Distributed Model Predictive Control: Theory and Applications

by

Aswin N. Venkat

A dissertation submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

(Chemical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2006

For my family . . .

Distributed Model Predictive Control: Theory and Applications

Aswin N. Venkat

Under the supervision of Professor James B. Rawlings

At the University of Wisconsin–Madison

Most standard model predictive control (MPC) implementations partition the plant into several units and apply MPC individually to these units. It is known that such a completely decentralized control strategy may result in unacceptable control performance, especially if the units interact strongly. Completely centralized control of large, networked systems is viewed by most practitioners as impractical and unrealistic. In this dissertation, a new framework for distributed, linear MPC with guaranteed closed-loop stability and performance properties is presented. A modeling framework that quantifies the interactions among subsystems is employed. One may think that modeling the interactions between subsystems and exchanging trajectory information among MPCs (communication) is sufficient to improve controller performance. We show that this idea is incorrect and may not provide even closed-loop stability. A cooperative distributed MPC framework, in which the objective functions of the local MPCs are modified to achieve systemwide control objectives is proposed. This approach allows practitioners to tackle large, interacting systems by building on local MPC systems already in place. The iterations generated by the proposed distributed MPC algorithm are systemwide feasible, and the controller based on any intermediate termination of the algorithm is closed-loop sta-

ble. These two features allow the practitioner to terminate the distributed MPC algorithm at the end of the sampling interval, even if convergence is not achieved. If iterated to convergence, the distributed MPC algorithm achieves optimal, centralized MPC control.

Building on results obtained under state feedback, we tackle next, distributed MPC under output feedback. Two distributed estimator design strategies are proposed. Each estimator is stable and uses only local measurements to estimate subsystem states. Feasibility and closed-loop stability for all distributed MPC algorithm iteration numbers are established for the distributed estimator-distributed regulator assembly in the case of decaying estimate error. A subsystem-based disturbance modeling framework to eliminate steady-state offset due to modeling errors and unmeasured disturbances is presented. Conditions to verify suitability of chosen local disturbance models are provided. A distributed target calculation algorithm to compute steady-state targets locally is proposed. All iterates generated by the distributed target calculation algorithm are feasible steady states. Conditions under which the proposed distributed MPC framework, with distributed estimation, distributed target calculation and distributed regulation, achieves offset-free control at steady state are described. Finally, the distributed MPC algorithm is augmented to allow asynchronous optimization and asynchronous feedback. Asynchronous feedback distributed MPC enables the practitioner to achieve performance superior to centralized MPC operated at the slowest sampled rate. Examples from chemical engineering, electrical engineering and civil engineering are examined and benefits of employing the proposed distributed MPC paradigm are demonstrated.

Acknowledgments

At the University of Wisconsin, I have had the opportunity to meet some wonderful people. First, I'd like to thank my advisor Jim Rawlings. I cannot put into words what I have learnt from him. His intellect, attitude to research and career have been a great source of inspiration. I have always been amazed by his ability to distill the most important issues from complex problems. It has been a great honor to work with him and learn from him.

I've been fortunate to have had the chance to collaborate with two fine researchers-Steve Wright and Ian Hiskens. I thank Steve for being incredibly patient with me from the outset. Steve's understanding of optimization is unparalleled, and his quickness in comprehending and critically analyzing material has constantly amazed me. I thank Ian for listening to my crazy ideas, for teaching me the basics of power systems, and for constantly encouraging me to push the bar higher. I have enjoyed our collaboration immensely. I'd like to thank Professors Mike Graham, Regina Murphy and Christos Maravelias for taking the time to serve on my thesis committee.

I'm grateful to my undergraduate professors Dr. R. D. Gudi and Dr. K. P. Madhavan for teaching me the basics of process control. Their lectures attracted me to this field initially. Dr. Gudi also made arrangements so that I could work in his group, and encouraged me to pursue graduate school. Dr. Vijaysai, Thank you for being such a great coworker and friend.

Over the years, the Rawlings group has been quite an assorted bunch. I thank Eric Haseltine for his friendship and for showing me the ropes when I first joined the group. I am indebted to John Eaton for answering all my Octave and Linux questions, and for providing invaluable computer support. I miss the lengthy discussions on cricket with Dan Patience. Brian Odelson generously devoted time to answer all my computer questions. Thank you Matt Tenny for answering my naive control questions. Jenny was always cheerful and willing to lend a helping hand. It was nice to meet Dennis Bonne. Gabriele, I've enjoyed the discussions we've had. It has been nice to get to know Paul Larsen, Ethan Mastny and Murali Rajamani. Murali, I hope that your "KK curse" is lifted one day. I wish Brett Stewart the best of luck in his studies. I've enjoyed our discussions, though I regret we did not have more time.

Thank you Nishant "Nanga" Bhasin for being a close friend all through undergrad and grad school. I miss our late night expeditions on Market street, the many trips to Pats and yearly camping trips. I could always count on Ashish Batra for sound advice on a range of topics. In the past five years, I have also made some lifelong friends in Madison, WI. Cliff, I will never forget those late nights in town, the lunch trips to Jordans and those Squash games. I'll also miss your "home made beer and cider", and the many excuses we conjured up to go try them. Angela was always a willing partner to Nams and to hockey games. I will keep my promise and take you to a cricket game sometime. Gova, I could always count on you for a game of Squash and/or beer. Thank you Paul, Erin, Amy, Maritza, Steve, Rajesh "Pager" and Mike for your friendship. I'd like to also thank the Madison cricket team for some unforgettable experiences over the last four summers.

I owe a lot to my family. Thank you Mum, Dad, Kanchan for your love, and for always being there. I thank my family in the states: my grandparents, Pushpa, Bobby and the "kids"-

Nathan and Naveen for their unfailing love and encouragement. Finally, I thank Shilpa Panth for her love and support through some trying times, especially the last year or so. I am so lucky to have met you, and I hope I can be as supportive when you need it.

ASWIN N. VENKAT

University of Wisconsin–Madison

October 2006

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Large engineering systems typically consist of a number of subsystems that interact with each other as a result of material, energy and information flows. A high performance control technology such as model predictive control (MPC) is employed for control of these subsystems. Local models and objectives are selected for each individual subsystem. The interactions among the subsystems are ignored during controller design. In plants where the subsystems interact weakly, local feedback action provided by these subsystem (decentralized) controllers may be sufficient to overcome the effect of interactions. For such cases, a decentralized control strategy is expected to work adequately. For many plants, ignoring the interactions among subsystems leads to a significant loss in control performance. An excellent illustration of the hazards of such a decentralized control structure was the failure of the North American power system resulting in the blackout of August $14, 2003$. The decentralized control structure prevented the interconnected control areas from taking emergency control actions such as selective load shedding. As each subsystem tripped, the overloading of the remaining subsystems became progressively more severe, leading finally to the blackout. It has been reported by the U.S.-Canada Power System Outage Task Force (2004) that the extent of the cascading

system failure was so drastic that within 7 minutes the blackout rippled from the Cleveland-Akron area in northern Ohio to much of northeastern USA and Canada. In many situations, such catastrophic network control failures are prevented by employing conservative design choices. Conservative controller design choices are expensive and reduce productivity.

An obvious recourse to decentralized control is to attempt centralized control of large-scale systems. Centralized controllers, however, are viewed by most practitioners as monolithic and inflexible. For most large-scale systems, the primary hurdles to centralized control are not computational but organizational. Operators are usually unwilling to deal with the substantial data collection and data handling effort required to design and maintain a valid centralized control system for a large plant. To the best of our knowledge, no such centralized control systems are operational today for any large, networked system. Operators of large, networked systems also want to be able to take the different subsystems offline for routine maintenance and repair without affecting a complete plantwide control system shutdown. This is not easily accomplished under centralized MPC. In many applications, plants are already in operation with decentralized MPCs in place. Plant personnel do not wish to engage in a complete control system redesign required to implement centralized MPC. In some cases, different parts of the networked system are owned by different organizations making the model development and maintenance effort required for centralized control impractical. Unless these organizational impediments change in the future, centralized control of large, networked systems is useful primarily as a benchmark against which other control strategies can be compared and assessed.

For each decentralized MPC, a sequence of open-loop controls are determined through the solution of a constrained optimal control problem. A local objective is used. A subsystem

model, which ignores the interactions, is used to obtain a prediction of future process behavior along the control horizon. Feedback is usually obtained by injecting the first input move. When new local measurements become available, the optimal control problem is resolved and a fresh forecast of the subsystem trajectory is generated. For distributed control, one natural advantage that MPC offers over other controller paradigms is its ability to generate a prediction of future subsystem behavior. If the likely influence of interconnected subsystems is known, each local controller can possibly determine suitable feedback action that accounts for these external influences. Intuitively, one expects this additional information to help improve systemwide control performance. In fact one of the questions that we will answer in this dissertation is the following: *Is communication of predicted behavior of interconnected subsystems sufficient to improve systemwide control performance?*

The goal of this dissertation is to develop a framework for control of large, networked systems through the suitable integration of subsystem-based MPCs. For the distributed MPC framework proposed here, properties such as feasibility, optimality and closed-loop stability are established. The approach presented in this dissertation is aimed at allowing practitioners to build on existing infrastructure. The proposed distributed MPC framework also serves to equip the practitioner with a low-risk strategy to explore the benefits attainable with centralized control using subsystem-based MPCs.

1.1 Organization and highlights of this dissertation

The remainder of this dissertation is organized as follows:

Chapter 2.

Current literature on distributed MPC is reviewed in this chapter. Shortcomings of available distributed MPC formulations are discussed. Developments in the area of distributed state estimation are investigated. Finally, contributions to closed-loop stability theory for MPC are examined.

Chapter 3.

This chapter motivates distributed MPC methods developed in this dissertation. Two examples are also provided. First, an example consisting of two interacting chemical plants is presented to illustrate the disparity in performance between centralized and decentralized MPC. Next, a four area power system is used to show that modeling the interactions between subsystems and exchange of trajectories among MPCs (pure communication) is insufficient to provide even closed-loop stability.

Chapter 4.

A state feedback distributed MPC framework with guaranteed feasibility, optimality and closed-loop stability properties is described. An algorithm for distributed MPC is presented. It is shown that the distributed MPC algorithm can be terminated at any intermediate iterate; on iterating to convergence, optimal, centralized MPC performance is achieved.

Chapter 5.

The distributed MPC framework described in Chapter 4 is expanded to include state estimation. Two distributed state estimation strategies are described. Robustness of the distributed

estimator-distributed regulator combination to decaying state estimate error is demonstrated.

Chapter 6.

In this chapter, we focus on the problem of achieving zero offset objectives with distributed MPC. For large, networked systems, the number of measurements typically exceeds the number of manipulated variables. Offset-free control can be achieved for at most a subset of the measured variables. Conditions for appropriate choices of controlled variables that enable offset-free control with local disturbance models are described. A distributed target calculation algorithm that enables calculation of the steady-state targets at the subsystem level is presented.

Chapter 7.

The control actions generated by the MPCs are not usually injected directly into the plant but serve as setpoints for lower level flow controllers. In addition to horizontal integration across subsystems, system control performance may be improved further by vertically integrating each subsystem's MPC with its lower level flow controllers. Structural simplicity of the resulting controller network is a key consideration for vertical integration. The concept of *partial cooperation* is introduced to tackle vertical integration between MPCs.

Chapter 8.

The distributed MPC algorithm introduced in Chapter 4 is augmented to allow asynchronous optimization. This feature enables the integration of MPCs with disparate computational time requirements without forcing all MPCs to operate at the slowest computational rate. This fea-

ture also avoids the need for synchronized clock keeping at each iterate. Because all MPCs are required to exchange information periodically only, the communication load (between MPCs) is reduced.

Chapter 9.

Algorithms for distributed constrained LQR (DCLQR) are described in this chapter. These algorithms achieve infinite horizon optimal control performance at convergence using finite values of the control horizon, N. To formulate a tractable DCLQR optimization problem, the system inputs are parameterized using the unconstrained, optimal, centralized feedback control law. Two flavors for implementable DCLQR algorithms are considered. First, an algorithm in which a terminal set constraint is enforced explicitly is described. Next, algorithms for which the terminal set constraint remains implicit through the choice of N are presented. Advantages and disadvantages of either approach are discussed.

Chapter 10.

In this chapter, we utilize distributed MPC for power system automatic generation control (AGC). A modeling framework suitable for power networks is used. Both terminal penalty and terminal control distributed MPC are evaluated. It is shown that the distributed MPC strategies proposed also allow coordination of the flexible AC transmission system controls with AGC.

Chapter 11.

In this chapter, we consider the problem of integrating MPCs with different sampling rates. Asynchronous feedback distributed MPC allows MPCs to inject appropriate control actions at their respective sampling rates. This feature enables one to achieve performance superior to centralized MPC designed at the slowest sampling rate. Algorithms for fast sampled and slow sampled MPCs are described. Nominal asymptotic stability for the asynchronous feedback distributed MPC control law is established.

Chapter 12.

This chapter summarizes the contributions of this dissertation and outlines possible directions for future research.

Chapter 2

Literature review

Model predictive control (MPC) is a process control technology that is being increasingly employed across several industrial sectors (Camacho and Bordons, 2004; Morari and Lee, 1997; Qin and Badgwell, 2003; Young, Bartusiak, and Fontaine, 2001). The popularity of MPC in industry stems in part from its ability to tackle multivariable processes and handle process constraints. At the heart of MPC is the process model and the concept of open-loop optimal feedback. The process model is used to generate a prediction of future subsystem behavior. At each time step, past measurements and inputs are used to estimate the current state of the system. An optimization problem is solved to determine an optimal open-loop policy from the present (estimated) state. Only the first input move is injected into the plant. At the subsequent time step, the system state is re-estimated using new measurements. The optimization problem is resolved and the optimal open-loop policy is recomputed. Figure 2.1 presents a conceptual picture of MPC.

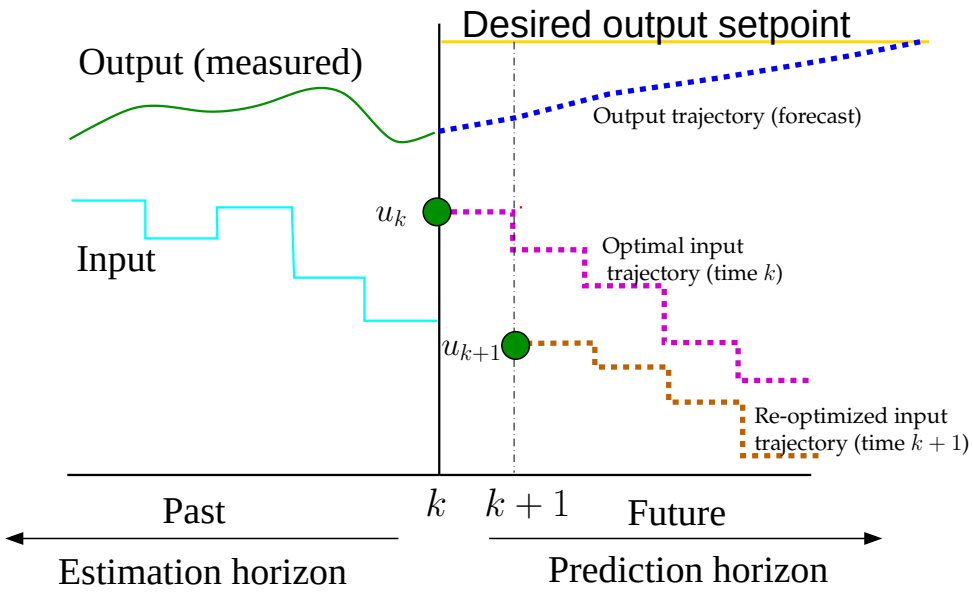

Figure 2.1: A conceptual picture of MPC. Only u_k is injected into the plant at time k. At time $k+1$, a new optimal trajectory is computed.

Distributed MPC.

The benefits and requirements for cross-integration of subsystem MPCs has been discussed in Havlena and Lu (2005); Kulhavý, Lu, and Samad (2001). A two level decomposition-coordination strategy for generalized predictive control, based on the master-slave paradigm was proposed in Katebi and Johnson (1997). A plantwide control strategy that involves the integration of linear and nonlinear MPC has been described in Zhu and Henson (2002); Zhu, Henson, and Ogunnaike (2000). A distributed MPC framework, for control of systems in which the dynamics of each of the subsystems are independent (decoupled) but the local state and control variables of the subsystems are nonseparably coupled in the cost function, was proposed in Keviczky, Borelli, and Balas (2005). In the distributed MPC framework described in Keviczky et al. (2005), each subsystem's MPC computes optimal input trajectories for itself and for all its neighbors. A sufficient condition for stability has also been established. Ensur-

ing the stability condition in Keviczky et al. (2005) is satisfied is, however, a nontrivial exercise. Furthermore, as noted by the authors, the stability condition proposed in Keviczky et al. (2005) has some undesirable consequences: (i) Satisfaction of the stability condition requires increasing information exchange rates as the system approaches equilibrium; this information exchange requirement to preserve nominal stability is counter-intuitive. (ii) Increasing the prediction horizon may lead to instability due to violation of the stability condition; closed-loop performance deteriorates after a certain horizon length. A globally feasible, continuous time distributed MPC framework for multi-vehicle formation stabilization was proposed in Dunbar and Murray (2006). In this problem, the subsystem dynamics are decoupled but the states are nonseparably coupled in the cost function. Stability is assured through the use of a compatibility constraint that forces the assumed and actual subsystem responses to be within a pre-specified bound of each other. The compatibility constraint introduces a fair degree of conservatism and may lead to performance that is quite different from the optimal, centralized MPC performance. Relaxing the compatibility constraint leads to an increase in the frequency of information exchange among subsystems required to ensure stability. The authors claim that each subsystem's MPC needs to communicate only with its neighbors is a direct consequence of the assumptions made: the subsystem dynamics are decoupled and only the states of the neighbors affect the local subsystem stage cost. A decentralized MPC algorithm for systems in which the subsystem dynamics and cost function are independent of the influence of other subsystem variables but have coupling constraints that link the state and input variables of different subsystems has been proposed in Richards and How (2004). Robust feasibility is established when the disturbances are assumed to be independent, bounded and a fixed, sequential ordering for the subsystems' MPC optimizations is allowed.

A distributed MPC algorithm for unconstrained, linear time-invariant (LTI) systems in which the dynamics of the subsystems are influenced by the states of interacting subsystems has been described in Camponogara, Jia, Krogh, and Talukdar (2002); Jia and Krogh (2001). A contractive state constraint is employed in each subsystem's MPC optimization and asymptotic stability is guaranteed if the system satisfies a matrix stability condition. An algorithmic framework for partitioning a plant into suitably sized subsystems for distributed MPC has been described in Motee and Sayyar-Rodsari (2003). An unconstrained, distributed MPC algorithm for LTI systems is also described. However, convergence, optimality and closed-loop stability properties, for the distributed MPC framework described in Motee and Sayyar-Rodsari (2003), have not been established. A distributed MPC strategy, in which the effects of the interacting subsystems are treated as bounded uncertainties, has been described in Jia and Krogh (2002). Each subsystem's MPC solves a min-max optimization problem to determine local control policies. The authors show feasibility of their distributed MPC formulation; optimality and closed-loop stability properties are, however, unclear. Recently in Dunbar (2005), an extension of the distributed MPC framework described in Dunbar and Murray (2006) that handles systems with interacting subsystem dynamics was proposed. At each time step, existence of a feasible input trajectory is assumed for each subsystem. This assumption is one limitation of the formulation. Furthermore, the analysis in Dunbar (2005) requires at least 10 agents for closed-loop stability. This lower bound on the number of agents (MPCs) is an undesirable and artificial restriction and limits the applicability of the method. In Magni and Scattolini (2006), a completely decentralized state feedback MPC framework for control of nonlinear systems was proposed. A contractive state constraint is used to ensure stability. It is assumed in Magni and Scattolini (2006) that no information exchange among subsystems is

possible. An attractive feature of this approach is the complete decentralization of the MPCs. The requirement of stability with no communication leads to rather conservative conditions for feasibility of the contractive constraint and closed-loop stability, that may be difficult to verify in practice. Optimality properties of the formulation have not been established and remain unclear. For the distributed MPC strategies available in the literature, nominal properties such as feasibility, optimality and closed-loop stability have not all been established for any single distributed MPC framework. Moreover, all known distributed MPC formulations assume perfect knowledge of the states (state feedback) and do not address the case where the states of each subsystem are estimated from local measurements (output feedback). In Chapters 5 and 6, we investigate distributed MPC with state estimation and disturbance modeling.

To arrive at distributed MPC algorithms with guaranteed feasibility, stability and performance properties, we also examine contributions to the area of plantwide decentralized control. Several contributions have been made in the area. A survey of decentralized control methods for large-scale systems can be found in Sandell-Jr., Varaiya, Athans, and Safonov (1978). Performance limitations arising due to the decentralized control framework has been described in Cui and Jacobsen (2002). Several decentralized controller design approaches approximate or ignore the interactions between the various subsystems and lead to a suboptimal plantwide control strategy (Acar and Ozguner, 1988; Lunze, 1992; Samyudia and Kadiman, 2002; Siljak, 1991). The required characteristics of any problem solving architecture in which the agents are autonomous and influence one another's solutions has been described in Talukdar, Baerentzen, Gove, and de Souza (1996).

State estimation, disturbance modeling and target calculation for distributed MPC.

All the states of a large, interacting system cannot usually be measured. Consequently, estimating the subsystem states from available measurements is a key component in any practical MPC implementation. Theory for centralized linear estimation is well understood. For large-scale systems, organizational and geographic constraints may preclude the use of centralized estimation strategies. The centralized Kalman filter requires measurements from all subsystems to estimate the state. For large, networked systems, the number of measurements is usually large to meet redundancy and robustness requirements. One difficulty with centralized estimation is communicating voluminous local measurement data to a central processor where the estimation algorithm is executed. Another difficulty is handling the vast amounts of data associated with centralized processing. Parallel solution techniques for estimation are available (Lainiotis, 1975; Lainiotis, Plataniotis, Papanikolaou, and Papaparaskeva, 1996). While these techniques reduce the data transmission requirement, a central processor that updates the overall system error covariances at each time step is still necessary. Analogous to centralized control, the optimal, centralized estimator is a benchmark for evaluating the performance of different distributed estimation strategies. A decentralized estimator design framework for large-scale systems was proposed in Sundareshan (1977); Sundareshan and Elbanna (1990); Sundareshan and Huang (1984). Local estimators were designed based on the decentralized dynamics and additional compensatory inputs were included for each estimator to account for the interactions between the subsystems. Estimator convergence was established under assumptions on either the strength of the interconnections or the structure of the interconnection matrix. A decentralized estimator design strategy, in which the interconnections are

treated as unknown inputs was proposed in Saif and Guan (1992); Viswanadham and Ramakrishna (1982) for a restricted class of systems where the interconnections satisfy certain algebraic conditions and the number of outputs, for each subsystem, is greater than the number of interacting inputs.

Disturbance models are used to eliminate steady-state offset in the presence of nonzero mean, constant disturbances and/or plant-model mismatch. The output disturbance model is the most widely used disturbance model in industry to achieve zero offset control performance at steady state (Cutler and Ramaker, 1980; García and Morshedi, 1986; Richalet, Rault, Testud, and Papon, 1978). It is well known that output disturbance models cannot be used in plants with integrating modes as the effects of the augmented disturbance cannot be distinguished from the plant integrating modes. An alternative is to use input disturbance models (Davison and Smith, 1971), where the disturbances are assumed to enter the system through the inputs. For single (centralized) MPCs, Muske and Badgwell (2002); Pannocchia and Rawlings (2002) derive conditions that guarantee zero offset control, using suitable disturbance models, in the presence of unmodelled effects and/or nonzero mean disturbances. In a distributed MPC framework, many choices for disturbance models exist. From a practitioner's standpoint, it is usually convenient to use local integrating disturbances. To track nonzero output setpoints, we require input and state targets that bring the system to the desired output targets at steady state. One option for determining the optimal steady-state targets in a distributed MPC framework is to perform a centralized target calculation (Muske and Rawlings, 1993) using the composite model for the plant. Alternatively, the target calculation problem can be formulated in a distributed manner with all the subsystem targets computed locally. A discussion on distributed target calculation is provided in Chapter 6.

Closed-loop stability for MPC.

The idea of designing optimal, open-loop, feedback controllers has been studied in the automatic control community for nearly four decades. The focus of initial work in the area was stabilization of unconstrained linear time-varying systems (Kleinman, 1970; Kwon and Pearson, 1978; Kwon, Bruckstein, and Kailath, 1983). The earliest known stability result for constrained systems was by Chen and Shaw (1982), who used a terminal equality constraint to stabilize nonlinear discrete time systems. The initial popularity of MPC was primarily due to interesting applications in the process industry (Cutler and Ramaker, 1980; García and Prett, 1986; Richalet et al., 1978). Several MPCs were successfully implemented, even though no stability guarantees were available at the time.

Theory for MPC has evolved significantly over the years. Review articles tracing the progress in the area are available (García, Prett, and Morari, 1989; Mayne, Rawlings, Rao, and Scokaert, 2000; Morari and Lee, 1997). A few well known recipes for guaranteeing closed-loop stability with MPC are available for single (centralized) MPCs. The commonly used techniques for ensuring stability with MPC are terminal constraint MPC, terminal penalty MPC and terminal control MPC. Terminal constraint MPC (Kwon and Pearson, 1978) achieves stability by employing an additional state constraint that forces the predicted system state at the end of the control horizon to be at the origin. In Keerthi and Gilbert (1986), a general stability analysis for terminal constraint MPC of constrained nonlinear discrete time systems is provided. An alternative strategy to guarantee closed-loop stability for MPC is to employ a stabilizing terminal penalty (Rawlings and Muske, 1993). Neither terminal constraint MPC nor terminal penalty MPC achieves infinite horizon optimal performance for a finite control horizon (N).

Subsequently for small values of N, there may be significant mismatch between the predicted system trajectory and the actual closed-loop response. This mismatch is known to complicate controller tuning. To achieve infinite horizon optimal performance with finite values of N, a terminal control MPC framework was proposed (Chmielewski and Manousiouthakis, 1996; Scokaert and Rawlings, 1998; Sznaier and Damborg, 1990). Terminal control MPC relies on solving a finite dimensional optimization problem to compute a set of inputs that drives the predicted system state inside an invariant set in which the optimal unconstrained feedback law is feasible (and therefore optimal). Characterization of the maximum invariant set satisfying the above mentioned property is possible in most cases and algorithms to approximate the maximal admissible set have been proposed in Gilbert and Tan (1991); Gutman and Cwikel (1987).

Exponential stability for the output feedback MPC control law is established in Scokaert, Rawlings, and Meadows (1997) using perturbed stability results for linear systems (Halanay, 1963). Lipschitz continuity of the control law w.r.t the state is a key requirement to establish exponential stability. In Meadows (1994), Lipschitz continuity of the control law for a single (centralized) linear MPC is proved using (Hager, 1979, Theorem 3.1). A limitation of the approach in Meadows (1994) is that (Hager, 1979, Theorem 3.1) assumes the set of active constraints to be linearly independent. This is usually difficult to ensure in practice and consequently, Lipschitz properties of the control law are difficult to establish in general. In a recent work, Choi and Kwon (2003) prove exponential stability for single (centralized) output feedback MPCs by constructing a single Lyapunov function. The attractive feature of the approach presented in Choi and Kwon (2003) is that Lipschitz continuity of the control law is not assumed.

Chapter 3

Motivation

Decentralized MPC is attractive to practitioners because it requires only local process data for controller design and model maintenance. Furthermore, routine maintenance operations such as taking units offline for repair are achieved easily under decentralized MPC. There is one well known caveat however: the performance of decentralized MPC is usually far from optimal when the subsystems interact significantly. Centralized MPC, on the other hand, achieves optimal nominal control for any system. However, as discussed in Chapter 1, centralized MPC viewed by most practitioners as impractical and unsuitable for control of large, networked systems.

To impact today's highly competitive markets, practitioners are constantly striving to push limits of performance. In cases where the subsystems are interacting, the control performance of centralized and decentralized MPC may differ significantly. In Section 3.1, we consider an example consisting of two networked styrene polymerization plants. In this example the control performance of decentralized and centralized MPC differ significantly. Such examples are not uncommon. Integrating subsystem-based MPCs has been recognized as a possible avenue for improving systemwide control performance (Havlena and Lu, 2005; Kulhavý et al.,

2001; Lu, 2000). One of the goals of this dissertation is to develop a distributed MPC framework with guaranteed performance properties *i.e.*, an assured performance improvement over decentralized MPC, and capable of approaching centralized MPC performance.

Another motivation for this work is the current state of distributed MPC. Most distributed MPC formulations in the literature are based on the assumption that transmitting predicted trajectory information among subsystems (pure *communication*) is sufficient to improve systemwide control performance. In Section 3.2, a four area power system for which communication-based MPC is closed-loop unstable is presented. In this dissertation, other examples for which communication-based MPC either fails or gives unacceptable closed-loop performance are provided. A reliable distributed MPC strategy with provable feasibility, optimality and closed-loop stability properties is required. Furthermore, all known distributed MPC frameworks require state feedback and do not address the more realistic scenario in which the subsystem states are estimated from measurements. To the best of our knowledge, ensuring offset-free control with distributed MPC is an important issue that has not been addressed in the literature.

3.1 Networked chemical processes with large potential for performance improvement

An example in which chemical systems are integrated as a result of material and energy flows between them is considered. In today's increasingly competitive markets, such inter-plant integration (in addition to plantwide integration) may assume significant economic relevance. A simplified scenario with two interacting plants shown in Figure 3.1 is considered. The first

plant consists of a styrene polymerization reactor producing grade A (a lower grade) of the polymer. Representative publications on the modeling and control of styrene polymerization reactors are Hidalgo and Brosilow (1990) and Russo and Bequette (1998). Based on supply and demand economics, a fraction of the lower grade polymer is transported to the second plant where a higher grade polymer is produced (grade B). The unreacted monomer and initiator are separated from the product polymer and recycled to the first plant. Transport of material between the plants is associated with significant time delays. Time delays coupled with complications arising due to recycle dynamics make the control problem challenging (Luyben, 1993a,b, 1994). Reductions in inventory and operational costs may dictate the need for such integrated schemes however. In the absence of integration, either MPC is a centralized controller. The MPCs employ a model obtained by linearizing the corresponding plant around the desired steady state. In this example, the two polymerization plants operate at different steady states, each of which correspond to the grade of polymer to be produced. Once the two plants are linked, the MPCs ignore the inter-plant interactions and function as two decentralized MPCs. The MPC in the first plant controls the temperature of the styrene polymerization reactor (T_1) by manipulating the initiator flow rate F_{init_0}. The MPC in the second plant controls the temperature in the polymerization reactor (T_2), monomer concentration at the top of the distillation column (C_{m_r}) and the sum of the monomer and initiator concentrations at the bottom of the column $(C_{\text{m}_{\text{bot}}} + C_{\text{init}_{\text{bot}}})$ by manipulating the initiator flow rate to the reactor (F_{init_2}), recycle flow rate (F_{recy}) and vapor boilup flow rate (V). The time delay due to transport of material between the plants is 1.1 hrs. Each MPC employs a Kalman filter to estimate the states of the system from local measurements. Input constraints are given in Table 3.1.

The control performance of decentralized and centralized MPC are compared when

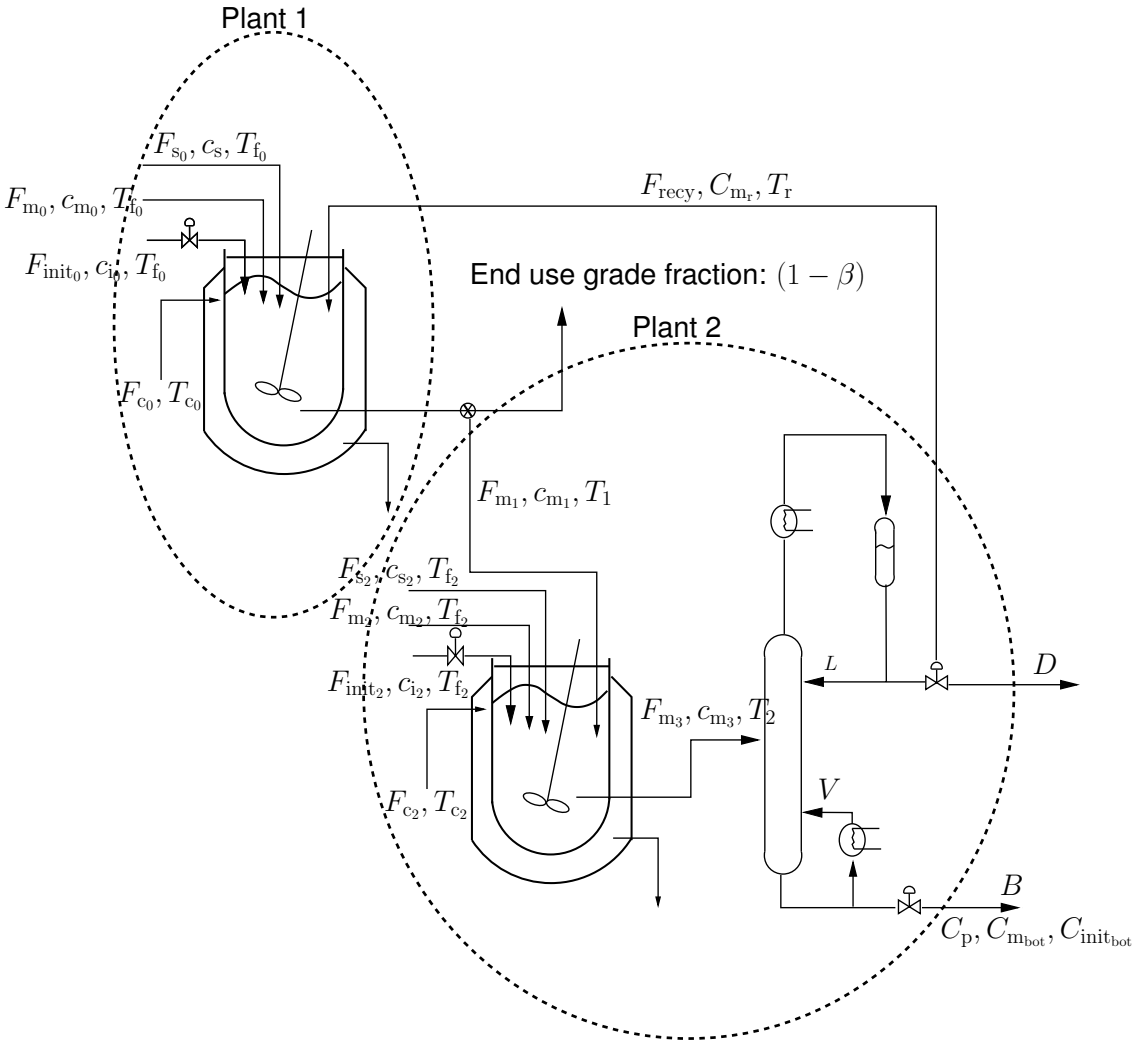

Figure 3.1: Interacting styrene polymerization processes. Low grade manufacture in first plant. High grade manufacture in second plant with recycle of monomer and solvent.

Table 3.1: Two integrated styrene polymerization plants. Input constraints.

$-0.25 \leq F_{\text{init}_0} \leq 0.25$
$-0.4 \leq F_{\text{init}_2} \leq 0.4$
$-0.4 \leq F_{\text{recy}} \leq 0.4$
$-0.4 \leq V \leq 0.4$

the setpoint temperatures of the two polymerization reactors are decreased by $-10°C$ and $-5°C$ respectively. The performance of decentralized and centralized MPC for temperature control in the two polymerization reactors is shown in Figure 3.2. Under decentralized MPC,

the inputs F_{init_0} and F_{recy} remain saturated until time $\sim$ 42 hrs and $\sim$ 20 hrs respectively. Consequently, the reactors under decentralized MPC take more than 50 hrs to settle at their new temperature setpoints. Under centralized MPC, the reactor temperatures are within 0.5% of the respective temperature setpoints in about $\sim$ 7 hrs. The same qualitative behavior is also observed with the other two outputs. Under centralized MPC, outputs C_{m_r} and $C_{\text{m}_{\text{bot}}} + C_{\text{init}_{\text{bot}}}$ track their respective set points in $\sim$ 25 hrs. Tracking the concentration setpoints takes over 60 hrs with decentralized MPC. The control costs with decentralized and centralized MPC are given in Table 3.2. Decentralized MPC gives unacceptable closed-loop performance for this example.

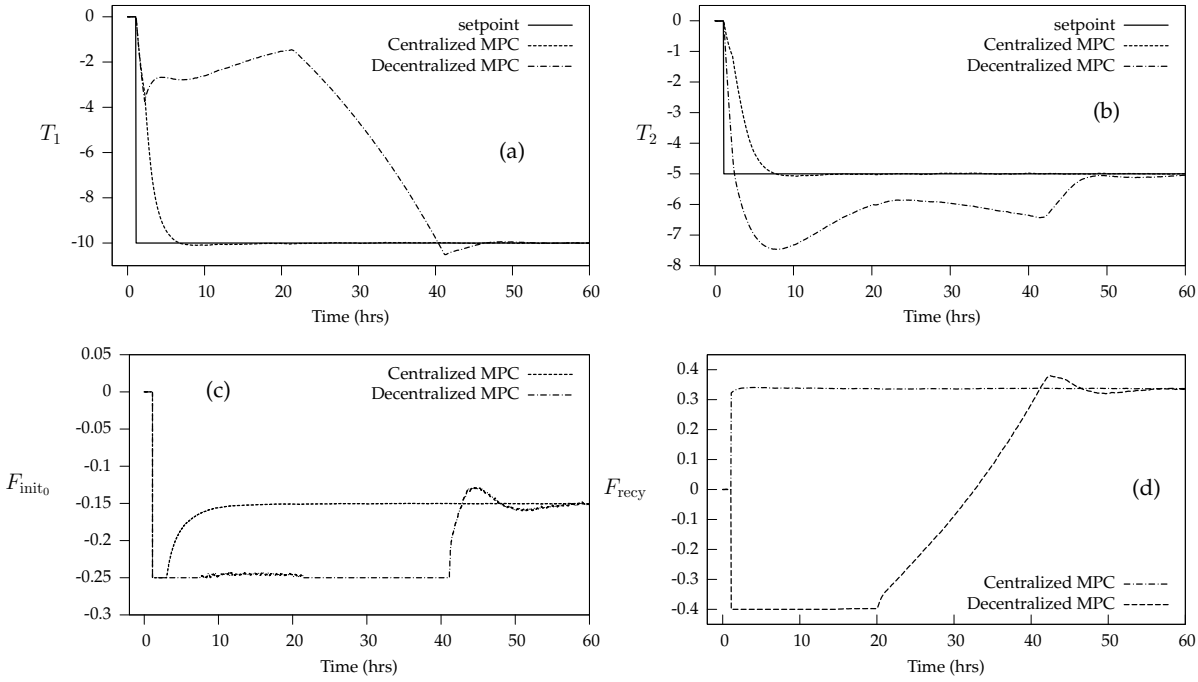

Figure 3.2: Interacting polymerization processes. Temperature control in the two polymerization reactors. (a) Temperature control in reactor 1. (b) Temperature control in reactor 2. (c) Initiator flowrate to reactor 1. (d) Recycle flowrate.

Table 3.2: Performance comparison of centralized and decentralized MPC

	Λ_{cost}	Performance loss (w.r.t centralized MPC)
Centralized-MPC	18.84	-
Decentralized-MPC	1608	8400%

3.2 Instability with communication-based MPC : Example of a four area power system

Consider the four area power system shown in Figure 3.3. A description of the model for each control area is given in Section 10.4 (Chapter 10, p. 243). Model parameters are given in Table A.1 (Appendix A). In each control area, a change in local power demand (load) alters the nominal operating frequency. The MPC in each control area i manipulates the load reference setpoint P_{ref_i} to drive the frequency deviations $\Delta\omega_i$ and tie-line power flow deviations $\Delta P_{\text{tie}}^{ij}$ to zero. Power flow through the tie lines gives rise to interactions among the control areas. Hence a load change in area 1, for instance, causes a transient frequency change in all control areas.

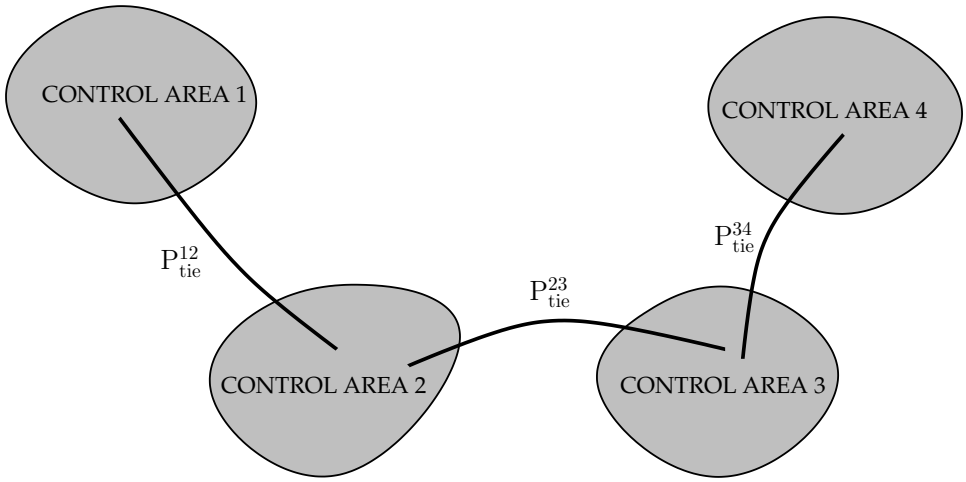

Figure 3.3: Four area power system.

The performance of centralized MPC (cent-MPC) and communication-based MPC (comm MPC) are compared for a 25% load increase in area 2 and a simultaneous 25% load drop in area 3. This load disturbance occurs at 5 sec. For each MPC, we choose a prediction horizon $N = 20$. In comm-MPC, the load reference setpoint $(\Delta P_{\mathrm{ref}_i})$ in each area is manipulated to reject the load disturbance and drive the change in local frequencies $(\Delta\omega_i)$ and tie-line power flows $(\Delta P_{\mathrm{tie}}^{ij})$ to zero. In the cent-MPC framework, a single MPC manipulates all four $\Delta P_{\mathrm{ref}_i}$. The load reference setpoint for each area is constrained between ± 0.5.

The performance of cent-MPC and comm-MPC are shown in Figure 3.4. Only $\Delta\omega_2$ and $\Delta P_{\mathrm{tie}}^{23}$ are shown as the frequency and tie-line power flow deviations in the other areas display similar qualitative behavior. Likewise, only $\Delta P_{\mathrm{ref}_2}$ and $\Delta P_{\mathrm{ref}_3}$ are shown as other load reference setpoints behave similarly. Under comm-MPC, the load reference setpoints for areas 2 and 3 switch repeatedly between their upper and lower saturation limits. Consequently, the power system network is unstable under comm-MPC. Cent-MPC, on the other hand, is able to reject the load disturbance and achieves good closed-loop performance.

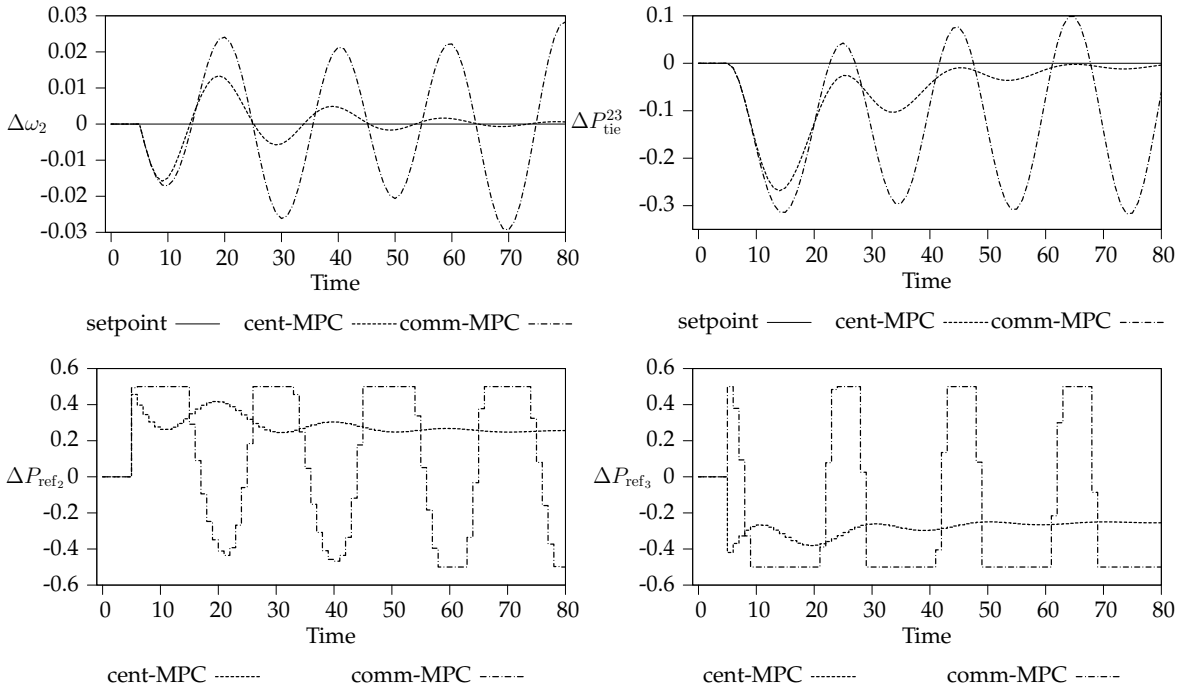

Figure 3.4: Four area power system. Performance of centralized and communication-based MPC rejecting a load disturbance in areas 2 and 3. Change in frequency $\Delta\omega_2$, tie-line power flow $\Delta P_{\text{tie}}^{23}$ and load reference setpoints $\Delta P_{\text{ref}_2}, \Delta P_{\text{ref}_3}$.

Chapter 4

State feedback distributed MPC [1]

In this chapter, we describe a new approach for controlling large, networked systems through the integration of subsystem-based MPCs. The proposed distributed MPC framework is iterative with the subsystem-based MPC optimizations executed in parallel. It is assumed that the interactions between the subsystems are stable. This assumption presumes the feasibility of a decentralized, manipulated variable (MV)-controlled variable (CV) design in which open-loop unstable modes, if any, are controlled and not allowed to evolve open loop. System redesign is recommended if such an initial design is not possible. The term *iterate* indicates a set of MPC optimizations executed in parallel (one for each subsystem) followed by an exchange of information among interconnected subsystems. We show that the distributed MPC algorithm can be terminated at any intermediate iterate to allow for computational or communication limits. At convergence, the distributed MPC algorithm is shown to achieve optimal, centralized MPC performance.

This chapter is organized as follows. First, a modeling framework suitable for distributed MPC is described. Next, the different candidate MPC formulations for systemwide

[1] **Portions of this chapter appear in Venkat, Rawlings, and Wright (2005b) and in Venkat, Rawlings, and Wright (2006f).**

control are described. We provide further proof and answer why modeling the interactions between subsystems and exchanging trajectory information among MPCs ((pure) communication) is insufficient to provide even closed-loop stability. We then proceed to characterize optimality conditions for distributed MPC and present an algorithm for distributed MPC. Closed-loop properties for the distributed MPC framework under state feedback are established subsequently. Three examples are presented to highlight the benefits of the described approach. Finally, we summarize the contributions of this chapter and present some extensions for the proposed distributed MPC framework.

4.1 Interaction modeling

Consider a plant comprised of M subsystems. The symbol $\mathbb{I}_M$ denotes the set of integers $1, 2, \ldots, M$.

Decentralized models. Let the decentralized (local) model for each subsystem $i \in \mathbb{I}_M$ be represented by a discrete, linear time invariant (LTI) model of the form

$$x_{ii}(k+1) = A_{ii}x_{ii}(k) + B_{ii}u_i(k)$$

$$y_i(k) = C_{ii}x_{ii}(k)$$

in which k is discrete time, and we assume $(A_{ii} \in \mathbb{R}^{n_{ii}\times n_{ii}}, B_{ii} \in \mathbb{R}^{n_{ii}\times m_i}, C_{ii} \in \mathbb{R}^{z_i\times n_{ii}})$ is a realization for each (u_i, y_i) input-output pair such that (A_{ii}, B_{ii}) is stabilizable and (A_{ii}, C_{ii}) is detectable.

Interaction models (IM). Consider any subsystem $i \in \mathbb{I}_M$. We represent the effect of any interacting subsystem $j \in \mathbb{I}_M, j \neq i$ on subsystem i through a discrete LTI model of the form

$$x_{ij}(k+1) = A_{ij}x_{ij}(k) + B_{ij}u_j(k)$$

The output equation for each subsystem is written as $y_i(k) = \sum_{j=1}^{M} C_{ij}x_{ij}(k)$. The model $(A_{ij} \in \mathbb{R}^{n_{ij}\times n_{ij}}, B_{ij} \in \mathbb{R}^{n_{ij}\times m_j}, C_{ij} \in \mathbb{R}^{z_i \times n_{ij}})$ is a minimal realization of the input-output pair (u_j, y_i).

Composite models (CM). The combination of the decentralized model and the interaction models for each subsystem yields the composite model (CM). The decentralized state vector x_{ii} is augmented with states arising due to the effects of all other subsystems.

Let $x_i = [x_{i1}', \ldots, x_{ii}', \ldots, x_{iM}']' \in \mathbb{R}^{n_i}$ denote the CM states for subsystem i. For notational simplicity, we represent the CM for subsystem i as

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j\neq i} W_{ij}u_j(k) \tag{4.1a}$$

$$y_i(k) = C_i x_i(k) \tag{4.1b}$$

in which $C_i = [C_{i1} \dots C_{ii} \dots C_{iM}]$ and

$$A_i = \begin{bmatrix} A_{i1} & & & & \\ & \ddots & & & \\ & & A_{ii} & & \\ & & & \ddots & \\ & & & & A_{iM} \end{bmatrix}, \quad B_i = \begin{bmatrix} 0 \\ \vdots \\ B_{ii} \\ 0 \\ \vdots \end{bmatrix}, \quad W_{ij} = \begin{pmatrix} 0 \\ \vdots \\ B_{ij} \\ 0 \\ \vdots \end{pmatrix}$$

The composite model (CM) for the entire plant is written as

$$\underbrace{\left[\begin{array}{c} x_{11} \\ \vdots \\ x_{1M} \\ \hline \vdots \\ \hline x_{M1} \\ \vdots \\ x_{MM} \end{array}\right]}_{x_{\text{cm}}} (k+1) = \underbrace{\left[\begin{array}{ccc|c|c} A_{11} & & & & \\ & \ddots & & & \\ & & A_{1M} & & \\ \hline & & & \ddots & \\ \hline & & & A_{M1} & \\ & & & & \ddots \\ & & & & A_{MM} \end{array}\right]}_{A_{\text{cm}}} \left[\begin{array}{c} x_{11} \\ \vdots \\ x_{1M} \\ \hline \vdots \\ \hline x_{M1} \\ \vdots \\ x_{MM} \end{array}\right] (k)$$

$$+ \underbrace{\left[\begin{array}{ccc} B_{11} & & \\ & \ddots & \\ & & B_{1M} \\ \hline & \vdots & \\ \hline B_{M1} & & \\ & \ddots & \\ & & B_{MM} \end{array}\right]}_{B_{\text{cm}}} \left[\begin{array}{c} u_1 \\ \vdots \\ u_M \end{array}\right] (k) \tag{4.2a}$$

$$\left[\begin{array}{c} y_1 \\ \vdots \\ y_M \end{array}\right] (k) = \underbrace{\left[\begin{array}{ccc|c|ccc} C_{11} & \cdots & C_{1M} & & & & \\ & & & \ddots & & & \\ & & & & C_{M1} & \cdots & C_{MM} \end{array}\right]}_{C_{\text{cm}}} \left[\begin{array}{c} x_{11} \\ \vdots \\ x_{1M} \\ \hline \vdots \\ \hline x_{M1} \\ \vdots \\ x_{MM} \end{array}\right] (k) \tag{4.2b}$$

After identification of the significant interactions from closed-loop operating data, we expect that many of the interaction terms will be zero. In the decentralized model, all of the interaction terms are zero. Further discussion of closed-loop identification procedures for distributed MPC can be found in Gudi and Rawlings (2006).

Centralized model. The centralized model is represented as

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k)$$

4.2 Notation and preliminaries

For any $i \in \mathbb{I}_M$ the notation $j \neq i$ indicates that j can take all values in $\mathbb{I}_M$ except $j = i$. Let $\mathbb{I}_+$ denote the set of positive integers. Given a bounded set Λ, $\mathrm{int}(\Lambda)$ denotes the interior of the set. For any two vectors $r, s \in \mathbb{R}^n$, the notation $\langle r, s \rangle$ represents the inner product of the two vectors. For any arbitrary, finite set of vectors $a_1, a_2, \ldots, a_s$, define $\mathrm{vec}(a_1, a_2, \ldots, a_s) = [a_1{}', a_2{}', \ldots, a_s{}']'$.

Lemma 4.1. *Let $\mathcal{A}x = b$ be a system of linear equations with $\mathcal{A} \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m,\ m \leq n$. Consider $\mathcal{X} \subset \mathbb{R}^n$ nonempty, compact, convex with $0 \in \mathrm{int}(\mathcal{X})$. The set $\mathcal{B} \subseteq \mathrm{range}(\mathcal{A})$ is defined as $\mathcal{B} = \{b \mid \mathcal{A}x = b, x \in \mathcal{X}\}$. For every $b \in \mathcal{B}$, $\exists\ \overline{x}(b)$ dependent on b, and $K > 0$ independent of b such that $\mathcal{A}\overline{x}(b) = b,\ \overline{x}(b) \in \mathcal{X}$ and $\|\overline{x}(b)\| \leq K\|b\|$.*

A proof is given in Appendix 4.10.1.

Let the current (discrete) time be k. For any subsystem $i \in \mathbb{I}_M$, let the predicted state and input at time instant $k+j,\ j \geq 0$, based on data at time k be denoted by $x_i(k+j|k) \in \mathbb{R}^{n_i}$ and $u_i(k+j|k) \in \mathbb{R}^{m_i}$, respectively. The stage cost is defined as

$$L_i(x_i, u_i) = \frac{1}{2}\left[{x_i}' Q_i x_i + {u_i}' R_i u_i\right] \tag{4.3}$$

in which $Q_i \geq 0$, $R_i > 0$. Denote a closed **ball** of radius $\varepsilon > 0$ centered at $a \in \mathbb{R}^n$ by $B_\varepsilon(a) = \{x|\ \|x - a\| \leq \varepsilon\}$.

The notation $\mu(k)$ denotes the set of CM states $x_1(k), x_2(k), \ldots x_M(k)$ *i.e.*,

$$\mu(k) = [x_1(k), x_2(k), \ldots, x_M(k)].$$

With slight abuse of notation, we write $\mu(k) \in \mathcal{X}$ to denote $\mathrm{vec}(\mu(k)) = \mathrm{vec}(x_1(k), x_2(k), \ldots, \ldots, x_M(k)) \in \mathcal{X}$. The norm operator for $\mu(k)$ is defined as

$$\|\mu(k)\| = \|\mathrm{vec}(x_1(k), x_2(k), \ldots, x_M(k))\| = \sqrt{\sum_{i=1}^{M} \|x_i(k)\|^2}.$$

The following notation represents the predicted infinite horizon state and input trajectory vectors in the different MPC frameworks

$$\begin{aligned}
\text{Centralized state trajectory: } \boldsymbol{x}(k)^{'} &= \left[x(k+1|k)', x(k+2|k)', \ldots\ldots\right] \\
\text{Centralized input trajectory: } \boldsymbol{u}(k)^{'} &= \left[u(k|k)', u(k+1|k)', \ldots\ldots\right] \\
\text{CM state trajectory (subsystem } i\text{): } \boldsymbol{x}_i(k)^{'} &= \left[x_i(k+1|k)', x_i(k+2|k)', \ldots\ldots\right] \\
\text{Input trajectory (subsystem } i\text{): } \boldsymbol{u}_i(k)^{'} &= \left[u_i(k|k)', u_i(k+1|k)', \ldots\ldots\right] \\
\text{Decentralized state trajectory (subsystem } i\text{): } \boldsymbol{x}_{ii}(k)^{'} &= \left[x_{ii}(k+1|k)', x_{ii}(k+2|k)', \ldots\ldots\right]
\end{aligned}$$

Let N denote the control horizon. The finite horizon trajectories use an over bar to distinguish them from the corresponding infinite horizon trajectories *i.e.,*

$$\begin{aligned}
\text{Centralized state trajectory: } \boldsymbol{x}(k)^{'} &= \left[x(k+1|k)', x(k+2|k)', \ldots\ldots\right] \\
\text{Centralized input trajectory: } \boldsymbol{u}(k)^{'} &= \left[u(k|k)', u(k+1|k)', \ldots\ldots\right] \\
\text{CM state trajectory (subsystem } i\text{): } \boldsymbol{x}_i(k)^{'} &= \left[x_i(k+1|k)', x_i(k+2|k)', \ldots\ldots\right] \\
\text{Input trajectory (subsystem } i\text{): } \boldsymbol{u}_i(k)^{'} &= \left[u_i(k|k)', u_i(k+1|k)', \ldots\ldots\right] \\
\text{Decentralized state trajectory (subsystem } i\text{): } \boldsymbol{x}_{ii}(k)^{'} &= \left[x_{ii}(k+1|k)', x_{ii}(k+2|k)', \ldots\ldots\right]
\end{aligned}$$

Define

$$\mathcal{C}_N(A,B) = \begin{bmatrix} B & AB & \ldots & A^{N-1}B \end{bmatrix}.$$

Assumption 4.1. All interaction models are stable *i.e.,* for each $i, j \in \mathbb{I}_M$, $|\lambda_{\max}(A_{ij})| < 1, \forall j \neq i$.

4.3 Systemwide control with MPC

In this section, four MPC based systemwide control formulations are described. In each case, the controller is defined by implementing the first input in the solution to the corresponding optimization problem. Let $\Omega_i \subset \mathbb{R}^{m_i}$, the set of admissible controls for subsystem i, be a nonempty, compact, convex set containing the origin in its interior. The set of admissible controls for the whole plant Ω is the Cartesian product of the admissible control sets of each of the subsystems. It follows that Ω is a compact, convex set containing the origin in its interior. The constrained stabilizable set $\mathbb{X}$ is the set of all initial subsystem states $\mu = [x_1, x_2, \ldots, x_M]$ that can be steered to the origin by applying a sequence of admissible controls (see (Sznaier and Damborg, 1990, Definition 2)). In each MPC based framework, $\mu(0) \in \mathbb{X}$. Hence a feasible solution exists to the corresponding optimization problem.

$$\mathcal{P}_1: \quad \textbf{Centralized MPC}$$

$$\min_{\boldsymbol{x}(k),\boldsymbol{u}(k)} \quad \phi\left(\boldsymbol{x}(k), \boldsymbol{u}(k); x(k)\right) = \sum_i w_i \phi_i\left(\boldsymbol{x}_i(k), \boldsymbol{u}_i(k); x_i(k)\right)$$

subject to

$$x(l+1|k) = Ax(l|k) + Bu(l|k), \;\; k \leq l$$

$$u_i(l|k) \in \Omega_i, \;\; k \leq l, \forall \;\; i \in \mathbb{I}_M$$

$$x(k) = \widehat{x}(k)$$

in which $\boldsymbol{x}(k),\ \boldsymbol{u}(k)$ represents the centralized state and input trajectories. The cost function for subsystem i is ϕ_i. The system objective is a convex combination of the local objectives in

which $w_i > 0, i \in \mathbb{I}_M$, $\sum_i w_i = 1$. The vector $\widehat{x}(k)$ represents the current estimate of the centralized model states $x(k)$ at discrete time k.

$$\mathcal{P}_2(i): \quad \textbf{Decentralized MPC}$$

$$\min_{\boldsymbol{x}_{ii}(k), \boldsymbol{u}_i(k)} \quad \phi_i^d\left(\boldsymbol{x}_{ii}(k), \boldsymbol{u}_i(k); x_{ii}(k)\right)$$

subject to

$$x_{ii}(l+1|k) = A_{ii}x_{ii}(l|k) + B_{ii}u_i(l|k), \quad k \le l$$

$$u_i(l|k) \in \Omega_i, k \le l$$

$$x_{ii}(k) = \widehat{x}_{ii}(k)$$

in which $(\boldsymbol{x}_{ii}, \boldsymbol{u}_i)$ represents the decentralized state and input trajectories for subsystem $i \in \mathbb{I}_M$. The notation $\widehat{x}_{ii}(k)$ represents the estimate of the decentralized model states at discrete time k. The subsystem cost function $\phi_i^d\left(\boldsymbol{x}_{ii}(k), \boldsymbol{u}_i(k); x_{ii}(k)\right)$, in the decentralized MPC framework is defined as

$$\phi_i^d\left(\boldsymbol{x}_{ii}(k), \boldsymbol{u}_i(k); x_{ii}(k)\right) = \frac{1}{2}\sum_{t=k}^{\infty}\left[x_{ii}(t|k)'Q_{ii}x_{ii}(t|k) + u_i(t|k)'R_iu_i(t|k)\right]$$

in which $Q_{ii} \ge 0$, $R_i > 0$ and $(A_{ii}, Q_{ii}^{1/2})$ is detectable.

