprocessor. The velocity reference is updated internally in the microprocessor at the same rate as the velocity controller. The reference is updated according to $v_r(t) = v_r(t_0) + u(t_0)(t - t_0)$ where $v_r(t_0)$ is the integrated velocity reference and $u(t_0)$ is the acceleration reference at sampling time t_0 , and $t \in [t_0, t_0 + h]$.

This cascaded control structure is suitable when fast closed loop dynamics from v_r to v is desired. Since the PC communication is performed at a much slower frequency than the velocity controller updates, $v_r = v$ is a good approximation. Using this approximation, the cart motion can be modeled as a double integrator from acceleration reference to cart position.

3.2 System Modeling

Due to the low level control previously described, the cart position p depends on the control signal u according to

$$\ddot{p} = u. \quad (2)$$

A pendulum is attached to the cart. When the pendulum is swinging, reaction forces in the mounting point creates disturbances to the cart motion. These disturbances are attenuated by the cascaded control structure on the cart, making the double integrator model of the cart accurate despite these disturbances. The pendulum is modeled as a simple gravity pendulum in which the weight of the rod is neglected. The pendulum dynamics are well known; let θ be the pendulum angle and the dynamics are described by

$$\ddot{\theta} = -\frac{g}{l} \sin \theta + \frac{u}{l} \cos \theta, \quad (3)$$

where $\theta = 0$ is defined to be the pendulum downward position, g is the gravitational acceleration, l is the length of the pendulum, and u is the cart acceleration. The full system dynamics are described by Equations (2) and (3). Note that since the cart acceleration is used as control signal,

the cart and pendulum dynamics are decoupled. They can be seen as two separate dynamical systems that are driven by the same control signal.

The position of the cart and the pendulum angle are defined such that the pendulum end point in the horizontal direction, x_p , and in the vertical direction, y_p , are given by

$$x_p = p - l \sin \theta, \quad y_p = -l \cos \theta.$$

4. Optimal Feedforward Trajectories

Two different minimum-time optimal control problems are considered in this paper. The first problem is a minimum-time swing-up problem with additional constraints on cart position and control signal magnitude. The second problem is a path-constrained minimum-time problem. The optimization problems are solved using the JModelica.org platform [Åkesson *et al.*, 2009] which allow for solving dynamic optimization problems by specifying the dynamical model, the cost function and constraints using a high-level language. The optimal control problems and the solutions obtained by the JModelica.org platform are presented below. For more information on how these optimal control problems were solved, see [Giselsson *et al.*, 2009].

4.1 Pendulum Swing-Up

The optimization objective is to reach the inverted position as fast as possible, starting from the downward position. Further constraints include that the cart should start and stop at the same position. The cart and angular velocities should be zero when the pendulum has reached the inverted position. The applied control signal, i.e., the cart acceleration, u , is limited to be in the interval $\pm 5\text{m/s}^2$ and its derivative must satisfy $-100\text{m/s}^3 \leq \dot{u} \leq 100\text{m/s}^3$. Since the cart track is finite, the cart position must satisfy $-0.5\text{m} \leq p \leq 0.5\text{m}$. The optimization problem is stated mathematically below.

$$\begin{aligned} & \text{minimize} && t_f \\ & \text{subject to} && \ddot{\theta} = -\frac{g}{l} \sin \theta + \frac{u}{l} \cos \theta \\ & && \ddot{p} = u \\ & && -0.5 \leq p \leq 0.5 \\ & && |u| \leq 5 \quad |\dot{u}| \leq 100 \\ & && \theta(0) = 0 \quad \dot{\theta}(0) = 0 \quad p(0) = 0 \quad \dot{p}(0) = 0 \\ & && \theta(t_f) = \pi \quad \dot{\theta}(t_f) = 0 \quad p(t_f) = 0 \quad \dot{p}(t_f) = 0 \end{aligned} \tag{4}$$

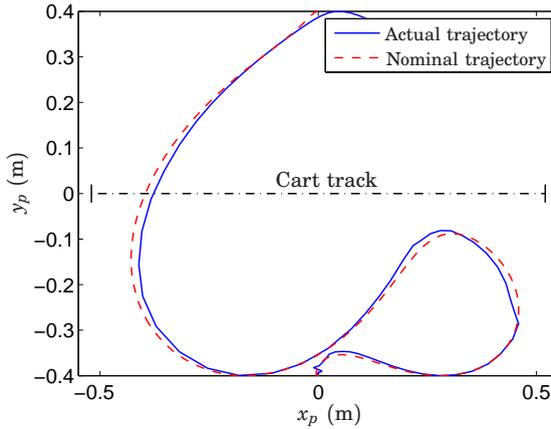


Figure 2. Pendulum end point trajectories for swing-up problem. Both the optimal trajectory and the trajectory obtained when optimal control trajectory is applied to the physical pendulum system in open loop, are plotted.

where t_f is the final time. To analyze the plant-model accuracy, the optimal feedforward trajectory was applied to the physical plant with the same initial conditions as in the optimization. The resulting pendulum end point trajectory, together with the optimal trajectory, is found in Figure 2. The optimal control trajectory for the swing-up example is found in Figure 4(a).

4.2 Optimization with Path-Constraints

In this optimization problem, the cart should move from one side of the track to the other side, while the end point of the pendulum must avoid an obstacle defined by

$$\left(\frac{x_p + 0.3}{0.05}\right)^2 + \left(\frac{y_p + 0.4}{0.3}\right)^2 = 1.$$

The pendulum should start and stop at rest in the downward position. Track and control limitations are equivalent to in the swing-up problem.

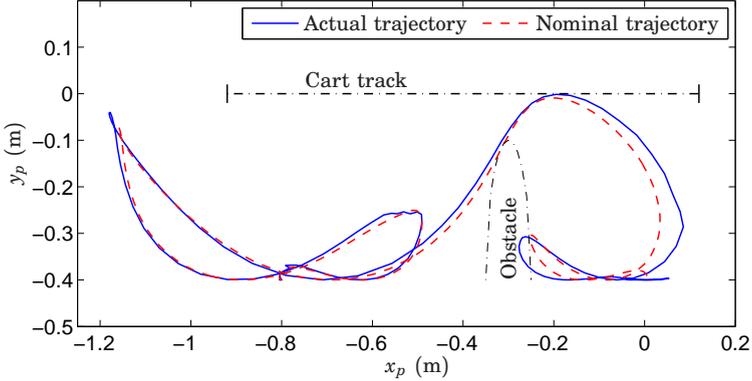


Figure 3. Pendulum end point trajectories for path-constrained problem. Both the optimal trajectory and the trajectory obtained when optimal control trajectory is applied to the physical pendulum system in open loop, are plotted.

