PLANTWIDE COOPERATIVE DISTRIBUTED MODEL PREDICTIVE CONTROL

by

Brett Taylor Stewart

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Chemical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2010

To my wife Heather.

ACKNOWLEDGMENTS

In preparing this thesis, I have had the pleasure and honor to work with James Rawlings. He has the uncanny ability to immediately understand complex problems and find the path toward realistic solutions. Only by an unrelenting patience has he shaped me into a researcher. His suggestions, talk deadlines, and paper requests always seemed to come at the right time. I was also fortunate to have the constant help of Stephen Wright. He was always willing to devote time to my optimization problems. Because of his effort, the proofs in this thesis are immeasurably better. I also thank Christos Maravelias, Jennifer Reed, and Ross Swaney for serving on my thesis committee.

While in Madison, I had the opportunity to work with some wonderful people. Aswin Venkat got me started during the difficult first year. Ethan Mastny, Paul Larsen, and Murali Rajamani were great sources of encouragement while I learned the ropes. Cliff Amundsen was a true friend who was always willing to listen to whatever had entered my head, whether it be in music, science, or random factoids. I will look fondly on the lunch trips to Jordon's with Aaron Lowe and the gang, and our conversations. Fernando Lima was a real mentor during his time in the group. He can continually be relied upon for advice and encouragement. Rishi Amrit has saved me too many times to mention. I knew who to bug when my code gave me odd error messages or was not working quite right. And he could always share in my excitement at a new working proof or algorithm, even if I only later discovered that sometimes 'working' is a relative term. I enjoyed my time with Kaushik Subramanian. Our conversations straightened out my research many times. And I thank Rishi Srivastiva for enthusiastically helping me with my questions in probability. It has been a real pleasure getting to know Ankur Gupta. Our occasional excursions into philosophy were fun and insightful. I wish the best of luck to Luo Ji. Even though the

time was short, Antonio Ferramosca and Emil Sokoler have been great friends. And of course I must thank John Eaton, who saved me countless hours of frustration. He always had the right solution for any problems with LaTeX, Octave, Gnuplot, and Linux.

I thank Kiran Sheth, Tyler Soderstrom, and Don Bartusiak at Exxon Mobil for giving me the opportunity to learn about real plants and participate in solving industrial problems. My experiences in Baytown have been a constant reality check for me. I must also thank Prof. Bart De Schutter and Prof. Tamas Keviczky for inviting me to Delft. The hospitality of the group at the Delft Center for Systems and Control made my stay both a great learning opportunity and a source of fond memories. Dang Doan deserves extra thanks for being such a kind host during my stay. A special thanks to Moritz Diehl and Wolfgang Marquardt for allowing me to travel and present to their groups.

I am also thankful to all of the friends I have made during my time in Madison. Thanks to Angela Jones for encouraging the dancing lessons and Sean Delcambre for putting up with me. Joe Bernacki, Matt Colvin, and I shared some truly penetrating conversations. I hope we can keep them up. Playing in the Madison Brass Band and the Perfect Fifth Brass Quintet was some of the most fun I have been able to have. I am truly grateful for all of the rehearsals which gave me a break from my studies and a chance to make some music. I especially thank John Stevens for taking a chance on me as an amateur musician. Then there are those friends who have not only helped me grow personally, but spiritually as well. Mark and Audrey Klingele taught me the true meaning of friendship and have kept me on the right path more often than I know. I am so happy to have known both of them and Monica. Marc and Julie Laudonio have been a constant support. They have allowed me to develop with patience and guidance. And I am grateful to Msgr. Kevin Holmes, who is not only a wealth of knowledge, but a true mentor in faith. I thank him for all his blessings. I will keep the Cathedral Parish in my prayers.

None of this would be possible without my parents, who taught and encouraged me early on to pursue my goals. I cannot imagine my life without all the trips to rehearsals, the pressure to complete my homework, and the love to support me through my troubling times. There are too many moments for me to list, but I thank them for them all.

Finally, I must express my gratitude for my wife, Heather. She has offered me indefatigable support, her love is infinite, and she is a constant joy in my life. She has truly made me a better man than I ever wanted to be. This thesis would not have been possible without her.

BTS - Madison, WI

DISCARD THIS PAGE

TABLE OF CONTENTS

Page

Appendix

Page

DISCARD THIS PAGE

LIST OF TABLES

DISCARD THIS PAGE

LIST OF FIGURES

Figure Page

DISCARD THIS PAGE

NOMENCLATURE

A, B, C system matrices $x^+ = Ax + Bu,\ y = Cx$

A_{ij} state transition matrix of subsystem j to subsystem i

A_i state transition matrix of subsystem i

A_{Li} estimate error transition matrix $A_i - L_i C_i$

B_{ij} input matrix of subsystem j to subsystem i

B_i input matrix of subsystem i

$\mathbb{B}_r$ ball of radius r around the origin

C_{ij} output matrix of subsystem j to subsystem i

C_i output matrix of subsystem i

e_i estimation error of subsystem i

f_i model of subsystem i, $x^+ = f_i(x, u)$

g input sequence transition function, $\mathbf{u}^+ = g(x, \mathbf{u})$

$\mathbf{h}_i$ initialization function of subsystem i

$\mathbf{h}_i^+$ reinitialization function of subsystem i

$\mathbb{I}_{l:m}$ the integer interval $\{l, l+1, \ldots, m-1, m\}$

$\mathbb{I}_{\geq 0}$ the positive integers

k sampling time

$\ell_i(x_i, u_i)$ stage cost of subsystem i

L_i optimal estimator gain

m_i input dimension of subsystem i

M number of subsystems

n_i state dimension of subsystem i

N finite control horizon

p iterate of optimization algorithm

p_i output dimension of subsystem i

P_f terminal penalty matrix

Q_i state penalty matrix of subsystem i

R_i input penalty matrix of subsystem i

$\mathbb{R}$ real numbers

$\mathbb{R}_+$ positive real numbers

$\mathbb{R}^n$ real-valued n-vectors

$\mathbb{R}^{n\times m}$ real-valued $n \times m$ matrices

u_i input (manipulated variable) vector of subsystem i

$\mathbf{u}_i$ input trajectory of subsystem i

$\tilde{\mathbf{u}}_i^+$ warm start of subsystem i

$\mathbf{u}^+$ improved input sequence

$\mathbb{U}$ input constraint set of subsystem i

V objective function

V_f terminal penalty

w_i weighting of subsystem i in optimization

x_i state vector of subsystem i

x_{ij} state interaction vector of subsystem j to subsystem i

$\mathbb{X}_f$ terminal set

$\mathbb{X}_N$ stabilizable set

y_i output vector of subsystem i

$|\cdot|$ Euclidian 2-norm (unless otherwise noted)

Greek Letters

α step length in optimization

Δ sampling time

κ control law

κ_f control law in terminal region $\mathbb{X}_f$

ρ_i objective function weighting of subsystem i

Σ_i solution to Lyapunov equation for subsystem i

$\phi(k;x,u)$ state at time k given initial state x and input sequence $\mathbf{u}$

$\phi(k;x)$ closed-loop state at time k given initial state x

$\boldsymbol{v}$ control optimization decision variable

Subscripts, Superscripts, and Accents

$\hat{x}$ estimated state

$\hat{x}^-$ estimated state before measurement

x^+ state at next time step

x_i state of subsystem i

$\mathbf{u}^0$ optimal input trajectory

ABSTRACT

This thesis presents a plantwide model predictive control strategy in which subsystems provide feedback by solving a local optimization and exchanging information across the plant. The plant model is required to satisfy a stabilizability condition that does not depend upon the strength of the open-loop, inter-subsystem interactions. This control strategy has the following features: hard input constraints are satisfied; terminating the distributed optimization iteration prior to convergence does not affect nominal stability; the distributed optimization converges to the Pareto optimal (centralized) solution; no coordinating optimization is employed. Exponential stability of the closed-loop plantwide system is proven for the state and output feedback cases. A modification is presented in which constraints sparsely coupled between subsystems can be handled without loss of stability or optimality.

An extension to the cooperative controller is provided in which communication between subsystems occurs at multiple time schedules, as in traditional hierarchical control. The hierarchical control scheme is shown to provide exponentially stable performance under state and output feedback. This extension uses a modification of the distributed optimization. The optimization is shown to converge to the Pareto optimum and can be terminated early without affecting nominal stability.

To demonstrate the flexibility of cooperative control, a series of examples are presented. These examples show that cooperative control can nominally stabilize subsystems optimizing, communicating, and sampling on any time scale. Further examples are provided to illustrate that

the modeling required for cooperative control can be identified over time and that the models can be generated from a centralized first-principles model.

Finally, a novel distributed optimization for nonconvex problems is presented that provides convergence to stationary points. This optimization is combined with a new result in suboptimal MPC to develop a plantwide distributed nonlinear controller. This controller provides asymptotically stablizing feedback and an example is used to illustrate performance.

Chapter 1

Introduction

Model predictive control (MPC) has become a popular control strategy in the petrochemical industry. Large chemical plants and refineries benefit from several properties of MPC: its ability to account for hard process constraints, the ease of tuning MPC for processes with many variables, and the robustness of MPC to modeling error. Moreover, MPC is adaptable to a variety of systems and operating objectives. Multiple vendors and industrial plants have successfully designed and implemented MPC (Qin and Badgwell, 2003; Young, Bartusiak, and Fontaine, 2001).

In an MPC strategy, inputs are chosen by solving an optimization for each time instant feedback is requested by the plant. This optimization is usually solved online, accounting for the current conditions of the plant, any process or safety constraints, and disturbances affecting the system. Using a model, a forecast of output dynamics is constructed for each feasible input sequence, and the optimization chooses the input sequence that best achieves a design

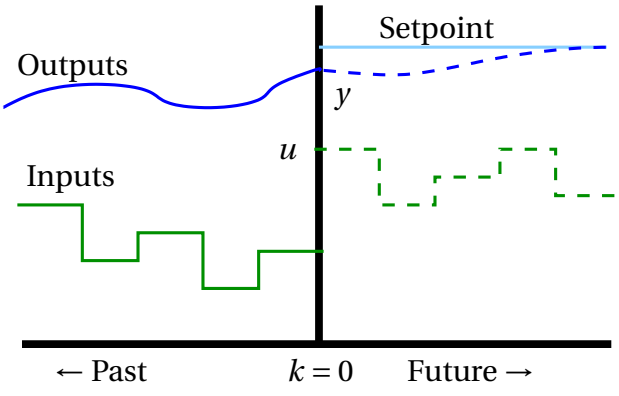


Figure 1.1: The optimization horizon in MPC.

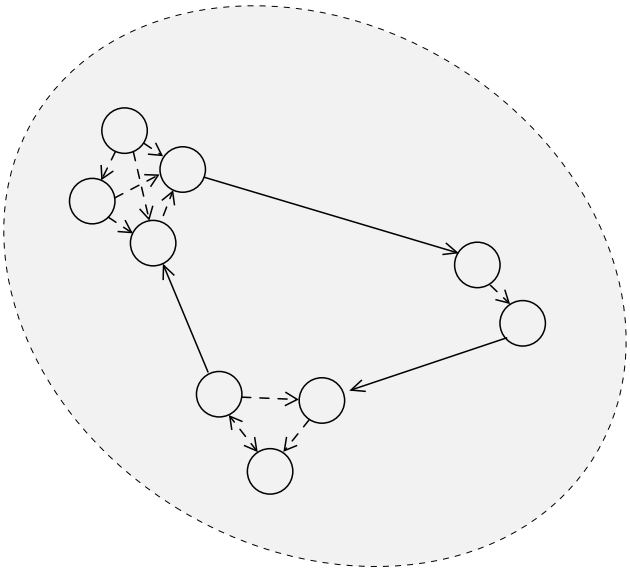


Figure 1.2: Interacting subsystems in a plant.

goal (see Figure 1.1). The first move of the chosen input sequence is injected into the plant and measurements are received from sensors. These measurements are used to obtain better parameters for the model and a more accurate estimate of current state of the plant. A new optimization is formed and the process is repeated (Mayne, Rawlings, Rao, and Scokaert, 2000).

Distributed control

An industrial plant is usually composed of a number of interacting subsystems (see Figure 1.2). An example is an oil refinery, in which a catalytic cracker is used to break crude oil into lighter hydrocarbons which are separated in a network of distillation columns. Often some heavier products are recycled back into the crackers to be broken into lighter products. Hence, the cracker and distillation columns form two interconnected subsystems within the refinery.

In distributed control, the subsystems are controlled independently and the plantwide open-loop interactions are modeled explicitly. Subsystem data, usually open-loop input trajectories, are exchanged between the controllers and a distributed optimization iterate is taken. This information allows each controller to anticipate the future behavior of the plant and take appropriate action to maintain stability and improve performance (Scattolini, 2009).

Other plantwide control strategies are decentralized and centralized control. In a decentralized strategy, no information is exchanged between the subsystems (Sandell Jr., Varaiya, Athans, and Safonov, 1978). Although decentralized control does not have this communication requirement, it is well known that strongly interacting subsystems are difficult to stabilize with decentralized control. Alternatively in centralized control, the plantwide feedback decisions are made by a single controller. Increased computational power, faster optimization software, and algorithms designed specifically for large-scale plantwide control have made centralized control more viable (Bartlett, Biegler, Backstrom, and Gopal, 2002; Pannocchia, Rawlings, and Wright, 2007). Centralized controllers are difficult to implement, however. Organizing subsystems into a centralized control optimization is often impractical or impossible for large-scale plants, and during maintenance of the controller the entire plantwide MPC must be offline. Plantwide coordinating optimizations have also been proposed to increase plantwide performance (Mesarović, Macko, and Takahara, 1970; Scattolini, 2009). Coordinators suffer from the same disadvantages as centralized control, however, because the plantwide control depends upon the reliability of the central optimization layer.

Distributed control offers a middle ground between decentralized and centralized control: the subsystems are independent yet practical performance properties exist. Distributed control can be split into noncooperative and cooperative strategies. In noncooperative control, the exchanged information is used to predict local performance, and each subsystem chooses feedback that improves a local objective. Nominal stability can be shown only for weakly interacting subsystems (Rawlings and Stewart, 2007). Alternatively, for cooperative control each of the subsystem controllers seek to improve a shared plantwide objective. Nominal stability holds for even strongly interacting subsystems and for any finite number of optimization iterates. Under sufficient optimization iteration, the controllers converge to the Pareto optimal, or centralized, plantwide feedback. The extra requirement for a cooperative control implementation is that the subsystem models and objective functions are shared within the plant as they are updated.

Academic impact of cooperative control

Although cooperative control is motivated above in the context of distributed control, it may also be viewed as a class of suboptimal control. A suboptimal controller requires its online optimization to satisfy two criteria: candidate solutions are feasible and the objective function improves at each optimization iterate. Any optimization of this class, coupled with suboptimal control theory, is able to be terminated before convergence and provide stabilizing feedback. If this optimization is *distributed* then the suboptimal controller is a cooperative distributed controller. It is this combination of distributed optimization theory with suboptimal control theory that gives cooperative control its properties.

In this thesis we show the flexibility of this approach to distributed control. In general, the stability of cooperative distributed MPC is shown as a specific case of suboptimal MPC. We provide distributed optimizations, both in the linear and nonlinear case, that satisfy the criteria for a suboptimal controller.

Industrial impact of cooperative control

In a traditional multivariable PID control framework, it is necessary to find input and output pairings that best control the system. Tools such as RGA analysis aid the practitioner to select these pairings, but for strongly interacting systems, it is a nontrivial task to find the best input and output pairs (Larsson and Skogestad, 2000). It is not necessary to pair inputs and outputs in an MPC, however, because the optimization accounts for all the interactions simultaneously and selects stabilizing inputs for each time step. Tuning an MPC is more natural because the pairing question is unnecessary and nominal stability is guaranteed for a wide choice of tuning parameters.

Similarly in plantwide MPC, decentralized control theory guides the practitioner to select plant subsystem decompositions based on the open-loop interaction strength of the subsystems and offers tuning strategies for reducing the closed-loop effects of subsystem interactions. In cooperative MPC, the choice of plant decomposition is widened because the magnitude of the open-loop interactions does not affect stability. Tuning in cooperative MPC is more natural

because nominal closed-loop stability is guaranteed without needing to detune the subsystem controllers. Cooperative control has the potential to be a successful alternative to traditional decentralized control.

Overview of Thesis

Chapter 2 – Literature Review: In the following chapter, we summarize the foundational literature used in this thesis and give an overview of recent developments in distributed MPC.

Chapter 3 – Cooperative Control for Linear Systems: This chapter provides the fundamental theory of linear cooperative MPC. Nominal stability is established for both the state and output feedback cases. An extension to the method is given for plants with sparsely coupled constraints.

Chapter 4 – Hierarchical Cooperative Control: Drawing upon the framework of Chapter 3, an extension to linear cooperative MPC is given in which the subsystems are divided into neighborhoods. Communication is delayed between the neighborhoods, achieving properties similar to hierarchical control. Stability is shown for both the state and output feedback cases.

Chapter 5 – Implementing Plantwide Cooperative Control: We show the flexibility of cooperative MPC as a plantwide control method. Several examples are given that show time scale separation in the sampling, optimization, and communication of the subsystem controllers. We also show how to move to cooperative control from a pre-existing decentralized and centralized control strategy.

Chapter 6 – Cooperative Control for Nonlinear Systems: In this chapter, we relax the linear modeling assumption. The nonlinearity of the models leads to a nonconvex objective function, and therefore we give a novel nonconvex distributed optimization. Stability is established for the state feedback case.

Chapter 7 – Conclusions and Future Work: We end with a summary of the contributions of this thesis and give recommendations for further work in cooperative control.

Chapter 2

Literature Review

2.1 Previous Work

The controllers proposed in this thesis are model predictive controllers. The theory of centralized model predictive control (MPC) has reached a mature stage. Several texts on MPC are available (Maciejowski, 2002; Camacho and Bordons, 2004; Rossiter, 2004; Wang, 2009). In particular, this thesis makes use of the monograph by Rawlings and Mayne (2009). Rawlings and Mayne (2009, pp.409–482) present an extensive study of distributed control, including a discussion of the differences between noncooperative and cooperative control. Additionally, Rawlings and Mayne (2009, pp.89–186) investigate the MPC regulation problem for the general nonlinear case. This study provides key control principles, such as terminal control and suboptimal MPC, used in this thesis.

In addition to control theory, cooperative MPC relies upon distributed optimization theory. An encyclopedic study of distributed and parallel optimization algorithms is presented by Bertsekas and Tsitsiklis (1997). Convergence results are provided for convex Jacobi distributed optimization algorithms (Bertsekas and Tsitsiklis, 1997, pp.219–223), the class of algorithms we propose for linear cooperative control. For nonlinear cooperative control, we utilize gradient projection algorithms, which converge to stationary points of the nonconvex objective function. This optimization method is investigated by Nocedal and Wright (2006, pp.485–490) and Bertsekas (2008, pp.228–249).

The first comprehensive analysis of cooperative MPC was given by Venkat (2006). A number of results are provided, including exponential closed-loop stability for state and output feedback and an extension for asynchronous feedback and optimization. The particular applications of interest are power distribution networks and chemical processes, which are illustrated in several examples.

This thesis uses the results of Venkat as a foundation, but we seek to clarify the presentation and notation of cooperative control while expanding and generalizing the results. In Chapter 3, we extend the stability of cooperative control to a broader class of plants. Venkat assumes that the inter-subsystem open-loop dynamics are stable. We weaken this requirement to a stabilizability assumption. Moreover, Venkat does not show optimal convergence of the distributed optimization with coupled constraints in the control problem. We provide an extension to cooperative control for sparsely coupled constraints. Chapter 4 builds upon the asynchronous feedback scheme of Venkat and presents a general theory of cooperative control with asynchronous communication and reduced plantwide data exchange. For all of the above extensions, we maintain the exponential stability result.

2.2 Review of Distributed Control Literature

Scattolini (2009) has published a review of large scale, plantwide MPC strategies. This paper contains over 100 references and examines decentralized, distributed, and hierarchical control architectures. The distributed control methods are separated into noniterating and iterating categories. The papers (Venkat, Rawlings, and Wright, 2005, 2006) are offered as examples of cooperative control. In the review these papers are classed as iterative methods because negotiation is required between the subsystem controllers. It is shown in Chapter 3, however, that cooperative control is both iterative *and* noniterative. Although the subsystems negotiate to find the plantwide feedback, the negotiation step takes place offline by exchanging modeling and objective information. For the online controller, one iterate of the distributed optimization is sufficient for stability. If time allows, more optimization iterates may be taken.

In the following sections, we provide further comments on some of the literature covered by Scattolini (2009) and additional background on recently published work.

2.2.1 Linear Distributed Control

2.2.1.1 Noncooperative Control

A noncooperative distributed control architecture for transportation networks is investigated in (Negenborn, De Schutter, and Hellendoorn, 2008). Gauss-Seidel and Jacobi, also often referred to as serial and parallel, optimization algorithms are proposed for the coordination of the controllers through an augmented Lagrangian method.[1] Mercangöz and Doyle (2007) develop a noncooperative controller for an experimental four-tank system. The experimental model is plantwide stable. Closed-loop stability is shown for a simulation and through a disturbance rejection experiment. For the experimental system, the proposed controller is shown to improve performance over decentralized control. Al-Ghewi, Budman, and Elkamel (2010) compare different plant subsystem decompositions for centralized, noncooperative, and decentralized control. They develop a mixed-integer nonlinear optimization that optimizes the plant performance based on a performance metric. The metric can be tuned to trade-off between controller performance and subsystem structure simplicity.

2.2.1.2 Cooperative Control

An article length presentation of cooperative control is given in (Venkat, Rawlings, and Wright, 2007). Exponential stability is proven for the state feedback case and for output feedback with decentralized Kalman filtering. Additionally, the paper includes a discussion of partial cooperation, in which some inter-subsystem models are removed from the distributed controller. The proposed controller is evaluated for chemical plant examples. Cooperative control is also applied to power distribution systems in (Venkat, Hiskens, Rawlings, and Wright, 2008). Necoara, Doan, and Suykens (2008) propose a novel distributed optimization algorithm using proximal

[1] See (Nocedal and Wright, 2006, pp.514–526) for a discussion of augmented Lagrangian methods.

center decomposition in order to speed up the optimization performance of cooperative control. Although the algorithm converges more quickly, it requires a coordinating dual optimization to be solved for the plantwide MPC problem. Another optimization speed up for the cooperative distributed optimization is presented in (Pannocchia, Wright, Stewart, and Rawlings, 2009). The proposed algorithm uses partial enumeration, in which a subset of the solutions to the subsystem control optimizations are stored for fast retrieval. An alternative to the work of Venkat et al. is the game theory approach of Maestre, Muñoz de la Peña, and Camacho (2010), in which two subsystems optimize a local objective function independently and the feedback is chosen by a cooperative game. The solutions are traded between the subsystems and the plantwide cost is evaluated for each solution. The subsystems then choose the control feedback that provides the largest decrease of the objective function from a table of the possible feedback choices. Camponogara and de Oliveira (2009) develop a decomposition method for the plantwide control problem. Feedback is determined using a Gauss-Seidel version of gradient projection.

2.2.2 Nonlinear Distributed Control

2.2.2.1 Noncooperative Control

The book by Başar and Olsder (1999) provides a fundamental analysis of the challenges present in noncooperative control for nonlinear systems. Popular references for nonlinear noncooperative control are (Camponogara, Jia, Krogh, and Talukdar, 2002) and (Jia and Krogh, 2002). In these articles stability is shown for weakly interacting, unconstrained systems. The authors provide a summary of Gauss-Seidel and Jacobi optimization algorithms for distributed control, but do not give convergence results for the control algorithm. Camponogara and Talukdar (2007) explicitly discuss the algorithm properties, proving the convergence of the Gauss-Seidel algorithm. They also provide heuristics to promote convergence of the Jacobi algorithm iterates to the Pareto set of plantwide feedback strategies.

Many of the results in noncooperative control have been presented for multi-vehicle applications, such as coordinating robots and unmanned aeronautical vehicles (UAVs). Noncooperative control often performs well for these applications because the vehicles do not dynamically interact. In several articles (Keviczky, Borrelli, and Balas, 2006, 2007; Keviczky, Borrelli, Fregene, Godbole, and Balas, 2008), a distributed control strategy for dynamically decoupled subsystems is presented. The subsystems are coupled through the objective function and exchange information with their neighbors. Each subsystem solves for the local feedback and also the feedback of its neighbors. Keviczky et al. prove asymptotic stability at the origin. Another approach is by Dunbar and Murray (2006), in which the subsystems optimization problems are augmented with a compatibility constraint that requires the actual feedback and the assumed feedback of the neighboring subsystems to stay within a bound. In (Dunbar, 2007), this method is extended to weakly coupled interacting subsystems. Franco, Magni, Parisini, Polycarpou, and Raimondo (2008) show stability for dynamically uncoupled agents that exchange information with each other and optimize a local objective. The coupling of the control problem is through the objective function, which is constructed so that the agents move toward a common objective. Stability is ensured if the coupling in the objective function is sufficiently small in the terminal region.

2.2.2.2 Cooperative Control

In a series of papers, Liu, Muñoz de la Peña, and Christofides develop a nonlinear distributed controller using Lyapunov-based control theory. The original idea for the controller is provided in (Liu, Muñoz de la Peña, Ohran, Christofides, and Davis, 2008) in which control of nonlinear systems is split between two tiers of controllers. The lower tier uses a Lyapunov-based controller to stabilize the plant, and an upper tier controller augments the plantwide control by providing additional feedback. A stability constraint is added to the upper tier to keep it from destabilizing the feedback from the lower tier. The plantwide control feedback is shown to be *practically* stable[2], which establishes that the plant stays in an invariant set containing

[2]For a definition of practical stability see (Lakshmikantham, Leela, and Martynyuk, 1990, pp.9-10)

the origin. In the paper (Liu, Muñoz de la Peña, and Christofides, 2009) the idea is expanded so that the lower tier is an MPC. The Lyapunov-based controller is used by the lower tier as an initial condition for the MPC, which tries to improve performance while meeting the stability constraint. After computing its feedback, the lower tier MPC sends its solution to the upper tier. As before, the upper tier MPC provides additional feedback. In both approaches, the lower tier controller is assumed to stabilize the entire plant without the upper tier controller. In (Liu, Chen, Muñoz de la Peña, and Christofides, 2010a), the authors expand the theory to account for asynchronous and delayed sampling. The estimates are not filtered, hence to maintain stability, the stabilizable set is shrunken to provide enough back-off for the error associated with the delay. In the paper (Chilin, Liu, Muñoz de la Peña, Christofides, and Davis, 2010), the two tier controller method is expanded to be stable during actuator faults. A backup controller, using a predefined Lyapunov-based controller, is designed and turned on during these faults.

A further expansion of the method is given in (Liu et al., 2010b) in which the number of subsystems is increased to any finite number of controllers and the assumption that one controller can stabilize the plant is removed. Two distributed optimization algorithms are proposed. The first algorithm is a Gauss-Seidel optimization strategy, in which each of the subsystem controllers solves for feedback one-at-a-time. The other algorithm is of the Jacobi class and allows the controllers to solve for feedback simultaneously. In both cases, the controllers are initialized with the feedback law from a pre-defined, stabilizing Lyapunov-based controller. Each controller tries to improve performance and then sends the solution of its local optimization to the other subsystems. It is not shown, however, that the optimization schemes improve the plantwide Lyapunov function. Therefore, to prove stability, the stabilizable set contains only those states in which the subsystems do not significantly interact.

Another Lyapunov-based approach is provided by Hermans, Lazar, and Jokić (2010a). Each subsystem is required to satisfy a Lyapunov stability constraint that involves only neighboring subsystems, leading to low communication. The distributed controller is applied to power networks in (Hermans, Lazar, Jokić, and van den Bosch, 2010b).

A weakness of this approach is that there does not exist a general method in the literature to find Lyapunov-based controllers. The Lyapunov-based controller is critical for the stability of the plant under feedback from the distributed MPC, because it is used to define the subsystem stability constraint. Moreover, it is assumed that the Lyapunov-based controller is defined for every state of interest. Developing *a priori* control laws is an active area of research and includes methods such as explicit MPC (Rawlings and Mayne, 2009, pp.483–519). Once a stabilizing plantwide control law is defined, however, the need for distributed control is weakly motivated. In contrast to this approach, we propose a nonlinear distributed controller in Chapter 6 that, given a feasible input sequence at initialization, generates a stabilizing warm start online. A pre-defined stabilizing control law is assumed to exist only in the terminal region.

2.2.3 Hierarchical and Coordinated Control

A classic reference for hierarchical control architectures is (Mesarović et al., 1970). Hierarchical and coordinating MPC is an active area of research. Aske, Strand, and Skogestad (2008) propose a coordinating controller for maximizing the economic plant profit, which is assumed to be achieved for the highest plant throughput. The coordinating controller tries to increase flow through the bottleneck by moving the plant steady state. Cheng, Forbes, and Yip (2007) develop a coordinated distributed optimization to solve for plantwide optimal steady states. The optimization is based on augmented Lagrangian algorithms. In (Marcos, Forbes, and Guay, 2009) the coordinated distributed optimization is extended to provide plantwide dynamic feedback. Negenborn, Leirens, De Schutter, and Hellendoorn (2009) propose a hierarchical MPC design for voltage control in a power network. The supervisory controller uses a pattern search algorithm to determine setpoints of the lower level controllers.

Chapter 3

Cooperative Control for Linear Systems

Note: Excluding Section 3.7, the text of this chapter appears in (Stewart, Venkat, Rawlings, Wright, and Pannocchia, 2010b).

3.1 Introduction

In this chapter, we state and prove the stability properties for cooperative distributed control under state and output feedback. In Section 3.2, we provide relevant theory for suboptimal control. Section 3.3 provides stability theory for cooperative control under state feedback. For ease of exposition, we introduce the theorems for the case of two controllers only. Section 3.4 extends these results to the output feedback case. The results are modified to handle coupled input constraints in Section 3.5. We then show how the theory extends to cover any finite number of controllers. We conclude with an example comparing the performance of cooperative control with other plantwide control strategies.

3.2 Suboptimal Model Predictive Control

Requiring distributed MPC strategies to converge is equivalent to implementing centralized MPC with the optimization distributed over many processors. Alternatively, we allow the subsystems to inject suboptimal inputs. This property increases the flexibility of distributed MPC, and the plantwide control strategy can be treated as a suboptimal MPC. In this section, we provide the definitions and theory of suboptimal MPC and draw upon these results in the sequel to establish stability of cooperative MPC.

We define the current state of the system as $x \in \mathbb{R}^n$, the trajectory of inputs

$$\mathbf{u} = \{u(0), u(1), \ldots, u(N-1)\} \in \mathbb{R}^{Nm}$$

and the state and input at time k as $(x(k), u(k))$. For the latter, we often abbreviate the notation as (x, u). Denote the input constraints as $\mathbf{u} \in \mathbb{U}^N$ in which $\mathbb{U}$ is compact, convex, and contains the origin in its interior. Denote $\mathbb{X}_N$ as the set of all x for which there exists a feasible $\mathbf{u}$. Initialized with a feasible input trajectory $\tilde{\mathbf{u}}$, the controller performs p iterations of a feasible path algorithm and computes $\mathbf{u}$ such that some performance metric is improved. At each sample time, the first input in the (suboptimal) trajectory is applied, $u = u(0)$. The state is updated by the state evolution equation $x^+ = f(x, u)$, in which x^+ is the state at the next iterate.

For any initial state $x(0)$, we initialize the suboptimal MPC with a feasible input sequence $\tilde{\mathbf{u}}(0) = \mathbf{h}(x(0))$ with $\mathbf{h}(\cdot)$ continuous. For subsequent decision times, we denote $\tilde{\mathbf{u}}^+$ as the *warm start*, a feasible input sequence for x^+ used to initialize the suboptimal MPC algorithm. Here, we set $\tilde{\mathbf{u}}^+ = \{u(1), \ldots, u(N-1), 0\}$. This sequence is obtained by discarding the first input, shifting the rest of the sequence forward one step and setting the last input to zero.

We observe that the input sequence at termination $\mathbf{u}^+$ is a function of the state initial condition x^+ and of the warm start $\tilde{\mathbf{u}}^+$. Noting that x^+ and $\tilde{\mathbf{u}}^+$ are both functions of x and $\mathbf{u}$, the input sequence $\mathbf{u}^+$ can be expressed as a function of only $(x, \mathbf{u})$ by $\mathbf{u}^+ = g(x, \mathbf{u})$. We refer to the function g as the iterate update.

Given a system $x^+ = f(x)$, with equilibrium point at the origin $0 = f(0)$, denote $\phi(k, x(0))$ as the solution $x(k)$ given the initial state $x(0)$. We consider the following definition.

Definition 3.1 (Exponential stability on a set $\mathbb{X}$)**.** *The origin is exponentially stable on the set* $\mathbb{X}$ *if for all* $x(0) \in \mathbb{X}$*, the solution* $\phi(k, x(0)) \in \mathbb{X}$ *and there exists* $\alpha > 0$ *and* $0 < \gamma < 1$ *such that*

$$\left|\phi(k, x(0))\right| \leq \alpha \, |x(0)| \, \gamma^k$$

for all $k \geq 0$.

The following lemma is an extension of (Scokaert, Mayne, and Rawlings, 1999, Theorem 1) for exponential stability.

Lemma 3.2 (Exponential stability of suboptimal MPC)**.** *Consider a system*

$$\begin{pmatrix} x^+ \\ \mathbf{u}^+ \end{pmatrix} = \begin{pmatrix} F(x,\mathbf{u}) \\ g(x,\mathbf{u}) \end{pmatrix} = \begin{pmatrix} f(x,u) \\ g(x,\mathbf{u}) \end{pmatrix} \qquad (x(0),\mathbf{u}(0)) \text{ given} \tag{3.2.1}$$

with a steady-state solution $(0,0) = (f(0,0), g(0,0))$*. Assume that the function* $V(\cdot) : \mathbb{R}^n \times \mathbb{R}^{Nm} \to \mathbb{R}_+$ *and input trajectory* $\mathbf{u}$ *satisfy*

$$a\,|(x,\mathbf{u})|^2 \le V(x,\mathbf{u}) \le b\,|(x,\mathbf{u})|^2 \tag{3.2.2a}$$

$$V(x^+,\mathbf{u}^+) - V(x,\mathbf{u}) \le -c\,|(x,u(0))|^2 \tag{3.2.2b}$$

$$|\mathbf{u}| \le d\,|x| \quad x \in \mathbb{B}_r \tag{3.2.2c}$$

in which $a,b,c,d,r > 0$*. If* $\mathbb{X}_N$ *is forward invariant for the system* $x^+ = f(x,u)$*, the origin is exponentially stable for all* $x(0) \in \mathbb{X}_N$*.*

Notice in the second inequality (3.2.2b), only the first input appears in the norm $|(x,u(0))|^2$. Note also that the last inequality applies only for x in a ball of radius r, which may be chosen arbitrarily small.