For communication and cooperation-based MPC, an iteration and exchange of variables between subsystems is performed during a sample time. We may choose not to iterate to convergence. We denote this iteration number as p. The cost function for communication-

based MPC is defined over an infinite horizon and written as

$$\phi_i\left(\boldsymbol{x}_i(k), \boldsymbol{u}_i(k); x_i(k)\right) = \sum_{t=k}^{\infty} L_i\left(x_i(t|k), u_i(t|k)\right) \tag{4.5}$$

in which $Q_i \geq 0$, $R_i > 0$ are symmetric weighting matrices with $(A_i, Q_i^{1/2})$ detectable. For each subsystem i and iterate p, the optimal state-input trajectory $(\boldsymbol{x}_i^p(k), \boldsymbol{u}_i^p(k))$ is obtained as the solution to the optimization problem $\mathcal{P}_3(i)$ defined as

$$\mathcal{P}_3(i): \quad \textbf{Communication-based MPC}$$

$$\min_{\boldsymbol{x}_i^p(k), \boldsymbol{u}_i^p(k)} \quad \phi_i\left(\boldsymbol{x}_i^p(k), \boldsymbol{u}_i^p(k); x_i(k)\right)$$

subject to

$$x_i^p(l+1|k) = A_i x_i^p(l|k) + B_i u_i^p(l|k) + \sum_{j \neq i} W_{ij} u_j^{p-1}(l|k), \quad k \leq l$$

$$u_i^p(l|k) \in \Omega_i, \quad k \leq l$$

$$x_i(k) = \widehat{x}_i(k)$$

in which $\boldsymbol{x}_i^p(k)' = [x_i^p(k+1|k)', x_i^p(k+2|k)', \ldots, \ldots]$, $\boldsymbol{u}_i^p(k)' = [u_i^p(k|k)', u_i^p(k+1|k)', \ldots, \ldots]$ and $\widehat{x}_i(k)$ represents the current estimate of the composite model states. Notice that the input sequence for subsystem i, $\boldsymbol{u}_i^p(k)$, is optimized to produce its value at iteration p, but the other subsystems' inputs are not updated during this optimization; they remain at iterate $p-1$. The objective function is the one for subsystem i only. For notational simplicity, we drop the time dependence of the state and input trajectories in each of the MPC frameworks described above. For instance, we write $(\boldsymbol{x}_i^p, \boldsymbol{u}_i^p) \equiv (\boldsymbol{x}_i^p(k), \boldsymbol{u}_i^p(k))$.

Each communication-based MPC [2] transmits current state and input trajectory information to all interconnected subsystems' MPCs. Competing agents have no knowledge of each others cost/utility functions. From a game theoretic perspective, the equilibrium of such a strategy, if it exists, is called a noncooperative equilibrium or Nash equilibrium Başar and Olsder (1999). The objectives of each subsystem's MPC controller are frequently in conflict with the objectives of other interacting subsystems' controllers. The best achievable performance is characterized by a Pareto optimal path which represents the set of optimal trade-offs among these conflicting/competing controller objectives. It is well known that the Nash equilibrium is usually suboptimal in the Pareto sense (Cohen (1998); Dubey and Rogawski (1990); Neck and Dockner (1987)).

4.3.1 Geometry of Communication-based MPC

We illustrate possible scenarios that can arise under communication-based MPC. In each case, $\Phi_i(\cdot)$ denotes the subsystem cost function obtained by eliminating the states from the cost function $\phi_i(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i)$ using the subsystem CM equation (see p. 39). The Nash equilibrium (NE) and the Pareto optimal solution are denoted by n and p, respectively. To allow a 2-dimensional representation, a unit control horizon $(N = 1)$ is used. In each example, existence of the NE follows using (Başar and Olsder, 1999, Theorem 4.4, p. 176). The NE n is the point of intersection of the reaction curves of the two cost functions (see (Başar and Olsder, 1999, p. 169)). The Pareto optimal path is the locus of (u_1, u_2) obtained by minimizing the weighted sum $w_1\Phi_1 + w_2\Phi_2$ for each $0 \leq w_1, w_2 \leq 1, w_1 + w_2 = 1$. If $(w_1, w_2) = (1, 0)$, the Pareto optimal solution is at point a, and if $(w_1, w_2) = (0, 1)$, the Pareto optimal solution is at point b.

[2] Similar strategies have been proposed in Camponogara et al. (2002); Jia and Krogh (2001)

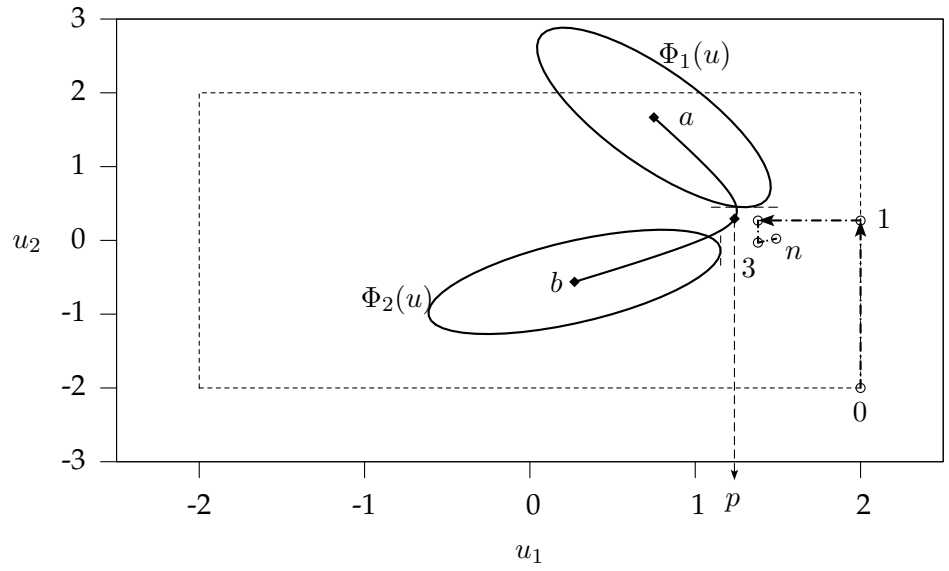

Figure 4.1: A stable Nash equilibrium exists and is near the Pareto optimal solution. Communication based iterates converge to the stable Nash equilibrium.

Example 1. Figure 4.1 illustrates the best case scenario for pure communication strategies. The NE n is located near the Pareto optimal solution p. For initial values of u_1 and u_2 located at point 0, the first communication-based iterate steers u_1 and u_2 to point 1. On iterating further, the sequence of communication-based iterates converges to n. In this case, the NE is stable *i.e.*, if the system is displaced from n, the sequence of communication-based iterates brings the system back to n. The closed-loop system will likely behave well in this case.

Example 2. Here, the initial values of the inputs are located near the Pareto optimal solution (Point 0 in Figure 4.2). However, as observed from Figure 4.2, the NE n for this system is not near p and therefore, the sequence of communication-based iterates drives the system away from the Pareto optimal solution. Even though the NE is stable, the solution obtained at convergence (n) of the communication-based strategy is far from optimal. Consequently, a stable NE need not imply closed-loop stability.

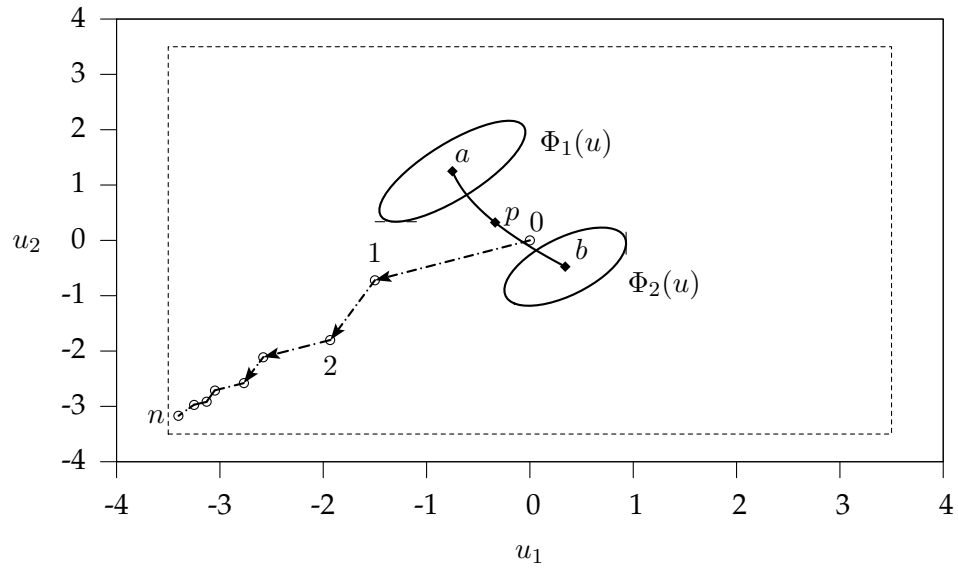

Figure 4.2: A stable Nash equilibrium exists but is not near the Pareto optimal solution. The converged solution, obtained using a communication-based strategy, is far from optimal.

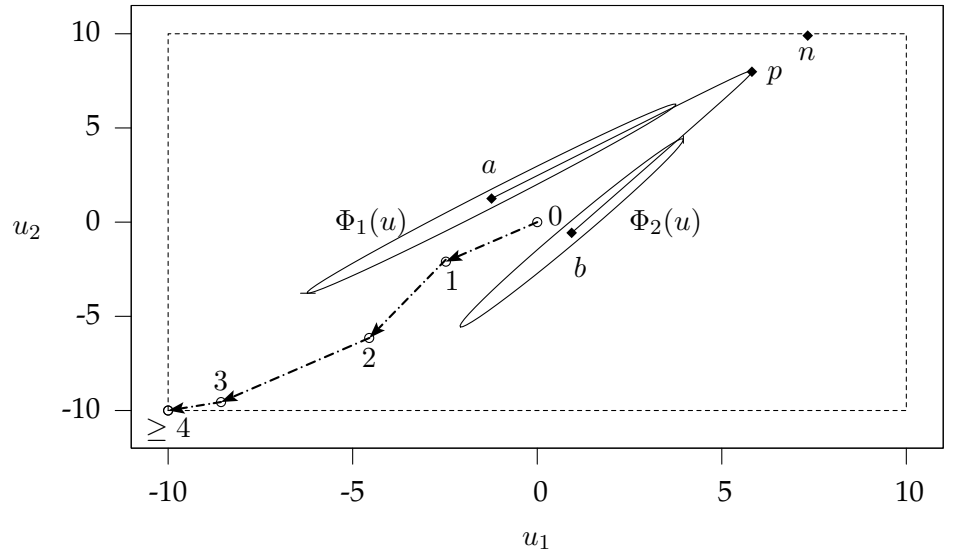

Figure 4.3: A stable Nash equilibrium does not exist. Communication-based iterates do not converge to the Nash equilibrium.

Example 3. We note from Figure 4.3 that the NE (n) for this system is in the proximity of the Pareto optimal solution p. For initial values of u_1 and u_2 at the origin and in the absence of

input constraints, the sequence of communication-based iterates diverges. For a compact feasible region (the box in Figure 4.3), the sequence of communication-based iterates is trapped at the boundary of the feasible region (Point 4) and does not converge to n. Here, a stable NE for a (pure) communication-based strategy, in the sense of (Başar and Olsder, 1999, Definition 4.5, p. 172), does not exist. The closed-loop system is likely to be unstable in this case.

For strongly coupled systems, the NE may not be close to the Pareto optimal solution. In some situations (Example 3), communication-based strategies do not converge to the NE. In fact, it is possible to construct simple examples where communication-based MPC leads to closed-loop instability (Section 4.7). Communication-based MPC is therefore, an unreliable strategy for systemwide control. The unreliability of the communication-based MPC formulation as a systemwide control strategy motivates the need for an alternate approach. We next modify the objective functions of the subsystems' controllers in order to provide a means for cooperation among the controllers. We replace the objective ϕ_i with an objective that measures the systemwide impact of local control actions. Many suitable objectives are possible. Here we choose the simplest case, the overall plant objective, which is a strict convex combination of the individual subsystems' objectives, $\phi = \sum_i w_i \phi_i, \;\; w_i > 0, \;\; \sum_{i=1}^{M} w_i = 1$.

In practical situations, the process sampling interval may be insufficient for the computation time required for convergence of a cooperation-based iterative algorithm. In such situations, the cooperation-based distributed MPC algorithm has to be terminated prior to convergence of the state and input trajectories (*i.e.*, when time runs out). The last calculated input trajectory is used to arrive at a suitable control law. To allow intermediate termination, all iterates generated by the distributed MPC algorithm must be plantwide feasible, and the resulting controller must be closed-loop stable. By plantwide feasibility, we mean that the state–

input sequence $\{\boldsymbol{x}_i, \boldsymbol{u}_i\}_{i=1}^{M}$ satisfies the model and input constraints of each subsystem. To guarantee plantwide feasibility of the intermediate iterates, we eliminate the states $\boldsymbol{x}_i, i \in \mathbb{I}_M$ from each of the optimization problems using the set of CM equations (Equation (4.1)). Subsequently, the cost function $\phi_i(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i(k))$ can be re-written as a function of all the interacting subsystem input trajectories with the initial subsystem state as a parameter *i.e.*,

$$\phi_i(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i(k)) \equiv \Phi_i\left(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_i, \ldots, \boldsymbol{u}_M; x_i(k)\right).$$

For each subsystem i, the optimal input trajectory $\boldsymbol{u}_i^{*(p)}$ is obtained as the solution to the feasible cooperation-based MPC (FC-MPC) optimization problem defined as

$$\mathcal{P}_4(i): \quad \textbf{Feasible cooperation-based MPC}$$

$$\boldsymbol{u}_i^{*(p)}(k) \in \ \arg(\mathcal{F}_i) \ \text{ where}$$

$$\mathcal{F}_i \triangleq \min_{\boldsymbol{u}_i} \quad \sum_{r=1}^{M} w_r \Phi_r \left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x_r(k)\right)$$

subject to

$$u_i(l|k) \in \Omega_i, \ \ k \leq l$$

$$x_r(k) = \widehat{x}_r(k), \forall\, r \in \mathbb{I}_M$$

4.4 Distributed, constrained optimization

Consider the following centralized MPC optimization problem.

$$\min_{\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_M} \quad \Phi\left(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_M;\mu(k)\right)=\sum_{i=1}^{M} w_i\Phi_i\left(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_M;x_i(k)\right) \tag{4.6a}$$

subject to

$$u_i(l|k)\in\Omega_i,\ \ k\le l\le k+N-1, \tag{4.6b}$$

$$u_i(l|k)=0,\ \ k+N\le l, \tag{4.6c}$$

$$x_i(k)=\widehat{x}_i(k),\quad \forall\, i\in\ \mathbb{I}_M$$

For open-loop integrating/unstable systems, an additional terminal state constraint that forces the unstable modes to the origin at the end of the control horizon is necessary to ensure stability (Rawlings and Muske (1993)).

Definition 4.1. The *normal cone* to a convex set Ω at a point $x\in\Omega$ is denoted by $N(x,\Omega)$ and defined by $N(x;\Omega)=\{s\,|\,\langle s,y-x\rangle\le 0\ \text{ for all } y\in\Omega\}$.

Let $(\boldsymbol{u}_1^*,\boldsymbol{u}_2^*,\ldots,\boldsymbol{u}_M^*)$ denote the solution to the centralized optimization problem of Equation (4.6). By definition, $\boldsymbol{u}_i^{*\prime}=[\overline{\boldsymbol{u}}_i^{*\prime},0,0,\ldots]\,,\ \forall\ i\in\mathbb{I}_M$. For each subsystem $i\in\mathbb{I}_M$, define $\mathcal{U}_i\in\mathbb{R}^{m_iN}$ as $\mathcal{U}_i=\Omega_i\times\Omega_i\times\ldots\times\Omega_i$. Hence, $\overline{\boldsymbol{u}}_i^*\in\mathcal{U}_i,\ \forall\, i\in\mathbb{I}_M$. The results presented here are valid also for $\Phi(\cdot)=\sum_{i=1}^{M}w_i\Phi_i(\cdot)$ convex and differentiable on some open neighborhood of $\mathcal{U}_1\times\mathcal{U}_2\times\ldots\times\mathcal{U}_M$ [3]. Optimality is characterized by the following result (which uses convexity but does not assume that the solution is unique).

[3]The assumptions on $\Phi(\cdot)$ imply that $\Phi\left(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_M\,;\mu(k)\right)\ >\ -\infty$ for all $\text{vec}(u_1(j|k),u_2(j|k),\ldots,u_M(j|k))\ \in\ \Omega_1\times\Omega_2\times\ldots\times\Omega_M,\forall\ j\ \ge\ k$ and that $\Phi(\cdot)$ is a proper convex function in the sense of (Rockafellar, 1970, p. 24).

Lemma 4.2. *$(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*)$ is optimal for the optimization problem of* Equation (4.6) *if and only if*

$$-\nabla_{\overline{\boldsymbol{u}}_i} \Phi\left(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*\right]; \mu(k)) \in N(\overline{\boldsymbol{u}}_i^*; \mathcal{U}_i), \;\; \textit{for all } i \in \mathbb{I}_M.$$

Proof. By definition (Equation (4.6)), $\boldsymbol{u}_i{}' = [\overline{\boldsymbol{u}}_i{}', 0, 0, \ldots], \; \forall \; i \in \mathbb{I}_M$. We note that $\Phi(\cdot)$ is a proper convex function, that $\mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \times \mathcal{U}_M \subset \text{dom}\,(\Phi(\cdot))$, and that the relative interior of $\mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \times \mathcal{U}_M$ is nonempty (see (Rockafellar, 1970, Theorem 6.2, p. 45)). Hence, the result is a consequence of (Rockafellar, 1970, Theorem 27.4, p. 270). □

Suppose that the following level set is bounded and closed (hence compact):

$$\mathcal{L} = \left\{ (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M) \;\middle|\; \Phi\left(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M \,; \mu(k)\right) \leq \Phi\left(\boldsymbol{u}_1^0, \boldsymbol{u}_2^0, \ldots, \boldsymbol{u}_M^0 \,; \mu(k)\right), \right.$$
$$\left. u_i(k+j|k) \in \Omega_i, \; 0 \leq j \leq N-1, \; u_i(k+j|k) = 0, \; N \leq j, \; i \in \mathbb{I}_M \right\} \tag{4.7}$$

We have the following result concerning the limiting set of a sequence of normal cones of a closed convex set.

Lemma 4.3. *Let $\Omega \in \mathbb{R}^n$ be closed and convex. Let $x \in \Omega$ and let $\{x_i\}$ be a sequence of points satisfying $x_i \in \Omega$ and $x_i \to x$. Let $\{v_i\}$ be any sequence satisfying $v_i \in N(x_i; \Omega)$ for all i. Then all limit points of the sequence $\{v_i\}$ belong to $N(x; \Omega)$.*

Proof. Let v be a limit point of $\{v_i\}$ and let $\mathcal{S}$ be a subsequence such that $\lim_{i \in \mathcal{S}} v_i = v$. By definition of normal cone, we have $\langle v_i, y - x_i \rangle \leq 0$ for all $y \in \Omega$ and all $i \in \mathcal{S}$. By taking limits as $i \to \infty$, $i \in \mathcal{S}$, we have $\langle v, y - x \rangle \leq 0$ for all $y \in \Omega$, proving that $v \in N(x, \Omega)$, as claimed. □

4.5 Feasible cooperation-based MPC (FC-MPC)

The FC-MPC optimization problem for subsystem $i \in \mathbb{I}_M$ is defined as

$$\mathcal{F}_i \triangleq \min_{\boldsymbol{u}_i} \quad \sum_{r=1}^{M} w_r \Phi_r \left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x_r(k) \right) \tag{4.8a}$$

subject to

$$u_i(l|k) \in \Omega_i, \;\; k \leq l \leq k+N-1 \tag{4.8b}$$

$$u_i(l|k) = 0, \;\; k+N \leq l \tag{4.8c}$$

in which $x_i(k) = \widehat{x}_i(k), \forall\, i \in \mathbb{I}_M$. For $\Phi_i(\cdot)$ quadratic and obtained by eliminating the CM states x_i from Equation (4.5) using the subsystem CM (Equation (4.1)), $\forall\, i \in \mathbb{I}_M$, the FC-MPC optimization problem (Equation (4.8)) can be rewritten as

$$\mathcal{F}_i \quad \triangleq \quad \min_{\overline{\boldsymbol{u}}_i} \quad \frac{1}{2}\overline{\boldsymbol{u}}_i(k)'\mathcal{R}_i\overline{\boldsymbol{u}}_i(k) \;+\; \left(r_i(k) + \sum_{j=1, j\neq i}^{M} \mathcal{H}_{ij}\overline{\boldsymbol{u}}_j^{p-1}(k) \right)'\overline{\boldsymbol{u}}_i(k) \;+\; \text{constant} \tag{4.9a}$$

subject to

$$u_i(l|k) \in \Omega_i, \;\; k \leq l \leq k+N-1 \tag{4.9b}$$

in which $\mathbb{Q}_i = \text{diag}(Q_i(1), \ldots, Q_i(N-1), \overline{Q}_i)$, $\mathbb{R}_i = \text{diag}(R_i(0), R_i(1), \ldots, R_i(N-1))$,

$$\mathcal{R}_i = w_i\mathbb{R}_i + w_i E_{ii}'\mathbb{Q}_i E_{ii} + \sum_{j\neq i}^{M} w_j E_{ji}'\mathbb{Q}_j E_{ji}, \qquad \mathcal{H}_{ij} = \sum_{l=1}^{M} w_l E_{li}'\mathbb{Q}_l E_{lj},$$

$$r_i(k) = w_i E_{ii}'\mathbb{Q}_i f_i x_i(k) + \sum_{j\neq i}^{M} w_j E_{ji}'\mathbb{Q}_j f_j x_j(k),$$

$$E_{ii} = \begin{bmatrix} B_i & 0 & \dots & \dots & 0 \\ A_iB_i & B_i & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_i^{N-1}B_i & \dots & \dots & \dots & B_i \end{bmatrix}, \; E_{ij} = \begin{bmatrix} W_{ij} & 0 & \dots & \dots & 0 \\ A_iW_{ij} & W_{ij} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_i^{N-1}W_{ij} & \dots & \dots & \dots & W_{ij} \end{bmatrix}, \; f_i = \begin{bmatrix} A_i \\ A_i^2 \\ \vdots \\ \vdots \\ A_i^N \end{bmatrix}$$

with $\overline{Q}_i$ denoting an appropriately chosen terminal penalty, as described in the sequel.

An implementable algorithm for FC-MPC is now described. Convergence and optimality properties of the proposed FC-MPC algorithm are established subsequently. Some additional notation is required. Let $\Phi(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \dots, \boldsymbol{u}_M^p; \mu(k))$ represent the cooperation-based cost function at iterate p and discrete time k with the initial set of subsystem states $\mu(k)$. Let $\overline{\boldsymbol{u}}_i^{*(p)}, \forall\, i \in \mathbb{I}_M$ denote the solution to the FC-MPC optimization problem [4] (Equation (4.9)). By definition, the corresponding infinite horizon input trajectory $\boldsymbol{u}_i^{*(p)} = [\overline{\boldsymbol{u}}_i^{*(p)\prime}, 0, 0, \dots], \forall\, i \in \mathbb{I}_M$. The cost associated with the input trajectory $\boldsymbol{u}_i^{*(p)}$ constructed from the solution to $\mathcal{F}_i$ (Equation (4.9)), is represented as $\Phi(\boldsymbol{u}_1^{p-1}, \dots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i^{*(p)}, \boldsymbol{u}_{i+1}^{p-1}, \dots, \boldsymbol{u}_M^{p-1}; \mu(k))$. The state sequence for subsystem i generated by the subsystems' input trajectories $(\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_M)$ and initial set of subsystem states μ is represented as $\boldsymbol{x}_i(\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_M; \mu)$. For notational convenience, we write $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i(\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_M; \mu)$ and $\overline{\boldsymbol{x}}_i \leftarrow \overline{\boldsymbol{x}}_i(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \dots, \overline{\boldsymbol{u}}_M; \mu)$. At discrete time k, let $p_{\max}(k)$ denote the maximum number of permissible iterates during the sampling interval. By definition, $0 < p(k) \leq p_{\max}(k), \forall\, k \geq 0$.

Algorithm 4.1. Given $\overline{\boldsymbol{u}}_i^0, x_i(k), \mathbb{Q}_i \geq 0, \mathbb{R}_i > 0, \; \forall\, i \in \mathbb{I}_M, p_{\max}(k) > 0$ and $\epsilon > 0$

$p \leftarrow 1, \kappa_i \leftarrow \Gamma\epsilon, \Gamma \gg 1$

[4] For notational simplicity, we drop the functional dependence of $\boldsymbol{u}_i^{*(p)}$ on $\mu(k)$ and $\overline{\boldsymbol{u}}_j^{p-1}, j \neq i$.

while $\kappa_i > \epsilon$ for some $i \in \mathbb{I}_M$ and $p \leq p_{\max}(k)$

do $\forall\ i \in \mathbb{I}_M$

$\overline{\boldsymbol{u}}_i^{*(p)} \in \ \arg(\mathcal{F}_i)$ (see Equation (4.9))

$\overline{\boldsymbol{u}}_i^{p} \leftarrow w_i \overline{\boldsymbol{u}}_i^{*(p)} + (1 - w_i)\, \overline{\boldsymbol{u}}_i^{p-1}$

Transmit $\overline{\boldsymbol{u}}_i^{p}$ to each interconnected subsystem $j \neq i$

$\kappa_i \leftarrow \|\overline{\boldsymbol{u}}_i^{p} - \overline{\boldsymbol{u}}_i^{p-1}\|$

end (do)

$p \leftarrow p + 1$

end (while)

In Algorithm 4.1 above, the state trajectory for subsystem i at iterate p is obtained as $\overline{\boldsymbol{x}}_i^{p} \leftarrow \overline{\boldsymbol{x}}_i^{p}\left(\overline{\boldsymbol{u}}_1^{p}, \overline{\boldsymbol{u}}_2^{p}, \ldots, \overline{\boldsymbol{u}}_M^{p}; \mu(k)\right)$. The maximum allowable iterates in each sampling interval $p_{\max}(k)$ is a design limit; one may choose to terminate Algorithm 4.1 prior to this limit.

Assumption 4.2. $p \in \mathbb{I}_+, 0 < p_{\max}(k) \leq p^* < \infty$.

Assumption 4.3. $N \geq \max(\alpha, 1)$, in which $\alpha = \max_{i \in \mathbb{I}_M} \alpha_i$ and $\alpha_i \geq 0$ denotes the number of unstable modes for subsystem $i \in \mathbb{I}_M$.

Lemma 4.4. *Consider the FC-MPC formulation of Equations* (4.8), (4.9). *The sequence of cost functions* $\left\{\Phi(\boldsymbol{u}_1^{p}, \boldsymbol{u}_2^{p}, \ldots, \boldsymbol{u}_M^{p}; \mu(k))\right\}$ *generated by Algorithm 4.1 is a nonincreasing function of the iteration number* p.

Proof. From Algorithm 4.1 we know, for all $i \in \mathbb{I}_M$ that

$$\Phi\left(\boldsymbol{u}_1^{p-1}, \ldots \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i^{*(p)}, \boldsymbol{u}_{i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)\right) \leq \Phi\left(\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)\right)$$

Therefore, from the definition of $\boldsymbol{u}_i^p$ (Algorithm 4.1) we have

$$\begin{aligned}
\Phi\left(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; \mu(k)\right) &= \Phi\left(w_1\boldsymbol{u}_1^{*(p)} + (1-w_1)\boldsymbol{u}_1^{p-1}, \ldots, w_M\boldsymbol{u}_M^{*(p)} + (1-w_M)\boldsymbol{u}_M^{p-1}; \mu(k)\right) \\
&= \Phi\Big(w_1(\boldsymbol{u}_1^{*(p)}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}) + w_2(\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{*(p)}, \ldots, \boldsymbol{u}_M^{p-1}) + \ldots \\
&\quad \ldots + w_M(\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{*(p)}); \mu(k)\Big) \\
\text{(by convexity of } \Phi(\cdot)) \quad &\leq \sum_{r=1}^{M} w_r \Phi\left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{r-1}^{p-1}, \boldsymbol{u}_r^{*(p)}, \boldsymbol{u}_{r+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)\right) \\
&\leq \Phi\left(\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)\right) \qquad (4.10)
\end{aligned}$$

□

Lemma 4.5. *All limit points of Algorithm 4.1 are optimal*

Proof. Let $(\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_M)$ be a limit point and $\mathcal{S}$ be a subsequence for which

$$\lim_{p \in \mathcal{S}} (\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}) = (\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_M).$$

By definition, $\boldsymbol{t}_i{}' = [\bar{\boldsymbol{t}}_i{}', 0, 0, \ldots], \ \forall\, i \in \mathbb{I}_M$, where $\bar{\boldsymbol{t}}_i \in \mathcal{U}_i$. By taking a further subsequence of $\mathcal{S}$ if necessary and using compactness of the level set $\mathcal{L}$ (Equation (4.7)), we can define $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M)$ such that

$$\lim_{p \in \mathcal{S}} (\boldsymbol{u}_1^{p}, \boldsymbol{u}_2^{p}, \ldots, \boldsymbol{u}_M^{p}) = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M).$$

We have, $\boldsymbol{v}_i{}' = [\overline{\boldsymbol{v}}_i{}', 0, 0, \ldots]$, $\forall\, i \in \mathbb{I}_M$, where $\overline{\boldsymbol{v}}_i \in \mathcal{U}_i$. From Equation (4.10) (Lemma 4.4), by taking limits and noting that $\Phi\left(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p\,; \mu(k)\right)$ is bounded below, we have that

$$\Phi\left(\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_M\,; \mu(k)\right) = \Phi\left(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\,; \mu(k)\right)$$

From Algorithm 4.1, we have $\boldsymbol{u}_i^{*(p)} = \frac{1}{w_i}\boldsymbol{u}_i^p - \left(\frac{1}{w_i} - 1\right)\boldsymbol{u}_i^{p-1},\ i \in \mathbb{I}_M$. Using the definitions of $\boldsymbol{v}_i$ and $\boldsymbol{t}_i$, define $\boldsymbol{z}_i = \frac{1}{w_i}\boldsymbol{v}_i - \left(\frac{1}{w_i} - 1\right)\boldsymbol{t}_i,\ \forall\, i \in\ \mathbb{I}_M$, we have by taking limits in Equation (4.10) (Lemma 4.4) and using, $\forall\, i \in \mathbb{I}_M$,

$$\Phi\left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i^{*(p)}, \boldsymbol{u}_{i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)\right) \leq \Phi\left(\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)\right)$$

that in fact

$$\begin{aligned}\Phi\left(\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_M\,; \mu(k)\right) &= \Phi\left(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{i-1}, \boldsymbol{z}_i, \boldsymbol{t}_{i+1}, \ldots, \boldsymbol{t}_M\,; \mu(k)\right) \\ &= \Phi\left(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\,; \mu(k)\right),\ \forall\, i \in \mathbb{I}_M \end{aligned} \tag{4.11}$$

From Algorithm 4.1, we have for each i,

$$-\nabla_{\overline{\boldsymbol{u}}_i}\Phi\left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i^{*(p)}, \boldsymbol{u}_{i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}\,; \mu(k)\right) \in N(\overline{\boldsymbol{u}}_i^{*(p)}, \mathcal{U}_i).$$

Taking limits and invoking Lemma 4.3, we see that

$$-\nabla_{\overline{\boldsymbol{u}}_i}\Phi\left(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{i-1}, \boldsymbol{z}_i, \boldsymbol{t}_{i+1}, \ldots, \boldsymbol{t}_M\,; \mu(k)\right) \in N(\overline{\boldsymbol{z}}_i, \mathcal{U}_i),\ \forall\, i \in \mathbb{I}_M$$

in which $\boldsymbol{z}_i{}' = [\overline{\boldsymbol{z}}_i{}', 0, 0, \ldots]$. From Equation (4.11), we know

$$\Phi\left(\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_M\,; \mu(k)\right) = \Phi\left(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{i-1}, \boldsymbol{z}_i, \boldsymbol{t}_{i+1}, \ldots, \boldsymbol{t}_M\,; \mu(k)\right).$$

Therefore, $\boldsymbol{t}_i$ is also a minimizer of $\Phi\left(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{i-1}, \boldsymbol{z}_i, \boldsymbol{t}_{i+1}, \ldots, \boldsymbol{t}_M\,; \mu(k)\right)$ over $\mathcal{U}_i$ and so, we have $-\nabla_{\overline{\boldsymbol{u}}_i}\Phi\left(\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_M\,; \mu(k)\right) \in N(\overline{\boldsymbol{t}}_i, \mathcal{U}_i),\ \forall\ i \in \mathbb{I}_M$. Hence, $(\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_M)$ satisfies the optimality condition. □

For the iterates confined to the level set $\mathcal{L}$ (Equation (4.7)), a limit point is guaranteed to exist. From Lemma 4.5, all limit points are optimal. Using strict convexity of the objective in Equation (4.9), it follows that $(\boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_M^*)$ is in fact the limit.

4.6 Closed-loop properties of FC-MPC under state feedback

At time k, let the FC-MPC scheme be terminated after $p(k) = q$ iterates. Let

$$\boldsymbol{u}_i^q(\mu(k))' = \left[u_i^q(\mu(k), 0)', u_i^q(\mu(k), 1)', \ldots, u_i^q(\mu(k), N-1)',\ 0,\ 0, \ldots\right]$$

for each $i \in \mathbb{I}_M$ represent the solution to the FC-MPC algorithm (Algorithm 4.1) after q iterates. The distributed MPC control law is obtained through a receding horizon implementation of optimal control whereby the input applied to subsystem i is $u_i(k) = u_i^q(\mu(k), 0)$.

Assumption 4.4. For each $i \in \mathbb{I}_M$, (A_{ii}, B_{ii}) is stabilizable.

Assumption 4.5. $Q_i(0) = Q_i(1) = \ldots = Q_i(N-1) = Q_i > 0$ and $R_i(0) = R_i(1) = \cdots = R_i(N-1) = R_i > 0,\ \forall\, i \in \mathbb{I}_M$.

Lemmas 4.4 and 4.5 lead to the following results on closed-loop stability.

4.6.1 Nominal stability for systems with stable decentralized modes

Feasibility of FC-MPC optimizations and domain of attraction. For open-loop stable systems, the domain of the controller is $\mathbb{R}^n$, $n = \sum_{i=1}^{M} n_i$. Convexity of each of the admissible input sets $\Omega_i,\ i \in \mathbb{I}_M$ and Algorithm 4.1 guarantee that if a feasible input trajectory exists for each subsystem $i \in \mathbb{I}_M$ at time $k = 0$ and $p(0) = 0$, then a feasible input trajectory exists for all subsystems at all future times. One trivial choice for a feasible input trajectory at $k = 0$ is $u_i(k + l|k) = 0,\ \ l \geq 0,\ \forall\, i \in \mathbb{I}_M$. This choice follows from our assumption that Ω is nonempty and $0 \in \text{int}(\Omega)$. The domain of attraction for the closed-loop system is $\mathbb{R}^n$.

Initialization. At time $k = 0$, let $\boldsymbol{u}_i^0(0) = [0,\ 0, \ldots\ldots]',\ \forall\, i \in \mathbb{I}_M$. Since $0 \in \text{int}(\Omega)$, this sequence of inputs is feasible. Define $\widetilde{J}_N(\mu(0)) = \Phi(\boldsymbol{u}_1^0(0), \ldots, \boldsymbol{u}_M^0(0); \mu(0))$ to be the value of the cooperation-based cost function with the set of zero input initialization trajectories and the set of initial subsystem states $\mu(0)$. At time $k > 0$, define $\forall\, i \in \mathbb{I}_M$

$$\boldsymbol{u}_i^0(k)' = \left[u_i^{p(k-1)}(\mu(k-1), 1)', \ldots, u_i^{p(k-1)}(\mu(k-1), N-1)',\ 0,\ 0, \ldots\right] \tag{4.12}$$

$(\boldsymbol{u}_1^0(k), \boldsymbol{u}_2^0(k), \ldots, \boldsymbol{u}_M^0(k))$ constitutes a set of feasible subsystem input trajectories with an associated cost function $J_N^0(\mu(k)) = \Phi\left(\boldsymbol{u}_1^0(k), \boldsymbol{u}_2^0(k), \ldots, \boldsymbol{u}_M^0(k); \mu(k)\right)$. The value of the cooperation based cost function after $p(k)$ iterates is denoted by

$$J_N^{p(k)}(\mu(k)) = \Phi(\boldsymbol{u}_1^{p(k)}(k), \ldots, \boldsymbol{u}_M^{p(k)}(k); \mu(k)).$$

The following lemma establishes the relationship between the different cost function values.

Lemma 4.6. *Given Algorithm 4.1, employing the FC-MPC optimization problem of Equation (4.9), for a system with stable decentralized modes. At time $k = 0$, let Algorithm 4.1 be initialized with input $u_i(k+l|k) = 0,\ l \geq 0,\ \forall\ i \in \mathbb{I}_M$. If for all times $k > 0$, each FC-MPC optimization problem is initialized with the strategy described in Equation (4.12), then we have,*

$$J_N^{p(k)}(\mu(k)) \leq J_N^0(\mu(k)) \leq \widetilde{J}_N(\mu(0)) - \sum_{j=0}^{k-1}\sum_{i=1}^{M} w_i L_i(x_i(j), 0) \leq \widetilde{J}_N(\mu(0)) \tag{4.13}$$

$$\forall\ p(k) \geq 0 \textit{ and all } k \geq 0$$

A proof is available in Appendix 4.10.2.

Lemma 4.6 can be used to show stability in the sense of Lyapunov (Vidyasagar, 1993, p. 136). Attractivity of the origin follows from the cost relationship $0 \leq J_N^{p(k+1)}(\mu(k+1)) \leq J_N^0(\mu(k)) = J_N^{p(k)}(\mu(k)) - \sum_{i=1}^{M} w_i L_i(x_i(k), u_i^{p(k)}(k))$. Asymptotic stability, therefore, follows.

Assumption 4.6. For each $i \in \mathbb{I}_M$, A_{ii} is stable, $\mathbb{Q}_i = \text{diag}\left(Q_i(1), \ldots, Q_i(N-1), \overline{Q}_i\right)$, in which $\overline{Q}_i$ is the solution of the Lyapunov equation $A_i{}'\overline{Q}_i A_i - \overline{Q}_i = -Q_i$

Remark 4.1. Consider a ball $B_\varepsilon(0), \varepsilon > 0$ such that the input constraints in each FC-MPC optimization problem are inactive. Because $0 \in \text{int}(\Omega_1 \times \cdots \times \Omega_M)$ and the distributed MPC control law is stable and attractive, an $\varepsilon > 0$ exists. Let Assumption 4.6 hold. For $\mu \in B_\varepsilon(0)$, $\overline{\boldsymbol{u}}_i^p(\mu), i \in \mathbb{I}_M$ is linear in $x_i, i \in \mathbb{I}_M$. Also, the initialization strategy for Algorithm 4.1 is independent of μ. The input trajectory $\overline{\boldsymbol{u}}_i^p(\mu),\ i \in \mathbb{I}_M$ generated by Algorithm 4.1 is, therefore, a Lipschitz continuous function of μ for all $p \in \mathbb{I}_+$. If $p \leq p^* < \infty$ (Assumption 4.2), a global Lipschitz constant (independent of p) can be estimated.

A stronger, exponential stability result is established using the following theorem.

Theorem 4.1 (Stable decentralized modes)**.** *Consider Algorithm 4.1 under state feedback employing the FC-MPC optimization problem of Equation* (4.9)*, $\forall\, i \in \mathbb{I}_M$. Let Assumptions 4.1 to 4.6 be satisfied. The origin is an exponentially stable equilibrium for the nominal closed-loop system*

$$x_i(k+1) = A_i x_i(k) + B_i u_i^{p(k)}(\mu(k), 0) + \sum_{j \neq i}^{M} W_{ij} u_j^{p(k)}(\mu(k), 0),\ i \in \mathbb{I}_M,$$

for all $\mu(0) \in \mathbb{R}^n$ and all $p(k) = 1, 2, \ldots, p_{\max}(k)$.

A proof is available in Appendix 4.10.4.

4.6.2 Nominal stability for systems with unstable decentralized modes

From Assumption 4.1, unstable modes may be present only in the decentralized model. For systems with some decentralized model eigenvalues on or outside the unit circle, closed-loop stability under state feedback can be achieved using a terminal state constraint that forces the unstable decentralized modes to zero at the end of the control horizon. Define

$$\mathbb{S}_i = \{x_{ii} \,|\, \exists\, \overline{\boldsymbol{u}}_i \in \mathcal{U}_i \text{ such that } {S_{u_i}}'[\mathcal{C}_N(A_{ii}, B_{ii})\, \overline{\boldsymbol{u}}_i + A_{ii}^N x_{ii}] = 0\} \qquad \text{steerable set}$$

to be the set of decentralized states x_{ii} that can be steered to the origin in N moves. From Assumption 4.1 and because the domain of each $x_{ij}, i, j \in \mathbb{I}_M, j \neq i$ is $\mathbb{R}^{n_{ij}}$, we define

$$\mathbb{D}_{R_i} = \mathbb{R}^{n_{i1}} \times \cdots \times \mathbb{R}^{n_{i(i-1)}} \times \mathbb{S}_i \times \mathbb{R}^{n_{i(i+1)}} \times \cdots \times \mathbb{R}^{n_{iM}} \subseteq \mathbb{R}^{n_i},\ i \in \mathbb{I}_M, \qquad \text{domain of regulator}$$

to be the of all x_i for which an admissible input trajectory $\overline{\boldsymbol{u}}_i$ exists that drives the unstable decentralized modes ${U_{u_i}}' x_i$ to zero (in N moves). The domain of the controller for the nominal

closed-loop system

$$x_i^+ = A_i x_i + B_i u_i^p(\mu, 0) + \sum_{j \neq i}^{M} u_j^p(\mu, 0), \quad i \in \mathbb{I}_M$$

is given by

$$\mathbb{D}_C = \{\mu \,|x_i \in \mathbb{D}_{R_i},\, i \in \mathbb{I}_M\}. \qquad \text{domain of controller}$$

The set $\mathbb{D}_C$ is positively invariant for the nominal system.

Initialization. Since $\mu(0) \in \mathbb{D}_C$, a feasible input trajectory exists and can be computed by solving the following simple quadratic program (QP) for each $i \in \mathbb{I}_M$.

$$\overline{\boldsymbol{u}}_i^0 = \arg\min_{\overline{\boldsymbol{u}}_i} \quad \|\overline{\boldsymbol{u}}_i\|^2 \tag{4.14a}$$

subject to

$${U_{u_i}}' \left(\mathcal{C}_N(A_i, B_i)\, \overline{\boldsymbol{u}}_i + A_i^N x_i(0) \right) = 0 \tag{4.14b}$$

$$u_i(l|0) \in \Omega_i, \;\; 0 \leq l \leq N-1 \tag{4.14c}$$

in which U_{u_i} is obtained through a Schur decomposition (Golub and Van Loan, 1996, p. 341) of A_i [5]. Since unstable modes, if any, are present only in the decentralized model (Assumption 4.1), we have ${U_{u_i}}' \left(\mathcal{C}_N(A_i, B_i)\, \overline{\boldsymbol{u}}_i + A_i^N x_i(0) \right) = {S_{u_i}}' \left(\mathcal{C}_N(A_{ii}, B_{ii})\, \overline{\boldsymbol{u}}_i + A_{ii}^N x_{ii}(0) \right)$.

[5] The Schur decomposition of $A_i = \begin{bmatrix} U_{s_i} & U_{u_i} \end{bmatrix} \begin{bmatrix} A_{s_i} & \\ & \bigotimes A_{u_i} \end{bmatrix} \begin{bmatrix} {U_{s_i}}' \\ {U_{u_i}}' \end{bmatrix}$ and $A_{ii} = \begin{bmatrix} S_{s_i} & S_{u_i} \end{bmatrix} \begin{bmatrix} A_{s_{ii}} & \\ & \bigoplus A_{u_{ii}} \end{bmatrix} \begin{bmatrix} {S_{s_i}}' \\ {S_{u_i}}' \end{bmatrix}$. The eigenvalues of $A_{s_i}, A_{s_{ii}}$ are strictly inside the unit circle and the eigenvalues of $A_{u_i}, A_{u_{ii}}$ are on or outside the unit circle.

Feasibility of FC-MPC optimizations and domain of attraction. In the nominal case, the initialization QP (Equation (4.14)) needs to be solved only once for each subsystem *i.e.*, at time $k = 0$. Nominal feasibility is assured for all $k \geq 0$ and $p(k) > 0$ if the initialization QP at $k = 0$ is feasible for each $i \in \mathbb{I}_M$. At time $k + 1$, the the initial input trajectory is given by Equation (4.12) for all $i \in \mathbb{I}_M$. The domain of attraction for the closed-loop system is the set $\mathbb{D}_C$.

Consider subsystem i with $\alpha_i \geq 0$ unstable modes. Since all interaction models are stable, all unstable modes arise from the decentralized model matrix A_{ii}. To have a bounded objective, the predicted control trajectory for subsystem i at iterate $p(k)$, $\overline{\boldsymbol{u}}_i^{p(k)}$, must bring the unstable decentralized modes ${U_{u_i}}' x_i$ to the origin at the end of the control horizon. Boundedness of the infinite horizon objective can be ensured by adding an end constraint

$$ {U_{u_i}}' \left(\mathcal{C}_N(A_i, B_i)\, \overline{\boldsymbol{u}}_i + A_i^N x_i(k) \right) = 0 $$

to the FC-MPC optimization problem (Equations (4.8), (4.9)) within the framework of Algorithm 4.1. Feasibility of the above end constraint follows because $N \geq \alpha_i$ and (A_i, B_i) is stabilizable [6].

At time $k = 0$, let the FC-MPC formulation be initialized with the feasible input trajectory $\boldsymbol{u}_i^0(0) = [v_i(0)', v_i(1)', \ldots]'$ obtained as the solution to Equation (4.14), in which $v_i(s) = 0,\ N \leq s,\ \forall\ i \in \mathbb{I}_M$, and let $\overline{J}_N(\mu(0)) = \Phi(\boldsymbol{u}_1^0(0), \ldots, \boldsymbol{u}_M^0(0); \mu(0))$ denote the associated cost function value. We have $J_N^{p(k)}(\mu(k)) \leq J_N^0(\mu(k)) \leq \overline{J}_N(\mu(0)) - \sum_{j=0}^{k-1} w_i L_i(x_i(k), 0) \leq \overline{J}_N(\mu(0)),\ \forall\ k \geq 0, p(k) > 0$. The proof for the above claim is identical to the proof for

[6] Stabilizability of (A_i, B_i) follows from Assumptions 4.1 and 4.4.

Lemma 4.6. Asymptotic stability can be established using arguments identical to those outlined in Section 4.6.1.

Assumption 4.7. $\alpha > 0$ (see Assumption 4.3). For each $i \in \mathbb{I}_M$,

$$\mathbb{Q}_i = \text{diag}\left(Q_i(1), \ldots, Q_i(N-1), \overline{Q}_i\right),$$

in which $\overline{Q}_i = U_{s_i}\Sigma_i {U_{s_i}}'$ with Σ_i obtained as the solution of the Lyapunov equation ${A_{s_i}}'\Sigma_i A_{s_i} - \Sigma_i = -{U_{s_i}}' Q_i U_{s_i}$.

The following theorem establishes exponential stability for systems with unstable decentralized modes.

Theorem 4.2 (Unstable decentralized modes)**.** *Consider Algorithm 4.1 under state feedback, employing the FC-MPC optimization problem of Equation* (4.9)*, $\forall\ i \in \mathbb{I}_M$, with an additional terminal constraint ${U_{u_i}}' x_i(k+N|k) = {U_{u_i}}' \left(\mathcal{C}_N(A_i, B_i)\,\overline{\boldsymbol{u}}_i + A_i^N x_i(k)\right) = 0$ enforced on the unstable decentralized modes. Let Assumptions 4.1 to 4.5 and Assumption 4.7 hold. The origin is an exponentially stable equilibrium point for the nominal closed-loop system*

$$x_i(k+1) = A_i x_i(k) + B_i u_i^{p(k)}(\mu(k), 0) + \sum_{j \neq i}^{M} W_{ij} u_j^{p(k)}(\mu(k), 0),\ i \in \mathbb{I}_M,$$

for all $\mu(0) \in \mathbb{D}_C$ and all $p(k) = 1, 2, \ldots, p_{\max}(k)$.

For positive semidefinite penalties on the states $x_i, i \in \mathbb{I}_M$, we have the following results:

Remark 4.2. Let Assumption 4.1 hold and let $Q_i \geq 0,\ (A_i, Q_i^{1/2}),\ i \in \mathbb{I}_M$ be detectable. The nominal closed-loop system $x_i(k+1) = A_i x_i(k) + B_i u_i^{p(k)}(\mu(k), 0) + \sum_{j \neq i} W_{ij} u_j^{p(k)}(\mu(k), 0),\ i \in$

$\mathbb{I}_M$ is exponentially stable under the state feedback distributed MPC control law defined by either Theorem 4.1 or Theorem 4.2.

Remark 4.3. Let Assumption 4.1 hold and let $Q_i = \text{diag}(Q_{1i}, \ldots, Q_{Mi})$ with $Q_{ii} > 0, Q_{ij} \geq 0,\ \forall\, i, j \in \mathbb{I}_M, j \neq i$. The nominal closed-loop system $x_i(k+1) = A_i x_i(k) + B_i u_i^{p(k)}(\mu(k), 0) + \sum_{j \neq i} W_{ij} u_j^{p(k)}(\mu(k),\, 0),\ \ i \in \mathbb{I}_M$ is exponentially stable under the state feedback distributed MPC control law defined by either Theorem 4.1 or Theorem 4.2.

4.7 Examples

Controller performance index. For the examples presented in this paper, the controller performance index for each systemwide control configuration is calculated as

$$\Lambda_{\text{cost}}(k) = \frac{1}{k} \sum_{j=0}^{k} \sum_{i=1}^{M} \frac{1}{2} \left[x_i(j)' Q_i x_i(j) + u_i(j)' R_i u_i(j) \right]$$

$$\Delta\Lambda_{\text{cost}}(\text{config})\% = \frac{\Lambda_{\text{cost}}(\text{config}) - \Lambda_{\text{cost}}(\text{cent})}{\Lambda_{\text{cost}}(\text{cent})} \times 100$$

in which $Q_i = C_i' Q_{y_i} C_i + \varepsilon_i I \geq 0, R_i > 0, \varepsilon_i \geq 0$ and k is the simulation time.

4.7.1 Distillation column control

Table 4.1: Constraints on inputs L, V and regulator parameters.

-1.5 $\leq$ V $\leq$ 1.5

-2 $\leq$ L $\leq$ 2

$Q_{y_1} = 50$	$Q_{y_2} = 50$
$R_1 = 1$	$R_2 = 1$
$\varepsilon_1 = 10^{-6}$	$\varepsilon_2 = 10^{-6}$

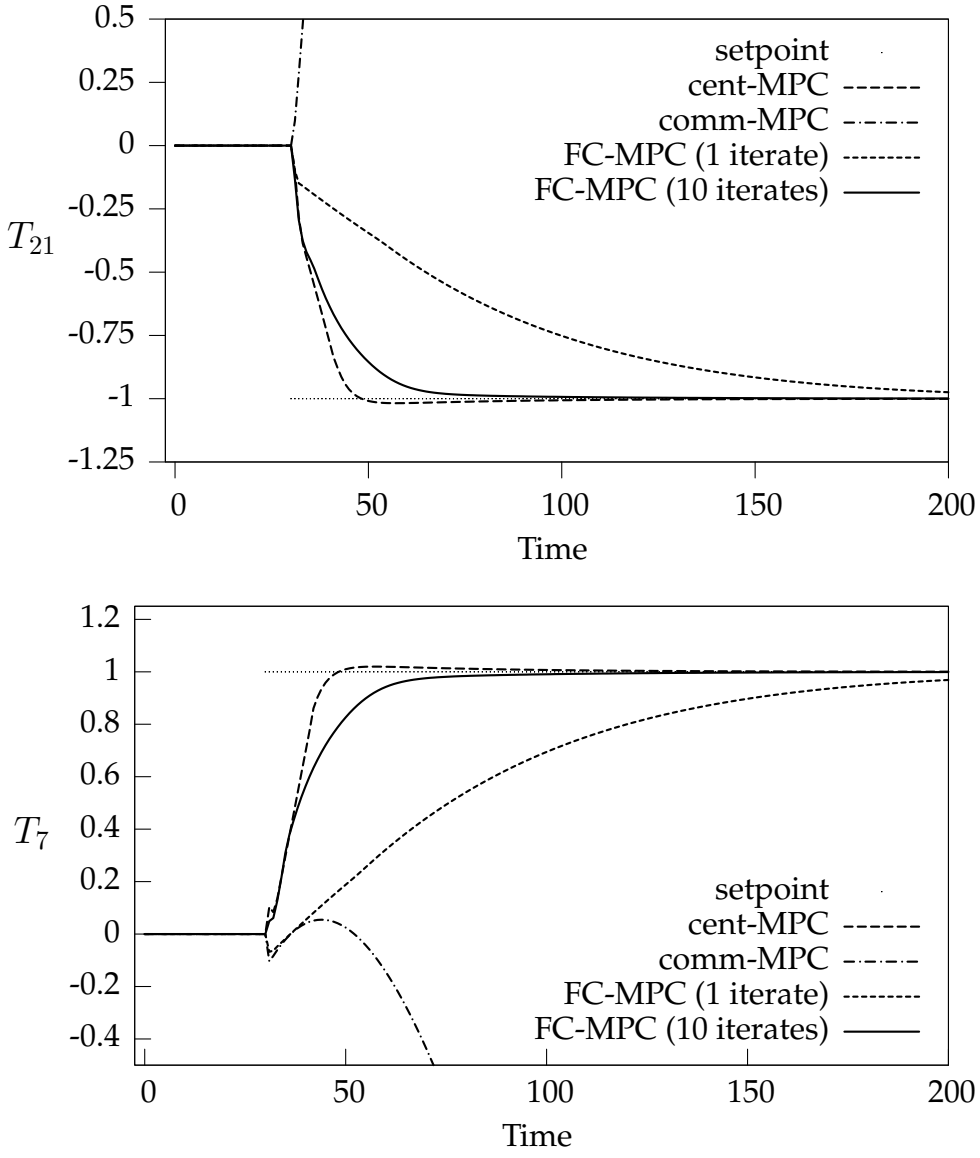

Figure 4.4: Setpoint tracking performance of centralized MPC, communication-based MPC and FC-MPC. Tray temperatures of the distillation column (Ogunnaike and Ray (1994)).

Consider the distillation column of (Ogunnaike and Ray, 1994, p. 813). Tray temperatures act as inferential variables for composition control. The outputs T_{21}, T_7 are the temperatures of trays 21 and 7, respectively and the inputs L, V denote the reflux flowrate and the vapor boilup flowrate to the distillation column. The sampling rate is 1 sec. The implications

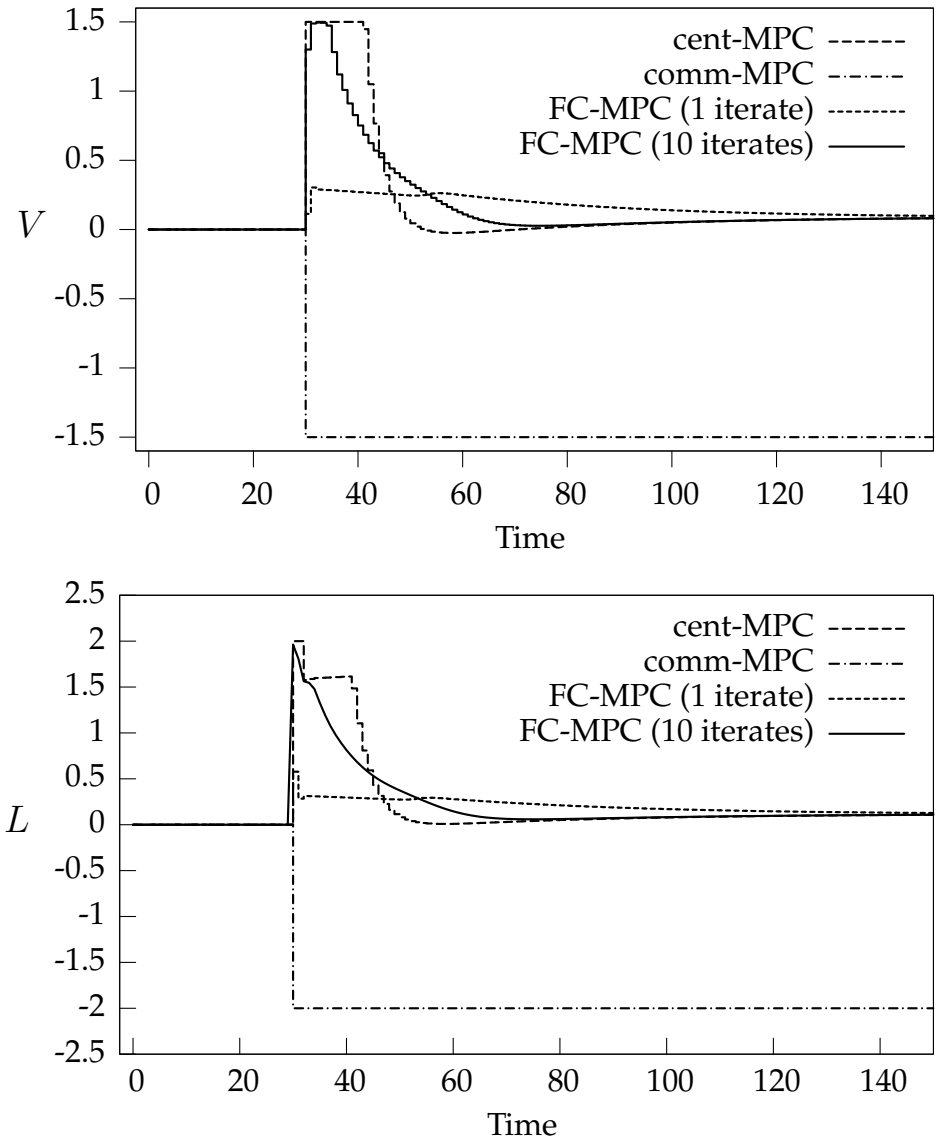

Figure 4.5: Setpoint tracking performance of centralized MPC, communication-based MPC and FC-MPC. Input profile (V and L) for the distillation column (Ogunnaike and Ray (1994)).

of the relative gain array (RGA) elements on controller design has been studied in Skogestad and Morari (1987). While the RGA for this system suggests pairing L with T_{21} and V with T_7, we intentionally choose a bad control variable–manipulated variable pairing. While dealing with subsystem-based control of large-scale systems, situations arise in which an optimal

pairing policy for the controlled and manipulated variable sets (CVs and MVs) either does not exist or is infeasible due to physical or operational constraints. Such situations are not uncommon. From a control perspective, one prerequisite of a reliable subsystem-based systemwide control strategy is the ability to overcome bad CV-MV choices. Figures 4.4 and 4.5 depict the closed-loop performance of centralized MPC (cent-MPC), communication-based MPC (comm-MPC) and FC-MPC when the temperature setpoint of trays 21 and 7 are altered by $-1^{\circ}C$ and $1^{\circ}C$, respectively. For each MPC, a control horizon $N = 25$ is used. The nominal plant model is available in Appendix A (see Table A.2). Input constraints and regulator parameters and constraints are given in Table 4.1.

Table 4.2: Closed-loop performance comparison of centralized MPC, decentralized MPC, communication-based MPC and FC-MPC.

	Λ_{cost}	$\Delta\Lambda_{\text{cost}}\%$
Cent-MPC	1.72	$--$
Comm-MPC	∞	∞
FC-MPC (1 iterate)	6.35	269.2%
FC-MPC (10 iterates)	1.74	1.32%

In the comm-MPC framework, inputs V and L saturate at their constraints and the resulting controller is closed-loop unstable. In principle, the situation here is similar to that depicted in Figure 4.3 (Example 3). The distributed controller derived by terminating the FC-MPC algorithm after just 1 iterate stabilizes the closed-loop system. However, the closed-loop performance of this distributed controller is significantly worse than the performance of centralized MPC. The control costs incurred using the different MPC frameworks are given in Table 4.2. The distributed controller defined by terminating the FC-MPC algorithm after 10 iterates, on the other hand, achieves performance that is within $\sim 1.4\%$ of centralized MPC performance. On iterating the FC-MPC algorithm to convergence, the distributed MPCs achieve

performance that is within a pre-specified tolerance of centralized MPC performance.

4.7.2 Two reactor chain with flash separator

We consider a plant consisting of two continuous stirred tank reactors (CSTRs) followed by a nonadiabatic flash. A schematic of the plant is shown in Figure 4.6. In each of the CSTRs, the desired product B is produced through the irreversible first order reaction $A \xrightarrow{\mathrm{k_1}} B$. An undesirable side reaction $B \xrightarrow{\mathrm{k_2}} C$ results in the consumption of B and in the production of the unwanted side product C. The product stream from CSTR-2 is sent to a nonadiabatic flash to separate the excess A from the product B and the side product C. Reactant A has the highest relative volatility and is the predominant component in the vapor phase. A fraction of the vapor phase is purged and the remaining (A rich) stream is condensed and recycled back to CSTR-1. The liquid phase (exiting from the flash) consists mainly of B and C. The first principles model and parameters for the plant are given in Appendix A (see Tables A.3 to A.5). Input constraints are given in Table 4.3. A linear model for the plant is obtained by linearizing the plant around the steady state corresponding to the maximum yield of B, which is the desired operational objective.