We get the following optimization problem

$$\begin{aligned}
 & \text{minimize} && t_f \\
 & \text{subject to} && \ddot{\theta} = -\frac{g}{l} \sin \theta + \frac{u}{l} \cos \theta \\
 & && \ddot{p} = u \\
 & && x_p = p - l \sin \theta && y_p = -l \cos \theta \\
 & && \left(\frac{x_p + 0.3}{0.05} \right)^2 + \left(\frac{y_p + 0.4}{0.3} \right)^2 \geq 1 \\
 & && -0.9 \leq p \leq 0.1 && |u| \leq 5 && |\dot{u}| \leq 100 \\
 & && p(0) = -0.8 && \dot{p}(0) = 0 && \theta(0) = 0 && \dot{\theta}(0) = 0 \\
 & && p(t_f) = 0 && \dot{p}(t_f) = 0 && \theta(t_f) = 0 && \dot{\theta}(t_f) = 0
 \end{aligned} \tag{5}$$

where t_f again is the final time. This is a highly nonconvex problem due to the nonlinear dynamics and, more significantly, due to the obstacle. To solve this problem using the JModelica.org platform, an initial guess needs to be constructed and sent to the solver. An initial guess is created by dividing the optimization problem into two parts. The first part has the same initial condition as the original problem and the terminal point constraint at a position above the obstacle. The second part has the initial condition at the position above the obstacle and the same terminal constraint as the original problem. These trajectories are merged and sent to the solver as initial condition. For more details on how these trajectories were created, see [Giselsson *et al.*, 2009]. Optimization results, as well as the trajectory obtained when applying the control action to the physical

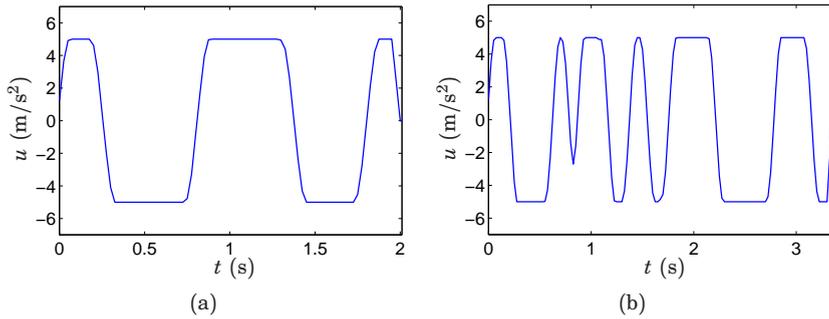


Figure 4. Optimal control trajectories for the swing-up problem (a) and the path-constrained problem (b).

system with the same initial condition as in the optimization, are found in Figure 3. The optimal control trajectory for the path-constrained problem is found in Figure 4(b).

4.3 Discretization of the Optimal Trajectories

The results from optimization problems (4) and (5) are continuous time state and control trajectories which we denote by $p^*(t), \dot{p}^*(t), \theta^*(t)$ and $\dot{\theta}^*(t)$ respectively. The sampling time of the PC communication is denoted by h and we introduce the sampling counter $n \in \mathbb{N}$. This implies that the actual time t at sampling instant n is $t = hn$. We define the discrete time variables $p^0, \dot{p}^0, \theta^0, \dot{\theta}^0$ and u^0 at the sampling instants as

$$\begin{aligned} p^0(n) &:= p^*(nh), & \dot{p}^0(n) &:= \dot{p}^*(nh), \\ \theta^0(n) &:= \theta^*(nh), & \dot{\theta}^0(n) &:= \dot{\theta}^*(nh), \\ u^0(n) &:= u^*(nh), \end{aligned}$$

for every $n \in \mathbb{N}$ such that $nh \leq t_f$. We also define

$$x^0(n) = [p^0(n) \dot{p}^0(n) \theta^0(n) \dot{\theta}^0(n)]^T.$$

Using these definitions, discrete time trajectories are created and used as feedforward trajectories to the pendulum system.

5. Model Predictive Control

The feedforward control trajectories from the previous section gives close to optimal state trajectories when applied to the physical pendulum sys-

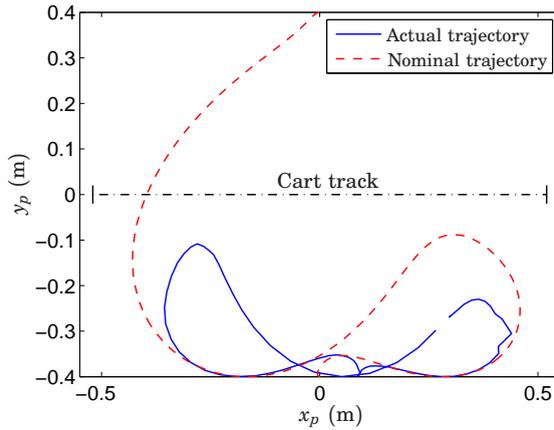


Figure 5. Pendulum end point trajectory for the swing-up problem when the real pendulum is swinging initially and no feedback is used.

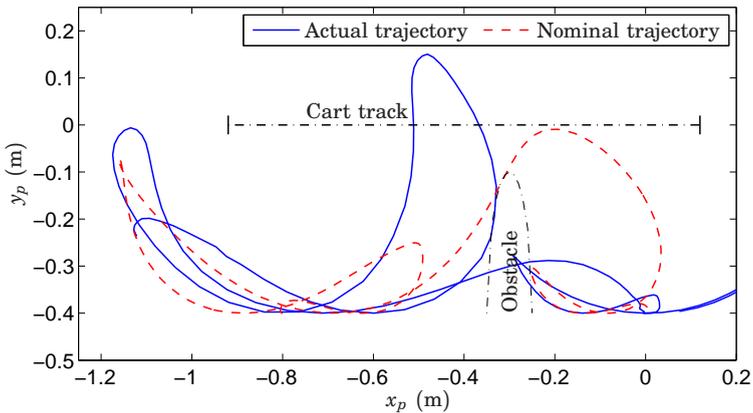


Figure 6. Pendulum end point trajectory for the path-constrained problem when the real pendulum is swinging initially and no feedback is used.

tem, see Figures 2 and 3. This behavior cannot be expected when disturbances are present. In the optimization problems in the previous section it is specified that the pendulum should start at rest in the downward position. When disturbances are present in the initial condition of the pendulum, the resulting pendulum end point trajectories are very far from the optimal ones. This is shown in Figure 5 for the swing-up problem and in Figure 6 for the path constrained problem. In the experiments, the pendulum was initially swinging back and forth with a magnitude of approximately 45° . In this section we introduce MPC feedback to cope with

such disturbances.