Proof of Lemma 3.2. First we establish that the origin of the extended system (3.2.1) is exponentially stable for all $(x(0),\mathbf{u}(0)) \in \mathbb{X}_N \times \mathbb{U}^N$. For $x \in \mathbb{B}_r$, we have $|\mathbf{u}| \le d\,|x|$. Consider the optimization

$$s = \max_{\mathbf{u} \in \mathbb{U}^N} |\mathbf{u}|$$

The solution exists by the Weierstrass theorem since $\mathbb{U}^N$ is compact and by definition we have that $s > 0$. Then we have $|\mathbf{u}| \le (s/r)\,|x|$ for $x \notin \mathbb{B}_r$. Therefore, for all $x \in \mathbb{X}_N$, we have $|\mathbf{u}| \le \bar{d}\,|x|$ in which $\bar{d} = \max(d, s/r)$, and

$$|(x,\mathbf{u})| \le |x| + |\mathbf{u}| \le (1+\bar{d})\,|x| \le (1+\bar{d})\,|(x,u(0))|$$

which gives $|(x,u(0))| \ge \bar{c}\,|(x,\mathbf{u})|$ with $\bar{c} = 1/(1+\bar{d}) > 0$. Therefore the extended state $(x,\mathbf{u})$ satisfies

$$V(x^+,\mathbf{u}^+) - V(x,\mathbf{u}) \le -\tilde{c}\,|(x,\mathbf{u})|^2 \quad (x,\mathbf{u}) \in \mathbb{X}_N \times \mathbb{U}^N \tag{3.2.3}$$

in which $\tilde{c} = c(\bar{c})^2$. Together with (3.2.2), (3.2.3) establishes that $V(\cdot)$ is a Lyapunov function of the extended state $(x, \mathbf{u})$ for all $x \in \mathbb{X}_N$ and $\mathbf{u} \in \mathbb{U}^N$. Hence for all $(x(0), \mathbf{u}(0)) \in \mathbb{X}_N \times \mathbb{U}^N$ and $k \geq 0$, we have

$$|(x(k), \mathbf{u}(k))| \leq \alpha\, |(x(0), \mathbf{u}(0))|\, \gamma^k$$

in which $\alpha > 0$ and $0 < \gamma < 1$. Notice that $\mathbb{X}_N \times \mathbb{U}^N$ is forward invariant for the extended system (3.2.1).

Finally, we remove the input sequence and establish that the origin is exponentially stable for the closed-loop system. We have for all $x(0) \in \mathbb{X}_N$ and $k \geq 0$

$$\begin{aligned} \big|\phi(k, x(0))\big| = |x(k)| &\leq |(x(k), \mathbf{u}(k))| \leq \alpha\, |(x(0), \mathbf{u}(0))|\, \gamma^k \\ &\leq \alpha(|x(0)| + |\mathbf{u}(0)|)\gamma^k \leq \alpha(1 + \bar{d})\, |x(0)|\, \gamma^k \\ &\leq \bar{\alpha}\, |x(0)|\, \gamma^k \end{aligned}$$

in which $\bar{\alpha} = \alpha(1 + \bar{d}) > 0$, and we have established exponential stability of the origin by observing that $\mathbb{X}_N$ is forward invariant for the closed-loop system $\phi(k, x(0))$. □

Remark 3.3. *For Lemma 3.2, we use the fact that* $\mathbb{U}$ *is compact. For unbounded* $\mathbb{U}$ *exponential stability may instead be established by compactness of* $\mathbb{X}_N$.

3.3 Cooperative Model Predictive Control

We now show that cooperative MPC is a form of suboptimal MPC and establish stability. To simplify the exposition and proofs, in Sections 3.3–3.5 we assume that the plant consists of only two subsystems. We then establish in Section 3.6 that the results extend to any finite number of subsystems.

3.3.1 Definitions

3.3.1.1 Models

We assume for each subsystem i that there exist a collection of linear models denoting the effects of inputs of subsystem j on the states of subsystem i for all $(i,j) \in \mathbb{I}_{1:2} \times \mathbb{I}_{1:2}$

$$x_{ij}^+ = A_{ij}x_{ij} + B_{ij}u_j$$

in which $x_{ij} \in \mathbb{R}^{n_{ij}}$, $u_j \in \mathbb{R}^{m_j}$, $A_{ij} \in \mathbb{R}^{n_{ij}\times n_{ij}}$, and $B_{ij} \in \mathbb{R}^{n_{ij}\times m_j}$. For a discussion of identification of this model choice, see (Gudi and Rawlings, 2006). In Section 5.2.2, we show how these subsystem models are related to the centralized model. Considering subsystem 1, we collect the states to form

$$\begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}^+ = \begin{bmatrix} A_{11} & \\ & A_{12} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} B_{11} \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ B_{12} \end{bmatrix} u_2$$

which denotes the model for subsystem 1. To simplify the notation, we define the equivalent model

$$x_1^+ = A_1 x_1 + \bar{B}_{11}u_1 + \bar{B}_{12}u_2$$

for which

$$x_1 = \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} \quad A_1 = \begin{bmatrix} A_{11} & \\ & A_{12} \end{bmatrix} \quad \bar{B}_{11} = \begin{bmatrix} B_{11} \\ 0 \end{bmatrix} \quad \bar{B}_{12} = \begin{bmatrix} 0 \\ B_{12} \end{bmatrix}$$

in which $x_1 \in \mathbb{R}^{n_1}$, $A_1 \in \mathbb{R}^{n_1\times n_1}$, and $\bar{B}_{1j} \in \mathbb{R}^{n_1\times m_j}$ with $n_1 = n_{11} + n_{12}$. Forming a similar model for subsystem 2, the plantwide model is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \bar{B}_{11} \\ \bar{B}_{21} \end{bmatrix} u_1 + \begin{bmatrix} \bar{B}_{12} \\ \bar{B}_{22} \end{bmatrix} u_2$$

We further simplify the plantwide model notation to

$$x^+ = Ax + B_1u_1 + B_2u_2$$

for which

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad A = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} \quad B_1 = \begin{bmatrix} \bar{B}_{11} \\ \bar{B}_{21} \end{bmatrix} \quad B_2 = \begin{bmatrix} \bar{B}_{12} \\ \bar{B}_{22} \end{bmatrix}$$

3.3.1.2 Objective Functions

Consider subsystem 1, for which we define the quadratic stage cost and terminal penalty, respectively

$$\ell_1(x_1, u_1) = \frac{1}{2}(x_1' Q_1 x_1 + u_1' R_1 u_1) \tag{3.3.1a}$$

$$V_{1f}(x_1) = \frac{1}{2} x_1' P_{1f} x_1 \tag{3.3.1b}$$

in which $Q_1 \in \mathbb{R}^{n_1 \times n_1}$, $R_1 \in \mathbb{R}^{m_1 \times m_1}$, and $P_{1f} \in \mathbb{R}^{n_1 \times n_1}$. We define the objective function for subsystem 1

$$V_1\big(x_1(0), \mathbf{u}_1, \mathbf{u}_2\big) = \sum_{k=0}^{N-1} \ell_1\big(x_1(k), u_1(k)\big) + V_{1f}\big(x_1(N)\big)$$

Notice V_1 is implicitly a function of both $\mathbf{u}_1$ and $\mathbf{u}_2$ because x_1 is a function of both u_1 and u_2. For subsystem 2, we similarly define an objective function V_2. We define the plantwide objective function

$$V\big(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2\big) = \rho_1 V_1\big(x_1(0), \mathbf{u}_1, \mathbf{u}_2\big) + \rho_2 V_2\big(x_2(0), \mathbf{u}_1, \mathbf{u}_2\big)$$

in which $\rho_1, \rho_2 > 0$ are relative weights. For notational simplicity, we write $V(x, \mathbf{u})$ for the plant objective.

3.3.1.3 Constraints

We require that the inputs satisfy

$$u_1(k) \in \mathbb{U}_1 \qquad u_2(k) \in \mathbb{U}_2 \quad k \in \mathbb{I}_{0:N-1}$$

in which $\mathbb{U}_1$ and $\mathbb{U}_2$ are compact and convex such that 0 is in the interior of $\mathbb{U}_i \ \forall i \in \mathbb{I}_{1:2}$.

Remark 3.4. *The constraints are termed uncoupled because the feasible region of* $\mathbf{u}_1$ *is not affected by* $\mathbf{u}_2$*, and vice-versa.*

3.3.1.4 Assumptions

For every $i \in \mathbb{I}_{1:2}$, let $\underline{\mathrm{A}}_i = \mathrm{diag}(A_{1i}, A_{2i})$ and $\underline{\mathrm{B}}_i = \left[\begin{smallmatrix} B_{1i} \\ B_{2i} \end{smallmatrix}\right]$. The following assumptions are used to establish stability.

Assumption 3.5. *For all $i \in \mathbb{I}_{1:2}$*

(a) The systems $(\underline{A}_i, \underline{B}_i)$ are stabilizable.

(b) The input penalties $R_i > 0$.

(c) The state penalties $Q_i \succeq 0$.

(d) The systems (A_i, Q_i) are detectable.

(e) $N \geq \max_{i \in \mathbb{I}_{1:2}} (\underline{n}_i^u)$, in which $\underline{n}_i^u$ is the number of unstable modes of $\underline{A}_i$, i.e., the number of $\lambda \in eig(\underline{A}_i)$ such that $|\lambda| \geq 1$.

The assumption 3.5(e) is required so that the horizon N is sufficiently large to zero the unstable modes.

3.3.1.5 Unstable Modes

For an unstable plant, we constrain the unstable modes to be zero at the end of the horizon to maintain closed-loop stability. To construct this constraint, consider the real Schur decomposition of A_{ij} for each $(i, j) \in \mathbb{I}_{1:2} \times \mathbb{I}_{1:2}$

$$A_{ij} = \begin{bmatrix} S_{ij}^s & S_{ij}^u \end{bmatrix} \begin{bmatrix} A_{ij}^s & - \\ - & A_{ij}^u \end{bmatrix} \begin{bmatrix} S_{ij}^{s\prime} \\ S_{ij}^{u\prime} \end{bmatrix} \tag{3.3.2}$$

in which A_{ij}^s is stable and A_{ij}^u has all unstable eigenvalues.

3.3.1.6 Terminal Penalty

Given the definition of the Schur decomposition (3.3.2), we define the matrices

$$S_i^s = \mathrm{diag}(S_{i1}^s, S_{i2}^s) \qquad A_i^s = \mathrm{diag}(A_{i1}^s, A_{i2}^s) \quad \forall i \in \mathbb{I}_{1:2} \tag{3.3.3a}$$

$$S_i^u = \mathrm{diag}(S_{i1}^u, S_{i2}^u) \qquad A_i^u = \mathrm{diag}(A_{i1}^u, A_{i2}^u) \quad \forall i \in \mathbb{I}_{1:2} \tag{3.3.3b}$$

Lemma 3.6. *The matrices (3.3.3) satisfy the Schur decompositions*

$$A_i = \begin{bmatrix} S_i^s & S_i^u \end{bmatrix} \begin{bmatrix} A_i^s & - \\ & A_i^u \end{bmatrix} \begin{bmatrix} S_i^{s\prime} \\ S_i^{u\prime} \end{bmatrix} \quad \forall i \in \mathbb{I}_{1:2}$$

Let Σ_1 and Σ_2 denote the solution of the Lyapunov equations

$$A_1^{s\prime}\Sigma_1 A_1^s - \Sigma_1 = -S_1^{s\prime} Q_1 S_1^s \quad A_2^{s\prime}\Sigma_2 A_2^s - \Sigma_2 = -S_2^{s\prime} Q_2 S_2^s \tag{3.3.4}$$

We then choose the terminal penalty for each subsystem to be the cost to go under zero control, such that

$$P_{1f} = S_1^s \Sigma_1 S_1^{s\prime} \qquad P_{2f} = S_2^s \Sigma_2 S_2^{s\prime} \tag{3.3.5}$$

3.3.1.7 Cooperative Model Predictive Control Algorithm

Let $\boldsymbol{v}^0$ be the initial condition for the cooperative MPC algorithm (see Section 3.3.2 for the discussion of initialization). At each iterate $p \geq 0$, the following optimization problem is solved for subsystem i, $i \in \mathbb{I}_{1:2}$

$$\min_{\boldsymbol{v}_i} V(x_1(0), x_2(0), \boldsymbol{v}_1, \boldsymbol{v}_2) \tag{3.3.6a}$$

subject to

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \bar{B}_{11} \\ \bar{B}_{21} \end{bmatrix} v_1 + \begin{bmatrix} \bar{B}_{12} \\ \bar{B}_{22} \end{bmatrix} v_2 \tag{3.3.6b}$$

$$\boldsymbol{v}_i \in \mathbb{U}_i^N \tag{3.3.6c}$$

$$S_{ji}^{u\,\prime} x_{ji}(N) = 0 \quad j \in \mathbb{I}_{1:2} \tag{3.3.6d}$$

$$|\boldsymbol{v}_i| \leq d_i \sum_{j \in \mathbb{I}_{1:2}} \left| x_{ji}(0) \right| \quad \text{if } x_{ji}(0) \in \mathbb{B}_r \quad \forall j \in \mathbb{I}_{1:2} \tag{3.3.6e}$$

$$\boldsymbol{v}_j = \boldsymbol{v}_j^p \quad j \in \mathbb{I}_{1:2} \setminus i \tag{3.3.6f}$$

in which we include the hard input constraints, the stabilizing constraint on the unstable modes, and the Lyapunov stability constraint. We denote the solutions to these problems as

$$\boldsymbol{v}_1^*(x_1(0), x_2(0), \boldsymbol{v}_2^p), \qquad \boldsymbol{v}_2^*(x_1(0), x_2(0), \boldsymbol{v}_1^p)$$

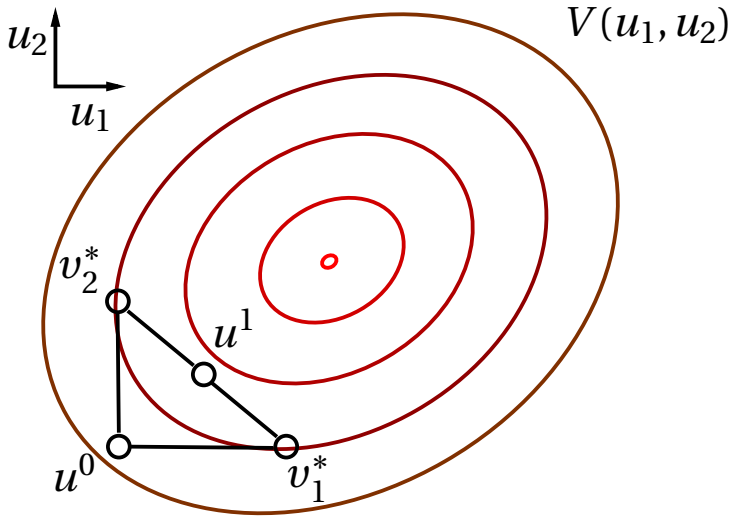


Figure 3.1: An iterate of the distributed optimization.

Given the prior, feasible iterate $(\boldsymbol{v}_1^p, \boldsymbol{v}_2^p)$, the next iterate is defined to be

$$(\boldsymbol{v}_1^{p+1}, \boldsymbol{v}_2^{p+1}) = w_1\Big(\boldsymbol{v}_1^*(\boldsymbol{v}_2^p), \boldsymbol{v}_2^p\Big) + w_2\Big(\boldsymbol{v}_1^p, \boldsymbol{v}_2^*(\boldsymbol{v}_1^p)\Big) \tag{3.3.7}$$

$$w_1 + w_2 = 1, \quad w_1, w_2 > 0$$

for which we omit the state dependence of $\boldsymbol{v}_1^*$ and $\boldsymbol{v}_2^*$ to reduce notation. This distributed optimization is of the Gauss-Jacobi type (see Bertsekas and Tsitsiklis, 1997, pp.219–223). A graphical representation of the algorithm is shown in Figure 3.1. At the last iterate $\bar{p}$, we set $\mathbf{u} \leftarrow (\boldsymbol{v}_1^{\bar{p}}, \boldsymbol{v}_2^{\bar{p}})$ and inject $u(0)$ into the plant.

The following properties follow immediately.

Lemma 3.7 (Feasibility)**.** *Given a feasible initial guess, the iterates satisfy*

$$(\boldsymbol{v}_1^p, \boldsymbol{v}_2^p) \in \mathbb{U}_1^N \times \mathbb{U}_2^N$$

for all $p \geq 1$.

Proof. By assumption, the initial guess is feasible. Because $\mathbb{U}_1$ and $\mathbb{U}_2$ are convex, the convex combination (3.3.7) with $p = 0$ implies $(\boldsymbol{v}_1^1, \boldsymbol{v}_2^1)$ is feasible. Feasibility for $p > 1$ follows by induction. □

Lemma 3.8 (Convergence)**.** *The cost $V(x(0),\boldsymbol{v}^p)$ is nonincreasing for each iterate p and converges as $p\to\infty$.*

Proof. For every $p\geq 0$, the cost function satisfies the following

$$\begin{aligned} V(x(0),\boldsymbol{v}^{p+1}) &= V(x(0), w_1(\boldsymbol{v}_1^*,\boldsymbol{v}_2^p) + w_2(\boldsymbol{v}_1^p,\boldsymbol{v}_2^*)) \\ &\leq w_1 V(x(0),(\boldsymbol{v}_1^*,\boldsymbol{v}_2^p)) + w_2 V(x(0),(\boldsymbol{v}_1^p,\boldsymbol{v}_2^*)) \qquad (3.3.8a) \\ &\leq w_1 V(x(0),(\boldsymbol{v}_1^p,\boldsymbol{v}_2^p)) + w_2 V(x(0),(\boldsymbol{v}_1^p,\boldsymbol{v}_2^p)) \qquad (3.3.8b) \\ &\leq V(x(0),\boldsymbol{v}^p) \end{aligned}$$

The first equality follows from (3.3.7). The inequality (3.3.8a) follows from convexity of $V(\cdot)$. The next inequality (3.3.8b) follows from the optimality of $\boldsymbol{v}_i^*\ \forall\, i\in\mathbb{I}_{1:2}$, and the final line follows from $w_1+w_2=1$. Because the cost is bounded below, it converges. □

Lemma 3.9 (Optimality)**.** *As $p\to\infty$ the cost $V(x(0),\boldsymbol{v}^p)$ converges to the optimal value $V^0(x(0))$, and the iterates $(\boldsymbol{v}_1^p,\boldsymbol{v}_2^p)$ converge to $(\mathbf{u}_1^0,\mathbf{u}_2^0)$ in which $\mathbf{u}^0=(\mathbf{u}_1^0,\mathbf{u}_2^0)$ is the Pareto (centralized) optimal solution.*

Proof. We give a proof that requires only closedness (not compactness) of $\mathbb{U}_i$, $i\in\mathbb{I}_{1:2}$. From Lemma 3.8, the cost converges, say to $\underline{V}$. Since V is quadratic and strongly convex, its sublevel sets $\mathrm{lev}_{\leq a}(V)$ are compact and bounded for all a. Hence, all iterates belong to the compact set $\mathrm{lev}_{\leq V(\boldsymbol{v}^0)}(V)\cap\mathbb{U}$, so there is at least one accumulation point. Let $\bar{\boldsymbol{v}}$ be any such accumulation point, and choose a subsequence $\mathscr{P}\subset\{1,2,3,\ldots\}$ such that $\{\boldsymbol{v}^p\}_{p\in\mathscr{P}}$ converges to $\bar{\boldsymbol{v}}$. We obviously have that $V(x(0),\bar{\boldsymbol{v}})=\underline{V}$, and moreover that

$$\lim_{p\in\mathscr{P},\,p\to\infty} V(x(0),\boldsymbol{v}^p) = \lim_{p\in\mathscr{P},\,p\to\infty} V(x(0),\boldsymbol{v}^{p+1}) = \underline{V} \qquad (3.3.9)$$

By strict convexity of V and compactness of $\mathbb{U}_i$, $i\in\mathbb{I}_{1:2}$, the minimizer of $V(x(0),\cdot)$ is attained at a unique point $\mathbf{u}^0=(\mathbf{u}_1^0,\mathbf{u}_2^0)$. By taking limits in (3.3.8) as $p\to\infty$ for $p\in\mathscr{P}$, and using $w_1>0$, $w_2>0$, we deduce directly that

$$\lim_{p\in\mathscr{P},\,p\to\infty} V(x(0),(\boldsymbol{v}_1^*(\boldsymbol{v}_2^p),\boldsymbol{v}_2^p)) = \underline{V} \qquad (3.3.10a)$$

$$\lim_{p\in\mathscr{P},\,p\to\infty} V(x(0),(\boldsymbol{v}_1^p,\boldsymbol{v}_2^*(\boldsymbol{v}_1^p))) = \underline{V} \qquad (3.3.10b)$$

We suppose for contradiction that $\underline{V} \neq V(x(0), \mathbf{u}^0)$ and thus $\bar{\boldsymbol{v}} \neq \mathbf{u}^0$. Because $V(x(0), \cdot)$ is convex, we have

$$\nabla V(x(0), \bar{\boldsymbol{v}})'(\mathbf{u}^0 - \bar{\boldsymbol{v}}) \leq \Delta V := V(x(0), \mathbf{u}^0) - V(x(0), \bar{\boldsymbol{v}}) < 0$$

where $\nabla V(x(0), \cdot)$ denotes the gradient of $V(x(0), \cdot)$. It follows immediately that either

$$\nabla V(x(0), \bar{\boldsymbol{v}})' \begin{bmatrix} \mathbf{u}_1^0 - \bar{\boldsymbol{v}}_1 \\ 0 \end{bmatrix} \leq (1/2)\Delta V \text{ or} \tag{3.3.11a}$$

$$\nabla V(x(0), \bar{\boldsymbol{v}})' \begin{bmatrix} 0 \\ \mathbf{u}_2^0 - \bar{\boldsymbol{v}}_2 \end{bmatrix} \leq (1/2)\Delta V \tag{3.3.11b}$$

Suppose first that (3.3.11a) holds. Applying Taylor's theorem to V

$$\begin{aligned} &V(x(0), (\boldsymbol{v}_1^p + \epsilon(\mathbf{u}_1^0 - \boldsymbol{v}_1^p), \boldsymbol{v}_2^p)) \\ &= V(x(0), \boldsymbol{v}^p) + \epsilon \nabla V(x(0), \boldsymbol{v}^p)' \begin{bmatrix} \mathbf{u}_1^0 - \boldsymbol{v}_1^p \\ 0 \end{bmatrix} \\ &\quad + \frac{1}{2}\epsilon^2 \begin{bmatrix} \mathbf{u}_1^0 - \boldsymbol{v}_1^p \\ 0 \end{bmatrix}' \nabla^2 V(x(0), \boldsymbol{v}_1^p + \gamma\epsilon(\mathbf{u}_1^0 - \boldsymbol{v}_1^p), \boldsymbol{v}_2^p) \begin{bmatrix} \mathbf{u}_1^0 - \boldsymbol{v}_1^p \\ 0 \end{bmatrix} \\ &\leq \underline{V} + (1/4)\epsilon\Delta V + \beta\epsilon^2 \end{aligned} \tag{3.3.12}$$

in which $\gamma \in (0, \epsilon)$. (3.3.12) applies for all $p \in \mathscr{P}$ sufficiently large, for some β independent of ϵ and p. By fixing ϵ to a suitably small value (certainly less than 1), we have both that the right-hand side of (3.3.12) is strictly less than $\underline{V}$ and that $\boldsymbol{v}_1^p + \epsilon(\mathbf{u}_1^0 - \boldsymbol{v}_1^p) \in \mathbb{U}_1$. By taking limits in (3.3.12) and using (3.3.10) and the fact that $\boldsymbol{v}_1^*(\boldsymbol{v}_2^p)$ is optimal for $V(x(0), (\cdot, \boldsymbol{v}_2^p))$ in $\mathbb{U}_1$, we have

$$\begin{aligned} \underline{V} &= \lim_{p \in \mathscr{P},\, p \to \infty} V(x(0), (\boldsymbol{v}_1^*(\boldsymbol{v}_2^p), \boldsymbol{v}_2^p)) \\ &\leq \lim_{p \in \mathscr{P},\, p \to \infty} V(x(0), (\boldsymbol{v}_1^p + \epsilon(\mathbf{u}_1^0 - \boldsymbol{v}_1^p), \boldsymbol{v}_2^p)) \\ &< \underline{V} \end{aligned}$$

giving a contradiction. By identical logic, we obtain the same contradiction from (3.3.11b). We conclude that $\underline{V} = V(x(0), \mathbf{u}^0)$ and thus $\bar{\boldsymbol{v}} = \mathbf{u}^0$. Since $\bar{\boldsymbol{v}}$ was an arbitrary accumulation point of

the sequence $\{\boldsymbol{v}^p\}$, and since this sequence is confined to a compact set, we conclude that the whole sequence converges to $\mathbf{u}^0$. □

Remark 3.10. *We present the distributed optimization algorithm with subproblem (3.3.6) and iterate update (3.3.7) so that the Lemmas 3.7–3.9 are satisfied. This choice is nonunique and other optimization methods may exist satisfying these properties.*

3.3.2 Stability of Cooperative Model Predictive Control

We define the steerable set $\mathbb{X}_N$ as the set of all x such that there exists a $\mathbf{u} \in \mathbb{U}^N$ satisfying (3.3.6d).

Assumption 3.11. *Given $r > 0$, for all $i \in \mathbb{I}_{1:2}$, d_i is chosen large enough such that there exists a $\mathbf{u}_i \in \mathbb{U}^N$ satisfying $|\mathbf{u}_i| \leq d_i \sum_{j \in \mathbb{I}_{1:2}} \left| x_{ij} \right|$ and (3.3.6d) for all $x_{ij} \in \mathbb{B}_r \ \forall j \in \mathbb{I}_{1:2}$.*

Remark 3.12. *Given Assumption 3.11, $\mathbb{X}_N$ is forward invariant.*

We now establish stability of the closed-loop system by treating cooperative MPC as a form of suboptimal MPC. We define the warm start for each subsystem as

$$\tilde{\mathbf{u}}_1^+ = \{u_1(1), u_1(2), \ldots, u_1(N-1), 0\}$$
$$\tilde{\mathbf{u}}_2^+ = \{u_2(1), u_2(2), \ldots, u_2(N-1), 0\}$$

The warm start $\tilde{\mathbf{u}}_i^+$ is used as the initial condition for the cooperative MPC problem in each subsystem i. We define the functions g_1^p and g_2^p as the outcome of applying the cooperative control iteration (3.3.7) p times

$$\mathbf{u}_1^+ = g_1^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) \qquad \mathbf{u}_2^+ = g_2^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2)$$

The system evolution is then given by

$$\begin{pmatrix} x_1^+ \\ x_2^+ \\ \mathbf{u}_1^+ \\ \mathbf{u}_2^+ \end{pmatrix} = \begin{pmatrix} A_1 x_1 + \bar{B}_{11} u_1 + \bar{B}_{12} u_2 \\ A_2 x_2 + \bar{B}_{21} u_1 + \bar{B}_{22} u_2 \\ g_1^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) \\ g_2^p(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) \end{pmatrix}$$

which we simplify to

$$\begin{pmatrix} x^+ \\ \mathbf{u}^+ \end{pmatrix} = \begin{pmatrix} Ax + B_1 u_1 + B_2 u_2 \\ g^p(x, \mathbf{u}) \end{pmatrix}$$

Theorem 3.13 (Exponential stability)**.** *Given Assumptions 3.5 and 3.11, the origin* $(x = 0)$ *of the closed-loop system* $x^+ = Ax + B_1 u_1 + B_2 u_2$ *is exponentially stable on the set* $\mathbb{X}_N$*.*

Proof. By eliminating the states in $\ell_i(\cdot)$, we can write V in the form $V(x, \mathbf{u}) = \frac{1}{2} x' \mathcal{Q} x + \frac{1}{2} \mathbf{u}' \mathcal{R} \mathbf{u} + x' \mathcal{S} \mathbf{u}$. Defining $\mathcal{H} = \left[\begin{smallmatrix} \mathcal{Q} & \mathcal{S} \\ \mathcal{S}' & \mathcal{R} \end{smallmatrix}\right] > 0$, $V(\cdot)$ satisfies (3.2.2a) by choosing $a = \frac{1}{2} \min_i(\lambda_i(\mathcal{H}))$ and $b = \frac{1}{2} \max_i(\lambda_i(\mathcal{H}))$. Next we show that $V(\cdot)$ satisfies (3.2.2b). Using the warm start at the next sample time, we have the following cost

$$\begin{aligned} V(x^+, \tilde{\mathbf{u}}^+) = V(x, \mathbf{u}) &- \frac{1}{2} \rho_1 \ell_1(x_1, u_1) - \frac{1}{2} \rho_2 \ell_2(x_2, u_2) \\ &+ \frac{1}{2} \rho_1 x_1(N)' \Big(A_1' P_{1f} A_1 - P_{1f} + Q_1 \Big) x_1(N) \\ &+ \frac{1}{2} \rho_2 x_2(N)' \Big(A_2' P_{2f} A_2 - P_{2f} + Q_2 \Big) x_2(N) \end{aligned} \tag{3.3.13}$$

Using the Schur decomposition defined in Lemma 3.6, and the constraints (3.3.6d) and (3.3.5), the last two terms of (3.3.13) can be written as

$$\begin{aligned} &\frac{1}{2} \rho_1 x_1(N)' S_1^s \Big(A_1^{s\prime} \Sigma_1 A_1^s - \Sigma_1 + S_1^{s\prime} Q_1 S_1^s \Big) S_1^{s\prime} x_1(N) \\ +&\frac{1}{2} \rho_2 x_2(N)' S_2^s \Big(A_2^{s\prime} \Sigma_2 A_2^s - \Sigma_2 + S_2^{s\prime} Q_2 S_2^s \Big) S_2^{s\prime} x_2(N) = 0 \end{aligned}$$

These terms are zero because of (3.3.4). Using this result and applying the iteration of the controllers gives

$$V(x^+, \mathbf{u}^+) \leq V(x, \mathbf{u}) - \frac{1}{2} \rho_1 \ell_1(x_1, u_1) - \frac{1}{2} \rho_2 \ell_2(x_2, u_2)$$

Because ℓ_i is quadratic in both arguments, there exists a $c > 0$ such that

$$V(x^+, \mathbf{u}^+) - V(x, \mathbf{u}) \leq -c\, |(x, u)|^2$$

The Lyapunov stability constraint (3.3.6e) for $x_{11}, x_{12}, x_{21}, x_{22} \in \mathbb{B}_r$ implies for $(x_1, x_2) \in \mathbb{B}_r$ that $|(\mathbf{u}_1, \mathbf{u}_2)| \leq 2\hat{d}\, |(x_1, x_2)|$ in which $\hat{d} = \max(d_1, d_2)$, satisfying (3.2.2c). Therefore the closed-loop system satisfies Lemma 3.2. Hence the closed-loop system is exponentially stable. □

3.4 Output Feedback

We now consider the stability of the closed-loop system with estimator error.

3.4.1 Models

For all $(i,j) \in \mathbb{I}_{1:2} \times \mathbb{I}_{1:2}$

$$x_{ij}^+ = A_{ij}x_{ij} + B_{ij}u_j \tag{3.4.1a}$$

$$y_i = \sum_{j \in \mathbb{I}_{1:2}} C_{ij}x_{ij} \tag{3.4.1b}$$

in which $y_i \in \mathbb{R}^{p_i}$ is the output of subsystem i and $C_{ij} \in \mathbb{R}^{p_i \times n_{ij})}$. Consider subsystem 1. As above, we collect the states to form $y_1 = \begin{bmatrix} C_{11} & C_{12} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}$ and use the simplified notation $y_1 = C_1 x_1$ to form the output model for subsystem 1.

Assumption 3.14. *For all $i \in \mathbb{I}_{1:2}$, (A_i, C_i) is detectable.*

3.4.2 Estimator

We construct a decentralized estimator. Consider subsystem 1, for which the local measurement y_1 and both inputs u_1 and u_2 are available, but x_1 must be estimated. The estimate satisfies

$$\hat{x}_1^+ = A_1\hat{x}_1 + \bar{B}_{11}u_1 + \bar{B}_{12}u_2 + L_1(y_1 - C_1\hat{x}_1)$$

in which $\hat{x}_1$ is the estimate of x_1 and L_1 is the Kalman filter gain. Defining the estimate error as $e_1 = x_1 - \hat{x}_1$ we have $e_1^+ = (A_1 - L_1C_1)e_1$. By Assumptions 3.5 and 3.14 there exists an L_1 such that $(A_1 - L_1C_1)$ is stable and therefore the estimator for subsystem 1 is stable. Defining e_2 similarly, the estimate error for the plant evolves

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix}^+ = \begin{bmatrix} A_{L1} & \\ & A_{L2} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

in which $A_{Li} = A_i - L_iC_i$. We collect the estimate error of each subsystem together and write $e^+ = A_L e$.

3.4.3 Reinitialization

We define the reinitialization step required to recover feasibility of the warm start for the perturbed state terminal constraint. For each $i \in \mathbb{I}_{1:2}$, define

$$\mathbf{h}_i^+(e) = \arg\min_{\mathbf{u}_i} \left\{ \left| \mathbf{u}_i - \tilde{\mathbf{u}}_i^+ \right|_{\mathscr{R}_i}^2 \;\middle|\; \begin{matrix} F_{ji}(\mathbf{u}_i - \tilde{\mathbf{u}}_i^+) = f_{ji} e_j \;\forall\, j \in \mathbb{I}_{1:2} \\ \mathbf{u}_i \in \mathbb{U}^N \end{matrix} \right\}$$

in which $\mathscr{R}_i = \operatorname{diag}(R_i)$, $F_{ji} = S_{ji}^{u\,\prime} \mathscr{C}_{ji}$, $f_{ji} = -S_{ji}^{u\,\prime} A_{ji}^N L_{ji}$, and $\mathscr{C}_{ji} = [B_{ji}\ A_{ji}B_{ji}\ \cdots\ A_{ji}^{N-1}B_{ji}]$ for all $i, j \in \mathbb{I}_{1:2}$. We use $\mathbf{h}_i^+(e)$ as the initial condition for the control optimization (3.3.6) for all $i \in \mathbb{I}_{1:2}$.

Proposition 3.15. *The reinitialization* $\mathbf{h}^+(\cdot) = (\mathbf{h}_1^+(\cdot), \mathbf{h}_2^+(\cdot))$ *is Lipschitz continuous on bounded sets.*

Proof. The proof follows from Prop. 7.13 (Rawlings and Mayne, 2009, p.499). □

3.4.4 Stability with Estimate Error

We consider the stability properties of the extended closed-loop system

$$\begin{pmatrix} \hat{x} \\ \mathbf{u} \\ e \end{pmatrix}^+ = \begin{pmatrix} F(\hat{x}, \mathbf{u}) + Le \\ g^p(\hat{x}, \mathbf{u}, e) \\ A_L e \end{pmatrix} \tag{3.4.2}$$

in which $F(\hat{x}, \mathbf{u}) = A\hat{x} + B_1 u_1 + B_2 u_2$ and $L = \operatorname{diag}(L_1 C_1, L_2 C_2)$. The function g^p includes the reinitialization step. Because A_L is stable there exists a Lyapunov function $J(\cdot)$ with the following properties

$$\bar{a}|e|^\sigma \le J(e) \le \bar{b}|e|^\sigma$$
$$J(e^+) - J(e) \le -\bar{c}|e|^\sigma$$

in which $\sigma > 0$, $\bar{a}, \bar{b} > 0$, and the constant $\bar{c} > 0$ can be chosen as large as desired by scaling $J(\cdot)$. For the remainder of this section, we choose $\sigma = 1$ in order to match the Lipschitz continuity of the plantwide objective function $V(\cdot)$. From the nominal properties of cooperative MPC, the

origin of the nominal closed-loop system $x^+ = Ax + B_1 u_1 + B_2 u_2$ is exponentially stable on $\mathbb{X}_N$ if the suboptimal input trajectory $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2)$ is computed using the actual state x, and the cost function $V(x, \mathbf{u})$ satisfies (3.2.2). We require the following feasibility assumption.