In the distributed MPC frameworks, there are 3 MPCs, one each for the two CSTRs and one for the nonadiabatic flash. In the centralized MPC framework, a single MPC controls the entire plant. The manipulated variables (MVs) for CSTR-1 are the feed flowrate F_0 and the cooling duty Q_{r}. The measured variables are the level of liquid in the reactor H_{r}, the exit mass fractions of A and B *i.e.*, $x_{\mathrm{A_r}}, x_{\mathrm{B_r}}$, respectively and the reactor temperature T_{r}. The controlled variables (CVs) for CSTR-1 are H_{r} and T_{r}. The MVs for CSTR-2 are the feed flowrate F_1 and the reactor cooling load Q_{m}. The (local) measured variables are the level H_{m}, the mass fractions

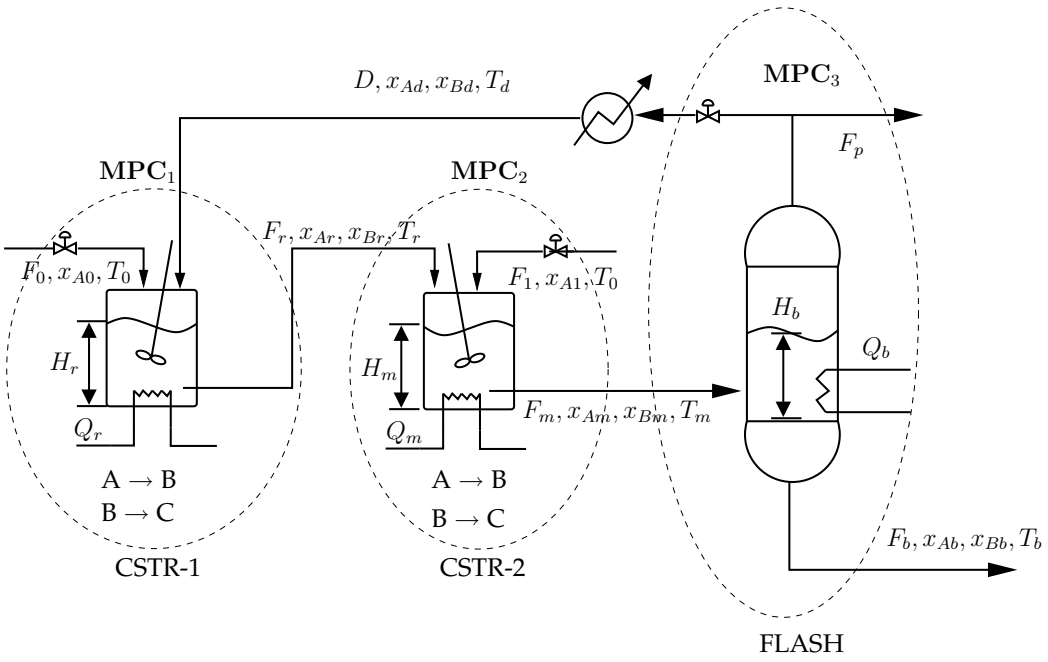

Figure 4.6: Two reactor chain followed by nonadiabatic flash. Vapor phase exiting the flash is predominantly A. Exit flows are a function of the level in the reactor/flash.

of A and B $x_{\mathrm{A_m}}, x_{\mathrm{B_m}}$ at the outlet, and the reactor temperature T_{m} . The CVs are H_{m} and T_{m}. For the nonadiabatic flash, the MVs are the recycle flowrate D and the cooling duty for the flash Q_{b}. The CVs are the level in the flash H_{b} and the temperature T_{b}. The measurements are $H_{\mathrm{b}}, T_{\mathrm{b}}$ and the product stream mass fractions of A and B ($x_{\mathrm{A_b}}$ and $x_{\mathrm{B_b}}$).

Table 4.3: Input constraints for Example 4.7.2. The symbol Δ represents a deviation from the corresponding steady-state value.

$-0.1 \leq \Delta F_0 \leq 0.1$	$-0.15 \leq \Delta Q_r \leq 0.15$
$-0.04 \leq \Delta F_1 \leq 0.04$	$-0.15 \leq \Delta Q_r \leq 0.15$
$-0.1 \leq \Delta D \leq 0.1$	$-0.15 \leq \Delta Q_b \leq 0.15$

Table 4.4: Closed-loop performance comparison of centralized MPC, communication-based MPC and FC-MPC.

	$\Lambda_{\mathrm{cost}} \times 10^{-2}$	$\Delta\Lambda_{\mathrm{cost}}\%$
Cent-MPC	2.0	$--$
Comm-MPC	∞	∞
FC-MPC (1 iterate)	2.13	6%
FC-MPC (10 iterates)	~ 2.0	$< 0.1\%$

The performance of centralized MPC (Cent-MPC), communication-based MPC (Comm-MPC) and FC-MPC are evaluated when a setpoint change corresponding to a 42% increase in the level H_m is made at time 15. The control horizon for each MPC is $N = 15$. Figures 4.7 and 4.8 depict the performance of the different MPC frameworks for the prescribed setpoint change. In the Comm-MPC framework, the flowrate F_1 switches continually between its upper and lower bounds. Subsequently, Comm-MPC leads to unstable closed-loop performance. Both Cent-MPC and FC-MPC (1 iterate) stabilize the closed-loop system. In response to an increase in the setpoint of H_m, the FC-MPC for CSTR-2 orders a maximal increase in flowrate F_1. The flowrate F_1, therefore, saturates at its upper limit. The FC-MPCs for CSTR-1 and the flash cooperate with the FC-MPC for CSTR-2 by initially increasing F_0 and later increasing D, respectively. This feature *i.e.*, cooperation among MPCs is absent under Comm-MPC and is the likely reason for its failure. A performance comparison of the different MPC frameworks is given in Table 4.4. If Algorithm 4.1 is terminated after just 1 iterate, the FC-MPC framework incurs a performance loss of 6% compared to cent-MPC performance. If 10 iterates per sampling interval are possible, the performance of FC-MPC is almost identical to cent-MPC performance.

4.7.3 Unstable three subsystem network

Consider a plant consisting of three subsystems. The nominal subsystem models are available in Appendix A (see Table A.6). Input constraints and regulator parameters are given in Table 4.5. For each MPC, a control horizon $N = 15$ is used. Since each of the subsystems has an unstable decentralized mode, a terminal state constraint that forces the unstable mode to the

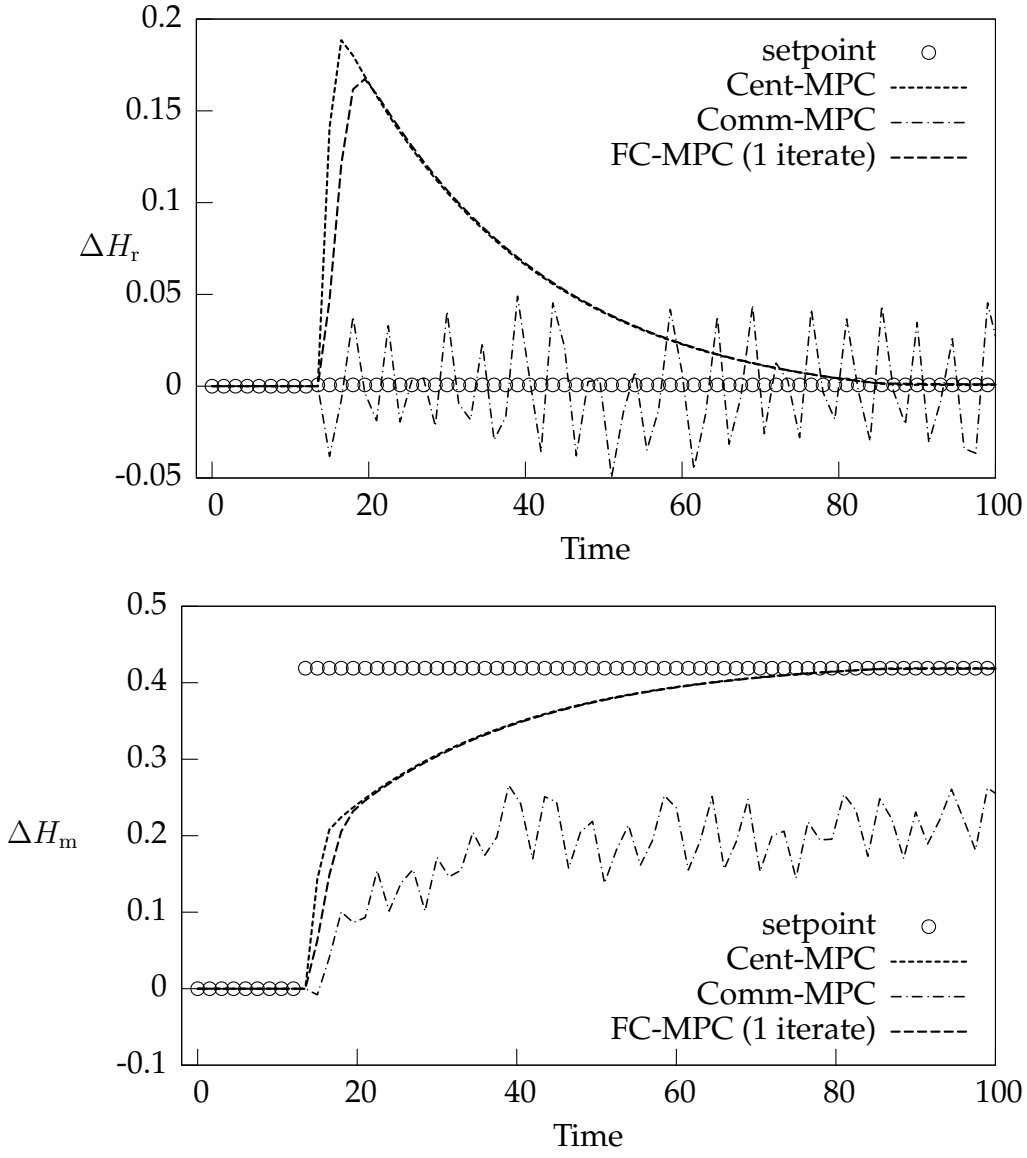

Figure 4.7: Performance of cent-MPC, comm-MPC and FC-MPC when the level setpoint for CSTR-2 is increased by 42%. Setpoint tracking performance of levels H_r and H_m.

origin at the end of the control horizon is employed (Theorem 4.2 for the FC-MPC framework). A setpoint change of 1 and -1 is made to outputs y_1 and y_5, respectively at time = 6. The performance of the distributed controller derived by terminating the FC-MPC algorithm after 1 and 5 iterates, respectively is shown in Figures 4.9-4.11. A closed-loop performance compar-

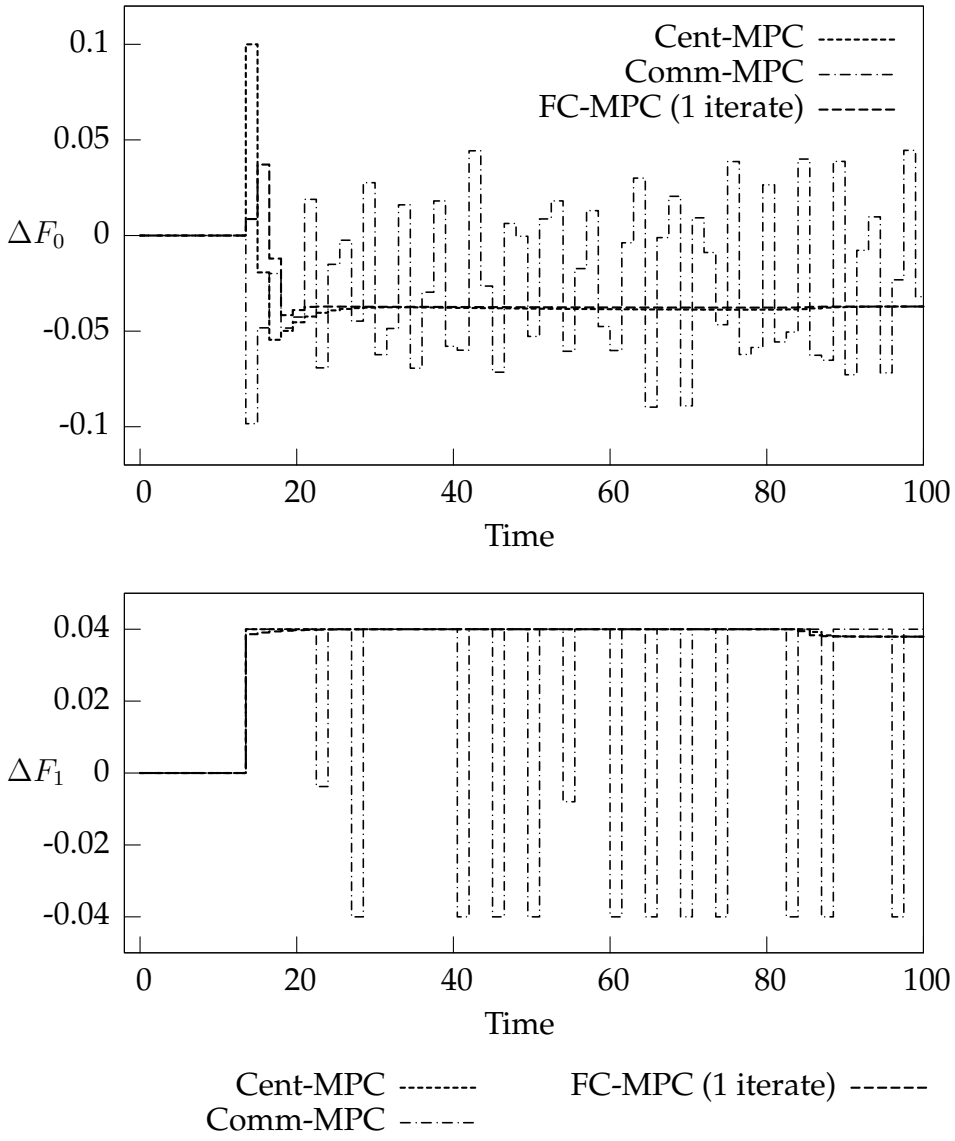

Figure 4.8: Performance of cent-MPC, comm-MPC and FC-MPC when the level setpoint for CSTR-2 is increased by 42%. Setpoint tracking performance of input flowrates F_0 and F_m.

ison of the different MPC based frameworks, for the described setpoint change is given in Table 4.6. The performance loss, compared to centralized MPC, incurred under the FC-MPC formulation terminated after just 1 iterate is $\sim 14\%$, which is a substantial improvement over the performance of decentralized and communication-based MPC. Both the decentralized and

Table 4.5: Input constraints and regulator parameters.

$$\begin{array}{c} \text{-1} \leq u_1 \leq \ 1 \\ \text{-0.15} \leq u_2 \leq 0.15 \\ \text{-1.5} \leq u_3 \leq \ 1.5 \\ \text{-0.2} \leq u_4 \leq \ 0.2 \\ \text{-0.75} \leq u_5 \leq 0.75 \end{array}$$

$$\left| \begin{array}{c} Q_{y_1} = \ 25 \\ R_1 = \ 1 \\ \varepsilon_1 \ = 10^{-6} \end{array} \right\| \left| \begin{array}{c} Q_{y_2} = \ 25 \\ R_2 = \ 1 \\ \varepsilon_2 \ = 10^{-6} \end{array} \right\| \left| \begin{array}{c} Q_{y_3} = \ 1 \\ R_3 = \ 1 \\ \varepsilon_3 \ = 10^{-6} \end{array} \right|$$

Table 4.6: Closed-loop performance comparison of centralized MPC, decentralized MPC, communication-based MPC and FC-MPC.

	Λ_{cost}	$\Delta\Lambda_{\text{cost}}\%$
Cent-MPC	1.78	$--$
Decent-MPC	3.53	98.3%
Comm-MPC	3.53	98.2%
FC-MPC (1 iterate)	2.03	13.9%
FC-MPC (5 iterates)	1.8	0.8%

communication-based MPC frameworks incur a performance loss of $\sim 98\%$ relative to centralized MPC performance. The behavior of the cooperation-based cost function with iteration number at time = 6 is shown in Figure 4.11. At time = 6, convergence to the centralized MPC solution is achieved after ~ 10 iterates.

4.8 Discussion and conclusions

In this chapter, a new distributed, linear MPC framework with guaranteed feasibility, optimality and closed-loop stability properties was described. It is shown that communication-based MPC strategies are unreliable for systemwide control and can lead to closed-loop instability. A cooperation-based distributed MPC algorithm was proposed. The intermediate iterates generated by this cooperation-based MPC algorithm are feasible and the state feedback distributed MPC control law based on any intermediate iterate is nominally closed-loop stable. Therefore,

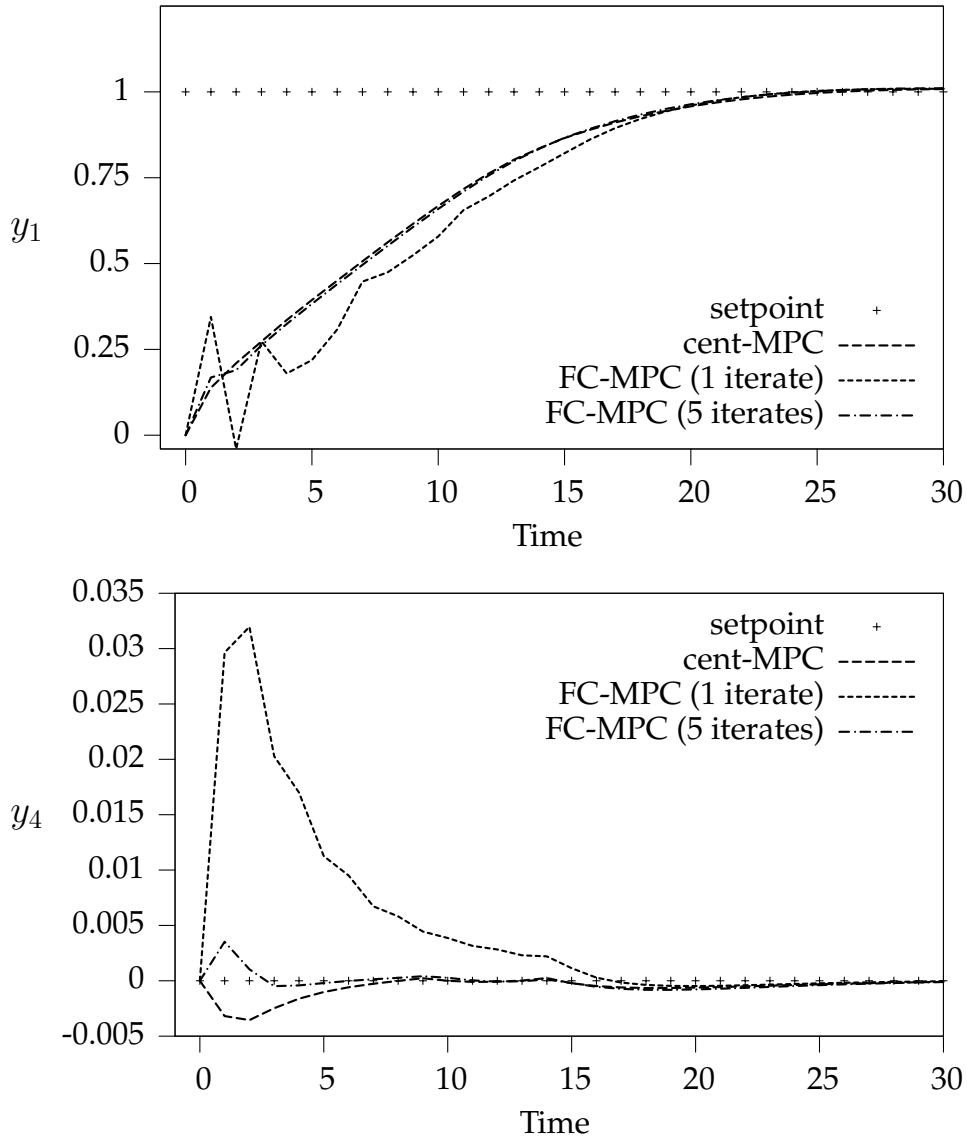

Figure 4.9: Performance of centralized MPC and FC-MPC for the setpoint change described in Example 4.7.3. Setpoint tracking performance of outputs y_1 and y_4.

one can terminate Algorithm 4.1 at end of each sampling interval, irrespective of convergence. At each time k, the states of each subsystem are relayed to all the interconnected subsystems' MPCs. At each iterate p, the MPC for subsystem $i \in \mathbb{I}_M$ calculates its optimal input trajectory $\boldsymbol{u}_i^p$ assuming the input trajectories generated by the interacting subsystems' MPCs remain at

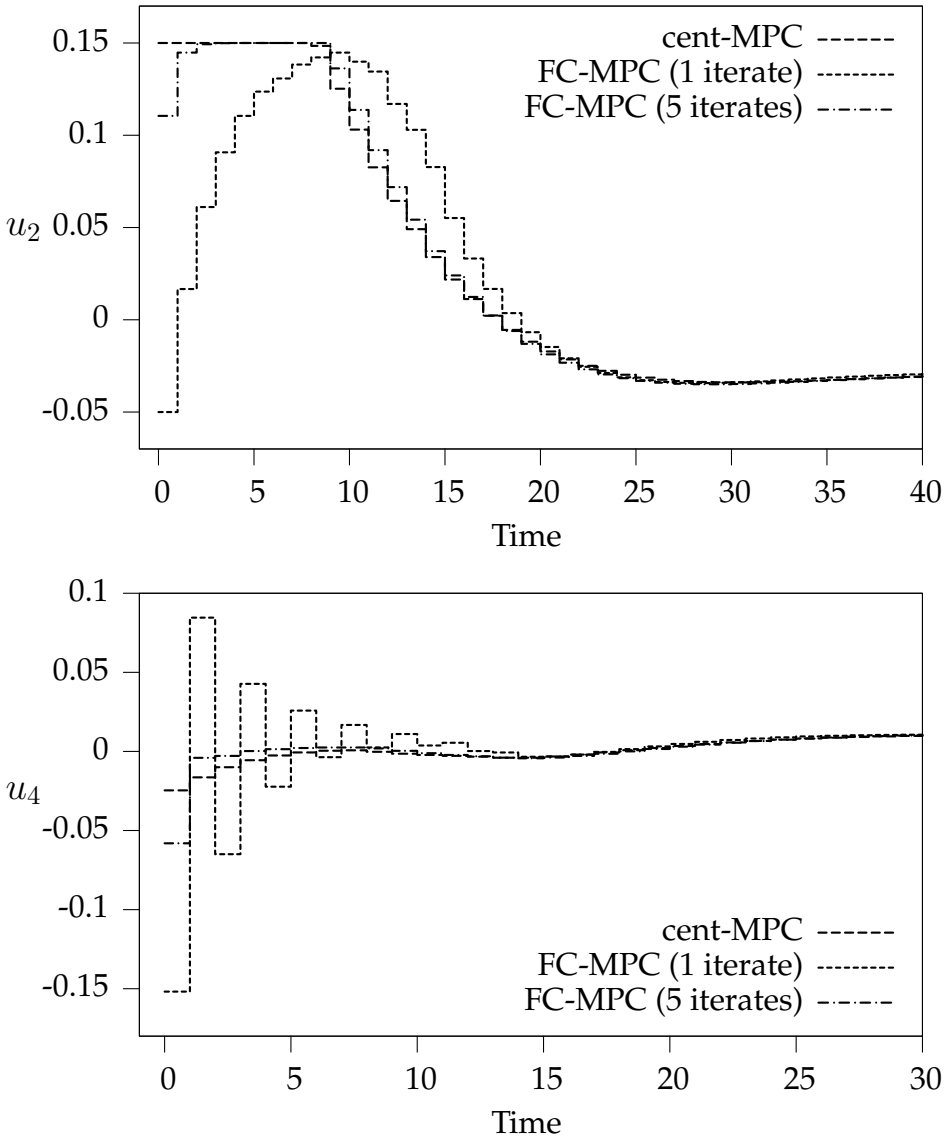

Figure 4.10: Performance of centralized MPC and FC-MPC for the setpoint change described in Example 4.7.3. Inputs u_2 and u_4.

$\boldsymbol{u}_j^{p-1}$, $\forall\, j \neq i$. The recomputed trajectory $\boldsymbol{u}_i^p$ is subsequently communicated to each interconnected subsystem's MPC.

Implementation. For a plant with M subsystems employing decentralized MPCs, conversion to the FC-MPC framework involves the following tasks. First, the interaction models

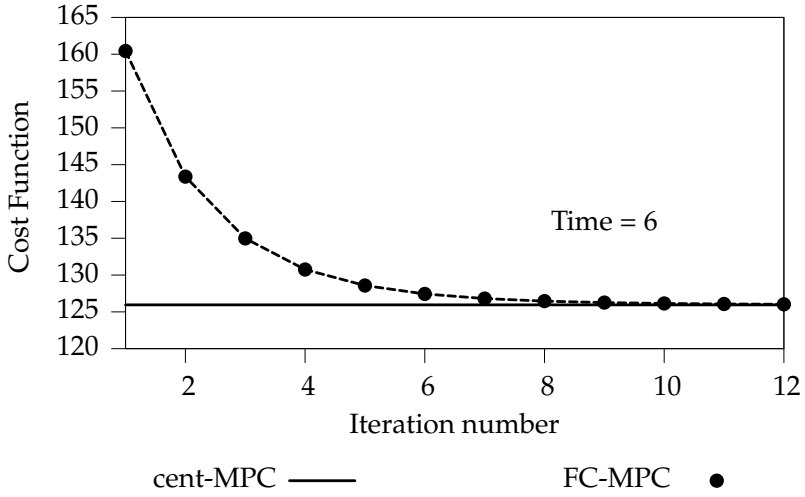

Figure 4.11: Behavior of the FC-MPC cost function with iteration number at time 6. Convergence to the optimal, centralized cost is achieved after ~ 10 iterates.

must be identified. Techniques for identifying the interaction models under closed-loop operating conditions have been described in Gudi and Rawlings (2006). Next, the Hessian and the linear term in the QP for each subsystem MPC need to be modified as shown in Equation 4.8. Notice that in the decentralized MPC framework, the linear term in the QP is modified after each time step; in the FC-MPC framework, the linear term is updated after each iterate. The Hessian in both frameworks is a constant and requires modification only if the models change. Also, unlike centralized MPC, both decentralized MPC and FC-MPC do not require any information regarding the constraints on the external input variables. Finally, a communication protocol must be established for relaying subsystem state information (after each time step) and input trajectory information (after each iterate). This communication protocol can range from data transfer over wireless networks to storing and retrieval of information from a central database. One may also consider utilizing recent developments in technology for control over networks (Baliga and Kumar, 2005; Casavola, Papini, and Franzé, 2006; Imer, Yuksel, and Basar, 2004) to establish a communication protocol suitable for distributed MPC. This issue is

beyond the scope of this work and remains an open research area. Along similar lines, developing reliable buffer strategies in the event of communication disruptions is another important research problem.

The FC-MPC framework allows the practitioner to seamlessly transition from completely decentralized control to completely centralized control. For each subsystem i, by setting $w_i = 1$, $w_j = 0$, $j \neq i$, and by switching off the communication between the subsystems' MPCs, the system reverts to decentralized MPC. On the other hand, iterating Algorithm 4.1 to convergence gives the optimal, centralized MPC solution. By terminating Algorithm 4.1 at intermediate iterates, we obtain performance that lies between the decentralized MPC (base case) and centralized MPC (best case) performance limits allowing the practitioner to investigate the potential control benefits of centralized control without requiring the large control system restructuring and maintenance effort needed to implement and maintain centralized MPC. Taking subsystems off line and bringing subsystems back online are accomplished easily in the FC-MPC framework. Through simple modifications, the FC-MPC framework can be geared to focus on operational objectives (at the expense of optimality), in the spirit of modular multivariable control Meadowcroft, Stephanopoulos, and Brosilow (1992). For instance, it is possible to modify the FC-MPC framework such that only local inputs are used to track a certain output variable. Details of such a modified FC-MPC framework are available in Chapter 7.

4.9 Extensions

Several extensions for the proposed FC-MPC framework are possible. Here, we present two simple extensions.

4.9.1 Rate of change of input penalty and constraint

Constraints and penalties on the rate of change of each subsystem's inputs can be included in the FC-MPC framework. For subsystem $i \in \mathbb{I}_M$, define $\Delta u_i(k) = u_i(k) - u_i(k-1)$. Let $\Delta u_i^{\min} \leq \Delta u_i \leq \Delta u_i^{\max}$, $\forall\, i \in \mathbb{I}_M$. Bound constraints on the rate of change of each subsystem's inputs represent limits on how rapidly the corresponding actuators/valves can move in practice. The stage cost is defined as

$$L_i(x_i, u_i, \Delta u_i) = \frac{1}{2}[x_i' Q_i x_i + u_i' R_i u_i + \Delta u_i' S_i \Delta u_i] \tag{4.15}$$

in which $Q_i \geq 0$, $R_i + S_i > 0$ are symmetric matrices and $(A_i, Q_i^{1/2})$ is detectable. To convert Equation (4.15) to the standard form (see Equation (4.3)), we use a strategy similar to that described in Muske and Rawlings (1993) for single MPCs. The decentralized state x_{ii} for subsystem $i \in \mathbb{I}_M$ is augmented with the subsystem input u_i obtained at the previous time step. At time k, define $z_{ii}(k) = [x_{ii}'(k), u_i(k-1)']$ to be the augmented decentralized state for

subsystem $i \in \mathbb{I}_M$. The augmented decentralized model is

$$z_{ii}(k+1) = \widetilde{A}_{ii} z_{ii}(k) + \widetilde{B}_{ii} u_i(k), \tag{4.16a}$$

$$y_{ii}(k+1) = \widetilde{C}_{ii} z_{ii}(k), \tag{4.16b}$$

$$\text{in which } \widetilde{A}_{ii} = \begin{bmatrix} A_{ii} & 0 \\ 0 & 0 \end{bmatrix}, \qquad \widetilde{B}_{ii} \begin{bmatrix} B_{ii} \\ I \end{bmatrix}, \quad \widetilde{C}_{ii} = \begin{bmatrix} C_{ii} & 0 \end{bmatrix}$$

The augmented CM state is defined as $z_i = [x_{i1}', \ldots, z_{ii}', \ldots, z_{iM}']'$. The augmented CM for subsystem i is

$$z_i(k+1) = \widetilde{A}_i z_i(k) + \widetilde{B}_i u_i(k) + \sum_{j \neq i} \widetilde{W}_{ij} u_j(k) \tag{4.17a}$$

$$y_i(k) = \widetilde{C}_i z_i(k) \tag{4.17b}$$

in which

$$\widetilde{A}_i = \mathrm{diag}(A_{i1}, \ldots, \widetilde{A}_{ii}, \ldots, A_{iM}),\ \widetilde{B}_i = \begin{bmatrix} B_i \\ I \end{bmatrix},\ \widetilde{W}_{ij} = \begin{bmatrix} W_{ij} \\ 0 \end{bmatrix},\ \widetilde{C}_i = \mathrm{diag}(C_{i1}, \ldots, \widetilde{C}_{ii}, \ldots, C_{iM})$$

The stage cost defined in Equation (4.15) can be rewritten as

$$L_i(z_i, u_i) = \frac{1}{2}[z_i' \widetilde{Q}_i z_i + u_i' \widetilde{R}_i u_i + z_i' \widetilde{M}_i u_i] \tag{4.18}$$

where

$$\widetilde{Q}_i = \begin{pmatrix} Q_i & \\ & S_i \end{pmatrix}, \qquad \widetilde{R}_i = R_i + S_i, \qquad \widetilde{M}_i = \begin{bmatrix} 0 \\ -S_i \end{bmatrix}$$

Notice that if $S_i = 0$, we revert to the earlier definition of the stage cost (Equation (4.3), p. 30). The cost function $\phi_i(\cdot)$ (see Equation (4.5), p. 34) is obtained by using Equation (4.18) for the definition of $L_i(\cdot)$. Let $\overline{\boldsymbol{z}}_{ii} = [z_{ii}(1)', \ldots, z_{ii}(N)']'$. Using Equation (4.16), we write $\overline{\boldsymbol{z}}_{ii} = \widetilde{F}_{ii}\overline{\boldsymbol{u}}_i + \widetilde{e}_{ii}z_{ii}(0)$, in which

$$\widetilde{F}_{ii} = \begin{bmatrix} \widetilde{B}_{ii} & 0 & \ldots & \ldots & 0 \\ \widetilde{A}_{ii}\widetilde{B}_{ii} & \widetilde{B}_{ii} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \widetilde{A}_{ii}^{N-1}\widetilde{B}_{ii} & \ldots & \ldots & \ldots & \widetilde{B}_{ii} \end{bmatrix}, \ \widetilde{e}_{ii} = \begin{bmatrix} \widetilde{A}_{ii} \\ \widetilde{A}_{ii}^2 \\ \vdots \\ \widetilde{A}_{ii}^N \end{bmatrix}$$

Define $T_{ii} = \begin{bmatrix} 0 & -I \end{bmatrix}$ for each $i \in \mathbb{I}_M$. The bound constraint on Δu_i can be expressed as $\Delta u_i^{\min} \leq T_{ii}z_{ii} + u_i \leq \Delta u_i^{\max}$. The FC-MPC optimization problem for subsystem i is, therefore,

$$\overline{\boldsymbol{u}}_i^{*(p)} \in \arg \min_{\overline{\boldsymbol{u}}_i} \ \frac{1}{2}\overline{\boldsymbol{u}}_i(k)'\mathcal{R}_i\overline{\boldsymbol{u}}_i(k) + \left(r_i(k) + \sum_{j=1, j\neq i}^{M} \mathcal{H}_{ij}\overline{\boldsymbol{u}}_j^{p-1}(k) \right)' \overline{\boldsymbol{u}}_i(k) + \text{constant} \tag{4.19a}$$

subject to

$$\overline{\boldsymbol{u}}_i \in \mathcal{U}_i \tag{4.19b}$$

$$\Pi_i^{\min} \leq \mathbb{D}_i\overline{\boldsymbol{u}}_i + \mathbb{Z}_i\widetilde{e}_{ii}z_{ii}(k) \leq \Pi_i^{\max} \tag{4.19c}$$

in which $\mathbb{Q}_i = \text{diag}(\widetilde{Q}_i(1), \ldots, \widetilde{Q}_i(N-1), \overline{Q}_i)$, $\mathbb{R}_i = \text{diag}(\widetilde{R}_i(0), \widetilde{R}_i(1), \ldots, \widetilde{R}_i(N-1))$,

$$\mathbb{M}_i = \begin{pmatrix} 0 & \widetilde{M}_i & & & \\ & 0 & \widetilde{M}_i & & \\ & & 0 & \ddots & \\ & & & \ddots & \widetilde{M}_i \\ 0 & 0 & \ldots & \ldots & 0 \end{pmatrix}, \quad \widetilde{P}_i = \begin{bmatrix} \widetilde{M}_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \widetilde{f}_i = \begin{bmatrix} \widetilde{A}_i \\ \widetilde{A}_i^2 \\ \vdots \\ \widetilde{A}_i^N \end{bmatrix}$$

$$\Pi_i^{\min} = \begin{bmatrix} \Delta u_i^{\min} - T_{ii} z_{ii}(k) \\ \Delta u_i^{\min} \\ \vdots \\ \vdots \\ \Delta u_i^{\min} \end{bmatrix}, \quad \Pi_i^{\max} = \begin{bmatrix} \Delta u_i^{\max} - T_{ii} z_{ii}(k) \\ \Delta u_i^{\max} \\ \vdots \\ \vdots \\ \Delta u_i^{\max} \end{bmatrix}, \quad \mathbb{Z}_i = \begin{bmatrix} 0 & 0 & & 0 \\ T_{ii} & & & \\ & \ddots & & \\ & & T_{ii} & 0 \end{bmatrix},$$

$$\widetilde{E}_{ii} = \begin{bmatrix} \widetilde{B}_i & 0 & \ldots & \ldots & 0 \\ \widetilde{A}_i \widetilde{B}_i & \widetilde{B}_i & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \widetilde{A}_i^{N-1} \widetilde{B}_i & \ldots & \ldots & \ldots & \widetilde{B}_i \end{bmatrix}, \qquad \widetilde{E}_{ij} = \begin{bmatrix} \widetilde{W}_{ij} & 0 & \ldots & \ldots & 0 \\ \widetilde{A}_i \widetilde{W}_{ij} & \widetilde{W}_{ij} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \widetilde{A}_i^{N-1} \widetilde{W}_{ij} & \ldots & \ldots & \ldots & \widetilde{W}_{ij} \end{bmatrix},$$

for each $j \in \mathbb{I}_M, j \neq i$. Also,

$$\mathcal{R}_i = w_i[\mathbb{R}_i + \widetilde{E}_{ii}{}'\mathbb{Q}_i\widetilde{E}_{ii} + 2\widetilde{E}_{ii}{}'\mathbb{M}_i] + \sum_{j\neq i}^{M} w_j\widetilde{E}_{ji}{}'\mathbb{Q}_j\widetilde{E}_{ji}$$

$$\mathcal{H}_{ij} = \sum_{l=1}^{M} w_l\widetilde{E}_{li}{}'\mathbb{Q}_l\widetilde{E}_{lj} + \mathbb{M}_i{}'\widetilde{E}_{ij} + \widetilde{E}_{ji}{}'\mathbb{M}_j$$

$$r_i(k) = w_i[\widetilde{E}_{ii}{}'\mathbb{Q}_i f_i z_i(k) + \mathbb{M}_i{}' f_i z_i(k) + \widetilde{P}_i z_i(k)] + \sum_{j\neq i}^{M} w_j\widetilde{E}_{ji}{}'\mathbb{Q}_j f_j z_j(k)$$

$$\mathbb{D}_i = \mathcal{T}_i\widetilde{F}_{ii} + I$$

The terminal penalty is obtained using Theorem 4.1 for stable systems or Theorem 4.2 for systems with unstable decentralized modes, and replacing each A_i, Q_i with $\widetilde{A}_i$, $\widetilde{Q}_i$, respectively. For systems with unstable decentralized modes, a terminal decentralized state constraint is also required (see Section 4.6.2). All established properties (feasibility, optimality and closed-loop stability) apply for this case.

4.9.2 Coupled subsystem input constraints

The FC-MPC formulation can be employed for control of systems with coupled subsystem input constraints of the form $\sum_{i=1}^{M} H_i u_i \leq h, h > 0$. At time k and iterate p, the FC-MPC

optimization problem for subsystem $i \in \mathbb{I}_M$ is

$$\boldsymbol{u}_i^{*(p)}(k) \in \arg \min_{\boldsymbol{u}_i} \sum_{r=1}^{M} w_r \Phi_r \left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x_r(k) \right) \quad (4.20a)$$

subject to

$$u_i(l|k) \in \Omega_i, \;\; k \leq l \leq k+N-1 \quad (4.20b)$$

$$u_i(l|k) = 0, \;\; k+N \leq l \quad (4.20c)$$

$$H_i u_i(l|k) + \sum_{j=1, j\neq i}^{M} H_j u_j^{p-1}(l|k) \leq h, \;\; k \leq l \leq k+N-1 \quad (4.20d)$$

It can be shown that the sequence of cost functions generated by Algorithm 4.1 (solving the optimization problem of Equation (4.20) instead) is a nonincreasing function of the iteration number and converges. Also, the distributed MPC control law based on any intermediate iterate is guaranteed to be feasible and closed-loop stable. Let Φ^∞ be the converged cost function value and let $S^\infty = \{(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M) \,|\, \Phi(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M; \mu) = \Phi^\infty\}$ denote the limit set. Using strict convexity of the objective, it can be shown that Algorithm 4.1 converges to a point $\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_M^\infty$ in S^∞. Because a coupled input constraint is present, the converged solution $\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_M^\infty$ may be different from the optimal centralized solution. We present two examples to illustrate possible nonoptimality of Algorithm 4.1 in the presence of coupled subsystems' input constraints.

Example for nonoptimality of Algorithm 4.1 in the presence of coupled constraints. A simple optimization example is described here. Consider the following optimization problem in

decision variables u_1 and u_2

$$\min_{u_1,u_2} \ (u_1 - 1)^2 + (u_2 - 1)^2 \tag{4.21a}$$

subject to

$$1 \geq u_1 \geq 0 \tag{4.21b}$$

$$1 \geq u_2 \geq 0 \tag{4.21c}$$

$$1 - u_1 - u_2 \geq 0 \tag{4.21d}$$

A graphical representation of the optimization problem is given in Figure 4.12. The optimal solution to the optimization problem of Equation (4.21) is $(u_1^*, u_2^*) = (\frac{1}{2}, \frac{1}{2})$.

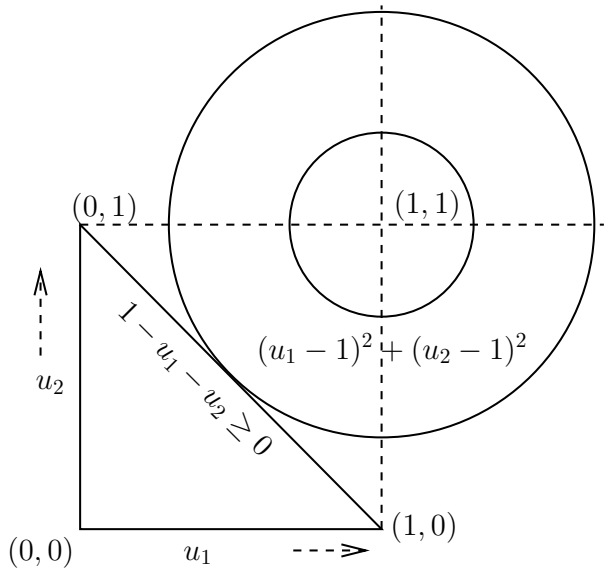

Figure 4.12: Example demonstrating nonoptimality of Algorithm 4.1 in the presence of coupled decision variable constraints.

In the cooperation-based distributed optimization framework, the optimizer for u_1 solves the following optimization problem at iterate p

$$\min_{u_1} (u_1 - 1)^2 + (u_2^{p-1} - 1)^2 \tag{4.22a}$$

subject to

$$1 \geq u_1 \geq 0 \tag{4.22b}$$

$$1 - u_1 - u_2^{p-1} \geq 0 \tag{4.22c}$$

Similarly, the optimizer for u_2 solves

$$\min_{u_2} (u_1^{p-1} - 1)^2 + (u_2 - 1)^2 \tag{4.23a}$$

subject to

$$1 \geq u_2 \geq 0 \tag{4.23b}$$

$$1 - u_1^{p-1} - u_2 \geq 0 \tag{4.23c}$$

At iterate p, let the solution to the optimization problems described in Equations (4.22) and (4.23) be $u_1^{*(p)}$ and $u_2^{*(p)}$, respectively. We consider three cases.

Case 1. $(u_1^0, u_2^0) = (1, 0)$

Using Algorithm 4.1 employing the optimization problems given by Equations (4.22) and (4.23) gives $(u_1^p, u_2^p) = (1, 0) = (u_1^{p-1}, u_2^{p-1}) = (u_1^0, u_2^0)$. Algorithm 4.1, therefore, gives a nonoptimal solution for all p.

Case 2. $(u_1^0, u_2^0) = (\frac{3}{4}, \frac{1}{4})$

Using Algorithm 4.1, we have $(u_1^p, u_2^p) = (\frac{3}{4}, \frac{1}{4}) = (u_1^{p-1}, u_2^{p-1}) = (u_1^0, u_2^0)$. Like case 1, Algorithm 4.1 gives a nonoptimal solution for all values of p.

Case 3. $(u_1^0, u_2^0) = (0, 0)$

For this case, we have using Algorithm 4.1 that $u_1^{*(1)} = 1$ and $u_2^{*(1)} = 1$. The first iterate, $(u_1^1, u_2^1) = (\frac{1}{2}, \frac{1}{2})$, the optimal solution. Hence, unlike cases 1 and 2, Algorithm 4.1 converges to the optimal solution after just 1 iterate.

Distributed MPC of distillation column with coupled constraints. We consider the distillation column described in Section 4.7.1 with an additional coupled input constraint $0 \leq L + V \leq 0.25$. The performance of FC-MPC at convergence is compared to centralized MPC (see Figure 4.13). While the coupled input constraint is active, the performance of FC-MPC (convergence) is different from centralized MPC. If the coupled input constraint is inactive, the performance of FC-MPC (convergence) is within a pre-specified tolerance of centralized MPC. The closed-loop control cost of FC-MPC (convergence) exceeds that of centralized MPC by nearly 1%.

4.10 Appendix

4.10.1 Proof for Lemma 4.1

Proof. It follows from compactness of $\mathcal{X}$ and continuity of the linear mapping $\mathcal{A}(\cdot)$ that the set $\mathcal{B}$ is compact. Let $\mathcal{A} = V\Sigma W'$ denote a singular value decomposition of $\mathcal{A}$. Also, let $r = \text{rank}(\mathcal{A})$. Therefore, $\Sigma W'x = V'b$. If $r < m$ then a necessary and sufficient condition for

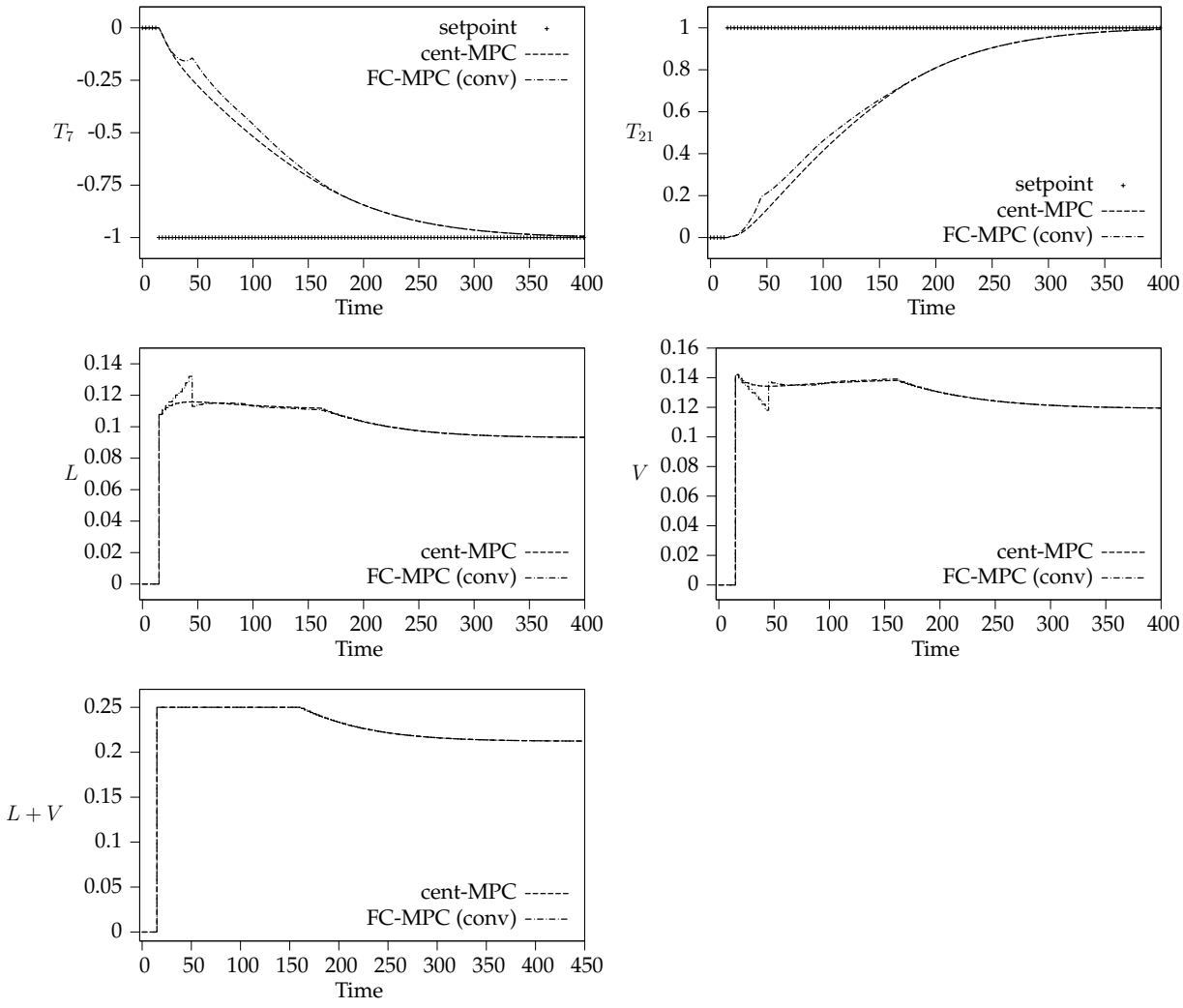

Figure 4.13: Setpoint tracking performance of centralized MPC and FC-MPC (convergence). An additional coupled input constraint $0 \leq L + V \leq 0.25$ is employed.

the system $\mathcal{A}x = b$ to be solvable is that the last $m - r$ left singular vectors are orthogonal to b. If $V = [v_1, v_2, \ldots, v_m]$, $W = [w_1, w_2, \ldots, w_n]$ and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots)$ then $\overline{x}(b) = \sum_{i=1}^{r} \frac{v_i' b}{\sigma_i} w_i$ is a solution to the system $\mathcal{A}x = b$ with minimum l_2 norm (Horn and Johnson, 1985, p. 429).

Since $0 \in \mathcal{X}$, there exists a $\varepsilon > 0$ such that $\overline{x}(b) \in \mathcal{X}$ for all $b \in B_\varepsilon(0)$. For $b \in B_\varepsilon(0)$, choose $K_1 = \sum_{i=1}^{r} \frac{\|v_i\| \|w_i\|}{\sigma_i}$. This choice gives $\|\overline{x}(b)\| \leq K_1 \|b\|, \mathcal{A}\overline{x}(b) = b, \overline{x}(b) \in \mathcal{X}$ with K_1

independent of the choice of $b \in B_\varepsilon(0)$.

Define $B \setminus B_\varepsilon(0) = \{b \mid b \in \mathcal{B}, \|b\| > \varepsilon\}$. Compactness of $\mathcal{X}$ implies $\exists\, R > 0$ such that $\|x\| \leq R, \forall x \in \mathcal{X}$. Therefore, $\|x\| \leq \frac{R}{\varepsilon}\|b\|,\ \forall\, b \in B \setminus B_\varepsilon(0), x \in \mathcal{X}$. The choice $K = \max(K_1, \frac{R}{\varepsilon})$ gives $\|\overline{x}(b)\| \leq K\|b\|,\ \forall\, b \in \mathcal{B}$. □

4.10.2 Proof for Lemma 4.6

Proof. The proof is by induction. At time $k = 0$, the FC-MPC algorithm is initialized with the input sequence $u_i(k+l|k) = 0,\ l \geq 0,\ \forall\ i \in \mathbb{I}_M$. Hence $J_N^0(\mu(0)) = \widetilde{J}_N(\mu(0))$. We know from Lemma 4.6 that $J_N^{p(0)}(\mu(0)) \leq J_N^0(\mu(0)) = \widetilde{J}_N(\mu(0))$. The relationship (Equation (4.13)), therefore, holds at $k = 0$. At time $k = 1$, we have

$$\begin{aligned} J_N^{p(1)}(\mu(1)) \leq J_N^0(\mu(1)) &= J_N^{p(0)}(\mu(0)) - \sum_{i=1}^{M} w_i L_i(x_i(0), u_i^{p(0)}(0)) \\ &\leq \widetilde{J}_N(\mu(0)) - \sum_{i=1}^{M} w_i L_i(x_i(0), 0) \\ &\leq \widetilde{J}_N(\mu(0)) \end{aligned}$$

Hence the relationship in Equation (4.13) is true for $k = 1$.

Assume now that the result is true for some time $k > 1$. At time $k + 1$,

$$\begin{aligned} J_N^{p(k+1)}(\mu(k+1) \leq J_N^0(\mu(k+1)) &= J_N^{p(k)}(\mu(k)) - \sum_{i=1}^{M} w_i L_i \left(x_i(k), u_i^{p(k)}(k) \right) \\ &\leq J_N^{p(k)}\left(\mu(k)\right) - \sum_{i=1}^{M} w_i L_i \left(x_i(k), 0 \right) \\ &\leq \widetilde{J}_N(\mu(0)) - \sum_{j=0}^{k} \sum_{i=1}^{M} w_i L_i(x_i(j), 0) \\ &\leq \widetilde{J}_N(\mu(0)) \end{aligned}$$

The result is, therefore, true for all $k \geq 0$, as claimed. □

4.10.3 Lipschitz continuity of the distributed MPC control law: Stable systems

Lemma 4.7. *Let the input constraints in the FC-MPC optimization problem of Equation (4.9) be specified in terms of a collection of linear inequalities such that the set of active constraints is linearly independent for each $i \in \mathbb{I}_M$. Let Assumption 4.6 hold. The input trajectory $\overline{\boldsymbol{u}}_i^p(\mu)$, $\forall\, i \in \mathbb{I}_M$ generated by Algorithm 4.1 is a Lipschitz continuous function of the set of subsystem states μ for all $p \in \mathbb{I}_+$, $p \leq p^*$.*

Proof. Lipschitz continuity of the control law in the set of subsystem states is proved in two steps. First, we show that the solution to the FC-MPC optimization problem (Equation (4.9)) for each subsystem i is Lipschitz continuous in the data. In the FC-MPC optimization problem of Equation (4.9), $\mathcal{R}_i > 0$. The solution to the FC-MPC optimization problem is, therefore, unique. The parameters that vary in the data are μ and the input trajectories Δ_{-i}^{p-1}, in which $\Delta_{-i}^{p-1} = \overline{\boldsymbol{u}}_1^{p-1}, \ldots, \overline{\boldsymbol{u}}_{i-1}^{p-1}, \overline{\boldsymbol{u}}_{i+1}^{p-1}, \ldots, \overline{\boldsymbol{u}}_M^{p-1}$. Let $\overline{\boldsymbol{u}}_i^{*(p)}(\mu; \Delta_{-i}^{p-1}(\mu))$ represent the solution to Equation (4.9) at iterate p and system state μ. Also, let $\zeta = [z_1, z_2, \ldots, z_M]$. By assumption, the set

of active constraints is linearly independent. From (Hager, 1979, Theorem 3.1), $\exists\ \rho < \infty$ such that

$$\|\overline{\boldsymbol{u}}_i^{*(p)}(\mu;\Delta_{-i}^{p-1}(\mu)) - \overline{\boldsymbol{u}}_i^{*(p)}(\zeta;\Delta_{-i}^{p-1}(\zeta)\| \leq \rho\left(\|\mu-\zeta\|^2 + \sum_{j\neq i}^{M}\|\overline{\boldsymbol{u}}_j^{p-1}(\mu) - \overline{\boldsymbol{u}}_j^{p-1}(\zeta)\|^2\right)^{1/2}$$

From Algorithm 4.1, we have

$$\begin{aligned}\|\overline{\boldsymbol{u}}_i^{p}(\mu) - \overline{\boldsymbol{u}}_i^{p}(\zeta)\| &\leq w_i\|\overline{\boldsymbol{u}}_i^{*(p)}(\mu;\ \cdot) - \overline{\boldsymbol{u}}_i^{*(p)}(\zeta;\ \cdot)\| \\ &\quad + (1-w_i)\,\|\overline{\boldsymbol{u}}_i^{p-1}(\mu) - \overline{\boldsymbol{u}}_i^{p-1}(\zeta)\| \\ &\leq \rho w_i\left(\|\mu-\zeta\|^2 + \sum_{j\neq i}^{M}\|\overline{\boldsymbol{u}}_j^{p-1}(\mu) - \overline{\boldsymbol{u}}_j^{p-1}(\zeta)\|^2\right)^{1/2} \\ &\quad + (1-w_i)\,\|\overline{\boldsymbol{u}}_i^{p-1}(\mu) - \overline{\boldsymbol{u}}_i^{p-1}(\zeta)\|,\ p\in\mathbb{I}_+ \end{aligned} \tag{4.24}$$

It follows from Equation (4.24) that if $\overline{\boldsymbol{u}}_i^{p-1}(\mu)$ is Lipschitz continuous w.r.t μ for all $i\in\mathbb{I}_M$ then $\overline{\boldsymbol{u}}_i^{p}(\mu)$ is Lipschitz continuous w.r.t μ.

If $k=0$, we choose $\boldsymbol{u}_i^0(0) = [0,0,\ldots]', \forall\ i\in\mathbb{I}_M$. For $k>0$, we have (Equation (4.12))

$$\overline{\boldsymbol{u}}_i^0(k) = \left[u_i^{p(k-1)}(\mu(k-1),1)', \ldots, u_i^{p(k-1)}(\mu(k-1),N-1)', 0\right]$$

Either initialization is independent of the current system state μ. Since the models are causal, $\overline{\boldsymbol{u}}_i^1(\mu)$ is Lipschitz continuous in μ. Subsequently, by induction, $\overline{\boldsymbol{u}}_i^p(\mu)$ is Lipschitz continuous in μ for all $p\in\mathbb{I}_+$. For $p_{\max}(k)\leq p^* < \infty$ for all $k\geq 0$, a global Lipschitz constant can be estimated. By definition, $\boldsymbol{u}_i^p(\mu) = [\overline{\boldsymbol{u}}_i^p(\mu)', 0, 0, \ldots]',\ i\in\mathbb{I}_M$. Hence, $\boldsymbol{u}_i^p(\mu)$ is a Lipschitz continuous function of μ for all $p\in\mathbb{I}_+,\ p\leq p_{\max}$. □

Corollary 4.7.1. *The distributed MPC control law $u_i^p(\mu, 0),\ i \in \mathbb{I}_M$ is Lipschitz continuous in μ for all $p \in \mathbb{I}_+,\ p \leq p^*$.*

4.10.4 Proof for Theorem 4.1

Proof. To prove exponential stability, the value function $J_N^{p(k)}(\mu(k))$ is a candidate Lyapunov function. We need to show (Vidyasagar, 1993, p. 267) there exists constants $a, b, c > 0$ such that

$$a \sum_{i=1}^{M} \|x_i(k)\|^2 \leq J_N^{p(k)}(\mu(k)) \leq b \sum_{i=1}^{M} \|x_i(k)\|^2 \tag{4.25a}$$

$$\Delta J_N^{p(k)}(\mu(k)) \leq -c \sum_{i=1}^{M} \|x_i(k)\|^2 \tag{4.25b}$$

in which $\Delta J_N^{p(k)}(\mu(k)) = J_N^{p(k+1)}(\mu(k+1)) - J_N^{p(k)}(\mu(k))$.

Let $\boldsymbol{x}_i^p = [x_i^p(\mu, 1)', x_i^p(\mu, 2)', \ldots]'$ denote the state trajectory for subsystem $i \in \mathbb{I}_M$ generated by the input trajectories $\boldsymbol{u}_1^p(\mu), \ldots, \boldsymbol{u}_M^p(\mu)$ obtained after $p \in \mathbb{I}_+$ (Algorithm 4.1) iterates and initial state μ. Rewriting the cooperation-based cost function in terms of the calculated state and input trajectories, we have

$$J_N^{p(k)}(\mu(k)) = \sum_{i=1}^{M} w_i \left[\sum_{l=0}^{N-1} L_i(x_i^{p(k)}(\mu(k), l), u_i^{p(k)}(\mu(k), l)) + \frac{1}{2} \|x_i^{p(k)}(\mu(k), N)\|_{\overline{Q}_i}^2 \right]$$

in which Q_i, R_i and $\overline{Q}_i$ are all positive definite and $x_i^{p(k)}(\mu(k), 0) = x_i(k), i \in \mathbb{I}_M$.

Because $Q_i > 0$, there exists an $a > 0$ such that $a \sum_{i=1}^{M} \|x_i(k)\|^2 \leq J_N^{p(k)}(\mu(k))$. One possible choice is $a = \min_{i \in \mathbb{I}_M} \frac{1}{2} w_i \lambda_{\min}(Q_i)$. From

$$\Delta J_N^{p(k)}(\mu(k)) \leq -\sum_{i=1}^{M} w_i L_i \left(x_i(k), u_i^{p(k)}(\mu(k), 0) \right) \leq -\sum_{i=1}^{M} w_i \frac{1}{2} x_i(k)' Q_i x_i(k),$$

there exists $c > 0$ such that $\Delta J_N^{p(k)}(\mu(k)) \leq -c \sum_{i=1}^{M} \|x_i(k)\|^2$. One possible choice for c is $c = \min_{i \in \mathbb{I}_M} \frac{1}{2} w_i \lambda_{\min}(Q_i)$.

At time $k = 0$, each subsystem's FC-MPC optimization is initialized with the zero input trajectory. Using Lemma 4.4, we have $J_N^{p(0)}(\mu(0)) \leq \sigma \sum_{i=1}^{M} \|x_i(0)\|^2$, in which $0 < \max_{i \in \mathbb{I}_M} w_i \lambda_{\max}(\overline{Q}_i) \leq \sigma$. Since $0 \in \text{int}\,(\Omega_1 \times \ldots \Omega_M)$ and the origin is Lyapunov stable and attractive with the cost relationship given in Lemma 4.6, there exists $\varepsilon > 0$ such that the input constraints remain inactive in each subsystem's FC-MPC optimization for any $\mu \in B_\varepsilon(0)$. From Remark 4.1, there exists $\rho > 0$ such that $\|\overline{\boldsymbol{u}}_i^p(\mu)\| \leq \sqrt{\rho}\|\mu\|, \forall \mu \in B_\varepsilon(0), 0 < p \leq p^*, i \in \mathbb{I}_M$ [7]. Using the definition of the norm operator on μ and squaring, we have $\|\overline{\boldsymbol{u}}_i^p(\mu)\|^2 \leq \rho \sum_{i=1}^{M} \|x_i\|^2$. Since $\Omega_i,\ i \in \mathbb{I}_M$ is compact, a constant $\mathcal{Z} > 0$ exists satisfying $\|u_i\| \leq \sqrt{\mathcal{Z}},\ \forall\ u_i \in \Omega_i$ and all $i \in \mathbb{I}_M$. For $\|\mu\| > \varepsilon$, we have $\|u_i\| \leq \frac{\sqrt{\mathcal{Z}}}{\varepsilon}\|\mu\|$. Choose $K = \max(\rho, \frac{\mathcal{Z}}{\varepsilon^2}, \sigma) > 0$. The constant K is independent of x_i and $\|u_i^p(\mu, j)\|^2 \leq K \sum_{i=1}^{M} \|x_i\|^2,\ \forall\, i \in \mathbb{I}_M, j \geq 0$ and all $0 < p \leq p^*$.