5.1 Discrete Time Pendulum Model

The continuous time dynamics of the pendulum system is discretized and linearized to be used for model predictive control. In each sampling instant, n , the system, (2)-(3), is linearized around the nominal states, $x^0(n)$, and the nominal control, $u^0(n)$. We introduce actual states $p(n)$, $\dot{p}(n)$, $\theta(n)$, $\dot{\theta}(n)$ and $x(n) = [p(n) \ \dot{p}(n) \ \theta(n) \ \dot{\theta}(n)]^T$ and the actual control $u(n)$. We also introduce the deviation between the actual states and nominal states

$$\begin{aligned} \Delta p(n) &:= p(n) - p^0(n), & \Delta \dot{p}(n) &:= \dot{p}(n) - \dot{p}^0(n), \\ \Delta \theta(n) &:= \theta(n) - \theta^0(n), & \Delta \dot{\theta}(n) &:= \dot{\theta}(n) - \dot{\theta}^0(n), \\ \Delta x(n) &:= x(n) - x^0(n), & \Delta u(n) &:= u(n) - u^0(n). \end{aligned}$$

Since the cart dynamics are linear, only the pendulum dynamics need to be linearized. To achieve this, we introduce $\Delta z_\theta(t)$ which is the deviation from the linearization point for the continuous pendulum states and $\Delta v(t)$ which is the continuous control signal for the linearized model. This gives the following linearized pendulum dynamics when linearized around pendulum angle θ^0

$$\begin{aligned} \Delta \dot{z}_\theta(t) &= \begin{pmatrix} 0 & 1 \\ -\frac{g}{l} \cos \theta^0 & 0 \end{pmatrix} \Delta z_\theta(t) + \begin{pmatrix} 0 \\ \frac{1}{l} \cos \theta^0 \end{pmatrix} \Delta v(t) \\ &= \underbrace{\begin{pmatrix} 0 & 1 \\ -(\omega^0)^2 & 0 \end{pmatrix}}_{A(\theta^0)} \Delta z_\theta(t) + \underbrace{\begin{pmatrix} 0 \\ \frac{(\omega^0)^2}{g} \end{pmatrix}}_{B(\theta^0)} \Delta v(t) \end{aligned}$$

where $(\omega^0)^2 = \frac{g}{l} \cos \theta^0$. The resulting linear time-varying dynamics depend on the nominal pendulum angle θ^0 only. To obtain a discrete time model for sampling instant n , the linearization is performed around pendulum angle $\theta^0(n)$ and the resulting linear model is discretized using zero-order-hold. The discrete time control signal, which we denote by $\Delta u(n)$ is constant during the sample. The discretized zero-order-hold equations becomes

$$\Delta x_\theta(n+1) = e^{A(\theta^0(n))h} \Delta x_\theta(n) + \int_{s=0}^h e^{A(\theta^0(n))(h-s)} B(\theta^0(n)) ds \Delta u(n) \quad (6)$$

where

$$\begin{aligned}
 e^{A(\theta^0(n))h} &= I + A(\theta^0(n))h + \frac{(A(\theta^0(n))h)^2}{2!} + \frac{(A(\theta^0(n))h)^3}{3!} + \dots \\
 &= I + \begin{pmatrix} 0 & 1 \\ -\omega^0(n)^2 & 0 \end{pmatrix} h - \begin{pmatrix} \omega^0(n)^2 & 0 \\ 0 & \omega^0(n)^2 \end{pmatrix} \frac{h^2}{2!} \\
 &\quad + \begin{pmatrix} 0 & -\omega^0(n)^2 \\ \omega^0(n)^4 & 0 \end{pmatrix} \frac{h^3}{3!} + \begin{pmatrix} \omega^0(n)^4 & 0 \\ 0 & \omega^0(n)^4 \end{pmatrix} \frac{h^4}{4!} \\
 &\quad + \begin{pmatrix} 0 & \omega^0(n)^4 \\ -\omega^0(n)^6 & 0 \end{pmatrix} \frac{h^5}{5!} - \begin{pmatrix} \omega^0(n)^6 & 0 \\ 0 & \omega^0(n)^6 \end{pmatrix} \frac{h^6}{6!} \\
 &\quad + \begin{pmatrix} 0 & -\omega^0(n)^6 \\ \omega^0(n)^8 & 0 \end{pmatrix} \frac{h^7}{7!} + \dots \\
 &= \begin{pmatrix} \sum_{l=0}^{\infty} \frac{(-1)^l}{(2l)!} (\omega^0(n)h)^{2l} & \frac{1}{\omega^0(n)} \sum_{l=0}^{\infty} \frac{(-1)^l}{(2l+1)!} (\omega^0(n)h)^{2l+1} \\ -\omega^0(n) \sum_{l=0}^{\infty} \frac{(-1)^l}{(2l+1)!} (\omega^0(n)h)^{2l+1} & \sum_{l=0}^{\infty} \frac{(-1)^l}{(2l)!} (\omega^0(n)h)^{2l} \end{pmatrix} \\
 &= \begin{pmatrix} \cos \omega^0(n)h & \frac{1}{\omega^0(n)} \sin \omega^0(n)h \\ -\omega^0(n) \sin \omega^0(n)h & \cos \omega^0(n)h \end{pmatrix}
 \end{aligned}$$

with $(\omega^0(n))^2 = \frac{g}{T} \cos \theta^0(n)$. The last equality comes from the Taylor series expansion of cosine and sine. The integral in (6) becomes

$$\begin{aligned}
 \int_{s=0}^h e^{A(\theta^0(n))(h-s)} B(\theta^0(n)) ds &= \int_{s=0}^h \begin{pmatrix} \frac{\omega^0(n)}{g} \sin(\omega^0(n)(h-s)) \\ \frac{\omega^0(n)^2}{g} \cos(\omega^0(n)(h-s)) \end{pmatrix} ds \\
 &= \left[\begin{array}{c} \frac{1}{g} \cos(\omega^0(n)(h-s)) \\ -\frac{\omega^0(n)}{g} \sin(\omega^0(n)(h-s)) \end{array} \right]_{s=0}^h \\
 &= \begin{pmatrix} \frac{1}{g}(1 - \cos \omega^0(n)h) \\ \frac{\omega^0(n)}{g} \sin \omega^0(n)h \end{pmatrix}.
 \end{aligned}$$

A discrete time model of the double integrator (2) is well known to be

$$\Delta x_c(n+1) = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} \Delta x_c(n) + \begin{pmatrix} \frac{h^2}{2} \\ h \end{pmatrix} \Delta u(n)$$

where $\Delta x_c = [\Delta p \ \Delta j]^T$. This gives the following full linearized model

$$\Delta x(n+1) = \Phi(\theta^0(n))\Delta x(n) + \Gamma(\theta^0(n))\Delta u(n) \quad (7)$$

where

$$\Phi(\theta^0(n)) = \begin{pmatrix} 1 & h & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos \omega^0(n)h & \frac{1}{\omega^0(n)} \sin \omega^0(n)h \\ 0 & 0 & -\omega^0(n) \sin \omega^0(n)h & \cos \omega^0(n)h \end{pmatrix}$$

$$\Gamma(\theta^0(n)) = \begin{pmatrix} \frac{h^2}{2} \\ h \\ \frac{1}{g}(1 - \cos \omega^0(n)h) \\ \frac{\omega^0(n)}{g} \sin \omega^0(n)h \end{pmatrix}$$

and $(\omega^0(n))^2 = \frac{g}{L} \cos \theta^0(n)$.