Assumption 3.16. *The set $\mathbb{X}_N$ is compact, and there exist two sets $\hat{\mathbb{X}}_N$ and $\mathscr{E}$ both containing the origin such that the following conditions hold: (i) $\hat{\mathbb{X}}_N \oplus \mathscr{E} \subseteq \mathbb{X}_N$, where $\oplus$ indicates the Minkowski sum; (ii) for each $\hat{x}(0) \in \hat{\mathbb{X}}_N$ and $\hat{e}(0) \in \mathscr{E}$, the solution of the extended closed-loop system* (3.4.2) *satisfies $\hat{x}(k) \in \mathbb{X}_N$ for all $k \geq 0$.*

Consider the sum of the two Lyapunov functions

$$W(\hat{x}, \mathbf{u}, e) = V(\hat{x}, \mathbf{u}) + J(e)$$

We next show that $W(\cdot)$ is a Lyapunov function for the perturbed system and establish exponential stability of the extended state origin $(\hat{x}, e) = (0, 0)$. From the definition of $W(\cdot)$ we have

$$\begin{aligned} a\,|(\hat{x}, \mathbf{u})|^2 + \bar{a}\,|e| \leq W(\hat{x}, \mathbf{u}, e) \leq b\,|(\hat{x}, \mathbf{u})|^2 + \bar{b}\,|e| \quad \Longrightarrow \\ \tilde{a}(|(\hat{x}, \mathbf{u})|^2 + |e|) \leq W(\hat{x}, \mathbf{u}, e) \leq \tilde{b}(|(\hat{x}, \mathbf{u})|^2 + |e|) \end{aligned} \tag{3.4.3}$$

in which $\tilde{a} = \min(a, \bar{a}) > 0$ and $\tilde{b} = \max(b, \bar{b}) > 0$. Next we compute the cost change

$$W(\hat{x}^+, \mathbf{u}^+, e^+) - W(\hat{x}, \mathbf{u}, e) = V(\hat{x}^+, \mathbf{u}^+) - V(\hat{x}, \mathbf{u}) + J(e^+) - J(e)$$

The Lyapunov function V is quadratic in $(\hat{x}, \mathbf{u})$ and, hence, Lipschitz continuous on bounded sets. By Proposition 3.15

$$\begin{aligned} \left|V(F(\hat{x}, \mathbf{u}) + Le, \mathbf{h}^+(e)) - V(F(\hat{x}, \mathbf{u}) + Le, \tilde{\mathbf{u}}^+)\right| \leq L_h L_{V_u}\,|e| \\ \left|V(F(\hat{x}, \mathbf{u}) + Le, \tilde{\mathbf{u}}^+) - V(F(\hat{x}, \mathbf{u}), \tilde{\mathbf{u}}^+)\right| \leq L_{V_x}\,|Le| \end{aligned}$$

in which L_h, L_{V_u}, and L_{V_x} are Lipschitz constants for $\mathbf{h}^+$ and the first and second arguments of V, respectively. Combining the above inequalities

$$\left|V(F(\hat{x}, \mathbf{u}) + Le, \mathbf{h}^+(e)) - V(F(\hat{x}, \mathbf{u}), \tilde{\mathbf{u}}^+)\right| \leq \bar{L}_V\,|e|$$

in which $\bar{L}_V = L_h L_{V_u} + L_{V_x} |L|$. Using the system evolution we have

$$V(\hat{x}^+, \mathbf{h}^+(e)) \leq V(F(\hat{x}, \mathbf{u}), \tilde{\mathbf{u}}^+) + \bar{L}_V |e|$$

and by Lemma 3.8

$$V(\hat{x}^+, \mathbf{u}^+) \leq V(F(\hat{x}, \mathbf{u}), \tilde{\mathbf{u}}^+) + \bar{L}_V |e|$$

Subtracting $V(\hat{x}, \mathbf{u})$ from both sides and noting that $\tilde{\mathbf{u}}^+$ is a stabilizing input sequence for $e = 0$ gives

$$\begin{aligned} V(\hat{x}^+, \mathbf{u}^+) - V(\hat{x}, \mathbf{u}) &\leq -c\,|(\hat{x}, u(0))|^2 + \bar{L}_V |e| \\ W(\hat{x}^+, \mathbf{u}^+, e^+) - W(\hat{x}, \mathbf{u}, e) &\leq -c\,|(\hat{x}, u(0))|^2 + \bar{L}_V |e| - \bar{c}\,|e| \\ &\leq -c\,|(\hat{x}, u(0))|^2 - (\bar{c} - \bar{L}_V)\,|e| \\ W(\hat{x}^+, \mathbf{u}^+, e^+) - W(\hat{x}, \mathbf{u}, e) &\leq -\tilde{c}(|(\hat{x}, u(0))|^2 + |e|) \end{aligned} \tag{3.4.4}$$

in which we choose $\bar{c} > \bar{L}_V$ and $\tilde{c} = \min(c, \bar{c} - \bar{L}_V) > 0$. This choice is possible because $\bar{c}$ can be chosen arbitrarily large. Notice this step is what motivated the choice of $\sigma = 1$. Lastly, we require the constraint

$$|\mathbf{u}| \leq d\,|\hat{x}|\,, \qquad \hat{x} \in \mathbb{B}_r \tag{3.4.5}$$

Theorem 3.17 (Exponential stability of perturbed system)**.** *Given Assumptions 3.5, 3.14, 3.16, for each $\hat{x}(0) \in \hat{\mathbb{X}}_N$ and $e(0) \in \mathscr{E}$, there exist constants $\alpha > 0$ and $0 < \gamma < 1$, such that the solution of the perturbed system* (3.4.2) *satisfies, for all $k \geq 0$*

$$|(\hat{x}(k), e(k)| \leq \alpha |(\hat{x}(0), e(0)| \gamma^k \tag{3.4.6}$$

Proof. Using the same arguments as for Lemma 3.2, we write:

$$W(\hat{x}^+, \mathbf{u}^+, e^+) - W(\hat{x}, \mathbf{u}, e) \leq -\hat{c}(|(\hat{x}, \mathbf{u})|^2 + |e|) \tag{3.4.7}$$

in which $\hat{c} \geq \tilde{c} > 0$. Therefore $W(\cdot)$ is a Lyapunov function for the extended state $(\hat{x}, \mathbf{u}, e)$ with mixed norm powers. The standard exponential stability argument can be extended for the

mixed norm power case to show that the origin of the extended closed-loop system (3.4.2) is exponentially stable (Rawlings and Mayne, 2009, p.420). Hence, for all $k \geq 0$

$$|(\hat{x}(k), \mathbf{u}(k), e(k))| \leq \tilde{\alpha}\,|(\hat{x}(0), \mathbf{u}(0), e(0))|\,\gamma^k$$

in which $\tilde{\alpha} > 0$ and $0 < \gamma < 1$. Notice that Assumption 3.16 implies that $\mathbf{u}(k)$ exists for all $k \geq 0$ because $\hat{x}(k) \in \mathbb{X}_N$.

We have, using the same arguments used in Lemma 3.2

$$\begin{aligned} |(\hat{x}(k), e(k))| &\leq |(\hat{x}(k), \mathbf{u}(k), e(k))| \leq \tilde{\alpha}\,|(\hat{x}(0), \mathbf{u}(0), e(0))|\,\gamma^k \\ &\leq \alpha\,|(\hat{x}(0), e(0))|\,\gamma^k \end{aligned}$$

in which $\alpha = \tilde{\alpha}(1 + \bar{d}) > 0$. □

Corollary 3.18. *Under the assumptions of Theorem 3.17, for each $x(0)$ and $\hat{x}(0)$ such that $e(0) = x(0) - \hat{x}(0) \in \mathscr{E}$ and $\hat{x}(0) \in \hat{\mathbb{X}}_N$, the solution of the closed-loop state $x(k) = \hat{x}(k) + e(k)$ satisfies:*

$$|x(k)| \leq \bar{\alpha}|x(0)|\gamma^k \tag{3.4.8}$$

for some $\bar{\alpha} > 0$ and $0 < \gamma < 1$.

Proof. We first note that: $|x(k)| \leq |\hat{x}(k)| + |e(k)| \leq \sqrt{2}|(\hat{x}(k), e(k))|$. From Theorem 3.17 we can write:

$$|x(k)| \leq \sqrt{2}\alpha\,|(\hat{x}(0), e(0))|\,\gamma^k \leq \bar{\alpha}\,|\hat{x}(0) + e(0)|\,\gamma^k$$

with $\bar{\alpha} = \sqrt{2}\alpha$, which concludes the proof by noticing that $x(0) = \hat{x}(0) + e(0)$. □

3.5 Coupled Constraints

In Remark 3.4, we commented that the constraint assumptions imply uncoupled constraints, because each input is constrained by a separate feasible region so that the full feasible space is defined $(\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{U}^N = \mathbb{U}_1^N \times \mathbb{U}_2^N$. This assumption, however, is not always practical. Consider two subsystems sharing a scarce resource for which we control the distribution. There then exists an availability constraint spanning the subsystems. This constraint is *coupled* because each local resource constraint depends upon the amount requested by the other subsystem.

Remark 3.19. *For plants with coupled constraints, implementing MPC problem (3.3.6) gives exponentially stable, yet suboptimal, feedback.*

In this section, we relax the assumption so that $(\mathbf{u}_1, \mathbf{u}_2) \in \mathbb{U}^N$ for any $\mathbb{U}$ compact, convex and containing the origin in its interior. Consider the decomposition of the inputs $\mathbf{u} = (\mathbf{u}_{U1}, \mathbf{u}_{U2}, \mathbf{u}_C)$ such that there exists a $\mathbb{U}_{U1}$,$\mathbb{U}_{U2}$, and $\mathbb{U}_C$ for which

$$\mathbb{U} = \mathbb{U}_{U1} \times \mathbb{U}_{U2} \times \mathbb{U}_C$$

and

$$\mathbf{u}_{U1} \in \mathbb{U}_{U1}^N, \quad \mathbf{u}_{U2} \in \mathbb{U}_{U2}^N, \quad \mathbf{u}_C \in \mathbb{U}_C^N$$

for which $\mathbb{U}_{U1}$, $\mathbb{U}_{U2}$, and $\mathbb{U}_C$ are compact and convex. We denote $\mathbf{u}_{Ui}$ the uncoupled inputs for subsystem i, $i \in \mathbb{I}_{1:2}$, and $\mathbf{u}_C$ the coupled inputs.

Remark 3.20. $\mathbb{U}_{U1}$,$\mathbb{U}_{U2}$, *or* $\mathbb{U}_C$ *may be empty, and therefore such a decomposition always exists.*

We modify the cooperative MPC problem (3.3.6) for the above decomposition. Define the augmented inputs $(\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2)$

$$\hat{\mathbf{u}}_1 = \begin{bmatrix} \mathbf{u}_{U1} \\ \mathbf{u}_C \end{bmatrix} \quad \hat{\mathbf{u}}_2 = \begin{bmatrix} \mathbf{u}_{U2} \\ \mathbf{u}_C \end{bmatrix}$$

The implemented inputs are

$$\mathbf{u}_1 = \hat{E}_1 \hat{\mathbf{u}}_1 \quad \mathbf{u}_2 = \hat{E}_2 \hat{\mathbf{u}}_2, \quad \hat{E}_1 = \begin{bmatrix} I & \\ & I_1 \end{bmatrix} \quad \hat{E}_2 = \begin{bmatrix} I & \\ & I_2 \end{bmatrix}$$

in which (I_1, I_2) are diagonal matrices with either 0 or 1 diagonal entries and satisfy $I_1 + I_2 = I$. For simplicity, we summarize the previous relations as $\mathbf{u} = \hat{E}\hat{\mathbf{u}}$ with $\hat{E} = \text{diag}(\hat{E}_1, \hat{E}_2)$. The objective function is

$$\hat{V}(x_1(0), x_2(0), \hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2) = V(x_1(0), x_2(0), \hat{E}_1\hat{\mathbf{u}}_1, \hat{E}_2\hat{\mathbf{u}}_2) \tag{3.5.1}$$

We solve the augmented cooperative MPC problem for $i \in \mathbb{I}_{1:2}$

$$\min_{\hat{\boldsymbol{v}}_i} \hat{V}(x_1(0), x_2(0), \hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2) \tag{3.5.2a}$$

subject to

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \bar{B}_{11} \\ \bar{B}_{21} \end{bmatrix} \hat{E}_1 \hat{v}_1 + \begin{bmatrix} \bar{B}_{12} \\ \bar{B}_{22} \end{bmatrix} \hat{E}_2 \hat{v}_2 \tag{3.5.2b}$$

$$\hat{\boldsymbol{v}}_i \in \mathbb{U}_{Ui}^N \times \mathbb{U}_C^N \tag{3.5.2c}$$

$$S_{ji}^{u\,\prime} x_{ji}(N) = 0 \quad j \in \mathbb{I}_{1:2} \tag{3.5.2d}$$

$$|\hat{\boldsymbol{v}}_i| \le d_i |x_i(0)| \quad \text{if } x_i(0) \in \mathbb{B}_r \tag{3.5.2e}$$

$$\hat{\boldsymbol{v}}_j = \hat{\boldsymbol{v}}_j^p \quad j \in \mathbb{I}_{1:2} \setminus i \tag{3.5.2f}$$

The update (3.3.7) is used to determine the next iterate.

Lemma 3.21. *As $p \to \infty$ the cost $\hat{V}(x(0), \hat{\boldsymbol{v}}^p)$ converges to the optimal value $V^0(x(0))$, and the iterates $(\hat{E}_1 \hat{\boldsymbol{v}}_1^p, \hat{E}_2 \hat{\boldsymbol{v}}_2^p)$ converge to the Pareto optimal centralized solution $\mathbf{u}^0 = (\mathbf{u}_1^0, \mathbf{u}_2^0)$.*

Proof. Because $\hat{V}(\cdot)$ is convex and bounded below, the proof follows from Lemma 3.9 and from noticing that the point $\mathbf{u}^0 = (\hat{E}_1 \hat{\mathbf{u}}_1^0, \hat{E}_2 \hat{\mathbf{u}}_2^0)$, with $\hat{\mathbf{u}}_i^0 = \lim_{p\to\infty} \hat{\boldsymbol{v}}_i, \quad i \in \mathbb{I}_{1:2}$, is Pareto optimal. □

Therefore, problem (3.5.2) gives optimal feedback and may be used for plants with coupled constraints.

3.6 M Subsystems

In this section, we show that the stability theory of cooperative control extends to any finite number of subsystems. For $M > 0$ subsystems, the plantwide variables are defined

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_M \end{bmatrix} \quad B_i = \begin{bmatrix} \bar{B}_{1i} \\ \bar{B}_{2i} \\ \vdots \\ \bar{B}_{Mi} \end{bmatrix} \quad \forall\, i \in \mathbb{I}_{1:M}$$

$$V(x, \mathbf{u}) = \sum_{i \in \mathbb{I}_{1:M}} \rho_i V_i(x_i, \mathbf{u}_i) \quad A = \operatorname{diag}(A_1, \ldots, A_M)$$

Each subsystem solves the optimization

$$\min_{\boldsymbol{v}_i} V(x(0), \boldsymbol{v})$$

subject to

$$x^+ = Ax + \sum_{i \in \mathbb{I}_{1:M}} B_i v_i$$

$$\boldsymbol{v}_i \in \mathbb{U}_i^N$$

$$S_{ji}^{u\,\prime} x_{ji}(N) = 0 \quad j \in \mathbb{I}_{1:M}$$

$$|\boldsymbol{v}_i| \le d_i \sum_{j \in \mathbb{I}_{1:M}} \left| x_{ji}(0) \right| \quad \text{if } x_{ji}(0) \in \mathbb{B}_r \quad j \in \mathbb{I}_{1:M}$$

$$\boldsymbol{v}_j = \boldsymbol{v}_j^p \quad j \in \mathbb{I}_{1:M} \setminus i$$

The controller iteration is given by

$$\boldsymbol{v}^{p+1} = \sum_{i \in \mathbb{I}_{1:M}} w_i(\boldsymbol{v}_1^p, \ldots, \boldsymbol{v}_i^*, \ldots, \boldsymbol{v}_M^p)$$

in which $\boldsymbol{v}_i^* = \boldsymbol{v}_i^*\left(x(0); \boldsymbol{v}_j^p, j \in \mathbb{I}_{1:M} \setminus i\right)$. After $\bar{p}$ iterates, we set $\mathbf{u} \leftarrow \boldsymbol{v}^{\bar{p}}$ and inject $u(0)$ into the plant.

The warm start is generated by purely local information

$$\tilde{\mathbf{u}}_i^+ = \{u_i(1), u_i(2), \ldots, u_i(N-1), 0\} \quad \forall\, i \in \mathbb{I}_{1:M}$$

The plantwide cost function then satisfies for any $\bar{p} \geq 0$

$$V(x^+, \mathbf{u}^+) \leq V(x, \mathbf{u}) - \sum_{i \in \mathbb{I}_{1:M}} \rho_i \ell_i(x_i, u_i)$$

$$|\mathbf{u}| \leq d\,|x| \quad x \in \mathbb{B}_r$$

Generalizing Assumption 3.5 to all $i \in \mathbb{I}_{1:M}$, we find that Theorem 3.13 applies and cooperative MPC of M subsystems is exponentially stable.

Moreover, expressing the M subsystem outputs as

$$y_i = \sum_{j \in \mathbb{I}_{1:M}} C_{ij} x_{ij} \quad i \in \mathbb{I}_{1:M}$$

and generalizing Assumption 3.14 for $i \in \mathbb{I}_{1:M}$, cooperative MPC for M subsystems satisfies Theorem 3.17. Finally, for systems with coupled constraints, we can decompose the feasible space such that $\mathbb{U} = (\prod_{i \in \mathbb{I}_{1:M}} \mathbb{U}_{Ui}) \times \mathbb{U}_C$. Hence, the input augmentation scheme of Section 3.5 is applicable to plants of M subsystems. Notice that, in general, this approach may lead to augmented inputs for each subsystem that are larger than strictly necessary to achieve optimal control. The most parsimonious augmentation scheme is described elsewhere (Pannocchia et al., 2009).

3.7 Distributed Control Example

In this example, we compare the distributed control strategies of noncooperative and cooperative control with decentralized and centralized control. Consider a distillation column for separating methanol from water. A two input/two output model for LV control of the column was proposed by Wood and Berry (1973). The reported transfer function matrix for the process is

$$\begin{pmatrix} x_D(s) \\ x_B(s) \end{pmatrix} = \begin{pmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21.0s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{pmatrix} \begin{pmatrix} R(s) \\ S(s) \end{pmatrix}$$

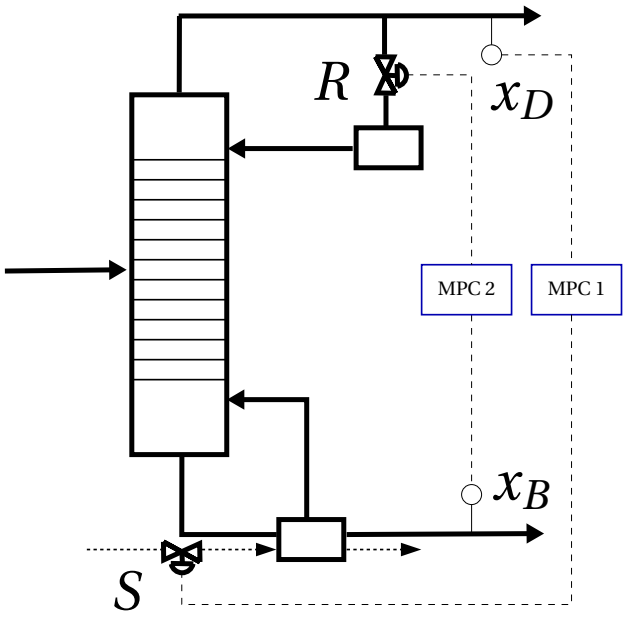


Figure 3.2: LV control of distillation column with bad pairing.

in which

x_D – overhead mole fraction methanol

x_B – bottoms mole fraction methanol

R – overhead reflux flow rate

S – bottoms steam flow rate

Using RGA, the recommended pairing is to control x_D with R and x_B with S. As in Figure 3.2, however, we intentionally choose the poor pairing and use the subsystems

$$y_1 = x_D \quad y_2 = x_B$$

$$u_1 = S \quad u_2 = R$$

We use the sampling time of 1 minute, and the controllers are tuned with the following parameters

$$Q_{y1} = 10 \quad Q_{y2} = 100 \qquad Q_i = C_i' Q_{yi} C_i + 0.001 I \quad \forall i \in \mathbb{I}_{1:2} \qquad R_1 = 0.1 \quad R_2 = 0.1 \qquad N = 10$$

Table 3.1: Performance comparison. Noncooperative and decentralized control are not able to reject the disturbance.

	Cost	Performance loss (%)
Centralized MPC	75.8	0
Cooperative MPC (10 iterates)	76.1	0.388
Cooperative MPC (1 iterate)	87.5	15.4
Noncooperative MPC	382	404
Decentralized MPC	364	380

The reflux flow is constrained such that $0 \leq R \leq 0.15$.

We simulate the response of the plant to an output disturbance. At 5 minutes a disturbance of 0.5 enters the bottoms mole fraction. As shown in Figure 3.3, noncooperative and decentralized control are not able to reject the disturbance, and, in this example, noncooperative control performs worse than decentralized control. Even given the poor pairing, cooperative control is robust the poor pairing and is able to successfully reject the disturbance. Performance improves as the number of optimization iterates increases (see Table 3.1).

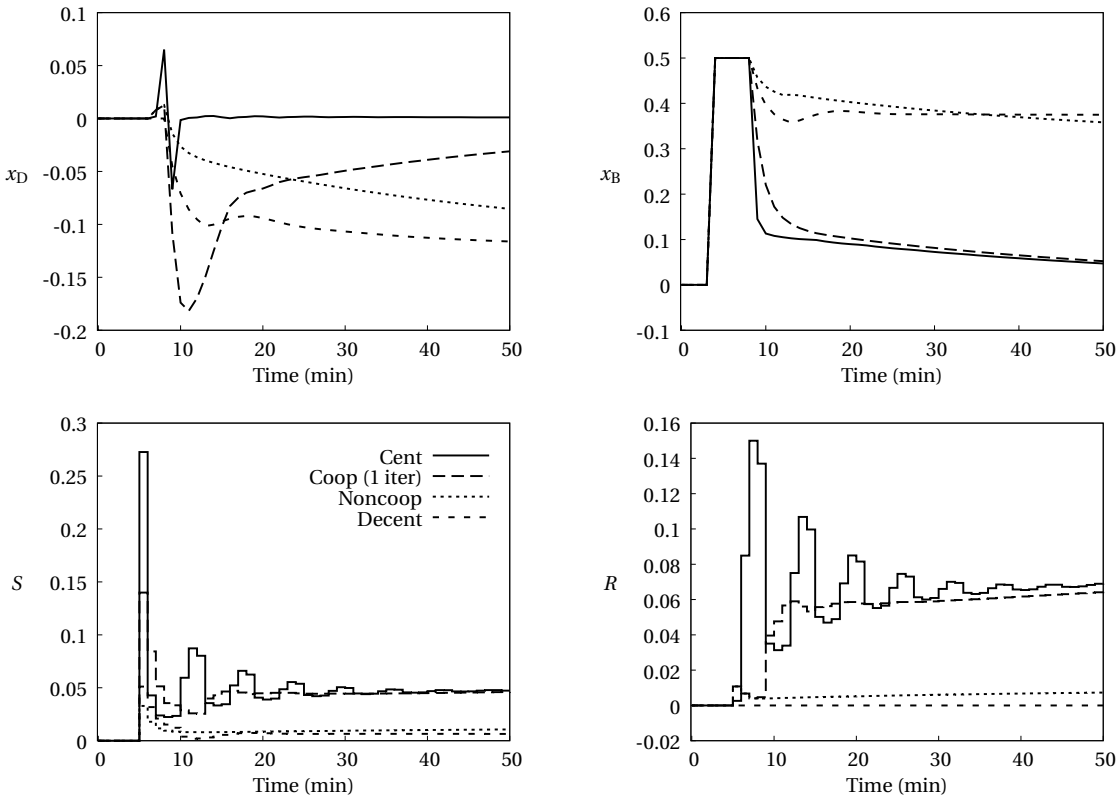


Figure 3.3: Disturbance rejection example.

Chapter 4

Hierarchical Cooperative Control

Note: With the exception of Sections 4.6 and 4.7, the text of this chapter appears in (Stewart, Rawlings, and Wright, 2010a).

4.1 Introduction

Chemical plants comprise many interacting subsystems connected through a network of material, energy, and information streams. Interactions propagate through the network and all subsystems affect each other. One useful characterization of this network is the density of the inter-subsystem connections. Densely interconnected plants are complex and the interactions quickly spread throughout the network. Commonly, however, there exists some sparsity in the plant interconnections. In sparsely connected plants, local interactions propagate quickly in groups of subsystems and affect the rest of the plant only on a longer time scale. For these plants, a network topology naturally arises in which subsystems are grouped into neighborhoods. Neighborhoods usually correspond to processing areas, in which several processes create raw materials for another set of subsystems. Performance in these plants is often improved using a hierarchy of controllers. Each neighborhood has a coordinating controller, which is itself coordinated with the other neighborhoods by a central plantwide controller (Scattolini, 2009). Hierarchical control has the advantage of maintaining independence of the subsystems and neighborhoods, yet designing the hierarchical control scheme is often difficult. Model and objective mismatch occurs because the control is separated into layers.

We have shown previously in Chapter 3 that cooperative control provides plantwide stability without a coordinator, but did not consider the underlying topology of the plant. The control scheme requires all subsystems to be synchronous and to communicate with all other subsystems at every optimization iterate. These requirements limit the application of cooperative control to plants in which information can be reliably exchanged plantwide, and they do not allow integration of subsystems with multiple time scales. In this chapter, we weaken these requirements by grouping the subsystems into hierarchies and allowing the subsystems to optimize and exchange information on *any* time scale. Our main contribution is to achieve the advantages of hierarchical control schemes without requiring additional coordinating controllers.

We begin in the next section with an introduction to the distributed optimization used for hierarchical cooperative control. In Section 4.3, we provide definitions, then show the control algorithm properties in Section 4.4. To simplify the exposition, we introduce these concepts with only four subsystems, but point out that these results extend to any number of subsystems. In Section 4.5 we provide a physical definition of neighborhoods, which is utilized to reduce communication in the plant. We conclude with an example showing performance of a chemical plant under multiple communication schedules.

4.2 Hierarchical Optimization

Consider the optimization

$$\min_{u} V(u_1, u_2, u_3, u_4) \tag{4.2.1}$$

in which $u_i \in \mathbb{R}$ and $V : \mathbb{R}^4 \to \mathbb{R}_+$ is strictly convex. The optimization is performed as a Jacobi iteration (Bertsekas and Tsitsiklis, 1997, pp.219-223), in which there is a suboptimization performed for each variable and the solutions are traded between optimizers. We create a two-level hierarchy and define the neighborhoods $\mathbb{N}_1 = \{1,2\}$ and $\mathbb{N}_3 = \{3,4\}$ (see Fig. 4.1). At inner iterates, the suboptimizations utilize the latest iterates from variables within their own neighborhood, while the other variables are updated at outer iterates. Consider the optimization

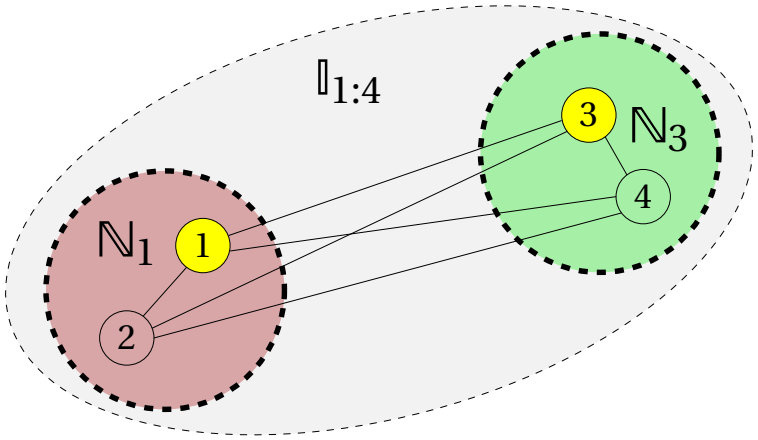


Figure 4.1: Two level hierarchy: two neighborhoods, each with two subsystems such that $\mathbb{N}_1 = \{1,2\}$ and $\mathbb{N}_3 = \{3,4\}$. The set of leaders $\mathbb{L} = \{1,3\}$.

between outer iterates t and $t+1$. Initializing $v^0 \leftarrow u^t = (u_1^t, u_2^t, u_3^t, u_4^t)$, at iterate $p \in \mathbb{I}_{0:\bar{p}-1}$

$$v_1^* = \arg\min_{v_1} V(v_1, v_2^p, u_3^t, u_4^t) \qquad v_3^* = \arg\min_{v_3} V(u_1^t, u_2^t, v_3, v_4^p)$$

$$v_2^* = \arg\min_{v_2} V(v_1^p, v_2, u_3^t, u_4^t) \qquad v_4^* = \arg\min_{v_4} V(u_1^t, u_2^t, v_3^p, v_4)$$

Inner iterates are generated by the convex step

$$v_i^{p+1} = w_i v_i^* + (1 - w_i) v_i^p \quad \forall i \in \mathbb{I}_{1:4}$$

in which for $l \in \mathbb{L} = \{1,3\}, \sum_{j \in \mathbb{N}_l} w_j = 1,\ w_i > 0\ \forall i \in \mathbb{I}_{1:4}$. We clarify the definition the leader set $\mathbb{L}$ in the sequel. After $\bar{p}$ iterates, the outer iterate is performed

$$u_i^{t+1} = \lambda_i v_i^{\bar{p}} + (1 - \lambda_i) u_i^t \quad \forall i \in \mathbb{I}_{1:4}$$

in which the weighting λ_i is defined for each neighborhood $\sum_{l \in \mathbb{L}} \lambda_l = 1,\ \lambda_i = \lambda_j\ \forall i, j \in \mathbb{N}_l,\ \forall l \in \mathbb{L}$.

Lemma 4.1 (Convergence)**.** *The cost $V(u_1^t, u_2^t, u_3^t, u_4^t)$ is nonincreasing at each outer iterate t and converges as $t \to \infty$.*

Proof of Lemma 4.1.

$V(u_1^{t+1}, u_2^{t+1}, u_3^{t+1}, u_4^{t+1}) =$

$$V\big(\lambda_1 v_1^{\bar{p}} + (1-\lambda_1) u_1^t, \lambda_2 v_2^{\bar{p}} + (1-\lambda_2) u_2^t, \lambda_3 v_3^{\bar{p}} + (1-\lambda_3) u_3^t, \lambda_4 v_4^{\bar{p}} + (1-\lambda_4) u_4^t\big)$$

Noticing $\lambda_1 = \lambda_2$, $\lambda_3 = \lambda_4$, and $\lambda_1 + \lambda_3 = 1$

$$
\begin{aligned}
&= V\big(\lambda_1 v_1^{\bar{p}} + (1-\lambda_1)u_1^t, \lambda_1 v_2^{\bar{p}} + (1-\lambda_1)u_2^t, \lambda_3 v_3^{\bar{p}} + (1-\lambda_3)u_3^t, \lambda_3 v_4^{\bar{p}} + (1-\lambda_3)u_4^t\big) \\
&\leq \lambda_1 V\big(v_1^{\bar{p}}, v_2^{\bar{p}}, u_3^t, u_4^t\big) + (1-\lambda_1)V(u_1^t, u_2^t, v_3^{\bar{p}}, v_4^{\bar{p}}\big) \\
&\leq \lambda_1 V\big(w_1 v_1^* + (1-w_1)v_1^{\bar{p}-1}, w_2 v_2^* + (1-w_2)v_2^{\bar{p}-1}, u_3^t, u_4^t\big) + \\
&\quad (1-\lambda_1)V(u_1^t, u_2^t, w_3 v_3^* + (1-w_3)v_3^{\bar{p}-1}, w_4 v_4^* + (1-w_4)v_4^{\bar{p}-1}\big)
\end{aligned}
$$

Noticing $w_1 = 1 - w_2$ and $w_3 = 1 - w_4$

$$
\begin{aligned}
&\leq \lambda_1 V\big(w_1 v_1^* + (1-w_1)v_1^{\bar{p}-1}, (1-w_1)v_2^* + w_1 v_2^{\bar{p}-1}, u_3^t, u_4^t\big) + \\
&\quad (1-\lambda_1)V(u_1^t, u_2^t, w_3 v_3^* + (1-w_3)v_3^{\bar{p}-1}, (1-w_3)v_4^* + w_3 v_4^{\bar{p}-1}\big) \\
&\leq \lambda_1\Big(w_1 V(v_1^*, v_2^{\bar{p}-1}, u_3^t, u_4^t) + (1-w_1)V(v_1^{\bar{p}-1}, v_2^*, u_3^t, u_4^t)\Big) + \\
&\quad (1-\lambda_1)\Big(w_3 V(u_1^t, u_2^t, v_3^*, v_4^{\bar{p}-1}) + (1-w_3)V(u_1^t, u_2^t, v_3^{\bar{p}-1}, v_4^*)\Big) \\
&\leq \lambda_1\Big(w_1 V(v_1^{\bar{p}-1}, v_2^{\bar{p}-1}, u_3^t, u_4^t) + (1-w_1)V(v_1^{\bar{p}-1}, v_2^{\bar{p}-1}, u_3^t, u_4^t)\Big) + \\
&\quad (1-\lambda_1)\Big(w_3 V(u_1^t, u_2^t, v_3^{\bar{p}-1}, v_4^{\bar{p}-1}) + (1-w_3)V(u_1^t, u_2^t, v_3^{\bar{p}-1}, v_4^{\bar{p}-1})\Big) \\
&\leq \lambda_1\Big(V(v_1^{\bar{p}-1}, v_2^{\bar{p}-1}, u_3^t, u_4^t)\Big) + (1-\lambda_1)\Big(V(u_1^t, u_2^t, v_3^{\bar{p}-1}, v_4^{\bar{p}-1})\Big)
\end{aligned}
$$

Continuing by induction

$$
\leq \lambda_1 V\big(v_1^0, v_2^0, u_3^t, u_4^t\big) + (1-\lambda_1)V(u_1^t, u_2^t, v_3^0, v_4^0\big)
$$

Because $v_i^0 = u_i^t, \forall i \in \mathbb{I}_{1:4}$

$$
= V\big(u_1^t, u_2^t, u_3^t, u_4^t\big)
$$

Consolidating, this gives the relationship

$$
V(u_1^{t+1}, u_2^{t+1}, u_3^{t+1}, u_4^{t+1}) \leq V\big(u_1^t, u_2^t, u_3^t, u_4^t\big) \tag{4.2.2}
$$

□

Lemma 4.2 (Optimality)**.** *The cost* $V(u_1^t, u_2^t, u_3^t, u_4^t)$ *converges to the optimal value* $V(u_1^*, u_2^*, u_3^*, u_4^*)$*, and the iterates* $(u_1^t, u_2^t, u_3^t, u_4^t)$ *converge to the optimizer* $(u_1^*, u_2^*, u_3^*, u_4^*)$ *as* $t \to \infty$*.*

The proof of Lemma 4.2 is equivalent to Lemma 3.9 and is omitted here.