Using stability of $A_i,\ \forall\, i \in \mathbb{I}_M$ and (Horn and Johnson, 1985, 5.6.13, p. 299), there exists $\bar{c} > 0$ and $\max_{i \in \mathbb{I}_M} \lambda_{\max}(A_i) \leq \lambda < 1$ such that $\|A_i^j\| \leq \bar{c}\lambda^j,\ \forall\, i \in \mathbb{I}_M, j \geq 0$. For any $i \in \mathbb{I}_M$

[7] Lipschitz continuity of $\overline{\boldsymbol{u}}_i^p(\mu), i \in \mathbb{I}_M$ also follows from Lemma 4.10.3. For $\mu \in B_\varepsilon(0)$, none of the input constraints are active. The requirement of linear independence of active constraints (in Lemma 4.10.3) is trivially satisfied.

and $0 \leq l \leq N$, therefore,

$$\begin{aligned}
\|x_i^{p(k)}(\mu(k),l)\| &\leq \|A_i^l\|\|x_i(k)\| + \sum_{j=0}^{l-1} \|A_i^{l-1-j}\| \Big[\|B_i\| \|u_i^{p(k)}(\mu(k),j)\| \\
&\qquad + \sum_{s\neq i} \|W_{is}\| \|u_j^{p(k)}(\mu(k),j)\| \Big] \\
&\leq \overline{c}\lambda^l \|x_i(k)\| + \sum_{j=0}^{l-1} \overline{c}\lambda^{l-1-j}\gamma\sqrt{K} \left(\sum_{i=1}^{M} \|x_i(k)\|^2 \right)^{1/2} \\
&\leq \overline{c} \left(\lambda^l + \frac{\gamma\sqrt{K}}{1-\lambda} \right) \left(\sum_{i=1}^{M} \|x_i(k)\|^2 \right)^{1/2} \\
&\leq \sqrt{\Gamma} \left(\sum_{i=1}^{M} \|x_i(k)\|^2 \right)^{1/2}
\end{aligned}$$

in which $\gamma = \max_{i\in\mathbb{I}_M} \|B_i\| + \sum_{s\neq i}^{M} \|W_{is}\|$ and $\Gamma = \overline{c}^2 \left(1 + \frac{\gamma\sqrt{K}}{1-\lambda}\right)^2$. Hence,

$$\begin{aligned}
J_N^{p(k)}(\mu(k)) &\leq \sum_{i=1}^{M} w_i \Big[\frac{1}{2} \sum_{j=0}^{N-1} \lambda_{\max}(Q_i) \|x_i^{p(k)}(\mu(k),j)\|^2 + \lambda_{\max}(R_i) \|u_i^{p(k)}(\mu(k),j)\|^2 \\
&\qquad + \frac{1}{2} \lambda_{\max}(\overline{Q}_i) \|x_i^{p(k)}(\mu(k),N)\|^2 \Big] \\
&\leq \frac{1}{2} \sum_{i=1}^{M} w_i \left[\sum_{j=0}^{N-1} \left(\lambda_{\max}(Q_i)\Gamma + \lambda_{\max}(R_i)K \right) + \lambda_{\max}(\overline{Q}_i)\Gamma \right] \sum_{i=1}^{M} \|x_i(k)\|^2 \\
&= b \sum_{i=1}^{M} \|x_i(k)\|^2
\end{aligned}$$

in which the positive constant $b = \frac{1}{2} \sum_{i=1}^{M} w_i \left[N(\lambda_{\max}(Q_i)\Gamma + \lambda_{\max}(R_i)K) + \lambda_{\max}(\overline{Q}_i)\Gamma \right]$. □

4.10.5 Lipschitz continuity of the distributed MPC control law: Unstable systems

Lemma 4.8. *Let Ω_i, $i \in \mathbb{I}_M$ be specified in terms of a collection of linear inequalities. For each $i \in \mathbb{I}_M$, consider the FC-MPC optimization problem of Equation* (4.9) *with a terminal state constraint*

$U_{u_i}{}'\left(\mathcal{C}_N(A_i,B_i)\,\overline{\boldsymbol{u}}_i + A_i^N x_i(k)\right) = 0$. Let $B_\varepsilon(0),\ \varepsilon > 0$ be defined such that the input inequality constraints in each FC-MPC optimization problem and initialization QP (Equation (4.14)) remain inactive for $\mu \in B_\varepsilon(0)$. Let Assumptions 4.1, 4.4 and 4.7 hold. The input trajectory $\overline{\boldsymbol{u}}_i^p(\mu),\ i \in \mathbb{I}_M$ generated by Algorithm 4.1 is a Lipschitz continuous function of μ for all $p \in \mathbb{I}_+, p \leq p^*$.

Proof. Since $0 \in \text{int}(\Omega_1 \times \cdots \times \Omega_M)$ and the distributed MPC control law is stable and attractive, an $\varepsilon > 0$ exists. The main difference between the proof for this lemma and the proof for Lemma 4.7 is showing that the initialization using the solution to the QP of Equation (4.14) (at initial time *i.e.*, $k = 0$) is Lipschitz continuous in the initial subsystem state x_i for $\mu \in B_\varepsilon(0)$. From Assumptions 4.1 and 4.4, $(A_i, B_i),\ i \in \mathbb{I}_M$ is stabilizable. Because (A_i, B_i) is stabilizable and U_{u_i} is obtained from a Schur decomposition, the rows of $U_{u_i}{}'\mathcal{C}_N(A_i, B_i)$ are independent (hence, the active constraints are independent).

Consider two sets of initial subsystem states $\mu(0), \zeta(0) \in \mathbb{D}_C$ containing the initial subsystem states $x_i(0)$ and $z_i(0)$, $i \in \mathbb{I}_M$, respectively. Let the solution to the initialization QP (Equation (4.14)) for the two initial states be $\overline{\boldsymbol{v}}_i^*(x_i(0))$ and $\overline{\boldsymbol{v}}_i^*(z_i(0))$, respectively. From (Hager, 1979, Theorem 3.1), $\exists\ \rho < \infty$ satisfying $\|\overline{\boldsymbol{v}}_i^*(x_i(0)) - \overline{\boldsymbol{v}}_i^*(z_i(0))\| \leq \rho\|x_i(0) - z_i(0)\|$. We have $\overline{\boldsymbol{u}}_i^0(\cdot) = \overline{\boldsymbol{v}}_i^*(x_i(0)),\ i \in \mathbb{I}_M$. This gives $\|\overline{\boldsymbol{u}}_i^0(\mu(0)) - \overline{\boldsymbol{u}}_i^0(\zeta(0))\| = \|\overline{\boldsymbol{v}}_i^*(x_i(0)) - \overline{\boldsymbol{v}}_i^*(z_i(0))\| \leq \rho\|x_i(0) - z_i(0)\| \leq \rho\|\mu(0) - \zeta(0)\|, \forall\ i \in \mathbb{I}_M$. Also, by definition $\boldsymbol{u}_i^0 = [\overline{\boldsymbol{u}}_i^{0\prime}, 0, \ldots]'$. The remainder of the proof follows the arguments in the proof for Lemma 4.7. □

4.10.6 Proof for Theorem 4.2

Proof. To show exponential stability for the set of initial subsystem states $\mu(k) \in \mathbb{D}_C$, we need to show that there exists positive constants a, b, c satisfying Equations (4.25a) and (4.25b). Determination of constants a and c closely follows the argument presented in the proof for The-

orem 4.1. To complete the proof, we need to show that a constant $b > 0$ exists such that

$$J_N^{p(k)}(\mu(k)) \leq b \sum_{i=1}^{M} \|x_i(k)\|^2$$

Let $\boldsymbol{x}_i^p = [x_i^p(\mu,1)', x_i^p(\mu,2)', \ldots]'$ denote the state trajectory for subsystem $i \in \mathbb{I}_M$ generated by the input trajectories $\boldsymbol{u}_1^p(\mu), \ldots, \boldsymbol{u}_M^p(\mu)$ obtained after $p \in \mathbb{I}_+$ (Algorithm 4.1) iterates and initial state μ. Let $x_i^{p(k)}(\mu(k),0) = x_i(k),\ i \in \mathbb{I}_M$. At time $k = 0$, we have $\mu(0) \in \mathbb{D}_C$. From the definition of the initialization QP (Equation (4.14)) and Lemma 4.1, there exists constant $K_{u_i} > 0$ independent of x_i such that the N input sequence $\widetilde{\boldsymbol{u}}_i(0)' = [\widetilde{u}_i(0|0)', \widetilde{u}_i(1|0)', \ldots, \widetilde{u}_i(N-1|0)']$ obtained as the solution to Equation (4.14) satisfies $\|\widetilde{u}_i(l|0)\| \leq \sqrt{K_{u_i}}\|x_i(0)\| \leq \sqrt{K_{u_i}}\|\mu(0)\|, 0 \leq l \leq N-1,\ i \in \mathbb{I}_M$. Let $K_u = \max_{i \in \mathbb{I}_M} K_{u_i}$. Since $0 \in \mathrm{int}(\Omega_1 \times \ldots \times \Omega_M)$ and the origin is Lyapunov stable and attractive with the cost relationship given in Section 4.6.2, there exists an $\varepsilon_1 > 0$ such that all the input inequality constraints in the FC-MPC optimization for each subsystem remain inactive for any $\mu \in B_{\varepsilon_1}(0)$. Similarly, choose $\varepsilon_2 > 0$ such that the minimum l_2 norm solution is feasible (and hence optimal) for the initialization QP (Equation (4.14)) for all $i \in \mathbb{I}_M$ and any $\mu \in B_{\varepsilon_2}(0)$. Feasibility of the minimum l_2 norm solution implies none of the input inequality constraints are active. Pick $\varepsilon = \min(\varepsilon_1, \varepsilon_2) > 0$. For $\mu \in B_\varepsilon(0)$, the only active constraint, for each subsystem $i \in \mathbb{I}_M$, is the terminal equality constraint ${U_{u_i}}' \left(\mathcal{C}_N(A_i, B_i)\, \overline{\boldsymbol{u}}_i + A_i^N x_i(k) \right) = 0$. In the above constraint, U_{u_i} is obtained from a Schur decomposition of A_i and (A_i, B_i) is stabilizable. The rows of ${U_{u_i}}'\mathcal{C}_N(A_i, B_i)$ are, therefore, linearly independent. From Lemma 4.8, $\boldsymbol{u}_i(\cdot)$ is Lipschitz continuous in μ for $\mu \in B_\varepsilon(0)$. Hence, there exists $\rho > 0$ such that $\|\boldsymbol{u}_i^p(\mu)\|^2 \leq \rho \sum_{i=1}^{M} \|x_i\|^2$,

$\forall\, 0 < p \leq p^*$, $i \in \mathbb{I}_M$ [8]. Using arguments identical to those described in the proof for Theorem 4.1, we have $K = \max(\rho, \frac{\mathcal{Z}}{\varepsilon}^2, K_u)$ where $K > 0$ and independent of $x_i, i \in \mathbb{I}_M$ such that $\|u_i^p(\mu, j)\|^2 \leq K \sum_{i=1}^M \|x_i\|^2$, $\forall\, i \in \mathbb{I}_M$, $j \geq 0$, and all $0 < p \leq p^*$.

Define $\mathcal{A}_i = \max_{0 \leq j \leq N} \|A_i^j\|$. For any $i \in \mathbb{I}_M$ and $0 \leq l \leq N$

$$
\begin{aligned}
\|x_i^{p(k)}(\mu(k), l)\| &= \|A_i^l x_i^{p(k)}(\mu(k), 0) + \sum_{j=0}^{l-1} A_i^{l-1-j} B_i u_i^{p(k)}(\mu(k), j) \\
&\qquad + \sum_{s \neq i} \sum_{j=0}^{l-1} A_i^{l-1-j} W_{is} u_s^{p(k)}(\mu(k), j)\| \\
&\leq \|A_i^l\| \|x_i(k)\| + \sum_{j=0}^{l-1} \|A_i^{l-1-j}\| \left[\|B_i\| + \sum_{s \neq i} \|W_{is}\| \right] \sqrt{K} \left(\sum_{i=1}^M \|x_i(k)\|^2 \right)^{1/2} \\
&\leq \mathcal{A}_i \|x_i(k)\| + \mathcal{A}_i \sum_{j=0}^{l-1} \gamma \sqrt{K} \left(\sum_{i=1}^M \|x_i(k)\|^2 \right)^{1/2} \\
&\leq \mathcal{A}_i \left(1 + \gamma N \sqrt{K} \right) \left(\sum_{i=1}^M \|x_i(k)\|^2 \right)^{1/2} \\
&= \sqrt{\Gamma_i} \left(\sum_{i=1}^M \|x_i(k)\|^2 \right)^{1/2}
\end{aligned} \tag{4.26}
$$

in which $\gamma = \max_{i \in \mathbb{I}_M} \|B_i\| + \sum_{s \neq i}^M \|W_{is}\|$ and $\Gamma_i = \mathcal{A}_i \left(1 + \gamma N \sqrt{K} \right)^2$.

Define Σ_i to be the solution to the Lyapunov equation ${A_{s_i}}' \Sigma_i A_{s_i} - \Sigma_i = -{U_{s_i}}' Q_i U_{s_i}$ and let $\overline{Q}_i = U_{s_i} \Sigma_i {U_{s_i}}'$. The infinite sum $\sum_{i=1}^M w_i \sum_{j=N}^\infty L_i(A_i^l x_i^{p(k)}(\mu(k), j), 0)$ subject to the terminal state constraint ${U_{u_i}}' x_i^{p(k)}(\mu(k), N) = 0$ is equal to $\sum_{i=1}^M w_i \frac{1}{2} x_i^{p(k)}(\mu(k), N)' \overline{Q}_i x_i^{p(k)}(\mu(k), N)$.

[8] The details are available in the proof for Theorem 4.1 and are, therefore, omitted.

$$
\begin{aligned}
J_N^{p(k)}(\mu(k)) &= \sum_{i=1}^{M} w_i \Bigg[\sum_{j=0}^{N-1} L_i \left(x_i^{p(k)}(\mu(k), j), u_i^{p(k)}(\mu(k), j) \right) \\
&\qquad + \frac{1}{2} x_i^{p(k)}(\mu(k), N)' \overline{Q}_i x_i^{p(k)}(\mu(k), N) \Bigg] \\
&\leq \sum_{i=1}^{M} w_i \Bigg[\frac{1}{2} \sum_{j=0}^{N-1} \lambda_{\max}(Q_i) \| x_i^{p(k)}(\mu(k), j) \|^2 + \lambda_{\max}(R_i) \| u_i^{p(k)}(\mu(k), j) \|^2 \\
&\qquad + \frac{1}{2} \lambda_{\max}(\overline{Q}_i) \| x_i^{p(k)}(\mu(k), N) \|^2 \Bigg] \\
&\leq \frac{1}{2} \sum_{i=1}^{M} w_i \left[\sum_{j=0}^{N-1} (\lambda_{\max}(Q_i) \Gamma_i + \lambda_{\max}(R_i) K) + \lambda_{\max}(\overline{Q}_i) \Gamma_i \right] \sum_{i=1}^{M} \| x_i(k) \|^2 \\
&\leq \frac{1}{2} \sum_{i=1}^{M} w_i \left[N \lambda_{\max}(Q_i) \Gamma_i + N \lambda_{\max}(R_i) K + \lambda_{\max}(\overline{Q}_i) \Gamma_i \right] \sum_{i=1}^{M} \| x_i(k) \|^2 \\
&\leq b \sum_{i=1}^{M} \| x_i(k) \|^2
\end{aligned}
$$

in which the constant b is independent of the initialization strategy used for the subsystem input trajectories and selected such that

$$
0 \leq \frac{1}{2} \sum_{i=1}^{M} w_i \left[N \lambda_{\max}(Q_i) \Gamma_i + N \lambda_{\max}(R_i) K + \lambda_{\max}(\overline{Q}_i) \Gamma_i \right] \leq b
$$

Hence, the closed-loop system is exponentially stable, as claimed. □

Chapter 5

Output feedback distributed MPC [1]

In this chapter, we consider the FC-MPC framework with distributed state estimation. Subsystem based Kalman filters are used to estimate subsystem states from local measurements. Two distributed estimation strategies are presented here. Feasibility, optimality and closed-loop stability properties for the distributed estimator-distributed regulator combination in the case of decaying estimate error are investigated. Finally, two examples are presented to illustrate the effectiveness of the proposed approach

5.1 Notation and preliminaries

The symbol $\mathbb{I}_M$ represents the set of integers $\{1, 2, \ldots, M\}$. The notation used in this chapter is consistent with the notation introduced in Chapter 4. Some additional notation is required. The $\mathrm{vec}(\cdot)$ operator defined in Section 4.2 is extended for a finite set of (compatible) matrices $Y_j \in \mathbb{R}^{s_j \times r},\ j = 1, 2, \ldots, J,\ J \geq 1$ and finite *i.e.*, $\mathrm{vec}(Y_1, Y_2, \ldots, Y_J) = \begin{bmatrix} Y_1{}' & Y_2{}' & \ldots & Y_J{}' \end{bmatrix}'$.

[1] **Portions of this chapter appear in Venkat, Rawlings, and Wright (2006e) and in Venkat, Rawlings, and Wright (2006g).**

For any matrix P, the symbol $\lambda_{\max}(P)$ denotes the maximum eigenvalue of P. Define

$$\mathcal{C}_N(A,B) = \begin{bmatrix} B & AB & \dots & A^{N-1}B \end{bmatrix},$$

in which N represents the control horizon. Let $\widehat{x}_i$ denote an estimate of the states of subsystem i. The notation $\widehat{\mu}$ denotes the set of estimated composite model (CM) states $\widehat{x}_1, \widehat{x}_2, \dots \widehat{x}_M$ *i.e.,*

$$\widehat{\mu} = [\widehat{x}_1, \widehat{x}_2, \dots, \widehat{x}_M].$$

We use $\widehat{\mu} \in \mathcal{X}$ to imply $\text{vec}(\widehat{\mu}) = \text{vec}(\widehat{x}_1, \dots, \widehat{x}_M) \in \mathcal{X}$. The norm operator for $\widehat{\mu}$ is defined as $\|\widehat{\mu}\| = \|\text{vec}(\widehat{x}_1, \dots, \widehat{x}_M)\| = \sqrt{\sum_{i=1}^{M} \|\widehat{x}_i\|^2}$. Let e_i denote the state estimate error for subsystem $i \in \mathbb{I}_M$. Let $(A_{\text{cm}} \in \mathbb{R}^{n\times n}, B_{\text{cm}} \in \mathbb{R}^{n\times m}, C_{\text{cm}} \in \mathbb{R}^{n_y\times n})$ denote the A, B and C matrices respectively for the composite model (CM) for the entire plant (see Chapter 4, Equation (4.2)) with $n = \sum_{i=1}^{M} n_i$, $m = \sum_{i=1}^{M} m_i$ and $n_y = \sum_{i=1}^{M} n_{y_i}$. It is assumed that $(A_{\text{cm}}, B_{\text{cm}})$ is stabilizable and $(A_{\text{cm}}, C_{\text{cm}})$ is detectable. For each (A_{ij}, B_{ij}, C_{ij}) in the CM for subsystem i, we have $A_{ij} \in \mathbb{R}^{n_{ij}\times n_{ij}}$, $B_{ij} \in \mathbb{R}^{n_{ij}\times m_j}$, $C_{ij} \in \mathbb{R}^{n_{y_i}\times n_{ij}}$ with $n_i = \sum_{j=1}^{M} n_{ij}$.

Consider the nonminimal, LTI system (A_m, B_m, C_m, G_m) defined as

$$x(k+1) = A_m x(k) + B_m u(k) + G_m w(k) \tag{5.1a}$$

$$y(k) = C_m x(k) + \nu(k) \tag{5.1b}$$

in which $A_m \in \mathbb{R}^{n\times n}, B_m \in \mathbb{R}^{n\times m}, C_m \in \mathbb{R}^{n_y\times n}, G_m \in \mathbb{R}^{n\times g}$. The terms $w(k)$ and $\nu(k)$ are zero-mean disturbances, with covariances Q_x, R_v respectively, that affect the state and output equation. It is assumed that (A_m, B_m) is stabilizable and (A_m, C_m) is detectable. The

noise shaping matrix G_m and the noise covariances Q_x, R_v are usually unknown and have to be estimated from process data. Let T be a similarity transform for the LTI system and let $(\widetilde{A}_m, \widetilde{B}_m, \widetilde{C}_m, \widetilde{G}_m) = (TA_mT^{-1}, TB_m, C_mT^{-1}, TG_m)$ be the transformed system.

Lemma 5.1. *Detectability (and hence stabilizability) is invariant under a similarity transformation.*

This result follows from (Chen, 1999, Theorems 5.15 and 5.16, p. 200). An alternate proof is given in Appendix 5.7.1.

5.2 State estimation for FC-MPC

Assumption 5.1. All interaction models are stable *i.e.*, for each $i, j \in \mathbb{I}_M$, $|\lambda_{\max}(A_{ij})| < 1, \forall j \neq i$.

In the context of FC-MPC, the goal of distributed estimation is to ascertain the states of each subsystem (decentralized and interaction) from local measurements. Observability for the subsystem CM follows from Lemma 5.2, which is a mere restatement of the Hautus Lemma (Sontag, 1998, p. 272).

Lemma 5.2 (Observability)**.** *For each subsystem $i \in \mathbb{I}_M$, (A_i, C_i) is observable if*

$$\operatorname{rank} \begin{bmatrix} \lambda I - A_i \\ C_i \end{bmatrix} = n_i, \ \forall \, \lambda = \lambda(A_i)$$

A sufficient condition for CM observability is stated below.

Corollary 5.2.1 (CM observability)**.** *For subsystem i, let $(A_{ij}, C_{ij}), \ \forall \ j \in \mathbb{I}_M$ be observable. If $\lambda(A_{i1}) \cap \lambda(A_{i2}) \cap \ldots \cap \lambda(A_{iM}) = \varnothing$, the CM (A_i, C_i) is observable.*

Proof. For any $\lambda \in \lambda(A_i)$, the conditions of the corollary imply, $\lambda \in \lambda(A_{ij})$ for some unique $\overline{j} \in \mathbb{I}_M$. If the columns of $\begin{bmatrix} \lambda I - A_i \\ C_i \end{bmatrix}$ are not independent, we contradict observability of $(A_{i\overline{j}}, C_{i\overline{j}})$, which proves the corollary. □

Observability is usually a stronger than necessary requirement. The following lemma provides a necessary and sufficient condition for detectability of each subsystem CM.

Lemma 5.3 (Detectability)**.** *Let Assumption 5.1 hold. The CM (A_i, C_i) is detectable if and only if (A_{ii}, C_{ii}) is detectable.*

A proof is given in Appendix 5.8.1.

Lemma 5.4. *Let the noise shaping matrix, G_{cm}, for the plant CM (see Equation (4.2)) be defined as $G_{\mathrm{CM}} = \begin{bmatrix} G_1{}' & G_2{}' & \dots & G_M{}' \end{bmatrix}'$. If $(A_{\mathrm{cm}}, G_{\mathrm{cm}})$ is stabilizable then (A_i, G_i), $\forall\, i \in \mathbb{I}_M$ is stabilizable.*

A proof is given in Appendix 5.8.2.

5.2.1 Method 1. Distributed estimation with subsystem-based noise shaping matrices

We consider first, a distributed estimation framework in which the noise shaping matrix $G_i \in \mathbb{R}^{n_i \times g_i}$ and noise covariances Q_{x_i}, R_{v_i} are estimated locally for each $i \in \mathbb{I}_M$. The steady-state subsystem-based Kalman filters designed subsequently, require only local measurements. For each subsystem $i \in \mathbb{I}_M$, let

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j \neq i}^{M} W_{ij} u_j(k) + G_i w_{x_i}(k), \qquad w_{x_i}(k) \sim N(0, Q_{x_i}) \tag{5.2a}$$

$$y_i(k) = C_i x_i(k) + \nu_i(k), \qquad \nu_i(k) \sim N(0, R_{v_i}) \tag{5.2b}$$

denote the CM employed by each subsystem's Kalman filter, in which $w_{x_i} \sim N(0, Q_{x_i}) \in \mathbb{R}^{g_i}$ and $\nu_i \sim N(0, R_{v_i}) \in \mathbb{R}^{n_{y_i}}$ represent zero-mean white noise disturbances affecting the CM state equation and output equation, respectively.

Assumption 5.2. For each $i \in \mathbb{I}_M$, (A_{ii}, C_{ii}) is detectable.

Assumption 5.3. For each $i \in \mathbb{I}_M$, (A_{ii}, B_{ii}) is stabilizable.

For the CM $(A_i, B_i, \{W_{ij}\}_{j \neq i}, C_i, G_i)$ with (A_{ii}, C_{ii}) detectable and (A_{ii}, B_{ii}) stabilizable, Lemma 5.3 gives (A_i, C_i) is detectable and (A_i, B_i) is stabilizable. There exists a similarity transformation T_i that converts the CM for subsystem i into observability canonical form (Kailath, 1980). Let $(\widehat{A}_i, \widehat{C}_i) = (T_i A_i T_i^{-1}, C_i T_i^{-1})$ be the A and C matrices of the CM in observability canonical form, where

$$\widehat{A}_i = \begin{bmatrix} A_i^{\mathrm{o}} & 0 \\ A_i^{12} & A_i^{\bar{\mathrm{o}}} \end{bmatrix}, \; \widehat{C}_i = \begin{bmatrix} C_i^{\mathrm{o}} & 0 \end{bmatrix} \tag{5.3}$$

From Lemma 5.1, $(\widehat{A}_i, \widehat{C}_i)$ is detectable. Therefore, $A_i^{\bar{\mathrm{o}}}$, which corresponds to the unobservable partition of the subsystem CM, is stable. The observable subsystem CM is

$$x_i^o(k+1) = A_i^{\mathrm{o}} x_i^o(k) + B_i^{\mathrm{o}} u_i(k) + \sum_{j \neq i} W_{ij}^{\mathrm{o}} u_j(k) + G_i^{\mathrm{o}} w_i(k), \quad y_i(k) = C_i^{\mathrm{o}} x_i^{\mathrm{o}}(k), \quad x_i^{\mathrm{o}} \in \mathbb{R}^{n_i^o}.$$

The noise covariances $G_i^{\mathrm{o}} Q_{x_i} G_i^{\mathrm{o}\prime}$ and R_{v_i} can be determined for the observable CM above, using any of autocovariance least squares (ALS) methods available in the literature (Carew and Bélanger, 1973; Mehra, 1970, 1972; Odelson, Rajamani, and Rawlings, 2006; Sims, Lainiotis, and Magill, 1969). Here, we use the procedure described in Odelson et al. (2006). Since (A_i, C_i) is detectable, a stable estimator gain $\mathcal{L}_i$ exists and (Odelson et al., 2006, Assumptions 1 and

2) are satisfied. The closed-loop data for estimating the covariances is generated using any stable filter gain for each estimator $i \in \mathbb{I}_M$. The FC-MPC algorithm (Algorithm 4.1, p. 43) is used for regulation. Two possible scenarios arise during estimation of the noise covariances. In the first case, $G_i^{\mathrm{o}} Q_{x_i} G_i^{\mathrm{o}\prime}$ and R_{v_i} can be estimated uniquely. A necessary and sufficient rank condition under which the ALS procedure gives unique estimates is given in (Odelson et al., 2006, Lemma 4). For the observable subsystem model $(A_i^{\mathrm{o}}, C_i^{\mathrm{o}})$ used in ALS estimation, unique estimates of $G_i^{\mathrm{o}} Q_{x_i} G_i^{\mathrm{o}\prime}$ and R_{v_i} are obtained only if $n_{y_i} \geq n_i^o$. For the case $n_{y_i} < n_i^o$, the estimates of $G_i^{\mathrm{o}} Q_{x_i} G_i^{\mathrm{o}\prime}$ and R_{v_i} are not unique. In this case, several choices for disturbance covariances that generate the same output data exist. One may choose any solution to the constrained ALS estimation problem (Odelson et al., 2006, Equation 13, p. 307) to calculate the estimator gain. Let $\mathcal{G}_i^{\mathrm{o}} \mathcal{Q}_{x_i} \mathcal{G}_i^{\mathrm{o}\prime}$ and $\mathcal{R}_{v_i}$ represent a solution to the constrained ALS estimation problem of (Odelson et al., 2006, Equation 13, p. 307) for subsystem $i \in \mathbb{I}_M$. A possible choice for the noise shaping matrix and the noise covariances is $G_i^{\mathrm{o}} \leftarrow I_{n_i}$, $Q_{x_i} \leftarrow \mathcal{G}_i^{\mathrm{o}} \mathcal{Q}_{x_i} \mathcal{G}_i^{\mathrm{o}\prime}$ and $R_{v_i} \leftarrow \mathcal{R}_{v_i}$. Another choice is $G_i^{\mathrm{o}} \leftarrow \mathcal{G}_i^{\mathrm{o}} \sqrt{\mathcal{Q}_{x_i}}$, $Q_{x_i} \leftarrow I_{g_i}$ and $R_{v_i} = \mathcal{R}_{v_i}$.

Lemma 5.5. *Let Assumptions 5.1 and 5.2 be satisfied. Define* $\widehat{G}_i = \begin{bmatrix} G_i^{\mathrm{o}} \\ 0 \end{bmatrix}$. $(\widehat{A}_i, \widehat{G}_i)$ *is stabilizable if and only if* $(A_i^{\mathrm{o}}, G_i^{\mathrm{o}})$ *is stabilizable.*

A proof is given in Appendix 5.8.3.

Define $G_i = T_i^{-1} \widehat{G}_i$. From Lemma 5.1, (A_i, G_i) is stabilizable if and only if $(\widehat{A}_i, \widehat{G}_i)$ is stabilizable.

Corollary 5.5.1. *Let Assumptions 5.1 and 5.2 be satisfied. Let* $\widehat{G}_i = \begin{bmatrix} G_i^{\mathrm{o}} \\ 0 \end{bmatrix}$. $(\widehat{A}_i, \widehat{G}_i Q_{x_i}^{1/2})$ *is stabilizable if and only if* $(A_i^{\mathrm{o}}, G_i^{\mathrm{o}} Q_{x_i}^{1/2})$ *is stabilizable.*

The proof for Corollary 5.5.1 is similar to the proof for Lemma 5.5 and is omitted.

Remark 5.1. For each subsystem $i \in \mathbb{I}_M$, the conditions for the existence of a stable, steady-state Kalman filter are identical to those described in (Poubelle, Bitmead, and Gevers, 1988, Theorem 1) for a single (centralized) Kalman filter. Thus, if $R_{v_i} > 0$, $Q_{x_i} \geq 0$, (A_i, C_i) is detectable and $(A_i, G_i Q_{x_i}^{1/2})$ is stabilizable, the steady-state Kalman filter for subsystem i exists and is a stable estimator. If $Q_{x_i} > 0$, the requirements for stability of the steady-state Kalman filter reduce to $R_{v_i} > 0$, (A_i, C_i) is detectable and (A_i, G_i) is stabilizable (Bertsekas, 1987).

Remark 5.2. The steady-state estimate error covariance for subsystem $i \in \mathbb{I}_M$, P_i, is the solution to the algebraic Riccati equation

$$P_i = G_i Q_{x_i} G_i{}' + A_i P_i A_i{}' - A_i P_i C_i{}' \left(R_{v_i} + C_i P_i C_i{}'\right)^{-1} C_i P_i A_i{}'.$$

The steady-state Kalman filter gain $\mathcal{L}_i$ for subsystem i is calculated as,

$$\mathcal{L}_i = P_i C_i{}' \left(R_{v_i} + C_i P_i C_i{}'\right)^{-1}.$$

Under the conditions of Remark 5.1, we have $|\lambda_{\max}(A_i - A_i \mathcal{L}_i C_i)| < 1$.

In this distributed estimation framework, the noise shaping matrix and noise covariances for each subsystem are identified using local process data. The estimators are decoupled, stable, and require only local measurement information. The estimates generated by each local Kalman filter may not be optimal, however.

5.2.2 Method 2. Distributed estimation with interconnected noise shaping matrices

In this estimation framework, the model used by each subsystem-based Kalman filter is

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j \neq i}^{M} W_{ij} u_j(k) + G_{ii} w_{x_i}(k) + \sum_{j \neq i}^{M} G_{ij} w_{x_j}(k)$$

$$y_i(k) = C_i x_i(k) + \nu_i(k),$$

in which $w_x{}' = [w_{x_1}{}', w_{x_2}{}', \ldots, w_{x_M}{}']$ and $\nu' = [\nu_1{}', \nu_2{}', \ldots, \nu_M{}']$ denote zero mean white noise disturbances affecting each CM state equation and output equation respectively, with $w_x(k) \sim N(0, \mathbb{Q}_x)$, $\nu(k) \sim N(0, \mathbb{R}_v)$.

There exists a similarity transformation $T_{\rm kf}$ that converts the plant CM $(A_{\rm cm}, B_{\rm cm}, C_{\rm cm})$ into the LTI system $(A_{\rm kf}, B_{\rm kf}, C_{\rm kf})$ in Kalman decomposition form (Kailath, 1980), in which

$$A_{\rm kf} = \begin{bmatrix} A_{\rm co} & 0 & A_{13} & 0 \\ A_{21} & A_{\rm c\bar{o}} & A_{23} & A_{24} \\ 0 & 0 & A_{\rm \bar{c}o} & 0 \\ 0 & 0 & A_{43} & A_{\rm \bar{c}\bar{o}} \end{bmatrix}, \; B_{\rm kf} = \begin{bmatrix} B_{\rm co} \\ B_{\rm c\bar{o}} \\ 0 \\ 0 \end{bmatrix}, \; C_{\rm kf} = \begin{bmatrix} C_{\rm co} & 0 & C_{\rm \bar{c}o} & 0 \end{bmatrix}. \tag{5.4}$$

Let $G_{\rm kf} = \begin{bmatrix} G_{\rm co}{}' & G_{\rm c\bar{o}}{}' & G_{\rm \bar{c}o}{}' & G_{\rm \bar{c}\bar{o}}{}' \end{bmatrix}'$ be the noise shaping matrix for the overall plant CM in Kalman decomposition form. The matrix $G_{\rm kf}$, is unknown and has to be determined from available closed-loop data. Using Lemma 5.1, we know $(A_{\rm kf}, C_{\rm kf})$ is detectable and $(A_{\rm kf}, B_{\rm kf})$ is stabilizable. Therefore, $A_{\rm c\bar{o}}, A_{\rm \bar{c}o}$ and $A_{\rm \bar{c}\bar{o}}$, which correspond to the uncontrollable and/or unobservable modes are stable matrices. Using any autocovariance least squares (ALS) tech-

niques the plant noise covariances $G_{\text{co}}\mathbb{Q}_x{G_{\text{co}}}'$ and $\mathbb{R}_v$ can be determined using closed-loop data for the minimal plant CM ($A_{\text{co}} \in \mathbb{R}^{n_{\text{co}} \times n_{\text{co}}}, B_{\text{co}} \in \mathbb{R}^{n_{\text{co}} \times m}, C_{\text{co}} \in \mathbb{R}^{n_y \times n_{\text{co}}}, G_{\text{co}} \times \mathbb{R}^{n_{\text{co}} \times g_{\text{co}}}$). The closed-loop data for ALS estimation is generated by using any stable filter gain for each subsystem-based estimator and the FC-MPC framework of Section 4.5 for each regulator. In fact, the estimators designed using Section 5.2.1 may be used to generate closed-loop data for this estimation framework. Similar to the scenario in Section 5.2.1, two cases arise here too. In the first case, $G_{\text{co}}\mathbb{Q}_x{G_{\text{co}}}'$ and $\mathbb{R}_v$ can be uniquely determined from closed-loop data by solving the constrained ALS estimation problem (Odelson et al., 2006, Equation 13). This scenario corresponds to the case $n_y \geq n_{\text{co}}$. For $n_y < n_{\text{co}}$, the solution the constrained ALS estimation problem is not unique. In this case, a solution to the ALS estimation problem is used. Let $\mathcal{G}_{\text{co}}\mathcal{Q}_x{\mathcal{G}_{\text{co}}}', \mathcal{R}_v$ denote a solution to the constrained ALS estimation problem (Odelson et al., 2006, Equation 13). Two possible choices for the covariances and noise shaping matrix are $G_{\text{co}} \leftarrow I_{n_{\text{co}}}$, $\mathcal{Q}_x \leftarrow \mathcal{G}_{\text{co}}\mathcal{Q}_x{\mathcal{G}_{\text{co}}}'$, $\mathbb{R}_v \leftarrow \mathcal{R}_v$ and $G_{\text{co}} \leftarrow \mathcal{G}_{\text{co}}\sqrt{\mathcal{Q}_x}$, $\mathbb{Q}_x \leftarrow I_{g_{\text{co}}}$, $\mathbb{R}_v = \mathcal{R}_v$.

Lemma 5.6. *Let* $(A_{\text{co}}, G_{\text{co}})$ *be stabilizable. If* $G_{\text{kf}}' = \left[G_{\text{co}}', 0, 0, 0\right]$, *then* $(A_{\text{kf}}, G_{\text{kf}})$ *is stabilizable.*

The proof for Lemma 5.6 is similar to the proof for Lemma 5.5, and is omitted for brevity.

By definition, $G_{\text{cm}} = \begin{bmatrix} {G_1}' & {G_2}' & \dots & {G_M}' \end{bmatrix}' = T_{\text{kf}}^{-1} G_{\text{kf}}$. Lemmas 5.1 and 5.6 give $(A_{\text{cm}}, G_{\text{cm}})$ is stabilizable. From Lemma 5.4, we know $(A_i, G_i),\ \forall\ i \in \mathbb{I}_M$ is stabilizable. Define $G_i = [G_{i1}, G_{i2}, \dots, G_{iM}]$.

Remark 5.3. For (A_i, C_i) detectable and $(A_i, G_i\mathbb{Q}_x^{1/2})$ stabilizable, the steady-state error covariance $\overline{P}_i$, for subsystem $i \in \mathbb{I}_M$ is the solution to the algebraic Riccati equation

$$\overline{P}_i = G_i\mathbb{Q}_x{G_i}' + A_i\overline{P}_i{A_i}' - A_i\overline{P}_i{C_i}'\left(R_{v_i} + C_i\overline{P}_i{C_i}'\right)^{-1} C_i\overline{P}_i{A_i}'$$

The steady-state Kalman filter gain is

$$\mathcal{L}_i = \overline{P}_i C_i' \left(R_{v_i} + C_i \overline{P}_i C_i'\right)^{-1},\ i \in \mathbb{I}_M$$

Under the conditions specified in Remark 5.1 (with Q_{x_i} replaced by $\mathbb{Q}_x$), the steady-state Kalman filter for subsystem i is a stable estimator.

This distributed estimation framework described here is suboptimal, but admits a wider class of structures for the noise shaping matrix as compared to the framework described in Section 5.2.1. A systemwide computation of the noise covariances is required however. In fact the estimation framework of Section 5.2.1 is a special case, in which each $G_{ij} = 0,\ j \neq i,\ \forall\, i \in \mathbb{I}_M$ and each off-diagonal block of $\mathbb{Q}_x$ is a zero submatrix.

5.3 Output feedback FC-MPC for distributed regulation

For the set of estimated states $\widehat{\mu}$, let $\Phi(\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p; \widehat{\mu}) = \sum_{r=1}^{M} w_r \Phi_r(\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p; \widehat{x}_r)$ represent the value of the cooperation-based cost function after p (Algorithm 4.1) iterates. We assume the following:

Assumption 5.4. For Algorithm 4.1, $p_{\max}(k) \equiv p_{\max} = p^*,\ k \geq 0$, $p_{\max} \in \mathbb{I}_+$ and $0 < p_{\max} < \infty$.

Assumption 5.5. $N \geq \max(\alpha, 1)$, in which $\alpha = \max(\alpha_1, \ldots, \alpha_M)$ and $\alpha_i \geq 0$ denotes the number of unstable modes for subsystem $i \in \mathbb{I}_M$.

Assumption 5.6. $Q_i(0) = Q_i(1) = \ldots = Q_i(N-1) = Q_i > 0$ and $R_i(0) = R_i(1) = \cdots = R_i(N-1) = R_i > 0,\ \forall\, i \in \mathbb{I}_M$.

Assumption 5.7. For each $i \in \mathbb{I}_M$, $(A_i - A_i \mathcal{L}_i C_i)$ is stable.

The evolution of the estimate error is given by $e_i(k+1) = (A_i - A_i\mathcal{L}_i C_i)e_i(k)$, in which $e_i(k)$ is the state estimate error for subsystem $i \in \mathbb{I}_M$ at time k. From Assumption 5.7 and Equation (5.12) (see Appendix 5.9.1), $Z_i = \mathcal{L}_i C_i(A_i - A_i\mathcal{L}_i C_i), i \in \mathbb{I}_M$.

FC-MPC control law under output feedback At time k and set of estimated subsystem states $\widehat{\mu}(k)$, let the FC-MPC algorithm (Algorithm 4.1) be terminated after $p(k) = q \geq 1$ cooperation-based iterates. Let

$$\boldsymbol{u}_i^q(\widehat{\mu}(k)) = \left[u_i^q(\widehat{\mu}(k),0)', u_i^q(\widehat{\mu}(k),1)', \ldots, u_i^q(\widehat{\mu}(k),N-1)',\ 0,\ 0, \ldots\right]',\ \forall\, i \in \mathbb{I}_M$$

represent the solution to Algorithm 4.1 after q iterates. The input injected into subsystem $i \in \mathbb{I}_M$ is $u_i^q(\widehat{\mu}(k), 0)$. Let

$$\boldsymbol{u}_i^+(\widehat{\mu}(k)) = [u_i^p(\widehat{\mu}(k),1)', \ldots, u_i^p(\widehat{\mu}(k),N-1)', 0, 0, \ldots]' \tag{5.5}$$

represent a shifted version of $\boldsymbol{u}_i^p(\widehat{\mu}(k)), i \in \mathbb{I}_M$.

5.3.1 Perturbed stability of systems with stable decentralized modes

Initialization. At time 0, each MPC is initialized with the zero input trajectory $u_i(j|0) = 0,\ 0 \leq j, \forall\, i \in \mathbb{I}_M$. At time $k+1$, the initial input trajectory for each subsystem's MPC is $\boldsymbol{u}_i^0(k+1) = \boldsymbol{u}_i^+(\widehat{\mu}(k)),\ i \in \mathbb{I}_M$. The cost function value for the set of feasible initial subsystem input trajectories at $k+1$ is $J_N^0(\widehat{\mu}(k+1)) = \Phi\left(\boldsymbol{u}_1^0(k+1), \boldsymbol{u}_2^0(k+1), \ldots, \boldsymbol{u}_M^0(k+1); \widehat{\mu}(k+1)\right)$.

Feasibility and domain of attraction. For $(\widehat{x}_{ii}(0), e_i(0)) \in \mathbb{R}^{n_{ii}} \times \mathbb{R}^{n_i}$, the zero input trajectory is feasible for each $i \in \mathbb{I}_M$. Existence of a feasible input trajectory for each $i \in \mathbb{I}_M$ at $k = 0$ and $p(0) = 0$ guarantees feasibility of $\mathcal{F}_i$, $\forall\, i \in \mathbb{I}_M$ (Equation (4.9), p. 42) at all $k \geq 0$, $p(k) > 0$. This result follows from the initialization procedure, convexity of $\Omega_i, \forall\, i \in \mathbb{I}_M$ and Algorithm 4.1. The controllable domain for the nominal closed-loop system is $\mathbb{R}^n \times \mathbb{R}^n$.

Assumption 5.8. For each $i \in \mathbb{I}_M$, A_{ii} is stable, $\mathbb{Q}_i = \text{diag}\left(Q_i(1), \ldots, Q_i(N-1), \overline{Q}_i\right)$, in which $\overline{Q}_i$ is the solution of the Lyapunov equation $A_i{}'\overline{Q}_i A_i - \overline{Q}_i = -Q_i$

Exponential stability for the closed-loop system under the output feedback distributed MPC control law is stated in the following theorem, which requires that the local estimators are exponentially stable but makes no assumptions on the optimality of the estimates.

Theorem 5.1 (Stable modes)**.** *Consider Algorithm 4.1 employing the FC-MPC optimization problem of Equation (4.9). Let Assumptions 5.1 to 5.8 hold. The origin is an exponentially stable equilibrium for the perturbed closed-loop system*

$$\widehat{x}_i(k+1) = A_i\widehat{x}_i(k) + B_i u_i^p(\widehat{\mu}(k), 0) + \sum_{j \neq i} W_{ij} u_j^p(\widehat{\mu}(k), 0) + Z_i e_i,$$

$$e_i(k+1) = (A_i - A_i\mathcal{L}_i C_i)e_i(k), \qquad i \in \mathbb{I}_M,$$

for all $((\widehat{x}_i(0), e_i(0)),\ i \in \mathbb{I}_M) \in \mathbb{R}^n \times \mathbb{R}^n$ *and all* $p = 1, 2, \ldots, p_{\max}$.

The proof is given in Appendix 5.9.3.

5.3.2 Perturbed closed-loop stability for systems with unstable decentralized modes

For systems with unstable modes, a terminal state constraint that forces the unstable modes to the origin at the end of the control horizon is employed in each FC-MPC optimization problem (Equation (4.9)). This terminal state constraint is necessary for stability. From Assumption 5.1, unstable modes, if any, are present only in the decentralized model. We have, therefore, that ${U_{u_i}}'\widehat{x}_i = {S_{u_i}}'\widehat{x}_{ii}, i \in \mathbb{I}_M$, in which U_{u_i} and S_{u_i}, are obtained through a Schur decomposition of A_i and A_{ii} respectively [2]. For $i \in \mathbb{I}_M$, define,

$$\mathbb{S}_i = \{x_{ii} \,|\, \exists\, \overline{\boldsymbol{u}}_i \in \mathcal{U}_i \text{ such that } {S_{u_i}}'[\mathcal{C}_N(A_{ii}, B_{ii})\, \overline{\boldsymbol{u}}_i + A_{ii}^N x_{ii}] = 0\} \qquad \text{steerable set}$$

to be the set of unstable decentralized modes that can be steered to zero in N moves. For stable systems, $\mathbb{S}_i = \mathbb{R}^{n_{ii}},\ i \in \mathbb{I}_M$. By definition, ${U_{u_i}}'[\mathcal{C}_N(A_i, B_i)\, \overline{\boldsymbol{u}}_i + A_i^N \widehat{x}_i] = {S_{u_i}}'[\mathcal{C}_N(A_{ii}, B_{ii})\, \overline{\boldsymbol{u}}_i + A_{ii}^N \widehat{x}_{ii}]$. From Assumption 5.1 and because the domain of each $x_{ij}, i, j \in \mathbb{I}_M, j \neq i$ is $\mathbb{R}^{n_{ij}}$,

$$\mathbb{D}_{R_i} = \mathbb{R}^{n_{i1}} \times \cdots \times \mathbb{R}^{n_{i(i-1)}} \times \mathbb{S}_i \times \mathbb{R}^{n_{i(i+1)}} \times \cdots \times \mathbb{R}^{n_{iM}} \subseteq \mathbb{R}^{n_i},\ i \in \mathbb{I}_M, \qquad \text{domain of regulator}$$

represents the set of all x_i for which an admissible input trajectory $\overline{\boldsymbol{u}}_i$ exists that drives the unstable decentralized modes ${U_{u_i}}'x_i$ to the origin. A positively invariant set, $\mathbb{D}_C$, for the perturbed closed-loop system

$$\widehat{x}_i^+ = A_i\widehat{x}_i + B_i u_i^p(\widehat{\mu}, 0) + \sum_{j \neq i}^{M} W_{ij} u_j^p(\widehat{\mu}, 0) + Z_i e_i, \quad e_i^+ = (A_i - A_i \mathcal{L}_i C_i) e_i, \quad i \in \mathbb{I}_M \qquad (5.6)$$

[2] The Schur decomposition of $A_{ii} = \begin{bmatrix} S_{s_i} & S_{u_i} \end{bmatrix} \begin{bmatrix} A_{s_{ii}} & \bigoplus \\ & A_{u_{ii}} \end{bmatrix} \begin{bmatrix} {S_{s_i}}' \\ {S_{u_i}}' \end{bmatrix}$, $A_i = \begin{bmatrix} U_{s_i} & U_{u_i} \end{bmatrix} \begin{bmatrix} A_{s_i} & \bigotimes \\ & A_{u_i} \end{bmatrix} \begin{bmatrix} {U_{s_i}}' \\ {U_{u_i}}' \end{bmatrix}$. Eigenvalues of $A_{u_{ii}}, A_{u_i}$ are on or outside the unit circle. Eigenvalues of $A_{s_{ii}}, A_{s_i}$ are strictly inside the unit circle.

is given by

$$\mathbb{D}_C = \{((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M) \mid \left((\widehat{x}_i^+, e_i^+),\ i \in \mathbb{I}_M\right) \in \mathbb{D}_C,\ \widehat{x}_i \in \mathbb{D}_{R_i}, i \in \mathbb{I}_M\} \quad \text{domain of controller} \tag{5.7}$$

A subsystem-based procedure to construct $\mathbb{D}_C$ is described below. Let the current decentralized state for subsystem i be $\widehat{x}_{ii}$. The perturbed decentralized model for subsystem $i \in \mathbb{I}_M$ is

$$\widehat{x}_{ii}^+ = A_{ii}\widehat{x}_{ii} + B_{ii}u_i^p(\widehat{\mu}, 0) + \mathfrak{T}_{ii}Z_i e_i, \tag{5.8a}$$

$$e_i^+ = (A_i - A_i\mathcal{L}_i C_i)e_i, \tag{5.8b}$$

in which e_i is the estimate error for subsystem i, Z_i is defined in Equation (5.12) (see Appendix 5.9.1), and $\mathfrak{T}_{ii} = [0, \ldots, \underbrace{I}_{i^{\text{th}}}, \ldots, 0] \in \mathbb{R}^{n_{ii} \times n_i}$. A positively invariant set can be constructed for the system described by Equation (5.8) using any of the techniques available in the literature for backward construction of polytopic sets under state and control constraints (Blanchini, 1999; Gutman and Cwikel, 1987; Keerthi and Gilbert, 1987; Rakovic, Kerrigan, Kouramas, and Mayne, 2004). The positively invariant set $\mathbb{D}_{D_i}$ is defined as

$$\mathbb{D}_{D_i} = \{(\widehat{x}_{ii}, e_i) \,|\, (\widehat{x}_{ii}^+, e_i^+) \in \mathbb{D}_{D_i}, \quad \widehat{x}_{ii} \in \mathbb{S}_i\}, \quad \text{domain of steerable decentralized states}$$

in which $(\widehat{x}_{ii}^+, e_i^+)$ is obtained using Equation (5.8). A brief synopsis of the construction is given in Appendix 5.9.4. The positively invariant set, $\mathbb{D}_C$, for the perturbed closed-loop system

(Equation (5.6)) can also be obtained as

$$\mathbb{D}_C = \{((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M) \mid (\widehat{x}_{ii}^{+}, e_i^{+}) \in \mathbb{D}_{D_i},\ \ \widehat{x}_i \in \mathbb{D}_{R_i},\ i \in \mathbb{I}_M\}.$$

Initialization. At time 0, let $((\widehat{x}_i(0), e_i(0)),\ i \in \mathbb{I}_M) \in \mathbb{D}_C$. A feasible input trajectory, therefore, exists for each $i \in \mathbb{I}_M$, and can be computed by solving the following quadratic program (QP).

$$\overline{\boldsymbol{u}}_i^0(0) = \arg\min_{\overline{\boldsymbol{u}}_i} \quad \|\overline{\boldsymbol{u}}_i\|^2$$

subject to

$$\overline{\boldsymbol{u}}_i \in \mathcal{U}_i$$

$$U_{u_i}{}'[\mathcal{C}_N(A_i, B_i)\,\overline{\boldsymbol{u}}_i + A_i^N \widehat{x}_i(0)] = 0$$

We have $\boldsymbol{u}_i^0(0) = [\overline{\boldsymbol{u}}_i^0(0), 0, 0, \ldots\ldots]$. Let $\overline{\boldsymbol{u}}_i^{+}(\widehat{\mu}(k)) = [u_i^{p(k)}(\widehat{\mu}(k), 1)', \ldots, u_i^{p(k)}(\widehat{\mu}(k), N-1)', 0]'$ represent a shifted version of $\overline{\boldsymbol{u}}_i^{p(k)}(\widehat{\mu}(k))$. For times $k > 0$, $\overline{\boldsymbol{u}}_i^0(k) = \overline{\boldsymbol{u}}_i^{+}(\widehat{\mu}(k-1)) + \overline{\boldsymbol{v}}_i(k)$, in which $\overline{\boldsymbol{v}}_i(k) \in \mathbb{R}^{m_i N}$ is calculated by solving the following QP for each $i \in \mathbb{I}_M$.

$$\overline{\boldsymbol{v}}_i(k) = \arg\min_{\overline{\boldsymbol{v}}_i(k)} \quad \|\overline{\boldsymbol{v}}_i(k)\|^2 \tag{5.9a}$$

subject to

$$\overline{\boldsymbol{u}}_i^{+}(\widehat{\mu}(k-1)) + \overline{\boldsymbol{v}}_i(k) \in \mathcal{U}_i \tag{5.9b}$$

$$U_{u_i}^{'}[\mathcal{C}_N(A_i, B_i)\ \big(\overline{\boldsymbol{u}}_i^{+}(\widehat{\mu}(k-1), 1) + \overline{\boldsymbol{v}}_i(k)\big) + A_i^N \widehat{x}_i(k)] = 0 \tag{5.9c}$$

Since $\mathbb{D}_C$ is positively invariant (by construction) and $((\widehat{x}_i(0), e_i(0)),\ i \in \mathbb{I}_M) \in \mathbb{D}_C$, a solution to the optimization problem of Equation (5.9) exists for all $i \in \mathbb{I}_M$ and all $k > 0$. Define $\boldsymbol{v}_i(k) = [\overline{\boldsymbol{v}}_i(k)', 0, 0, \ldots]'$.

Assumption 5.9. $\alpha > 0$ (see Assumption 5.5).

For each $i \in \mathbb{I}_M$, $\mathbb{Q}_i = \text{diag}\left(Q_i(1), \ldots, Q_i(N-1), \overline{Q}_i\right)$, in which $\overline{Q}_i = U_{s_i}\Sigma_i {U_{s_i}}'$ with Σ_i obtained as the solution of the Lyapunov equation ${A_{s_i}}'\Sigma_i A_{s_i} - \Sigma_i = -{U_{s_i}}' Q_i U_{s_i}$.

The following theorem establishes exponential closed-loop stability under output feedback for systems with unstable modes.

Theorem 5.2 (Unstable modes)**.** *Let Assumptions 5.1 to 5.7 and Assumption 5.9 hold. Consider Algorithm 4.1, using the FC-MPC optimization problem of Equation (4.9) with an additional end constraint* ${U_{u_i}}'\widehat{x}_i(k+N|k) = {U_{u_i}}'[\mathcal{C}_N(A_i, B_i)\,\overline{\boldsymbol{u}}_i + A_i^N \widehat{x}_i(k)] = 0$ *enforced on the unstable decentralized modes. The origin is an exponentially stable equilibrium for the perturbed closed-loop system*

$$\widehat{x}_i(k+1) = A_i\widehat{x}_i(k) + B_i u_i^p(\widehat{\mu}(k), 0) + \sum_{j \neq i} W_{ij} u_j^p(\widehat{\mu}(k), 0) + Z_i e_i,$$

$$e_i(k+1) = (A_i - A_i\mathcal{L}_i C_i)e_i(k), \qquad i \in \mathbb{I}_M,$$

for all $((\widehat{x}_i(0), e_i(0)),\ i \in \mathbb{I}_M) \in \mathbb{D}_C$ *(Equation* (5.7)*) and all* $p = 1, 2, \ldots, p_{\max}$.

The proof is given in Appendix 5.9.5.

Remark 5.4. In Venkat et al. (2006f), it is shown that the nominal distributed MPC control law is exponentially stable. Lipschitz continuity of the nominal distributed MPC control law in the subsystem states μ is established. Asymptotic stability of the output feedback distributed MPC control law under decaying perturbations follows using (Scokaert et al., 1997, Theorem 3).

5.4 Example: Integrated styrene polymerization plants

We revisit the integrated styrene polymerization plants example described in Section 3.1. The performance of the FC-MPC framework under output feedback is evaluated and compared against the performance of decentralized and centralized MPC. The evolution of the system states is affected by zero mean random disturbances (noise). These random disturbances also corrupt available measurements. Subsystem-based Kalman filters are employed to estimate the composite model states. Two cases for the FC-MPC framework are considered, In the first case, the FC-MPC algorithm (Algorithm 4.1) is terminated after 1 iterate and in the second case, Algorithm 4.1 is terminated after 10 iterates. The performance of the FC-MPC framework tracking the temperature of each polymerization reactor in the first plant is shown in Figure 5.1. The closed-loop costs are compared in Table 5.1. The FC-MPC framework, with Algorithm 4.1 terminated after 1 iterate achieves performance that is within 1.2% of the optimal, centralized MPC performance. All plant outputs track their respective set points in ~ 25 hrs, roughly two-fifths the time required under decentralized MPC.

	Λ_{cost}	Performance loss (w.r.t centralized MPC)
Centralized-MPC	18.84	-
Decentralized-MPC	1608	8400%
FC-MPC (1 iterate)	18.94	0.54%
FC-MPC (5 iterates)	18.84	0%

Table 5.1: Closed-loop performance comparison of centralized MPC, decentralized MPC and FC-MPC.

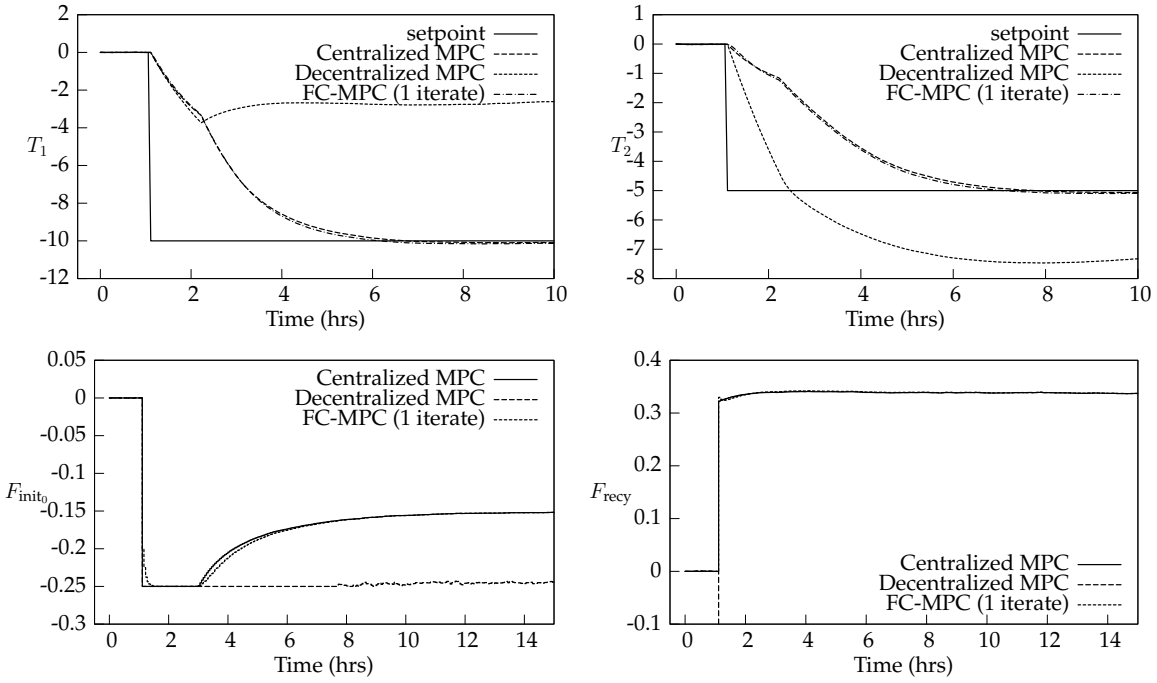

Figure 5.1: Interacting polymerization processes. Temperature control in the two polymerization reactors. Performance comparison of centralized MPC, decentralized MPC and FC-MPC (1 iterate).

5.5 Distillation column control

We revisit the distillation column considered in Section 4.7.1 (p. 54). The performance of centralized MPC, communication-based MPC and FC-MPC is investigated under output feedback. For communication and cooperation-based MPC, a local Kalman filter is employed for each subsystem. For centralized MPC, a single Kalman filter is used to estimate system states. Stochastic disturbances affect the evolution of the states and corrupts process measurements. The state and measurement noise covariances for each local estimator are $Q_{x_i} = 0.5I_{n_i}$ and $R_{v_i} = 0.1I_{n_{y_i}}$, $i = 1, 2$. For the centralized estimator $Q_x = \text{diag}(Q_{x_1}, Q_{x_2})$, $R_v = \text{diag}(R_{v_1}, R_{v_2})$.

At time 0, $x_i(0) = 0_{n_i}$, $\widehat{x}_i(0) = -0.1I_{n_i}$, $i = 1, 2$. The performance of the different MPC frameworks, in the presence of estimate error, is shown in Figure 5.2. Communication-

based MPC is unstable for this case. FC-MPC (1 iterate) stabilizes the system but achieves poor closed-loop performance relative to centralized MPC. FC-MPC (10 iterates) achieves improved closed-loop performance that is within 30% of the optimal centralized MPC. On iterating Algorithm 4.1 to convergence, the performance of FC-MPC is within a pre-specified tolerance of centralized MPC.

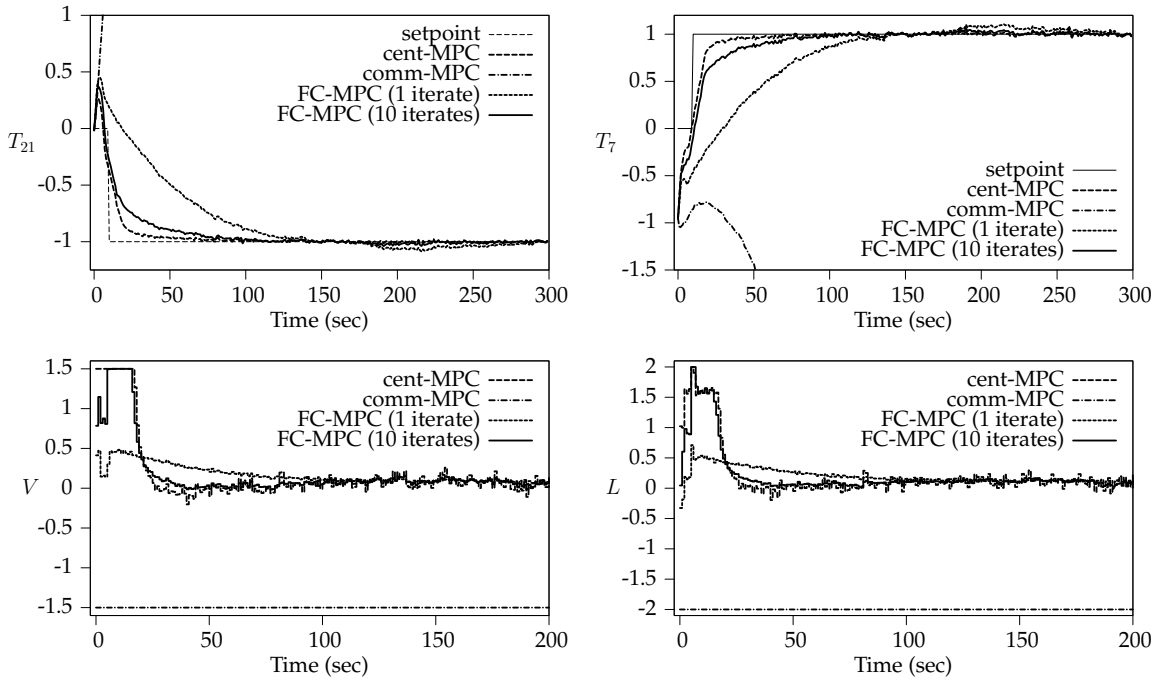

Figure 5.2: Setpoint tracking performance of centralized MPC, communication-based MPC and FC-MPC under output feedback. The prior model state at $k = 0$ underestimates the actual system states by 10%.