5.2 MPC Optimization Problem

The model developed in the previous section is unstable for some pendulum angles. Due to this, predicting future states directly with (7) may result in poor predictions. To avoid that, a discrete time LQ-feedback term that depends on the nominal pendulum angle is introduced, $u_{fb}(n) = -L(\theta^0(n))\Delta x(n)$, where $L(\theta^0(n))$ is the optimal LQ-feedback for (7). The prediction model becomes

$$\begin{aligned} \Delta x(n+1) &= (\Phi(\theta^0(n)) - \Gamma(\theta^0(n))L(\theta^0(n))) \Delta x(n) + \Gamma(\theta^0(n))\Delta u(n) \\ &= \Phi_L(\theta^0(n))\Delta x(n) + \Gamma(\theta^0(n))\Delta u(n) \end{aligned} \quad (8)$$

where $\Phi_L(\theta^0(n)) := \Phi(\theta^0(n)) - \Gamma(\theta^0(n))L(\theta^0(n))$. This model is stable for every nominal pendulum angle $\theta^0(n)$. The decision variables in the MPC problem are state and control signal deviations from the nominal trajectories. In every sample instant, $u(n) = u^0(n) + \Delta u(n) + u_{fb}(n)$, is sent as control signal (acceleration reference) to the system. The maximal allowed acceleration is $\pm 7m/s^2$ which is the maximal acceleration for which the inner control loops do not saturate. The track on which the cart is attached is one meter. The track length and control magnitude constraints are captured in the following sample dependent constraint set

$$\begin{aligned} \mathcal{X}(n) = \{ \Delta u(n) \in \mathbb{R}, \Delta x(n) \in \mathbb{R}^4 \mid & |\Delta u(n) + u^0(n) - L(\theta^0(n))\Delta x(n)| \leq 7, \\ & \Delta p(n) + p^0(n) \leq 1 - p_0, \\ & \Delta p(n) + p^0(n) \geq -p_0 \} \end{aligned} \quad (9)$$

where $p_0 \in [0, 1]$ is the initial position of the cart on the track. We use a quadratic cost, hence the MPC problem to be solved in each sampling

instant, n , is

$$\begin{aligned}
 \min_{\Delta x, \Delta u} \quad & \sum_{l=n}^{n+N} \Delta x(l)^T Q \Delta x(l) + \Delta u(l)^T R \Delta u(l) & (10) \\
 \text{s.t.} \quad & \Delta x(l+1) = \Phi_L(\theta^0(l)) \Delta x(l) + \Gamma(\theta^0(l)) \Delta u(l), \quad l = n, \dots, n+N-1, \\
 & (x(l), u(l)) \in \mathcal{X}(l), \quad l = n, \dots, n+N, \\
 & \Delta x(n) = \bar{x}
 \end{aligned}$$

where $Q \succeq 0$ and $R \succ 0$. The optimal $\Delta u(n+N) \equiv 0$ and can hence be removed from the optimization. Since the objective function is quadratic and the dynamics and constraints are linear, the resulting optimization problem is a quadratic program.

5.3 Optimization Algorithm

The optimization problem (10) is solved using the algorithm presented in [Giselsson, 2012a]. The algorithm in [Giselsson, 2012a] is an accelerated gradient algorithm that is applied to the dual of a condensed version of (10). A condensed version it obtained by eliminating the state variables by expressing them in the control variables. We present the condensed version of (10) and the optimization algorithm from [Giselsson, 2012a] below. To this end we introduce the following matrices

$$\mathbf{A}(n) := \begin{pmatrix} I \\ \mathbf{A}_1(n) \\ \vdots \\ \mathbf{A}_N(n) \end{pmatrix}, \quad \mathbf{B}(n) := \begin{pmatrix} 0 & \cdots & 0 \\ \mathbf{B}_{11}(n) & \cdots & \mathbf{B}_{1N}(n) \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{N1}(n) & \cdots & \mathbf{B}_{NN}(n) \end{pmatrix}$$

where

$$\begin{aligned}
 \mathbf{A}_i(n) &:= \prod_{l=n}^{n+i-1} \Phi_L(\theta^0(l)), \\
 \mathbf{B}_{ij}(n) &:= \begin{cases} \left[\prod_{l=n+j}^{n+i-1} \Phi_L(\theta^0(l)) \right] \Gamma(\theta^0(n+j-1)) & j < i, \\ \Gamma(\theta^0(n+j-1)) & j = i, \\ 0 & j > i. \end{cases}
 \end{aligned}$$

We denote the predicted future state and control variables by

$$\Delta \mathbf{x}(n) = \begin{bmatrix} \Delta x(n) \\ \vdots \\ \Delta x(n+N) \end{bmatrix}, \quad \Delta \mathbf{u}(n) = \begin{bmatrix} \Delta u(n) \\ \vdots \\ \Delta u(n+N-1) \end{bmatrix}.$$

The predicted future state variables can be expressed in the current state $\Delta x(n) = \bar{x}$ and in the control variables $\Delta \mathbf{u}(n)$ as

$$\Delta \mathbf{x}(n) = \mathbf{A}(n)\bar{x} + \mathbf{B}(n)\Delta \mathbf{u}(n). \quad (11)$$

By introducing the matrices $I_p := [1, 0, 0, 0]$, $C_p := \text{blkdiag}(I_p, \dots, I_p)$ and

$$C_L(n) := \text{blkdiag}(-L(\theta^0(n)), \dots, -L(\theta^0(n+N-1)))$$

the constraint set (9) for all $n, \dots, n+N$ can be written as

$$\underbrace{\begin{pmatrix} C_p \\ -C_p \\ C_L(n) \\ -C_L(n) \end{pmatrix}}_{C_x(n)} \Delta \mathbf{x}(n) + \underbrace{\begin{pmatrix} 0 \\ 0 \\ I \\ -I \end{pmatrix}}_{C_u} \Delta \mathbf{u}(n) \leq \underbrace{\begin{pmatrix} \mathbf{1} - \mathbf{p}_0 - \mathbf{p}^0(n) \\ \mathbf{p}_0 + \mathbf{p}^0(n) \\ \mathbf{7} - \mathbf{u}^0(n) \\ \mathbf{7} + \mathbf{u}^0(n) \end{pmatrix}}_{d(n)} \quad (12)$$

where

$$\begin{aligned} \mathbf{p}_0 &= [p_0, \dots, p_0]^T, & \mathbf{p}^0(n) &= [p^0(n), \dots, p^0(n+N)]^T, \\ \mathbf{1} &= [1, \dots, 1]^T, & \mathbf{u}^0(n) &= [u^0(n), \dots, u^0(n+N-1)]^T, \\ \mathbf{7} &= [7, \dots, 7]^T. \end{aligned}$$

The constraints in (12) can, using the state predictions in (11), be written as

$$(C_x(n)\mathbf{B}(n) + C_u) \Delta \mathbf{u}(n) \leq d(n) - C_x(n)\mathbf{A}(n)\bar{x}. \quad (13)$$