4.2.1 Example

To illustrate the cost decrease at each outer iterate, consider Fig. 4.2. In Fig. 4.2a, the optimizers in neighborhood 1 seek a solution to the optimization problem $\min_{v_1,v_2} V(v_1,v_2,u_3^0,u_4^0)$ and after taking $\bar{p}=2$ steps, arrive at the suboptimal point $\bar{v}_1^2=(v_1^2,v_2^2)$. Similarly in neighborhood 3, the suboptimal point $\bar{v}_3^2=(v_3^2,v_4^2)$ is found (see Fig. 4.2b). These two points are used to generate the next outer iterate u^1. We show this convex step in Fig. 4.2c, for which the x-axis and y-axis are, respectively, a representation of (u_1,u_2) and (u_3,u_4). Notice that $V(u^1)\leq V(u^0)$.

4.3 Preliminaries

We now provide definitions required to show properties of the control algorithm in the next section. For simplicity, we assume a plant composed of only four subsystems and relax this assumption in the sequel.

4.3.1 Models

Consider the decentralized linear model

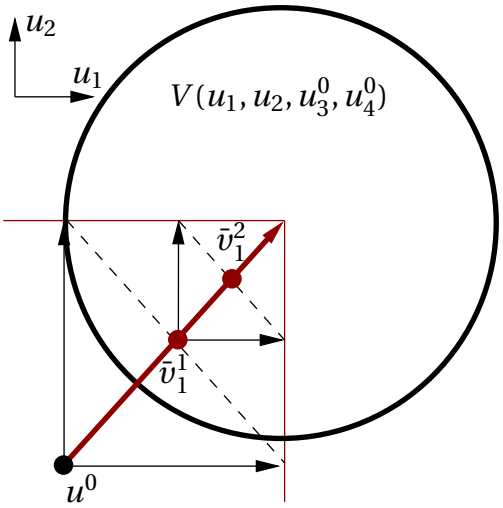
$$x_{ij}^+ = A_{ij}x_{ij} + B_{ij}u_j \quad \forall (i,j)\in \mathbb{I}_{1:4}\times\mathbb{I}_{1:4} \tag{4.3.1}$$

in which $x_{ij}\in\mathbb{R}^{n_{ij}}$, $u_j\in\mathbb{R}^{m_j}$, $A_{ij}\in\mathbb{R}^{n_{ij}\times n_{ij}}$, and $B_{ij}\in\mathbb{R}^{n_{ij}\times m_j}$. This model captures the effect from the inputs of subsystem $j\in\mathbb{I}_{1:4}$ on the states of subsystem $i\in\mathbb{I}_{1:4}$. For notational simplicity, we collect the states to form the subsystem model for each $i\in\mathbb{I}_{1:4}$

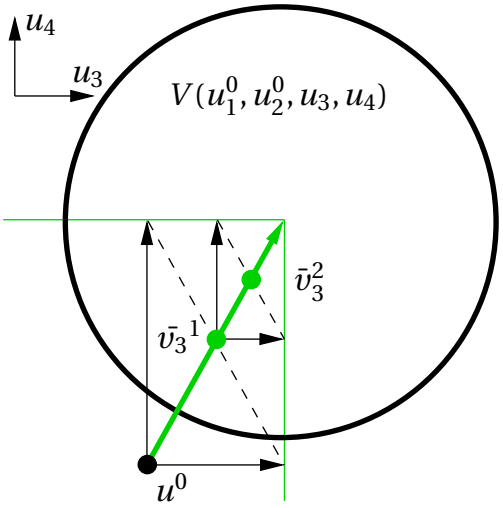
$$x_i^+ = A_i x_i + \sum_{j\in\mathbb{I}_{1:4}} \bar{B}_{ij}u_j$$

in which $x_i=[x_{i1}'\cdots x_{i4}']'\in\mathbb{R}^{n_i}$, $n_i=\sum_{j\in\mathbb{I}_{1:4}} n_{ij}$, $A_i=\operatorname{diag}(A_{i1},\ldots,A_{i4})\in\mathbb{R}^{n_i\times n_i}$ and

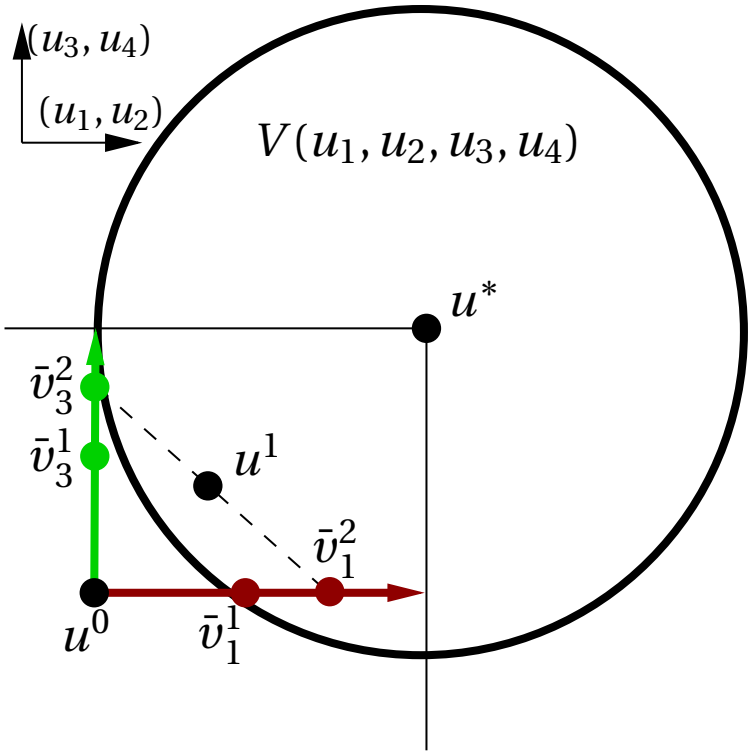
$$\bar{B}_{ij}=[0\cdots \underbrace{B_{ij}'}_{j\text{th position}}\cdots 0]'\in\mathbb{R}^{n_i\times m_j}$$



(a) Neighborhood 1



(b) Neighborhood 3



(c) Overall

Figure 4.2: Hierarchical optimization. Figures (a) and (b) show each neighborhood's 2-dimensional optimization. Figure (c) is a representation of the overall 4-dimensional optimization.

4.3.2 Objective Functions

For each $i \in \mathbb{I}_{1:4}$ define the quadratic stage cost and terminal penalty, respectively, as $\ell_i(x_i, u_i) = \frac{1}{2}(x_i' Q_i x_i + u_i' R_i u_i)$ and $V_{if}(x_i) = \frac{1}{2} x_i' P_{if} x_i$ in which $Q_i \in \mathbb{R}^{n_i \times n_i}$, $R_i \in \mathbb{R}^{m_i \times m_i}$, and $P_{if} \in \mathbb{R}^{n_i \times n_i}$. For subsystem $i \in \mathbb{I}_{1:4}$, the objective function is

$$V_i(x_i(0), \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = \sum_{k=0}^{N-1} \ell_i(x_i(k), u_i(k)) + V_{if}(x_i(N))$$

in which $\mathbf{u}_i = [u_i(0)', \ldots, u_i(N-1)']' \in \mathbb{R}^{Nm_i}$ and $N > 0$ is the control horizon. We define the plantwide objective function

$$V(x(0), \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = \sum_{i \in \mathbb{I}_{1:4}} \rho_i V_i(x_i(0), \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$$

in which $x = [x_1', \ldots, x_4']'$ and $\rho_i > 0$ for all $i \in \mathbb{I}_{1:4}$.

4.3.3 Constraints

At each time step, the inputs satisfy the constraints $u_i(k) \in \mathbb{U}_i, \forall k \in \mathbb{I}_{0:N-1}, \forall i \in \mathbb{I}_{1:4}$ in which $\mathbb{U}_i$ is compact, convex, and contains the origin in its interior.

4.3.4 Terminal Penalties and Terminal Constraints

Consider the real Schur decomposition

$$A_{ij} = \begin{bmatrix} S^s_{ij} & S^u_{ij} \end{bmatrix} \begin{bmatrix} A^s_{ij} & \\ & A^u_{ij} \end{bmatrix} \begin{bmatrix} S^{s\,\prime}_{ij} \\ S^{u\,\prime}_{ij} \end{bmatrix}$$

in which A^s_{ij} and A^u_{ij} are composed of the stable and unstable blocks of A_{ij}, respectively. Defining $S^s_i = \mathrm{diag}(S^s_{i1}, \ldots, S^s_{i4})$ and $A^s_i = \mathrm{diag}(A^s_{i1}, \ldots, A^s_{i4})$ for each $i \in \mathbb{I}_{1:4}$, define

$$P_{if} = S^s_i \Sigma_i S^{s\prime}_i \tag{4.3.2}$$

in which Σ_i satisfies the Lyapunov equation $A^{s\prime}_i \Sigma_i A^s_i - \Sigma_i = -S^{s\prime}_i Q_i S^s_i$. We set the unstable modes of A_{ij} to zero at the end of the horizon with the constraint $S^{u\,\prime}_{ij} x_{ij}(N) = 0$.

4.4 Control Algorithm

We now apply the optimization presented in Section 4.2 to distributed MPC. Consider the case of two neighborhoods, each with two subsystems, i.e., $\mathbb{N}_1 = \{1,2\}$ and $\mathbb{N}_3 = \{3,4\}$ (see Fig. 4.1). With a slight abuse of notation we refer to $\mathbb{N}_2 \leftarrow \mathbb{N}_1$ and $\mathbb{N}_4 \leftarrow \mathbb{N}_3$. We define a two-level hierarchy in which the subsystems in neighborhood $\mathbb{N}_1$ communicate with each other at every time step, but for which they communicate with the subsystems in neighborhood $\mathbb{N}_3$ only after N_s steps. For example, if all subsystems perform an outer iterate after every time step, then $N_s = 0$. As in Section 4.2, we re-optimize this trajectory multiple times between outer iterates, but here only N_o input steps are optimized at a time.

4.4.1 Initialization

For plant initialization, we require an initial condition $\tilde{\mathbf{u}}_i$ for each subsystem $i \in \mathbb{I}_{1:4}$

$$\tilde{\mathbf{u}}_i = [\underbrace{\tilde{u}_i(0)', \tilde{u}_i(1)', \ldots, \tilde{u}_i(N_o-1)'}_{N_o}, \underbrace{0, \ldots, 0}_{N_s}]' \tag{4.4.1}$$

for which $\tilde{\mathbf{u}}_i \in \mathbb{U}_i^N$ and the first N_o inputs satisfy the terminal constraint ${S_{ji}^u}' x_{ji}(N_o) = 0$ for all $j \in \mathbb{I}_{1:4}$.

4.4.2 Control Problem

Without loss of generality, let the outer iterate $t = 0$. We initialize $\boldsymbol{v}_i^0 \leftarrow \mathbf{u}_i^0 \leftarrow \tilde{\mathbf{u}}_i$. Between outer iterates, for $k \in \mathbb{I}_{0:N_s}$, subsystem i for every $i \in \mathbb{I}_{1:4}$ solves the following optimization at inner iterate $p \in \mathbb{I}_{k\bar{p}:(k+1)\bar{p}-1}$

$$\min_{\boldsymbol{v}_i} V(x(0), \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4) \tag{4.4.2a}$$

subject to

$$x_i^+ = A_i x_i + \sum_{j \in \mathbb{I}_{1:4}} \bar{B}_{ij} v_j \tag{4.4.2b}$$

$$\boldsymbol{v}_i \in \mathbb{U}_i^N \tag{4.4.2c}$$

$$S_{ji}^{u\,\prime} x_{ji}(k+N_o) = 0 \quad j \in \mathbb{I}_{1:4} \tag{4.4.2d}$$

$$|\boldsymbol{v}_i| \leq d_i \sum_{j \in \mathbb{I}_{1:4}} \left| x_{ji}(0) \right| \quad \text{if } x_{ji}(0) \in \mathbb{B}_r \quad j \in \mathbb{I}_{1:4} \tag{4.4.2e}$$

$$\boldsymbol{v}_j = \boldsymbol{v}_j^p \quad j \in \mathbb{N}_i \setminus i \tag{4.4.2f}$$

$$\boldsymbol{v}_l = \boldsymbol{v}_l^0 \quad l \in \mathbb{I}_{1:4} \setminus \mathbb{N}_i \tag{4.4.2g}$$

$$v_i(\tau) = v_i^p(\tau) \quad \forall \tau \in \mathbb{I}_{0:k-1} \cup \mathbb{I}_{k+N_o:N-1} \tag{4.4.2h}$$

in which the radius $r > 0$ of $\mathbb{B}_r$ is chosen as small as desired.[1] Notice that although the horizon of inputs is N, only N_o of the inputs are decision variables at any time k due to constraint (4.4.2h). We denote the solution to these optimizations as $\boldsymbol{v}_i^* = \boldsymbol{v}_i^*(x(0); \boldsymbol{v}_j^p; \boldsymbol{v}_l^0)\ \forall\, j \in \mathbb{N}_i \setminus i, \forall\, l \in \mathbb{I}_{1:4} \setminus \mathbb{N}_i$. The next inner iterate is defined by

$$\boldsymbol{v}_i^{p+1} = w_i \boldsymbol{v}_i^* + (1 - w_i)\boldsymbol{v}_i^p \tag{4.4.3}$$

in which $\sum_{j \in \mathbb{N}_i} w_j = 1$ and $w_i > 0$.

At time $k \in \mathbb{I}_{0:N_s}$ each subsystem $i \in \mathbb{I}_{1:4}$ takes $\bar{p} > 0$ inner iterates and arrives at the point $\boldsymbol{v}_i^{(k+1)\bar{p}}$. Each subsystem $i \in \mathbb{I}_{1:4}$ computes

$$\mathbf{u}_i^k = \lambda_i \boldsymbol{v}_i^{(k+1)\bar{p}} + (1 - \lambda_i)\tilde{\mathbf{u}}_i$$

in which $\sum_{j \in \mathbb{L}} \lambda_j = 1, \lambda_i = \lambda_j\ \forall\, j \in \mathbb{N}_i$, and $\mathbb{L}$ is the set of all leaders, i.e., $\mathbb{L} = \{1, 3\}$. Each subsystem injects the input

$$u_i(k) = u_i^k(k) = \lambda_i v_i^{(k+1)\bar{p}}(k) + (1 - \lambda_i) v_i^0(k) \tag{4.4.4}$$

in which $u_i^k(k)$ is the kth component of $\mathbf{u}_i^k$.

[1] A detailed discussion of this constraint is provided in Section 3.2.

Remark 4.3 (Weighting)**.** *Whereas w_i measures the relative weight of a subsystem in $\mathbb{N}_i$, λ_i measures the relative importance of neighborhood $\mathbb{N}_i$ in the plantwide topology.*

Remark 4.4. *Between outer iterates, each subsystem input is re-optimized $(N_s+1)\bar{p}$ times.*

Lemma 4.5 (Feasibility)**.** *The input trajectories satisfy $(\mathbf{u}_1^k, \mathbf{u}_2^k, \mathbf{u}_3^k, \mathbf{u}_4^k) \in \mathbb{U}_1^N \times \mathbb{U}_2^N \times \mathbb{U}_3^N \times \mathbb{U}_4^N$ for all $k \geq 0$.*

Proposition 4.6 (Terminal constraint feasibility)**.** *Given an input trajectory*

$$\mathbf{u}_i = [u_i(0)', \ldots, u_i(k+N_o-1)', 0, \ldots, 0]'$$

satisfying the constraint ${S_{ji}^u}' x_{ji}(\tau + N_o) = 0 \ \forall\, j \in \mathbb{I}_{1:4}$ for $\tau = k$, the constraint is satisfied for all $\tau \in \mathbb{I}_{k:N_s}$.

The plant moves to the next time step, using $\boldsymbol{\nu}_i^{(k+1)\bar{p}}$ as the initial condition for control problem (4.4.2) at time $k+1$. The point $\boldsymbol{\nu}_i^{(k+1)\bar{p}}$ is feasible by Lemma 4.5 and Proposition 4.6.

Let $\tilde{\mathbf{u}}^{N_s+} = (\tilde{\mathbf{u}}_1^{N_s+}, \tilde{\mathbf{u}}_2^{N_s+}, \tilde{\mathbf{u}}_3^{N_s+}, \tilde{\mathbf{u}}_4^{N_s+})$ denote the warm start, in which for $i \in \mathbb{I}_{1:4}$

$$\tilde{\mathbf{u}}_i^{N_s+} = [\underbrace{u_i(N_s+1)', \ldots, u_i(N-1)'}_{N_o-1}, \underbrace{0, \ldots, 0}_{N_s+1}]' \tag{4.4.5}$$

At $k = N_s$, the neighborhoods exchange $\tilde{\mathbf{u}}_i^{N_s+}$ and initialize $\boldsymbol{\nu}_i^{N_s+1} \leftarrow \tilde{\mathbf{u}}_i^{N_s+}$ for all $i \in \mathbb{I}_{1:4}$.

Remark 4.7. *For $N_s = 0$, the controller performs as a standard cooperative controller with only one optimization step taken between iterates.*

4.4.3 Input iteration

Here we show how the input trajectory is re-optimized by the controller at each time step. Consider the initial condition for each subsystem $i \in \mathbb{I}_{1:4}$ defined by (4.4.1)

$$\mathbf{u}_i^0 = \{u_i^0(0), u_i^0(1), \ldots, u_i^0(N-1), 0, \ldots, 0\}$$

We assume the initial condition is feasible and hence the first N steps satisfy the constraint (4.4.2d) with $k = 0$.

After $\bar{p} \geq 0$ steps of the iterate update (4.4.3), the controllers arrive at the following suboptimal point

$$\boldsymbol{v}_i^{\bar{p}} = \{v_i^{\bar{p}}(0), v_i^{\bar{p}}(1), \ldots, v_i^{\bar{p}}(N-1), 0, \ldots, 0\}$$

Only the first N steps are optimized, due to constraint (4.4.2h), leaving N_s trailing zeros. By (4.4.4), the input

$$u_i(0) = \lambda_i v_i^{\bar{p}}(0) + (1-\lambda_i) v_i^0(0)$$

is injected into the plant.

Moving to the next time step, the control algorithm (4.4.2) is initialized with $\boldsymbol{v}_i^{\bar{p}}$. By Proposition 4.6, $\boldsymbol{v}_i^{\bar{p}}$ is feasible for $k = 1$. After $\bar{p} \geq 0$ further iterates, the controller arrives at the point

$$\boldsymbol{v}_i^{2\bar{p}} = \{v_i^{\bar{p}}(0), v_i^{2\bar{p}}(1), \ldots, v_i^{2\bar{p}}(N), 0, \ldots, 0\}$$

Notice that the first input $v_i^{2\bar{p}}(0) \leftarrow v_i^{\bar{p}}(0)$ and that $v_i^{2\bar{p}}(N)$ has been optimized so that there is one fewer trailing zero. In this way, the control problem (4.4.2) always has N decision variables, but this optimization envelope moves down the trajectory of $N + N_s$ inputs. The implemented input is

$$u_i(1) = \lambda_i v_i^{2\bar{p}}(1) + (1-\lambda_i) v_i^0(1)$$

and the controller problem (4.4.2) is initialized at the next time step with $\boldsymbol{v}_i^{2\bar{p}}$. These steps are repeated until $k = N_s$, at which time the controllers compute the point

$$\boldsymbol{v}_i^{(N_s+1)\bar{p}} = \{v_i^{\bar{p}}(0), \ldots, v_i^{(N_s+1)\bar{p}}(N_s), \ldots, v_i^{(N_s+1)\bar{p}}(N+N_s-1)\}$$

This point is used to form the input sequence

$$\mathbf{u}_i^{N_s} = \lambda_i \boldsymbol{v}_i^{(N_s+1)\bar{p}} + (1-\lambda_i)\mathbf{u}_i^0, \quad \forall\, i \in \mathbb{I}_{1:4}$$

Starting from $x(0)$, the implemented inputs until time $k = N_s$ are contained in this input sequence

$$\mathbf{u}_i^{N_s} = \{\underbrace{u_i(0), \ldots, u_i(N_s)}_{N_s+1}, \underbrace{u_i(N_s+1), \ldots, u_i(N+N_s-1)}_{N-1}\}$$

for each $i \in \mathbb{I}_{1:4}$. The first N_s+1 inputs are implemented, and the last $N-1$ are used for the warm start, which we define in the sequel.

4.4.4 Objective Decrease

To prove stability, we show that the objective function decreases with every outer iterate and is bounded above between outer iterates.

Lemma 4.8 (Upperbound)**.** *Let t be the last outer iterate. Then the cost $V(x, \mathbf{u}_1^k, \mathbf{u}_2^k, \mathbf{u}_3^k, \mathbf{u}_4^k)$ is upperbounded by $V(x, \mathbf{u}_1^t, \mathbf{u}_2^t, \mathbf{u}_3^t, \mathbf{u}_4^t)$ for each $k \in \mathbb{I}_{t:t+N_s}$.*

Proof. The proof is similar to Lemma 4.1. □

Lemma 4.9 (Convergence and optimality)**.** *The cost $V(x, \mathbf{u}_1^t, \mathbf{u}_2^t, \mathbf{u}_3^t, \mathbf{u}_4^t)$ decreases at each outer iterate t and converges to the optimal value $V^0(x)$ as $t \to \infty$. The inputs $\mathbf{u}^t \to \mathbf{u}^0(x)$, the optimal centralized input sequence, as $t \to \infty$.*

Proof. The proof is similar to Lemma 4.1 and Lemma 4.2. □

Remark 4.10 (Asynchronous)**.** *Lemmas 4.8 and 4.9 do not require evenly spaced outer iterates, nor do they require $\bar{p}$ be the same in each neighborhood.*

Given Remark 4.10, the neighborhoods may exchange information after any number of iterates, and each neighborhood may be optimized on its own time schedule.

4.4.5 Stability

For every $i \in \mathbb{I}_{1:4}$, let $\underline{\mathrm{A}}_i = \mathrm{diag}(A_{1i}, \ldots, A_{4i})$ and $\underline{\mathrm{B}}_i = [B_{1i}', \ldots, B_{4i}']'$

Assumption 4.11. *For all $(i, j) \in \mathbb{I}_{1:4} \times \mathbb{I}_{1:4}$*

1. *The systems $(\underline{A}_i, \underline{B}_i)$ are stabilizable.*
2. *The input penalties $R_i > 0$.*
3. *The state penalties $Q_i \geq 0$.*

4. *The systems* (A_i, Q_i) *are detectable.*

5. $N_o \geq \max_{i\in\mathbb{I}_{1:4}}(\underline{n}_i^u)$*, in which* $\underline{n}_i^u$ *is the number of unstable modes of* $\underline{A}_i$*, i.e., number of* $\lambda \in eig(\underline{A}_i)$ *such that* $|\lambda| \geq 1$.

Notice we make no assumption on the strength of the interactions between subsystems. Let $\mathbb{X}_N$ be the forward invariant set of all initial states such that the control problem (4.4.2) is feasible.

Theorem 4.12 (Exponential stability)**.** *Given Assumption 4.11, the origin of the closed-loop system* $\phi(k,x)$ *is exponentially stable on the set* $\mathbb{X}_N$.

Proof. The proof closely follows that of the discussion of suboptimal MPC in (Rawlings and Mayne, 2009, pp.415-420). Consider the initialization of the control algorithm at the next outer iterate with the warm start

$$V(x^{N_s+},\tilde{\mathbf{u}}^{N_s+}) = V(x,\mathbf{u}^{N_s}) - \sum_{i\in\mathbb{I}_{1:4}}\sum_{k=0}^{N_s}\frac{\rho_i}{2}\ell_i(x_i(k),u_i(k)) + \sum_{i\in\mathbb{I}_{1:4}}\frac{\rho_i}{2}x_i(N)'\Psi_i(N_s)x_i(N)$$

in which $x^{N_s+} = \phi(N_s+1,x)$ and

$$\Psi_i(N_s) = (A_i^{N_s+1})'P_{if}A_i^{N_s+1} - P_{if} + \sum_{k=0}^{N_s}(A_i^k)'Q_iA_i^k$$

By the terminal constraint (4.4.2d) and the terminal penalty (4.3.2),

$$x_i(N+N_s)'\Psi_i(N_s)x_i(N+N_s) = \\ x_i(N+N_s)'S_i^s\Big((A_{ii}^{s\,\prime})^{N_s+1}\Sigma_iA_{ii}^{s\;N_s+1} - \Sigma_i + \sum_{k\in\mathbb{I}_{0:N_s}}(A_{ii}^{s\,\prime})^kS_i^{s\prime}Q_iS_i^s(A_{ii}^s)^k\Big)S_i^{s\prime}x_i(N+N_s)$$

Proposition 4.13. P_f *satisfies*

$$A'P_fA - P_f = -Q$$

for any A, Q if and only if*, for all* $N > 0$*,* P_f *satisfies*

$$(A^N)'P_fA^N - P_f = -\sum_{k=0}^{N-1}(A^k)'QA^k$$

Proof. The proof is by induction. □

By Proposition 4.13

$$x_i(N)'\Psi_i(N_s)x_i(N) = 0 \quad \forall i \in \mathbb{I}_{1:4}$$

therefore

$$V(x^{N_s+}, \tilde{\mathbf{u}}^{N_s+}) = V(x, \mathbf{u}^{N_s}) - \sum_{i\in\mathbb{I}_{1:4}} \sum_{k=0}^{N_s} \frac{\rho_i}{2} \ell_i(x_i(k), u_i(k))$$

By Lemma 4.9

$$V(x^{N_s+}, \mathbf{u}^{N_s+}) \leq V(x, \mathbf{u}^{N_s}) - \sum_{i\in\mathbb{I}_{1:4}} \sum_{k=0}^{N_s} \frac{\rho_i}{2} \ell_i(x_i(k), u_i(k))$$

Hence, there exists a c such that

$$V(x^{N_s+}, \mathbf{u}^{N_s+}) - V(x, \mathbf{u}^{N_s}) \leq -c \left| x, v^{N_s} \right|^2 \tag{4.4.6}$$

in which $v^{N_s} = \{u(0), \ldots, u(N_s - 1)\}$. Between outer iterates, by Lemma 4.8

$$V(x, \mathbf{u}^k) - V(x, \mathbf{u}^0) \leq 0 \quad k \in \mathbb{I}_{0:N_s}$$

The standard upper and lower bounding norms on $V(\cdot)$ follow from the compactness of each $\mathbb{U}_i$. We remove the appearance of the input $u(0)$ in the norm of (4.4.6) using constraint (4.4.2e) and complete the proof similarly to Lemma 6.4 in (Rawlings and Mayne, 2009, p.418). A visual representation of the proof is shown in Fig. 4.3. Therefore we may construct an exponentially decaying function upperbounding V at each time step, and exponential stability follows. □

Remark 4.14 (M Subsystems and L Levels)**.** *The above arguments are presented with a plant composed of* 4 *subsystems split into* 2 *neighborhoods. These arguments are extended to any number of subsystems, however, by replacing* 4 *in the above arguments with any integer* $M > 0$*. Levels are added to the hierarchy by treating the outer iterate for level* L *as an inner iterate for level* $L+1$*.*

For the remainder of the chapter, we will assume M subsystems.

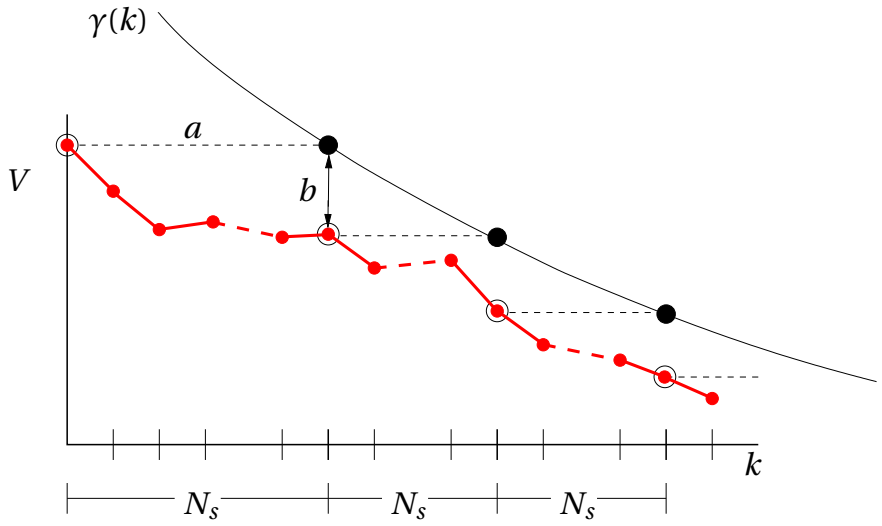


Figure 4.3: Exponential decaying upperbound of objective function. The upperbound given by line (a) is shown by Lemma 4.8. The cost drop given by the line (b) is show by Lemma 4.9. These properties allow the construction of the exponentially decaying function $\gamma(k)$.

4.5 Reducing Communication

In addition to reducing the frequency of information exchange between subsystems, it is also advantageous to reduce the complexity of the plantwide communication topology. In this section, we explicitly define neighborhoods and provide a change of variables that allows exchange of information between only a subset of the subsystems.

4.5.1 Time Delays

Consider the model

$$\begin{bmatrix} x_{ij} \\ z_{ij} \end{bmatrix}^+ = \begin{bmatrix} A_{ij} & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{ij} \\ z_{ij} \end{bmatrix} + \begin{bmatrix} 0 \\ B_{ij} \end{bmatrix} u_j$$

In this model, one time step elapses before a change in u_j contributes to x_{ij}. We then say that the x_{ij} model contains one *time delay* and abbreviate the model as

$$x_{ij}^+ = A_{ij} x_{ij} + B_{ij} u_j^-$$

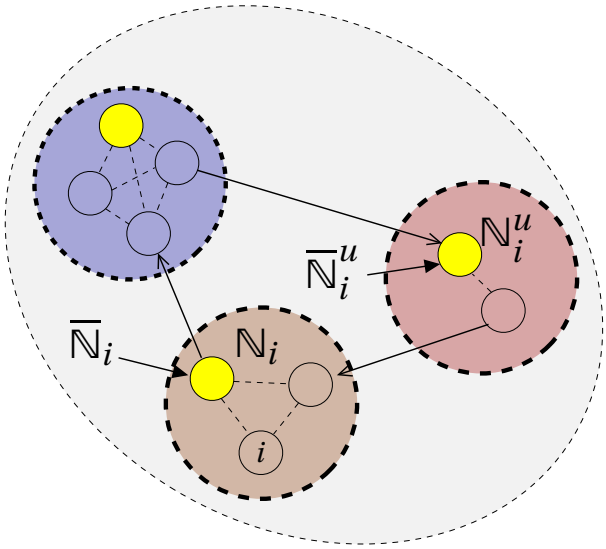


Figure 4.4: Subsystems grouped into neighborhoods with leaders. The neighborhood for subsystem i is $\mathbb{N}_i$ with leader $\overline{\mathbb{N}}_i$. The neighborhood upstream of subsystem i is $\mathbb{N}_i^u$ with leader $\overline{\mathbb{N}}_i^u$.

Similarly, a model with q time delays is abbreviated

$$x_{ij}^+ = A_{ij}x_{ij} + B_{ij}u_j^{q-}$$

4.5.2 Network Structure

Definition 4.15 (Neighborhood)**.** *The set* $\mathbb{N}$ *is denoted a neighborhood if for all* $(i,j) \in \mathbb{N} \times \mathbb{N}$ *the* x_{ij} *model has* zero *time delays.*

Definition 4.16 (Upstream Neighborhood)**.** *The neighborhood* $\mathbb{N}^u$ *is defined as upstream to neighborhood* $\mathbb{N}$ *if, for all* $i \in \mathbb{N}$ *and* $j \in \mathbb{N}^u$*, the model* x_{ij} *has one time delay.*

For all $i \in \mathbb{I}_{1:M}$, denote the neighbors of subsystem i as $\mathbb{N}_i$ and the upstream neighbors as $\mathbb{N}_i^u$. For each neighborhood $\mathbb{N}_i$, we denote the (singlet) leader set for the neighborhood as $\overline{\mathbb{N}}_i$ for all $i \in \mathbb{I}_{1:M}$. Similarly, the leader set for the upstream neighborhoods of subsystem i is denoted $\overline{\mathbb{N}}_i^u$. For example, see Fig. 4.4.

4.5.3 Models

Consider an alternative model to (4.3.1) in which for every $i \in \mathbb{I}_{1:M}$, the model for subsystem i is

$$\begin{aligned} x_{ij}^+ &= A_{ij}x_{ij} + B_{ij}u_j \quad \forall j \in \mathbb{N}_i \\ x_{il}^+ &= A_{il}x_{il} + B_{il}u_l^- \quad \forall l \in \mathbb{N}_i^u \\ x_{is}^+ &= A_{is}x_{is} + B_{is}u_s^{q-} \quad \forall s \in \mathbb{N}_i^{[q]}, \forall q > 1 \end{aligned} \tag{4.5.1}$$

in which the qth upstream neighborhood is $\mathbb{N}_i^{[q]} = \bigcup_{j\in\mathbb{N}_i^u} \mathbb{N}_j^{[q-1]}$ with $\mathbb{N}_i^{[1]} = \mathbb{N}_i^u$. Without loss of generality[2], we may write the model in the following form

$$x_{ij}^+ = A_{ij}x_{ij} + B_{ij}u_j \quad \forall j \in \mathbb{N}_i \tag{4.5.2a}$$

$$x_{il}^+ = A_{il}x_{il} + E_{il}x_{sl} \quad \forall l \in \mathbb{N}_i^u, s \in \overline{\mathbb{N}}_l \tag{4.5.2b}$$

in which $E_{il} \in \mathbb{R}^{n_{ij}\times n_{sl}}$. Collecting the states into vectors $x_i = [\cdots, x_{ij}', \cdots]'$ for all $i \in \mathbb{I}_{1:M}$, $j \in \mathbb{N}_i \cup \mathbb{N}_i^u$, and we form the following matrices for all $i \in \mathbb{I}_{1:M}$

$$\begin{aligned} \bar{A}_{ii} &= \text{diag}(\cdots, A_{ij}, \cdots) \quad \forall j \in \mathbb{N}_i \cup \mathbb{N}_i^u \\ \bar{A}_{il} &= [\cdots, \bar{E}_{ij}', \cdots, 0, \cdots, 0]' \quad \forall j \in \mathbb{N}_l \text{ s.t. } l \in \overline{\mathbb{N}}_i^u \end{aligned}$$

in which $\bar{A}_{ij} \in \mathbb{R}^{n_i\times n_j}$ and

$$\bar{E}_{ij} = [0 \cdots \underbrace{E_{ij}'}_{j\text{th position}} \cdots 0]' \in \mathbb{R}^{n_i \times n_{ij}} \ \forall j \in \mathbb{N}_i^u$$

For all $\bar{A}_{ij}$ not defined above, $\bar{A}_{ij} = 0$. These matrices form the model

$$x_i^+ = \bar{A}_{ii}x_i + \sum_{j\in\mathbb{N}_i} \bar{B}_{ij}u_j + \sum_{l\in\overline{\mathbb{N}}_i^u} \bar{A}_{il}x_l \quad \forall i \in \mathbb{I}_{1:M} \tag{4.5.3}$$

Notice that each state is affected only by inputs within its neighborhood and the states of upstream leaders.