5.6 Discussion and conclusions

An output feedback distributed MPC framework with guaranteed feasibility, optimality and perturbed closed-loop stability properties was described in this chapter. Two distributed state estimation strategies were proposed for estimating the subsystem states using local measure-

ments. An attractive feature of the distributed estimator design procedure described in Section 5.2.1 is that it requires only local process data. The subsystem-based estimation strategy proposed in Section 5.2.2 allows a more general structure for the noise covariances and the noise shaping matrix. The latter strategy, however, requires a systemwide computation of the noise covariances, which may not be feasible in some cases. The distributed estimation strategies presented here do not need a *master* processor. The designed subsystem-based Kalman filters are stable estimators. Only local measurements are required for estimator updates. The trade-off here is the suboptimality of the generated estimates; the obtained estimates, however, converge to the optimal (centralized) estimates exponentially. The FC-MPC algorithm (Algorithm 4.1, p 43) is used for distributed regulation. Closed-loop stability under decaying perturbations for all (Algorithm 4.1) iteration numbers was established. The perturbed closed-loop stability result guarantees that the distributed estimator-distributed regulator assembly is stabilizing under intermediate termination of the FC-MPC algorithm.

5.7 Appendix: Preliminaries

5.7.1 Proof of Lemma 5.1

Proof. Let T be a similarity transform for the LTI system (A_m, B_m, C_m, G_m) with (A_m, B_m) stabilizable and (A_m, C_m) detectable. Let the transformed LTI system be

$$(\widetilde{A}_m, \widetilde{B}_m, \widetilde{C}_m, \widetilde{G}_m) = (TA_mT^{-1}, TB_m, C_mT^{-1}, TG_m).$$

We know from the Hautus lemma Sontag (1998) that

$$\text{rank}(\mathbb{H}[\lambda]) = \text{rank} \begin{bmatrix} \lambda I - A_m \\ \\ C_m \end{bmatrix} = n, \ \forall \, \lambda \in \lambda(A_m), \ |\lambda| \geq 1$$

From the definition of T and $\mathbb{H}[\lambda]$, we have

$$\mathbb{H}[\lambda] = \begin{bmatrix} \lambda I - A_m \\ \\ C_m \end{bmatrix} = \begin{bmatrix} \lambda I - T^{-1}\widetilde{A}_m T \\ \\ \widetilde{C}_m T \end{bmatrix} = \begin{bmatrix} T^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} \lambda I - \widetilde{A}_m \\ \\ \widetilde{C}_m \end{bmatrix} T$$

Let $\widetilde{\mathbb{H}}[\lambda] = \begin{bmatrix} \lambda I - \widetilde{A}_m \\ \\ \widetilde{C}_m \end{bmatrix}$. Therefore, $\widetilde{\mathbb{H}}[\lambda] = \begin{bmatrix} T & \\ & I \end{bmatrix} \mathbb{H}[\lambda] \ T^{-1}$. Suppose $(\widetilde{A}_m, \widetilde{C}_m)$ is not detectable. By assumption, there exists $\lambda_1, \ |\lambda_1| \geq 1$ and z such that $\widetilde{\mathbb{H}}[\lambda_1] z = 0, z \neq 0$, which gives

$$\begin{bmatrix} T & \\ & I \end{bmatrix} \mathbb{H}[\lambda_1] \ T^{-1} z = 0, \ z \neq 0$$

Let $v = T^{-1}z$. Since $z \neq 0$ and T is full rank, $v \neq 0$. This gives $\mathbb{H}[\lambda_1] v = 0, v \neq 0$, which contradicts detectability of (A_m, C_m). The arguments establishing the implication $(\widetilde{A}_m, \widetilde{C}_m)$ detectable $\implies$ (A_m, C_m) detectable are similar to those used earlier with T replaced by T^{-1}. Since stabilizability of $(\widetilde{A}_m, \widetilde{B}_m) \equiv$ detectability of $(\widetilde{A}_m{}', \widetilde{B}_m{}')$, stabilizability is also invariant under a similarity transformation. $\square$

5.8 Appendix: State estimation for FC-MPC

Theorem 5.3. *Let*

$$\mathbb{A} = \begin{pmatrix} \mathcal{A} & \\ & \mathcal{A}_s \end{pmatrix} \in \mathbb{R}^{(n+n_s)\times(n+n_s)}, \quad \mathbb{C} = \begin{pmatrix} \mathcal{C} & \mathcal{C}_s \end{pmatrix} \in \mathbb{R}^{n_y\times(n+n_s)}, \tag{5.10}$$

in which $\mathcal{A}_s$ is stable, $\mathcal{A} \in \mathbb{R}^{n\times n}$ and $\mathcal{C} \in \mathbb{R}^{n_y\times n}$. The pair $(\mathcal{A},\mathcal{C})$ is detectable if and only if $(\mathbb{A},\mathbb{C})$ is detectable.

Proof. From the Hautus lemma for detectability (Sontag, 1998, p. 318), $(\mathcal{A},\mathcal{C})$ is detectable iff $\operatorname{rank}\begin{pmatrix}\begin{bmatrix}\lambda I - \mathcal{A}\\ \mathcal{C}\end{bmatrix}\end{pmatrix} = n,\ \forall\, |\lambda| \geq 1$.

$(\mathcal{A},\mathcal{C})$ **detectable** $\implies$ $(\mathbb{A},\mathbb{C})$ **detectable.** Consider $|\lambda| \geq 1$. Detectability of $(\mathcal{A},\mathcal{C})$ implies the columns of $\begin{bmatrix}\lambda I - \mathcal{A}\\ \mathcal{C}\end{bmatrix}$ are independent. Hence, $\begin{bmatrix}\lambda I - \mathcal{A}\\ 0\\ \mathcal{C}\end{bmatrix}$ has independent columns. Since $\mathcal{A}_s$ is stable, the columns of $\lambda I - \mathcal{A}_s$ are independent, which implies the columns of $\begin{bmatrix}0\\ \lambda I - \mathcal{A}_s\\ \mathcal{C}_s\end{bmatrix}$ are also independent. Due to the positions of the zeros, the columns of

$$\begin{bmatrix} \lambda I - \mathcal{A} & 0 \\ 0 & \lambda I - \mathcal{A}_s \\ \mathcal{C} & \mathcal{C}_s \end{bmatrix}$$

are also independent. Hence, $(\mathbb{A},\mathbb{C})$ is detectable.

$(\mathbb{A}, \mathbb{C})$ **detectable** $\implies$ $(\mathcal{A}, \mathcal{C})$ **detectable.** We have from the Hautus lemma for detectability that the columns of

$$\begin{bmatrix} \lambda I - \mathcal{A} & \\ & \lambda I - \mathcal{A}_s \\ \mathcal{C} & \mathcal{C}_s \end{bmatrix}$$

are independent for all $|\lambda| \geq 1$. The columns of $\begin{bmatrix} \lambda I - \mathcal{A} \\ 0 \\ \mathcal{C} \end{bmatrix}$ are, therefore, independent. Hence, the columns of $\begin{bmatrix} \lambda I - \mathcal{A} \\ \mathcal{C} \end{bmatrix}$ are independent, which gives $(\mathcal{A}, \mathcal{C})$ is detectable. □

5.8.1 Proof for Lemma 5.3

Proof. Let $\mathcal{A} = A_{ii}, \mathcal{C} = C_{ii}$,

$$\mathcal{A}_s = \mathrm{diag}(A_{i1}, \ldots, A_{i(i-1)}, A_{i(i+1)}, \ldots, A_{iM}) \text{ and } \mathcal{C}_s = [C_{i1}, \ldots, C_{i(i-1)}, C_{i(i+1)}, \ldots, C_{iM}].$$

Also, let $\mathbb{A}$, $\mathbb{C}$ be given by Equation (5.10). We note that $A_i = U\mathbb{A}U$, $C_i = \mathbb{C}U$, in which U is a unitary matrix (hence a similarity transform). Invoking Theorem 5.3 and Lemma 5.1, we have the required result. □

5.8.2 Proof for Lemma 5.4

Proof. Define $\mathbb{A}_l = \mathrm{diag}(A_1, A_2, \ldots, A_{l-1})$ and $\mathbb{G}_l = [G_1{}', G_2{}', \ldots, G_{l-1}{}']'$, $l \in \mathbb{I}_M$. We have for all $\lambda \in \lambda(A_{\mathrm{cm}})$, $|\lambda| \geq 1$

$$\mathrm{rank}\begin{bmatrix}\lambda I - A_{\mathrm{cm}} & G_{\mathrm{cm}}\end{bmatrix} = \mathrm{rank}\begin{bmatrix}\lambda I - \mathbb{A}_M & & \mathbb{G}_M \\ & \lambda I - A_M & G_M\end{bmatrix} = n$$

This gives $\mathrm{rank}\begin{bmatrix}\lambda I - A_M & G_M\end{bmatrix} = n_M$ or we violate stabilizability of $(A_{\mathrm{cm}}, G_{\mathrm{cm}})$. Hence, (A_M, G_M) is stabilizable. We also have $\mathrm{rank}\begin{bmatrix}\lambda I - \mathbb{A}_M & \mathbb{G}_M\end{bmatrix} = n - n_M = s_{M-1}$. Now consider $\begin{bmatrix}\lambda I - \mathbb{A}_M & \mathbb{G}_M\end{bmatrix}$. We have

$$\mathrm{rank}\begin{bmatrix}\lambda I - \mathbb{A}_M & \lambda I - \mathbb{G}_M\end{bmatrix} \mathrm{rank}\begin{bmatrix}\lambda_1 I - \mathbb{A}_{M-1} & & \mathbb{G}_{M-1} \\ & \lambda_1 I - A_{M-1} & G_{M-1}\end{bmatrix} = s_{M-1} = s_{M-2} + n_{M-1},$$

which gives $\mathrm{rank}\begin{bmatrix}\lambda I - A_{M-1} & G_{M-1}\end{bmatrix} = n_{M-1}$ *i.e.,* (A_{M-1}, G_{M-1}) is stabilizable. Proceeding in this manner, we have (A_i, G_i), $\forall\, i \in \mathbb{I}_M$ is detectable. □

5.8.3 Proof for Lemma 5.5

Proof. $(\widehat{A}_i, \widehat{G}_i)$ **stabilizable** $\implies$ (A_i^o, G_i^o) **stabilizable.** By assumption, we have

$$\mathrm{rank}\begin{bmatrix}\lambda I - A_i^o & 0 & G_i^o \\ -A_i^{12} & \lambda I - A_i^{\overline{o}} & 0\end{bmatrix} = n_i = n_i^o + n_i^{\overline{o}},\ \forall\, |\lambda| \geq 1,$$

in which $A_i^{\bar{o}} \in \mathbb{R}^{n_i^{\bar{o}} \times n_i^{\bar{o}}}$. Consider $|\lambda| \geq 1$. From the rank condition above, we have that the rows of $[\lambda I - A_i^o,\ 0,\ G_i^o]$ are independent. Hence, the rows of $\begin{bmatrix} \lambda I - A_i^o & G_i^o \end{bmatrix}$ are independent *i.e.*, (A_i^o, G_i^o) is stabilizable.

(A_i^o, G_i^o) **stabilizable** $\implies$ $(\widehat{A}_i, \widehat{G}_i)$ **stabilizable.** Since (A_i^o, G_i^o) is stabilizable, the rows of $\begin{bmatrix} \lambda I - A_i^o & G_i^o \end{bmatrix}$ are independent for all $|\lambda| \geq 1$. Hence, the rows of $\begin{bmatrix} \lambda I - A_i^o & 0 & G_i^o \end{bmatrix}$ are also independent. From Lemma 5.3, (A_i, C_i) is detectable. Since (A_i, C_i) is detectable, its observability canonical form $(\widehat{A}_i, \widehat{C}_i)$ is also detectable (Lemma 5.1). From Equation (5.3), $A_i^{\bar{o}}$ is stable. The rows of $\lambda I - A_i^{\bar{o}}$ are independent, which implies the rows of $\begin{bmatrix} -A_i^{12} & \lambda I - A_i^{\bar{o}} & 0 \end{bmatrix}$ are also independent. Due to the positions of the zeros, the rows of

$$\begin{bmatrix} \lambda I - A_i^o & 0 & G_i^o \\ -A_i^{12} & \lambda I - A_i^{\bar{o}} & 0 \end{bmatrix}$$

are independent, which gives $(\widehat{A}_i, \widehat{G}_i)$ is stabilizable. □

5.9 Appendix: Perturbed closed-loop stability

Lemma 5.7 (Choi and Kwon (2003))**.** *Suppose Z is a positive semidefinite $n \times n$ matrix and a, b are n-dimensional vectors. Then given $\delta > 0$,*

$$(a+b)'Z(a+b) \leq (1+\delta)\, a'Za + \left(1 + \frac{1}{\delta}\right) b'Zb \tag{5.11}$$

Table 5.2: Two valid expressions for α_i

Case	α_1	α_2	α_3
1	$(1+\delta)$	$(1+\delta)\left(1+\frac{1}{\delta}\right)$	$\left(1+\frac{1}{\delta}\right)^2$
2	$(1+\delta)^2$	$(1+\delta)\left(1+\frac{1}{\delta}\right)$	$\left(1+\frac{1}{\delta}\right)$

Consider the case in which $Z = I$. Lemma 5.7 gives

$$\|a+b\|^2 \leq (1+\delta)\|a\|^2 + \left(1+\frac{1}{\delta}\right)\|b\|^2$$

Now, consider $\|a+b+c\|^2$. Repeated use of Lemma 5.7 gives

$$\|a+b+c\|^2 \leq \alpha_1\|a\|^2 + \alpha_2\|b\|^2 + \alpha_3\|c\|^2$$

There are six valid expressions for the α_i. Two valid expressions are given in Table 5.2.

Definition 5.1 (Hölder's inequality). For any set of nonnegative quantities a_i and b_i, $i = 1, 2, \ldots, n$, we have

$$(a_1^p + a_2^p + \ldots + a_n^p)^{1/p}(b_1^q + b_2^q + \ldots + b_n^q)^{1/q} \geq a_1b_1 + a_2b_2 + \ldots + a_nb_n$$

in which p and q are related by

$$\frac{1}{p} + \frac{1}{q} = 1$$

Corollary 5.1.1. For any set of nonnegative quantities a_i, $i = 1, 2, \ldots, n$, $\quad n^{p-1}\left(\sum_{i=1}^n a_i^p\right) \geq \left(\sum_{i=1}^n a_i\right)^p$.

Proof. The result follows by choosing $b_i = 1$, $i = 1, 2, \ldots, n$ in Hölder's inequality (Definition 5.1), and noting that $\frac{p}{q} = p - 1$. □

5.9.1 Preliminaries

Nominal closed-loop subsystem. Let Algorithm 4.1 be terminated after $p \in \mathbb{I}_+$ iterates. The evolution of each nominal closed-loop subsystem $i \in \mathbb{I}_M$ follows $x_i^+ = A_i x_i + B_i u_i^p(\mu, 0) + \sum_{j \neq i} W_{ij} u_j^p(\mu, 0) = F_i^p(\mu)$, in which $u_i^p(\mu, 0)$ is the control law for subsystem i.

Perturbed closed-loop subsystem. Let $e_i = x_i - \widehat{x}_{i,\ominus}$ denote the current estimate error for subsystem $i \in \mathbb{I}_M$. The symbol $\widehat{x}_{i,\ominus}$ denotes the estimate of x_i before current measurement y_i is available; $\widehat{x}_i$ represents the estimate of x_i after y_i is available. Let e_i^+ denotes the estimate error at the subsequent time step.

Assumption 5.10. $e_i^+ = \mathcal{A}_i^L e_i,\ |\lambda_{\max}(\mathcal{A}_i^L)| < 1, i \in \mathbb{I}_M$,

For Algorithm 4.1 terminated after p iterates, the control law for subsystem $i \in \mathbb{I}_M$ is $u_i^p(\widehat{\mu}, 0)$ (see Section 5.3). We have the following equations for the filter for subsystem i

$$\widehat{x}_i = \widehat{x}_{i,\ominus} + \mathfrak{L}_i(y_i - C_i \widehat{x}_{i,\ominus}),\ \ \widehat{x}_{i,\ominus}^+ = A_i \widehat{x}_i + B_i u_i^p(\widehat{\mu}, 0) + \sum_{j \neq i}^{M} W_{ij} u_j^p(\widehat{\mu}, 0),$$

in which $\widehat{x}_{i,\ominus}^+$ represents an estimate of the successor subsystem state x_i^+ before new measurement y_i^+ is available and $\mathfrak{L}_i, i \in \mathbb{I}_M$ is the filter gain. For each subsystem $i \in \mathbb{I}_M$, we have

$$\xi_i(1) = \widehat{x}_i^+ - \widehat{x}_{i,\ominus}^+ = (\widehat{x}_{i,\ominus}^+ + \mathfrak{L}_i C_i e_i^+) - \widehat{x}_{i,\ominus}^+ = Z_i e_i, \tag{5.12}$$

in which $Z_i = \mathfrak{L}_i C_i \mathcal{A}_i^L$ and $\widehat{x}_i^+$ represents an estimate of x_i^+ after y_i^+ is available. Consider Figure 5.3. Let $\boldsymbol{x}_i^p = [\rho_i^p(1)', \rho_i^p(2)', \ldots]'$ be the state trajectory for subsystem $i \in \mathbb{I}_M$ generated by $\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p$ and initial subsystem state $\widehat{x}_i$ (trajectory $\mathfrak{A}_i^p$ in Figure 5.3). We have $\rho_i^p(1) = \widehat{x}_{i,\ominus}^+, i \in \mathbb{I}_M$. The evolution of $\rho_i^p(j), j \geq 1$ in $\mathfrak{A}_i^p$ is

$$\rho_i^p(j) = A_i^{j-1}\rho_i^p(1) + \sum_{l=1}^{j-1} A_i^{j-1-l} B_i u_i^p(\widehat{\mu}, l) + \sum_{s \neq i}^{M} \sum_{l=1}^{j-1} A_i^{j-1-l} W_{is} u_s^p(\widehat{\mu}, l) \tag{5.13}$$

The state estimate for subsystem $i \in \mathbb{I}_M$ at the subsequent time step is $\widehat{x}_i^+ = \rho_i^p(1) + Z_i e_i$. Let $z_i(1) = \widehat{x}_i^+$. For each $i \in \mathbb{I}_M$, let $\boldsymbol{w}_i = [w_i(1)', w_i(2)', \ldots]'$, $w_i(j) \in \Omega_i,\ j \geq 1$ be an admissible input trajectory from $z_i(1)$. Let $\boldsymbol{z}_i = [z_i(2)', z_i(3), \ldots]'$ be the state trajectory for subsystem $i \in \mathbb{I}_M$ generated by $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M$ and initial subsystem state $z_i(1)$ (trajectory $\mathfrak{B}_i^0$ in Figure 5.3). For $z_i(j)$ in $\mathfrak{B}_i^0$, we write

$$z_i(j) = A_i^{j-1} z_i(1) + \sum_{l=1}^{j-1} A_i^{j-1-l} B_i w_i(l) + \sum_{s \neq i}^{M} \sum_{l=1}^{j-1} A_i^{j-1-l} W_{is} w_s(l) \tag{5.14}$$

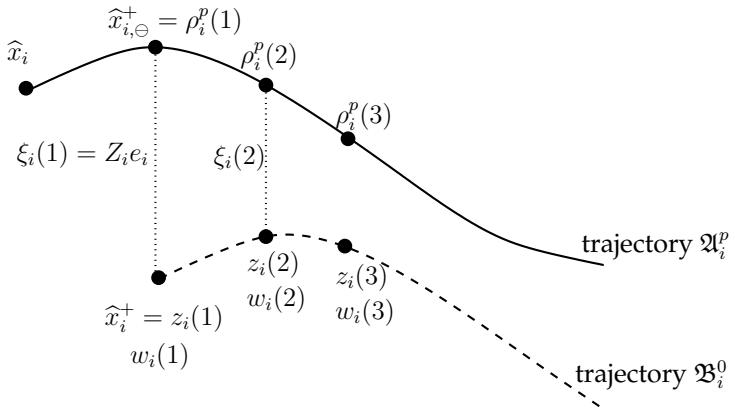

Figure 5.3: Trajectory $\mathfrak{A}_i^p$ is the state trajectory for subsystem i generated by $\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p$ and initial subsystem state $\widehat{x}_i$. The state trajectory $\mathfrak{B}_i^0$ for subsystem i is generated by $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M$ from initial state $z_i(1)$.

Define $\xi_i(j) = z_i(j) - \rho_i^p(j), v_i(j) = w_i(j) - u_i^p(\widehat{\mu}, j),\ j \geq 1$ and all $i \in \mathbb{I}_M$. For $j = 1$, we know from Equation (5.12) that $\xi_i(1) = Z_i e_i,\ i \in \mathbb{I}_M$. For $j > 1$, we have from Equations (5.13) and (5.14) that

$$\xi_i(j) = A_i^{j-1}\xi_i(1) + \sum_{l=1}^{j-1} A_i^{j-1-l} B_i v_i(l) + \sum_{s\neq i}^{M}\sum_{l=1}^{j-1} A_i^{j-1-l} W_{is} v_s(l) \tag{5.15}$$

For Algorithm 4.1 terminated after p iterates, the evolution of each perturbed closed-loop subsystem $i \in \mathbb{I}_M$ follows

$$\widehat{x}_i^+ = A_i\widehat{x}_i + B_i u_i^p(\widehat{\mu}, 0) + \sum_{j\neq i}^{M} W_{ij} u_j^p(\widehat{\mu}, 0) + Z_i e_i, \quad e_i^+ = \mathcal{A}_i^L e_i \tag{5.16}$$

5.9.2 Main result

Let $\widehat{\mu}^+ = [\widehat{x}_1^+, \ldots \widehat{x}_M^+]$ and $p, q \in \mathbb{I}_+$. For the set of estimated subsystem states $\widehat{\mu}$, we assume (WLOG) that Algorithm 4.1 is terminated after p iterates. At the subsequent time step with estimated state $\widehat{\mu}^+$), let q (possibly different from p) iterates be performed. Let the distributed MPC control law $u_i^p(\widehat{\mu}, 0),\ i \in \mathbb{I}_M$ be defined for $\widehat{\mu} \in \mathbb{X}_0$. Define $\mathbb{X}_u = \{\widehat{\mu} \,|\, \widehat{\mu} \in \mathbb{X}_0, u_i^p(\widehat{\mu}, 0) \in \Omega_i,\ i \in \mathbb{I}_M\}$.

Assumption 5.11. For the nominal closed-loop system $x_i^+ = F_i^p(\mu),\ i \in \mathbb{I}_M$,

$$J_N^p(\mu) = \Phi(\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p; \mu)$$

is a Lyapunov function satisfying

$$a_r \sum_{i=1}^{M} \|x_i\|^2 \leq J_N^p(\mu) \leq b_r \sum_{i=1}^{M} \|x_i\|^2 \tag{5.17a}$$

$$\Delta J_N(\mu) \leq -c_r \sum_{i=1}^{M} \|x_i\|^2 \tag{5.17b}$$

in which $a_r, b_r, c_r > 0$ and $\Delta J_N(\mu) = J_N^q(\mu^+) - J_N^p(\mu)$.

Define the set

$$\mathbb{Z} = \{ ((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M) \mid \big((\widehat{x}_i^+, e_i^+),\ i \in \mathbb{I}_M\big) \in \mathbb{Z},\ \widehat{\mu} \in \mathbb{X}_u\}, \tag{5.18}$$

in which $(\widehat{x}_i^+, e_i^+)$ is given by Equation (5.16). Let $((\widehat{x}_i(0), e_i(0)),\ i \in \mathbb{I}_M)$ represent the set of initial (estimated) subsystem states and initial estimate errors, respectively.

Theorem 5.4. *Let Assumptions 5.6, 5.10 and 5.11 hold. Consider the auxiliary system*

$$\xi_i(j+1) = A_i \xi_i(j) + B_i v_i(j) + \sum_{l \neq i}^{M} W_{il} v_l(j), \quad v_i(j) + u_i^p(\widehat{\mu}(0), j) \in \Omega_i,\ \forall\, i \in \mathbb{I}_M,\ j \geq 1,$$

with initial condition $\xi_i(1) = Z_i e_i(0)$. Suppose a set of perturbation trajectories

$$\boldsymbol{v}_i = [v_i(1)', v_i(2)', \ldots]',\ i \in \mathbb{I}_M$$

and a constant $\sigma_r > 0$ exist such that

$$\sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} L_i(\xi_i(j), v_i(j)) \leq \sigma_r \|e_i(0)\|^2, \tag{5.19}$$

the perturbed closed-loop system

$$\widehat{x}_i(k+1) = A_i\widehat{x}_i(k) + B_i u_i^p(\widehat{\mu}(k),0) + \sum_{l\neq i}^{M} W_{il}u_l^p(\widehat{\mu}(k),0) + Z_i e_i(k),$$

$$e_i(k+1) = \mathcal{A}_i^L e_i(k), \qquad i \in \mathbb{I}_M,$$

is exponentially stable for all $((\widehat{x}_i(0), e_i(0)),\ i \in \mathbb{I}_M) \in \mathbb{Z}$ *(Equation (5.18)).*

Proof. To establish exponential stability, we choose a candidate Lyapunov function that combines the regulator cost function and the subsystem state estimation errors (Choi and Kwon, 2003) employ a similar idea to show exponential stability of a single (centralized) MPC under output feedback). Define $V_N^p\left((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M\right) = J_N^p(\widehat{\mu}) + \frac{1}{2}\sum_{i=1}^{M} w_i e_i' \Psi_i e_i$ to be a candidate Lyapunov function, in which Ψ_i is the solution of the Lyapunov equation $\mathcal{A}_i^{L\prime}\Psi_i\mathcal{A}_i^L - \Psi_i = -\Pi_i$ and $\Pi_i > 0$ is a user-defined matrix. Since $\mathcal{A}_i^L$ is a stable matrix and $\Pi_i > 0$, it follows that Ψ_i exists, is unique and positive definite (p.d.) (Sontag, 1998, p. 230) for all $i \in \mathbb{I}_M$. Consider any $\widehat{x}_i(0) = \widehat{x}_i$ and $e_i(0) = e_i$, $i \in \mathbb{I}_M$ such that $((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M) \in \mathbb{Z}$. We need to show (Vidyasagar, 1993, p. 267) that there exists constants $a, b, c > 0$ such that

$$a\sum_{i=1}^{M}\left[\|\widehat{x}_i\|^2 + \|e_i\|^2\right] \leq V_N^p\left((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M\right) \leq b\sum_{i=1}^{M}\left[\|\widehat{x}_i\|^2 + \|e_i\|^2\right] \tag{5.20a}$$

$$\Delta V_N\left((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M\right) \leq -c\sum_{i=1}^{M}\left[\|\widehat{x}_i\|^2 + \|e_i\|^2\right], \tag{5.20b}$$

in which $\Delta V_N^p\left((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M\right) = V_N^q\left((\widehat{x}_i^+, e_i^+),\ i \in \mathbb{I}_M\right) - V_N^p\left((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M\right)$.

For subsystem $i \in \mathbb{I}_M$, let $\boldsymbol{x}_i^p = [\rho_i^p(1)', \rho_i^p(2)', \ldots]'$, $\rho_i^p(1) = \widehat{x}_{i,\ominus}^+ = F_i^p(\widehat{x}_i)$ be the state trajectory generated by the input trajectories $\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p$, obtained after p Algorithm 4.1 iter-

ates, and initial subsystem state $\widehat{x}_i$ (see Figure 5.3). Let $\boldsymbol{w}_i = \boldsymbol{u}_i^+(\mu) + \boldsymbol{v}_i,\ i \in \mathbb{I}_M$ (see Equation (5.5)) be a set of feasible subsystem input trajectories from $\widehat{\mu}^+$. The set of input trajectories $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M$ is used to initialize Algorithm 4.1 at the subsequent time step (from $\widehat{\mu}^+$). Let $\boldsymbol{z}_i = [z_i(2)', z_i(3)', \ldots]'$ denote the state trajectory generated by the set of feasible input trajectories $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M$, in which $z_i(2) = A_i\widehat{x}_i^+ + B_i w_i(1) + \sum_{j\neq i}^{M} w_j(1)$. For convenience, we define $z_i(1) = \widehat{x}_i^+,\ \forall\, i \in \mathbb{I}_M$. By definition (see p. 116), we have $z_i(j) = \rho_i^p(j) + \xi_i(j),\ j \geq 1,\ i \in \mathbb{I}_M$, and from Equation (5.12), $\xi_i(1) = Z_i e_i,\ i \in \mathbb{I}_M$. Using Lemma 4.4, we have

$$
\begin{aligned}
J_N^q(\widehat{\mu}^+) = \Phi([\boldsymbol{u}_1^q, \ldots, \boldsymbol{u}_M^q]; \widehat{\mu}^+) &\leq \Phi([\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M]; \widehat{\mu}^+) \\
&= \sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} L_i(z_i(j), w_i(j)) \\
&= \sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} L_i(\rho_i^p(j) + \xi_i(j), u_i^p(j) + v_i(j))
\end{aligned}
$$

Invoking Lemma 5.7 gives,

$$
\leq \sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} \left[(1+\delta) L_i(\rho_i^p(j), u_i^p(j)) + \left(1 + \frac{1}{\delta}\right) L(\xi_i(j), v_i(j)) \right]
$$

Hence, we have

$$
\begin{aligned}
J_N^q(\widehat{\mu}^+) &\leq (1+\delta) \sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} L_i(\rho_i^p(j), u_i^p(j)) + \left(1 + \frac{1}{\delta}\right) \sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} L_i(\xi_i(j), v_i(j)) \\
&\leq (1+\delta) \left[J_N^p(\widehat{\mu}) - \sum_{i=1}^{M} w_i L_i(\widehat{x}_i, u_i^p(\mu, 0)) \right] + \left(1 + \frac{1}{\delta}\right) \sigma_r \sum_{i=1}^{M} \|e_i\|^2
\end{aligned}
$$

We know $J_N^p(\mu) \leq b_r \sum_{i=1}^M \|x_i\|^2$. Let $\omega = \min_{i\in\mathbb{I}_M} w_i \frac{1}{2}\lambda_{\min}(Q_i)$. Therefore,

$$\begin{aligned} J_N^q(\widehat{\mu}^+) - J_N^p(\mu) &\leq \delta J_N^p(\widehat{\mu}) - \sum_{i=1}^M w_i L_i(\widehat{x}_i, u_i^p(\mu,0)) + \left(1+\frac{1}{\delta}\right)\sigma_r \sum_{i=1}^M \|e_i\|^2 \\ &\leq -(\omega - \delta b_r)\sum_{i=1}^M \|\widehat{x}_i\|^2 + \left(1+\frac{1}{\delta}\right)\sigma_r \sum_{i=1}^M \|e_i\|^2 \end{aligned}$$

Since $w_i, Q_i > 0,\ \forall\, i \in \mathbb{I}_M$, $\omega > 0$. Subsequently, we can choose $0 < c < \omega$ and $\delta_* = \frac{\omega - c}{b_r} > 0$. Let $d = \sigma_r\left(1 + \frac{1}{\delta_*}\right)$. We have

$$J_N^q(\widehat{\mu}^+) - J_N^p(\mu) \leq -c\sum_{i=1}^M \|x_i\|^2 + d\sum_{i=1}^M \|e_i\|^2$$

Define

$$\begin{aligned} \Delta e &= \frac{1}{2}\sum_{i=1}^M w_i {e_i^+}'\Psi_i e^+(k+1) - \frac{1}{2}\sum_{i=1}^M w_i {e_i}'\Psi_i e_i \\ &= \frac{1}{2}\sum_{i=1}^M w_i \left\{ {e_i}'\left[\mathcal{A}_i^{L\prime}\Psi_i\mathcal{A}_i^L - \Psi_i\right] e_i \right\} \\ &= -\frac{1}{2}\sum_{i=1}^M w_i {e_i}'\Pi_i e_i \end{aligned}$$

Let $w_{\min} = \min_{i\in\mathbb{I}_M} w_i$. The restriction $w_i > 0, i \in \mathbb{I}_M$ implies $w_{\min} > 0$. Since Π_i is a user-defined matrix, we can choose $\Pi_i = \Pi, \forall\, i \in \mathbb{I}_M$ such that $\lambda_{\min}(\Pi) = \frac{2}{w_{\min}}(d+c)$ [3]. Noting that

[3] e.g., choose Π to be any diagonal matrix with the smallest diagonal entry equal to $\frac{2}{w_{\min}}(d+c)$.

$\Delta V_N(\cdot) = J_N^q(\widehat{\mu}^+) - J_N^p(\widehat{\mu}) + \Delta e$ gives

$$\begin{aligned}\Delta V_N\left((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M\right) &\leq -c\sum_{i=1}^{M}\|\widehat{x}_i\|^2 + d\sum_{i=1}^{M}\|e_i\|^2 - \frac{1}{2}\left(\min_{i\in\mathbb{I}_M} w_i\lambda_{\min}(\Pi_i)\right)\sum_{i=1}^{M}\|e_i\|^2 \\ &= -c\sum_{i=1}^{M}\|\widehat{x}_i\|^2 - \left(\frac{1}{2}w_{\min}\lambda_{\min}(\Pi) - d\right)\sum_{i=1}^{M}\|e_i\|^2 \\ &= -c\sum_{i=1}^{M}\left[\|\widehat{x}_i\|^2 + \|e_i\|^2\right]\end{aligned}$$

Since $\Psi_i > 0,\ \forall\, i \in \mathbb{I}_M$, there exists constants $a_e, b_e > 0$ such that

$$a_e\sum_{i=1}^{M}\|e_i\|^2 \leq \frac{1}{2}\sum_{i=1}^{M} w_i {e_i}'\Psi_i e_i \leq b_e\sum_{i=1}^{M}\|e_i\|^2.$$

The choice $a = \min(a_r, a_e)$, $b = \max(b_r, b_e)$ satisfies Equation (5.20a).

□

5.9.3 Proof for Theorem 5.1

Proof for Theorem 5.1. From Lemma 5.3, we have (A_i, C_i) is detectable. It follows from Section 5.2 that there exists $\mathcal{L}_i, \forall i \in \mathbb{I}_M$ such that $\mathcal{A}_i^L = (A_i - A_i\mathcal{L}_i C_i)$ is a stable matrix. From (Sontag, 1998, p. 231), $\overline{Q}_i > 0,\ i \in \mathbb{I}_M$. Using arguments identical to the state feedback case (see proof for Theorem 4.1, p. 81), constants a_r, b_r and c_r that satisfy Equation (5.17) can be determined. We note that $\boldsymbol{u}_i^+(\widehat{\mu}),\ i \in \mathbb{I}_M$ is a set of feasible input trajectories for the successor subsystem states $\widehat{\mu}^+$. For the choice $\boldsymbol{v}_i = [0, 0, \ldots]',\ \forall\, i \in \mathbb{I}_M$, Equation (5.15) gives

$\xi_i(j) = A_i^{j-1}\xi_i(1), 1 \leq j$. Hence,

$$\begin{aligned}\sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} L_i(\xi_i(j), v_i(j)) = \sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} L_i(\xi_i(j), 0) &= \sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} \xi_i(j)' A_i^{j-1\prime} Q_i A_i^{j-1} \xi_i(j) \\ &= \sum_{i=1}^{M} w_i \xi_i(1)' \overline{Q}_i \xi_i(1) \\ &\leq \sum_{i=1}^{M} w_i \lambda_{\max}(\overline{Q}_i) \|Z_i\|^2 \|e_i\|^2\end{aligned}$$

The choice $\sigma_r = \max(w_1\lambda_{\max}(\overline{Q}_i)\|Z_i\|^2, \ldots, w_M\lambda_{\max}(\overline{Q}_M)\|Z_M\|^2)$ satisfies Equation (5.19). Invoking Theorem 5.4 with $\mathbb{Z} = \mathbb{R}^n \times \mathbb{R}^n$ completes the proof. □

5.9.4 Construction of $\mathbb{D}_{D_i}$ for unstable systems

Let $\varepsilon_i > 0$, selected arbitrarily small. Choose $\widehat{x}_{ii} \in \mathbb{X}_i^0 \subseteq \mathbb{S}_i$ and $e_i \in B_{\varepsilon_i}(0)$. Define

$$\mathfrak{X}_0^i = \{(\widehat{x}_{ii}, e_i) \,|\, \widehat{x}_{ii} \in \mathbb{X}_i^0, e_i \in B_{\varepsilon_i}(0)\}$$

Let

$$\mathfrak{X}_{-1}^i = \{(\widehat{x}_{ii}, e_i) \,|\, \exists\, u_i^p(\widehat{\mu}, 0) \in \Omega_i \text{ such that } (\widehat{x}_{ii}^+, e_i^+) \in \mathfrak{X}_0^i, \widehat{x}_{ii} \in \mathbb{S}_i\}$$

The set $\mathfrak{X}_{-1}^i$ consists of $(\widehat{x}_{ii}, e_i)$ pairs for which an admissible distributed MPC control law for subsystem i exists that drives the successor decentralized state and estimate error inside $\mathfrak{X}_0^i$. Proceeding backwards in this manner, we have for any $l \in \mathbb{I}_+$ that

$$\mathfrak{X}_{-l}^i = \{(\widehat{x}_{ii}, e_i) \,|\, \exists\, u_i^p(\widehat{\mu}, 0) \in \Omega_i \text{ such that } (\widehat{x}_{ii}^+, e_i^+) \in \mathfrak{X}_{-l+1}^i, \widehat{x}_{ii} \in \mathbb{S}_i\}$$

The maximal, positively invariant stabilizable set for subsystem i is $\mathfrak{X}^i_{-\infty} = \bigcup_{l=1}^{\infty} \mathfrak{X}_{-l+1}$. Finite determination of this maximal, positively invariant stabilizable set is possible if and only if $\mathfrak{X}^i_{-l} = \mathfrak{X}^i_{-l+1)}$ for some $l \in \mathbb{I}_+$ (Blanchini, 1999; Kolmanovsky and Gilbert, 1998; Rakovic et al., 2004). If finite determination of $\mathfrak{X}^i_{-\infty}$ is possible, we set $\mathbb{D}_{D_i} = \mathfrak{X}^i_{-\infty}$. If $\mathfrak{X}^i_{-\infty}$ cannot be determined finitely, we make the conservative choice $\mathbb{D}_{D_i} = \mathfrak{X}_{-L}$ for some L large.

5.9.5 Proof for Theorem 5.2

Proof for Theorem 5.2. Existence of $\mathfrak{L}_i = \mathcal{L}_i, i \in \mathbb{I}_M$ such that $\mathcal{A}_i^L = (A_i - A_i\mathcal{L}_i C_i)$ is a stable matrix follows from Lemma 5.3. A procedure for determining constants a_r, b_r and c_r satisfying Equation (5.17) is given in the proof for Theorem 4.2 (see Appendix 4.10.6, p. 84) and is omitted for brevity. Consider Figure 5.3. We have using Equation (5.13) and the definition of $\boldsymbol{u}_i^p(\mu)$ that ${U_{u_i}}'\rho_i^p(N) = {U_{u_i}}'\rho_i^p(N+1) = 0$. For $((\widehat{x}_i, e_i),\ i \in \mathbb{I}_M) \in \mathbb{D}_C$ (see Section 5.3.2), $\overline{\boldsymbol{v}}_i, i \in \mathbb{I}_M$ exists. One possible choice for $\overline{\boldsymbol{v}}_i$ is the solution to the QP of Equation (5.9). Let $\boldsymbol{v}_i = [\overline{\boldsymbol{v}}_i', 0, \ldots]'$ and define $\boldsymbol{w}_i = \boldsymbol{u}_i^+(\widehat{\mu}) + \boldsymbol{v}_i,\ i \in \mathbb{I}_M$ to be admissible input trajectories from $\widehat{\mu}^+$ satisfying ${U_{u_i}}'z_i(N+1) = 0,\ i \in \mathbb{I}_M$. Hence, ${U_{u_i}}'\xi_i(N+1) = 0,\ \forall\, i \in \mathbb{I}_M$. Let $j \in \mathbb{I}_+ \cup \{0\}$. From Lemma 4.1 (p. 29), a constant K_{e_i} independent of e_i exists for each $i \in \mathbb{I}_M$ such that $\|v_i(j)\| \leq K_{e_i}\|e_i\|,\ 0 \leq j$ and ${U_{u_i}}'\xi_i(N+1) = 0$. Let $\mathcal{A}_i = \max_{0\leq j\leq N} \|A_i^j\|$ and $\mathcal{A} = \max_{i\in\mathbb{I}_M} \mathcal{A}_i$. For each

subsystem $i \in \mathbb{I}_M$, we have from Equation (5.15) that

$$\begin{aligned}
\|\xi_i(j)\| &\leq \|A_i^{j-1}\|\|\xi_i(1)\| + \sum_{l=1}^{j-1} \|A_i^{j-1-l}\|\|B_i\|\|v_i(l)\| + \sum_{l=1}^{j-1}\sum_{s\neq i}^{M} \|A_i^{j-1-l}\|\|W_{is}\|\|v_s(l)\| \\
&= \mathcal{A}\|Z_i\|\|e_i\| + \sum_{l=1}^{j-1} \mathcal{A}\|B_i\|K_{e_i}\|e_i\| + \sum_{s\neq i}^{M}\sum_{l=1}^{j-1} \mathcal{A}\|W_{is}\|K_{e_i}\|e_s\| \\
&\leq \mathcal{A}(\|Z_i\| + N\|B_i\|K_{e_i})\|e_i\| + \sum_{s\neq i}^{M} N\mathcal{A}\|W_{is}\|K_{e_s}\|e_s\| \\
&\leq \beta_{e_i} \sum_{s=1}^{M} \|e_s\|, \; \forall\, 1 \leq j \leq N+1,
\end{aligned}$$

in which $\beta_{e_i} = \max(\mathcal{A}(\|Z_i\| + N\|B_i\|K_{e_i}), \Xi_i)$ and $\Xi_i = N\mathcal{A}\max_{s\in\mathbb{I}_M} \|W_{is}\|K_{e_s}$. Let $\mathcal{F}^\infty = \sum_{i=1}^{M} w_i \sum_{j=1}^{\infty} L_i\left(\xi_i(j), v_i(j)\right)$. We have

$$\begin{aligned}
\mathcal{F}^\infty &= \sum_{i=1}^{M} w_i \left[\sum_{j=1}^{N} L_i(\xi_i(j), v_i(j)) + \sum_{j=N+1}^{\infty} L_i(\xi_i(j), v_i(j)) \right] \\
&= \sum_{i=1}^{M} w_i \left[\sum_{j=1}^{N} L_i(\xi_i(j), v_i(j)) + \frac{1}{2}\xi_i(N+1)'\overline{Q}_i\xi_i(N+1) \right] \\
&\leq \sum_{i=1}^{M} w_i \frac{1}{2} \left\{ \left[\lambda_{\max}(Q_i)N + \lambda_{\max}(\overline{Q}_i)\right] \beta_{e_i}^2 \left(\sum_{s=1}^{M} \|e_s\| \right)^2 + N\lambda_{\max}(R_i)K_{e_i}^2\|e_i\|^2 \right\}
\end{aligned}$$

Invoking Corollary 5.1.1 with $p, q = 2$ and $n = M$ gives, $\left(\sum_{i=1}^{M} \|e_i\|\right)^2 \leq M \sum_{i=1}^{M} \|e_i\|^2$. Hence,

$$\mathcal{F}^\infty \leq \sum_{i=1}^{M} w_i \frac{1}{2} \left\{ \left[\lambda_{\max}(Q_i)N + \lambda_{\max}(\overline{Q}_i)\right] \beta_{e_i}^2 M \sum_{s=1}^{M} \|e_s\|^2 + N\lambda_{\max}(R_i)K_{e_i}^2\|e_i\|^2 \right\}$$

Define the constants

$$\eta_a = \max_{i \in \mathbb{I}_M} \frac{1}{2} w_i \left[\lambda_{\max}(Q_i) N + \lambda_{\max}(\overline{Q}_i) \right] \beta_{e_i}^2 M$$

$$\eta_b = \max_{i \in \mathbb{I}_M} \frac{1}{2} w_i N \lambda_{\max}(R_i) K_{e_i}^2$$

and $\eta = M\eta_a + \eta_b$. This gives $\mathcal{F}^\infty \leq \eta \sum_{i=1}^{M} \|e_i\|^2,\ \eta > 0$. Choosing $\sigma_r = \eta$ and invoking Theorem 5.4 with $\mathbb{Z} = \mathbb{D}_C$ proves the theorem. □

Chapter 6

Offset-free control with FC-MPC [1]

This chapter addresses the issue of achieving offset-free control objectives with distributed MPC. The standard practice in MPC is to augment the states in the process model with integrating disturbances. In a distributed MPC framework, many choices for disturbance models exist. From a practitioner's standpoint, it is usually convenient to use local integrating disturbances. We consider first, a disturbance modeling framework to achieve zero-offset steady-state control in the presence of nonzero mean disturbances and/or plant model mismatch. Next, a distributed algorithm for computing the steady-state input, state and output targets locally is described. Conditions that ensure offset-free control at steady state are discussed subsequently. Two examples are presented to illustrate the efficacy of the distributed MPC framework with distributed estimation, local disturbance modeling and distributed target calculation. Finally, the main contributions of this chapter are summarized.

We make the following assumptions:

Assumption 6.1. All interaction models are stable *i.e.*, for each $i, j \in \mathbb{I}_M$, $|\lambda_{\max}(A_{ij})| < 1, \forall j \neq i$.

Assumption 6.2. For each $i \in \mathbb{I}_M$, (A_{ii}, C_{ii}) is detectable.

[1] **Portions of this chapter appear in Venkat et al. (2006e) and in Venkat et al. (2006g).**

Assumption 6.3. For each $i \in \mathbb{I}_M$, (A_{ii}, B_{ii}) is stabilizable.

6.1 Disturbance modeling for FC-MPC

For each subsystem $i \in \mathbb{I}_M$, the CM state is augmented with the integrating disturbance. The augmented CM $(\widetilde{A}_i, \widetilde{B}_i, \{\widetilde{W}_{ij}\}_{j\neq i}, \widetilde{C}_i, \widetilde{G}_i)$ for subsystem i is

$$\underbrace{\begin{bmatrix} x_i \\ d_i \end{bmatrix}}_{\widetilde{x}_i}(k+1) = \underbrace{\begin{pmatrix} A_i & B_i^d \\ 0 & I \end{pmatrix}}_{\widetilde{A}_i}\begin{bmatrix} x_i \\ d_i \end{bmatrix}(k) + \underbrace{\begin{pmatrix} B_i \\ 0 \end{pmatrix}}_{\widetilde{B}_i} u_i(k) + \sum_{j\neq i}^{M} \underbrace{\begin{pmatrix} W_{ij} \\ 0 \end{pmatrix}}_{\widetilde{W}_{ij}} u_j(k) + \underbrace{\begin{pmatrix} G_i & \\ & I \end{pmatrix}}_{\widetilde{G}_i}\begin{bmatrix} w_{x_i} \\ w_{d_i} \end{bmatrix}(k),$$

$$y_i(k) = \underbrace{\begin{pmatrix} C_i & C_i^d \end{pmatrix}}_{\widetilde{C}_i}\begin{bmatrix} x_i \\ d_i \end{bmatrix}(k) + \nu_i(k),$$

in which $d_i \in \mathbb{R}^{n_{d_i}}$, $B_i^d \in \mathbb{R}^{n_i \times n_{d_i}}$, $C_i^d \in \mathbb{R}^{n_{y_i} \times n_{d_i}}$. The vectors $w_{x_i}(k) \sim N(0, Q_{x_i}) \in \mathbb{R}^{n_{g_i}}$, $w_{d_i}(k) \sim N(0, Q_{d_i}) \in \mathbb{R}^{n_{d_i}}$ and $\nu_i(k) \sim N(0, R_{v_i}) \in \mathbb{R}^{n_{y_i}}$ are zero mean white noise disturbances affecting the augmented CM state equation and output equation, respectively. The notation (B_i^d, C_i^d) represents the input–output disturbance model pair for subsystem i, in which $B_i^d = \text{vec}(B_{i1}^d, \ldots, B_{ii}^d, \ldots, B_{iM}^d)$. The augmented decentralized model $(\widetilde{A}_{ii}, \widetilde{B}_{ii}, \widetilde{C}_{ii})$ is obtained by augmenting the decentralized state x_{ii} with the integrating disturbance d_i. It is assumed that the augmented decentralized model with the input-output disturbance model pair $(\mathcal{B}_{ii}^d, \mathcal{C}_i^d)$ is detectable (hence, $n_{d_i} \leq n_{y_i}$) [2].

Lemma 6.1. *Let Assumptions 6.1 and 6.2 hold. For each subsystem $i \in \mathbb{I}_M$, let the augmented decentralized model $(\widetilde{A}_{ii}, \widetilde{C}_{ii})$ with the input-output disturbance model pair $(\mathcal{B}_{ii}^d, \mathcal{C}_i^d)$ be detectable. The*

[2] Conditions for detectability of the augmented decentralized model are given in (Pannocchia and Rawlings, 2002, Lemma 1, p. 431)

augmented CM $(\widetilde{A}_i, \widetilde{C}_i)$ *with input disturbance model* $B_i^d = \mathrm{vec}(B_{i1}^d, \ldots B_{ii}^d, \ldots, B_{iM}^d)$*, in which* $B_{ii}^d = \mathcal{B}_{ii}^d$*,* $B_{ij}^d = 0,\ j \in \mathbb{I}_M,\ j \neq i$*, and output disturbance model* $\mathcal{C}_i^d$*, is detectable.*

A proof is presented in Appendix 6.6.1.

In view of the internal model principle of Francis and Wonham (1976), it may be preferable to choose disturbance models that best represent the actual plant disturbances. Hence, in certain cases, it may be useful to use a more general input disturbance model of the form $B_i^d = \mathrm{vec}(B_{i1}^d, \ldots, B_{ii}^d, \ldots, B_{iM}^d)$ in conjunction with the output disturbance model C_i^d. The following lemma gives a general condition for detectability of the augmented CM $(\widetilde{A}_i, \widetilde{C}_i)$.

Lemma 6.2. *Let Assumption 6.1 and 6.2 hold. The augmented CM* $(\widetilde{A}_i, \widetilde{C}_i)$*, with input disturbance model* $B_i^d = \mathrm{vec}(B_{i1}^d, \ldots, B_{ii}^d, \ldots, B_{iM}^d)$ *and output disturbance model* C_i^d*, is detectable if and only if*

$$\mathrm{rank} \begin{bmatrix} I - A_i & -B_i^d \\ C_i & C_i^d \end{bmatrix} = n_i + n_{d_i} \tag{6.1}$$

A proof is given in Appendix 6.6.2.

One method to satisfy the rank condition of Equation (6.1) is to pick B_i^d and C_i^d such that $\mathrm{range} \left(\begin{bmatrix} B_i^d \\ -C_i^d \end{bmatrix} \right) \subseteq \mathrm{null} \left(\begin{bmatrix} I - A_i \\ C_i \end{bmatrix}' \right)$. Let $(\widetilde{A}_i, \widetilde{C}_i)$ be detectable and let the steady-state estimator gain for the state and integrating disturbance vector for subsystem i be denoted by

$\mathcal{L}_{\mathrm{x}_i}$ and $\mathcal{L}_{\mathrm{d}_i}$ respectively. The filter equations for subsystem i are

$$\begin{bmatrix} \widehat{x}_i \\ \widehat{d}_i \end{bmatrix}(k|k) = \begin{bmatrix} \widehat{x}_i \\ \widehat{d}_i \end{bmatrix}(k|k-1) + \begin{bmatrix} \mathcal{L}_{\mathrm{x}_i} \\ \mathcal{L}_{\mathrm{d}_i} \end{bmatrix} \left(y_i(k) - C_i \widehat{x}_i(k|k-1) - C_i^d \widehat{d}_i(k|k-1) \right) \tag{6.2a}$$

$$\begin{bmatrix} \widehat{x}_i \\ \widehat{d}_i \end{bmatrix}(k+1|k) = \begin{bmatrix} A_i & B_i^d \\ & I \end{bmatrix} \begin{bmatrix} \widehat{x}_i \\ \widehat{d}_i \end{bmatrix}(k|k) + \begin{bmatrix} B_i \\ 0 \end{bmatrix} u_i(k) + \sum_{j \neq i}^{M} \begin{bmatrix} W_{ij} \\ 0 \end{bmatrix} u_j(k) \tag{6.2b}$$

in which $\widehat{x}_i, \widehat{d}_i$ denotes an estimate of the state and integrating disturbance respectively, for subsystem i.

6.2 Distributed target calculation for FC-MPC

For robustness and redundancy, the number of measurements is typically chosen greater than the number of manipulated inputs. Consequently, one can achieve offset-free control for only a subset of the measured variables. Define $z_i = H_i y_i, z_i \in \mathbb{R}^{n_{c_i}}, H_i \in \mathbb{R}^{n_{c_i} \times n_{y_i}}$ to be the set of controlled variables (CVs) for each subsystem $i \in \mathbb{I}_M$. The choice of CVs is presumed to satisfy:

Assumption 6.4.

$$\operatorname{rank} \begin{bmatrix} I - A_{ii} & -B_{ii} \\ H_i C_{ii} & 0 \end{bmatrix} = n_{ii} + n_{c_i},\ i \in \mathbb{I}_M, \tag{6.3}$$

Assumption 6.4 implies that the number of CVs for each subsystem $i \in \mathbb{I}_M$ cannot exceed either the number of manipulated variables (MVs) m_i or the number of measurements n_{y_i}, and that $H_i C_{ii}$ must be full row rank.

Lemma 6.3. *Let Assumptions 6.1 and 6.4 hold.*

$$\text{rank} \begin{bmatrix} I - A_i \\ H_i C_i \end{bmatrix} = n_i \;\; \textit{if and only if} \;\; \text{rank} \begin{bmatrix} I - A_{ii} \\ H_i C_{ii} \end{bmatrix} = n_{ii}.$$

The proof is similar to the proof for Lemma 5.3, and is omitted.

Assumption 6.5. $\text{rank} \begin{bmatrix} I - A_{ii} \\ H_i C_{ii} \end{bmatrix} = n_{ii},\ i \in \mathbb{I}_M.$

Assumption 6.5 is a weaker restriction than detectability of $(A_{ii}, H_i C_{ii})$. In the distributed target calculation framework, the steady-state targets are computed at the subsystem level. At each iterate, an optimization and exchange of calculated steady-state information among interacting subsystems is performed. For subsystem $i \in \mathbb{I}_M$, let z_i^{sp} denote the setpoint for the CVs and let u_i^{ss} represent the corresponding steady-state value for the MVs. Hence, we write $z_i^{\text{sp}} = \mathcal{G}_i u_i^{\text{ss}}$, where $\mathcal{G}_i$ is a steady-state gain matrix. The triplet $(y_{s_i}, x_{s_i}, u_{s_i})$ represents the steady-state output, state and input target for subsystem i. The target objective for subsystem $i \in \mathbb{I}_M$, Ψ_i, is defined as $\Psi_i(u_{s_i}) = \frac{1}{2}(u_i^{\text{ss}} - u_{s_i})' R_{u_i} (u_i^{\text{ss}} - u_{s_i})$, in which $R_{u_i} > 0$. Each subsystem $i \in \mathbb{I}_M$, solves the following QP at iterate t.

$$(x_{s_{ii}}^{*(t)}, u_{s_i}^{*(t)}) \in \;\; \arg \min_{x_{s_{ii}}, u_{s_i}} \frac{1}{2}(u_i^{\text{ss}} - u_{s_i})' R_{u_i} (u_i^{\text{ss}} - u_{s_i}) \tag{6.4a}$$

subject to

$$u_{s_i} \in \Omega_i \tag{6.4b}$$

$$\begin{bmatrix} I - A_{ii} & -B_{ii} \\ H_i C_{ii} & \end{bmatrix} \begin{bmatrix} x_{s_{ii}} \\ u_{s_i} \end{bmatrix} = \begin{bmatrix} B_{ii}^d \widehat{d}_i \\ z_i^{\text{sp}} - H_i C_i^d \widehat{d}_i - \sum_{j \neq i}^{M} \left(\overline{g}_{ij} u_{s_j}^{t-1} + \overline{h}_{ij} \widehat{d}_i \right) \end{bmatrix} \tag{6.4c}$$

in which $\overline{g}_{ij} = H_i C_{ij}(I - A_{ij})^{-1} B_{ij},\ \overline{h}_{ij} = H_i C_{ij}(I - A_{ij})^{-1} B^d_{ij}, \forall\, i, j \in \mathbb{I}_M,\ j \neq i.$

Existence. Let Assumption 6.4 be satisfied. Consider

$$\begin{bmatrix} I - A_{ii} & B_{ii} \\ H_i C_{ii} & \end{bmatrix} \begin{pmatrix} x_{s_{ii}} \\ u_{s_i} \end{pmatrix} = \begin{bmatrix} B^d_{ii}\widehat{d}_i \\ z^{\text{sp}}_i - H_i C^d_i \widehat{d}_i - \sum_{j \neq i}^{M} \left(\overline{g}_{ij} u_{s_j} + \overline{h}_{ij} \widehat{d}_i \right) \end{bmatrix},\ i \in \mathbb{I}_M \tag{6.5}$$

$$\mathbb{D}_T = \Big\{ \left((z^{\text{sp}}_i, \widehat{d}_i),\ i \in \mathbb{I}_M \right) \mid \exists\ ((x_{s_{ii}}, u_{s_i}), i \in \mathbb{I}_M) \text{ satisfying Equation (6.5)} \\ \text{and } u_{s_j} \in \Omega_j,\ \forall\, j \in \mathbb{I}_M \Big\} \qquad \text{domain of target}$$

If $\mathbb{D}_T$ is empty, the constraints are too stringent to meet $z^{\text{sp}}_i,\ i \in \mathbb{I}_M$. For $\mathbb{D}_T$ nonempty and $\left((z^{\text{sp}}_i, \widehat{d}_i), i \in \mathbb{I}_M \right) \in \mathbb{D}_T$, the feasible region for Equation (6.4) is nonempty for each $i \in \mathbb{I}_M$. Since $R_{u_i} > 0$, the objective is bounded below. A solution to Equation (6.4), therefore, exists for all $i \in \mathbb{I}_M$ (Frank and Wolfe, 1956).

Uniqueness.

Lemma 6.4. *For each subsystem $i \in \mathbb{I}_M$, let H_i satisfy Assumption 6.4. Let Assumption 6.1 hold. The solution to the target optimization problem (Equation* (6.4)*) for each $i \in \mathbb{I}_M$, if it exists, is unique if and only if Assumption 6.5 is satisfied.*

A proof is given in Appendix 6.6.4.

Corollary 6.4.1. $x^{*(t)}_{s_i} = [x^{*(t)\prime}_{s_{i1}}, \ldots, x^{*(t)\prime}_{s_{iM}}]', i \in \mathbb{I}_M$ *is unique.*

Remark 6.1. It can be shown that (A_i, H_iC_i) is detectable if and only if (A_{ii}, H_iC_{ii}) is detectable. For subsystem i, if H_i satisfies Assumption 6.4, (A_{ii}, H_iC_{ii}) is detectable and Assumption 6.1 holds, the solution to the optimization problem of Equation (6.4) is unique.

The steady-state targets for each $i \in \mathbb{I}_M$ are obtained using the distributed target calculation algorithm given below.

Algorithm 6.1. Given $\left(u_{s_i}^0,\ z_i^{\mathrm{sp}},\ u_i^{\mathrm{ss}}\right),\ R_{u_i} > 0,\ \forall\, i \in \mathbb{I}_M, t_{\max} > 0,\ \epsilon > 0$

$t \leftarrow 1,\ \kappa_i \leftarrow \Gamma\epsilon,\ \Gamma \gg 1$

while $\kappa_i > \epsilon$ for some $i \in \mathbb{I}_M$ and $t \leq t_{\max}$

do $\forall\, i \in \mathbb{I}_M$

Determine $(x_{s_{ii}}^{*(t)}, u_{s_i}^{*(t)})$ from Equation (6.4)

$(x_{s_{ii}}^t, u_{s_i}^t) \leftarrow w_i(x_{s_{ii}}^{*(t)}, u_{s_i}^{*(t)}) + (1 - w_i)(x_{s_{ii}}^{t-1}, u_{s_i}^{t-1})$

$\kappa_i \leftarrow \|(x_{s_{ii}}^t, u_{s_i}^t) - (x_{s_{ii}}^{t-1}, u_{s_i}^{t-1})\|$

Transmit $(x_{s_{ii}}^t, u_{s_i}^t)$ to each interconnected subsystem $j \in \mathbb{I}_M,\ j \neq i$

end (do)

$t \leftarrow t + 1$

end (while)

For each subsystem $i \in \mathbb{I}_M$ at iterate t, the target state $x_{s_{ij}}^t, j \neq i$ is calculated using $x_{s_{ij}}^t = (I - A_{ij})^{-1}\left[B_{ij}u_{s_j}^t + B_{ij}^d\widehat{d}_i\right]$ and by definition, $x_{s_i}^t = [x_{s_{i1}}^t{}', \ldots, x_{s_{iM}}^t{}']'$. All iterates generated by Algorithm 6.1 are feasible steady states. Furthermore, the target calculation objective $\Psi(\cdot) = \sum_{i=1}^M w_i\Psi_i(u_{s_i}^t)$, in which $w_i > 0,\ i \in \mathbb{I}_M$ and $\sum_{i=1}^M w_i = 1$, is a nonincreasing function of the iteration number t. Since $\Psi_i(\cdot), i \in \mathbb{I}_M$ is bounded below, the sequence of costs $\{\Psi(u_{s_1}^t, \ldots, u_{s_M}^t)\}$ converges. The proof for monotonicity and convergence is identical to the

proof for Lemma 4.4 and is omitted. Let $\Psi(u^t_{s_1}, \ldots, u^t_{s_M}) \to \Psi^\infty$. Because an equality constraint that couples input targets of different subsystems is included in the target optimization (Equation (6.4)), optimality at convergence may not apply for Algorithm 6.1. Define the limit set $S^\infty = \{((x_{s_{ii}}, u_{s_i}),\ i \in \mathbb{I}_M) \mid \Psi(\cdot) = \Psi^\infty\}$. It can be shown that the sequence $(x^t_{s_{ii}}, u^t_{s_i}),\ i \in \mathbb{I}_M$ (generated by Algorithm 6.1) converges to a point $\left((x^\infty_{s_{ii}}, u^\infty_{s_i}),\ i \in \mathbb{I}_M\right) \in S^\infty$. The targets $(x^\infty_{s_{ii}}, u^\infty_{s_i}),\ i \in \mathbb{I}_M$ may be different from $(x^*_{s_i}, u^*_{s_i}),\ i \in \mathbb{I}_M$, the optimal state and input targets obtained using a centralized target calculation (Venkat, 2006). However, $z^{\mathrm{sp}}_i = C_i x^\infty_{s_i} + C^d_d \widehat{d}_i$ and $(I - A_i)x^\infty_{s_i} = B_i u^\infty_{s_i} + \sum_{j \neq i}^M W_{ij} u^\infty_{s_j} + B^d_i \widehat{d}_i,\ i \in \mathbb{I}_M$.