By further defining

$$\begin{aligned} \mathbf{Q} &:= \text{blkdiag}(Q, \dots, Q), & \mathbf{R} &:= \text{blkdiag}(R, \dots, R), \\ C(n) &:= C_x(n)\mathbf{B}(n) + C_u, & g(n, \bar{x}) &:= d(n) - C_x(n)\mathbf{A}(n)\bar{x} \end{aligned}$$

the optimization problem (10) at sample instant n is equivalently written as

$$\begin{aligned} \min_{\Delta \mathbf{u}(n)} & \quad \frac{1}{2} \Delta \mathbf{u}(n)^T H(n) \Delta \mathbf{u}(n) + \bar{x}^T G(n) \Delta \mathbf{u}(n) \\ \text{s.t.} & \quad C(n) \Delta \mathbf{u}(n) \leq g(n, \bar{x}) \end{aligned} \quad (14)$$

where $H(n) = \mathbf{B}(n)^T \mathbf{Q} \mathbf{B}(n) + \mathbf{R}$ and $G(n) = \mathbf{A}(n)^T \mathbf{Q} \mathbf{B}(n)$. To solve (14) we introduce dual variables $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^p$ for the inequality constraints where p is the total number of constraints. We get the following dual problem

$$\max_{\boldsymbol{\mu} \geq 0} \min_{\Delta \mathbf{u}(n)} \frac{1}{2} \Delta \mathbf{u}(n)^T H(n) \Delta \mathbf{u}(n) + \bar{x}^T G(n) \Delta \mathbf{u}(n) + \boldsymbol{\mu}^T (C(n) \Delta \mathbf{u}(n) - g(n, \bar{x}))$$

which, as shown in [Giselsson *et al.*, 2012], can explicitly be written as

$$\max_{\boldsymbol{\mu} \geq 0} -\frac{1}{2} (C(n)^T \boldsymbol{\mu} + G(n)^T \bar{x})^T (H(n))^{-1} (C(n)^T \boldsymbol{\mu} + G(n)^T \bar{x}) - \boldsymbol{\mu}^T g(n, \bar{x}). \quad (15)$$

We define the dual function as the maximand in (15) and denote the dual function by $D_N(\bar{x}, \boldsymbol{\mu}, n)$. The dual function D_N has Lipschitz continuous gradient with Lipschitz constant $L(n) = \|C(n)(H(n))^{-1}C(n)^T\|$ and the gradient is given by

$$\nabla D_N(\bar{x}, \boldsymbol{\mu}, n) = -C(n)(H(n))^{-1}(C(n)^T \boldsymbol{\mu} + G(n)^T \bar{x}) - g(n, \bar{x}).$$

As shown in [Giselsson *et al.*, 2012, Giselsson, 2012a] this implies that the dual function can be maximized using an accelerated gradient method. The iterations defining the accelerated gradient algorithm applied to the dual problem (15) are stated below (cf. [Giselsson, 2012a]).

$$\Delta \mathbf{u}^k = -(H(n))^{-1} (C(n)^T \boldsymbol{\mu}^k + G(n)^T \bar{x}) \quad (16)$$

$$\Delta \tilde{\mathbf{u}}^k = \Delta \mathbf{u}^k + \frac{k-1}{k+2} (\Delta \mathbf{u}^k - \Delta \mathbf{u}^{k-1}) \quad (17)$$

$$\boldsymbol{\mu}^{k+1} = \max \left\{ 0, \boldsymbol{\mu}^k + \frac{k-1}{k+2} (\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}) + \frac{1}{L(n)} (C(n) \Delta \tilde{\mathbf{u}}^k - g(n, \bar{x})) \right\} \quad (18)$$

where k is the iteration number. The algorithm converges as $O(1/k^2)$ in both dual function value and in distance between the primal variables $\Delta \mathbf{u}^k$ and the optimal primal variables (cf. [Giselsson *et al.*, 2012, Giselsson, 2012a]). For more on accelerated gradient methods the reader is referred to [Nesterov, 2003, Beck and Teboulle, 2009, Tseng, 2008, Giselsson *et al.*, 2012].

5.4 Implementational Aspects and Stopping Conditions

The MPC controller is implemented in Matlab/Simulink and communicates with the second microprocessor on the cart via the serial interface. Not all state variables can be measured directly, only cart and pendulum positions are measured. The cart velocity is estimated in the second

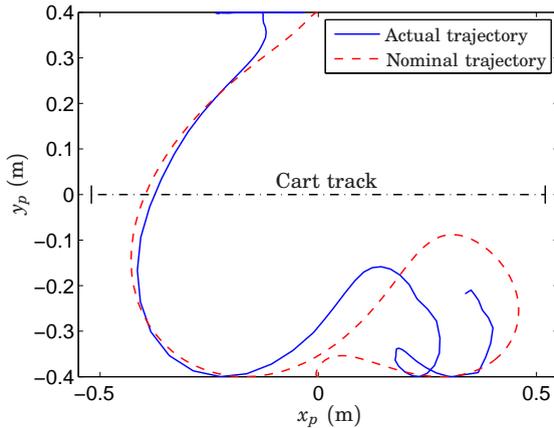


Figure 7. Pendulum end point trajectory for the swing-up problem when the real pendulum is swinging initially and feedback is used.

microprocessor and is sent to the PC when asked for by the MPC controller. The pendulum angular velocity is estimated by a derivative filter in Simulink that is updated ones in every MPC sample. This gives accurate enough pendulum angular velocity estimates since the pendulum dynamics are quite slow. The control horizon is chosen to $N = 40$. The sampling time, which is chosen to $h = 0.025\text{s}$, sets hard limitations on the allowed execution time of the MPC controller. In each sampling instant, the matrices $A(n)$, $B(n)$, $C(n)$ and $g(n, \bar{x})$ are built. These matrices are sampling dependent, but can be precomputed and stored for faster online execution. The optimization algorithm (16)-(18) is warm-started in every sample with the solution to the optimization problem in the previous sample, shifted one step, as initial guess. A constraint tightening approach is used to guarantee a feasible solution with finite number of iterations. We use a relative constraint tightening of 0.005, i.e., if the actual constraint is $x \leq 0.5$ the corresponding constraint is set to $x \leq (1 - 0.005)0.5 = 0.4975$ in the optimization problem. The stopping condition of the algorithm is to have primal feasibility, i.e., $x \leq 0.5$ in the example above and a relative duality gap less than 0.005. By construction of the optimization problem, the equality constraints originating from the dynamic equations always hold.