[2] See the discussion in Appendix A

4.5.4 Interaction Graph

To represent the state-to-state interactions of (4.5.3), we construct a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ in which each subsystem is a vertex, i.e., $\mathscr{V} = \mathbb{I}_{1:M}$. For each subsystem $i \in \mathbb{I}_{1:M}$, there exists an edge $(i,i) \in \mathscr{E}$ and an edge $(j,i) \in \mathscr{E}$ for all $j \in \overline{\mathbb{N}}_i^u$.

A path is a sequence of vertices connected by edges. A path $P = \{i, j, l, \ldots, m, n\}$ such that $i, j, l, \ldots, m, n \in \mathscr{V}$ and $(i,j), (j,l), \ldots, (m,n) \in \mathscr{E}$. Let the path P_{ij} be a path from subsystem i to subsystem j and the set $\mathbb{P}_{ij}^k = \{P_{ij}\}$ be the set of all paths from i to j having k connecting edges. Define

$$\bar{A}_{ij}^{[k]} = \sum_{P_{ij} \in \mathbb{P}_{ij}^k} \prod_{(m,n) \in P_{ij}} \bar{A}_{mn}$$

This matrix gives the total effect of the state-to-state interactions from subsystem i to subsystem j that propagate in k steps.

4.5.5 Reduced Communication

Communication is reduced by rearranging model (4.5.3) in the following way. For all $i \in \mathbb{L}$

$$\alpha_i^+ = \bar{A}_{ii}\alpha_i + \sum_{j \in \mathbb{N}_i} \bar{B}_{ij} u_j \tag{4.5.4}$$

for which $\alpha_i(0) = x_i(0)$. Notice that α is defined only for the neighborhood leaders, and its computation needs only information, e.g., states and inputs, available within the neighborhood. At time $k \geq 0$, the states of subsystem $i \in \mathbb{I}_{1:M}$ are

$$x_i(k) = \bar{A}_{ii}^k x_i(0) + \sum_{\tau=0}^{k-1} \sum_{j \in \mathbb{N}_i} \bar{A}_{ii}^{k-\tau-1} \bar{B}_{ij} u_j(\tau) + \sum_{\tau=0}^{k-1} \sum_{l \in \mathbb{L}} \sum_{s \in \mathbb{I}_{1:M} \setminus l} \bar{A}_{is}^{[k-\tau-1]} \bar{A}_{sl} \alpha_l(\tau)$$

To compute its state, each subsystem communicates only with the neighborhood leaders. This communication topology reduces the complexity of information sharing in the network. The total plantwide communication required in cooperative control is Nm bytes per exchange. By exchanging only the α trajectories between neighborhoods, the communication is $N(\sum_{i \in \mathbb{L}} n_i)$ bytes per exchange.

4.5.6 Control Algorithm

As previously, assume the last outer iterate is $k = 0$. First the leaders communicate and compute the values of α_l^0 for all $l \in \mathbb{L}$. Then for all $i \in \mathbb{I}_{1:M}$, the following optimization is performed for all $p \in \mathbb{I}_{k\tilde{p}:(k+1)\tilde{p}}$

$$\min_{\boldsymbol{v}_i} V(x(0), \boldsymbol{v}_1, \ldots, \boldsymbol{v}_i, \ldots, \boldsymbol{v}_M) \tag{4.5.5a}$$

subject to (4.5.5a)

$$\boldsymbol{v}_i \in \mathbb{U}_i^N \tag{4.5.5b}$$

$$S_{ji}^{u\,\prime} x_{ji}(k + N_o) = 0 \quad j \in \mathbb{I}_{1:M} \tag{4.5.5c}$$

$$|\boldsymbol{v}_i| \leq d_i \sum_{j \in \mathbb{I}_{1:M}} \left| x_{ji}(0) \right| \quad \text{if } x_{ji}(0) \in \mathbb{B}_r \quad j \in \mathbb{I}_{1:M} \tag{4.5.5d}$$

$$\boldsymbol{v}_j = \boldsymbol{v}_j^p \quad j \in \mathbb{N}_i \setminus i \tag{4.5.5e}$$

$$\boldsymbol{v}_l = \boldsymbol{v}_l^0 \quad l \in \mathbb{I}_{1:M} \setminus \mathbb{N}_i \tag{4.5.5f}$$

$$v_i(\tau) = v_i^p(\tau) \quad \forall \tau \in \mathbb{I}_{0:k-1} \cup \mathbb{I}_{k+N_o:N-1} \tag{4.5.5g}$$

The inner and outer iterates are performed as in Section 4.4. At an outer iterate, the leaders recompute α. The procedure is then repeated.

Remark 4.17. *Because the neighborhoods exchange the trajectory of α values instead of the input trajectories, the future input plans are not shared between neighborhoods.*

Remark 4.18. *The control algorithm and optimization problem remains the same and, given the previous assumptions remain satisfied, all properties proven in the previous sections are maintained.*

4.6 Output feedback

In general the model (4.3.1) is nonminimal and the states x_{ij} are unobservable. This problem motivates the an output feedback scheme in which the states are estimated at each time

step. For all $(i,j) \in \mathbb{I}_{1:M} \times \mathbb{I}_{1:M}$, consider the model

$$x_{ij}^+ = A_{ij}x_{ij} + B_{ij}u_j$$
$$y_i = \sum_{i \in \mathbb{I}_{1:M}} C_{ij}x_{ij}$$

in which $y_i \in \mathbb{R}^{p_i}$ is the output of subsystem i and $C_{ij} \in \mathbb{R}^{p_i \times n_{ij}}$. For notational simplicity, we write $y_i = C_i x_i$ in which $C_i = [C_{i1}, \ldots, C_{iM}]$.

Assumption 4.19. *For all $i \in \mathbb{I}_{1:M}$, (A_i, C_i) is detectable.*

4.6.1 Smoothing the Initial State

Consider an estimate of the initial state $\hat{x}_i(0)$ with covariance P_{i0}. As new measurements are obtained between synchronization times, we update the estimate of the initial state using the following relationships. Letting $\hat{x}_i^{[0]-}(0) = \hat{x}_i(0)$, $P_i^{[0]} = P_{i0}$ and $\Pi_i^{[0]} = Q_{wi}$ for all $i \in \mathbb{I}_{1:M}$

$$\hat{x}_i^{[k+1]}(0) = \hat{x}_i^{[k]}(0) + L_i^{[k]}(y_i(k) - C_i\hat{x}_i^{[k]}(k))$$
$$L_i^{[k]} = P_i^{[k]}(A_i^k)'C_i'(C_i\Pi_i^{[k]}C_i' + R_{vi})^{-1}$$
$$\Pi_i^{[k]} = A_i^k P_i^{[k]}(A_i^k)' + \sum_{t=0}^{k-1} A_i^t Q_{wi}(A^t)'$$
$$P_i^{[k+1]} = P_i^{[k]}(A_i^k)'C_i'(C_i\Pi_i^{[k]}C_i' + R_{vi})^{-1}C_iA_i^kP_i^{[k]}$$

in which $\hat{x}_i^{[k]}(k) = \phi_i(k; \hat{x}_i^{[k]}(0), \mathbf{u}^k)$ is the nominal estimate of $x_i(k)$, and Q_{wi} and R_{vi} are the decentralized input and output disturbance covariances, respectively.

4.6.2 Nominal Estimation Error

Between outer iterates, the input injected into the plant is not communicated between subsystems in differing neighborhoods. This reduction of communication leads to estimate error in the nominal system, because each subsystem estimates its state using old values of the inputs outside its neighborhood. Between outer iterates, the state estimate is

$$\hat{x}_i^+ = A_i\hat{x}_i + \sum_{j \in \mathbb{N}_i} \bar{B}_{ij}u_j + \sum_{j \in \mathbb{I}_{1:M} \setminus \mathbb{N}_i} \bar{B}_{ij}\tilde{u}_j \tag{4.6.1}$$

in which $\tilde{u}_j$ is the last communicated value of u_j. Defining the nominal error to be $e_i^{[k]} = x_i(0) - \hat{x}_i^{[k]}(0)$ for all $i \in \mathbb{I}_{1:M}$, this error propagates with the following model

$$e_i^{[k+1]} = e_i^{[k]} - L_i^{[k]} C_i \big(x_i(k) - \hat{x}_i^{[k]}(k) \big)$$

Using (4.6.1), we arrive at the model

$$e_i^{[k+1]} = \big(I - L_i^{[k]} C_i A_i^k \big) e_i^{[k]} - L_i^{[k]} C_i \sum_{j \in \mathbb{I}_{1:M} \setminus \mathbb{N}_i} \sum_{t=0}^{k-1} A_i^{k-t-1} \bar{B}_{ij} (u_j(t) - \tilde{u}_j(t))$$

To reduce notation, we use the following

$$e_i^{[k+1]} = A_{Li}^{[k]} e_i^{[k]} + \sum_{j \in \mathbb{I}_{1:M} \setminus \mathbb{N}_i} B_{Lij}^{[k]} (\mathbf{u}_j^{k-1} - \tilde{\mathbf{u}}_j)$$

in which

$$A_{Li}^{[k]} = (I - L_i^{[k]} C_i A_i^k)$$
$$B_{Lij}^{[k]} = -L_i^{[k]} C_i \sum_{j \in \mathbb{I}_{1:M} \setminus \mathbb{N}_i} \sum_{t=0}^{k-1} A_i^{k-t-1} \bar{B}_{ij}$$

Noting that nominally $e_i^{[0]} = 0$, the above equation is further simplified to

$$e_i^{[k+1]} = \sum_{j \in \mathbb{I}_{1:M} \setminus \mathbb{N}_i} E_{ij}^{[k-1]} (\mathbf{u}_j^{k-1} - \tilde{\mathbf{u}}_j)$$

in which

$$E_{ij}^{[k-1]} = \sum_{t=0}^{k-1} \Big(\prod_{s=t+1}^{k-1} A_{Li}^{[s]} \Big) B_{Lij}^{[t]}$$

We form the plantwide error as

$$e^{[k+1]} = E^{[k-1]} (\mathbf{u}^{k-1} - \tilde{\mathbf{u}}) \tag{4.6.2}$$

in which $E^{[k-1]} = \{E_{ij}^{[k-1]}\}$. Notice that the nominal error is affected by the inputs at least two time steps behind only.

4.6.3 Output Feedback Stability

The nominal stability of output feedback follows similar arguments to that of the state feedback case in Section 4.4.5. We first show that if a stability assumption holds, the plantwide cost is a Lyapunov function for the closed-loop system every N_s time steps. We then show that the plantwide objective is upperbounded at all other times.

Define the set of admissible state and input sequence pairs as

$$\mathbb{Z}_N = \{(x, \mathbf{u}) \in \mathbb{R}^n \times \mathbb{U}^N \mid V(x, \mathbf{u}) \leq a, \phi(N; x, \mathbf{u}) = 0\}$$

in which $a > 0$ is arbitrary. The set of initial states $\mathbb{X}_N$ is then

$$\mathbb{X}_N = \{x \mid \exists \mathbf{u} \text{ such that } (x, \mathbf{u}) \in \mathbb{Z}_N\}$$

To show stability, we require the nominal error to be small. First note that using (4.6.2) the nominal estimate at the next outer iterate satisfies

$$\hat{x}^{N_s+} = x^{N_s+} - e^{[N_s+]} = x^{N_s+} - E^{[N_s-1]}(\mathbf{u}^{N_s-1} - \tilde{\mathbf{u}})$$

We then make the following assumption on the nominal error.

Assumption 4.20. *For all $x \in \mathbb{X}_N$ and $\mathbf{u}^{N_s} \in \mathbb{U}^N$*

$$V(x^{N_s+}, \mathbf{h}(e^{[N_s+]})) \leq V(\hat{x}^{N_s+}, \tilde{\mathbf{u}}^{N_s+})$$

in which $x^{N_s+} = \phi(N_s + 1; x, \mathbf{u}^{N_s})$, $\mathbf{h}(\cdot)$ is the reinitialization function, and $\tilde{\mathbf{u}}^{N_s+}$ is the warm start formed from $\mathbf{u}^{N_s}$.

The following assumption bounds the nominal error between outer iterates.

Assumption 4.21. *For all $x \in \mathbb{X}_N$ and $\tilde{\mathbf{u}} \in \mathbb{U}^N$*

$$V(x, \mathbf{h}(e^{[k]}) \leq V(x, \tilde{\mathbf{u}}) \quad \forall k \in \mathbb{I}_{0:N_s}$$

Remark 4.22. *The above assumptions require the true state x and the nominal estimated state $\hat{x}^{[k]}$ to be close for all input sequences chosen by the distributed controller without communication. For example, these requirements are be met by requiring $E^{[k]}(\mathbf{u}^k - \tilde{\mathbf{u}}) = 0$ for all $k \in \mathbb{I}_{0:N_s}$.*

Theorem 4.23 (Output feedback nominal exponential stability)**.** *Let Assumptions 4.11, 4.20, and 4.21 hold. Then the origin of the closed-loop system $\phi(k,x)$ is exponentially stable.*

Proof. At the outer iterate, we have by Assumption 4.20 that

$$\begin{aligned} V(x^{N_s+},\mathbf{h}(e^{[N_s+]})) &\leq V(\hat{x}^{N_s+},\tilde{\mathbf{u}}^{N_s+}) \\ &\leq V(x,\mathbf{u}^{N_s} - \sum_{i\in\mathbb{I}_{1:M}}\sum_{k=0}^{N_s}\frac{\rho_i}{2}\ell_i(\hat{x}_i(k),u_i(k)) + \sum_{i\in\mathbb{I}_{1:M}}\frac{\rho_i}{2}\hat{x}_i(N)'\Psi_i(N_s)\hat{x}_i(N) \end{aligned}$$

in which $\hat{x}_i(k) = \hat{x}_i^{[k]}(k)$. Using similar arguments as Theorem 4.12, it follows that

$$\begin{aligned} V(x^{N_s+},\mathbf{h}(e^{[N_s+]})) &\leq V(x,\mathbf{u}^{N_s}) - \sum_{i\in\mathbb{I}_{1:M}}\frac{\rho_i}{2}\ell_i(x,u(0)) \\ V(x^{N_s+},\mathbf{u}^{N_s+}) &\leq V(x,\mathbf{u}^{N_s})w - \sum_{i\in\mathbb{I}_{1:M}}\frac{\rho_i}{2}\ell_i(x,u(0)) \end{aligned}$$

by Assumption 4.21. We can then find a constant c such that

$$V(x^{N_s+},\mathbf{u}^{N_s+}) \leq V(x,\mathbf{u}^{N_s}) - c\,|(x,u(0))|^2$$

By Assumption 4.21 between outer iterates the cost is upperbounded. The proof is completed as in Theorem 4.12. □

Remark 4.24. *The stability of output feedback with perturbation can be established as in Theorem 3.17 because the smoothing estimator given in Section 4.6.1 is stable and therefore a Lyapunov function for the error exists.*

4.7 Hierarchical Control Example

To show the performance characteristics of hierarchical cooperative control, we perform the same disturbance rejection examples as in Section 3.7. The control objective is to reject a disturbance in a distillation column using LV control. The model for the process is

$$\begin{pmatrix} x_D(s) \\ x_B(s) \end{pmatrix} = \begin{pmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21.0s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{pmatrix} \begin{pmatrix} R(s) \\ S(s) \end{pmatrix}$$

Table 4.1: Performance comparison. Performance deteriorates as N_s increases, but does not approach the poor performance of decentralized or noncooperative control.

	Cost	Performance loss (%)
$N_s = 0$	87.5	15.4
$N_s = 1$	104	37.4
$N_s = 5$	139	83.2
$N_s = 10$	177	134
$N_s = 20$	195	157

The process is controlled with the poor pairing of $x_D - S$ and $x_B - R$. At time $t = 5$ minutes a disturbance enters x_B, the bottoms mole fraction. The performance of the hierarchical cooperative controller with different communication time schedules is shown in Figure 4.5. For all time schedules, the disturbance is rejected, although performance deteriorates as the communication delay, N_s is increased (see Table 4.1). Notice that control with $N_s = 20$ is stable, even though this delay is larger than the optimization horizon $N_o = 10$. For this example, none of the cases approach the poor performance of decentralized or noncooperative control (see Table 3.1).

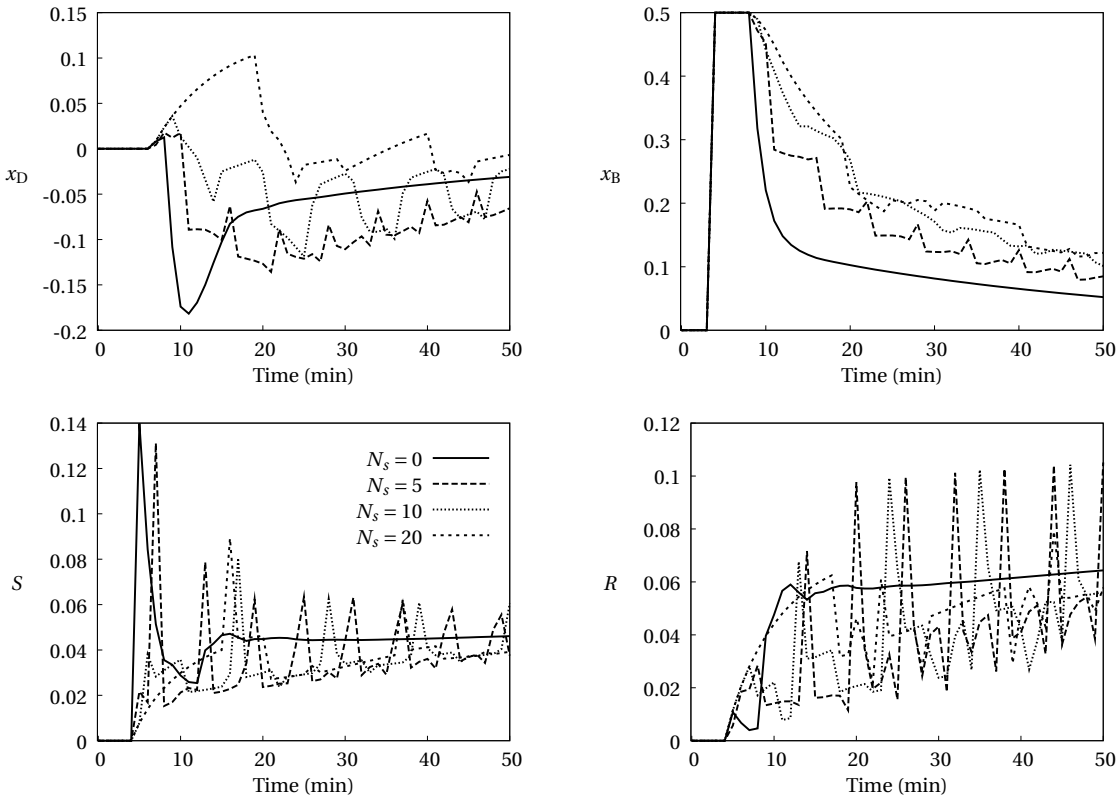


Figure 4.5: Disturbance rejection example.

Chapter 5

Implementing Plantwide Cooperative Control

5.1 Introduction

The goal of plantwide control is to coordinate many interacting subsystems. The subsystems are often designed independently from the rest of the plant or added over time to meet new production requirements. As each subsystem has its own implementation challenges, within a plant there is often a mix of economic objectives, time scales, and modeling choices. The challenge of plantwide control is to assimilate these objectives into a unifying control strategy.

The proceeding chapters have been primarily concerned with giving the theoretical background of cooperative control. In this chapter we explore the practical application of cooperative control and show the flexibility of the method. We begin by describing how to develop the models for cooperative control. We then provide a chemical plant example to illustrate that cooperative control can nominally coordinate subsystems operating at any time scale, sampling at any time scale, and can exchange information on any time scale.

5.2 Plantwide Modeling

5.2.1 From Decentralized to Cooperative Control

Many large-scale chemical plants implement decentralized control. This control strategy choice is often adequate if open-loop interactions between the subsystems in the plant are weakly coupled (Sandell Jr. et al., 1978). Yet even weakly interacting subsystems can benefit from distributed control, in which these interactions are taken explicitly into account.

Consider a plant composed of M subsystems. These subsystems interact and hence, for each pair of subsystems, we can represent the interactions with an input/output model $G_{ij}(s)$ capturing the affect of input u_j on the output y_i for all $(i,j) \in \mathbb{I}_{1:M} \times \mathbb{I}_{1:M}$. We can arrange these models into the matrix form

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1M}(s) \\ G_{21}(s) & G_{22}(s) & \cdots & G_{2M}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{M1}(s) & G_{M2}(s) & \cdots & G_{MM}(s) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{pmatrix}$$

In decentralized control the plantwide controller is implemented with the assumption that $G_{ij}(s) = 0$ for all $i \neq j$. Most likely, this assumption is an approximation of the true plant model. From the view of distributed control the decentralized assumption is a case of plant/model mismatch. Plantwide performance can be improved by identifying each $G_{ij}(s)$ and implementing distributed control. Gudi and Rawlings (2006) give a method for determining $G_{ij}(s)$ for plantwide control.

For many plants it is infeasible or impossible to identify all interaction models, however. It may also be easier, in practice, to develop the plant model over time. In both cases, plant/model mismatch can be reduced by identifying a subset of the interaction models. We now provide an example showing that distributed control provides better performance without identifying all interaction models.

5.2.1.1 Distillation Column Example

Consider the distillation column for separating methanol from water in Figure 5.1. A two input and two output model was proposed by Wood and Berry (1973). The Laplace domain form of the model is

$$\begin{pmatrix} x_D \\ x_B \end{pmatrix} = \begin{pmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21.0s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{pmatrix} \begin{pmatrix} R \\ S \end{pmatrix}$$

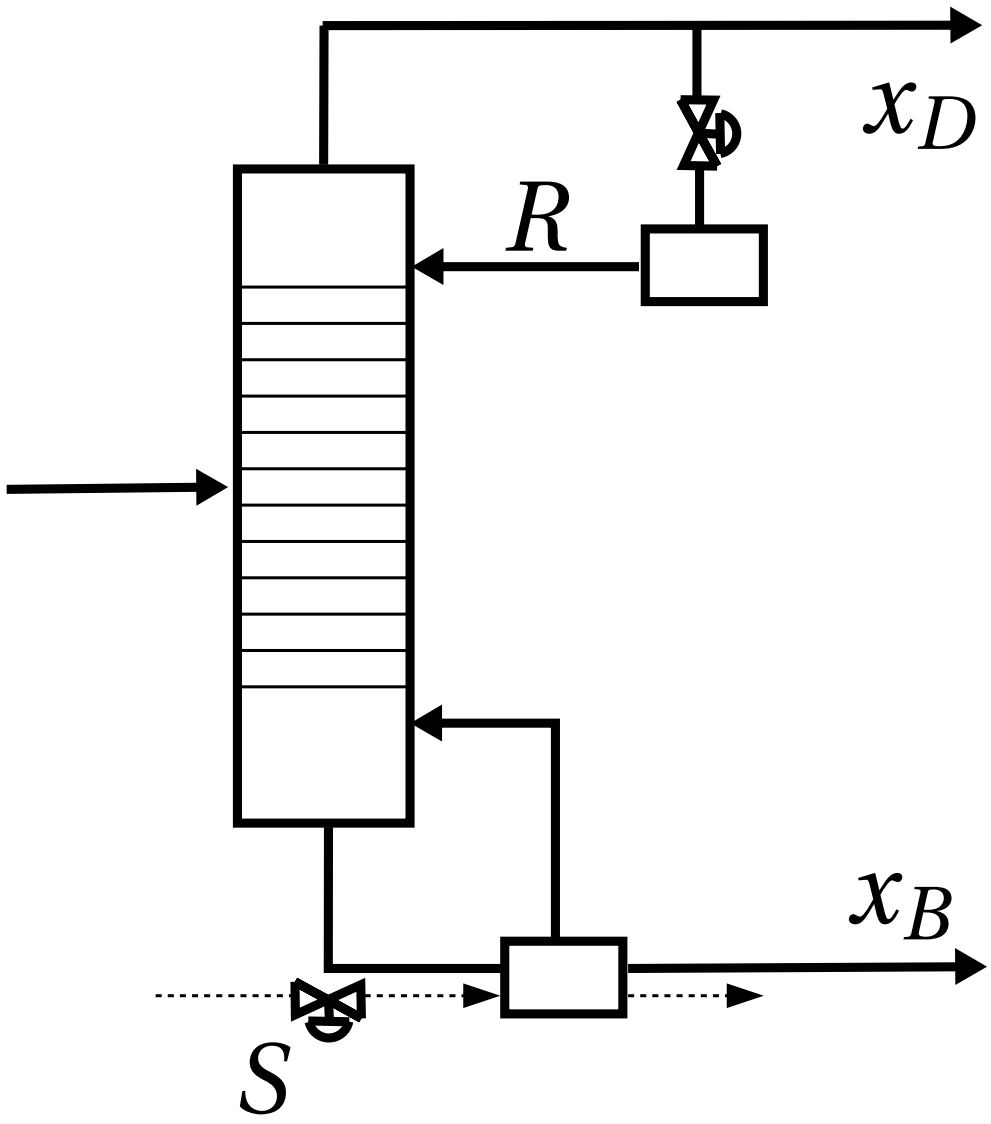


Figure 5.1: LV control of distillation column

Table 5.1: Performance comparison. Adding subsystem interaction model information improves controller performance.

	Cost	Performance loss (%)
Cooperative MPC	1.16	0
$G_{12} = 0$	1.7	46.9
$G_{21} = 0$	2.35	103
$G_{12} = G_{21} = 0$	2.53	119
Decentralized MPC	2.56	121

in which

$$x_D - \text{overhead mole fraction methanol}$$

$$x_B - \text{bottoms mole fraction methanol}$$

$$R - \text{overhead reflux flow rate}$$

$$S - \text{bottoms steam flow rate}$$

We choose the good pairing of subsystems, $y_1 = x_D$, $u_1 = R$, $y_2 = x_B$, $u_2 = S$. We simulate a setpoint change of y_1 from 0 to 0.75 subject to the input constraints $0 \geq |u_1| \geq 0.15$. Five simulations are conducted. We show the performance of cooperative control subject to full knowledge of the plant, but we also show performance with plant/model mismatch in which $G_{12} = 0$, $G_{21} = 0$, and both are zero. The simulation results are shown in Figure 5.2. Each controller is stabilizing, but performance increases as the model is made more like the plant (see Table 5.1). In all cases, cooperative control is an improvement on decentralized control.

5.2.2 From Centralized to Cooperative Control[1]

A plantwide distributed controller can also be designed using a first-principles model of the plant. This method is useful to evaluate decompositions of the plant into subsystems. A plant decomposition is chosen by dividing the plantwide inputs and outputs into a M sets of (u_i, y_i)

[1] This section appears in (Stewart et al., 2010b)

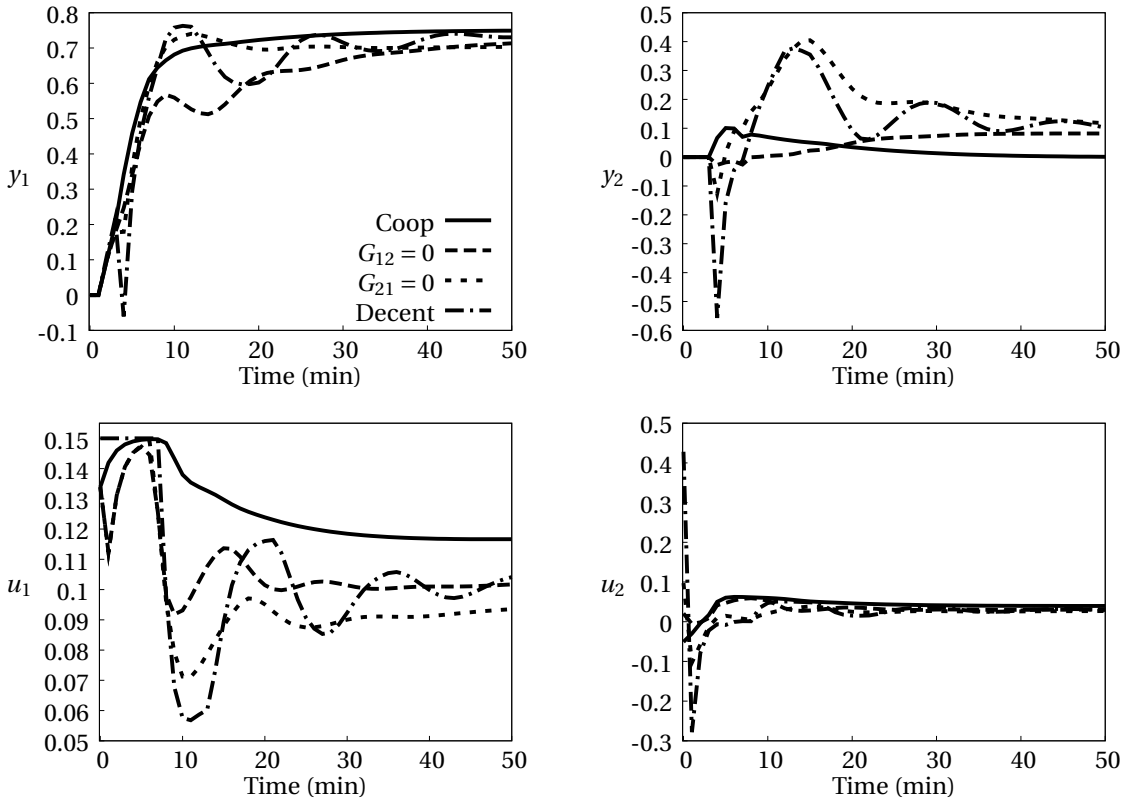


Figure 5.2: Performance of LV controller for the distillation column of Wood and Berry (1973). The control performance improves as model information is added.

pairs. Then consider the possibly nonminimal centralized model

$$x^+ = Ax + \sum_{j \in \mathbb{I}_{1:M}} B_j u_j \tag{5.2.1}$$

$$y_i = C_i x \quad \forall i \in \mathbb{I}_{1:M}$$

For each input/output pair (u_j, y_i) we transform the triple (A, B_j, C_i) into its Kalman canonical form (Antsaklis and Michel, 1997, p.270)

$$\begin{bmatrix} z_{ij}^{oc} \\ z_{ij}^{\bar{o}c} \\ z_{ij}^{o\bar{c}} \\ z_{ij}^{\bar{o}\bar{c}} \end{bmatrix}^+ = \begin{bmatrix} A_{ij}^{oc} & 0 & A_{ij}^{oc\bar{c}} & 0 \\ A_{ij}^{\bar{o}oc} & A_{ij}^{\bar{o}c} & A_{ij}^{\bar{o}co\bar{c}} & A_{ij}^{\bar{o}c\bar{c}} \\ 0 & 0 & A_{ij}^{o\bar{c}} & 0 \\ 0 & 0 & A_{ij}^{\bar{o}\bar{c}o} & A_{ij}^{\bar{o}\bar{c}} \end{bmatrix} \begin{bmatrix} z_{ij}^{oc} \\ z_{ij}^{\bar{o}c} \\ z_{ij}^{o\bar{c}} \\ z_{ij}^{\bar{o}\bar{c}} \end{bmatrix} + \begin{bmatrix} B_{ij}^{oc} \\ B_{ij}^{\bar{o}c} \\ 0 \\ 0 \end{bmatrix} u_j$$

$$y_{ij} = \begin{bmatrix} C_{ij}^{oc} & 0 & C_{ij}^{o\bar{c}} & 0 \end{bmatrix} \begin{bmatrix} z_{ij}^{oc} \\ z_{ij}^{\bar{o}c} \\ z_{ij}^{o\bar{c}} \\ z_{ij}^{\bar{o}\bar{c}} \end{bmatrix} \qquad y_i = \sum_{j \in \mathbb{I}_{1:M}} y_{ij}$$

The vector z_{ij}^{oc} captures the modes of A that are both observable by y_i and controllable by u_j. The distributed model in the form of (3.4.1) is then

$$A_{ij} \leftarrow A_{ij}^{oc} \quad B_{ij} \leftarrow B_{ij}^{oc} \quad C_{ij} \leftarrow C_{ij}^{oc} \quad x_{ij} \leftarrow z_{ij}^{oc}$$

In the following example, the above procedure is used to generate the distributed model.

5.3 Chemical Plant Example[2]

In this section, we construct a chemical plant example in order to evaluate the plantwide control strategies. Consider a plant consisting of two reactors and a separator. A stream of pure reactant A is added to each reactor and converted to the product B by a first-order reaction. The product is lost by a parallel first-order reaction to side product C. The distillate of the separator is split and partially redirected to the first reactor (see Figure 5.3). The model for the plant is

[2] A similar example is provided in (Stewart et al., 2010b)

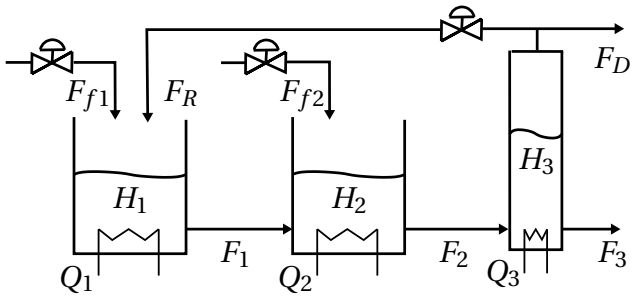


Figure 5.3: Two reactors in series with separator and recycle.

$$
\begin{aligned}
\frac{dH_1}{dt} &= \frac{1}{\rho A_1}(F_{f1} + F_R - F_1) \\
\frac{dx_{A1}}{dt} &= \frac{1}{\rho A_1 H_1}(F_{f1}x_{A0} + F_R x_{AR} - F_1 x_{A1}) - k_{A1}x_{A1} \\
\frac{dx_{B1}}{dt} &= \frac{1}{\rho A_1 H_1}(F_R x_{BR} - F_1 x_{B1}) + k_{A1}x_{A1} - k_{B1}x_{B1} \\
\frac{dT_1}{dt} &= \frac{1}{\rho A_1 H_1}(F_{f1}T_0 + F_R T_R - F_1 T_1) \\
&\quad - \frac{1}{C_p}(k_{A1}x_{A1}\Delta H_A + k_{B1}x_{B1}\Delta H_B) + \frac{Q_1}{\rho A_1 C_p H_1} \\
\frac{dH_2}{dt} &= \frac{1}{\rho A_2}(F_{f2} + F_1 - F_2) \\
\frac{dx_{A2}}{dt} &= \frac{1}{\rho A_2 H_2}(F_{f2}x_{A0} + F_1 x_{A1} - F_2 x_{A2}) - k_{A2}x_{A2} \\
\frac{dx_{B2}}{dt} &= \frac{1}{\rho A_2 H_2}(F_1 x_{B1} - F_2 x_{B2}) + k_{A2}x_{A2} - k_{B2}x_{B2} \\
\frac{dT_2}{dt} &= \frac{1}{\rho A_2 H_2}(F_{f2}T_0 + F_1 T_1 - F_2 T_2) \\
&\quad - \frac{1}{C_p}(k_{A2}x_{A2}\Delta H_A + k_{B2}x_{B2}\Delta H_B) + \frac{Q_2}{\rho A_2 C_p H_2} \\
\frac{dH_3}{dt} &= \frac{1}{\rho A_3}(F_2 - F_D - F_R - F_3) \\
\frac{dx_{A3}}{dt} &= \frac{1}{\rho A_3 H_3}(F_2 x_{A2} - (F_D + F_R)x_{AR} - F_3 x_{A3}) \\
\frac{dx_{B3}}{dt} &= \frac{1}{\rho A_3 H_3}(F_2 x_{B2} - (F_D + F_R)x_{BR} - F_3 x_{B3}) \\
\frac{dT_3}{dt} &= \frac{1}{\rho A_3 H_3}(F_2 T_2 - (F_D + F_R)T_R - F_3 T_3) + \frac{Q_3}{\rho A_3 C_p H_3}
\end{aligned}
$$

in which for all $i \in \mathbb{I}_{1:3}$

$$F_i = k_{vi} H_i \quad k_{Ai} = k_A \exp(-\frac{E_A}{RT_i}) \quad k_{Bi} = k_B \exp(-\frac{E_B}{RT_i})$$

The recycle flow and weight percents satisfy

$$F_D = 0.01 F_R \quad x_{AR} = \frac{\alpha_A x_{A3}}{\bar{x}_3} \quad x_{BR} = \frac{\alpha_B x_{B3}}{\bar{x}_3}$$
$$\bar{x}_3 = \alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3} \qquad x_{C3} = (1 - x_{A3} - x_{B3})$$

The output and input are denoted, respectively

$$y = \begin{bmatrix} H_1 \; x_{A1} \; x_{B1} \; T_1 \; H_2 \; x_{A2} \; x_{B2} \; T_2 \; H_3 \; x_{A3} \; x_{B3} \; T_3 \end{bmatrix}$$
$$u = \begin{bmatrix} F_{f1} \; Q_1 \; F_{f2} \; Q_2 \; F_R \; Q_3 \end{bmatrix}$$

We linearize the nonlinear plant model around the steady state defined by Table 5.2 and derive the following linear discrete-time model with sampling time $\Delta = 0.1$s

$$x^+ = Ax + Bu \quad y = x$$

5.3.1 Distributed Control

We choose to control the separator and each reactor independently and partition the plant into 3 subsystems by defining

$$y_1 = \begin{bmatrix} H_1 \; x_{A1} \; x_{B1} \; T_1 \end{bmatrix} \qquad u_1 = \begin{bmatrix} F_{f1} \; Q_1 \end{bmatrix}$$
$$y_2 = \begin{bmatrix} H_2 \; x_{A2} \; x_{B2} \; T_2 \end{bmatrix} \qquad u_2 = \begin{bmatrix} F_{f2} \; Q_2 \end{bmatrix}$$
$$y_3 = \begin{bmatrix} H_3 \; x_{A3} \; x_{B3} \; T_3 \end{bmatrix} \qquad u_3 = \begin{bmatrix} F_R \; Q_3 \end{bmatrix}$$

Following the distributed model derivation outlined in Section 5.2.2, we form the distributed model for the plant.