6.2.1 Initialization

At time $k = 0$, Algorithm 6.1 is initialized with any feasible $(x^0_{s_{ii}}(0), u^0_{s_i}(0)),\ i \in \mathbb{I}_M$. Let Algorithm 6.1 be terminated after $t \in \mathbb{I}_+, t \leq t_{\max}$ iterates. For the nominal or constant disturbance case, the steady-state pair $(x^t_{s_i}(k), u^t_{s_i}(k)),\ i \in \mathbb{I}_M$ is a feasible initial guess for the distributed target optimization problem (Equation (6.4)) at time $k + 1$. Using monotonicity and convergence properties for Algorithm 6.1, $(x^t_{s_i}(k), u^t_{s_i}(k))$ converges to $(x^\infty_{s_i}, u^\infty_{s_i}),\ i \in \mathbb{I}_M$ as $k \to \infty$.

6.3 Offset-free control with FC-MPC

The regulation problem described in Chapters 4 and 5 assumed the input and output targets are at the origin. The targets may of course need to be nonzero while tracking nonzero setpoints or rejecting nonzero constant disturbances. To achieve offset-free control in the above scenarios, a target calculation is performed and the target shifted states, inputs and outputs are used in the regulator. In the FC-MPC framework, Algorithm 6.1 may be terminated at in-

termediate iterates. For a large, networked system, the number of local measurements usually exceeds the number of subsystem inputs. Also, only the CVs typically have setpoints. Offset-free control can, therefore, be achieved for only the CVs. The choice of regulator parameters is restricted to enable offset-free control in the CVs. Accordingly, the stage cost for subsystem i is defined as

$$\begin{aligned} L_i(\omega_i, \nu_i) &= \frac{1}{2}\left[\|z_i - z_{s_i}^t\|_{Q_{z_i}}^2 + \|u_i - u_{s_i}^t\|_{R_i}^2\right] \\ &= \frac{1}{2}\left[{\omega_i}' Q_i \omega_i + {\nu_i}' R_i \nu_i\right] \end{aligned}$$

in which t is the number of distributed target calculation iterates, $\widehat{\omega}_i = \widehat{x}_i - x_{s_i}^t$, $\nu_i = u_i - u_{s_i}^t$, $z_i^{\mathrm{sp}} = H_i C_i x_{s_i}^t + H_i C_i^d \widehat{d}_i$, $Q_{z_i}, R_i > 0$, $Q_i = {C_i}' {H_i}' Q_{z_i} H_i C_i$ and $\left((z_i^{\mathrm{sp}}, \widehat{d}_i),\ i \in \mathbb{I}_M\right) \in \mathbb{D}_T$. The cost function for subsystem i (Equation (4.5)) is rewritten as $\phi_i(\widehat{\boldsymbol{\omega}}_i(k), \boldsymbol{\nu}_i(k); \widehat{\omega}_i(k))$, where $\widehat{\boldsymbol{\omega}}_i(k) = [\omega_i(k+1|k)', \omega_i(k+2|k)', \ldots]'$, $\boldsymbol{\nu}_i(k) = [\nu_i(k)', \nu_i(k+1|k), \ldots]$ and $\widehat{\omega}_i(k+j+1|k) = A_i\widehat{\omega}_i(k+j|k) + B_i\nu_i(k+j|k) + \sum_{s\neq i}^{M} W_{is}\nu_s(k+j|k)$, $0 \leq j$. It can be shown under the assumption $Q_{z_i} > 0$ that $(A_i, Q_i^{1/2})$ (with Q_i defined as above) is detectable if and only if $(A_{ii}, H_i C_{ii})$ is detectable. Let $\overline{\boldsymbol{\nu}}_i(k) = [\nu_i(k)', \ldots, \nu_i(k+N-1|k)]'$. The FC-MPC optimization problem for subsystem i is given by Equation (4.9), in which each $\overline{\boldsymbol{u}}_j$, $\widehat{x}_j$, $j \in \mathbb{I}_M$ replaced by $\overline{\boldsymbol{\nu}}_j$, $\widehat{\omega}_j$, respectively.

For the augmented subsystem model (see Section 6.1), detectability of $(\widetilde{A}_i, \widetilde{C}_i)$ implies that a steady-state estimator gain $\widetilde{\mathcal{L}}_i$ exists such that $\widetilde{A}_i - \widetilde{A}_i\widetilde{\mathcal{L}}_i\widetilde{C}_i$ is stable. We have $\widetilde{e}_i^+ = (\widetilde{A}_i - \widetilde{A}_i\widetilde{\mathcal{L}}_i\widetilde{C}_i)\widetilde{e}_i$ and $\widetilde{Z}_i = \widetilde{\mathcal{L}}_i\widetilde{C}_i(\widetilde{A}_i - \widetilde{A}_i\widetilde{\mathcal{L}}_i\widetilde{C}_i)$, where $\widetilde{e}_i$ is the estimate error for the augmented subsystem model. Let $\mu_s^t = [x_{s_1}^t, \ldots, x_{s_M}^t]$. The evolution of the perturbed augmented system

is given by

$$\begin{pmatrix} \widehat{x}_i \\ \widehat{d}_i \end{pmatrix}^+ = \widetilde{A}_i \begin{pmatrix} \widehat{x}_i \\ \widehat{d}_i \end{pmatrix} + \widetilde{B}_i u_i^p(\widehat{\mu} - \mu_s^t, 0) + \sum_{j \neq i}^{M} \widetilde{W}_{ij} u_j^p(\widehat{\mu} - \mu_s^t, 0) + \widetilde{Z}_i \widetilde{e}_i, \tag{6.6a}$$

$$\widetilde{e}_i^+ = (\widetilde{A}_i - \widetilde{A}_i \widetilde{\mathcal{L}}_i \widetilde{C}_i)\widetilde{e}_i, \quad z_i^{\mathrm{sp},+} = z_i^{\mathrm{sp}}, \qquad i \in \mathbb{I}_M, \tag{6.6b}$$

in which $\widehat{\mu} - \mu_s^t = [\widehat{x}_1 - x_{s_1}^t, \ldots, \widehat{x}_M - x_{s_M}^t]$. Define

$$\begin{aligned} \widetilde{\mathbb{D}}_C = \Big\{ \Big((\widehat{x}_i, \widehat{d}_i, \widetilde{e}_i, z_i^{\mathrm{sp}}),\ i \in \mathbb{I}_M \Big) \mid \Big((\widehat{x}_i^+, \widehat{d}_i^+, \widetilde{e}_i^+, z_i^{\mathrm{sp},+}),\ i \in \mathbb{I}_M \Big) \in \widetilde{\mathbb{D}}_C \\ \Big((z_i^{\mathrm{sp}}, \widehat{d}_i),\ i \in \mathbb{I}_M \Big) \in \mathbb{D}_T, \quad \widehat{x}_i - x_{s_i}(z_i^{\mathrm{sp}}, \widehat{d}_i) \in \mathbb{D}_{R_i},\ i \in \mathbb{I}_M \Big\} \quad \text{domain of controller} \end{aligned} \tag{6.7}$$

In Equation (6.7), $(\widehat{x}_i^+, \widehat{d}_i^+, \widetilde{e}_i^+, z_i^{\mathrm{sp},+}), i \in \mathbb{I}_M$ is calculated using Equation (6.6). The set $\widetilde{\mathbb{D}}_C$ is positively invariant. The set $\widetilde{\mathbb{D}}_C$ represents the maximal positively invariant stabilizable set for distributed MPC (with target calculation, state estimation and regulation) under constant disturbances and time invariant setpoints.

Let $(x_{s_i}^t, u_{s_i}^t)$ represent the state and input targets obtained for subsystem $i \in \mathbb{I}_M$ after $t \in \mathbb{I}_+, t \leq t_{\max}$ Algorithm 6.1 iterations. Let FC-MPC based on either Theorem 5.1 (stable systems) or Theorem 5.2 (unstable systems) be used for distributed regulation. Let Algorithm 4.1 (p. 43) be terminated after $p \in \mathbb{I}_+, p \leq p_{\max}$ iterations. The target shifted perturbed closed-loop system evolves according to

$$\widehat{\omega}_i(k+1) = A_i \widehat{\omega}_i(k) + B_i \nu_i(k) + \sum_{j \neq i}^{M} W_{ij} \nu_j(k) + \tau_i^x \widetilde{Z}_i \widetilde{e}_i(k), \tag{6.8a}$$

$$\widetilde{e}_i(k+1) = (\widetilde{A}_i - \widetilde{A}_i \widetilde{\mathcal{L}}_i \widetilde{C}_i)\widetilde{e}_i(k), \quad i \in \mathbb{I}_M, \tag{6.8b}$$

in which $\widehat{\omega}_i = \widehat{x}_i - x^t_{s_i}$, $\nu_i = u^p_i(\widehat{\mu} - \mu^t_s, 0)$ and $\tau^x_i[\widehat{x}_i{}', \widehat{d}_i{}']' = \widehat{x}_i$, $i \in \mathbb{I}_M$. The input injected into subsystem $i \in \mathbb{I}_M$, after p Algorithm 4.1 iterations and t Algorithm 6.1 iterations, is $u^p_i(\widehat{\mu} - \mu^t_s, 0) + u^t_{s_i}$. The evolution of the disturbance estimate follows $\widehat{d}_i(k+1) = \widehat{d}_i(k) + \tau^d_i \widetilde{Z}_i \widetilde{e}_i(k)$, $i \in \mathbb{I}_M$, where $\tau^d_i[\widehat{x}_i{}', \widehat{d}_i{}']' = \widehat{d}_i$.

Theorem 6.1. *Let $(\widetilde{A}_i, \widetilde{C}_i)$, $i \in \mathbb{I}_M$ be detectable. The origin is an exponentially stable equilibrium for the target shifted perturbed closed-loop system given by Equation* (6.8)*, in which $(\widetilde{A}_i - \widetilde{A}_i \widetilde{\mathcal{L}}_i \widetilde{C}_i)$, $i \in \mathbb{I}_M$ is stable and $\widehat{\omega}_i(0) = \widehat{x}_i(0) - x^t_{s_i}(0)$, for all $p = 1, 2, \ldots, p_{\max}$, $t = 1, 2, \ldots, t_{\max}$, $k \geq 0$ for all $\Big((\widehat{x}_i(0), \widehat{d}_i(0), \widetilde{e}_i(0), z^{\mathrm{sp}}_i),\ i \in \mathbb{I}_M\Big) \in \widetilde{\mathbb{D}}_C$.*

A proof is given in Appendix 6.6.5.

Lemma 6.5. *Let Assumptions 6.1 to 6.3 hold. Let $(\widetilde{A}_i, \widetilde{C}_i), i \in \mathbb{I}_M$ be detectable, $(\widetilde{A}_i - \widetilde{A}_i \widetilde{\mathcal{L}}_i \widetilde{C}_i), i \in \mathbb{I}_M$ be stable and $n_{d_i} = n_{y_i}, \forall\, i \in \mathbb{I}_M$. Also, let the input inequality constraints for each subsystem $i \in \mathbb{I}_M$ be inactive at steady state. If the closed-loop system under FC-MPC is stable, the FC-MPCs with subsystem-based estimators, local disturbance models and distributed target calculation, track their respective CV setpoints with zero offset at steady state i.e., $z^{\mathrm{sp}}_i = H_i y_i(\infty)$, where $y_i(\infty)$ is the output for subsystem i at steady state, and H_i satisfies Assumption 6.4.*

A proof is given in Appendix 6.6.6.

6.4 Examples

6.4.1 Two reactor chain with nonadiabatic flash

A plant consisting of two continuous stirred tank reactors (CSTRs) and a nonadiabatic flash is considered. A schematic of the plant is shown in Figure 6.2. A description of the plant as well

as MVs, CVs for each control subsystem was provided in Section 4.7.2 (p. 58). A linear model for the plant is obtained by linearizing the plant around the steady state corresponding to the maximum yield of B. The constraints on the manipulated variables are given in Table 6.1. In the decentralized and distributed MPC frameworks, there are 3 MPCs, one each for the two CSTRs and one for the nonadiabatic flash. In the centralized MPC framework, a single MPC controls the entire plant.

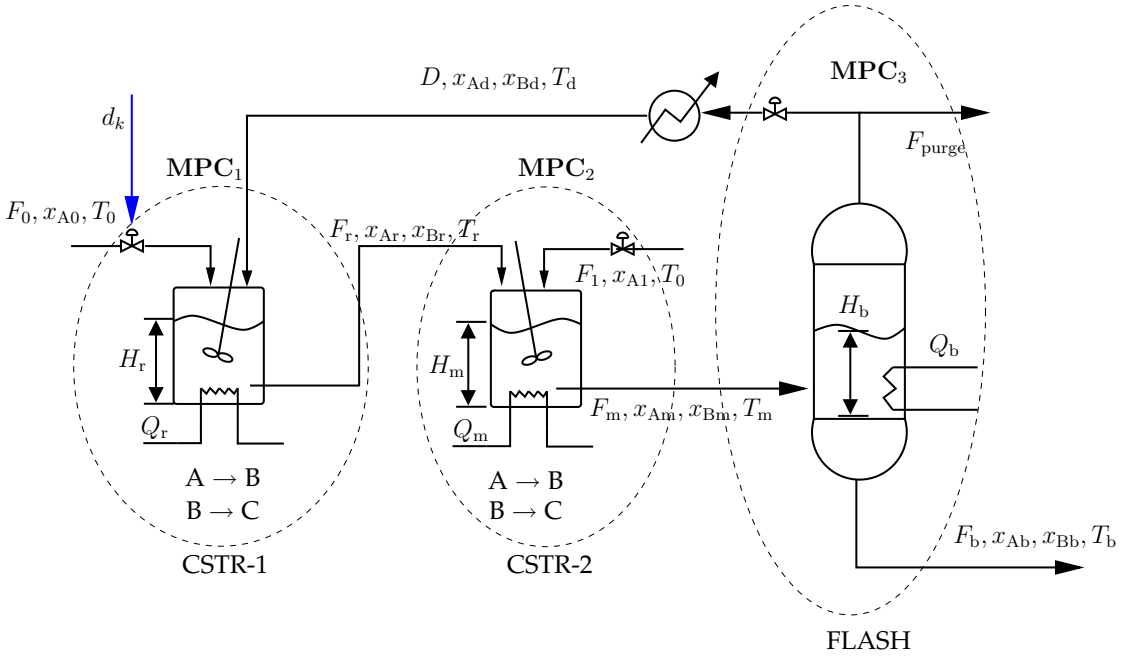

Figure 6.1: Two reactor chain followed by nonadiabatic flash. Vapor phase exiting the flash is predominantly A. Exit flows are a function of the level in the reactor/flash.

Table 6.1: Input constraints for Example 6.4.1. The symbol Δ represents a deviation from the corresponding steady-state value.

$-0.2 \leq \Delta F_0 \leq 0.2$	$-8 \leq \Delta Q_r \leq 8$
$-0.04 \leq \Delta F_1 \leq 0.04$	$-2 \leq \Delta Q_r \leq 2$
$-0.25 \leq \Delta D \leq 0.25$	$-8 \leq \Delta Q_b \leq 8$

The states and integrating disturbances for each subsystem are estimated from measurements. Input disturbance models are used to eliminate steady-state offset. The distur-

bance models employed in each MPC framework are given in Table 6.2. In the FC-MPC framework under output feedback, the states of each subsystem are estimated from local measurements using subsystem-based Kalman filters. The steady-state targets are calculated in a distributed manner employing Algorithm 6.1. Two cases for distributed target calculation in the FC-MPC framework are considered. In the first case, distributed target calculation algorithm is terminated after 10 iterates, and in the second case, the distributed target calculation algorithm is iterated to convergence.

Table 6.2: Disturbance models (decentralized, distributed and centralized MPC frameworks) for Example 6.4.1.

$B_{11}^d = 0.5 \begin{bmatrix} B_{11} & B_{11} \end{bmatrix}$ $B_1^d = \text{vec}(B_{11}^d, 0, 0)$	$C_{11}^d = 0.5 I_{z_1}$ $C_1^d = 0.5 I_{z_1}$
$B_{22}^d = 0.5 \begin{bmatrix} B_{22} & B_{22} \end{bmatrix}$ $B_2^d = \text{vec}(0, B_{22}^d, 0)$	$C_{22}^d = 0.5 I_{z_2}$ $C_2^d = 0.5 I_{z_2}$
$B_{33}^d = 0.5 \begin{bmatrix} B_{33} & B_{33} \end{bmatrix}$ $B_3^d = \text{vec}(0, 0, B_{33}^d)$ $B_d = \text{diag}(B_1^d, B_2^d, B_3^d)$	$C_{33}^d = 0.5 I_{z_3}$ $C_3^d = 0.5 I_{z_3}$ $C_d = \text{diag}(C_1^d, C_2^d, C_3^d)$

A feed flowrate disturbance affects CSTR-1 from time = 30. As a result of this flowrate disturbance, the feed flowrate to CSTR-1 is increased by 5% (relative to the steady-state value). The disturbance rejection performance of centralized MPC, decentralized MPC and FC-MPC is investigated for the described disturbance scenario. A control horizon $N = 15$ is used for each MPC. The sampling interval is 1.5. The weight for each CV is 10 and the weight for each MV is 1. The performance of the different MPC frameworks rejecting the feed flowrate disturbance to CSTR-1 (d_k in Figure 6.1) is shown in Figure 6.2. The resulting temperature and cooling duty changes are small and therefore, not shown. The closed-loop control costs incurred by the different MPC frameworks are compared in Table 6.3.

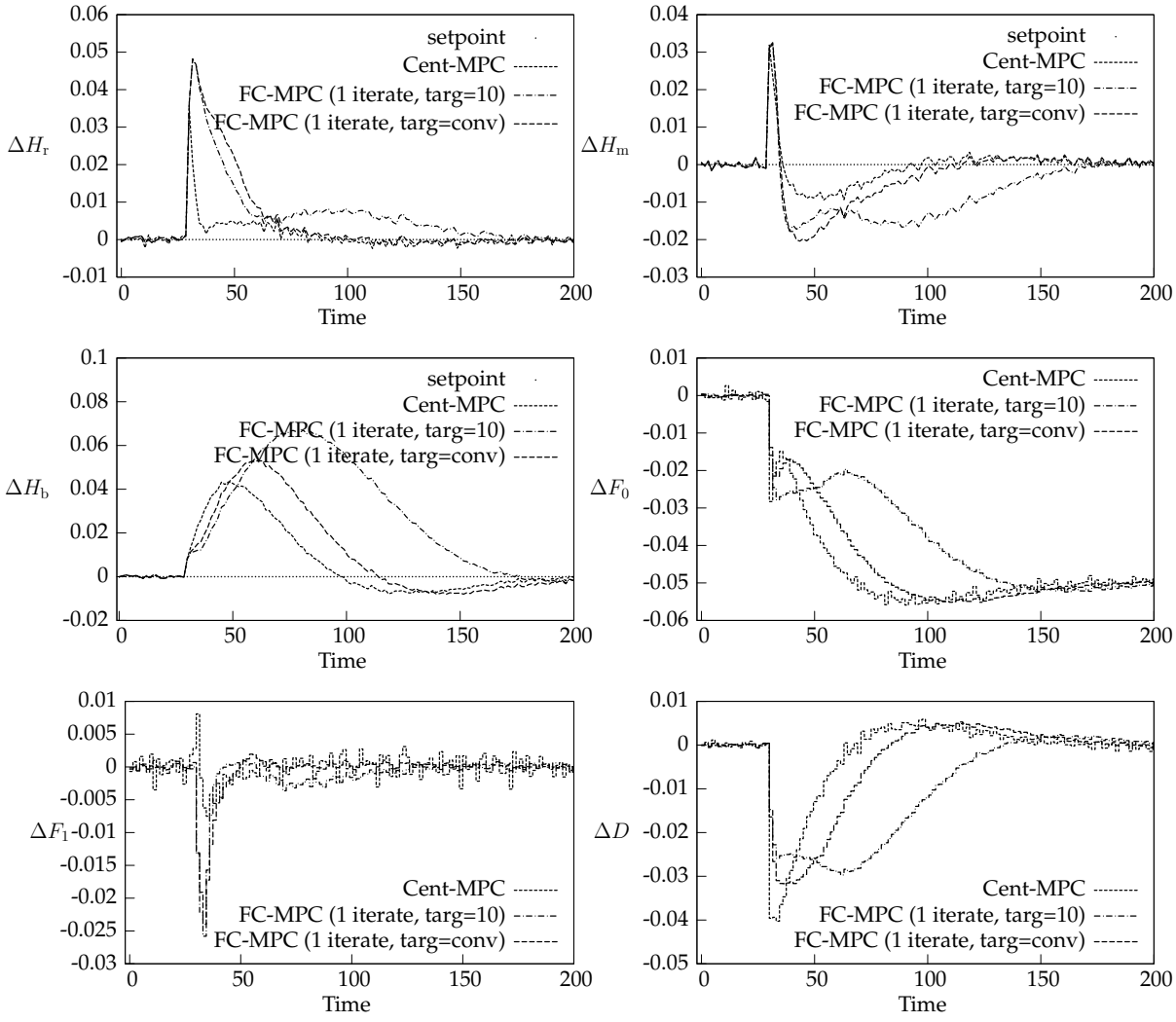

Figure 6.2: Disturbance rejection performance of centralized MPC, decentralized MPC and FC-MPC. For the FC-MPC framework, 'targ=conv' indicates that the distributed target calculation algorithm is iterated to convergence. The notation 'targ=10' indicates that the distributed target calculation algorithm is terminated after 10 iterates.

Under decentralized MPC, the feed flowrate disturbance causes closed-loop instability. With the centralized MPC and FC-MPC frameworks, the system is able to reject the feed flowrate disturbance. The feed flowrate disturbance d_k to CSTR-1 causes an increase in H_r. In the FC-MPC framework, MPC-1 lowers F_0 to compensate for the extra material flow into CSTR-1. MPC-3 cooperates with MPC-1 and helps drive H_r back to its setpoint by decreasing

the recycle flowrate D to CSTR-1. The initial increase in H_{r} results in an increase in F_{r}, which in turn increases H_{m}. Subsequently, F_{m} and hence, H_{b} also increase. To compensate for the initial increase in H_{m}, MPC-2 decreases F_1. The initial increase in H_{b} is due to an increase in F_{m} (MPC-2) and decrease in D (MPC-3). To lower H_{b}, MPC-3 subsequently increases D. MPC-1 continues to steer H_{r} to its setpoint, in spite of an increase in D (by MPC-3), through a corresponding (further) reduction in F_0.

Table 6.3: Closed-loop performance comparison of centralized MPC, decentralized MPC and FC-MPC. The distributed target calculation algorithm (Algorithm 6.1) is used to determine steady-state subsystem input, state and output target vectors in the FC-MPC framework.

	$\Lambda_{\mathrm{cost}} \times 10^2$	$\Delta\Lambda_{\mathrm{cost}}\%$
Cent-MPC	4.14	$--$
Decent-MPC	∞	$--$
FC-MPC (1 iterate, targ=conv)	5.74	38.4%
FC-MPC (1 iterate, targ=10)	7.12	71.2%
FC-MPC (10 iterates, targ=conv)	4.17	0.62%
FC-MPC (10 iterates, targ=10)	5.93	43.2%

The performance loss incurred under FC-MPC, with Algorithm 4.1 (p. 43) terminated after 1 cooperation-based iterate and Algorithm 6.1 (p. 132) terminated after 10 iterates, is $\sim 72\%$ relative to centralized MPC. If the distributed target calculation algorithm (Algorithm 6.1) is iterated to convergence, the performance loss w.r.t centralized MPC reduces to $\sim 38\%$. If both Algorithm 4.1 and Algorithm 6.1 are terminated after 10 iterates, the performance loss relative to centralized MPC is $\sim 43\%$. Iterating Algorithm 6.1 to convergence, and terminating Algorithm 4.1 after 10 iterates, improves performance to within 1% of centralized MPC performance.

6.4.2 Irrigation Canal Network

The key to agricultural productivity and the goal of irrigation canal networks is to provide the right quantity of water at the right time and place. The need for flexible, "on-demand" schedules has motivated the need for automatic control of these water networks. Each irrigation canal consists of a fixed number of reaches that are interconnected through control gates. In reach i (see Figure 6.3), the downstream water level y_i is controlled by manipulating the gate opening u_i. However the water level y_i is also affected by the gate opening u_{i+1} in reach $i+1$. At the downstream end of each reach i, an off-take discharge Q_i supplies water to meet local demands. For each reach, the off-take discharge is dictated by the local water demand. Variations in the off-take discharge are disturbances for the system. Representative publications on the modeling and control of canal networks include Garcia, Hubbard, and de Vries (1992); Sawadogo, Malaterre, and Kosuth (1995); Sawadogo, Faye, and Mora-Camino (2001); Schuurmans, Bosgra, and Brouwer (1995). In most cases, different sections of the irrigation canal network are administered by different governing bodies (e.g., different municipalities) making centralized control impractical and unrealizable. It is also well known that a decentralized control formulation in which each canal reach employs a local controller to regulate water levels may realize poor closed-loop performance as a result of the interaction between adjacent reaches.

The example we consider here is canal 2 of the test cases established by the ASCE task committee Clemmens et al. (1998). The canal under consideration is fed by a constant water level reservoir at its head. The canal consists of 8 interconnected reaches with the downstream end closed (Figure 6.4). Between times 0.5 hrs and 2.5 hrs, an off-take discharge disturbance

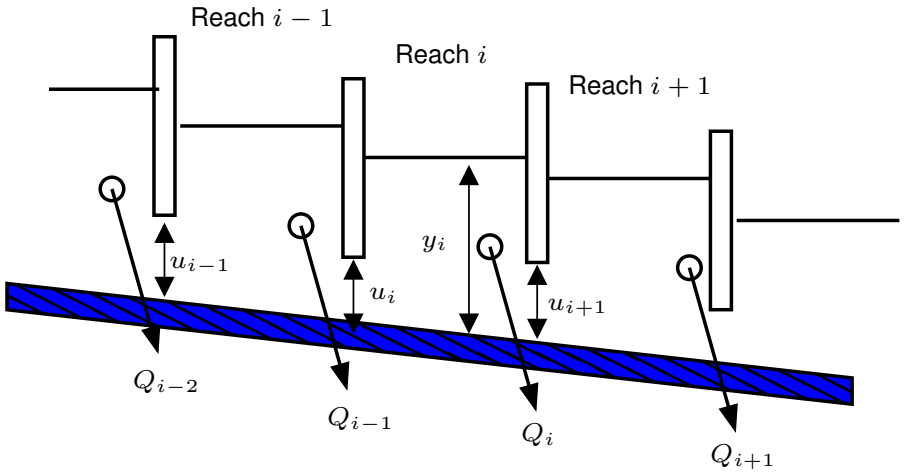

Figure 6.3: Structure of an irrigation canal. Each canal consists of a number of interconnected reaches.

affects reaches $1-8$. During this time, reaches $1, 3, 5$ and 7 experience an off-take discharge disturbance of $2.5\ \mathrm{m}^3/\mathrm{min}$ and simultaneously, a discharge disturbance $-2.5\ \mathrm{m}^3/\mathrm{min}$ affects reaches $2, 4, 6$ and 8. The closed-loop performance of centralized MPC, decentralized MPC and FC-MPC, rejecting this off-take discharge disturbance is assessed. The permissible gate opening Δu_i (deviation w.r.t to steady state) for each reach $i, i \in \{1, 2, \ldots, 8\}$ is given in Table 6.4. In the decentralized and distributed MPC frameworks, there are 8 MPCs, one for each reach.

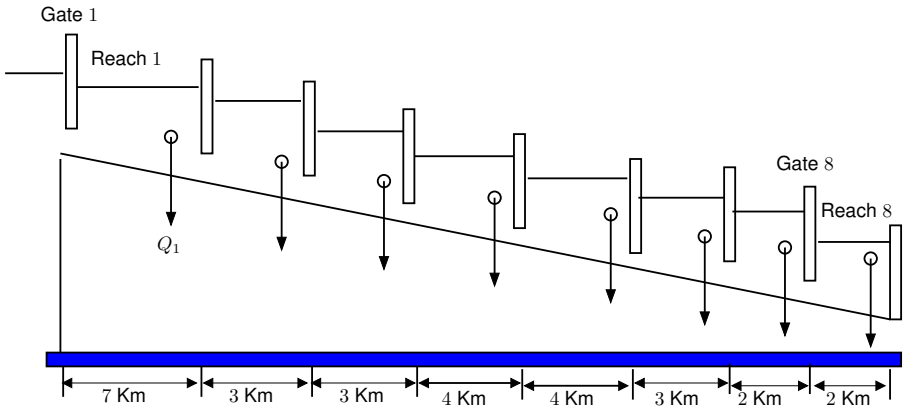

Figure 6.4: Profile of ASCE test canal 2 Clemmens et al. (1998). Total canal length 28 km.

Table 6.4: Gate opening constraints for Example 6.4.2. The symbol Δ denotes a deviation from the corresponding steady-state value.

$-1.5 \leq \Delta u_1 \leq 1.5$
$-1.5 \leq \Delta u_2 \leq 1.5$
$-0.4 \leq \Delta u_3 \leq 0.4$
$-0.1 \leq \Delta u_4 \leq 0.1$
$-0.25 \leq \Delta u_5 \leq 0.25$
$-0.15 \leq \Delta u_6 \leq 0.15$
$-0.1 \leq \Delta u_7 \leq 0.1$
$-0.1 \leq \Delta u_8 \leq 0.1$

In each MPC framework, the state and constant disturbances are estimated using a steady-state Kalman filter. Output disturbance models are used to eliminate steady-state offset due to the unmeasured off-take discharge disturbances. In the FC-MPC formulation, the distributed target calculation algorithm is iterated to convergence. The performance of the different MPC frameworks, rejecting the off-take discharge disturbance in reaches $3, 4$ and 6 is shown in Figure 6.5. For each MPC, $Q_i = 10, R_i = 0.1, i \in \{1, 2, \ldots, 8\}$. The sampling rate is 0.1 hrs (or every 6 minutes) and the control horizon N for each MPC is 30.

Table 6.5: Closed-loop performance of centralized MPC, decentralized MPC and FC-MPC rejecting the off-take discharge disturbance in reaches $1 - 8$. The distributed target calculation algorithm (Algorithm 6.1) is iterated to convergence.

	$\Lambda_{\text{cost}} \times 10^4$	$\Delta\Lambda_{\text{cost}}\%$
Cent-MPC	4.58	$--$
Decent-MPC	7.81	70.5%
FC-MPC (1 iterate)	7.41	61.8
FC-MPC (2 iterates)	5.88	28.4%
FC-MPC (5 iterates)	4.99	9.0%
FC-MPC (10 iterates)	4.72	3.1%

Based on control costs calculated at steady state (Table 6.5), decentralized MPC leads to a control performance loss of $\sim 70\%$ relative to centralized MPC performance. In the FC-MPC framework, with Algorithm 4.1 terminated after a single iterate, the incurred performance loss w.r.t centralized MPC is $\sim 62\%$. If the FC-MPC algorithm is terminated after 2 cooperation-

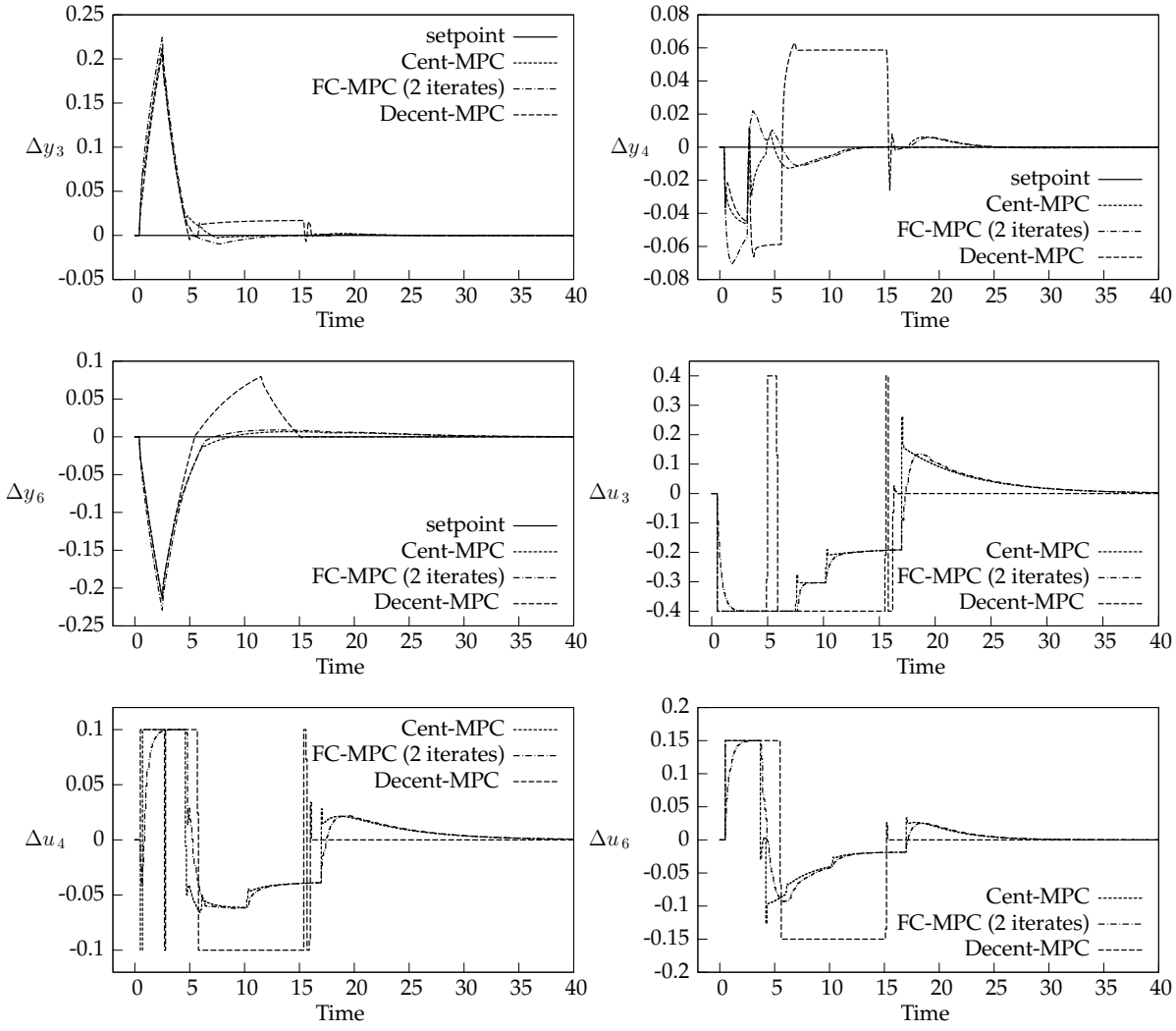

Figure 6.5: Control of ASCE test canal 2. Water level control for reaches $3, 4$ and 6.

based iterates, the performance loss reduces to $\sim 29\%$ of centralized MPC performance. The FC-MPC framework achieves performance that is within 3.25% of centralized MPC performance if 10 cooperation-based iterates are allowed.

6.5 Discussion and conclusions

A disturbance modeling framework that uses local integrating disturbances was employed. This choice of local integrating disturbances is motivated by considerations of simplicity and practical convenience. A simple rank test is shown to be necessary and sufficient to verify suitability of postulated subsystem disturbance models. Next, a distributed target calculation algorithm that computes steady-state input, state and output targets at the subsystem level was described. All iterates generated by the distributed target calculation algorithm are feasible steady states. Also, the target cost function is monotonically nonincreasing with iteration number. The attributes described above allow intermediate termination of the distributed target calculation algorithm. A maximal positively invariant stabilizable set for distributed MPC, with state estimation, target calculation and regulation, was defined. This positively invariant set characterizes system state, disturbance, estimate error and setpoint quadruples for which the system can be stabilized using the distributed MPC control law. Zero-offset control at steady state is established for the set of subsystem-based MPCs under the assumption that the input constraints for each subsystem are inactive at the target. An interesting result, which follows from Lemmas 6.1 and 6.5, is that disturbance models that achieve zero-offset steady-state control under decentralized MPC are sufficient to realize offset-free steady-state control in the FC-MPC framework.

Two examples were presented to illustrate the effectiveness of the proposed output feedback distributed MPC framework. In the first example, control of a chemical plant was investigated in the presence of a feed flowrate disturbance. The states of each subsystem were estimated using a Kalman filter. Decentralized MPC is unable to reject the disturbance and re-

sults in closed-loop unstable behavior. Here, the distributed target calculation algorithm was terminated at an intermediate iterate. The FC-MPC framework is able to reject the disturbance and achieves zero offset steady-state control performance for all values of distributed target calculation and distributed regulation algorithm iteration numbers. Next, the distributed MPC framework was employed to reject a discharge disturbance in a simulated irrigation canal network. Local output disturbance models were used to achieve zero offset steady-state control performance. The first iterate was observed to improve performance marginally ($\sim 9\%$) compared to decentralized MPC. A second iterate, however, leads to a significant improvement in performance ($\sim 41\%$) compared to decentralized MPC. For this example, it is recommended that at least two iterates per sampling interval be performed.

Implementation. The structure of the FC-MPC framework with distributed estimation, local disturbance modeling and distributed target calculation is shown in Figure 6.6. Each subsystem uses a local Kalman filter to estimate its states and integrating disturbances. The only external information required are the inputs injected into the interconnected subsystems. This input information, however, is available in the regulator and consequently, no information transfer between subsystems is needed at the estimation level. Next, the targets are calculated locally. The input target is relayed to all interacting subsystems after each iterate (Algorithm 6.1). The setpoint for subsystem CVs, local integrating disturbances and the decentralized state target need not be communicated to interconnected subsystems. For distributed regulation, the subsystem state estimate is communicated to all interconnected subsystems at each k; the recalculated input trajectories are broadcast to interacting subsystems after each iterate. For each subsystem $i \in \mathbb{I}_M$, by setting $w_i = 1$ and $w_j = 0,\ \forall\, j \neq i$ in the FC-MPC

regulator optimization problem and $A_{ij}, C_{ij} = 0, \forall\, j \neq i$ in the estimator and the target optimization problem, and by switching off the communication between the subsystems, we revert to decentralized MPC.

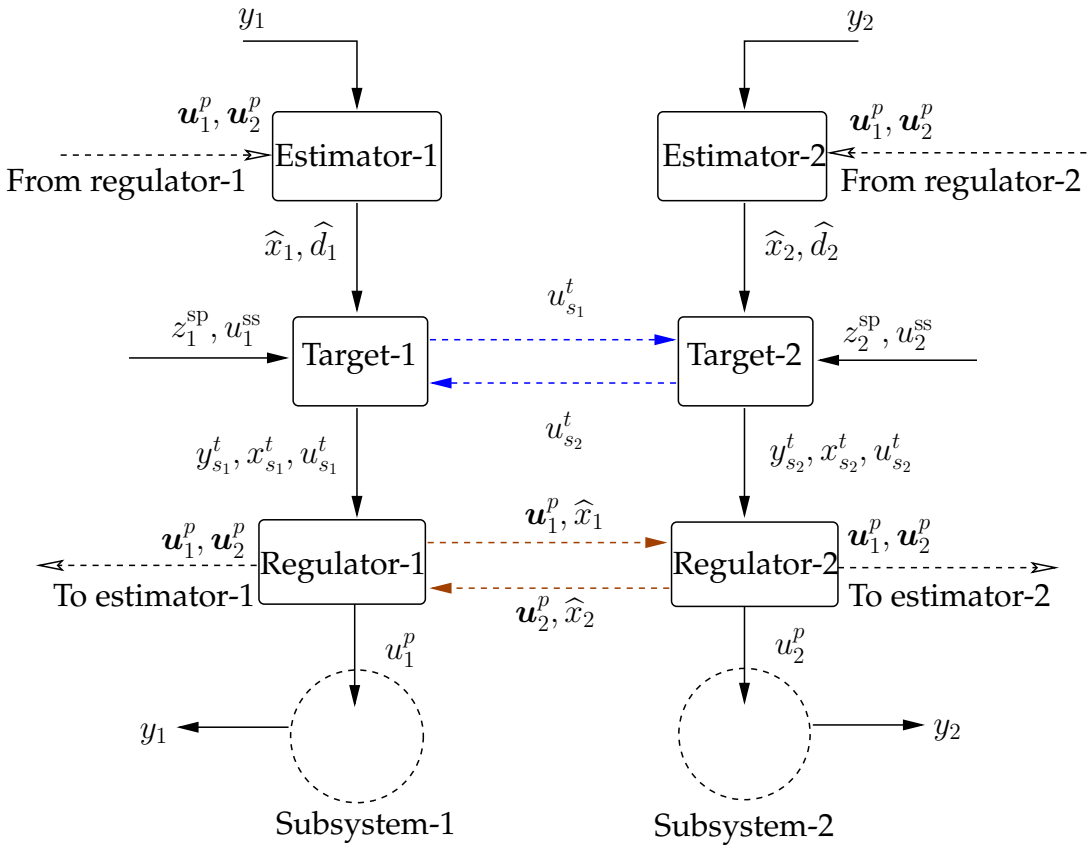

Figure 6.6: Structure of output feedback FC-MPC.

6.6 Appendix

6.6.1 Proof for Lemma 6.1

Proof for Lemma 6.1. From the Hautus Lemma for detectability (Sontag, 1998, p. 318), $(\widetilde{A}_i, \widetilde{C}_i)$ is detectable iff $\operatorname{rank}\left(\begin{bmatrix}\lambda I - \widetilde{A}_i \\ \widetilde{C}_i\end{bmatrix}\right) = n_i + n_{d_i},\ \forall\, |\lambda| \geq 1$. Define

$$\mathcal{S}(\lambda) = \begin{bmatrix} \lambda I - A_{ii} & -\mathcal{B}_{ii}^d & \\ & (\lambda-1)I & \\ C_{ii} & \mathcal{C}_i^d & \mathbb{C}_s \\ & & \lambda I - \mathbb{A}_s \end{bmatrix} = \begin{bmatrix} \lambda I - \widetilde{A}_{ii} & 0 \\ \widetilde{C}_{ii} & \mathbb{C}_s \\ 0 & \lambda I - \mathbb{A}_s \end{bmatrix},$$

in which $\widetilde{A}_{ii} = \begin{bmatrix} A_{ii} & \mathcal{B}_{ii}^d \\ & I \end{bmatrix}$, $\widetilde{C}_{ii}^d = \begin{bmatrix} C_{ii} & \mathcal{C}_{ii}^d \end{bmatrix}$ denote respectively, the A and C matrix for the augmented decentralized model and

$$\mathbb{A}_s = \operatorname{diag}(A_{i1}, \ldots, A_{i(i-1)}, A_{i(i+1)}, \ldots, A_{iM}),\ A_{ii} \in \mathbb{R}^{n_{ii} \times n_{ii}}$$

$$\mathbb{C}_s = [C_{i1}, \ldots, C_{i(i-1)}, C_{i(i+1)}, \ldots, C_{iM}]$$

Consider $|\lambda| \geq 1$. Since $\mathbb{A}_s$ is stable, the columns of $\lambda I - \mathbb{A}_s$ are independent. Hence the columns of $\begin{bmatrix} 0 \\ \mathbb{C}_s \\ \lambda I - \mathbb{A}_s \end{bmatrix}$ are independent. By assumption, $\begin{bmatrix} \lambda I - \widetilde{A}_{ii} \\ \widetilde{C}_{ii} \end{bmatrix}$ has $n_{ii} + n_{d_i}$ independent columns. Due to the positions of the zeros, $\mathcal{S}(\lambda)$ has $n_i + n_{d_i}$ independent columns. To complete

the proof, we note that

$$\begin{bmatrix} \lambda I - \widetilde{A}_i \\ \widetilde{C}_i \end{bmatrix} = U\mathcal{S}(\lambda)U,$$

in which U is an unitary matrix. Consequently, the columns of $\begin{bmatrix} \lambda I - \widetilde{A}_i \\ \widetilde{C}_i \end{bmatrix}$ are independent for all $|\lambda| \geq 1$. Hence, $(\widetilde{A}_i, \widetilde{C}_i)$ is detectable. □

6.6.2 Proof for Lemma 6.2

Proof. $(\widetilde{A}_i, \widetilde{C}_i)$ **detectable** $\implies$ **rank condition.** From the Hautus lemma for detectability (Sontag, 1998, p. 318), the columns of $\begin{bmatrix} \lambda I - A_i & -B_i^d \\ & (\lambda - 1)I \\ C_i & C_i^d \end{bmatrix}$ are independent for any λ satisfying $|\lambda| \geq 1$. The columns of $\begin{bmatrix} \lambda I - A_i & -B_i^d \\ C_i & C_i^d \end{bmatrix}$ are, therefore, independent for all $|\lambda| \geq 1$. The choice $\lambda = 1$ gives the desired rank relationship.

rank condition $\implies$ $(\widetilde{A}_i, \widetilde{C}_i)$ **detectable.** The assumed rank condition implies the columns of $\begin{bmatrix} I - A_i & -B_i^d \\ C_i & C_i^d \end{bmatrix}$ are independent. From Lemma 5.3, (A_i, C_i) is detectable. The columns of $\begin{bmatrix} \lambda I - A_i \\ C_i \end{bmatrix}$ are, therefore, independent for all $|\lambda| \geq 1$. Consider $\begin{bmatrix} \lambda I - A_i & -B_i^d \\ C_i & C_i^d \\ 0 & (\lambda - 1)I \end{bmatrix}$. For

$\lambda \neq 1$, the columns of $(\lambda - 1)I$ are independent and therefore, the columns of $\begin{bmatrix} -B_i^d \\ C_i^d \\ (\lambda-1)I \end{bmatrix}$ are independent. Due to the position of the zero, the columns of $\begin{bmatrix} \lambda I - A_i & -B_i^d \\ C_i & C_i^d \\ 0 & (\lambda-1)I \end{bmatrix}$ are independent. For $\lambda = 1$, we know that the columns of $\begin{bmatrix} I - A_i & -B_i^d \\ C_i & C_i^d \end{bmatrix}$ are independent (by assumption). Hence, $(\widetilde{A}_i, \widetilde{C}_i)$ is detectable, as claimed. □

6.6.3 Existence and uniqueness for a convex QP

Theorem 6.2. *Let $f(x) = \frac{1}{2}x'Qx + c'x + d$ and $-\infty < \underline{f} \leq f(x), \forall\, x$. Consider the constrained QP*

$$\min_x f(x) \quad \textit{subject to} \quad Ax = b, x \in \mathbb{X}$$

in which $x \in \mathbb{R}^n$, $b \in \mathbb{R}^p$, $Q \geq 0$, $A \in \mathbb{R}^{p\times n}$, and $\mathbb{X} \subseteq \mathbb{R}^{s\times n}$ is polygonal. Let the feasible region be nonempty. Let $\mathrm{rank}(A) = p$. A solution to this problem exists. Furthermore, the solution is unique if rank $\begin{bmatrix} Q \\ A \end{bmatrix} = n$.

Proof. Since the feasible region is nonempty and polygonal, and the QP is bounded below by $\underline{f}$, a solution exists (Frank and Wolfe, 1956). Suppose that there exists two solutions x and $\overline{x}$. Let $w = x - \overline{x}$. We have $Aw = A(x - \overline{x}) = b - b = 0$. The normal cone optimality conditions [3]

[3] We note that $f(\cdot)$ is a proper convex function in the sense of (Rockafellar, 1970, p. 24), that the relative interior of $Ax = b, x \in \mathbb{X}$ is nonempty and that the feasible region defined by $(Ax = b, x \in \mathbb{X}) \subset \mathrm{dom}(f(\cdot))$. The normal cone optimality conditions are, therefore, both necessary and sufficient (Rockafellar, 1970, Theorem 27.4, p. 270).

for x and $\overline{x}$ gives

$$(y - x)'(Qx + c) \geq 0 \quad \forall\, y | Ay = b, y \in \mathbb{X}$$

$$(y - \overline{x})'(Q\overline{x} + c) \geq 0 \quad \forall\, y | Ay = b, y \in \mathbb{X}$$

Substituting $y = \overline{x}$ in the first equation and $y = x$ in the second equation, we have $w'Qx \leq -w'c$ and $w'Q\overline{x} \geq -w'c$. These two equations together imply $w'Q\overline{x} \geq w'Qx$, and therefore $w'Qw \leq 0$. Because $Q \geq 0$, $w'Qw \geq 0$. Hence, $w'Qw = 0$, which implies $Qw = 0$. Using $Aw = 0$ and full column rank for $\begin{bmatrix} Q \\ A \end{bmatrix}$, we have that the only solution for $\begin{bmatrix} Q \\ A \end{bmatrix} w = 0$ is $w = 0$. This gives $x = \overline{x}$. □

6.6.4 Proof for Lemma 6.4

Proof for Lemma 6.4. Reverse implication. The objective function for the optimization problem of Equation (6.4) can be rewritten as

$$\Psi_i(\cdot) = \frac{1}{2} \begin{pmatrix} x_{s_{ii}} \\ u_{s_i} \end{pmatrix}' \begin{bmatrix} 0 & \\ & R_{u_i} \end{bmatrix} \begin{pmatrix} x_{s_{ii}} \\ u_{s_i} \end{pmatrix} + \begin{bmatrix} 0 \\ -R_{u_i} u_i^{\text{ss}} \end{bmatrix}' \begin{pmatrix} x_{s_{ii}} \\ u_{s_i} \end{pmatrix} + \frac{1}{2} u_i^{\text{ss}\prime} R_{u_i} u_i^{\text{ss}}.$$

From Theorem 6.2, the solution to the target optimization problem for each $i \in \mathbb{I}_M$ is unique if the columns of $\begin{bmatrix} 0 & \\ & R_{u_i} \\ I - A_{ii} & -B_{ii} \\ H_i C_{ii} & \end{bmatrix}$, $i \in \mathbb{I}_M$ are independent. Because $R_{u_i} > 0, i \in \mathbb{I}_M$, and due

to the position of the zeros, the columns of $\begin{bmatrix} 0 & \\ & R_{u_i} \\ I - A_{ii} & -B_{ii} \\ H_i C_{ii} & \end{bmatrix}$ are independent if and only if the columns of $\begin{bmatrix} I - A_{ii} \\ H_i C_{ii} \end{bmatrix}$ are independent.

Forward implication. Let $(x_{s_{ii}}^{*(t)}, u_{s_i}^{*(t)}),\ i \in \mathbb{I}_M$ be unique and assume $\operatorname{rank} \begin{bmatrix} I - A_{ii} \\ H_i C_{ii} \end{bmatrix} < n_{ii}$ for some $i \in \mathbb{I}_M$. By assumption, there exists v such that $\begin{bmatrix} I - A_{ii} \\ H_i C_{ii} \end{bmatrix} v = 0,\ v \neq 0$. The pair $(x_{s_{ii}}^{*(t)} + v, u_{s_i}^{*(t)})$ achieves the optimal cost $\frac{1}{2}\|u_i^{\mathrm{ss}} - u_{s_i}^{*(t)}\|_{R_{u_i}}^2$ and

$$\begin{bmatrix} I - A_{ii} \\ H_i C_{ii} \end{bmatrix} (x_{s_{ii}}^{*(t)} + v) = \begin{bmatrix} B_{ii} u_{s_i}^{*(t)} + B_{ii}^d \widehat{d}_i \\ z_i^{\mathrm{sp}} - H_i C_i^d \widehat{d}_i - \sum_{j \neq i}^{M} \left(\overline{g}_{ij} u_{s_j}^{t-1} + \overline{h}_{ij} d_i \right) \end{bmatrix},$$

which contradicts uniqueness of $x_{s_{ii}}^{*(t)}$. □

6.6.5 Proof for Theorem 6.1

Proof for Theorem 6.1. Since $(\widetilde{A}_i, \widetilde{C}_i),\ i \in \mathbb{I}_M$ is detectable, an estimator gain $\widetilde{\mathcal{L}}_i$ exists such that $(\widetilde{A}_i - \widetilde{A}_i \widetilde{\mathcal{L}}_i \widetilde{C}_i)$ is stable for each $i \in \mathbb{I}_M$. From the positive invariance of $\widetilde{\mathbb{D}}_C$,

$$\left((z_i^{\mathrm{sp}}, \widehat{d}_i(k)),\ i \in \mathbb{I}_M \right) \in \mathbb{D}_T$$

for all $k \geq 0$. The target optimization problem (Equation (6.4)) is feasible for each $i \in \mathbb{I}_M$ for all $k \geq 0$. In Theorem 5.4 (Appendix 5.9.2), replace e_i by $\widetilde{e}_i$, Z_i by $\tau_i^x \widetilde{Z}_i$, $\mathcal{A}_i^L$ by $(\widetilde{A}_i - \widetilde{A}_i\widetilde{\mathcal{L}}_i\widetilde{C}_i)$, $\widehat{x}_i$ by $\widehat{\omega}_i$, and $\widehat{\mu}$ by $\widehat{\mu} - \mu_s^t$. The model matrices $(A_i, B_i, \{W_{ij}\}_{j\neq i}, C_i)$ are unaltered. From the definition of $\mathbb{D}_T$, and positive invariance of $\widetilde{\mathbb{D}}_C$, feasible perturbation trajectories $\boldsymbol{v}_i, i \in \mathbb{I}_M$ exist such that $v_i(j) + (u_i^p(\widehat{\mu} - \mu_s^t, j) + u_{s_i}^t) \in \Omega_i, j \geq 1, i \in \mathbb{I}_M$. Existence of σ_r for stable and unstable systems can be demonstrated using arguments used to prove Theorem 5.1 and Theorem 5.2, respectively (with appropriate variable name changes as outlined above). Invoking Theorem 5.4 completes the proof. □

6.6.6 Proof for Lemma 6.5

Proof. From Lemma 5.3, (A_i, C_i) is detectable and (A_i, B_i) is stabilizable. Zero offset steady-state tracking performance can be established in the FC-MPC framework through an extension of either (Muske and Badgwell, 2002, Theorem 4) or (Pannocchia and Rawlings, 2002, Theorem 1). Let Algorithm 4.1 be terminated after $p \in \mathbb{I}_+,\ p < \infty$ iterates. At steady state, using Lemma 4.4 we have $u_i^p(\mu(\infty), 0) = u_i^\infty(\mu(\infty), 0),\ i \in \mathbb{I}_M,\ p \in \mathbb{I}_+$. Let the targets generated by Algorithm 6.1 at steady state be $(x_{s_i}^\infty, u_{s_i}^\infty), \forall\, i \in \mathbb{I}_M$ (see Section 6.2.1). Let $(\widehat{x}_i(\infty), \widehat{d}_i(\infty))$ denote an estimate of the subsystem state and integrating disturbance vectors at steady state. From Equation (6.2), we have

$$\widehat{x}_i(\infty) = A_i\widehat{x}_i(\infty) + B_i u_i^\infty(\mu(\infty), 0) + \sum_{j\neq i} W_{ij} u_j^\infty(\mu(\infty), 0) + B_{\mathrm{d}_i}\widehat{d}_i(\infty)$$
$$+ \mathcal{L}_{\mathrm{x}_i}\left(y_i(\infty) - C_i\widehat{x}_i(\infty) - C_i^d\widehat{d}_i(\infty)\right)$$
$$\text{and} \qquad \widehat{d}_i(\infty) = \widehat{d}_i(\infty) + \mathcal{L}_{\mathrm{d}_i}\left(y_i(\infty) - C_i\widehat{x}_i(\infty) - C_i^d\widehat{d}_i(\infty)\right)$$

Invoking (Pannocchia and Rawlings, 2002, Lemma 3) for each subsystem $i \in \mathbb{I}_M$ gives $\mathcal{L}_{\mathrm{d}_i}$ is full rank. Hence, $y_i(\infty) = C_i\widehat{x}_i(\infty) + C_i^d\widehat{d}_i(\infty)$ and $(\widehat{x}_i(\infty) - x_{s_i}^\infty) = A_i(\widehat{x}_i(\infty) - x_{s_i}^\infty) + B_i(u_i^\infty(\mu(\infty),0) - u_{s_i}^\infty) + \sum_{j\neq i} W_{ij}(u_j^\infty(\mu(\infty),0) - u_{s_j}^\infty)$, $i \in \mathbb{I}_M$. Because all input constraints are inactive at steady state, there exists $\mathcal{K}$ such that the solution to Algorithm 4.1 at steady state is

$$\begin{bmatrix} (u_1^\infty(\mu(\infty),0) - u_{s_1}^\infty) \\ (u_2^\infty(\mu(\infty),0) - u_{s_2}^\infty) \\ \vdots \\ (u_M^\infty(\mu(\infty),0) - u_{s_M}^\infty) \end{bmatrix} = -\mathcal{K} \begin{bmatrix} (\widehat{x}_1(\infty) - x_{s_1}^\infty) \\ (\widehat{x}_2(\infty) - x_{s_2}^\infty) \\ \vdots \\ (\widehat{x}_M(\infty) - x_{s_M}^\infty) \end{bmatrix}$$

Stability of the closed-loop system requires $A_{\mathrm{cm}} - B_{\mathrm{cm}}\mathcal{K}$ to be a stable matrix. Therefore,

$$(I - A_{\mathrm{cm}} - B_{\mathrm{cm}}\mathcal{K}) \begin{bmatrix} (\widehat{x}_1(\infty) - x_{s_1}^\infty) \\ (\widehat{x}_2(\infty) - x_{s_2}^\infty) \\ \vdots \\ (\widehat{x}_M(\infty) - x_{s_M}^\infty) \end{bmatrix} = 0,$$

which gives $(\widehat{x}_i(\infty) - x_{s_i}^\infty) = 0,\ i \in \mathbb{I}_M$ and $u_i^\infty(\mu(\infty),0) = u_{s_i}^\infty$. This implies

$$\begin{aligned} H_i y_i(\infty) - z_i^{\mathrm{sp}} &= \Big(H_i C_i \widehat{x}_i(\infty) + H_i C_i^d \widehat{d}_i(\infty)\Big) - \Big(H_i C_i x_{s_i}^\infty + H_i C_i^d \widehat{d}_i(\infty)\Big) \\ &= H_i C_i (\widehat{x}_i(\infty) - x_{s_i}^\infty) \\ &= 0 \end{aligned}$$

□

6.6.7 Simplified distributed target calculation algorithm for systems with nonintegrating decentralized modes

For systems without integrating decentralized modes, an alternative, simpler, distributed target calculation algorithm can be derived by eliminating the decentralized states $x_{s_{ii}},\ i \in \mathbb{I}_M$ from the optimization problem of Equation (6.4). For subsystem $i \in \mathbb{I}_M$ at iterate t, the following QP is solved

$$u_{s_i}^{*(t)} \in \quad \arg \quad \min_{u_{s_i}} \quad \frac{1}{2}(u_i^{\mathrm{ss}} - u_{s_i})' R_{u_i} (u_i^{\mathrm{ss}} - u_{s_i}) \tag{6.9a}$$

$$\text{subject to} \quad u_{s_i} \in \Omega_i \tag{6.9b}$$

$$\overline{g}_{ii} u_{s_i} = z_i^{\mathrm{sp}} - H_i C_i^d \widehat{d}_i - \sum_{j \neq i}^{M} \overline{g}_{ij} u_{s_j}^{t-1} - \sum_{j=1}^{M} \overline{h}_{ij} \widehat{d}_i, \tag{6.9c}$$

in which $\overline{g}_{ii} = H_i C_{ii} (I - A_{ii})^{-1} B_{ii}$ and $\overline{h}_{ii} = H_i C_{ii} (I - A_{ii})^{-1} B_{ii}^d$. The following algorithm may be employed to determine steady-state targets.

Algorithm 6.2. Given $\left(u_{s_i}^0,\ z_i^{\mathrm{sp}}, u_i^{\mathrm{ss}}\right), R_{u_i} > 0,\ i \in \mathbb{I}_M, t_{\max} > 0, \epsilon > 0$

$t \leftarrow 1, \kappa_i \leftarrow \Gamma\epsilon, \Gamma \gg 1$

while $\kappa_i > \epsilon$ for some $i \in \mathbb{I}_M$ and $t \leq t_{\max}$

do $\forall\ i \in \mathbb{I}_M$

Determine $u_{s_i}^{*(t)}$ from Equation (6.9)

$u_{s_i}^t \leftarrow w_i u_{s_i}^{*(t)} + (1 - w_i) u_{s_i}^{t-1}$

$\rho_i \leftarrow \|u_{s_i}^t - u_{s_i}^{t-1}\|$

Transmit $u_{s_i}^t$ to each interconnected subsystem $j \in \mathbb{I}_M,\ j \neq i$

end (do)

$t \leftarrow t+1$

end (while)

At each iterate t, the target state vector $x_{s_i}^t,\ \forall\ i \in \{1, M\}$ is calculated as $x_{s_i}^t \leftarrow (I - A_i)^{-1}B_i u_{s_i}^t + \sum_{j \neq i}(I - A_i)^{-1}W_{ij}u_{s_j}^t + (I - A_i)^{-1}B_i^d \widehat{d}_i$. For $\left((z_i^{\text{sp}}, \widehat{d}_i), i \in \mathbb{I}_M\right) \in \mathbb{D}_T$, a feasible solution for the target optimization problem above exists for each $i \in \mathbb{I}_M$. Because the objective is strictly convex, the solution to Equation (6.9) is unique.

Chapter 7

Distributed MPC with partial cooperation [1]

In the FC-MPC framework, the objective of each local MPC is known to all interconnected subsystem MPCs. This global sharing of objectives may not be desirable in some situations. As a simple example, consider the system depicted in Figure 7.1. Assume that the y_2 setpoint is unreachable and that u_2 is at its bound constraint. From a practitioner's standpoint, it is desirable to manipulate input u_1, to the largest extent possible, to achieve all future y_1 setpoint changes. Conversely, it is desirable to manipulate u_2 to track setpoint changes in y_2. By definition, a decentralized control structure is geared to realize this operational objective. However, the resulting closed-loop performance may be quite poor. Centralized control, on the other hand, utilizes an optimal combination of the inputs u_1, u_2 to achieve the new setpoint. The centralized MPC framework, though optimal, may manipulate both u_1 and u_2 significantly.