5.5 Experimental Results

In Figures 7 and 8 pendulum end point trajectories when feedforward and MPC feedback is used, are plotted. The experiments are initialized with the pendulum swinging back and forth with a magnitude of approx-

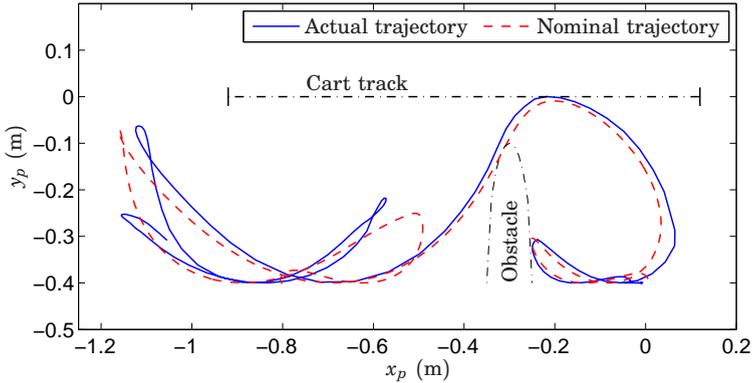


Figure 8. Pendulum end point trajectory for the path-constrained problem when the real pendulum is swinging initially and feedback is used.

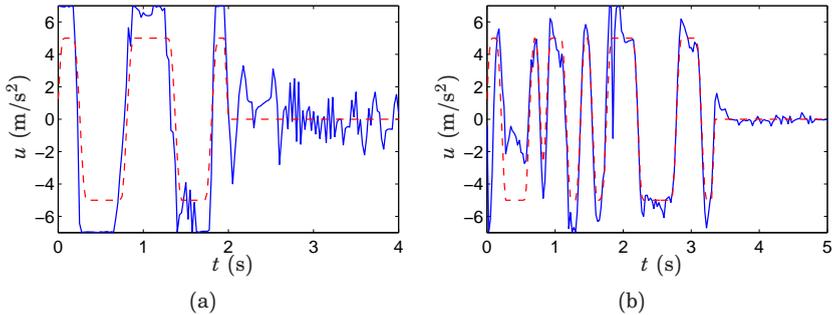


Figure 9. Control trajectory applied to the real system (solid) and feedforward trajectory (dashed) when initial swinging and feedback is used for the swing-up problem (a) and the path-constrained problem (b).

imately 45° as in the examples without feedback in Figures 5 and 6. The weight matrices are chosen to be $Q = \text{diag}(50, 0.1, 50, 0.1)$, and $R = 0.3$ in the path-constrained problem and $R = 1$ in the swing-up problem. The feedback gain vector L is the LQ-gain computed using unit weights on both states and control.

Due to the initial swinging of the pendulum, the trajectories are far from the optimal ones at start but the feedback brings the actual trajectories closer to the optimal trajectories with time. This shows that the introduced model approximations are accurate enough to achieve good performance in the physical pendulum system. Figures 9(a) and 9(b) show the control trajectories that are applied to the system for the swing-

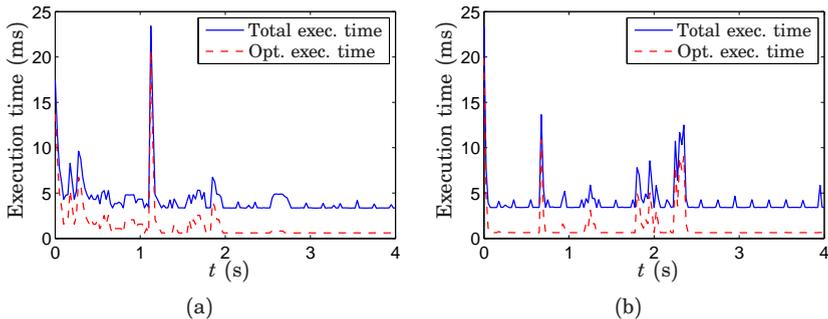


Figure 10. Total (solid) and optimization algorithm (dashed) execution times for the MPC algorithm when initial swinging and feedback is used for the swing-up problem (a) and the path-constrained problem (b).

up problem and the path-constrained problem respectively. Figures 10(a) and 10(b) show the execution time of the MPC controller for the swing-up problem and path-constrained problem respectively. Both total execution time, which include setup of the matrices used by the optimization algorithm and solving the problem, and execution time used by the optimization algorithm are plotted. Figures 10(a) and 10(b) show that the optimization algorithm is efficient enough to find a close to optimal solution well within the sampling time of $h = 25$ ms. Videos of similar experiments, with and without initial swinging of the pendulum, can be found in [Giselsson, 2012b].

6. Conclusions

We have developed an MPC controller that controls the actual system trajectories towards precomputed feedforward trajectories in a pendulum system. The feedforward trajectories take the system from one operating point to another. One swing-up problem and one path-constrained problem are considered and both have been applied to a physical pendulum system. The MPC optimization problem is solved using an accelerated gradient method technique presented in [Giselsson, 2012a]. The experiments show that the algorithm is efficient enough for real-time implementation in a pendulum system.

References

- Åkesson, J., M. Gäfvert, and H. Tummescheit (2009): “Jmodelica—an open source platform for optimization of modelica models.” In *Proceedings of MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*. Vienna, Austria.
- Beck, A. and M. Teboulle (2009): “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” *SIAM J. Imaging Sciences*, **2:1**, pp. 183–202.
- Bemporad, A. and C. Filippi (2001): “Suboptimal explicit MPC via approximate multiparametric quadratic programming.” In *Proceedings of 40th IEEE Conference on Decision and Control*, pp. 4851–4856. Orlando, Florida.
- Bemporad, A., M. Morari, V. Dua, and E. Pistikopoulos (2002): “The explicit linear quadratic regulator for constrained systems.” *Automatica*, **38:1**, pp. 3–20.
- Camacho, E. F. and C. A. Bordons (1997): *Model Predictive Control in the Process Industry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Giselsson, P. (2011): “Model predictive control in a pendulum system.” In *Proceedings of the 31st IASTED Conference on Modelling, Identification and Control*. Innsbruck, Austria.
- Giselsson, P. (2012a): “Execution time certification for gradient-based optimization in model predictive control.” In *Proceedings of the 51st IEEE Conference on Decision and Control*. Maui, HI. To appear.
- Giselsson, P. (2012b): “Pendulum videos.” <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Giselsson, P., J. Åkesson, and A. Robertsson (2009): “Optimization of a pendulum system using Optimica and Modelica.” In *Proceedings of the 7th International Modelica Conference 2009*, pp. 480–489. Como, Italy.
- Giselsson, P., M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer (2012): “Accelerated gradient methods and dual decomposition in distributed model predictive control.” *Automatica*. To appear. Available: <http://www.control.lth.se/Staff/PontusGiselsson.html>.
- Johansen, T. A. (2004): “Approximate explicit receding horizon control of constrained nonlinear systems.” *Automatica*, **40:2**, pp. 293–300.
- Maciejowski, J. M. (2002): *Predictive Control with Constraints*. Prentice Hall, Essex, England.