Table 5.2: Steady states and parameters

Parameter	Value	Units	Parameter	Value	Units
H_1	29.8	m	A_1	3	m^2
x_{A1}	0.542	wt.%	A_2	3	m^2
x_{B1}	0.393	wt.%	A_3	1	m^2
T_1	315	K	ρ	0.15	kg/m^3
H_2	30	m	C_p	25	kJ/kg·K
x_{A2}	0.503	wt.%	k_{v1}	2.5	kg/m·s
x_{B2}	0.421	wt.%	k_{v2}	2.5	kg/m·s
T_2	315	K	k_{v3}	2.5	kg/m·s
H_3	3.27	m	x_{A0}	1	wt.%
x_{A3}	0.238	wt.%	T_0	313	K
x_{B3}	0.570	wt.%	k_A	0.02	1/s
T_3	315	K	k_B	0.018	1/s
F_{f1}	8.33	kg/s	E_A/R	-100	K
Q_1	10	kJ/s	E_B/R	-150	K
F_{f2}	0.5	kg/s	ΔH_A	-40	kJ/kg
Q_2	10	kJ/s	ΔH_B	-50	kJ/kg
F_R	66.2	kg/s	α_A	3.5	
Q_3	10	kJ/s	α_B	1.1	
			α_C	0.5	

Table 5.3: Input constraints

Parameter	Lower bound	Steady state	Upper bound	Units
F_{f1}	0	8.33	10	kg/s
Q_1	0	10	50	kJ/s
F_{f2}	0	0.5	10	kg/s
Q_2	0	10	50	kJ/s
F_R	0	66.2	75	kg/s
Q_3	0	10	50	kJ/s

Table 5.4: Distributed control performance comparison

	Cost	Performance loss (%)
Centralized MPC	0.101	0
Cooperative MPC (10 iterates)	0.101	0.504
Cooperative MPC (1 iterate)	0.12	18.9
Noncooperative MPC	0.301	199
Decentralized MPC	1.69	1.58e+03

5.3.2 Simulation

Consider the performance of distributed control with the partitioning defined above. The tuning parameters are

$$Q_{yi} = \text{diag}(1,0,0,0.1)\ \forall i = \mathbb{I}_{1:2} \quad Q_{y3} = \text{diag}(1,0,10^3,0)$$

$$Q_i = C_i' Q_{yi} C_i + 0.001 I \quad R_i = 0.01 I \quad \forall i \in \mathbb{I}_{1:3}$$

The horizon length is 5 seconds, i.e., $N = 50$. The input constraints are defined in Table 5.3. We simulate a setpoint change in the output product weight percent x_{B3} at $t = 0.5s$.

5.3.3 Comparing Plantwide Control Methods

In Figure 5.4, the performance of the distributed control strategies are compared to the centralized control benchmark.[3] For this example, noncooperative control is an improvement over decentralized control (see Table 5.4). Cooperative control with only a single iteration is significantly better than noncooperative control, however, and approaches centralized control as more iteration is allowed.

5.4 Asynchronous Subsystem Communication

The communication in the distributed controller may be changed using the hierarchical control theory of Chapter 4. We show the performance of hierarchical control with three neighborhood configurations. In configuration 1, the reactors are paired in a neighborhood and the

[3] The cost is calculated as Cost $= \sum_{k=0}^{k_{\text{sim}}} \ell(x(k), u(k))$

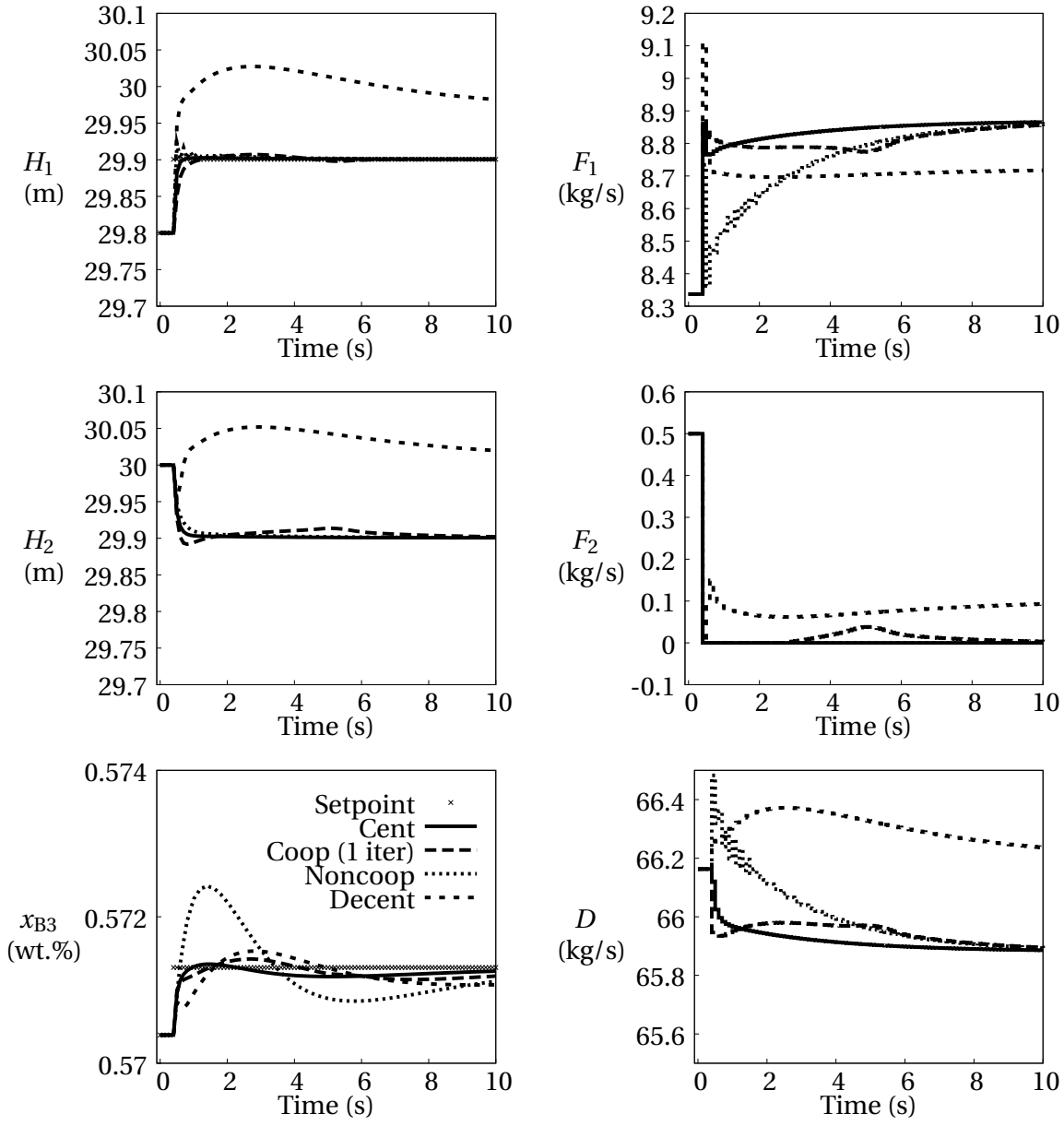


Figure 5.4: Performance of reactor and separator example.

Table 5.5: Hierarchical control performance comparison

	Cost ({1,2}{3})	Loss (%)	Cost ({1}{2,3})	Loss (%)	Cost ({1}{2}{3})	Loss (%)
$N_s = 0$	0.122	21.3	0.122	21.3	0.122	21.3
$N_s = 1$	0.132	31.2	0.133	31.8	0.131	30.5
$N_s = 5$	0.143	42.6	0.136	35.6	0.145	44.3
$N_s = 10$	0.129	28.6	0.122	21.1	0.134	33.1
$N_s = 20$	0.114	13.7	0.108	6.99	0.117	16.3

separator is in its own neighborhood. In configuration 2, the first reactor is in its own neighborhood and the second reactor and separator form the other neighborhood. For configuration 3, each subsystem is in its own neighborhood. Using the same chemical simulation, we vary the communication time schedule in the plant. The performance for three communications times in configuration 3 is given in Figure 5.5. As shown in Table 5.5, it is not possible to generalize the performance loss associated with reducing the communication frequency. Reducing the communication to every other or after every five time steps reduces performance. A reduction to every ten or twenty time steps significantly improves performance. This performance gain shows the benefit of the hierarchical control design.

5.5 Asynchronous Subsystem Optimization

Inside an industrial chemical plant, there are often subsystems with different dynamical time scales. In the slower time scale subsystems, it may be unnecessary or impossible to provide feedback at the fastest sampling time in the plant. Therefore, there is a motivation to allow the subsystem controllers to optimize their control problems at multiple rates. Venkat (2006, Chapter 11) proposed an asynchronous feedback scheme for cooperative control. In this framework, a slow and fast time scale is defined. The slower time scale subsystems implement a zero-order hold while the fast time scale subsystems re-solve their control optimization to provide updated feedback.

The hierarchical control scheme in Chapter 4 also allows this separation of feedback time scales. We divide the set of subsystems into fast and slow feedback sets, namely $\mathbb{I}_{\text{fast}}$ and $\mathbb{I}_{\text{slow}}$

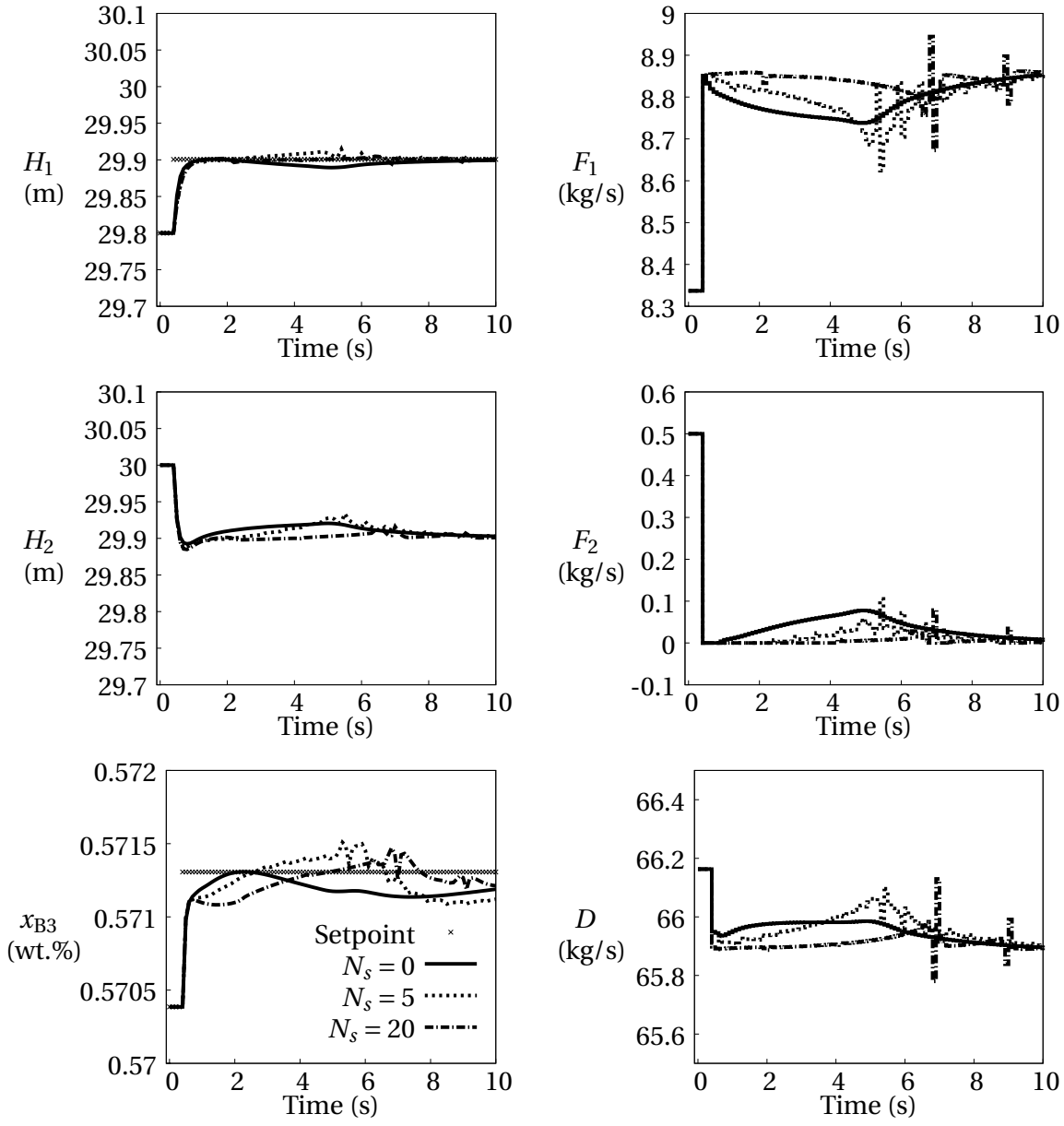


Figure 5.5: Hierarchical control performance of reactor and separator example. Each subsystem is in its own neighborhood.

such that $\mathbb{I}_{\text{fast}} \cup \mathbb{I}_{\text{slow}} = \mathbb{I}_{1:M}$. Let τ be the ratio of the slow to fast sampling times. Each subsystem $i \in \mathbb{I}_{\text{slow}}$ adds the following equality constraints to its control problem (4.4.2)

$$\begin{aligned} u_i(0) &= u_i(1) = \cdots = u(\tau - 1) \\ u_i(\tau) &= u_i(\tau + 1) = \cdots = u(2\tau - 1) \\ &\vdots \\ u_i(N - \tau) &= u_i(N - \tau + 1) = \cdots = u(N - 1) \end{aligned}$$

The synchronization time is $N_s = \tau - 1$. To reduce computation, it is only necessary to use the inputs $\{u_i(0), u_i(\tau), \ldots, u_i(N - \tau)\}$ as decision variables in the control problem. The faster subsystems optimize at the fast time scale. The horizon of the fast subsystems can be reduced to aid implementation by choosing a smaller horizon $N_{\text{fast}} < N - 1$ and using the constraint $u_i(N_{\text{fast}}) = \cdots = u_i(N - 1) = 0$ for $i \in \mathbb{I}_{\text{fast}}$.

In Figure 5.6, we show the performance of the chemical plant from Section 5.3 under asynchronous cooperative control. The reactors optimize and communicate at every time step, while the separator optimizes only every five time steps, i.e., $\tau = 5$. The performance cost of the asynchronous implementation is 0.148, which is 21% higher than fast, synchronous cooperative control.

5.6 Asynchronous Sampling

Chemical plants are often affected by high and low frequency disturbances. High frequency disturbances are rejected by sampling fast responding outputs and taking control action quickly. Disturbances that affect the plant over a longer time scale are rejected by measuring slower responding outputs less frequently. The traditional method for controlling a plant with multiple disturbance time scales is cascade control (Ogunnaike and Ray, 1994, pp.567–570). In cascade control a slow outer-loop controller sends setpoints to a fast inner-loop controller. Because the inner-loop controller's design goal is to track its setpoint, it may reject a disturbance that helps the inner-loop controller, limiting the performance of the plant. This objective mismatch

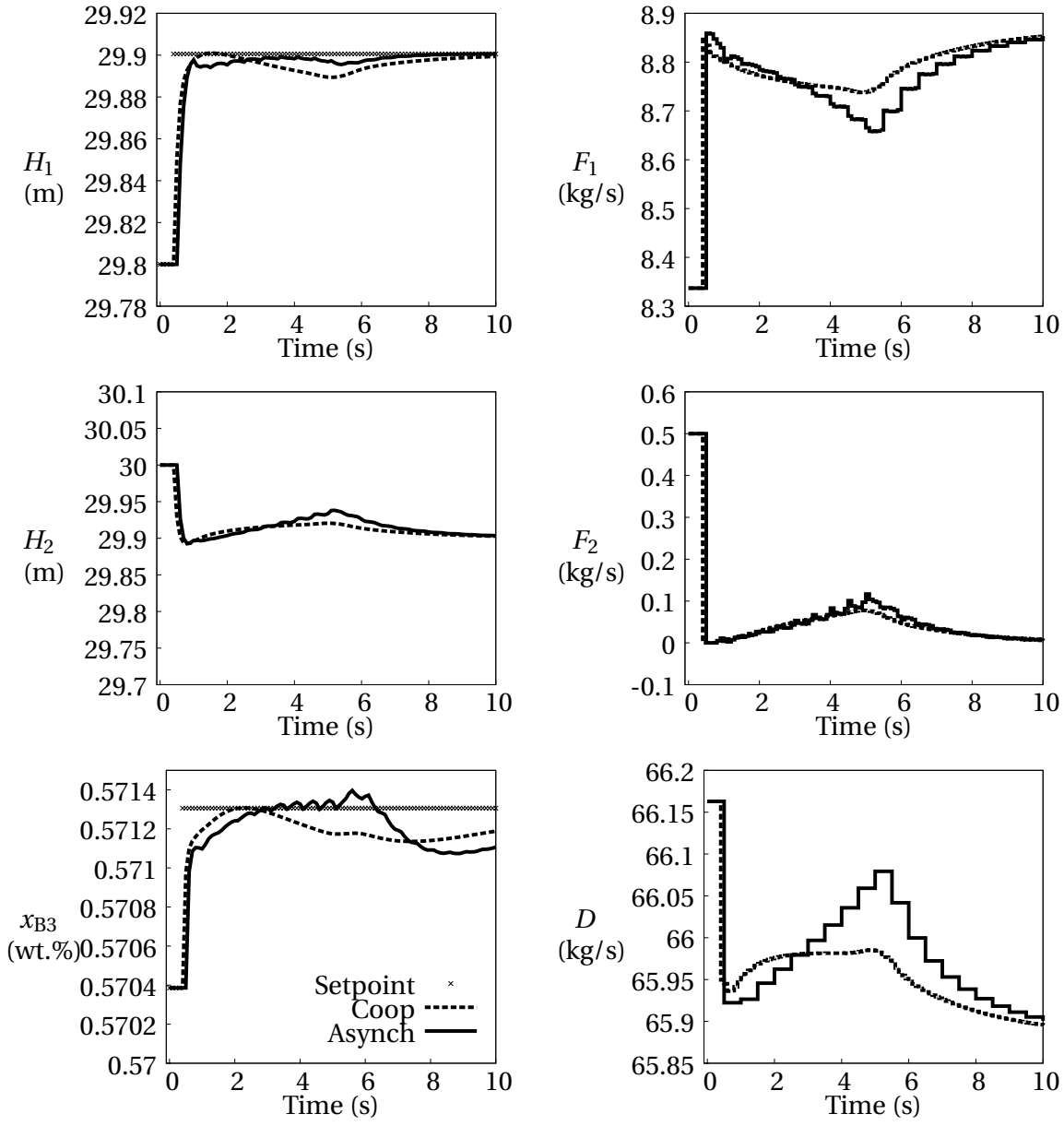
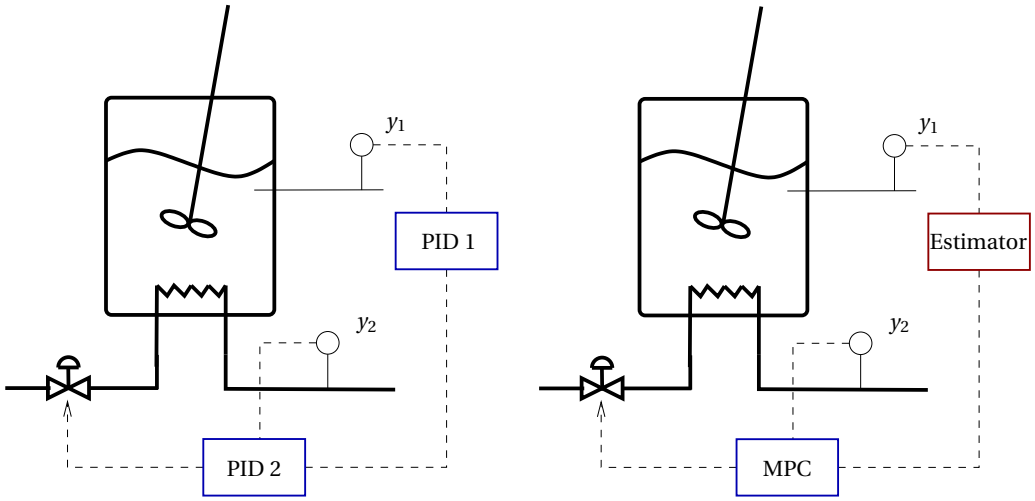


Figure 5.6: Asynchronous control performance of reactor and separator example.



(a) Cascade control

(b) MPC with asynchronous estimation

Figure 5.7: Asynchronous sampling. Traditional cascade control requires two levels of controllers. MPC can be implemented with asynchronous estimation.

is similar to the weakness of noncooperative control. This problem motivates a controller that can reject high frequency disturbances and ensure the slower outputs track their setpoints.

Nagrath, Prasad, and Bequette (2002) show that a single MPC can be used as an alternative to a multi-loop cascade controller (see Figure 5.7). The MPC replaces the fast, inner-loop controller and the outer-loop measurements are sampled on a slower time schedule. Because the outer-loop measurements are used in the MPC, there is no objective mismatch in the controller, and because only one controller is used, the tuning complexity is lower.

5.6.1 One Layer MPC with Delayed Estimation

Consider a single input/two output system

$$x_1^+ = A_1 x_1 + B_1 u \qquad x_2^+ = A_2 x_2 + B_2 u$$

$$y_1 = C_1 x_1 \qquad y_2 = C_2 x_2$$

Without loss of generality, we denote (y_1, u) as the slow system and (y_2, u) as the fast system. The goal of the overall control scheme is to bring y_1 to its setpoint y_{1sp}. The y_2 measurement

is used to detect high frequency input disturbances so that they can be rejected before significantly affecting y_1. It is usually not required to control y_2 to a setpoint. Because the outer-loop measurement is sampled at a slower rate, we require a modification of the estimator defined in Section 3.4. Let y_1 be sampled every τ time steps. Using the same techniques as Rawlings and Mayne (2009, pp.28–32), the outer-loop estimator for this case is defined by

$$\begin{aligned}
\hat{x}_1^- &= A_1^\Delta \hat{x}_1 + \sum_{j=0}^{\Delta-1} A_1^{\Delta-j-1} B_1 u(j) \\
P_1^- &= A_1^\Delta P_1 (A_1^\Delta)' + \sum_{j=0}^{\Delta-1} A_1^{\Delta-j-1} Q_1 (A_1^{\Delta-j-1})' \\
L_1 &= P_1^- C_1' (C_1 P_1^- C_1' + R_1)^{-1} \\
P_1 &= P_1^- - P_1^- C_1' (C_1 P_1^- C_1' + R_1)^{-1} C_1 P_1^- \\
\hat{x}_1 &= \hat{x}_1^- + L_1 \left(y_1 - C_1 \hat{x}_1^- \right)
\end{aligned}$$

5.6.2 Asynchronous Sampling Example

In this example, we compare a traditional cascade controller to MPC with asynchronous sampling. Consider the two output/one input plant with output disturbances

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{-4}{20s+1} \\ \dfrac{-2.5}{5s+1} \end{bmatrix} u$$

The goal of the control in this example is to keep y_1 at its setpoint under all disturbances.

To design the cascade controller, we use the method given by Russo and Bequette (1997). The cascade controller is constructed with a PI controller for the outer loop that controls y_1 by manipulating the setpoint of y_2. The inner-loop controller uses proportional control only and tries to move y_2 to its setpoint by directly manipulating u. The outer loop executes every 50 time steps. The cascade controllers are tuned using the gain of the LQR controller with parameters $Q_{y1} = 1$, $R_1 = 1$, $Q_{y2} = 20$, and $R_2 = 1$. The integrating time constant for the outer loop is $T_i = 0.04$. The MPC is tuned with parameters $Q_y = \text{diag}(1,0)$ and $R = 10$.

To illustrate a performance benefit of MPC over cascade control, we compare their rejection of a disturbance. At time $t = 5s$, a unit input disturbance enters the plant. The cascade controller responds quickly and moves y_2 closer to its setpoint. Because the outer loop does not execute for another 45s, the y_1 continues to drift away from its setpoint until it is sampled. Alternatively, the MPC brings y_1 back to setpoint even before sampling. This example show that the objective mismatch of the loops in cascade control can lead to significant performance losses.

Remark 5.1. *The above arguments are for one MPC, but the same asynchronous sampling framework can be applied to the decentralized estimator in each subsystem.*

Remark 5.2 (Data network)**.** *In order to implement plantwide cooperative control, it is necessary for the subsystem controllers and estimators to communicate. This communication can be accomplished with a plantwide data network to which every subsystem can write and read information. This communication framework contrasts the coordinating optimizations usually required in plantwide control (Scattolini, 2009).*

5.7 Conclusions

In this chapter, we present examples showing the flexibility and practical advantages of cooperative control over traditional plantwide control schemes. The examples show that the subsystems in the plant can communicate on any time schedule, the control problem solved in each subsystem can be solved on any time schedule, and measurements can be sampled asynchronously. Moreover, these design choices can be made independently and do not affect the nominal stability of the plant. We also show that it is not necessary to identify all plantwide subsystem interactions in order to benefit from cooperative control, and that alternatively, the plantwide model can be constructed from first principles.

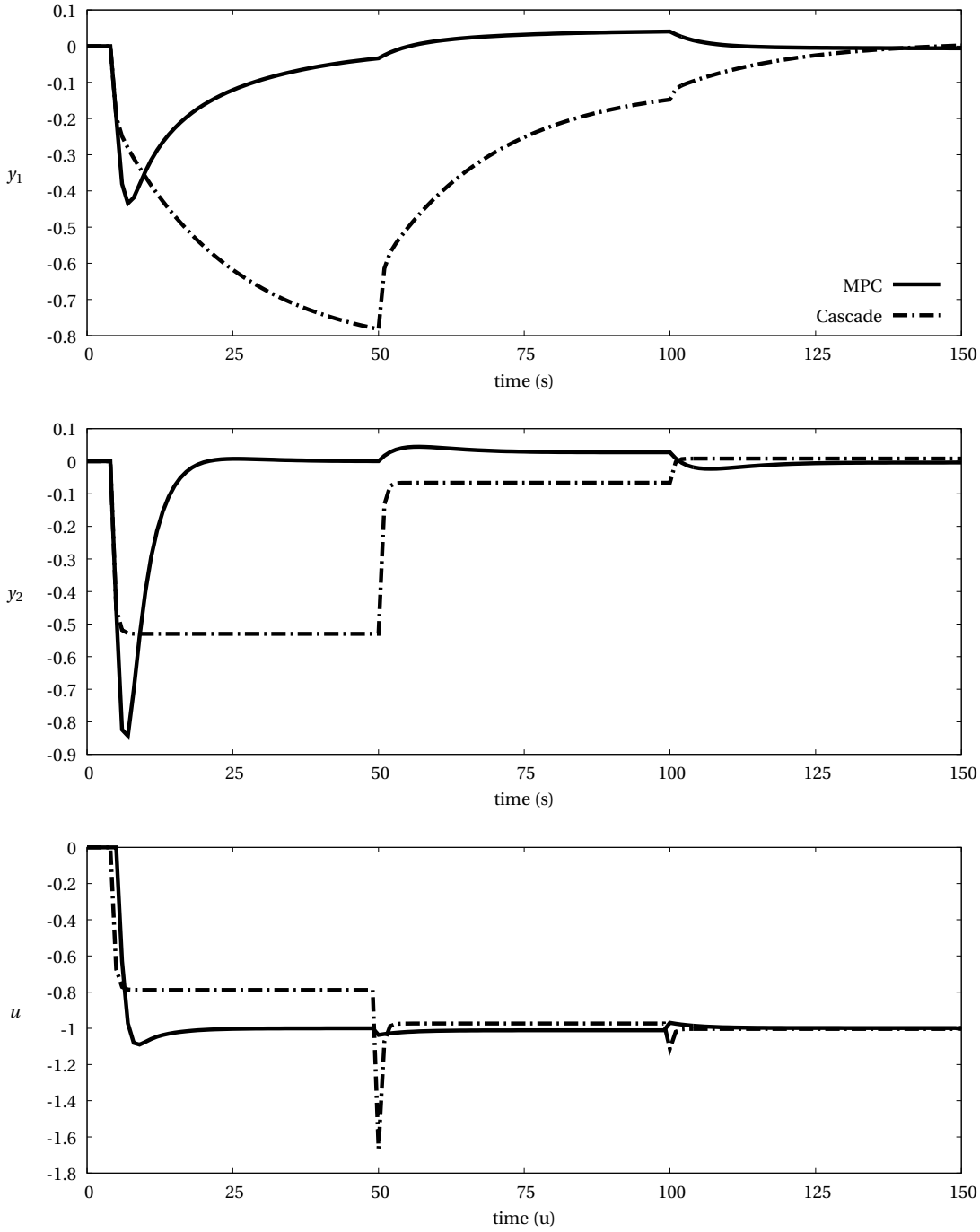


Figure 5.8: Cascade control

Chapter 6

Cooperative Control for Nonlinear Systems

Note: The text of this chapter appears in (Stewart, Wright, and Rawlings, 2010c).

6.1 Introduction

In this chapter, we extend the work in Chapter 1 to nonlinear plants. The main difference is that the plantwide objective function is nonconvex and therefore we propose a novel distributed nonconvex optimization that converges to stationary points without the use a of central coordinating optimization. We avoid a coordinating optimization based on the following two criteria for the plantwide control optimization: (i) the optimizers should *not* rely on a central coordinator and (ii) the exchange of information between the subsystems and the iteration of the subsystem optimizations should be able to terminate before convergence without compromising closed-loop properties. The first criterion is motivated by the practicality of industrial distributed control. Distributed control strategies are used for plants in which centralized control is often impractical or undesirable to implement, and a plantwide coordination layer is likely as difficult to implement as centralized control. The second criterion is motivated by the implementation of distributed control. A plantwide control strategy should be robust to communication disruptions and algorithm failures. Therefore these strategies cannot rely on iteration convergence in order to have an implementable input. In the absence of either of these properties, the alternative is usually centralized control.

The statement of the optimization follows in the next section. In Section 6.3, we present a distributed controller that uses the nonconvex optimization and show that it is asymptotically stable. We then present an illustrative example and follow with conclusions.

6.2 Distributed Nonconvex Optimization

Consider the optimization

$$\min_u V(u) \quad \text{s.t.} \quad u \in \mathbb{U} \tag{6.2.1}$$

in which $u \in \mathbb{R}^m$ and $V(\cdot) : \mathbb{R}^m \to \mathbb{R}_+$ is twice continuously differentiable. We assume $\mathbb{U}$ is closed, convex, and can be separated into M orthogonal subspaces such that $\mathbb{U} = \mathbb{U}_1 \times \cdots \times \mathbb{U}_M$, for which $\mathbb{U}_i \in \mathbb{R}^{m_i}$ for all $i \in \mathbb{I}_{1:M}$. We require approximate solutions to the following suboptimizations at iterate $p \geq 0$ for all $i \in \mathbb{I}_{1:M}$

$$\min_{u_i \in \mathbb{U}_i} V(u_i, u_{-i}^p)$$

in which $u_{-i} = (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_M)$. Let the approximate solution to these optimizations be $\overline{u}_i^p$. In the proposed algorithm, we compute the approximate solutions via line search with gradient projection. At iterate $p \geq 0$

$$\overline{u}_i^p = \mathscr{P}_i(u_i^p - \nabla_i V(u^p)) \tag{6.2.2}$$

in which $\nabla_i V(u^p)$ is the ith component of $\nabla V(u^p)$ and the function $\mathscr{P}_i(\cdot)$ denotes the projection onto the set $\mathbb{U}_i$. Define the step $v_i^p = \overline{u}_i^p - u_i^p$. To choose the stepsize α_i^p, each suboptimizer initializes the stepsize with $\overline{\alpha}_i$ and then uses backtracking with a factor of $\beta \in (0,1)$ until α_i^p satisfies the Armijo rule (Bertsekas, 2008, p.230)

$$V(u^p) - V(u_i^p + \alpha_i^p v_i^p, u_{-i}^p) \geq -\sigma \alpha_i^p \nabla_i V(u^p)' v_i^p \tag{6.2.3}$$

in which $\sigma \in (0,1)$. After all suboptimizers finish the backtracking process, they exchange steps. Each suboptimizer forms a candidate step

$$u_i^{p+1} = u_i^p + w_i \alpha_i^p v_i^p \quad \forall i \in \mathbb{I}_{1:M} \tag{6.2.4}$$

and checks the following inequality, which tests if $V(u^p)$ is convex-like

$$V(u^{p+1}) \leq \sum_{i \in \mathbb{I}_{1:M}} w_i V(u_i^p + \alpha_i^p v_i^p, u_{-i}^p) \tag{6.2.5}$$

in which $\sum_{i \in \mathbb{I}_{1:M}} w_i = 1$ and $w_i > 0$ for all $i \in \mathbb{I}_{1:M}$. If condition (6.2.5) is not satisfied, then we find the direction with the worst cost improvement $i_{\max} = \arg\max_i \{V(u_i^p + \alpha_i^p v_i^p, u_{-i}^p)\}$, and eliminate this direction by setting $w_{i_{\max}}$ to zero and repartitioning the remaining w_i so that they sum to 1. We then reform the candidate step (6.2.4) and check condition (6.2.5) again. We repeat until (6.2.5) is satisfied. At worst, condition (6.2.5) is satisfied with one direction only. The steps are formalized in Algorithm 1.