[1] **Portions of this chapter appear in Venkat, Rawlings, and Wright (2005a).**

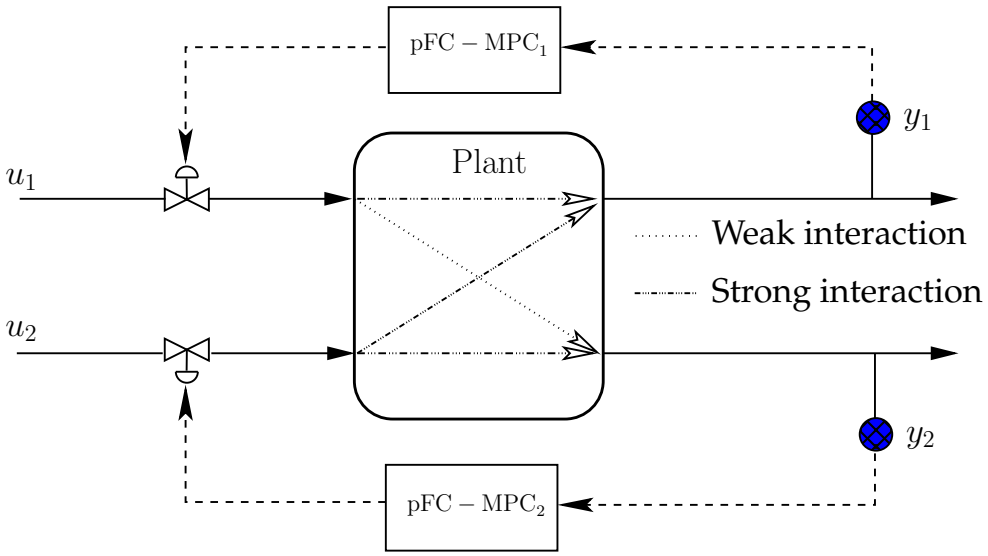

Figure 7.1: 2×2 interacting system. Effect of input u_1 on output y_2 is small compared to $u_1 - y_1, u_2 - y_1$ and $u_2 - y_2$ interactions.

7.1 Partial feasible cooperation-based MPC (pFC-MPC)

To track the setpoint of y_1 exclusively with input u_1 and setpoint of y_2 primarily with u_2, the concept of partial cooperation is employed. This approach of designing controllers to explicitly handle operational objectives is similar in philosophy to the modular multivariable controller (MMC) approach of Meadowcroft et al. (1992). The principal goal is to meet operational objectives, even if the resulting controller performance is not optimal. The partial cooperation-based MPC for subsystem 1 ($\mathrm{pFC}-\ \mathrm{MPC}_1$) manipulates u_1 but has access only to the local objective ϕ_1 that quantifies the cost of control action u_1 on y_1. The partial cooperation-based MPC for subsystem 2 ($\mathrm{pFC}-\mathrm{MPC}_2$) manipulates u_2 and retains access to both subsystem objectives ϕ_1 and ϕ_2. Therefore, $\mathrm{pFC}-\mathrm{MPC}_2$ evaluates the cost of control action u_2 on a global level $i.e.$, its effect on both system outputs y_1 and y_2.

7.1.1 Geometry of partial cooperation

To illustrate the behavior of partial feasible cooperation-based MPC (pFC-MPC), we consider a simple example consisting of two subsystems with cost functions $\Phi_1(\cdot)$ and $\Phi_2(\cdot)$. Both $\Phi_1(\cdot)$ and $\Phi_2(\cdot)$ are obtained by eliminating the states $x_i, i = 1, 2$ from the cost functions $\phi_1(\cdot)$ and $\phi_2(\cdot)$ respectively using the corresponding composite model equations (see p. 39, Chapter 4). The point p in Figure 7.2 represents the Pareto optimal solution for $w_1 = w_2 = \frac{1}{2}$. If both MPCs cooperate completely (FC-MPC), we know from Lemma 4.5 (p. 45) that the FC-MPC algorithm (Algorithm 4.1, p. 43) converges to p. Under partial cooperation, $\mathrm{pFC-MPC}_1$ utilizes only $\Phi_1(\cdot)$ to determine its control actions. For cases in which the $u_1 - y_2$ interactions are weak compared to the $u_1 - y_1$, $u_2 - y_2$ and $u_2 - y_1$ interactions, the pFC-MPC framework is observed to converge to a point in a neighborhood of p. In Figure 7.2, p' represents the converged solution obtained using partial cooperation. The displacement of p' relative to p is observed to be a function of the strength of the $u_1 - y_2$ interactions relative to the other interaction pairs. If the $u_1 - y_2$ interactions are identically zero, p and p' coincide. Unlike FC-MPC, there are no convergence guarantees for pFC-MPC however. For situations in which $u_1 - y_2$ interactions are much stronger than the other interaction pairs, partial cooperation may not converge. Employing pFC-MPCs for cases in which the $u_1 - y_2$ interactions are significant is a bad design strategy; FC-MPCs should be used instead.

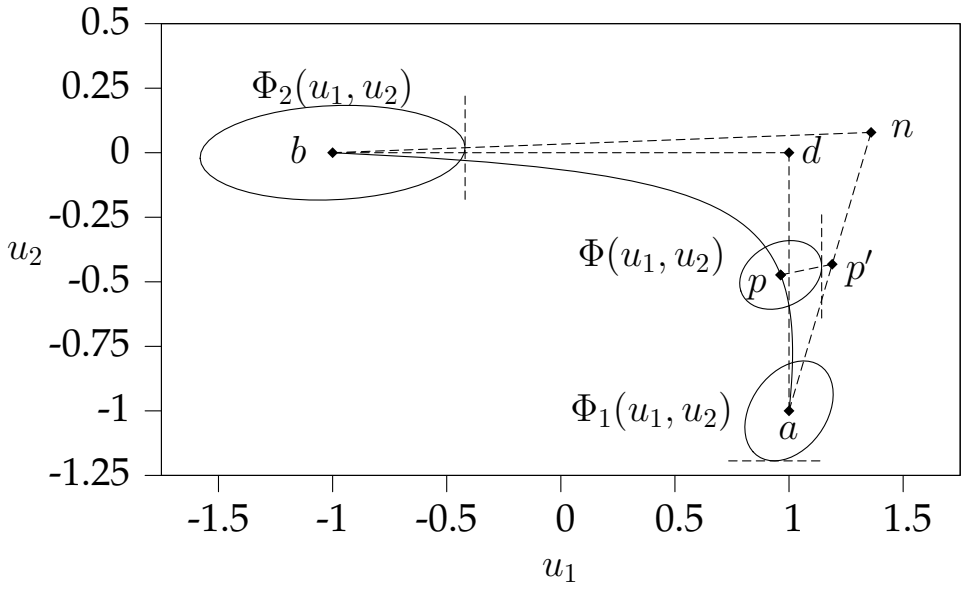

Figure 7.2: Geometry of partial cooperation. p denotes the Pareto optimal solution. p' represents the converged solution with partial cooperation. d is the solution obtained under decentralized MPC. n is the Nash equilibrium.

7.1.2 Example

We consider an example in which the $u_1 - y_2$ interaction is weak compared to the $u_1 - y_1$, $u_2 - y_2$ and $u_2 - y_1$ interactions. The plant is described by

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

in which

$$G_{11} = \frac{1.26}{9.6s+1} \qquad G_{12} = \frac{0.5(-5s+1)}{(18s+1)(5s+1)}$$

$$G_{21} = \frac{1.5(-20s+1)}{(10.5s+1)(20s+1)} \qquad G_{22} = \frac{3.9}{5.4s+1}$$

The input constraints are $|u_1| \leq 1.25$ and $|u_2| \leq 0.3$. The sampling rate is 1.5. An initial (reachable) setpoint change is made to y_2. Tracking errors for output y_2 are weighted 50 times more than tracking errors for output y_1. The regulator penalties on u_1 and u_2 are $R_1 = 1$ and $R_2 = 0.01$ respectively.

At times 4 and 7, unreachable y_2 setpoint changes are made. For each of the new y_2 setpoints, the input u_2 is at its upper bound at steady state. The pFC-MPC algorithm is terminated after 1 iterate. The closed-loop performance of cent-MPC and pFC-MPC are shown in Figure 7.3. Cent-MPC, in violation of the desired mode of operation, manipulates input u_1 (in addition to u_2) to track the y_2 target optimally. Since the y_1 setpoint is unchanged and $\mathrm{pFC-MPC}_1$ has access to objective ϕ_1 only, u_1 remains unaltered. To alter y_2, $\mathrm{pFC-MPC}_2$ needs to manipulate u_2. However, u_2 is already at its bound constraint and consequently, y_2 remains unchanged. Thus, the $\mathrm{pFC-MPC}$ formulation, though suboptimal, achieves desired operational objectives.

7.2 Vertical integration with pFC-MPC

In previous chapters, we were concerned with horizontal integration of the higher level subsystems' MPCs. In practice, the outputs from each MPC are almost never injected directly into the plant, but are setpoints for lower level flow controllers. Such a cascaded controller structure results in a vertical hierarchy within each subsystem. By integrating the lower level flow controllers with the higher level subsystem MPC, it may be possible to further improve systemwide control performance. Traditionally, PID controllers have been used in industry for control of fast flow loops. Recent advances in explicit MPC solution techniques (Bempo-

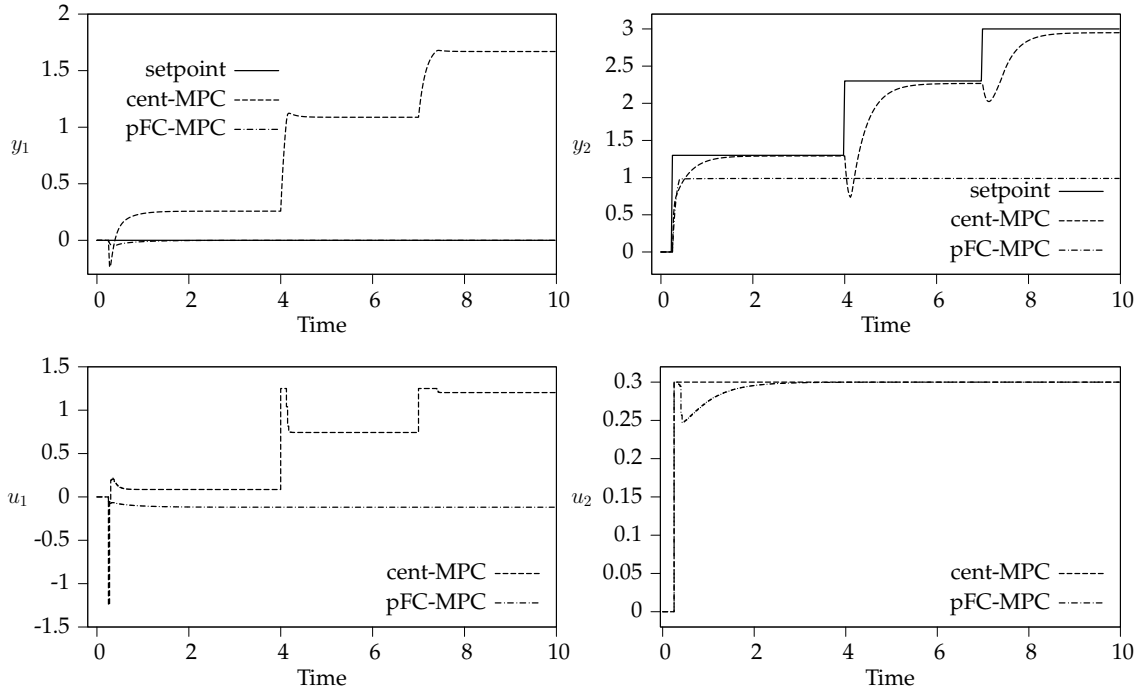

Figure 7.3: Closed-loop performance of pFC-MPC and cent-MPC for the system in Figure 7.1.

rad and Filippi, 2003; Bemporad, Morari, Dua, and Pistikopoulos, 2002; Pannocchia, Laachi, and Rawlings, 2005; Tondel, Johansen, and Bemporad, 2003) allows the evaluation of the optimal MPC control action within milliseconds for SISO systems. This development provides the practitioner with an option to replace conventional PID controllers with SISO MPCs for control of fast flow loops (Pannocchia et al., 2005). In addition, employing MPCs at each control level provides an opportunity for cooperative vertical integration within each subsystem.

An obvious choice for integrating different flow controllers (SISO MPCs) with the higher level multivariable MPC is to use FC-MPC (complete cooperation). For a large, networked system with a substantial number of these flow controllers, FC-MPC may result in an intricate network of interconnected and communicating flow controllers. By exploiting the time-scale separation between the flow control loops and the MV-CV loops directed by the higher level

MPC, this seemingly complicated network of controllers can be simplified using *partial cooperation*. The time constants for the flow loops are typically much smaller than those for the CV-MV loops controlled by the higher level MPC. A change in the valve position, for instance, has an almost instantaneous effect on the exit flowrate from the valve. The effect of a change in valve position has a significantly slower and damped effect on subsystem CVs such as temperature or concentration. In partial cooperation, each lower level flow controller optimizes its local objective. For cascade control, the objective of each flow controller is typically to manipulate the control valve opening such that the desired flowrate is achieved optimally. The desired flowrate for the flow controller is provided by the higher level FC-MPC, which uses a global objective to determine its control outputs. A schematic for the structure of partial cooperation-based cascade control is shown in Figure 7.4. Under partial cooperation, the only information exchange performed by each flow controller is with its (higher level) subsystem MPC; the different flow controllers are not required to communicate with each other.

7.2.1 Example: Cascade control of reboiler temperature

In this example, we investigate the disturbance rejection performance of pFC-MPC employed for cascade control of temperature in a distillation column reboiler. A schematic of the plant is given in Figure 7.5. To implement pFC-MPC, two interaction models are required. The first interaction model describes the effect of a change in the control valve position on the reboiler temperature. The second interaction model describes the effect of a change in the flowrate setpoint on the exit flow from the valve. These interaction models may be identified from operating data using closed-loop identification techniques (Gudi and Rawlings, 2006; Juang

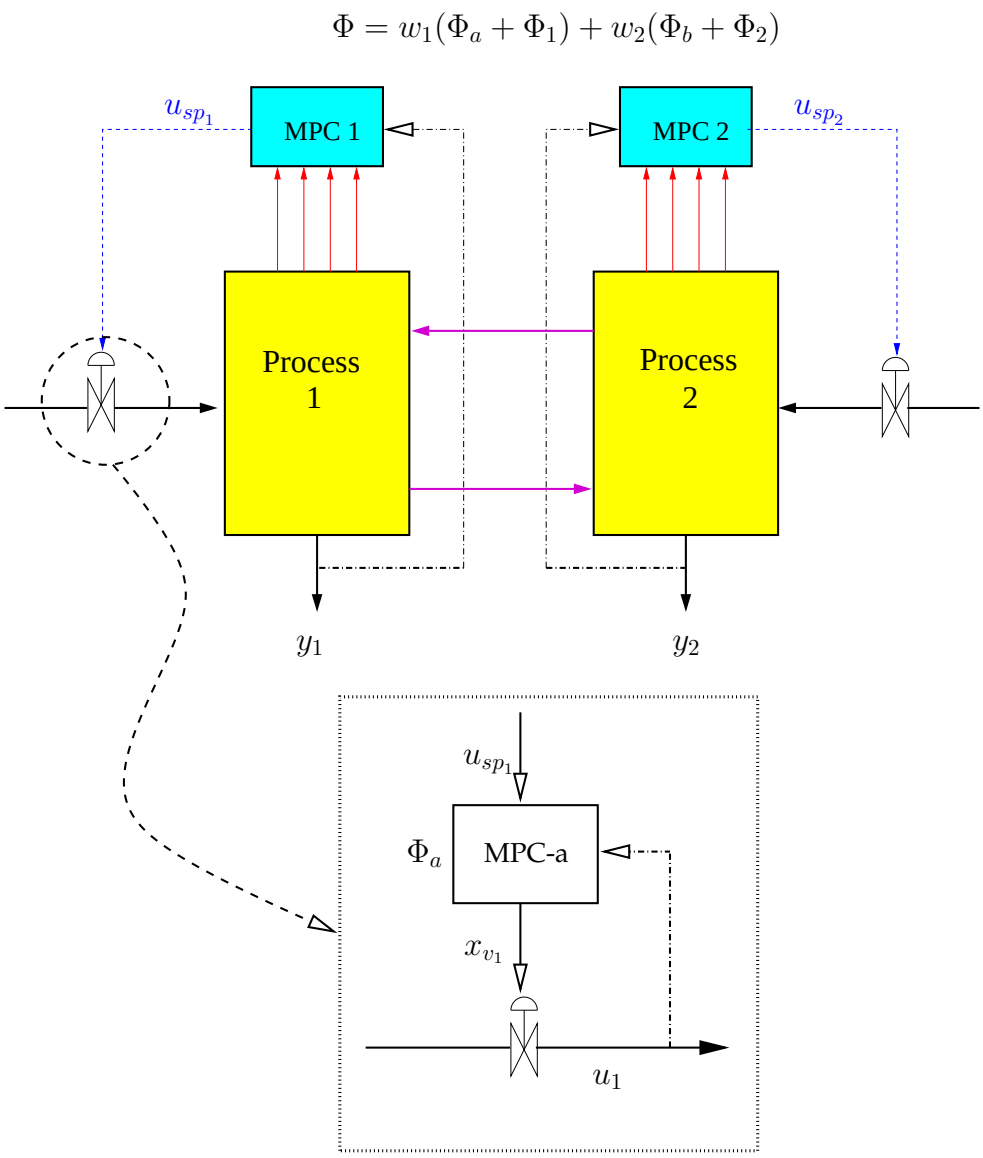

Figure 7.4: Structure for cascade control with pFC-MPC. $\Phi_i, i = 1, 2$ represents the local objective for each higher level MPC. Φ_a and Φ_b denote the local objective for the lower level MPCs a and b respectively. The overall objective is Φ. The notation $x_{v_i}, i = 1, 2$ denotes the percentage valve opening for flow control valve i. MPCs 1 and 2 use Φ to determine appropriate control outputs. MPCs a and b use Φ_a and Φ_b respectively to compute their control actions. MPC-a broadcasts trajectories to MPC-1 only. Similarly, MPC-b communicates with MPC-2 only.

and Phan, 1994; Lakshminarayanan, Emoto, Ebara, Tomida, and Shah, 2001; Verhaegen, 1993).

A complete description of the model is given below.

$$\begin{pmatrix} T \\ F \end{pmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{pmatrix} F_{sp} \\ x_v \end{pmatrix} + \begin{bmatrix} 0 \\ G_d \end{bmatrix} d$$

in which

$$G_{11} = \frac{0.25(s-20)}{(10s+1)(25s+1)} \qquad G_{12} = \frac{0.55(s-20)(s-1.5)}{(10s+1)(25s+1)(2s+1)}$$
$$G_{21} = \frac{0.75}{(5s+1)} \qquad G_{22} = \frac{2.2(s-1.5)}{2s+1}$$
$$G_d = \frac{1.5}{0.5s+1}$$

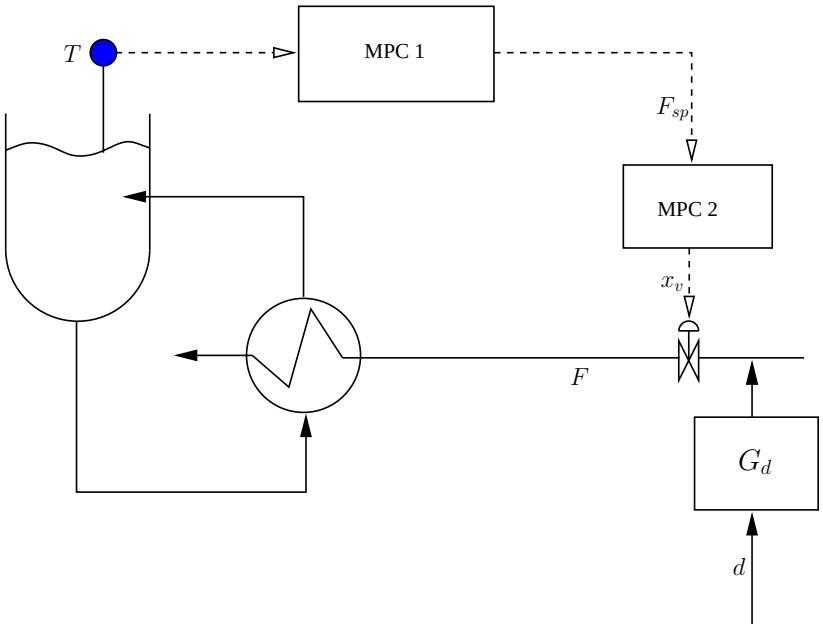

Figure 7.5: Cascade control of reboiler temperature.

The deviational flowrate from the valve is constrained as $|F| \leq 2$. At time $k = 0$, a valve pressure disturbance d of magnitude 0.1 affects the exit flowrate from the valve. The

disturbance rejection performance of pFC-MPC, when the exchange of information between the two nested MPCs is terminated after one iterate is investigated. The cascade control performance of pFC-MPC is compared against the performance of traditional cascade control, in which decentralized MPCs are used in place of PID controllers.

MPC-1 manipulates the flowrate setpoint F_{sp} to control reboiler temperature T. MPC-2 manipulates the valve opening x_v to control the exit flowrate from the valve F. The local objective for MPC-1 is to maintain T at its desired target by manipulating F_{sp}. The local objective for MPC-2 is to manipulate x_v to bring F as close as possible to F_{sp}. The higher level MPC *i.e.*, MPC-1 utilizes both its own local objective as well as the objective for MPC-2 to determine its control outputs. MPC-2, on the other hand, uses only its local objective to determine suitable control action. Cascade control performance of decentralized MPC and pFC-MPC rejecting the pressure disturbance in the valve is shown in Figure 7.6. A closed-loop performance comparison of the different MPCs is provided in Table 7.1. Cascade control with FC-MPC (1 iterate) achieves performance within 1.5% of the optimal centralized MPC performance. Cascade control with pFC-MPC (1 iterate) incurs a performance loss of about 7% relative to FC-MPC (1 iterate). Cascade control using decentralized MPCs gives unacceptable closed-loop performance.

Table 7.1: Closed-loop performance comparison of cascaded decentralized MPC, pFC-MPC and FC-MPC. Incurred performance loss measured relative to closed-loop performance of FC-MPC (1 iterate).

	$\Lambda_{\text{cost}} \times 10^2$	$\Delta\Lambda_{\text{cost}}\%$
FC-MPC (1 iterate)	2.16	$--$
Decent-MPC	82	$> 3000\%$
pFC-MPC (1 iterate)	2.3	6.6%

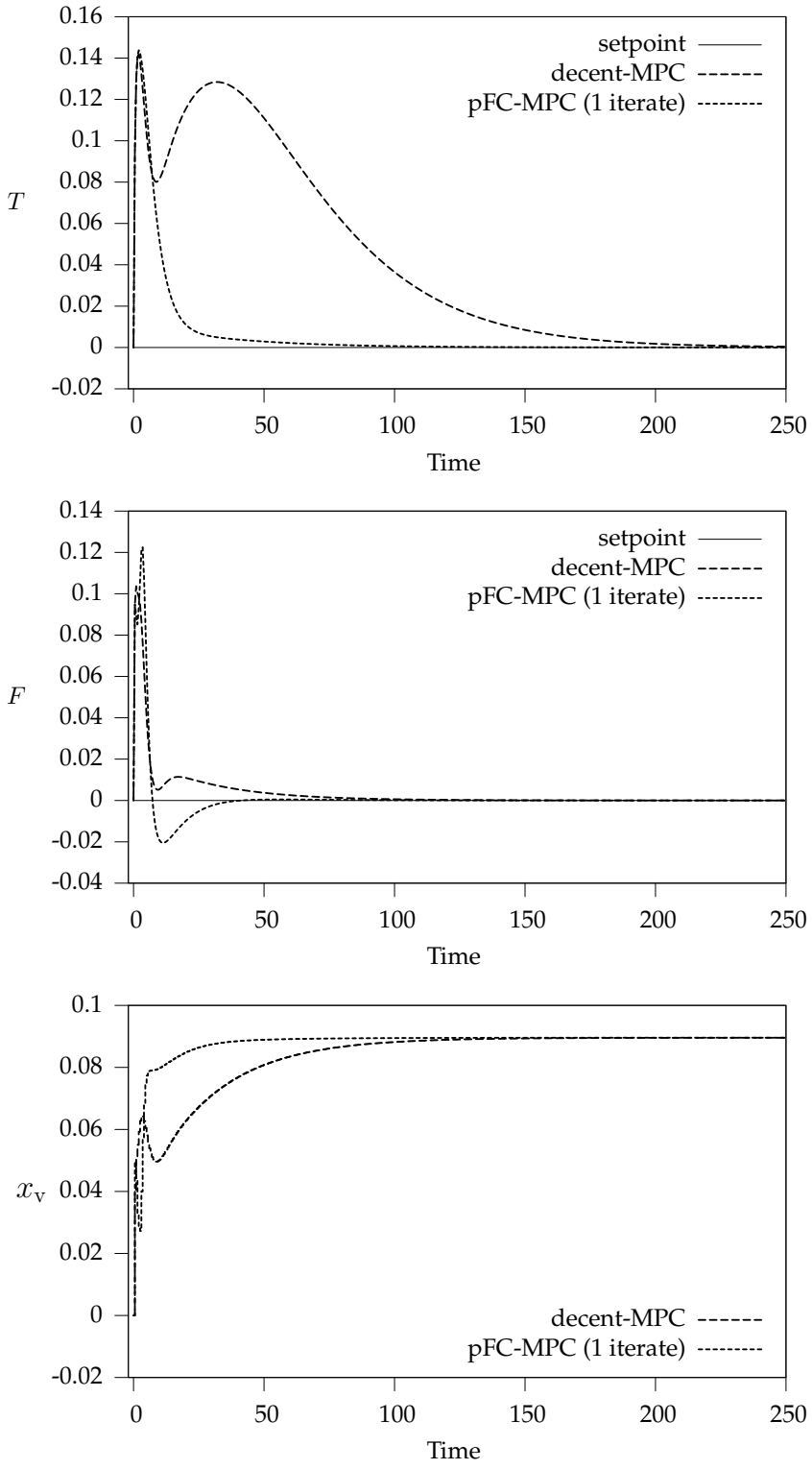

Figure 7.6: Disturbance rejection performance comparison of cascaded SISO decentralized MPCs and cascaded pFC-MPCs. Disturbance affects flowrate from valve.

7.3 Conclusions

The concept of distributed MPC with partial cooperation was introduced in this chapter. Partial cooperation is an attractive strategy for cases in which some of the interactions are significantly weaker than others. The structure of the resulting controller network is simpler, and communication requirements are reduced compared to FC-MPC. When the weak interactions are identically zero, the converged solution using partial cooperation is Pareto optimal. Two applications were considered here. In the first application, partial cooperation was used to incorporate operational objectives in distributed MPC. In the second application, partial cooperation was used to integrate lower level flow controllers with each subsystem's MPC. As an example, partial cooperation was employed to integrate controllers used for cascade control of reboiler temperature. Partial cooperation was observed to improve closed-loop performance significantly compared to conventional cascade control with decentralized SISO MPCs.

Chapter 8

Asynchronous optimization for distributed MPC.

In previous chapters, it was assumed that all MPCs perform their iterations synchronously *i.e.*, have the same computational time requirements and frequency of information exchange. Several factors determine the computational time necessary for each subsystem's MPC. These factors include model size, processor speed, hardware and software used etc. For many large networked systems, the demands on computational time may differ considerably from subsystem to subsystem. If all MPCs are forced to operate synchronously, the worst case computational time requirement for the slowest MPC is used. The essence of asynchronous optimization (for FC-MPC) is to exploit the difference in required computational times for subsystems' MPCs to further improve systemwide control performance.

To illustrate the idea behind asynchronous optimization for FC-MPC, we consider a simple example consisting of three MPCs (see Figure 8.1). We assume MPCs 1 and 2 can perform their respective MPC optimizations faster than MPC 3. During an iterate, MPCs 1 and

2 solve their respective MPC optimizations and transmit calculated input trajectories to the other subsystems. Under synchronous operation, MPCs 1 and 2 remain idle and commence a subsequent iterate only after they receive new input trajectories from MPC 3. Under asynchronous operation, MPCs 1 and 2 do not idle; they commence another iterate (termed an *inner iterate*) utilizing previously obtained input trajectories from MPC 3. Information exchange during the inner iterates occurs between MPCs 1 and 2 only. On receiving new input trajectories from MPC 3, the three MPCs synchronize to correct all assumed and calculated input trajectories. The corrected input trajectories are transmitted to all other MPCs. Further details on the synchronization procedure are provided in Section 8.2. Several synchronizations may be performed within a sampling interval; several inner iterations may be performed between any two synchronization iterates.

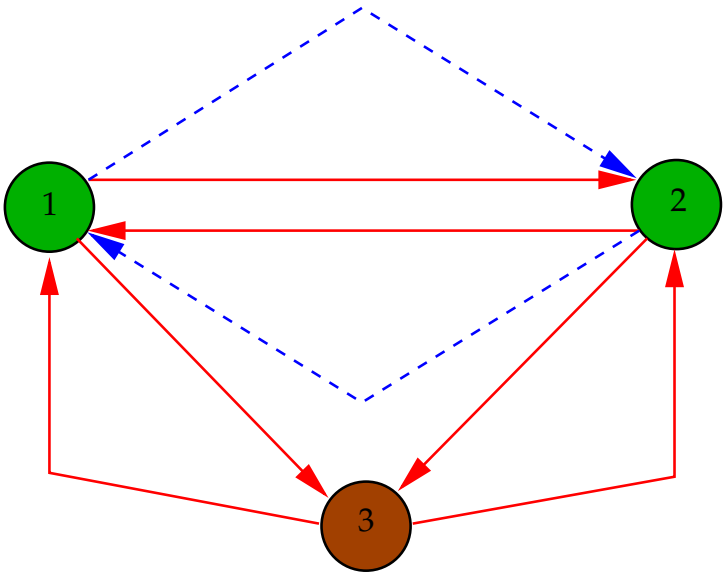

Figure 8.1: Asynchronous optimization for FC-MPC- a conceptual picture. MPCs 1 and 2 have shorter computational time requirements than MPC 3. Solid lines represent information exchange at synchronization. Dashed lines depict information exchange during inner iterations between MPCs 1 and 2.

8.1 Preliminaries

Lemma 8.1. *Let $\mathcal{X}$ be a convex, compact set and let $f(\cdot)$ be a continuous, strictly convex function on $\mathcal{X}$. Then, the solution x^* to the optimization problem*

$$\min_{x \in \mathcal{X}} f(x) \tag{8.1}$$

exists and is unique.

Proof. Since Equation (8.1) optimizes a continuous function $f(\cdot)$ over a compact set $\mathcal{X}$, a minimizer x^* exists. The second claim is well know and is stated in several textbooks without proof e.g., (Bertsekas, 1999, p. 193). A simple proof is presented here.

Assume there exists $\overline{x} \neq x^*$ such that $f(\overline{x}) = f(x^*)$. Since $f(\cdot)$ is strictly convex, we have for some $0 < \lambda < 1$ and $w = \lambda\overline{x} + (1-\lambda)x^*$ that

$$\begin{aligned} f(w) = f(\lambda\overline{x} + (1-\lambda)x^*) &< \lambda f(\overline{x}) + (1-\lambda) f(x^*) \\ &= f(x^*), \end{aligned}$$

which contradicts optimality of $f(x^*)$ and thereby establishes the lemma. □

Lemma 8.2. *Let $\mathbb{W} \subseteq \mathbb{R}^n$ be a nonempty, compact set. Consider an infinite sequence $w_k \in \mathbb{W}$. If w^* is the unique limit point of the sequence $\{w_k\}$, then $w_k \to w^*$.*

Proof. Suppose $w_k \nrightarrow w^*$, then $\exists$ an open ball $B^o_\varepsilon(w^*)$ such that $w_k \notin B^o_\varepsilon(w^*)$ infinitely often. It follows that $\mathbb{W} \backslash B^o_\varepsilon(w^*)$ is a closed, bounded set (hence compact) and therefore the infinite

subsequence $\{v_k = w_k \in \mathbb{W}\backslash B_\varepsilon^o(w^*)\}$ has a limit point $\overline{w}$ in $\mathbb{W}\backslash B_\varepsilon^o(w^*)$. By construction, $\overline{w} \neq w^*$, which is a contradiction. □

8.2 Asynchronous optimization for FC-MPC

Define $\mathbb{I}_+$ to be the set of positive integers. Let the collection of M subsystem-based MPCs be divided into $\mathcal{I} > 0$ groups. MPCs with similar computational time requirements are grouped together. We define the index set $\mathcal{J}_i \in \mathbb{I}_+^{s_i}$ to be the set of indices corresponding to subsystems in group i. Define the finite positive sequence $t_1 = s_1, t_2 = t_1 + s_2, \ldots, t_j = t_{j-1} + s_j, \ldots, t_\mathcal{I} = t_{\mathcal{I}-1} + s_\mathcal{I} = M$. WLOG we assume $\mathcal{J}_1 = \{1, 2, \ldots, t_1\},\ \mathcal{J}_2 = \{t_1 + 1, t_1 + 2, \ldots, t_2\}, \ldots, \mathcal{J}_\mathcal{I} = \{t_{\mathcal{I}-1} + 1, t_{\mathcal{I}-1} + 2, \ldots, t_\mathcal{I} = M\}$.

For asynchronous optimization, MPCs within a group perform a sequence of optimizations in parallel and exchange input trajectories with each other. For each group, a set of subsystems' MPC optimizations performed in parallel and the subsequent exchange of input trajectories between subsystems in the group is termed an *inner iterate*. During each inner iterate, MPCs in a group do not communicate with MPCs in other groups; information transfer is strictly within the group. The selected weight for each subsystem's MPC optimization is $w_i,\ i \in \mathbb{I}_M$ (see p. 38). The choice of subsystem weights satisfies $w_i > 0,\ \forall\, i \in \mathbb{I}_M$ and $\sum_{i=1}^M w_i = 1$. For each $\mathcal{J}_i, i = 1, 2, \ldots, \mathcal{I}$, q_i denotes the inner iteration number. A synchronization weight γ_i is selected for each group $\mathcal{J}_i, i = 1, 2, \ldots, \mathcal{I}$ such that $\gamma_i > 0,\ \forall\, i = 1, 2, \ldots, \mathcal{I}$ and $\sum_{i=1}^\mathcal{I} \gamma_i = 1$. Periodically, all MPCs exchange input trajectories; this is termed an *outer iterate* or a *synchronization iterate*. A local recalculation of all input trajectories is also performed during synchronization. In Algorithm 8.1, an explicit expression for this local recalculation is

provided. The notation p represents the synchronization (outer) iteration number.

For notational convenience, we define $\boldsymbol{z}_i$ to be the collection of subsystem input trajectories in group $i \in \mathcal{I}$ *i.e.*, $\boldsymbol{z}_i = [\boldsymbol{u}_{t_{i-1}+1}, \boldsymbol{u}_{t_{i-1}+2}, \ldots, \boldsymbol{u}_{t_i}]$. Likewise, $\overline{\boldsymbol{z}}_i = [\overline{\boldsymbol{u}}_{t_{i-1}+1}, \ldots, \overline{\boldsymbol{u}}_{t_i}]$. With slight abuse of notation, we define

$$\Pi_i^{q_i}(\boldsymbol{\zeta}_j) = [\boldsymbol{u}_{t_{i-1}+1}^{q_i}, \ldots, \boldsymbol{u}_{(j-1)}^{q_i}, \boldsymbol{\zeta}_j, \boldsymbol{u}_{(j+1)}^{q_i}, \ldots, \boldsymbol{u}_{t_i}^{q_i}] \tag{8.2}$$

for each subsystem $j \in \mathcal{J}_i$. Note in the definition of $\Pi_i^{q_i}(\boldsymbol{\zeta}_j)$ that the input trajectories corresponding to each subsystem $s \in \mathcal{J}_i, s \neq j$ are held constant at $\boldsymbol{u}_s^{q_i}$. The input trajectory for subsystem $j \in \mathcal{J}_i$ is $\boldsymbol{\zeta}_j$.

8.2.1 Asynchronous computation of open-loop policies

The asynchronous FC-MPC optimization problem for subsystem $j \in \mathcal{J}_i$, $\mathcal{F}_j^a$, is written as

$$\boldsymbol{\zeta}_j^{q_i}(k) \in \arg\min_{\boldsymbol{\zeta}_j} \quad \sum_{r=1}^{M} w_r \Phi_r \left(\boldsymbol{z}_1^{p-1}, \ldots, \boldsymbol{z}_{i-1}^{p-1}, \Pi_i^{q_i-1}(\boldsymbol{\zeta}_j), \boldsymbol{z}_{i+1}^{p-1}, \ldots, \boldsymbol{z}_{\mathcal{I}}^{p-1}; \mu(k) \right) \tag{8.3a}$$

subject to

$$u_i(t|k) \leq \Omega_i, \quad k \leq t \leq k+N-1 \tag{8.3b}$$

$$u_i(t|k) = 0, \quad k+N \leq t \tag{8.3c}$$

in which $\Phi_i(\cdot)$ is obtained by eliminating the subsystem states from the cost function $\phi_i(\cdot)$ (see Section 4.3, p. 39). In the asynchronous FC-MPC optimization problem for subsystem $j \in \mathcal{J}_i$, the input trajectories corresponding to each subsystem $l \notin \mathcal{J}_i$ are held constant at $\boldsymbol{u}_l^{p-1}$.

Let $\mathcal{U}_j = \Omega_j \times \ldots \times \Omega_j \in \mathbb{R}^{m_j N}, j \in \mathbb{I}_M$. For $\phi_j(\cdot)$ defined in Equation (4.5), p. 34, and $\Phi_j(\cdot)$ obtained by eliminating the CM states x_j from Equation (4.5) using the subsystem CM (Equation (4.1), p. 27), the FC-MPC optimization problem for subsystem $j \in \mathcal{J}_i$, $\mathcal{F}_j^a$, is

$$\overline{\boldsymbol{\zeta}}_j^{\,q_i} \in \arg\min_{\overline{\boldsymbol{\zeta}}_j} \frac{1}{2}\overline{\boldsymbol{u}}_j{}'\mathfrak{R}_j\overline{\boldsymbol{u}}_j + \left(r_j(k) + \sum_{s=1,s\neq j}^{M} \mathcal{H}_{js}\overline{\boldsymbol{v}}_s\right)' \overline{\boldsymbol{u}}_j + \text{constant} \tag{8.4a}$$

subject to

$$\overline{\boldsymbol{u}}_j \in \mathcal{U}_j \tag{8.4b}$$

in which

$$\overline{\boldsymbol{v}}_s = \begin{cases} \overline{\boldsymbol{u}}_s^{q_i-1} & \text{if } s \in \mathcal{J}_i,\ s \neq j, \\ \overline{\boldsymbol{u}}_s^{p-1} & \text{if } s \notin \mathcal{J}_i. \end{cases}$$

$$\mathfrak{R}_j = w_j\mathbb{R}_j + w_j E_{jj}{}'\mathbb{Q}_j E_{jj} + \sum_{l\neq j}^{M} w_l E_{lj}{}'\mathbb{Q}_l E_{lj}$$

$$\mathcal{H}_{js} = \sum_{l=1}^{M} w_l E_{lj}{}'\mathbb{Q}_l E_{ls}$$

$$r_j(k) = w_j E_{jj}{}'\mathbb{Q}_j f_j x_j(k) + \sum_{l\neq j}^{M} w_l E_{lj}{}'\mathbb{Q}_l f_l x_l(k)$$

The definitions of $E_{js}, f_j, \mathbb{Q}_j$ and $\mathbb{R}_j$, $\forall\, j, s \in \mathbb{I}_M$ are available in Section 4.5, p. 42. The terminal penalty for systems with stable decentralized modes is determined using Theorem 4.1, p. 49. If unstable decentralized modes are present, an additional terminal state constraint ${U_{u_i}}' x_i(k+N|k) = 0$ is required (in Equation (8.4)) to ensure closed-loop stability (see Section 4.6.2, p. 50). Closed-loop stability follows using Theorem 4.2, p. 53. The following algorithm is employed

for asynchronous optimization FC-MPC.

Algorithm 8.1 **(aFC-MPC)**. Given $\overline{\boldsymbol{u}}_i^0, \mathbb{Q}_i \geq 0, \mathbb{R}_i > 0, i \in \mathbb{I}_M$

$\mu(k), p_{\max}(k) \geq 0, p \leftarrow 1, \epsilon > 0$

$\mathcal{J}_j, q_j^{\max}(k) \geq 0, q_j \leftarrow 1,\ j = 1, 2, \ldots, \mathcal{I}$

$\kappa_j, \rho_s \leftarrow \Gamma\epsilon, j = 1, 2, \ldots, \mathcal{I}, s \in \mathbb{I}_M, \Gamma \gg 1$

$\overline{\boldsymbol{w}}_j^0 = \overline{\boldsymbol{u}}_j^0,\ \forall\, j \in \mathbb{I}_M$

while $\rho_s > \epsilon$ for some $s \in \mathbb{I}_M$ and $p \leq p_{\max}(k)$

do $\forall\, i = 1, 2, \ldots, \mathcal{I}$

Inner iterations.

while $\kappa_j > \epsilon$ for some $j \in \mathcal{J}_i$ and $q_i \leq q_i^{\max}(k)$

do $\forall\, j \in \mathcal{J}_i$

(i1) $\overline{\boldsymbol{\zeta}}_j^{\,q_i} \in \arg\ \mathcal{F}_i^a$ (Equation (8.4))

[1](i2) $\overline{\boldsymbol{w}}_j^{\,q_i} \leftarrow w_i \overline{\boldsymbol{\zeta}}_j^{\,q_i} + (1 - w_i)\, \overline{\boldsymbol{w}}_j^{q_i - 1}$

(i3) $\kappa_j \leftarrow \|\overline{\boldsymbol{w}}_j^{q_i} - \overline{\boldsymbol{w}}_j^{q_i - 1}\|$

(i4) Transmit $\overline{\boldsymbol{w}}_j^{q_i}$ to each subsystem $l \in \mathcal{J}_i,\ l \neq j$

end (do)

$q_i \leftarrow q_i + 1$

end (while)

end (do)

Synchronization (outer) iterations.

do $\forall\, i = 1, 2, \ldots, \mathcal{I}$

do $\forall\, j \in \mathcal{J}_i$

[1]In general, any strict convex combination (possibly different from w_i) may be used.

(o1) $\overline{\boldsymbol{u}}_j^p \leftarrow \gamma_i \overline{\boldsymbol{w}}_j^{q_i} + (1-\gamma_i)\,\overline{\boldsymbol{u}}_j^{p-1}$

(o2) $\rho_j \leftarrow \|\overline{\boldsymbol{u}}_j^p - \overline{\boldsymbol{u}}_j^{p-1}\|$

(o3) Transmit $\overline{\boldsymbol{u}}_j^p$ to each interconnected subsystem $l \in \mathbb{I}_M,\ l \neq j$

end (do)

end (do)

$p \leftarrow p+1$

$q_j \leftarrow 1,\ \overline{\boldsymbol{w}}_j^0 \leftarrow \overline{\boldsymbol{u}}_j^p,\ \forall\, j \in \mathbb{I}_M$

end (while)

The state trajectory for subsystem $j \in \mathcal{J}_i$ at inner iteration number q_i is $\overline{\boldsymbol{x}}_j^{q_i} \leftarrow \overline{\boldsymbol{x}}_j^{q_i}(\overline{\boldsymbol{z}}_1^{p-1}, \ldots, \overline{\boldsymbol{z}}_j^{q_i}, \ldots, \overline{\boldsymbol{z}}_{\mathcal{I}}^{p-1}; \mu(k))$. At each outer iterate p, the state trajectory for subsystem $l \in \mathbb{I}_M$ is obtained as $\overline{\boldsymbol{x}}_l^p \leftarrow \overline{\boldsymbol{x}}_l^p\left(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p; \mu(k)\right)$. By definition, $\boldsymbol{w}_i^{q_i} = [\overline{\boldsymbol{w}}_i^{q_i\prime}, 0, 0, \ldots]',\ i \in \mathbb{I}_M,\ q_i \in \mathbb{I}_+$.

8.2.2 Geometry of asynchronous FC-MPC

An example consisting of three subsystems is considered. MPCs 1 and 2 (for subsystems 1, 2) are assigned to $\mathcal{J}_1$ while MPC 3 (subsystem 3) is assigned to $\mathcal{J}_2$. We choose $q_1 = 3$ and $q_2 = 1$. The cost function for the three subsystems are Φ_1, Φ_2 and Φ_3 respectively. The decision variables for the three subsystems are u_1, u_2 and u_3. For the purpose of illustration, we project all relevant points on the $u_1 - u_2$ plane. Initially, the decision variables are assumed to have values $(u_1^0, u_2^0, u_3^0) = (2, 2, 0)$ (see Figure 8.2). MPCs 1 and 2 perform $q_1 = 3$ inner iterations (see steps (i1)-(i4) in Algorithm 8.1) assuming u_3 remains at its initial value $u_3^0 = 0$. The progress of the inner iterations is shown in Figure 8.2. During this time, MPC 3 performs an inner iteration ($q_2 = 1$) assuming $(u_1, u_2) = (u_1^0, u_2^0) = (2, 2)$.

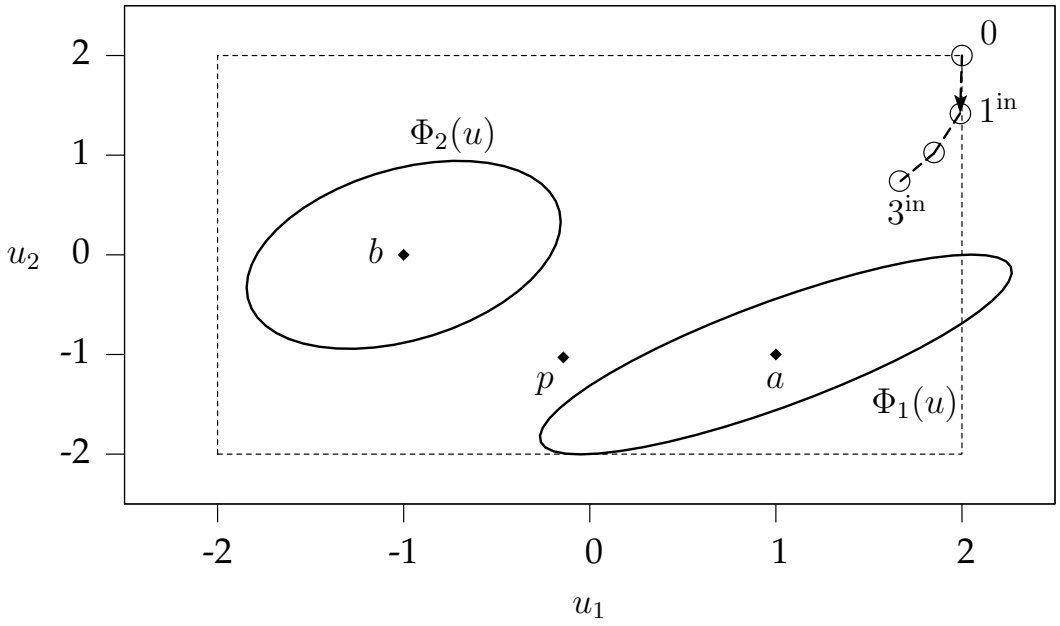

Figure 8.2: Progress of inner iterations performed by MPCs 1 and 2. Decision variable u_3 assumed to be at u_3^0. Point 3^{in} is obtained after three inner iterations for $\mathcal{J}_1$. p represents the Pareto optimal solution.

The first outer iteration is performed (steps (o1)-(o3) in Algorithm 8.1) next. In Figure 8.3, point 1 represents the result of the first synchronization iterate. The sequence of synchronization iterates is shown in Figure 8.4. Convergence to p, the Pareto optimal solution, is achieved after 4 synchronization iterates.

8.2.3 Properties

Lemma 8.3. *Consider any $\mathcal{J}_i, i = 1, 2, \ldots, \mathcal{I}$. Let $\widetilde{\boldsymbol{z}}_i^{q_i} = [\boldsymbol{w}_{t_{i-1}+1}^{q_i}, \ldots, \boldsymbol{w}_{t_i}^{q_i}]$. The sequence of cost functions*

$$\{\Phi(\boldsymbol{z}_1^{p-1}, \ldots, \boldsymbol{z}_{(i-1)}^{p-1}, \widetilde{\boldsymbol{z}}_i^{q_i}, \boldsymbol{z}_{(i+1)}^{p-1}, \ldots, \boldsymbol{z}_{\mathcal{I}}^{p-1}; \mu(k))\}$$

generated by Algorithm 8.1 is a nonincreasing function of the inner iteration number q_i.

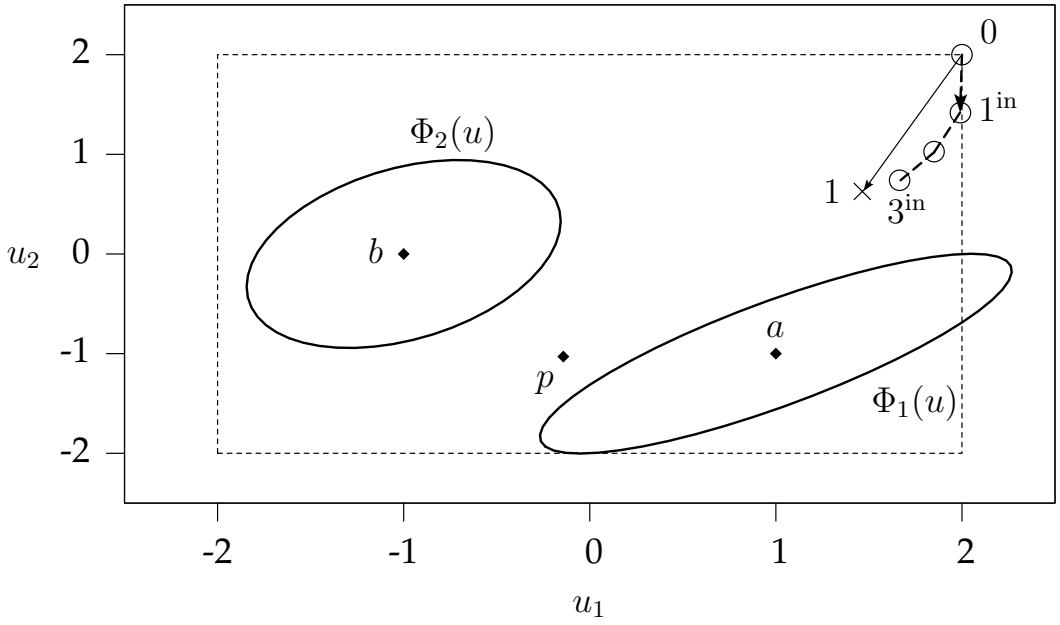

Figure 8.3: The first synchronization (outer) iterate. Point 1 represents the value of the decision variables after the first synchronization iterate.

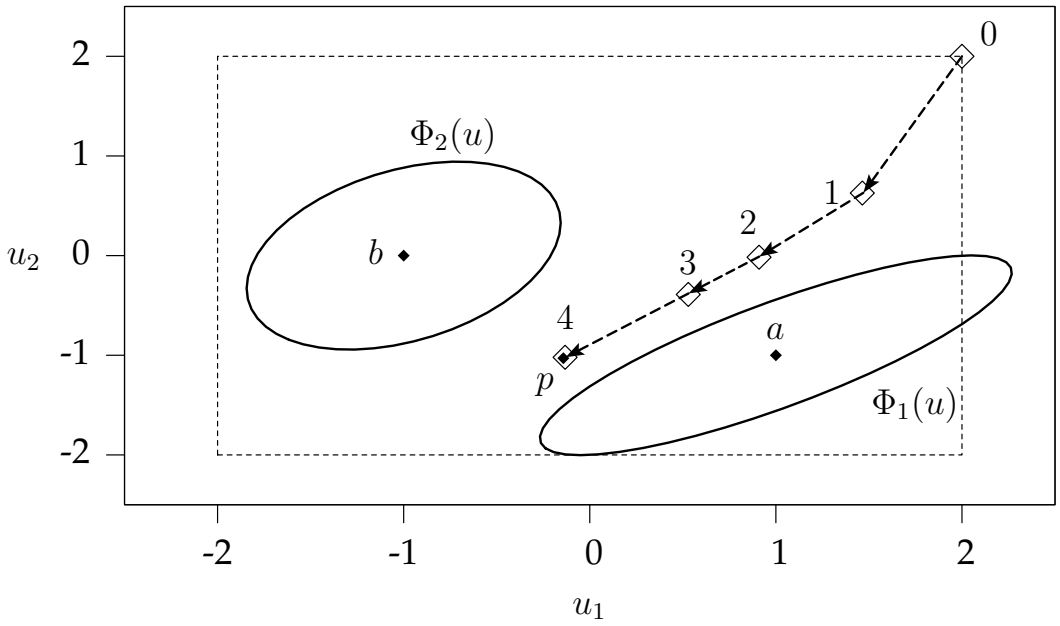

Figure 8.4: The sequence of synchronization (outer) iterations. Convergence to p is achieved after 4 synchronization iterates.

Proof. For any $l \in \mathcal{J}_i$, we have from Algorithm 8.1 that $\boldsymbol{u}_l^{q_i} = w_l \boldsymbol{\zeta}_i^{q_i} + (1 - w_l)\boldsymbol{w}_l^{q_i - 1}$. Hence,

$$
\begin{aligned}
&\Phi(\boldsymbol{z}_1^{p-1}, \ldots, \boldsymbol{z}_{(i-1)}^{p-1}, \widetilde{\boldsymbol{z}}_i^{q_i}, \boldsymbol{z}_{(i+1)}^{p-1}, \ldots, \boldsymbol{z}_{\mathcal{I}}^{p-1}; \mu(k)) \\
&\quad = \Phi(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{t_{i-1}}^{p-1}, \boldsymbol{w}_{t_{i-1}+1}^{q_i}, \ldots, \boldsymbol{w}_{t_i}^{q_i}, \boldsymbol{u}_{t_i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)) \\
&\quad = \Phi\Big(w_1(\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}) + \ldots \\
&\qquad + w_{t_{i-1}+1}(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{\zeta}_{t_{i-1}+1}^{q_i}, \boldsymbol{w}_{t_{i-1}+2}^{q_i - 1}, \ldots, \boldsymbol{w}_{t_i}^{q_i - 1}, \boldsymbol{u}_{t_i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}) + \ldots \\
&\qquad \ldots + w_{t_i}(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{w}_{t_{i-1}+1}^{q_i - 1}, \ldots, \boldsymbol{\zeta}_{t_i}^{q_i}, \boldsymbol{u}_{t_i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}) + \ldots \\
&\qquad \ldots + w_M(\boldsymbol{u}_1^{p-1}, \boldsymbol{u}_2^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}); \mu(k) \Big)
\end{aligned}
\tag{8.5a}
$$

Using convexity of $\Phi(\cdot)$, gives

$$
\begin{aligned}
&\leq \sum_{l=1}^{t_{i-1}} w_l \Phi(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)) \\
&\quad + \sum_{l=t_{i-1}+1}^{t_i} w_l \Phi(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{t_{i-1}}^{p-1}, \boldsymbol{w}_{t_{i-1}+1}^{q_i - 1}, \ldots, \boldsymbol{\zeta}_l^{q_i}, \ldots, \boldsymbol{w}_{t_i}^{q_i - 1}, \boldsymbol{u}_{t_i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)) \\
&\quad + \sum_{l=t_i+1}^{M} w_l \Phi(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k))
\end{aligned}
\tag{8.5b}
$$

From (i1) in Algorithm 8.1, we have

$$
\leq \Phi(\boldsymbol{z}_1^{p-1}, \ldots, \boldsymbol{z}_{(i-1)}^{p-1}, \widetilde{\boldsymbol{z}}_i^{q_i - 1}, \boldsymbol{z}_{(i+1)}^{p-1}, \ldots, \boldsymbol{z}_{\mathcal{I}}^{p-1}; \mu(k)) \tag{8.5c}
$$

Proceeding backwards to $q_i = 0$, and since $\boldsymbol{w}_j^0 = \boldsymbol{u}_i^{p-1},\ i \in \mathcal{J}_i$, we have

$$\leq \Phi(\boldsymbol{z}_1^{p-1}, \ldots, \boldsymbol{z}_{(i-1)}^{p-1}, \boldsymbol{z}_i^{p-1}, \boldsymbol{z}_{(i+1)}^{p-1}, \ldots, \boldsymbol{z}_{\mathcal{I}}^{p-1}; \mu(k)) \tag{8.5d}$$

□

Lemma 8.4. *The sequence of cost functions* $\{\Phi(\boldsymbol{z}_1^p, \ldots, \boldsymbol{z}_i^p, \ldots, \boldsymbol{z}_{\mathcal{I}}^p; \mu(k))\}$ *generated by Algorithm 8.1 is a nonincreasing function of the synchronization (outer) iteration number* p.

Proof. We have,

$$\begin{aligned}
&\Phi(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; \mu(k)) \\
&\quad = \Phi\Big(\gamma_1 \boldsymbol{u}_1^{q_1} + (1-\gamma_1)\boldsymbol{u}_1^{p-1}, \ldots\ldots, \gamma_2 \boldsymbol{u}_{t_1+1}^{q_2} + (1-\gamma_2)\boldsymbol{u}_{t_1+1}^{p-1}, && \text{(8.6a)} \\
&\qquad \ldots, \gamma_{\mathcal{I}} \boldsymbol{u}_{t_{\mathcal{I}}+1}^{q_{\mathcal{I}}} + (1-\gamma_{\mathcal{I}})\boldsymbol{u}_{t_{\mathcal{I}}+1}^{p-1}, \ldots, \gamma_{\mathcal{I}} \boldsymbol{u}_M^{q_{\mathcal{I}}} + (1-\gamma_{\mathcal{I}})\boldsymbol{u}_M^{p-1}; \mu(k)\Big) \\
&\quad = \Phi\Big(\gamma_1(\boldsymbol{u}_1^{q_1}, \ldots, \boldsymbol{u}_{t_1}^{q_1}, \boldsymbol{u}_{t_1+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}) \\
&\qquad + \gamma_2(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{t_1}^{p-1}, \boldsymbol{u}_{t_1+1}^{q_2}, \ldots, \boldsymbol{u}_{t_2}^{q_2}, \boldsymbol{u}_{t_2+1}^{p-1} \ldots, \boldsymbol{u}_M^{p-1}) + \ldots && \text{(8.6b)} \\
&\qquad \ldots + \gamma_{\mathcal{I}}(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{t_{\mathcal{I}}-1}^{p-1}, \boldsymbol{u}_{t_{\mathcal{I}-1}+1}^{q_{\mathcal{I}}} \ldots, \boldsymbol{u}_M^{q_{\mathcal{I}}}); \mu(k)\Big) \\
&\quad \leq \gamma_1 \Phi(\boldsymbol{u}_1^{q_1}, \ldots, \boldsymbol{u}_{t_1}^{q_1}, \boldsymbol{u}_{t_1+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)) \\
&\qquad + \gamma_2 \Phi(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{t_1}^{p-1}, \boldsymbol{u}_{t_1+1}^{q_2}, \ldots, \boldsymbol{u}_{t_2}^{q_2}, \boldsymbol{u}_{t_2+1}^{p-1} \ldots, \boldsymbol{u}_M^{p-1}; \mu(k)) + \ldots && \text{(8.6c)} \\
&\qquad \ldots + \gamma_{\mathcal{I}} \Phi(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{t_{\mathcal{I}}-1}^{p-1}, \boldsymbol{u}_{t_{\mathcal{I}-1}+1}^{q_{\mathcal{I}}} \ldots, \boldsymbol{u}_M^{q_{\mathcal{I}}}; \mu(k)) \\
&\quad \leq \sum_{r=1}^{\mathcal{I}} \gamma_r \Phi(\boldsymbol{z}_1^{p-1}, \ldots, \boldsymbol{z}_{r-1}^{p-1}, \boldsymbol{z}_r^{q_r}, \boldsymbol{z}_{r+1}^{p-1}, \ldots, \boldsymbol{z}_{\mathcal{I}}^{p-1}; \mu(k)) \quad \text{(from convexity of } \Phi(\cdot)) && \text{(8.6d)} \\
&\quad \leq \Phi(\boldsymbol{z}_1^{p-1}, \ldots\ldots, \boldsymbol{z}_{\mathcal{I}}^{p-1}; \mu(k)) \quad \text{(using Equation (8.5d))} && \text{(8.6e)}
\end{aligned}$$

□

From the definition of $\phi_j(\cdot),\ j \in \mathbb{I}_M$, we have $\mathbb{R}_j > 0,\ \forall\ j \in \mathbb{I}_M$. Therefore, $\Phi_j(\cdot)$ in Equation (8.4) is strictly convex. Convexity of each $\Omega_j, j \in \mathbb{I}_M$ implies that the Cartesian product $\Omega = \Omega_1 \times \Omega_2 \times \ldots \Omega_M$ is also convex. From Lemma 8.1, the solution $(\boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_M^*)$ to the centralized optimization problem

$$\min_{(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M)} \ \Phi(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M; \mu(k)) \tag{8.7a}$$

$$u_i(t|k) \in \Omega_i,\ k \leq t \leq k+N-1 \tag{8.7b}$$

$$u_i(t|k) = 0,\ k+N \leq t \tag{8.7c}$$

$$\forall\ i \in \mathbb{I}_M$$

exists and is unique.