- Morari, M. and J. H. Lee (1999): “Model predictive control: Past, present and future.” *Computers and Chemical Engineering*, **23**, pp. 667–682.
- Nesterov, Y. (2003): *Introductory Lectures on Convex Optimization: A Basic Course*, 1st edition. Springer Netherlands.
- Rawlings, J. and D. Mayne (2009): *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI.
- Richter, S., C. Jones, and M. Morari (2009): “Real-time input-constrained MPC using fast gradient methods.” In *Proceedings of the 48th Conference on Decision and Control*, pp. 7387–7393. Shanghai, China.
- Tseng, P. (2008): “On accelerated proximal gradient methods for convex-concave optimization.” Submitted to *SIAM J. Optim.* Available: <http://www.math.washington.edu/~tseng/papers/apgm.pdf>.
- Wang, Y. and S. Boyd (2010): “Fast model predictive control using online optimization.” *IEEE Transactions on Control Systems Technology*, **18:2**, pp. 267–278.

Supplement A

Specification of Randomly Generated Systems

In this section, the randomly generated systems used in the numerical sections of Paper III, Paper IV, Paper V, and Paper VI are presented. In Paper III, Paper IV, and Paper V the same random system is used.

A.1. Random System in Paper III, Paper IV, and Paper V

Dynamics

The dynamical system consists of three subsystems with five states and one input in each subsystem. The full system dynamics are described by

$$x(t+1) = Ax(t) + Bu(t)$$

where $x \in \mathbb{R}^{15}$, $u \in \mathbb{R}^3$, $x = [x_1, \dots, x_{15}]^T$, and $u = [u_1, u_2, u_3]^T$. The dynamics-matrix is block-diagonal while the input-matrix is sparse:

$$A = \begin{pmatrix} A_1 & & \\ & A_2 & \\ & & A_3 \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & 0 & B_{13} \\ B_{21} & B_{22} & B_{23} \\ 0 & B_{32} & B_{33} \end{pmatrix}.$$

The matrices for subsystem 1 are:

$$A_1 = \begin{pmatrix} 0.265 & 0.237 & 0.001 & 0.193 & 0.056 \\ 0.054 & 0.426 & 0.154 & 0.209 & 0.079 \\ 0.101 & 0.243 & 0.455 & 0.145 & 0.148 \\ 0.084 & 0.111 & 0.059 & 0.453 & 0.210 \\ 0.058 & 0.238 & 0.236 & 0.083 & 0.326 \end{pmatrix}$$

and

$$B_{11} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.403 \end{pmatrix}, \quad B_{13} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.827 \end{pmatrix}.$$

Supplement A. Specification of Randomly Generated Systems

The matrices for subsystem 2 are:

$$A_2 = \begin{pmatrix} 0.455 & 0.109 & 0.098 & 0.070 & 0.127 \\ 0.027 & 0.398 & 0.109 & 0.157 & 0.132 \\ 0.208 & 0.177 & 0.495 & 0.149 & 0.023 \\ 0.213 & 0.188 & 0.145 & 0.496 & 0.229 \\ 0.090 & 0.192 & 0.215 & 0.022 & 0.476 \end{pmatrix}$$

and

$$B_{21} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.476 \end{pmatrix}, \quad B_{22} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.439 \end{pmatrix}, \quad B_{23} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.620 \end{pmatrix}.$$

The matrices for subsystem 3 are:

$$A_3 = \begin{pmatrix} 0.455 & 0.095 & 0.102 & 0.182 & 0.100 \\ 0.250 & 0.499 & 0.160 & 0.122 & 0.251 \\ 0.040 & 0.246 & 0.502 & 0.162 & 0.102 \\ 0.060 & 0.163 & 0.141 & 0.477 & 0.167 \\ 0.178 & 0.217 & 0.236 & 0.050 & 0.481 \end{pmatrix}$$

and

$$B_{32} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.304 \end{pmatrix}, \quad B_{33} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.995 \end{pmatrix}.$$

Constraints

The states are subject to the following constraints:

$$\begin{aligned} -0.116 \leq x_1 \leq 1.246, & \quad -0.076 \leq x_2 \leq 1.023, & \quad -0.104 \leq x_3 \leq 1.462, \\ -0.120 \leq x_4 \leq 0.530, & \quad -0.056 \leq x_5 \leq 1.020, & \quad -0.083 \leq x_6 \leq 1.390, \\ -0.061 \leq x_7 \leq 0.730, & \quad -0.073 \leq x_8 \leq 0.811, & \quad -0.057 \leq x_9 \leq 1.152, \\ -0.078 \leq x_{10} \leq 0.775, & \quad -0.094 \leq x_{11} \leq 1.380, & \quad -0.110 \leq x_{12} \leq 1.256, \\ -0.061 \leq x_{13} \leq 1.283, & \quad -0.135 \leq x_{14} \leq 1.479, & \quad -0.097 \leq x_{15} \leq 0.551, \end{aligned}$$

and the controls are subject to the following constraints:

$$-0.608 \leq u_1 \leq 1.153, \quad -1.118 \leq u_2 \leq 0.536, \quad -1.462 \leq u_3 \leq 1.067.$$

Cost Function

Two different quadratic cost functions are used in Paper III, Paper IV, and Paper V; one with identity-matrices defining the cost for states and controls and one with randomly generated diagonal matrices. The randomly generated matrices are

$$Q = \begin{pmatrix} Q_1 & & \\ & Q_2 & \\ & & Q_3 \end{pmatrix},$$

where

$$Q_1 = \begin{pmatrix} 32.11 & & & & \\ & 7.53 & & & \\ & & 34.59 & & \\ & & & 24.88 & \\ & & & & 18.72 \end{pmatrix},$$

$$Q_2 = \begin{pmatrix} 59.63 & & & & \\ & 63.64 & & & \\ & & 8.19 & & \\ & & & 69.01 & \\ & & & & 36.26 \end{pmatrix},$$

$$Q_3 = \begin{pmatrix} 2.71 & & & & \\ & 10.90 & & & \\ & & 88.09 & & \\ & & & 37.31 & \\ & & & & 1.45 \end{pmatrix},$$

and

$$R = \begin{pmatrix} 53.66 & & \\ & 51.05 & \\ & & 83.70 \end{pmatrix}.$$

A.2. Random System in Paper VI

Dynamics

The dynamical system consists of six subsystems with five states, one input, and one output in each subsystem. The full system dynamics are

Supplement A. Specification of Randomly Generated Systems

described by

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}$$

where $x \in \mathbb{R}^{30}$, $u \in \mathbb{R}^6$, $y \in \mathbb{R}^6$ $x = [x_1, \dots, x_{30}]^T$, $u = [u_1, \dots, u_6]^T$, and $y = [y_1, \dots, y_6]^T$. The dynamics-matrix and the output-matrix are block-diagonal:

$$A = \begin{pmatrix} A_1 & & \\ & \ddots & \\ & & A_6 \end{pmatrix}, \quad C = \begin{pmatrix} C_1 & & \\ & \ddots & \\ & & C_6 \end{pmatrix},$$

while the input-matrix is sparse:

$$B = \begin{pmatrix} B_{11} & 0 & B_{13} & B_{14} & B_{15} & 0 \\ 0 & B_{22} & B_{23} & B_{24} & 0 & 0 \\ 0 & 0 & B_{33} & 0 & B_{35} & 0 \\ B_{41} & 0 & B_{43} & B_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & B_{55} & 0 \\ 0 & B_{62} & 0 & 0 & 0 & B_{66} \end{pmatrix}.$$