Remark 6.1. *In Chapter 3, we proposed a similar distributed algorithm for a convex optimization. The main difference in the nonconvex case is that poor suboptimizer steps must be eliminated to ensure the objective function decreases at each iterate.*

Remark 6.2 (Distributed)**.** *The test of inequality* (6.2.5) *does not need a coordinator. At each optimization iterate the subsystems exchange the solutions of the gradient projection. Each subsystem has a copy of the plantwide model and can evaluate the objection function independently. Therefore the while-loop in Algorithm 1, which is a series of conditional statements without optimization, can be run on each controller. This computation is likely a smaller overhead than a coordinating optimization.*

Lemma 6.3 (Feasibility)**.** *Given a feasible initial condition, the iterates u^p are feasible for all $p \geq 0$.*

Lemma 6.4 (Objective decrease)**.** *The objective function decreases at every iterate*

$$V(u^{p+1}) \leq V(u^p)$$

Lemma 6.5 (Convergence)**.** *Every accumulation point of the sequence $\{u^p\}$ is stationary.*

The proofs of Lemmas 6.3 and 6.4 follow by construction of the algorithm. We give the proof of Lemma 6.5 after establishing some preliminary propositions.

Algorithm 1 Distributed gradient projection

Given finite $\overline{p}$, $0 < \sigma < 1$, and $\overline{w}_i > 0$ for all $i \in \mathbb{I}_{1:M}$ such that $\sum_{i \in \mathbb{I}_{1:M}} \overline{w}_i = 1$.

for $p = 0, 1, \ldots, \overline{p}$ **do**
  for $i \in \mathbb{I}_{1:M}$ **do**
    Compute $\overline{u}_i^p$ using (6.2.2);
    Find α_i^p satisfying (6.2.3);
    $V_i^p \leftarrow V(u_i^p + \alpha_i^p v_i^p, u_{-i}^p)$;
  $v^p \leftarrow (v_1^p, \ldots, v_M^p)$;
  $k \leftarrow 1$, $\mathbb{I}_{\text{good}} \leftarrow \mathbb{I}_{1:M}$, $w_i \leftarrow \overline{w}_i$
  while $k < M$ **do**
    for $i \in \mathbb{I}_{1:M}$ **do**
      $u_i^{p+1} \leftarrow u_i^p + w_i \alpha_i^p v_i^p$;
    if u^{p+1} satisfies (6.2.5) **then**
      break;
    else
      $i_{\max} \in \arg\max_{i \in \mathbb{I}_{1:M}} \{V_i^p\}$;
      $\mathbb{I}_{\text{good}} \leftarrow \mathbb{I}_{\text{good}} \setminus i_{\max}$;
      $w_{i_{\max}} \leftarrow 0$;
      $\overline{w} \leftarrow \sum_{j \in \mathbb{I}_{\text{good}}} w_j$;
      for $i \in \mathbb{I}_{1:M}$ **do**
        $w_i \leftarrow w_i / \overline{w}$;
    $k \leftarrow k + 1$;

Proposition 6.6. *Given a closed, convex set $\mathbb{U}$ with any $y \in \mathbb{U}$ and any $z \in \mathbb{R}^m$, for the projection $\mathscr{P}(\cdot)$ onto $\mathbb{U}$*

$$(y-z)'(y-\mathscr{P}(z)) \geq 0$$

with equality if and only if $y = \mathscr{P}(z)$.

Proposition 6.7. *If $\hat{u}$ is nonstationary*

$$\nabla_i V(\hat{u})'\left[\hat{u}_i - \mathscr{P}_i(\hat{u}_i - \nabla_i V(\hat{u}))\right] \geq 0 \quad i \in \mathbb{I}_{1:M}$$

with strict inequality for at least one $i \in \mathbb{I}_{1:M}$.

Proof. Set $y = \hat{u}_i$ and $z = \hat{u}_i - \nabla_i V(\hat{u})$ in Proposition 6.6 to prove the first claim. To show the second claim, observe that if equality holds for all $i \in \mathbb{I}_{1:M}$, then from Proposition 6.6 we would have

$$\hat{u}_i = \mathscr{P}_i(\hat{u}_i - \nabla_i V(\hat{u})) \quad \forall\, i \in \mathbb{I}_{1:M}$$

and therefore $\hat{u} = \mathscr{P}(\hat{u} - \nabla V(\hat{u}))$, and $\hat{u}$ would be stationary. □

Proposition 6.8. *Suppose $\hat{u}$ is a nonstationary point. Then there are positive constants ρ and ϵ and an index $i \in \mathbb{I}_{1:M}$ such that for all u with $|u - \hat{u}| \leq \rho$, the ith suboptimizer chooses stepsize α_i for which*

$$V(u_i, u_{-i}) - V(u_i + \alpha_i v_i, u_{-i}) \geq \epsilon$$

Proof. Let i be an index such that strict inequality holds in Proposition 6.7. Using the continuity of $\nabla_i V(\cdot)$ and $\mathscr{P}_i(\cdot)$, define $\rho > 0$ and $\epsilon_i > 0$ such that

$$-\nabla_i V(u)' v_i = \nabla_i V(u)'\left[u_i - \mathscr{P}_i(u_i - \nabla_i V(u))\right] \geq \epsilon_i$$

for all u with $|u-\hat{u}| \le \rho$. From Taylor's theorem (Nocedal and Wright, 2006, p. 14), and using continuity of $\nabla_i V(\cdot)$, there is an $\hat{\alpha}_i > 0$ such that for all $\alpha_i \in [0, \hat{\alpha}_i]$

$$\begin{aligned} & V(u_i, u_{-i}) - V(u_i + \alpha_i v_i, u_{-i}) = \\ & \quad - \alpha_i \nabla_i V(u_i, u_{-i})' v_i - \alpha_i \left[\nabla_i V(u_i + t\alpha_i v_i, u_{-i}) - \nabla_i V(u_i, u_{-i})\right]' v_i \\ & = - \alpha_i \nabla_i V(u_i, u_{-i})' v_i + o(\alpha_i) \\ & \ge - \sigma \alpha_i \nabla_i V(u_i, u_{-i})' v_i \end{aligned} \tag{6.2.6}$$

in which $\hat{\alpha}_i$ is small enough to ensure that the remainder term satisfies

$$o(\alpha_i) \le -(1-\sigma)\alpha_i \nabla_i V(u_i, u_{-i})' v_i$$

a strictly positive multiple of α_i. Hence the backtracking process terminates at a value α_i greater than or equal to $\underline{\alpha}_i = \min(\overline{\alpha}_i, \beta\hat{\alpha}_i) > 0$. Hence, from (6.2.6), we have

$$V(u_i, u_{-i}) - V(u_i + \alpha_i v_i, u_{-i}) \ge -\sigma \underline{\alpha}_i \nabla_i V(u_i, u_{-i})' v_i \ge \sigma \underline{\alpha}_i \epsilon_i > 0$$

Therefore the Proposition holds with $\epsilon = \sigma \underline{\alpha}_i \epsilon_i$. □

We now proceed with the proof of the convergence result.

Proof of Lemma 6.5. Toward a contradiction, suppose that $\hat{u}$ is a nonstationary point, and let K be a subsequence such that $\{u^p\}_{p\in K} \to \hat{u}$. By taking a further subsequence if necessary, we have from Proposition 6.8 that there is an index i and a positive constant ϵ such that

$$V(u_i^p, u_{-i}^p) - V(u_i^p + \alpha_i^p v_i^p, u_{-i}^p) \ge \epsilon$$

for all $p \in K$. Let j^p be the index in $\mathbb{I}_{1:M}$ that attains the *best* decrease on V at iterate p. Since there are only finitely many possible values for j^p, at least one of them must recur infinitely often. By taking a further subsequence we can assure $j^p \equiv j$ for some $j \in \mathbb{I}_{1:M}$. We thus have

$$V(u^p) - V(u_j^p + \alpha_j^p v_j^p, u_{-j}^p) \ge V(u^p) - V(u_i + \alpha_i^p v_i^p, u_{-i}^p) \ge \epsilon$$

for all $p \in K$. Moreover, the index j remains in the set $\mathbb{I}_{\text{good}}$ for all inner iterations, at each major iteration $p \in K$. Since all terms in the summation on the right-hand side of (6.2.5) are

nonnegative and $w_j \geq \overline{w}_j > 0$, using (6.2.7), the right-hand side is bounded below by $\overline{w}_j \epsilon > 0$. Therefore

$$V(u^p) - V(u^{p+1}) \geq \overline{w}_j \epsilon > 0 \quad \forall\, p \in K$$

for which $\overline{w}_j \epsilon$ does not depend on p. This inequality implies that $V(u^p) \to -\infty$ over the *entire* sequence $\{u^p\}$, since $V(u^p)$ decreases at every iteration. This contradicts $\lim_{p \in K} V(u^p) = V(\hat{u})$, and the proof is complete. □

6.2.1 Example from Rawlings and Mayne

Consider the nonconvex function

$$V(u_1, u_2) = e^{-2u_1} - 2e^{-u_1} + e^{-2u_2} - 2e^{-u_2}$$
$$+ a \exp(-\beta((u_1 + 0.2)^2 + (u_2 + 0.2)^2))$$

in which $a = 1.1$ and $\beta = 0.4$ (Rawlings and Mayne, 2009, p.462). There are two global minimum located at $(0.007, 2.28)$ and $(2.28, 0, 007)$ and a local minimum at $(0.23, 0.23)$. The inputs are constrained such that $0.1 \leq u_i \leq 4$ for $i \in \mathbb{I}_{1:2}$. We start the algorithm at three initial conditions $(0.5, 0.5)$, $(3.9, 3.6)$ and $(3.5, 3.9)$. As shown in Figure 6.1, each of these points converges to a different local minimum.

6.3 Distributed Nonlinear Cooperative Control

In this section, we propose a controller that uses the distributed optimization described in the previous section. To facilitate the exposition, we assume the plant comprises only two subsystems.

6.3.1 Problem Formulation (Ideal Case)

In order to motivate the theory of our approach to distributed control, consider the ideal case in which each controller can optimize and exchange information infinitely quickly. In the sequel, we propose a controller with the goal of replicating the properties of the ideal case as closely as possible without relaxing the basic properties of the controller.

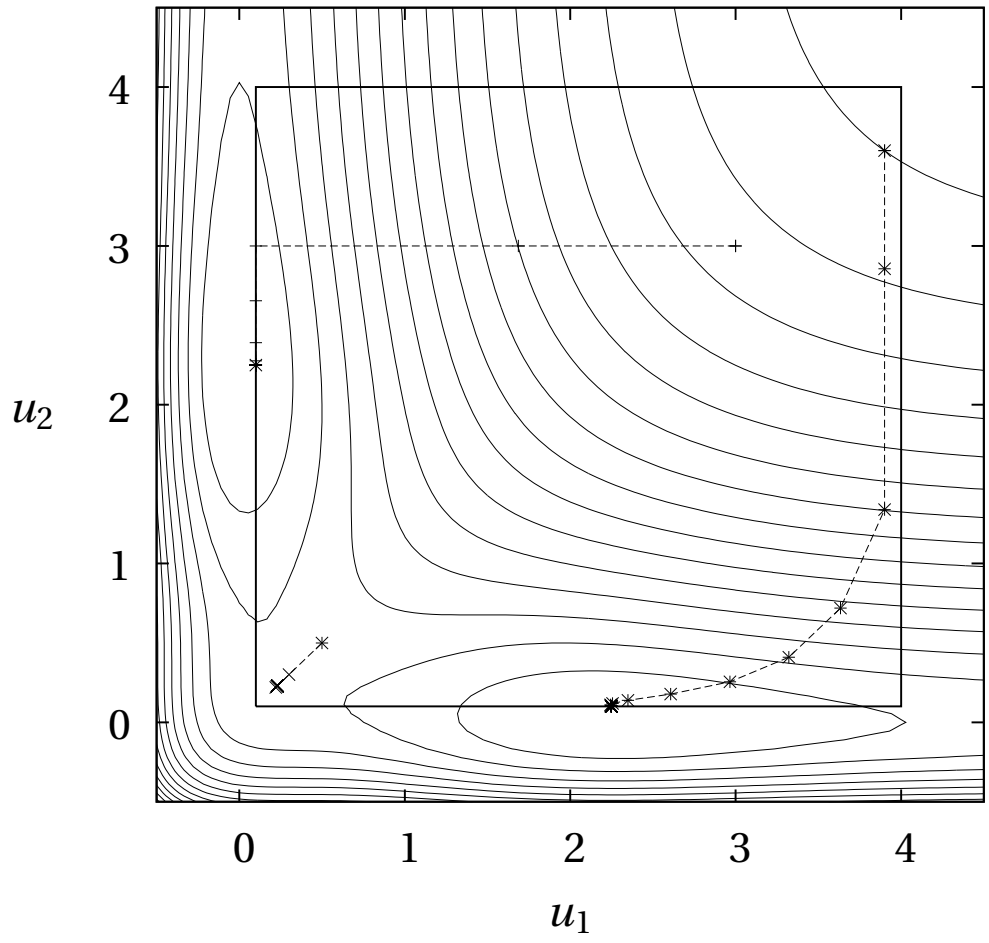


Figure 6.1: Nonconvex function presented in Rawlings and Mayne (2009) optimized with Algorithm 1

In general, we have models of the form

$$\dot{x}_1 = f_1(x_1, x_2, u_1, u_2) \qquad \dot{x}_2 = f_2(x_1, x_2, u_1, u_2)$$

and subsystem objective functions

$$V_1^\infty\big(x_1, x_2, u_1, u_2\big) = \int_0^\infty \ell_1\big(x_1(t), u_1(t)\big)\,dt \qquad V_2^\infty\big(x_1, x_2, u_1, u_2\big) = \int_0^\infty \ell_2\big(x_2(t), u_2(t)\big)\,dt$$

We define the plantwide objective function

$$V^\infty\big(x_1, x_2, u_1(\cdot), u_2(\cdot)\big) = \rho_1 V_1^\infty\big(x_1, x_2, u_1(\cdot), u_2(\cdot)\big) + \rho_2 V_2^\infty\big(x_1, x_2, u_1(\cdot), u_2(\cdot)\big)$$

At time t, the subsystems solve the following optimizations

$$\begin{aligned}
&\min_{u_1(\cdot)} \; V^\infty\big(x_1(0), x_2(0), u_1(\cdot), u_2\big) && \min_{u_2(\cdot)} \; V^\infty\big(x_1(0), x_2(0), u_1, u_2(\cdot)\big) \\
&\qquad \text{s.t.} \quad u_1(t) \in \mathbb{U}_1 \quad \forall\, t \geq 0 && \qquad \text{s.t.} \quad u_2(t) \in \mathbb{U}_2 \quad \forall\, t \geq 0 \\
&\qquad u_2(t) = u_2^0(t; x_1, x_2) \quad \forall\, t \geq 0 && \qquad u_1(t) = u_1^0(t; x_1, x_2) \quad \forall\, t \geq 0 \\
&\dot{x}_1 = f_1(x_1(t), x_2(t), u_1(t), u_2(t)) \quad \forall\, t \geq 0 && \dot{x}_1 = f_1(x_1(t), x_2(t), u_1(t), u_2(t)) \quad \forall\, t \geq 0 \\
&\dot{x}_2 = f_2(x_1(t), x_2(t), u_1(t), u_2(t)) \quad \forall\, t \geq 0 && \dot{x}_2 = f_2(x_1(t), x_2(t), u_1(t), u_2(t)) \quad \forall\, t \geq 0
\end{aligned}$$

in which $\mathbb{U}_1$ and $\mathbb{U}_2$ are the input constraints. We denote the solution to these problems as $u_1^0(\cdot; x_1, x_2)$ and $u_2^0(\cdot; x_1, x_2)$, respectively. This controller has several properties. The decision space of each controller is reduced to a subset of the inputs. The controller feedback converges to the centralized optimal feedback infinitely quickly. Notice that in the ideal case the control law is a function of both states, and hence initial states must be exchanged between controllers.

The assumptions of infinitely fast optimization and communication and infinite dimensional decision variables are unrealistic, however. We wish to replicate the properties of this ideal controller but weaken these assumptions. Therefore we use discrete time models, a finite number of optimization and communication steps, and a finite decision space. For the proposed controller, we retain the (ideal) properties of reducing the decision space of each controller and having an implementable, stabilizing input at each sampling time. Notice that the

same compromise exists in the centralized case when comparing an ideal infinite horizon controller to an implementable discrete-time finite horizon control. Notice also that the choice of a finite number of iterates of the optimization algorithm implies the distributed controller is suboptimal. We now give the theory of the proposed distributed controller.

6.3.2 Model

We assuming the following models exist

$$x_1^+ = f_1(x_1, x_2, u_1, u_2) \qquad x_2^+ = f_2(x_1, x_2, u_1, u_2) \tag{6.3.1}$$

in which $x_i \in \mathbb{R}^{n_i}$, $u_i \in \mathbb{R}^{m_i}$, and $f_i : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \to \mathbb{R}^{n_i}$ is continuous such that $f_i(0) = 0$ for all $i \in \mathbb{I}_{1:2}$. We collect these models to form the plantwide model

$$x^+ = f(x_1, x_2, u_1, u_2) = f(x, u)$$

in which

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad f(x,u) = \begin{bmatrix} f_1(x_1, x_2, u_1, u_2) \\ f_2(x_1, x_2, u_1, u_2) \end{bmatrix}$$

for which $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$.

6.3.3 Constraints

At each time step k, we require the inputs to satisfy

$$u_1(k) \in \mathbb{U}_1 \qquad u_2(k) \in \mathbb{U}_2 \quad k \in \mathbb{I}_{0:N-1}$$

in which each $\mathbb{U}_i \in \mathbb{R}^{m_i}$ is compact, convex, and contains the origin in its interior.

6.3.4 Objective Functions

Usually in distributed control implementations an objective function is defined for each subsystem. We construct the plantwide objective function from these objectives. For each subsystem $i \in \mathbb{I}_{1:2}$, we denote the positive definite function $\ell_i(x_i, u_i)$ as the stage cost such that

$\ell_i(0,0) = 0$ and $V_{if}(x)$ as the terminal cost such that $V_{if}(0) = 0$. The objective function for each subsystem $i \in \mathbb{I}_{1:2}$ is defined

$$V_i\big(x(0), \mathbf{u}_1, \mathbf{u}_2\big) = \sum_{k=0}^{N-1} \ell_i\big(x_i(k), u_i(k)\big) + V_{if}\big(x(N)\big)$$

in which $\mathbf{u}_i = \{u_i(0), \ldots, u_i(N-1)\} \in \mathbb{R}^{Nm_i}$, $x_i(k) = \phi_i(k; x_i, \mathbf{u}_1, \mathbf{u}_2)$, and $N > 0$. Because x_i is a function of both u_1 and u_2, V_i is implicitly a function of both $\mathbf{u}_1$ and $\mathbf{u}_2$. We define plantwide objective

$$V\big(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2\big) = \rho_1 V_1\big(x(0), \mathbf{u}_1, \mathbf{u}_2\big) + \rho_2 V_2\big(x(0), \mathbf{u}_1, \mathbf{u}_2\big)$$

in which $\rho_1, \rho_2 > 0$ are weighting factors. To simplify notation we use $V(x, \mathbf{u})$ for the plantwide objective.

Remark 6.9. *Alternatively, the plantwide objective function can be defined without reference to subsystem objective functions.*

Assumption 6.10. *For each $i \in \mathbb{I}_{1:2}$, there exists a $\mathcal{K}_\infty$ function $\alpha_i(\cdot)$ such that*

$$\ell_i(x_i, u_i) \geq \alpha_i(|x_i|) \quad \forall (x_i, u_i) \in \mathbb{R}^{n_i} \times \mathbb{U}_i \tag{6.3.2}$$

6.3.5 Terminal Controller

Denote the plantwide terminal penalty $V_f(x) = \rho_1 V_{1f}(x) + \rho_2 V_{2f}(x)$. We define the terminal region $\mathbb{X}_f$ to be a sublevel set of V_f. For $a > 0$, define

$$\mathbb{X}_f = \{x \mid V_f(x) \leq a\}$$

Assumption 6.11. *The plantwide terminal penalty $V_f(\cdot)$ satisfies*

$$\alpha_f(|x|) \leq V_f(x) \leq \gamma_f(|x|) \quad \forall x \in \mathbb{X}_f$$

in which $\alpha_f(\cdot)$ and $\gamma_f(\cdot)$ are $\mathcal{K}_\infty$ functions.

Defining $\ell(x, u) = \rho_1 \ell_1(x_1, u_1) + \rho_2 \ell_2(x_2, u_2)$, we require the following stability assumption.

Assumption 6.12. *The terminal cost $V_f(\cdot)$ satisfies*

$$\min_{(u_1,u_2)\in\mathbb{U}_1\times\mathbb{U}_2}\Big\{V_f\big(f(x,u_1,u_2)\big)+\ell(x,u)$$
$$s.t.\quad f(x,u_1,u_2)\in\mathbb{X}_f\Big\}\leq V_f(x)\quad\forall x\in\mathbb{X}_f$$

This assumption implies that for $x\in\mathscr{X}_f$ there exists a $\kappa_{if}(x)\in\mathbb{U}_i$ for all $i\in\mathbb{I}_{1:2}$ such that

$$V_f\big(f(x,\kappa_{1f}(x),\kappa_{2f}(x))\big)+\ell\big(x,\kappa_{1f}(x),\kappa_{2f}(x)\big)\leq V_f(x) \tag{6.3.3}$$
$$\text{s.t.}\quad f(x,\kappa_{1f}(x),\kappa_{2f}(x))\in\mathbb{X}_f$$

Each terminal controller $\kappa_{if}(\cdot)$ may be found via a centralized calculation offline. We next provide an example of such a terminal control law.

6.3.5.1 Distributed Terminal Control Example

In this example, we make a linear approximation of the nonlinear model around the origin and find a stabilizing linear control law. Let $f(\cdot)$ and $\ell(\cdot)$ be Lipschitz continuous in a neighborhood of the origin. Define $A=\nabla_x f(0,0)$, $B=\nabla_u f(0,0)$, $Q=\nabla^2_{xx}\ell(0,0)$, $R=\nabla^2_{uu}\ell(0,0)$, and $S=\nabla^2_{xu}\ell(0,0)$. Denote P_f as the solution to the centralized discrete time Riccati equation

$$P_f=A'P_fA+Q-(A'P_fB+S)(R+B'P_fB)^{-1}(B'P_fA+S')$$

and terminal controller gain K as

$$K=-(R+B'P_fB)^{-1}(B'P_fA+S')$$

In the terminal region, the unconstrained control law $u=Kx$ is used. Defining the stable matrix $A_K=(A+BK)$ and $Q^*=(Q+K'RK)$, let the matrix P satisfy the Lyapunov equation $A_K'PA_K+2Q^*=P$. Following (Rawlings and Mayne, 2009, pp.135-137), there exists an $a\in(0,\infty)$ such that

$$V_f(f(x,\kappa_{1f}(x),\kappa_{2f}(x)))+\frac{1}{2}x'Q^*x-V_f(x)\leq 0\quad\forall x\in W(a)$$

in which $W(a)=\{x\mid V_f(x)\leq a\}$

$$\kappa_{1f}(x_1,x_2)=K_{11}x_1+K_{12}x_2$$
$$\kappa_{2f}(x_1,x_2)=K_{21}x_1+K_{22}x_2$$

$$V_{1f}(x_1, x_2) = \frac{1}{2} x_1' P_{11} x_1 + \frac{1}{2} x_1' P_{12} x_2 \tag{6.3.4a}$$

$$V_{2f}(x_1, x_2) = \frac{1}{2} x_2' P_{21} x_2 + \frac{1}{2} x_2' P_{22} x_2 \tag{6.3.4b}$$

and

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \quad K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

We then define the terminal set $\mathbb{X}_f = W(a)$.

Remark 6.13. *For systems in which it is undesirable or impossible to calculate the centralized K and P matrices, a decentralized terminal controller can be used with the trade-off that $\mathbb{X}_f$ is smaller than for the centralized terminal controller case.*

6.3.6 Removing the Terminal Constraint in Suboptimal MPC

To show stability, we must ensure that $\phi(N; x, \mathbf{u}) \in \mathbb{X}_f$. Imposing a terminal constraint on the state, however, requires the use of coupled input constraints in each suboptimization of cooperative MPC. Such a constraint, in general, does not allow the distributed algorithm to converge to the optimal plantwide control feedback without a coordinator (see Remark 3.19). This terminal constraint can be removed from the control problem by modifying the terminal penalty, however. In the following, we show this feature for the general suboptimal MPC case (Rawlings and Mayne, 2009, pp.155–158), and note that the proposed distributed controller is of this class.

For some $\beta \geq 1$, let the objective function be defined

$$V^{\beta}(x, \mathbf{u}) = \sum_{k=0}^{N-1} \ell(x(k), u(k)) + \beta V_f(x(N)) \tag{6.3.5}$$

Define the set of admissible initial $(x, \mathbf{u})$ pairs as

$$\mathbb{Z}_0 = \{(x, \mathbf{u}) \in \mathbb{X} \times \mathbb{U}^N \mid V^{\beta}(x, \mathbf{u}) \leq \overline{V},\ \phi(N; x, \mathbf{u}) \in \mathbb{X}_f\} \tag{6.3.6}$$

in which $\overline{V} > 0$ is an arbitrary constant and $\mathbb{X} = \mathbb{R}^n$. Then the set of initial states $\mathbb{X}_0$ is the projection of $\mathbb{Z}_0$ onto $\mathbb{X}$

$$\mathbb{X}_0 = \{x \in \mathbb{X} \mid \exists \mathbf{u} \text{ such that } (x, \mathbf{u}) \in \mathbb{Z}_0\}$$

Proposition 6.14 (Terminal constraint satisfaction)**.** *Let* $\{(x(k), \mathbf{u}(k)) \mid k \in \mathbb{I}_{\geq 0}\}$ *denote the set of states and control sequences generated by the suboptimal system. There exists a* $\overline{\beta} > 1$ *such that for all* $\beta \geq \overline{\beta}$*, if* $(x(0), \mathbf{u}(0)) \in \mathbb{Z}_0$*, then* $(x(k), \mathbf{u}(k)) \in \mathbb{Z}_0$ *with* $\phi(N; x(k), \mathbf{u}(k)) \in \mathbb{X}_f$ *for all* $k \in \mathbb{I}_{\geq 0}$*.*

Proof. The proof is by induction.[1] We show that there is a finite value $\overline{\beta}$ such that the following property holds for all $\beta \geq \overline{\beta}$: For any state and input sequence $(x, u) \in \mathbb{Z}_0$, the successor state and input sequence $(x^+, u^+) \in \mathbb{Z}_0$. The successor state is $x^+ = f(x, u(0))$ and the warm start is

$$\tilde{\mathbf{u}}^+ = \{u(1), u(2), \ldots, u(N-1), \kappa_f\big(x(N)\big)\}$$

We know that $\tilde{\mathbf{u}}^+ \in \mathbb{U}^N$ because $\kappa(x(N)) \in \mathbb{U}$ for $x(N) \in \mathbb{X}_f$. We also have from the properties of $\kappa_f(\cdot)$ that $\phi(N; x^+, \tilde{\mathbf{u}}^+) \in \mathbb{X}_f$ and $V^\beta(x^+, \tilde{\mathbf{u}}^+) \leq \overline{V}$ by (6.3.3). Next consider any control sequence $\boldsymbol{v} \in \mathbb{U}^N$ meeting the suboptimal MPC cost requirement

$$V^\beta(x^+, \boldsymbol{v}) \leq V^\beta(x^+, \tilde{\mathbf{u}}^+)$$

Expanding the cost function on the left and using the bound on the right gives

$$\sum_{i=0}^{N-1} \ell(z(i), v(i)) + \beta V_f(z(N)) \leq \overline{V}$$

in which $z(i) = \phi(i; x^+, \boldsymbol{v})$. This inequality implies

$$\beta V_f(z(N)) \leq \overline{V}$$

and if we choose

$$\beta \geq \overline{\beta} = \max(1, \overline{V}/a)$$

we obtain $V_f(z(N)) \leq a$, which implies that $z(N) \in \mathbb{X}_f$. We have found a finite value of $\overline{\beta}$ such that the terminal state corresponding to any admissible $\mathbf{u}^+$ from state x^+ lies in $\mathbb{X}_f$ for $\beta \geq \overline{\beta}$. By induction, since $(x(0), \mathbf{u}(0)) \in \mathbb{Z}_0$, $(x(k), \mathbf{u}(k)) \in \mathbb{Z}_0$ for all $k \in \mathbb{I}_{\geq 0}$, and the result is established. □

[1] The proof of this Proposition is by Rawlings, Stewart, Wright, and Mayne (2010)

For the remainder of the chapter, we replace the plantwide objective with the modified objective $V(\cdot) \leftarrow V^{\overline{\beta}}(\cdot)$ and hence the terminal constraint is satisfied.

6.3.7 Cooperative Control Algorithm

Let $\tilde{\mathbf{u}} \in \mathbb{U}$ be the initial condition for the cooperative MPC algorithm such that $\phi(N; x(0), \tilde{\mathbf{u}}) \in \mathbb{X}_f$. At each iterate p, an approximate solution of the following optimization problem is found

$$\min_{\mathbf{u}} \; V\big(x_1(0), x_2(0), \mathbf{u}_1, \mathbf{u}_2\big) \tag{6.3.7a}$$

$$\text{s.t.} \quad x_1^+ = f_1(x_1, x_2, u_1, u_2) \tag{6.3.7b}$$

$$x_2^+ = f_2(x_1, x_2, u_1, u_2) \tag{6.3.7c}$$

$$\mathbf{u}_i \in \mathbb{U}_i^N \quad \forall\, i \in \mathbb{I}_{1:2} \tag{6.3.7d}$$

$$|\mathbf{u}_i| \le \delta_i(|x_i(0)|) \quad \text{if } x(0) \in \mathbb{B}_r \quad \forall\, i \in \mathbb{I}_{1:2} \tag{6.3.7e}$$

in which $\delta_i(\cdot)$ is a $\mathcal{K}_\infty$ function and $r > 0$ can be chosen as small as required. Constraint (6.3.7e) is needed for stability and is motivated in the sequel. We can write (6.3.7) in the form of (6.2.1) by substituting the model equations (6.3.7b) and (6.3.7c) into the objective function (6.3.7a). To achieve distributed control, we use Algorithm 1 to solve (6.3.7).

Let the input sequence returned by Algorithm 1 be $\mathbf{u}^{\bar{p}}(x, \tilde{\mathbf{u}})$. The first input of this sequence $\kappa^{\bar{p}}(x(0)) = u^{\bar{p}}(0; x(0), \tilde{\mathbf{u}})$ is injected into the plant and the state is moved forward. To reinitialize the algorithm at the next sampling time, we define the warm start

$$\tilde{\mathbf{u}}_1^+ = \{u_1(1), u_1(2), \ldots, u_1(N-1), \kappa_{1f}\big(x(N)\big)\}$$

$$\tilde{\mathbf{u}}_2^+ = \{u_2(1), u_2(2), \ldots, u_2(N-1), \kappa_{2f}\big(x(N)\big)\}$$

in which $x(N) = \phi(N; x(0), \mathbf{u}_1, \mathbf{u}_2)$. In general, it is not possible to solve optimization (6.3.7) to optimality because of the limited time available to obtain feedback. The distributed controller is therefore suboptimal, and the stability of the controller can be established by suboptimal MPC theory.

6.3.8 Stability of Distributed Nonlinear Cooperative Control

To establish stability of the control algorithm, we show that the plantwide objective cost decreases between sampling times. Without loss of generality, assume $k = 0$ and the input $u(0)$ is injected into the plant. Using the warm start as the initial condition at the next sampling time, we have

$$\begin{aligned}
V(x^+, \tilde{\mathbf{u}}^+) = V(x, \mathbf{u}) &- \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2) \\
&- \rho_1 V_{1f}(x(N)) - \rho_2 V_{2f}(x(N)) \\
&+ \rho_1 \ell_1\Big(x_1(N), \kappa_{1f}\big(x(N)\big)\Big) + \rho_2 \ell_2\Big(x_2(N), \kappa_{2f}\big(x(N)\big)\Big) \\
&+ \rho_1 V_{1f}\Big(f_1\Big(x_1(N), x_2(N), \kappa_{1f}\big(x(N)\big), \kappa_{2f}\big(x(N)\big)\Big)\Big) \\
&+ \rho_2 V_{2f}\Big(f_2\Big(x_1(N), x_2(N), \kappa_{1f}\big(x(N)\big), \kappa_{2f}\big(x(N)\big)\Big)\Big)
\end{aligned}$$

Using (6.3.3), the last six terms above are cumulatively nonpositive, giving

$$V(x^+, \tilde{\mathbf{u}}^+) \le V(x, \mathbf{u}) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)$$

By Lemma 6.4, the objective function cost decreases from this warm start, so that

$$V(x^+, \mathbf{u}^+) \le V(x, \mathbf{u}) - \rho_1 \ell_1(x_1, u_1) - \rho_2 \ell_2(x_2, u_2)$$

Hence

$$V(x^+, \mathbf{u}^+) - V(x, \mathbf{u}) \le -\alpha(|(x, u)|) \tag{6.3.8}$$

in which $\alpha(|(x,u)|) = \rho_1 \alpha_1(|(x_1, u_1)|) + \rho_2 \alpha_2(|(x_2, u_2)|)$.

We now give the main result of this chapter. Let $\mathbb{X}_N$ be the forward invariant set of all initial states for which the control optimization (6.3.7) is feasible.

Theorem 6.15 (Asymptotic stability)**.** *Let Assumptions 6.10-6.12 hold and let $V(\cdot) \leftarrow V^{\overline{\beta}}(\cdot)$ by Proposition 6.14. Then for every $x(0) \in \mathbb{X}_N$, the origin is asymptotically stable for the closed-loop system $x^+ = f(x, \kappa^{\bar{p}}(x))$.*

Proof. The proof follows from the stability of suboptimal MPC (Scokaert et al., 1999, Theorem 1), which requires satisfaction of three properties to prove asymptotic stability. 1) There exists a lower bound $\alpha(|x|) \leq V(x,\mathbf{u})$ by satisfaction of Assumptions 6.10 and 6.11. 2) The descent property has been shown above in (6.3.8). 3) The Lyapunov constraint (6.3.7e) is explicitly added to the optimization. We have accounted for each required property and the result is established. □

Remark 6.16 (M subsystems)**.** *The arguments for the controller have been given for the case of two subsystems only, but same arguments apply for any finite $M > 0$ number of subsystems.*

6.4 Illustrative Example

For this example, we use the stage cost

$$\ell_1(x_1,u_1) = \frac{1}{2}(x_1' Q_1 x_1 + u_1' R_1 u_1)$$
$$\ell_2(x_2,u_2) = \frac{1}{2}(x_2' Q_2 x_2 + u_2' R_2 u_2)$$

in which $Q_1, Q_2 > 0$ and $R_1, R_2 > 0$. This stage cost gives the objective function

$$V(x,\mathbf{u}) = \frac{1}{2}\sum_{k=0}^{N-1} x(k)'Qx(k) + u(k)'Ru(k) + V_f(x(N))$$

in which $Q = \mathrm{diag}(Q_1, Q_2)$, $R = \mathrm{diag}(R_1, R_2)$ and $V_f(\cdot) = V_{1f}(\cdot) + V_{2f}(\cdot)$ is defined by (6.3.4). The terminal region is defined as in Section 6.3.5.1.