The output-matrices satisfy $C_1, \dots, C_6 = [0 \ 0 \ 0 \ 0 \ 1]$ and the matrices for subsystem 1 are:

$$A_1 = \begin{pmatrix} 0.299 & 0.144 & 0.015 & 0.236 & 0.248 \\ 0.088 & 0.449 & 0.123 & 0.011 & 0.226 \\ 0.175 & 0.137 & 0.449 & 0.101 & 0.028 \\ 0.086 & 0.015 & 0.054 & 0.322 & 0.140 \\ 0.045 & 0.242 & 0.111 & 0.197 & 0.422 \end{pmatrix}$$

and

$$B_{11} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.169 \end{pmatrix}, \quad B_{13} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.740 \end{pmatrix},$$

$$B_{14} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.124 \end{pmatrix}, \quad B_{15} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.077 \end{pmatrix}.$$

The matrices for subsystem 2 are:

$$A_2 = \begin{pmatrix} 0.493 & 0.094 & 0.171 & 0.096 & 0.027 \\ 0.108 & 0.396 & 0.120 & 0.118 & 0.207 \\ 0.098 & 0.241 & 0.382 & 0.240 & 0.139 \\ 0.187 & 0.168 & 0.080 & 0.286 & 0.105 \\ 0.091 & 0.162 & 0.032 & 0.040 & 0.342 \end{pmatrix}$$

and

$$B_{22} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.748 \end{pmatrix}, \quad B_{23} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.127 \end{pmatrix}, \quad B_{24} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.712 \end{pmatrix}.$$

The matrices for subsystem 3 are:

$$A_3 = \begin{pmatrix} 0.382 & 0.069 & 0.240 & 0.146 & 0.071 \\ 0.228 & 0.416 & 0.167 & 0.100 & 0.230 \\ 0.040 & 0.162 & 0.275 & 0.155 & 0.246 \\ 0.058 & 0.123 & 0.119 & 0.311 & 0.142 \\ 0.244 & 0.155 & 0.070 & 0.250 & 0.489 \end{pmatrix}$$

Supplement A. Specification of Randomly Generated Systems

and

$$B_{33} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.135 \end{pmatrix}, \quad B_{35} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.586 \end{pmatrix}.$$

The matrices for subsystem 4 are:

$$A_4 = \begin{pmatrix} 0.356 & 0.147 & 0.015 & 0.160 & 0.190 \\ 0.076 & 0.463 & 0.008 & 0.223 & 0.042 \\ 0.034 & 0.072 & 0.326 & 0.063 & 0.177 \\ 0.115 & 0.033 & 0.158 & 0.306 & 0.140 \\ 0.252 & 0.139 & 0.173 & 0.118 & 0.318 \end{pmatrix}$$

and

$$B_{41} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.907 \end{pmatrix}, \quad B_{43} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.066 \end{pmatrix}, \quad B_{44} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.292 \end{pmatrix}.$$

The matrices for subsystem 5 are:

$$A_5 = \begin{pmatrix} 0.448 & 0.072 & 0.029 & 0.130 & 0.110 \\ 0.210 & 0.261 & 0.138 & 0.242 & 0.120 \\ 0.238 & 0.014 & 0.438 & 0.212 & 0.229 \\ 0.125 & 0.093 & 0.160 & 0.374 & 0.174 \\ 0.023 & 0.003 & 0.054 & 0.172 & 0.384 \end{pmatrix}, \quad B_{55} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.035 \end{pmatrix}.$$

The matrices for subsystem 6 are:

$$A_6 = \begin{pmatrix} 0.491 & 0.074 & 0.200 & 0.214 & 0.222 \\ 0.088 & 0.341 & 0.194 & 0.058 & 0.254 \\ 0.107 & 0.069 & 0.277 & 0.010 & 0.109 \\ 0.138 & 0.198 & 0.037 & 0.443 & 0.062 \\ 0.175 & 0.071 & 0.069 & 0.056 & 0.272 \end{pmatrix}$$

and

$$B_{62} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.374 \end{pmatrix}, \quad B_{66} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.432 \end{pmatrix}.$$

Constraints and Cost Function

The original state constraints are $-0.11 \leq x_1, \dots, x_{30} \leq 2$. After constraint tightening, which is performed to guarantee robust feasibility in the output feedback case, the state constraints are:

$$\begin{aligned} -0.047 \leq x_1 \leq 1.937, & \quad -0.045 \leq x_2 \leq 1.935, & \quad -0.049 \leq x_3 \leq 1.939, \\ -0.069 \leq x_4 \leq 1.959, & \quad -0.041 \leq x_5 \leq 1.930, & \quad -0.030 \leq x_6 \leq 1.920, \\ -0.031 \leq x_7 \leq 1.921, & \quad -0.015 \leq x_8 \leq 1.904, & \quad -0.040 \leq x_9 \leq 1.930, \\ -0.057 \leq x_{10} \leq 1.947, & \quad -0.037 \leq x_{11} \leq 1.927, & \quad -0.009 \leq x_{12} \leq 1.899, \\ -0.032 \leq x_{13} \leq 1.922, & \quad -0.046 \leq x_{14} \leq 1.936, & \quad -0.005 \leq x_{15} \leq 1.895, \\ -0.057 \leq x_{16} \leq 1.946, & \quad -0.062 \leq x_{17} \leq 1.952, & \quad -0.069 \leq x_{18} \leq 1.959, \\ -0.065 \leq x_{19} \leq 1.955, & \quad -0.051 \leq x_{20} \leq 1.941, & \quad -0.054 \leq x_{21} \leq 1.945, \\ -0.036 \leq x_{22} \leq 1.926, & \quad -0.029 \leq x_{23} \leq 1.915, & \quad -0.038 \leq x_{24} \leq 1.930, \\ -0.065 \leq x_{25} \leq 1.955, & \quad -0.018 \leq x_{26} \leq 1.913, & \quad -0.047 \leq x_{27} \leq 1.937, \\ -0.069 \leq x_{28} \leq 1.959, & \quad -0.041 \leq x_{29} \leq 1.931, & \quad -0.060 \leq x_{30} \leq 1.950. \end{aligned}$$

The control constraints are:

$$\begin{aligned} -0.1 \leq u_1 \leq 0.1, & \quad -0.1 \leq u_2 \leq 0.1, & \quad -0.1 \leq u_3 \leq 0.1, \\ -0.1 \leq u_4 \leq 0.1, & \quad -0.1 \leq u_5 \leq 0.1, & \quad -0.1 \leq u_6 \leq 0.1. \end{aligned}$$

The quadratic cost function has identity cost-matrices for both states and controls.