6.4.1 Simulation

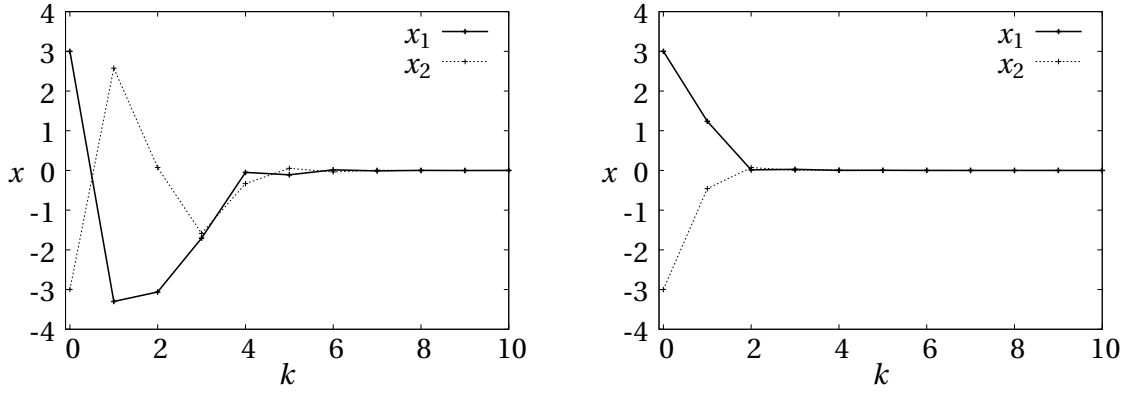
Consider the unstable nonlinear system

$$x_1^+ = x_1^2 + x_2 + u_1^3 + u_2$$
$$x_2^+ = x_1 + x_2^2 + u_1 + u_2^3$$

with initial condition $(x_1, x_2) = (3, -3)$. The control objective is to stabilize the system and drive the states to the origin. For the simulation we choose the parameters

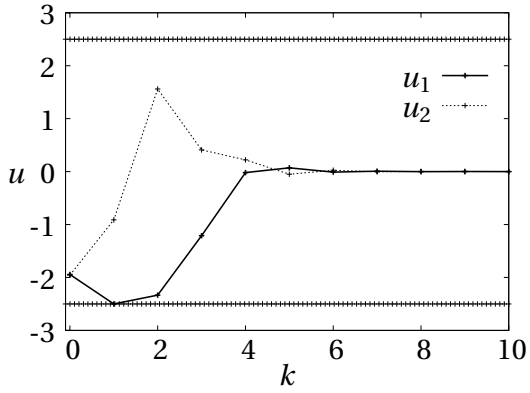
$$Q = I \quad R = I \quad N = 2 \quad \overline{p} = 3 \quad \beta = 1 \quad \mathbb{U}_i = [-2.5, 2.5] \quad \forall i \in \mathbb{I}_{1:2}$$

As shown in Figure 6.2, the control scheme is stabilizing. Increasing the maximum number of iterations significantly improves the performance. In Figure 6.2, we also show the performance for $\overline{p} = 10$. The cost difference is given in Figure 6.3. To elucidate the difficulty in optimizing the nonconvex objective function, the iterations of the zeroth stage control optimization are shown in Figure 6.4 for the $N = 1$ case. The terminal region, calculated as in Section 6.3.5.1, is shown in Figure 6.5.

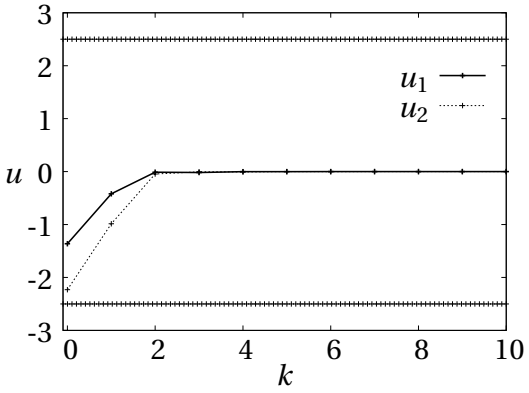


(a) State trajectory ($\overline{p} = 3$)

(b) Centralized state trajectory ($\overline{p} = 10$)



(c) Input trajectory ($\overline{p} = 3$)



(d) Centralized input trajectory ($\overline{p} = 10$)

Figure 6.2: Controller performance with $(x_1(0), x_2(0)) = (3, -3)$. Setting $\overline{p} = 10$ approximates a centralized controller solution.

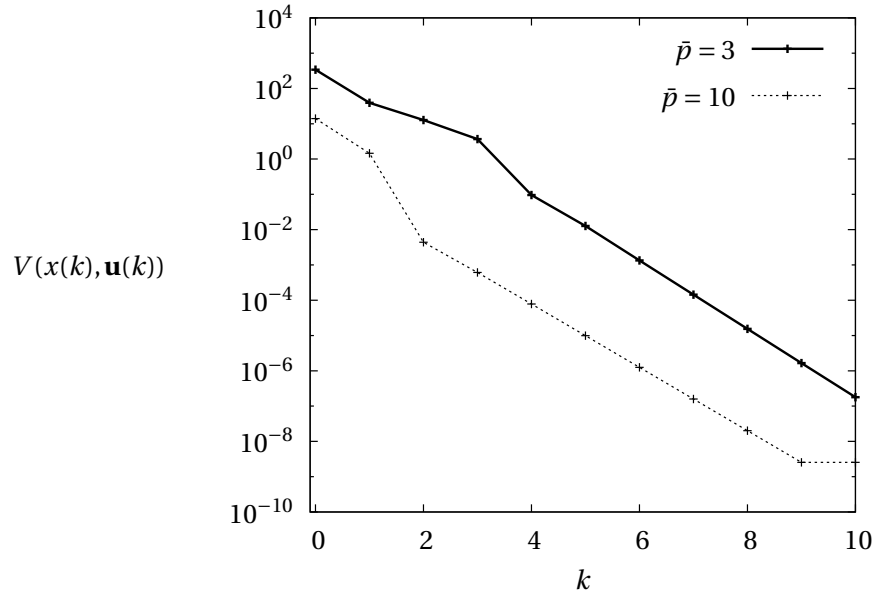


Figure 6.3: Open-loop cost to go versus time on the closed-loop trajectory for different numbers of iterations.

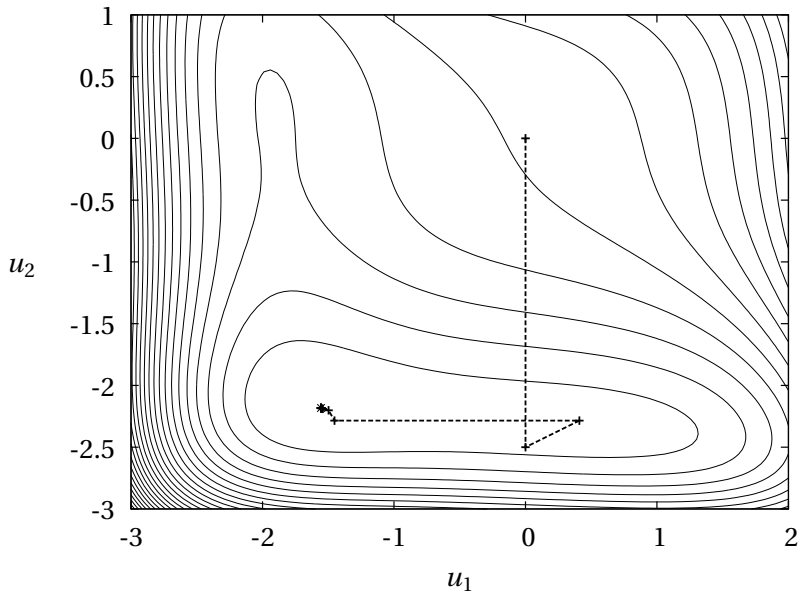


Figure 6.4: Contours of V with $N = 1$ for $k = 0$ with $(x_1(0), x_2(0)) = (3, -3)$. Iterations of the subsystem controllers with initial condition $(u_1^0, u_2^0) = (0, 0)$.

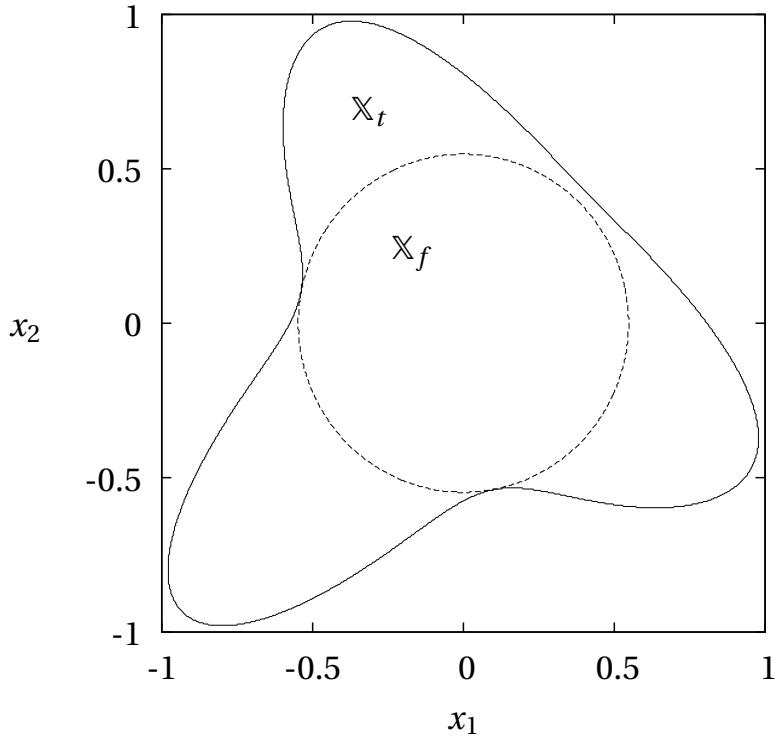


Figure 6.5: Terminal region. $\mathbb{X}_t$ are the points in which the terminal controller is stabilizing and $\mathbb{X}_f = \{x \mid V_f(x) \leq 0.485\} \subseteq \mathbb{X}_t$ is the terminal region.

Chapter 7

Conclusions and Future Work

We conclude with a summary of contributions and some suggestions for challenging problems motivated by this thesis.

Contributions

Plantwide offset-free feedback for linear plants: The theory of cooperative MPC was given in Chapter 3. This control strategy has the following features: hard input constraints are satisfied; terminating the distributed optimization iteration prior to convergence does not affect nominal stability; the distributed optimization converges to the Pareto optimal (centralized) solution; no coordinating optimization is needed. Exponential stability of the closed-loop plantwide system was proven for the state and output feedback cases. An extension is also shown in which sparsely coupled constraints between subsystems can be handled without loss of stability or optimality. Stability does not depend on subsystem interaction strength.

Integration of multiple communication and optimization time scales: In Chapter 4, an extension to cooperative control was provided in which communication between subsystems occurs at multiple time schedules, as in traditional hierarchical control. The hierarchical control scheme was shown to provide exponentially stable performance under state and output feedback. This extension uses a modification of the distributed optimization and was shown to converge to the Pareto optimum and can be terminated early without affecting nominal stability.

Plantwide implementation examples: To demonstrate the flexibility of cooperative control, a series of examples were given in Chapter 5. These examples have shown that cooperative control can nominally stabilize subsystems optimizing, communicating, and sampling on any time scale. Further examples were provided to illustrate that the modeling required for cooperative control can be identified over time and that the models can be generated from a centralized, first-principles model.

Plantwide nominally stable control for nonlinear plants: Chapter 6 presented a novel distributed optimization for nonconvex problems that provides convergence to stationary points. We combined this optimization with a new result in suboptimal MPC to develop a plantwide distributed nonlinear controller. This controller was shown to provide asymptotically stabilizing feedback and an example was used to illustrate its performance.

Future Work

Integration of cooperative and economic MPC: Recent work in economic MPC has led to a method in which MPCs can track plant profit directly (Diehl, Amrit, and Rawlings, 2010; Rawlings and Amrit, 2009). Real time optimization (RTO) is often used in plantwide control to find the economically optimal steady state. Using the theory of economic MPC, the nonconvex optimization defined in Section 6.2 can instead be used to optimize economics directly in the distributed controllers.

Systematic plant decomposition: In Chapter 5, two methods for constructing the plantwide interaction model are given. These models must satisfy the stabilizability and detectability assumptions (Assumptions 3.5 and 3.14). There is no systematic method to generate the best plant decomposition for each plant. Similarly, there is not a method to form the best neighborhood decomposition in hierarchical control. These methods would help guide practitioners to the best use of cooperative control.

Densely coupled input constraints: In Section 3.5, we provided a method for handling plants with sparsely coupled input constraints between the subsystems. Densely coupled constraints can occur in plants with strong dynamic coupling and hard output constraints. In the limit of plants with coupled constraints between all subsystems, the optimization method presented in this thesis requires each subsystem to solve the centralized control problem. This computational burden weakens the advantages of cooperative control. Therefore, there exists a need for an alternative optimization method that can improve the objective and stay feasible at each iterate while giving provable convergence to the global optimum.

Unstable models: To satisfy the stability assumption for linear systems (Assumption 3.5), unstable modes in the plant cannot be controllable by multiple subsystems. This stability assumption is required to satisfy the terminal constraint in the MPC problem. A relaxed stability assumption for linear systems can be formed by using terminal control MPC and removing the terminal constraint as in Section 6.3.6. This relaxed stability requirement would allow a greater diversity of plant decompositions to be controlled by cooperative MPC.

Nominal error bound in hierarchical control: The nominal stability of hierarchical control depends upon the propagation of nominal error in the plant. To show stability, Assumptions 4.20 and 4.21 are required. The assumptions bound the nominal error. More work is needed to determine which systems satisfy this bound.

Data network faults: In this thesis, we have assumed that the data communicated between subsystem controllers is transferred without faults. Real plants have networks with interference and errors that may lead to poor performance in plantwide control. The sensitivity of cooperative control to these communication errors should be investigated.

Bibliography

Walid Al-Ghewi, Hector Budman, and Ali Elkamel. Selection of control structure for distributed model predictive control in the presence of model errors. *J. Proc. Cont.*, 20(3):270–284, 2010.

Panos J. Antsaklis and Anthony N. Michel. *Linear Systems*. McGraw-Hill, New York, 1997.

E. M. B. Aske, S. Strand, and S. Skogestad. Coordinator MPC for maximizing plant throughput. *Comput. Chem. Eng.*, 32:195–204, 2008.

R. A. Bartlett, L. T. Biegler, J. Backstrom, and V. Gopal. Quadratic programming algorithms for large-scale model predictive control. *J. Proc. Cont.*, 12(7):775–795, 2002.

Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*. SIAM, Philadelphia, 1999.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, second edition, 2008.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation*. Athena Scientific, Belmont, Massachusetts, 1997.

Eduardo F. Camacho and C. Bordons. *Model Predictive Control*. Springer-Verlag, London, second edition, 2004.

Eduardo Camponogara and Nuno M. C. de Oliveira. Distributed optimization for model predictive control of linear-dynamic networks. *IEEE Trans. Syst. Man Cyber.*, 39(6):1331–1338, November 2009.

Eduardo Camponogara and Sarosh Talukdar. Distributed model predictive control. *IEEE Trans. Syst. Man Cyber.*, 37(5):732–745, September 2007.

Eduardo Camponogara, Dong Jia, Bruce H. Krogh, and Sarosh Talukdar. Distributed model predictive control. *IEEE Ctl. Sys. Mag.*, pages 44–52, February 2002.

R. Cheng, J. F. Forbes, and W. S. Yip. Price-driven coordination method for solving plant-wide MPC problems. *J. Proc. Cont.*, 17(5):429–438, 2007.

David Chilin, Jinfeng Liu, David Muñoz de la Peña, Panagiotis D. Christofides, and James F. Davis. Detection, isolation and handling of actuator faults in distributed model predictive control systems. *J. Proc. Cont.*, 20(9):1059–1075, 2010.

Moritz Diehl, Rishi Amrit, and James B. Rawlings. A Lyapunov function for economic optimizing model predictive control. *IEEE Trans. Auto. Cont.*, 2010. Accepted for publication.

William B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. Auto. Cont.*, 52(7):1249–1263, 2007.

William B. Dunbar and Richard M. Murray. Distributed receding horizon control with application to multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

Elisa Franco, Lalo Magni, Thomas Parisini, Marios M. Polycarpou, and Davide M. Raimondo. Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach. *IEEE Trans. Auto. Cont.*, 53 (1):324–338, 2008.

Ravindra D. Gudi and James B. Rawlings. Identification for Decentralized Model Predictive Control. *AIChE J.*, 52(6):2198–2210, 2006.

R. M. Hermans, M. Lazar, and A. Jokić. Almost decentralized Lyapunov-based nonlinear model predictive control. In *Proceedings of the American Control Conference*, Baltimore, Maryland, June 2010a.

R. M. Hermans, M. Lazar, A. Jokić, and P. P. J. van den Bosch. Almost decentralized Lyapunov-based model predictive control of power networks. In *Proceedings 15th IEEE Mediterranean Electrotechnical Conference*, Valletta, Malta, April 2010b.

Dong Jia and Bruce H. Krogh. Min-max feedback model predictive control for distributed control with communication. In *Proceedings of the American Control Conference*, pages 4507–4512, Anchorage,Alaska, May 2002.

T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G.J. Balas. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Trans. Cont. Sys. Tech.*, 16 (1):19–33, Jan. 2008.

Tamás Keviczky, Francesco Borrelli, and Gary J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(6):2105–2115, 2006.

Tamás Keviczky, Francesco Borrelli, and Gary J. Balas. Distributed predictive control: synthesis, stability and feasibility. In Jeff S. Shamma, editor, *Cooperative Control of Distributed Multi-Agent Systems*, pages 79–108. John Wiley & Sons, 2007.

V. Lakshmikantham, S. Leela, and A. A. Martynyuk. *The Mathematics of Nonlinear Programming*. World Scientific Publishing Co. Pte. Ltd., Singapore, 1990.

Truls Larsson and Sigurd Skogestad. Plantwide control- A review and a new design procedure. *Mod. Ident. Control*, 21(4):209–240, 2000.

Jinfeng Liu, David Muñoz de la Peña, Benjamin J. Ohran, Panagiotis D. Christofides, and James F. Davis. A two-tier architecture for networked process control. *Chem. Eng. Sci.*, 63(22):5394–5409, 2008.

Jinfeng Liu, David Muñoz de la Peña, and Panagiotis D. Christofides. Distributed model predictive control of nonlinear process systems. *AIChE J.*, 55(5):1171–1184, 2009.

Jinfeng Liu, X. Chen, David Muñoz de la Peña, and Panagiotis D. Christofides. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE J.*, 56(5):2137–2149, 2010a.

Jinfeng Liu, David Muñoz de la Peña, and Panagiotis D. Christofides. Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements. *Automatica*, 46(1):52–61, 2010b.

J. M. Maciejowski. *Predictive Control with Contraints*. Prentice-Hall, Harlow, UK, 2002.

J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Cont. Appl. Meth.*, In press, 2010.

Natalia I. Marcos, J. Fraser Forbes, and Martin Guay. Coordination of distributed model predictive controllers for constrained dynamic processes. In *ADCHEM 2009, International Symposium on Advanced Control of Chemical Processes*, Istanbul, Turkey, July 12-15, 2009.

David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

M. Mercangöz and F. J. Doyle. Distributed model predictive control of an experimental four-tank system. *J. Proc. Cont.*, 17(3):297–308, 2007.

MD Mesarović, D. Macko, and Y. Takahara. *Theory of hierarchical, multilevel systems*. Academic Press, New York, 1970.

Deepak Nagrath, Vinay Prasad, and B. Wayne Bequette. A model predictive formulation for control of open-loop unstable cascade systems. *Chem. Eng. Sci.*, 57:365–378, 2002.

Ion Necoara, Dang Doan, and J. A. K Suykens. Application of the proximal center decomposition method to distributed model predictive control. In *Proceedings of the IEEE Conference on Decision and Control*, Cancun, Mexico, December 9-11 2008.

R.R. Negenborn, B. De Schutter, and J. Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Eng. App. A.I.*, 21(3):353–366, 2008.

R.R. Negenborn, S. Leirens, B. De Schutter, and J. Hellendoorn. Supervisory nonlinear mpc for emergency voltage control using pattern search. *Control Eng. Prac.*, 17(7):841–848, 2009.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, second edition, 2006.

Babatunde A. Ogunnaike and W. Harmon Ray. *Process Dynamics, Modeling, and Control*. Oxford University Press, New York, 1994.

Gabriele Pannocchia, James B. Rawlings, and Stephen J. Wright. Fast, large-scale model predictive control by partial enumeration. *Automatica*, 43:852–860, 2007.

Gabriele Pannocchia, Stephen J. Wright, Brett T. Stewart, and James B. Rawlings. Efficient cooperative distributed MPC using partial enumeration. In *ADCHEM 2009, International Symposium on Advanced Control of Chemical Processes*, Istanbul, Turkey, July 12-15, 2009.

S. Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Eng. Prac.*, 11(7):733–764, 2003.

James B. Rawlings and Rishi Amrit. Optimizing process economic performance using model predictive control. In Lalo Magni, Davide Martino Raimondo, and Frank Allgöwer, editors, *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 119–138, Berlin, 2009. Springer.

James B. Rawlings and David Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI, 2009. 576 pages, ISBN 978-0-9759377-0-9.

James B. Rawlings and Brett T. Stewart. Coordinating multiple optimization-based controllers: new opportunities and challenges. In *DYCOPS*, Cancun, Mexico, June 2007.

James B. Rawlings, Brett T. Stewart, Stephen J. Wright, and David Q. Mayne. Suboptimal mpc: On replacing the terminal contraint by a terminal cost, 2010. Internal communication.

J. Anthony Rossiter. *Model-based predictive control: a practical approach*. CRC Press LLC, Boca Raton, FL, 2004.

Louis P. Russo and B. Wayne Bequette. State-space versus input/output representations for cascade control of unstable systems. *Ind. Eng. Chem. Res.*, 36(6):2271–2278, 1997. doi: 10.1021/ie960677o. URL `http://pubs.acs.org/doi/abs/10.1021/ie960677o`.

Nils R. Sandell Jr., Pravin Varaiya, Michael Athans, and Michael Safonov. Survey of decentralized control methods for large scale systems. *IEEE Trans. Auto. Cont.*, 23(2):108–128, 1978.

Riccardo Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *J. Proc. Cont.*, 19(5):723–731, May 2009. ISSN 0959-1524.

Pierre O. M. Scokaert, David Q. Mayne, and James B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. Auto. Cont.*, 44(3):648–654, March 1999.

Brett T. Stewart, James B. Rawlings, and Stephen J. Wright. Hierarchical cooperative distributed model predictive control. In *Proceedings of the American Control Conference*, Baltimore, Maryland, June 2010a.

Brett T. Stewart, Aswin N. Venkat, James B. Rawlings, Stephen J. Wright, and Gabriele Pannocchia. Cooperative distributed model predictive control. *Sys. Cont. Let.*, 59:460–469, 2010b.

Brett T. Stewart, Stephen J. Wright, and James B. Rawlings. Cooperative distributed model predictive control for nonlinear systems. *J. Proc. Cont.*, 2010c. Submitted for publication.

Aswin N. Venkat. *Distributed Model Predictive Control: Theory and Applications*. PhD thesis, University of Wisconsin–Madison, October 2006. URL `http://jbrwww.che.wisc.edu/theses/venkat.pdf`.

Aswin N. Venkat, James B. Rawlings, and Stephen J. Wright. Stability and optimality of distributed model predictive control. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, December 2005.

Aswin N. Venkat, James B. Rawlings, and Stephen J. Wright. Implementable distributed model predictive control with guaranteed performance properties. In *Proceedings of the American Control Conference*, Minneapolis, Minnesota, June 14-16 2006.

Aswin N. Venkat, James B. Rawlings, and Stephen J. Wright. Distributed model predictive control of large-scale systems. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, pages 591–605. Springer, 2007.

Aswin N. Venkat, Ian A. Hiskens, James B. Rawlings, and Stephen J. Wright. Distributed MPC strategies with application to power system automatic generation control. *IEEE Ctl. Sys. Tech.*, 16(6):1192–1206, November 2008.

L. Wang. *Model Predictive Control System Design and Implementation Using Matlab*. Springer, New York, 2009.

R. K. Wood and M. W. Berry. Terminal composition control of a binary distillation column. *Chem. Eng. Sci.*, 28:1707–1717, 1973.

Robert E. Young, R. Donald Bartusiak, and Robert W. Fontaine. Evolution of an industrial nonlinear model predictive controller. In James B. Rawlings, Babatunde A. Ogunnaike, and John W. Eaton, editors, *Chemical Process Control–VI: Sixth International Conference on Chemical Process Control*, pages 342–351, Tucson, Arizona, January 2001. AIChE Symposium Series, Volume 98, Number 326.

DISCARD THIS PAGE

Appendix A: From input-to-state to state-to-state models

In this appendix, we provide a general method for switching models from an input-to-state representation in (4.5.1) to the state-to-state representation in (4.5.2).

Example

Consider the network shown in Fig. A.1. In this network, each subsystem is its own neighborhood. From the definition of neighborhoods (see Definition 4.15), there is a time delay between subsystems for each edge in the network. This network is represented by the model

$$
\begin{aligned}
x_{11}^{+} &= A_{11}x_{11} + B_{11}u_1 & x_{21}^{+} &= A_{21}x_{21} + B_{21}u_1^{-} & x_{31}^{+} &= A_{31}x_{31} + B_{31}u_1^{2-} \\
x_{12}^{+} &= A_{12}x_{12} + B_{12}u_2^{2-} & x_{22}^{+} &= A_{22}x_{22} + B_{22}u_2 & x_{32}^{+} &= A_{32}x_{32} + B_{32}u_2^{-} \\
x_{13}^{+} &= A_{13}x_{13} + B_{13}u_3^{-} & x_{23}^{+} &= A_{23}x_{23} + B_{23}u_3^{2-} & x_{33}^{+} &= A_{33}x_{33} + B_{33}u_3
\end{aligned}
$$

in which the superscript $t-$ indicates a time delay of t steps. By augmenting the states of each subsystem, we change the input-to-state model to a model with state-to-state interactions.

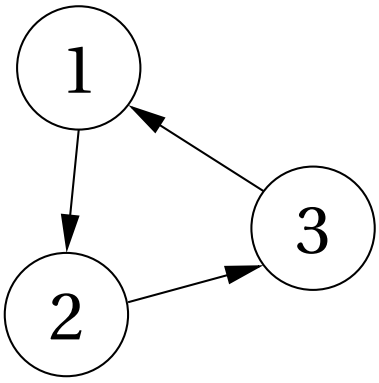


Figure A.1: Each node is its own neighborhood

Each state x_{ij} is augmented with the input u_j one time step behind for all $(i,j) \in \mathbb{I}_{1:3} \times \mathbb{I}_{1:3}$

$$\begin{bmatrix} x_{11} \\ u_1^- \end{bmatrix}^+ = \begin{bmatrix} A_{11} & \\ & 0 \end{bmatrix} \begin{bmatrix} x_{11} \\ u_1^- \end{bmatrix} + \begin{bmatrix} B_{11} \\ I \end{bmatrix} u_1$$

$$x_{12}^+ = A_{12} x_{12} + \begin{bmatrix} 0 & B_{12} \end{bmatrix} \begin{bmatrix} x_{32} \\ u_2^{2-} \end{bmatrix}$$

$$\begin{bmatrix} x_{13} \\ u_3^{2-} \end{bmatrix}^+ = \begin{bmatrix} A_{13} & \\ & 0 \end{bmatrix} \begin{bmatrix} x_{13} \\ u_3^{2-} \end{bmatrix} + \begin{bmatrix} 0 & B_{13} \\ 0 & I \end{bmatrix} \begin{bmatrix} x_{33} \\ u_3^- \end{bmatrix}$$

$$\begin{bmatrix} x_{21} \\ u_1^{2-} \end{bmatrix}^+ = \begin{bmatrix} A_{21} & \\ & 0 \end{bmatrix} \begin{bmatrix} x_{21} \\ u_1^{2-} \end{bmatrix} + \begin{bmatrix} 0 & B_{21} \\ 0 & I \end{bmatrix} \begin{bmatrix} x_{11} \\ u_1^- \end{bmatrix}$$

$$\begin{bmatrix} x_{22} \\ u_2^- \end{bmatrix}^+ = \begin{bmatrix} A_{22} & \\ & 0 \end{bmatrix} \begin{bmatrix} x_{22} \\ u_2^- \end{bmatrix} + \begin{bmatrix} B_{22} \\ I \end{bmatrix} u_2$$

$$x_{23}^+ = A_{23} x_{23} + \begin{bmatrix} 0 & B_{23} \end{bmatrix} \begin{bmatrix} x_{13} \\ u_3^{2-} \end{bmatrix}$$

$$x_{31}^+ = A_{31} x_{31} + \begin{bmatrix} 0 & B_{31} \end{bmatrix} \begin{bmatrix} x_{21} \\ u_1^{2-} \end{bmatrix}$$

$$\begin{bmatrix} x_{32} \\ u_2^{2-} \end{bmatrix}^+ = \begin{bmatrix} A_{32} & \\ & 0 \end{bmatrix} \begin{bmatrix} x_{32} \\ u_2^{2-} \end{bmatrix} + \begin{bmatrix} 0 & B_{32} \\ 0 & I \end{bmatrix} \begin{bmatrix} x_{22} \\ u_2^- \end{bmatrix}$$

$$\begin{bmatrix} x_{33} \\ u_3^- \end{bmatrix}^+ = \begin{bmatrix} A_{33} & \\ & 0 \end{bmatrix} \begin{bmatrix} x_{33} \\ u_3^- \end{bmatrix} + \begin{bmatrix} B_{33} \\ I \end{bmatrix} u_3$$

Notice that u_j^{3-} is not used because the longest path in the graph is three steps. Then we have the model

$$\begin{aligned}
x_{11}^+ &= A_{11} x_{11} + B_{11} u_1 & x_{21}^+ &= A_{21} x_{21} + B_{21} x_{11} & x_{31}^+ &= A_{31} x_{31} + B_{31} x_{21} \\
x_{12}^+ &= A_{12} x_{12} + B_{12} x_{32} & x_{22}^+ &= A_{22} x_{22} + B_{22} u_2 & x_{32}^+ &= A_{32} x_{32} + B_{32} x_{22} \\
x_{13}^+ &= A_{13} x_{13} + B_{13} x_{33} & x_{23}^+ &= A_{23} x_{23} + B_{23} x_{13} & x_{33}^+ &= A_{33} x_{33} + B_{33} u_3
\end{aligned}$$

for which we have replaced the following parameters

$$x_{11} \leftarrow \begin{bmatrix} x_{11} \\ u_1^- \end{bmatrix} \qquad x_{21} \leftarrow \begin{bmatrix} x_{21} \\ u_1^{2-} \end{bmatrix} \qquad x_{31} \leftarrow x_{31}$$

$$x_{12} \leftarrow x_{12} \qquad x_{22} \leftarrow \begin{bmatrix} x_{22} \\ u_2^- \end{bmatrix} \qquad x_{32} \leftarrow \begin{bmatrix} x_{32} \\ u_2^{2-} \end{bmatrix}$$

$$x_{13} \leftarrow \begin{bmatrix} x_{13} \\ u_3^{2-} \end{bmatrix} \qquad x_{23} \leftarrow x_{23} \qquad x_{33} \leftarrow \begin{bmatrix} x_{33} \\ u_3^- \end{bmatrix}$$

$$A_{11} \leftarrow \text{diag}(A_{11}, 0) \qquad A_{21} \leftarrow \text{diag}(A_{21}, 0) \qquad A_{31} \leftarrow A_{31}$$

$$A_{12} \leftarrow A_{12} \qquad A_{22} \leftarrow \text{diag}(A_{22}, 0) \qquad A_{32} \leftarrow \text{diag}(A_{32}, 0)$$

$$A_{13} \leftarrow \text{diag}(A_{13}, 0) \qquad A_{23} \leftarrow A_{23} \qquad A_{33} \leftarrow \text{diag}(A_{33}, 0)$$

$$B_{11} \leftarrow \begin{bmatrix} B_{11} \\ I \end{bmatrix} \qquad B_{21} \leftarrow \begin{bmatrix} 0 & B_{21} \\ 0 & I \end{bmatrix} \qquad B_{31} \leftarrow \begin{bmatrix} 0 & B_{31} \end{bmatrix}$$

$$B_{12} \leftarrow \begin{bmatrix} 0 & B_{12} \end{bmatrix} \qquad B_{22} \leftarrow \begin{bmatrix} B_{22} \\ I \end{bmatrix} \qquad B_{32} \leftarrow \begin{bmatrix} 0 & B_{32} \\ 0 & I \end{bmatrix}$$

$$B_{13} \leftarrow \begin{bmatrix} 0 & B_{13} \\ 0 & I \end{bmatrix} \qquad B_{23} \leftarrow \begin{bmatrix} 0 & B_{23} \end{bmatrix} \qquad B_{33} \leftarrow \begin{bmatrix} B_{33} \\ I \end{bmatrix}$$

Following this procedure, we can change any input-to-state model into a model with state-to-state interactions.

PLANTWIDE COOPERATIVE DISTRIBUTED MODEL PREDICTIVE CONTROL

Brett Taylor Stewart

Under the supervision of Professor James B. Rawlings

At the University of Wisconsin-Madison

This thesis presents a plantwide model predictive control strategy in which subsystems provide feedback by solving a local optimization and exchanging information across the plant. The plant model is required to satisfy a stabilizability condition that does not depend upon the strength of the open-loop, inter-subsystem interactions. This control strategy has the following features: hard input constraints are satisfied; terminating the distributed optimization iteration prior to convergence does not affect nominal stability; the distributed optimization converges to the Pareto optimal (centralized) solution; no coordinating optimization is employed. Exponential stability of the closed-loop plantwide system is proven for the state and output feedback cases. A modification is presented in which constraints sparsely coupled between subsystems can be handled without loss of stability or optimality.

An extension to the cooperative controller is provided in which communication between subsystems occurs at multiple time schedules, as in traditional hierarchical control. The hierarchical control scheme is shown to provide exponentially stable performance under state and output feedback. This extension uses a modification of the distributed optimization. The optimization is shown to converge to the Pareto optimum and can be terminated early without affecting nominal stability.

To demonstrate the flexibility of cooperative control, a series of examples are presented. These examples show that cooperative control can nominally stabilize subsystems optimizing, communicating, and sampling on any time scale. Further examples are provided to illustrate that

the modeling required for cooperative control can be identified over time and that the models can be generated from a centralized first-principles model.

Finally, a novel distributed optimization for nonconvex problems is presented that provides convergence to stationary points. This optimization is combined with a new result in suboptimal MPC to develop a plantwide distributed nonlinear controller. This controller provides asymptotically stablizing feedback and an example is used to illustrate performance.

James B. Rawlings