Lemma 8.5. *Consider $\Phi(\cdot)$ positive definite quadratic and $\Omega_i, \forall\, i \in \mathbb{I}_M$ is nonempty, convex and compact. Let the solution to Algorithm 8.1 after p synchronization iterates be $(\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p)$ with an associated cost function value $\Phi(\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_M^p; \mu(k))$, in which $\boldsymbol{u}_i^p = [\overline{\boldsymbol{u}}_i^{p\prime}, 0, 0, \ldots]'$. Denote the unique solution to Equation (8.7) by $(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*)$, in which $\boldsymbol{u}_i^* = [\overline{\boldsymbol{u}}_i^{*\prime}, 0, 0, \ldots]'$, and let $\Phi(\boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_M^*; \mu(k))$ represent the optimal cost function value. The solution obtained at convergence of Algorithm 8.1 satisfies*

$$\lim_{p\to\infty} \Phi(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; \mu(k)) = \Phi(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*; \mu(k)) \text{ and}$$

$$\lim_{p\to\infty} (\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p) = (\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*)$$

Proof. By the assumptions on $\Phi(\cdot)$, we have that the curvature norm $\|\nabla^2\Phi(\cdot)\|$ is bounded above (by K, say) on $\Omega_1 \times \Omega_2 \times \ldots \times \Omega_M$. Suppose for contradiction that there is some other point $(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty) \neq (\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*)$ that is an accumulation point of the sequence $\{(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p)\}$. That is, we have a subsequence $\mathcal{S}$ such that $\lim_{p\in\mathcal{S}}(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p) = (\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty)$. For the full sequence, we have from Lemma 8.4 that

$$\Phi(\boldsymbol{z}_1^p, \boldsymbol{z}_2^p, \ldots, \boldsymbol{z}_\mathcal{I}^p; \mu(k)) \leq \sum_{r=1}^{\mathcal{I}} \gamma_r \Phi(\boldsymbol{z}_1^{p-1}, \ldots, \boldsymbol{z}_{r-1}^{p-1}, \boldsymbol{z}_r^{q_r}, \boldsymbol{z}_{r+1}^{p-1}, \ldots, \boldsymbol{z}_\mathcal{I}^{p-1}; \mu(k))$$
$$\leq \Phi(\boldsymbol{z}_1^{p-1}, \boldsymbol{z}_2^{p-1}, \ldots, \boldsymbol{z}_\mathcal{I}^{p-1}); \mu(k)) \quad (8.8)$$

In the limit as $p \to \infty$ in Equation (8.8),

$$\Phi(\boldsymbol{z}_1^\infty, \boldsymbol{z}_2^\infty, \ldots, \boldsymbol{z}_\mathcal{I}^\infty; \mu(k)) \leq \sum_{r=1}^{\mathcal{I}} \gamma_r \Phi(\boldsymbol{z}_1^\infty, \ldots, \boldsymbol{z}_{r-1}^\infty, \boldsymbol{z}_r^{q_r}, \boldsymbol{z}_{r+1}^\infty, \ldots, \boldsymbol{z}_\mathcal{I}^\infty; \mu(k))$$
$$\leq \Phi(\boldsymbol{z}_1^\infty, \boldsymbol{z}_2^\infty, \ldots, \boldsymbol{z}_\mathcal{I}^\infty; \mu(k)) \quad (8.9)$$

From Equation (8.5) in Lemma 8.3 and taking limits as $p \to \infty$, we have for each $r = 1, 2, \ldots, \mathcal{I}$

$$\Phi(\boldsymbol{z}_1^\infty, \ldots, \boldsymbol{z}_{r-1}^\infty, \boldsymbol{z}_r^{q_r}, \boldsymbol{z}_{r+1}^\infty, \ldots, \boldsymbol{z}_\mathcal{I}^\infty; \mu(k)) \leq \Phi(\boldsymbol{z}_1^\infty, \boldsymbol{z}_2^\infty, \ldots, \boldsymbol{z}_\mathcal{I}^\infty; \mu(k)) \quad (8.10)$$

Using Equations (8.9) and (8.10), we have

$$\Phi(\boldsymbol{z}_1^{q_1}, \boldsymbol{z}_2^\infty, \ldots, \boldsymbol{z}_\mathcal{I}^\infty; \mu(k)) = \Phi(\boldsymbol{z}_1^\infty, \boldsymbol{z}_2^{q_2}, \ldots, \boldsymbol{z}_\mathcal{I}^\infty; \mu(k)) = \ldots\ldots = \Phi(\boldsymbol{z}_1^\infty, \boldsymbol{z}_2^\infty, \ldots, \boldsymbol{z}_\mathcal{I}^{q_\mathcal{I}}; \mu(k))$$
$$= \Phi(\boldsymbol{z}_1^\infty, \boldsymbol{z}_2^\infty, \ldots, \boldsymbol{z}_\mathcal{I}^\infty; \mu(k)) \quad (8.11)$$

In fact, from Equation (8.5) in Lemma 8.3, we can argue in the limit $p \to \infty$ that

$$\begin{aligned}
&\Phi(\boldsymbol{\zeta}_1^{q_1}, \boldsymbol{u}_2^{q_1-1}, \ldots, \boldsymbol{u}_{t_1}^{q_1-1}, \boldsymbol{u}_{t_1+1}^{\infty}, \ldots, \boldsymbol{u}_M^{\infty}; \mu(k)) \\
&\qquad = \Phi(\boldsymbol{u}_1^{q_1-1}, \boldsymbol{\zeta}_2^{q_1}, \ldots, \boldsymbol{u}_{t_1}^{q_1-1}, \boldsymbol{u}_{t_1+1}^{\infty}, \ldots, \boldsymbol{u}_M^{\infty}; \mu(k)) = \ldots\ldots \\
&= \ldots = \Phi(\boldsymbol{z}_1^{\infty}, \ldots, \boldsymbol{z}_{i-1}^{\infty}, \boldsymbol{u}_{t_{i-1}+1}^{q_i-1}, \ldots, \boldsymbol{\zeta}_j^{q_i}, \ldots, \boldsymbol{u}_{t_i}^{q_i-1}, \boldsymbol{z}_{i+1}^{\infty}, \ldots, \boldsymbol{z}_{\mathcal{I}}^{\infty}; \mu(k)) = \ldots \\
&\qquad = \ldots = \Phi(\boldsymbol{z}_1^{\infty}, \ldots, \boldsymbol{z}_{\mathcal{I}-1}^{\infty}, \boldsymbol{u}_{t_{\mathcal{I}-1}+1}^{q_{\mathcal{I}}}, \ldots, \boldsymbol{\zeta}^{q_{\mathcal{I}}}; \mu(k)) \\
&\qquad\qquad = \Phi(\boldsymbol{u}_1^{\infty}, \boldsymbol{u}_2^{\infty}, \ldots, \boldsymbol{u}_M^{\infty}; \mu(k)) \quad (8.12)
\end{aligned}$$

Because $(\boldsymbol{u}_1^{\infty}, \boldsymbol{u}_2^{\infty}, \ldots, \boldsymbol{u}_M^{\infty})$ is nonoptimal, and using first-order optimality conditions, at least one of the following conditions hold:

$$\nabla_{\boldsymbol{u}_j} \Phi(\boldsymbol{u}_1^{\infty}, \boldsymbol{u}_2^{\infty}, \ldots, \boldsymbol{u}_M^{\infty}; \mu(k))'(\boldsymbol{u}_j^* - \boldsymbol{u}_j^{\infty}) < 0, \ j \in \mathbb{I}_M \quad (8.13)$$

Suppose WLOG that Equation (8.13) is true for $j = s, \ s \in \mathbb{I}_M$. We thus have by Taylor's theorem that

$$\begin{aligned}
&\Phi(\boldsymbol{u}_1^{\infty}, \ldots, \boldsymbol{u}_{s-1}^{\infty}, \boldsymbol{u}_s^{\infty} + \alpha(\boldsymbol{u}_s^* - \boldsymbol{u}_s^{\infty}), \boldsymbol{u}_{s+1}^{\infty}, \ldots, \boldsymbol{u}_M^{\infty}; \mu(k)) \\
&\qquad = \Phi(\boldsymbol{u}_1^{\infty}, \boldsymbol{u}_2^{\infty}, \ldots, \boldsymbol{u}_M^{\infty}; \mu(k)) + \alpha \nabla_{\boldsymbol{u}_s} \Phi(\boldsymbol{u}_1^{\infty}, \boldsymbol{u}_2^{\infty}, \ldots, \boldsymbol{u}_M^{\infty})'(\boldsymbol{u}_s^* - \boldsymbol{u}_s^{\infty}) \\
&\qquad\quad + \frac{1}{2}\alpha^2 (\boldsymbol{u}_s^* - \boldsymbol{u}_s^{\infty})' \nabla_{\boldsymbol{u}_s}^2 \Phi(\boldsymbol{u}_1^{\infty}, \boldsymbol{u}_2^{\infty}, \ldots, \boldsymbol{u}_M^{\infty})'(\boldsymbol{u}_s^* - \boldsymbol{u}_s^{\infty}), \quad (8.14)
\end{aligned}$$

By choosing the minimizing α, we can identify a point $\widetilde{\boldsymbol{u}}_s$ with [2]

$$\begin{aligned}
&\Phi(\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_{s-1}^\infty, \widetilde{\boldsymbol{u}}_s, \boldsymbol{u}_{s+1}^\infty, \ldots, \boldsymbol{u}_M^\infty; \mu(k)) \\
&\quad = \Phi(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty; \mu(k)) - \frac{|\nabla_{\boldsymbol{u}_s} \Phi(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty; \mu(k))'(\boldsymbol{u}_s^* - \boldsymbol{u}_s^\infty)|^2}{2(\boldsymbol{u}_s^* - \boldsymbol{u}_s^\infty)' \nabla^2_{\boldsymbol{u}_1} \Phi(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty; \mu(k))(\boldsymbol{u}_s^* - \boldsymbol{u}_s^\infty)} \\
&\quad \leq \Phi(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty; \mu(k)) - \frac{1}{2K} \frac{|\nabla_{\boldsymbol{u}_1} \Phi(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty; \mu(k)'(\boldsymbol{u}_s^* - \boldsymbol{u}_s^\infty)|^2}{\|\boldsymbol{u}_s^* - \boldsymbol{u}_s^\infty\|^2} \\
&\quad = \Phi(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty; \mu(k)) - \epsilon, \quad (8.15)
\end{aligned}$$

for an obvious definition of $\epsilon > 0$. Hence, for any p we have

$$\begin{aligned}
&\Phi(\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_{s-1}^p, \boldsymbol{\zeta}_s^{p+1}, \boldsymbol{u}_{s+1}^p, \ldots, \boldsymbol{u}_M^p; \mu(k)) \\
&\qquad \leq \Phi(\boldsymbol{u}_1^p, \ldots, \boldsymbol{u}_{s-1}^p, \widetilde{\boldsymbol{u}}_s, \boldsymbol{u}_{s+1}^p, \ldots, \boldsymbol{u}_M^p; \mu(k)) \\
&\qquad = \Phi(\boldsymbol{u}_1^\infty, \ldots, \boldsymbol{u}_{s-1}^\infty, \widetilde{\boldsymbol{u}}_s, \boldsymbol{u}_{s+1}^\infty, \ldots, \boldsymbol{u}_M^\infty; \mu(k)) + O\left(\sum_{r=1, r\neq s}^{M} \|\boldsymbol{u}_r^\infty - \boldsymbol{u}_r^p\| \right) \\
&\qquad \leq \Phi(\boldsymbol{u}_1^\infty, \boldsymbol{u}_2^\infty, \ldots, \boldsymbol{u}_M^\infty; \mu(k)) + O\left(\sum_{r=1, r\neq s}^{M} \|\boldsymbol{u}_r^\infty - \boldsymbol{u}_r^p\| \right) - \epsilon. \quad (8.16)
\end{aligned}$$

By taking the limits of both sides over $p \in \mathcal{S}$, $p \to \infty$, we obtain $0 \leq -\epsilon$, a contradiction.

For the final statement of the proof, we note that the iterates are confined to the level set

$$S_0 = \{(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M) \,|\, \Phi(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M; \mu(k)) \leq \Phi(\boldsymbol{u}_1^0, \ldots, \boldsymbol{u}_M^0; \mu(k))\}$$

The iterates are, therefore, guaranteed to have an accumulation point. By the first part of the theorem all such accumulation points are optimal. Hence (from the statement of the lemma), the only limit point is $(\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*)$, and from Lemma 8.2 it is in fact the limit. □

[2] I would like to thank Professor Stephen J. Wright for showing me this construction.

8.2.4 Closed-loop properties

At time k, let Algorithm 8.1 be terminated after $p(k) = \sigma > 0$ synchronization iterates. Let

$$\boldsymbol{u}_i^{\sigma}(\mu(k)) = [u_i^{\sigma}(\mu(k), 0)', \ldots, u_i^{\sigma}(\mu(k), N-1)', 0, 0, \ldots]', \ i \in \mathbb{I}_M \tag{8.17}$$

represent the solution to Algorithm 8.1 after σ synchronization iterates. The input injected into subsystem $i \in \mathbb{I}_M$ under asynchronous optimization based FC-MPC is $u_i^{\sigma}(\mu(k), 0)$.

For open-loop stable systems, the initialization procedure described in Section 4.6.1 may be used to initialize Algorithm 8.1. Closed-loop stability under intermediate termination of Algorithm 8.1 at any synchronization (outer) iterate can be established under the conditions specified in Theorem 4.1. For systems with unstable decentralized modes, the initialization procedure described in Section 4.6.2 may be used to initialize Algorithm 8.1. Closed-loop stability follows under the conditions described in Theorem 4.2. The proof for closed-loop stability in either case is identical to the corresponding proof for closed-loop stability presented in Chapter 4 (proofs for Theorems 4.1 and 4.2).

8.2.5 Example: Two reactor chain with nonadiabatic flash

We revisit two reactor-flash example described in Section 4.7.2. We assume that MPCs 1 and 2 for the CSTRs have a smaller (worst case) computational time requirement than MPC 3. Two cases for asynchronous optimization based FC-MPC (aFC-MPC) are considered. In the first case, $q_1 = 2, q_2 = 1$ and in the second case $q_1 = 5, q_2 = 1$. A setpoint change is made to H_m. The performance of aFC-MPC is evaluated and compared against (synchronous) FC-MPC and

centralized MPC. The setpoint tracking performance using the different MPCs is shown in Figures 8.5 and 8.6. The closed-loop control costs are given in Table 8.1.

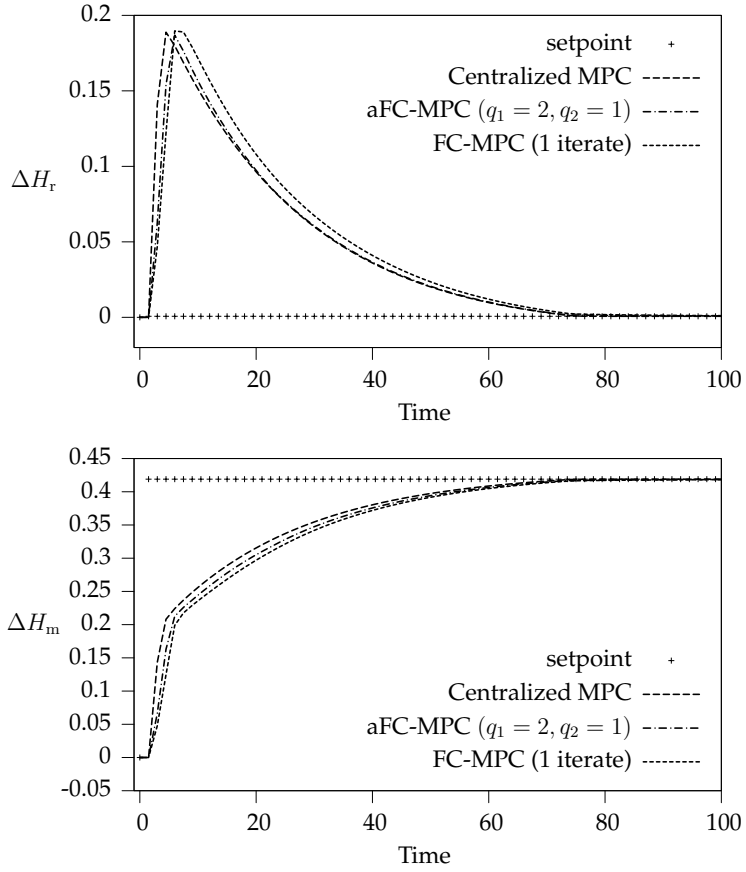

Figure 8.5: Setpoint tracking for levels in the two CSTRs.

From the closed-loop control costs, aFC-MPC $(q_1 = 2, q_2 = 1)$ gives a 40% improvement in performance compared to FC-MPC (1 iterate). For aFC-MPC with $q_1 = 5$ and $q_2 = 1$, a 66% improvement in performance is obtained over synchronous FC-MPC terminated after 1 iterate.

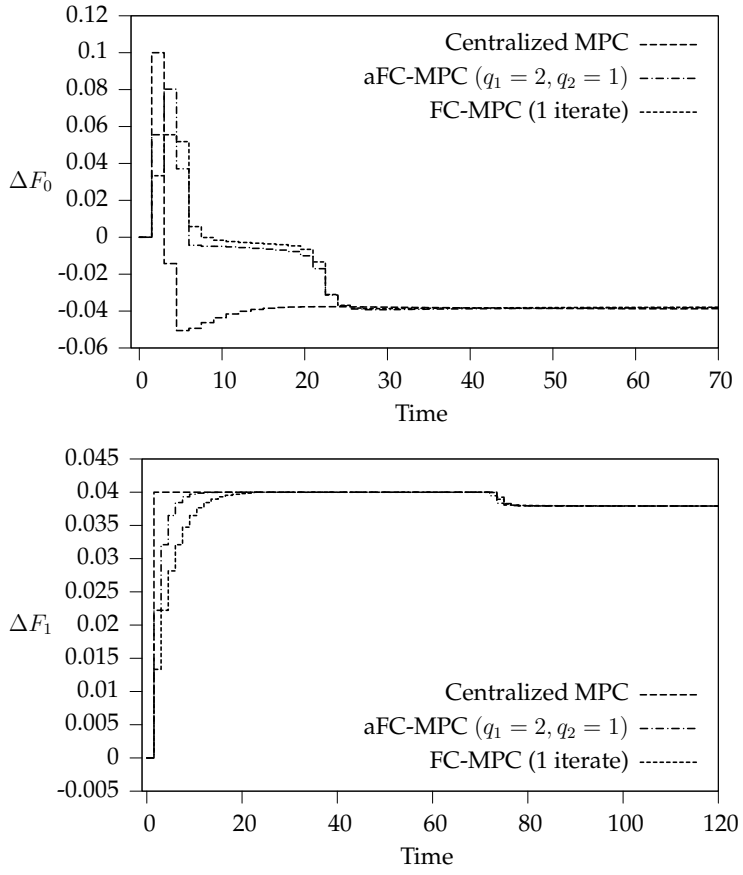

Figure 8.6: Manipulated feed flowrates for setpoint tracking of levels.

Table 8.1: Setpoint tracking performance of centralized MPC, FC-MPC and asynchronous FC-MPC.

	$\Lambda_{\text{cost}} \times 10^{-2}$	$\Delta\Lambda_{\text{cost}}\%$
Centralized MPC	2.41	$--$
FC-MPC (1 iterate)	3.15	30.7
aFC-MPC ($q_1 = 2, q_2 = 1$)	2.85	18.2
aFC-MPC ($q_1 = 5, q_2 = 1$)	2.65	10.3

8.3 Conclusions

An algorithm for asynchronous optimization based distributed MPC was described in this chapter. Asynchronous optimization avoids the need for synchronized clock keeping at each iterate by allowing MPCs with disparate computational time requirements to function at their

respective rates. In this framework, MPCs with similar computational time requirements are grouped together. Consider MPC $i \in \mathbb{I}_M$ in group $\mathcal{J}_\alpha$. At each inner iterate q_i, MPC i calculates its input trajectory assuming the input trajectories of MPCs $j \in \mathcal{J}_\alpha, j \neq i$ are at $\boldsymbol{u}_j^{q_i-1}, j \in \mathcal{J}_\alpha$; the input trajectories corresponding to MPCs in other groups are assumed to be at values obtained at the last synchronization iterate $p-1$ *i.e.*, $\boldsymbol{u}_s^{p-1}, s \notin \mathcal{J}_\alpha$. MPC $i \in \mathcal{J}_\alpha$ transmits $\boldsymbol{u}_i^{q_i}$ to each MPC $j \in \mathcal{J}_\alpha, j \neq i$ and receives $\boldsymbol{u}_j^{q_i}$ from MPC $j \in \mathcal{J}_\alpha, j \neq i$. No information is exchanged with MPCs outside the group. Periodically, all MPCs $i \in \mathbb{I}_M$ exchange input trajectories and calculate $\boldsymbol{u}_i^p, i \in \mathbb{I}_M$ at synchronization iterate p. It was shown that the asynchronous FC-MPC algorithm can be terminated at any synchronization iterate without affecting feasibility or nominal closed-loop stability. At convergence of the asynchronous FC-MPC algorithm, the optimal centralized MPC performance is achieved. The synchronous FC-MPC framework described in earlier chapters can be derived as a special case of Algorithm 8.1. For synchronous FC-MPC, all MPCs are assumed to belong to group $\mathcal{J}_\alpha$. By setting $\gamma_\alpha = 1, \gamma_j = 0, \; \forall \, j \neq \alpha$, we revert to the synchronous FC-MPC algorithm (Algorithm 4.1, p. 43) described in Chapter 4.

Chapter 9

Distributed constrained LQR

In Chapter 4, a terminal penalty distributed MPC framework with nominal stability and performance properties was described. Since terminal penalty FC-MPC is reliant on a suboptimal parameterization of the postulated control trajectories, it cannot achieve infinite horizon optimal performance for finite N, even at convergence. In this chapter, through a simple reformulation of the FC-MPC optimization problem in Chapter 4, a distributed MPC formulation that achieves infinite horizon optimal performance at convergence is described. A terminal state constraint constraint distributed MPC framework is also described.

9.1 Notation and preliminaries

The matrices $(A_{\mathrm{cm}}, B_{\mathrm{cm}}, C_{\mathrm{cm}})$ represent the A, B, C matrices respectively of the CM for the entire plant (see Equation (4.2), p. 28). Let x_{cm} denote the state vector for the plant CM. It is assumed that $(A_{\mathrm{cm}}, B_{\mathrm{cm}})$ is stabilizable and $(A_{\mathrm{cm}}, C_{\mathrm{cm}})$ is detectable.

Let $\mathbb{I}$ denote the set of integers. The notation $\mathbb{I}_+$ is used to represent the set of strictly positive integers. The closed ball $B_\varepsilon(x)$ is defined as

$$B_\varepsilon(x) = \{\, z \mid \|z - x\| \leq \varepsilon \}$$

The open ball $B^o_\varepsilon(x)$ is defined as

$$B_\varepsilon(x) = \{\, z \mid \|z - x\| < \varepsilon \}$$

The notation μ represents the set of subsystem states $[x_1, x_2, \ldots, x_M]$. By definition, $x_{\text{cm}} = [x_1{}', x_2{}', \ldots, x_M{}']'$. Let $\mathbb{X} \subseteq \mathbb{R}^n$ be a nonempty set. Define the operation $\mu \in \mathbb{X}$ to represent $[x_1{}', x_2{}', \ldots, x_M{}']' \in \mathbb{X}$. The control horizon is represented by N.

Lemma 9.1 (Minimum principle for constrained, convex optimization)**.** *Let $\mathcal{X}$ be a convex set and let f be a convex function over $\mathcal{X}$. A necessary and sufficient condition for x^* to be a global minimum of f over $\mathcal{X}$ is*

$$\nabla f(x^*)'(x - x^*) \geq 0, \; \forall \, x \in \mathcal{X}$$

A proof is given in (Bertsekas, 1999, p. 194)

Lemma 9.2. *Let*

$$\mathbb{A} = \begin{pmatrix} \mathcal{A} & & \\ & \ddots & \\ & & \mathcal{A}_s \end{pmatrix} \in \mathbb{R}^{(n+n_s)\times(n+n_s)} \qquad \mathbb{B} = \begin{pmatrix} \mathcal{B} \\ \vdots \\ \mathcal{B}_s \end{pmatrix} \in \mathbb{R}^{(n+n_s)\times m}$$

in which, A_s is stable, $\mathcal{A} \in \mathbb{R}^{n\times n}$ and $\mathcal{B} \in \mathbb{R}^{n\times m}$. The pair $(\mathbb{A}, \mathbb{B})$ is stabilizable if and only if $(\mathcal{A}, \mathcal{B})$ is stabilizable.

A proof is given in Appendix 9.6.1.

Assumption 9.1. All interaction models are stable *i.e.,* for each $i, j \in \mathbb{I}_M$, $|\lambda_{\max}(A_{ij})| < 1, \forall j \neq i$.

9.2 Infinite horizon distributed MPC

Cost function. The *stage cost* at stage $t \geq k$ along the prediction horizon and the *cost function* $\phi_i(\cdot)$ for subsystem i are given by Equations (4.3) and (4.5) respectively in Chapter 4.

9.2.1 The benchmark controller : centralized constrained LQR

For any system, centralized constrained LQR (CLQR) achieves infinite horizon optimal performance. The CLQR optimization problem is

$\mathcal{P}^0$: **Centralized constrained LQR**

$$\min_{\boldsymbol{x},\boldsymbol{u}} \ \phi\left(\boldsymbol{x}, \boldsymbol{u}; \mu(k)\right) = \ \sum_i w_i \phi_i\left(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i(k)\right) \tag{9.1a}$$

$$\text{subject to} \quad x(t+1) = Ax(t) + Bu(t), \tag{9.1b}$$

$$u_i(l|k) \in \Omega_i, \quad k \leq l, \tag{9.1c}$$

$$\forall\, i \in \mathbb{I}_M$$

$$\text{where } w_i > 0,\ i \in \mathbb{I}_M \text{ and } \sum_{i=1}^{M} w_i = 1$$

The problem with CLQR is the infinite number of decision variables and constraints in the optimization problem (Equation (9.1)). To avoid dealing with an intractable, infinite dimensional

optimization problem, we typically choose a suitably long control horizon N such that the system state at the end of the control horizon is inside a positively invariant set where the optimal unconstrained (LQR) control law is feasible and therefore, optimal. A suitable terminal penalty is calculated using this optimal unconstrained feedback law and the terminal set constraint remains implicit through the choice of N. The infinite dimensional optimization problem can, therefore, be replaced by an equivalent finite dimensional optimization problem. Implementable algorithms for CLQR have been described in Chmielewski and Manousiouthakis (1996); Scokaert and Rawlings (1998); Sznaier and Damborg (1990).

9.2.2 Distributed constrained LQR (DCLQR) .

We consider, in the spirit of the work of Chmielewski and Manousiouthakis (1996); Scokaert and Rawlings (1998); Sznaier and Damborg (1990), distributed constrained LQR, which achieves infinite horizon optimal performance at convergence. The central issue remains the same; to deal with infinite number of decision variables and constraints, we require a parameterization that allows us to replace the intractable, infinite dimensional distributed constrained LQR optimization problem with an equivalent finite dimensional one.

Several choices exist for parameterizing the input in a neighborhood of the origin. In special cases where the different subsystems interact weakly, one may be able to use decentralized feedback laws to stabilize the system in a neighborhood of the origin. Stabilization may be achieved either by employing a dual mode controller philosophy (see Michalska and Mayne (1993)), in which the decentralized feedback controllers take charge once the MPCs drive the system to a desired neighborhood of the origin (Dunbar, 2006) or by using the decentralized feedback control law to calculate an appropriate stabilizing terminal penalty, if it

exists. For many systems, decentralized feedback may not be sufficient to stabilize the system, even in a neighborhood of the origin. In fact, a set of stabilizing decentralized feedback gains may not even exist. Furthermore, the performance with a set of decentralized feedback laws is always suboptimal when the subsystems are interacting. In Chapter 4, the input trajectories were parameterized using $u_i(k+j|k) = 0,\ N \leq j, i \in \mathbb{I}_M$. This terminal penalty distributed MPC formulation is stabilizing for any number of iterates and converges to the solution of a modified infinite horizon centralized control problem. Terminal penalty FC-MPC (Chapter 4), however, achieves infinite horizon optimal performance only in the limit $N \to \infty$.

An alternative terminal feedback law that allows us to achieve infinite horizon optimal performance at convergence is the unconstrained centralized feedback law. The idea here is to force the collection of subsystem-based MPCs to drive the system state to a neighborhood of the origin in which the unconstrained, optimal plant CM feedback law is feasible. From Gilbert and Tan (1991), we know that such a neighborhood of the origin is well defined and can be computed offline.

Following the description in Gilbert and Tan (1991), we use $\mathcal{O}_\infty(A_{\mathrm{cm}}, C_{\mathrm{cm}})$ to denote the maximal output admissible set for the plant CM $(A_{\mathrm{cm}}, B_{\mathrm{cm}}, C_{\mathrm{cm}})$. Convexity of each $\Omega_i, \forall\, i \in \mathbb{I}_M$ implies that $\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_M$ is convex. Hence, $\mathcal{O}_\infty(\cdot)$ is convex (Gilbert and Tan, 1991, Theorem 2.1). Let $\Omega_i,\ i \in \mathbb{I}_M$ be a polytope given by

$$\Omega_i \triangleq \left\{ u_i \,\middle|\, D_i u_i \leq d_i, d_i > 0 \right\}$$

Determination of $\mathcal{O}_\infty(\cdot)$, in this case, involves the solution to a set of linear programs.

Consider the plant CM $(\widehat{A}, \widehat{B}, \widehat{C}) = (T^{-1}A_{\rm cm}T, T^{-1}B_{\rm cm}, C_{\rm cm}T)$ in observability canonical form, in which

$$\widehat{A} = \begin{bmatrix} A^{\rm o} & 0 \\ A^{12} & A^{\overline{\rm o}} \end{bmatrix}, \ \widehat{C} = \begin{bmatrix} C^{\rm o} & 0 \end{bmatrix} \tag{9.2}$$

and $A^{\rm o} \in \mathbb{R}^{n_o \times n_o}$, $A_{\overline{\rm o}} \in \mathbb{R}^{n-n_o \times n-n_o}$ denote the observable and unobservable modes, respectively. Since $(A_{\rm cm}, C_{\rm cm})$ is detectable, and using Lemma 5.1, $A^{\overline{\rm o}}$ is stable. The maximal admissible set $\mathcal{O}_\infty(\widehat{A}, \widehat{C}) = \mathcal{O}_\infty(A^{\rm o}, C^{\rm o}) \times \mathbb{R}^{n-n_{\rm o}}$. Also, $\mathcal{O}_\infty(A_{\rm cm}, C_{\rm cm}) = T\mathcal{O}_\infty(\widehat{A}, \widehat{C})$ (Gilbert and Tan, 1991). Because $(A_{\rm cm}, C_{\rm cm})$ is detectable only (and not observable), $\mathcal{O}_\infty(A_{\rm cm}, C_{\rm cm})$ is a cylinder with infinite extent along directions in the unobservable subspace.

Let $K_{\rm cm}$ denote the optimal, linear quadratic regulator (LQR) gain and let $\Pi_{\rm cm}$ denote the solution to the corresponding discrete steady-state Riccati equation for the plant CM *i.e.*,

$$\Pi_{\rm cm} = \mathcal{Q} + A_{\rm cm}{}'\Pi_{\rm cm}A_{\rm cm} - A_{\rm cm}{}'\Pi_{\rm cm}B_{\rm cm}(\mathcal{R} + B_{\rm cm}{}'\Pi_{\rm cm}B_{\rm cm})^{-1}B_{\rm cm}{}'\Pi_{\rm cm}A_{\rm cm} \tag{9.3a}$$

$$K_{\rm cm} = -(\mathcal{R} + B_{\rm cm}{}'\Pi_{\rm cm}B_{\rm cm})^{-1}B_{\rm cm}{}'\Pi_{\rm cm}A_{\rm cm} \tag{9.3b}$$

in which $\mathcal{Q} = \mathrm{diag}(w_1Q_1, w_2Q_2, \ldots, w_MQ_M)$ and $\mathcal{R} = \mathrm{diag}(w_1R_1, w_2R_2, \ldots, w_MR_M)$. Conditions for existence of a solution to Equation (9.3) are well known (Bitmead and Gevers, 1991; Bitmead, Gevers, Petersen, and Kaye, 1985; Chan, Goodwin, and Sin, 1984; de Souza, Gevers,

and Goodwin, 1986). Both K_{cm} and Π_{cm} are partitioned subsystem-wise and written as

$$K_{\mathrm{cm}} = \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1M} \\ K_{21} & K_{22} & \dots & K_{2M} \\ \vdots & \ddots & \ddots & \vdots \\ K_{M1} & K_{M2} & \dots & K_{MM} \end{bmatrix} \qquad \Pi_{\mathrm{cm}} = \begin{bmatrix} \Pi_{11} & \Pi_{12} & \dots & \Pi_{1M} \\ \Pi_{21} & \Pi_{22} & \dots & \Pi_{2M} \\ \vdots & \ddots & \ddots & \vdots \\ \Pi_{M1} & \Pi_{M2} & \dots & \Pi_{MM} \end{bmatrix} \tag{9.4}$$

Lyapunov stability of $(A_{\mathrm{cm}} + B_{\mathrm{cm}}K_{\mathrm{cm}})$ and $0 \in \mathrm{int}(\Omega)$ implies $0 \in \mathrm{int}(\mathcal{O}_\infty)$ (Gilbert and Tan, 1991, Theorem 2.1).

Optimization. Define for each subsystem $i \in \mathbb{I}_M$

$$\Phi\left(\boldsymbol{u}_1^{p-1}, \dots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^{p-1}, \dots, \boldsymbol{u}_M^{p-1}; \mu(k)\right) = \sum_{r=1}^{M} w_r \Phi_r\left(\boldsymbol{u}_1^{p-1}, \dots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^{p-1}, \dots, \boldsymbol{u}_M^{p-1}; x_r(k)\right)$$

The DCLQR optimization problem for subsystem i is

$$\min_{\boldsymbol{u}_i} \quad \Phi\left(\boldsymbol{u}_1^{p-1}, \dots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^{p-1}, \dots, \boldsymbol{u}_M^{p-1}; \mu(k)\right) \tag{9.5a}$$

subject to

$$u_i(t|k) \in \Omega_i, \;\; k \le t \le k+N-1 \tag{9.5b}$$

$$u_i(t|k) = K_{ii}x_i(t|k) + \sum_{j \neq i} K_{ij}x_j(t|k), \;\; k+N \le t \tag{9.5c}$$

To indicate explicitly, the N dependence of the finite input trajectory for each subsystem, we write

$$\overline{\boldsymbol{u}}_i(k;N) = \left[u_i(k|k)', u_i(k+1|k)', \ldots, u_i(k+N-1|k)'\right]', \ \forall\, i \in \mathbb{I}_M \tag{9.6}$$

The finite state trajectory for subsystem i, generated by the collection of input trajectories $\overline{\boldsymbol{u}}_1(k;N), \ldots, \overline{\boldsymbol{u}}_M(k;N)$ is denoted by $\overline{\boldsymbol{x}}_i(\overline{\boldsymbol{u}}_1(k;N), \ldots, \overline{\boldsymbol{u}}_M(k;N); \mu(k))$. For notational simplicity, we write $\overline{\boldsymbol{x}}_i(k;N) \leftarrow \overline{\boldsymbol{x}}_i(\overline{\boldsymbol{u}}_1(k;N), \overline{\boldsymbol{u}}_2(k;N) \ldots, \overline{\boldsymbol{u}}_M(k;N); \mu(k))$. Define

$$\mathbb{Q}_i = \text{diag}\left(w_i Q_i(1), \ldots, w_i Q_i(N-1), \Pi_{ii}\right)$$

$$\mathbb{R}_i = \text{diag}\left(w_i R_i(0), w_i R_i(1), \ldots, w_i R_i(N-1)\right)$$

$$\mathbb{T}_{ij} = \text{diag}\left(0, \ldots, 0, \Pi_{ij}\right)$$

Assumption 9.2. For each $i \in \mathbb{I}_M$, $Q_i(1) = \ldots = Q_i(N-1) = Q_i \geq 0$, $R_i(0) = \ldots = R_i(N-1) = R_i > 0$, $(A_i, Q_i^{1/2})$ detectable.

The symbol $\Psi(\cdot)$ represents the cost function expressed in terms of the finite horizon input trajectories $\overline{\boldsymbol{u}}_1(k;N), \ldots, \overline{\boldsymbol{u}}_M(k;N)$. For notational simplicity, we drop the functional dependence of $\overline{\boldsymbol{u}}_i$ and use $\Psi(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M; \mu(k))$ to represent $\Psi(\overline{\boldsymbol{u}}_1(k;N), \ldots, \overline{\boldsymbol{u}}_M(k;N); \mu(k))$. If the terminal control law specified in Equation (9.5c) is feasible, we have

$$\Psi(\overline{\boldsymbol{u}}_1, \ldots, \overline{\boldsymbol{u}}_M; \mu(k)) = \Phi(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M; \mu(k)).$$

An explicit expression for $\Psi(\cdot)$ for $\phi_i(\cdot)$, $i \in \mathbb{I}_M$ defined in Equation (4.5), p. 34 will be provided in Section 9.2.4.

Assumption 9.3. $(A_{ii}, C_{ii}),\ i \in \mathbb{I}_M$ is observable.

Assumption 9.4. $(A_{ii}, B_{ii}),\ i \in \mathbb{I}_M$ is controllable.

Let

$$G_{ji}(N) = \begin{bmatrix} A_{ji}^{N-1}B_{ji} & A_{ji}^{N-2}B_{ji} & \dots & B_{ji} \end{bmatrix} \qquad g_{ji}(N) = A_{ji}^{N}x_{ji}(k)$$

$$\boldsymbol{G}_i(N) = \begin{bmatrix} G_{1i} \\ G_{2i} \\ \vdots \\ G_{Mi} \end{bmatrix} \qquad \boldsymbol{g}_i(\mu(k); N) = \begin{bmatrix} g_{1i} \\ g_{2i} \\ \vdots \\ g_{Mi} \end{bmatrix}$$

$$\forall\, i, j \in \mathbb{I}_M$$

For each subsystem $i \in \mathbb{I}_M$, define

$$\xi_i(k; N) = \begin{bmatrix} x_{1i}(k+N|k) \\ x_{2i}(k+N|k) \\ \vdots \\ x_{Mi}(k+N|k) \end{bmatrix} = \boldsymbol{G}_i(N)\overline{\boldsymbol{u}}_i(k; N) + \boldsymbol{g}_i(\mu(k); N) \tag{9.7}$$

Thus, $\xi_i(k; 0) = [x_{1i}(k)', x_{2i}(k)', \dots, x_{Mi}(k)']'$. Define $\mathcal{O}_\infty^i,\ i \in \mathbb{I}_M$, closed and convex such that $\mathcal{O}_\infty^1 \times \cdots \times \mathcal{O}_\infty^M \subseteq \mathcal{O}_\infty(A_{\rm cm}, C_{\rm cm})$. Each $\mathcal{O}_\infty^i,\ i \in \mathbb{I}_M$ is a projection of $\mathcal{O}_\infty(A_{\rm cm}, C_{\rm cm})$ on the space of ξ_i, and has infinite extent along the directions that correspond to the unobservable modes of $(A_{\rm cm}, C_{\rm cm})$.

9.2.3 Initialization

For convenience, we use $\mathcal{O}_\infty$ to mean $\mathcal{O}_\infty(A_{\rm cm}, C_{\rm cm})$. To initialize the DCLQR algorithm at $k = 0$, it is necessary to calculate a set of subsystem input trajectories that steers the final predicted system state inside $\mathcal{O}_\infty$. Several formulations to compute such a set of input trajectories exist. A subsystem-based procedure for initialization is used here. Each MPC solves a quadratic program (QP) to calculate initial input trajectories. The collection of subsystems' input trajectories drives the final predicted system state (at the end of the control horizon) inside $\mathcal{O}_\infty$.

To initialize the DCLQR algorithm, each MPC $i \in \mathbb{I}_M$ solves the following QP

$$\overline{\boldsymbol{u}}_i^0(k;N) = \arg \mathcal{L}_i^{\rm N}(\mu(k)) \tag{9.8a}$$

in which

$$\mathcal{L}_i^{\rm N}(\mu(k)) = \min_{\overline{\boldsymbol{z}}_i(k;N)} \ \|\overline{\boldsymbol{z}}_i(k;N)\|^2$$

subject to

$$\boldsymbol{G}_i(N)\boldsymbol{z}_i(k;N) + \boldsymbol{g}_i(\mu(k);N) \in \mathcal{O}_\infty^i \tag{9.8b}$$

$$z_i(k+j|k) \in \Omega_i, \ \ j = 0, 1, \ldots, N-1 \tag{9.8c}$$

$$\forall\, i \in \mathbb{I}_M$$

where $\overline{\boldsymbol{z}}_i(k;N) = [z_i(k|k)', z_i(k+1|k)', \ldots, z_i(k+N-1|k)']'$. At time $k+1$, the input trajectory used for initialization is given by

$$\overline{\boldsymbol{u}}_i^{\,0}(k+1) = \left[u_i^{p(k)}(k+1|k)', \ldots, u_i^{p(k)}(k+N-1|k)', \left(\sum_{j=1}^{M} K_{ij} x_j^{p(k)}(k+N|k) \right)' \right]', \ \forall\, i \in \mathbb{I}_M \tag{9.9}$$

The trajectory $\boldsymbol{u}_i^0(\cdot)$, $i \in \mathbb{I}_M$ in Equation (9.9) is a shifted version of $\overline{\boldsymbol{u}}_i^{p(k)}(k;N)$. For the nominal case, each subsystem-based MPC needs to solve the initialization QP (Equation (9.8)) only once at $k = 0$.

Define

$$\mathcal{U}_i = \Omega_i \times \ldots \times \Omega_i \in \mathbb{R}^{m_i N}$$

Remark 9.1. The constrained stabilizable set $\mathbb{X}$ is defined as the set of subsystem states $x_1, \ldots, x_M$ that can be steered to the origin by applying an admissible set of subsystems' input trajectories $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M$. Let $\xi_i(\cdot;0) = [x_{1i}', x_{2i}', \ldots, x_{Mi}']'$. Define

$$\mathbb{S}_i = \{\xi_i(\cdot;0) \mid \exists\, \overline{\boldsymbol{u}}_i \in \mathcal{U}_i \text{ such that } \xi_i(\cdot;N) \in \mathcal{O}_\infty^i\}, \qquad \text{steerable set}$$

The set $\mathbb{S}_1 \times \ldots \times \mathbb{S}_M$ denotes the set of $\xi_1, \ldots, \xi_M$ for which the initialization QP (Equation (9.8)) is feasible for each $i \in \mathbb{I}_M$.

9.2.4 Method 1. DCLQR with set constraint

One approach to ensure feasibility of the terminal control law is to explicitly enforce the terminal set constraint $\boldsymbol{G}_i(N)\overline{\boldsymbol{u}}_i(k;N) + \boldsymbol{g}_i(N) \in \mathcal{O}_\infty^i$ for each subsystem $i \in \mathbb{I}_M$. It is assumed that $\xi_i(0;0) \in \mathbb{S}_i,\ i \in \mathbb{I}_M$ The initialization QP (Equation (9.8)) is feasible for each $i \in \mathbb{I}_M$. The DCLQR optimization problem of Equation (9.5) can be rewritten as an equivalent finite dimensional optimization problem given by

$$\mathcal{S}_i^N \triangleq \min_{\overline{\boldsymbol{u}}_i(k;N)} \Psi(\overline{\boldsymbol{u}}_1^{p-1}, \ldots, \overline{\boldsymbol{u}}_{i-1}^{p-1}, \overline{\boldsymbol{u}}_i, \overline{\boldsymbol{u}}_{i+1}^{p-1}, \ldots, \overline{\boldsymbol{u}}_M^{p-1}; \mu(k)) = \frac{1}{2}\overline{\boldsymbol{u}}_i(k;N)'\mathfrak{R}_i\overline{\boldsymbol{u}}_i(k;N) + \left(\boldsymbol{r}_i(\mu(k)) + \sum_{j\neq i}\mathcal{H}_{ij}\overline{\boldsymbol{u}}_j^{p-1}(k;N)\right)' \overline{\boldsymbol{u}}_i(k;N) + \text{constant} \quad (9.10a)$$

subject to

$$\overline{\boldsymbol{u}}_i(k;N) \in \mathcal{U}_i, \quad (9.10b)$$

$$\boldsymbol{G}_i(N)\overline{\boldsymbol{u}}_i(k;N) + \boldsymbol{g}_i(N) \in \mathcal{O}_\infty^i \quad (9.10c)$$

in which

$$\mathfrak{R}_i = \left(\mathbb{R}_i + E_{ii}'\mathbb{Q}_iE_{ii}\right) + \sum_{j\neq i}^{M} E_{ji}'\mathbb{Q}_jE_{ji} + \sum_{j=1}^{M} E_{ji}'\sum_{l\neq j}\mathbb{T}_{jl}E_{li}$$

$$\mathcal{H}_{ij} = \sum_{l=1}^{M} E_{li}'\mathbb{Q}_lE_{lj} + \sum_{l=1}^{M} E_{li}'\sum_{s\neq l}\mathbb{T}_{ls}E_{sj}$$

$$\boldsymbol{r}_i(\mu(k)) = E_{ii}'\mathbb{Q}_if_ix_i(k) + \sum_{j\neq i}^{M} E_{ji}'\mathbb{Q}_jf_jx_j(k) + \sum_{j=1}^{M} E_{ji}'\sum_{l\neq j}\mathbb{T}_{jl}f_lx_l(k)$$

$$E_{ii} = \begin{bmatrix} B_i & 0 & \dots & \dots & 0 \\ A_iB_i & B_i & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_i^{N-1}B_i & \dots & \dots & \dots & B_i \end{bmatrix} \quad E_{ij} = \begin{bmatrix} W_{ij} & 0 & \dots & \dots & 0 \\ A_iW_{ij} & W_{ij} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_i^{N-1}W_{ij} & \dots & \dots & \dots & W_{ij} \end{bmatrix} \quad f_i = \begin{bmatrix} A_i \\ A_i^2 \\ \vdots \\ \vdots \\ A_i^N \end{bmatrix}$$

$$\forall\, i, j \in \mathbb{I}_M,\ j \neq i$$

Remark 9.2. From the definition of $\phi_i(\cdot),\ i \in \mathbb{I}_M$ (Equation (4.5))), $\mathbb{Q}_i \geq 0$ and $\mathbb{R}_i > 0$. The optimization problem of Equation (9.10), therefore, minimizes a strictly convex function over a compact set. From Lemma 8.1 (p. 171), a solution to the optimization problem of Equation (9.10) exists and is unique.

Noting that $[\xi_1{}', \dots, \xi_M{}']' = \mathbb{U}x_{\text{cm}}$, in which $\mathbb{U}$ is a unitary matrix, we define

$$\mathbb{D}_C = \{\mu \,|\, \xi_i \in \mathbb{S}_i,\ i \in \mathbb{I}_M\} \qquad \text{domain of controller}$$

The set $\mathbb{D}_C$ is positively invariant for the nominal closed-loop system. An algorithm for DCLQR with the set constraint enforced explicitly is described below.

Algorithm 9.1 (**DCLQR (set constraint)**). Given : $N \in \mathbb{I}_+$, $[\xi_1(0;0)', \dots, \xi_M(0;0)']' \in \mathbb{D}_C$

$\overline{\boldsymbol{u}}_i^0, \mathbb{Q}_i \geq 0, \mathbb{R}_i > 0,\ \forall\, i \in \mathbb{I}_M, p_{\max}(k) \geq 0$ and $\epsilon > 0$

$p \leftarrow 1, \kappa_i \leftarrow \Gamma\epsilon, \Gamma \gg 1$

while $\kappa_i > \epsilon$ for some $i \in \mathbb{I}_M$ and $p \leq p_{\max}(k)$

do $\forall\ i \in \mathbb{I}_M$

$\overline{\boldsymbol{u}}_i^{*(p)}(k;N) \in \ \arg \mathcal{S}_i^N$ (Equation (9.10))

$\overline{\boldsymbol{u}}_i^p(k;N) \leftarrow w_i \overline{\boldsymbol{u}}_i^{*(p)}(k;N) + (1-w_i)\,\overline{\boldsymbol{u}}_i^{p-1}(k;N)$

$\kappa_i \leftarrow \|\overline{\boldsymbol{u}}_i^p(k;N) - \overline{\boldsymbol{u}}_i^{p-1}(k;N)\|$

Transmit $\overline{\boldsymbol{u}}_i^p(k;N)$ to each interconnected subsystem $j \in \mathbb{I}_M,\ j \neq i$

end (do)

$p \leftarrow p+1$

end (while)

In Algorithm 9.1, the state trajectory for subsystem $i \in \mathbb{I}_M$ at iterate p is obtained as $\overline{\boldsymbol{x}}_i^p \leftarrow \overline{\boldsymbol{x}}_i^p\left(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p; \mu(k)\right)$. At times $k > 0$, Algorithm 9.1 is initialized using Equation (9.9). It follows from the definition of $\mathbb{D}_C$ and Algorithm 9.1 that feasibility of the initialization QP (Equation (9.8)) for each $i \in \mathbb{I}_M$ at $k = 0$ ensures nominal feasibility for the optimization problem of Equation (9.10), $\forall\, i \in \mathbb{I}_M$ at all times $k > 0$ and all $p(k) > 0$.

Properties

Lemma 9.3. *Given the DCLQR formulation $\mathcal{S}_i^N,\ \forall\, i \in \mathbb{I}_M$ (see Equation* (9.10)*), the sequence of cost functions $\Phi(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \ldots, \overline{\boldsymbol{u}}_M^p; \mu(k))$ generated by Algorithm 9.1 is nonincreasing with iteration number p.*

The proof is identical to the proof for Lemma 4.4. See p. 44.

From the definition of $\phi_i(\cdot)$, we have $\mathbb{R}_i > 0$. Hence, $\mathfrak{R}_i > 0, \forall\, i \in \mathbb{I}_M$. It follows that $\Psi_i(\cdot)$ is strictly convex. Using convexity of $\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_M$ and Lemma 8.1, the solution $(\overline{\boldsymbol{u}}_1^*, \ldots, \overline{\boldsymbol{u}}_M^*)$ to the optimization problem

$$\min_{(\overline{\boldsymbol{u}}_1, \ldots, \overline{\boldsymbol{u}}_M) \in \mathcal{U}_1 \times \ldots \times \mathcal{U}_M} \Psi(\overline{\boldsymbol{u}}_1, \ldots, \overline{\boldsymbol{u}}_M; \mu(k)) \tag{9.11}$$

exists and is unique.

Lemma 9.4. *Consider $\Psi(\cdot)$ positive definite quadratic and $\Omega_i, i \in \mathbb{I}_M$ convex. Let the solution to Algorithm 9.2.4 after p iterates be $(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p)$ with the cost function value $\Psi(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p; \mu(k))$. Denote the unique solution for the optimization problem of Equation (9.11) by $(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*)$, and let $\Psi(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k))$ represent the corresponding cost function value. The solution obtained at convergence of Algorithm 9.1 satisfies*

$$\lim_{p\to\infty} \Psi(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p; \mu(k)) = \Psi(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k))$$

$$\lim_{p\to\infty} (\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p) = (\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*)$$

A proof is provided in Appendix 9.6.2.

From Lemma 9.4 and the principle of optimality (Bellman, 1957), we have

$$\lim_{p\to\infty} (\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p) = (\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \ldots, \boldsymbol{u}_M^*)$$

9.2.5 Method 2. DCLQR without explicit set constraint

For this case, the DCLQR optimization problem is defined as

$$\mathcal{P}_i^N \triangleq \min_{\overline{\boldsymbol{u}}_i(k;N)} \ \Psi(\overline{\boldsymbol{u}}_1^{p-1},\ldots,\overline{\boldsymbol{u}}_{i-1}^{p-1},\overline{\boldsymbol{u}}_i,\overline{\boldsymbol{u}}_{i+1}^{p-1},\ldots,\overline{\boldsymbol{u}}_M^{p-1};\mu(k)) = \frac{1}{2}\overline{\boldsymbol{u}}_i(k;N)'\mathfrak{R}_i\overline{\boldsymbol{u}}_i(k;N) + \left(\boldsymbol{r}_i(\mu(k)) + \sum_{j\neq i}\mathcal{H}_{ij}\overline{\boldsymbol{u}}_j^{p-1}(k;N)\right)' \overline{\boldsymbol{u}}_i(k;N) + \text{constant} \quad (9.12a)$$

subject to

$$\overline{\boldsymbol{u}}_i \in \mathcal{U}_i \quad (9.12b)$$

Two approaches for DCLQR without explicitly enforcing the terminal set constraint exist. In the first approach, the value of N is altered online. An initial horizon length N_0 is selected such that the initialization QP (Equation (9.8)) is feasible for all $i \in \mathbb{I}_M$. At each iterate p, feasibility of the terminal set constraint $\xi_i^p(k;N) = \boldsymbol{G}_i(N)\overline{\boldsymbol{u}}_i^{*(p)}(k;N) + \boldsymbol{g}_i(\mu(k);N) \in \mathcal{O}_\infty^i,\ i \in \mathbb{I}_M$ is verified. If for some $i \in \mathbb{I}_M$, the terminal set constraint above is violated, N is increased and the input trajectories for each subsystem $i \in \mathbb{I}_M$ are recomputed using the new value of N. An algorithm for DCLQR that is based on this approach is presented in Appendix 9.6.3.

Rather than increase N online, a different approach is adopted here. The idea is to restrict the set of permissible initial states to a positively invariant set in which the terminal set constraint is feasible for each subsystem $i \in \mathbb{I}_M$. This positively invariant set depends on the choice of N. Define

$$\mathbb{S} = \{\mu \,|\, \exists\, \overline{\boldsymbol{u}}_i \in \mathcal{U}_i \ni \boldsymbol{G}_i(N)\overline{\boldsymbol{u}}_i + \boldsymbol{g}_i(\mu;N) \in \mathcal{O}_\infty^i,\ \forall\, i \in \mathbb{I}_M\} \qquad \text{steerable set} \quad (9.13)$$

Let $\mu^+ = [x_1^+, \ldots, x_M^+]$, in which $x_i^+ = A_i x_i + B_i u_i^p(\mu, 0) + \sum_{j \neq i} u_j^p(\mu, 0)$ and $u_i^p(\mu, 0)$ is the control law for subsystem $i \in \mathbb{I}_M$ (see Section 9.2.6). Define

$$\mathbb{D}_C = \{\mu \,|\, \mu^+ \in \mathbb{D}_C, \mu \in \mathbb{S}\} \qquad \text{domain of controller} \tag{9.14}$$

To construct $\mathbb{D}_C$, one may employ standard techniques available in the literature for backward construction of polytopic invariant sets under state and control constraints (Blanchini, 1999; Gutman and Cwikel, 1987; Keerthi and Gilbert, 1987). A brief exposition of the idea is provided below. Let $\mathfrak{X}_0 = \mathbb{S}$. Define

$$\mathfrak{X}_{-1} = \{\mu \,|\, \mu \in \mathbb{S} \text{ and } \exists\, u_i^p(\mu, 0) \in \Omega_i,\ i \in \mathbb{I}_M \ \ni\ \mu^+ \in \mathfrak{X}_0\}$$

For any $l > 0$, we have

$$\mathfrak{X}_{-(l+1)} = \{\mu \,|\, \mu \in \mathbb{S} \text{ and } \exists\, u_i^p(\mu, 0) \in \Omega_i,\ i \in \mathbb{I}_M \ \ni\ \mu^+ \in \mathfrak{X}_{-l}\}$$

The maximal stabilizable (positively invariant) set is given by $\mathfrak{X}_{-\infty} = \cup_{l=0}^{\infty} \mathfrak{X}_l$. It is well known that $\mathfrak{X}_{-\infty}$ is finitely determined if and only if $\mathfrak{X}_{-(l)} = \mathfrak{X}_{-(l-1)}$ for some $l \in \mathbb{I}_+$. In this case, we have $\mathfrak{X}_{-\infty} = \mathfrak{X}_{-l}$. The following algorithm is used for DCLQR (without explicitly enforcing the terminal set constraint).

Algorithm 9.2 (**DCLQR (without terminal set constraint))**. Given : $N \in \mathbb{I}_+$, $\mu(0) \in \mathbb{D}_C$

$\overline{\boldsymbol{u}}_i^0, \mathbb{Q}_i \geq 0, \mathbb{R}_i > 0,\ \forall\, i \in \mathbb{I}_M, p_{\max}(k) \geq 0$ and $\epsilon > 0$

$p \leftarrow 1, \kappa_i \leftarrow \Gamma\epsilon, \Gamma \gg 1$

while $\kappa_i > \epsilon$ for some $i \in \mathbb{I}_M$ and $p \leq p_{\max}(k)$

do $\forall\ i \in \mathbb{I}_M$

$\overline{\boldsymbol{u}}_i^{*(p)}(k;N) \in \ \arg\mathcal{P}_i^N$ (Equation (9.12)

$\overline{\boldsymbol{u}}_i^{p}(\mu(k);N) \leftarrow w_i\overline{\boldsymbol{u}}_i^{*(p)}(k;N) + (1-w_i)\,\overline{\boldsymbol{u}}_i^{p-1}(\mu(k);N)$

$\kappa_i \leftarrow \|\overline{\boldsymbol{u}}_i^{p}(\mu(k);N) - \overline{\boldsymbol{u}}_i^{p-1}(\mu(k);N)\|$

Transmit $\overline{\boldsymbol{u}}_i^{p}$ to each interconnected subsystem $j \in \mathbb{I}_M,\ j \neq i$

end (do)

$p \leftarrow p+1$

end (while)

Let $\eta_i^p(k;N) = \boldsymbol{G}_i(N)\overline{\boldsymbol{u}}_i^p(k;N) + \boldsymbol{g}_i(\mu(k);N),\ \forall\ i \in \mathbb{I}_M$. Since $\eta_i^0 \in \mathcal{O}_\infty^i$ (using the initialization procedure, Section 9.2.3), we have, using convexity of $\mathcal{O}_\infty^i$ that if $\xi_i^p(k;N) \in \mathcal{O}_\infty^i$, then $\eta_i^p(k;N) \in \mathcal{O}_\infty^i$. Let $\eta = [\eta_1{}', \eta_2{}', \ldots, \eta_M{}']'$. From the definitions of $x_{\rm cm}$, η and the unitary transformation $\mathbb{U}$, $x_{\rm cm}^p(k+N|k) = \mathbb{U}\eta^p(k;N)$. Since the 2-norm is unitary invariant (Horn and Johnson, 1985, p. 292), $\|x_{\rm cm}^p(k+N|k)\| = \|\mathbb{U}\eta^p(k;N)\| = \|\eta^p(k;N)\|$. Hence, $x_{\rm cm}^p(k+N|k) \in \mathcal{O}_\infty^1 \times \cdots \times \mathcal{O}_\infty^M \subseteq \mathcal{O}_\infty$.

9.2.6 Closed-loop properties of DCLQR

DCLQR control law. At time k, let the DCLQR algorithm (Algorithm 9.1 or 9.2) be terminated after $p(k) = q$ iterates. Furthermore, let

$$\overline{\boldsymbol{u}}_i^q(\mu(k);N) = [u_i^q(\mu(k),0), u_i^q(\mu(k),1), \ldots, u_i^q(\mu(k),N-1)],\ \forall\, i \in \mathbb{I}_M \tag{9.15}$$

denote the input trajectories obtained by terminating the DCLQR algorithm (Algorithm 9.1 or 9.2) after q iterates. The control law under DCLQR is obtained through a receding hori-

zon implementation of optimal control. The input injected into subsystem i, at time k, is $u_i^q(\mu(k), 0)$.

Let $J^{p(k)}(\mu(k))$ denote the value of the cooperation-based cost function at time k and $p(k)$ iterates. We have,

$$\begin{aligned} J^{p(k)}(\mu(k)) &= \Psi(\overline{\boldsymbol{u}}_1^{p(k)}, \ldots, \overline{\boldsymbol{u}}_M^{p(k)}; \mu(k)) \\ &= \Phi(\boldsymbol{u}_1^{p(k)}, \ldots, \boldsymbol{u}_M^{p(k)}; \mu(k)) \\ &= \sum_{i=1}^{M} w_i \phi_i \left(\boldsymbol{x}_i^{p(k)}, \boldsymbol{u}_i^{p(k)}; x_i(k) \right) \\ &= \sum_{i=1}^{M} w_i \sum_{j=0}^{\infty} L_i \left(x_i^{p(k)}(k+j|k), u_i^{p(k)}(k+j|k) \right) \end{aligned}$$

Theorem 9.1. *Consider the DCLQR framework employing either Algorithm 9.1 or Algorithm 9.2. Let Assumptions 9.1, 9.2 and 9.4 be satisfied. The origin is an asymptotically stable equilibrium point for the nominal closed-loop system* $x_i(k+1) = A_i x_i(k) + B_i u_i^{p(k)}(\mu(k), 0) + \sum_{j \neq i}^{M} W_{ij} u_j^{p(k)}(\mu(k), 0),\ i \in \mathbb{I}_M$, *for all* $\mu(0) \in \mathbb{D}_C$ *and all* $p(k) = 1, 2, \ldots, p_{\max}(k)$.

Proof. Invariance of $B_{\kappa_i}(0)$ and asymptotic stability of the resulting closed loop system can be proved using the discrete version of La Salle's invariance and asymptotic stability theorems (LaSalle, 1976). A useful reference for the stability of ordinary difference equations is the paper by Hurt (1967). At discrete time $k+1$, using the definition of the set of shifted subsystems' input trajectories (Equation (9.9)), and invoking Lemma 9.3, we have

$$0 \leq J^{p(k+1)}(\mu(k+1)) \leq J^0(\mu(k+1)) = J^{p(k)}(\mu(k)) - \sum_{i=1}^{M} w_i L_i(x_i(k), u_i^{p(k)}(k)) \qquad (9.16)$$

$$\forall\, p(k) \geq 0 \ \text{ and all } k \geq 0$$

Since $R_i > 0, u_i^{p(k)}(\mu(k)) \rightarrow 0, \forall\, i \in \mathbb{I}_M$ as $k \rightarrow \infty$. Detectability of $(A_i, Q_i^{1/2})$ implies $x_i(k) \rightarrow 0, \forall\, i \in \mathbb{I}_M$ in the limit $k \rightarrow \infty$. □

Lemma 9.5. *Let $\mu(0) \in \mathbb{D}_C$ and $Q_i, R_i > 0, \forall\, i \in \mathbb{I}_M$. There exists a finite time $T^* \geq 0$, for the nominal closed-loop system under the DCLQR control law such that $x_{\mathrm{cm}}(T^*) \in \mathcal{O}_\infty$.*

A proof is given in Appendix 9.6.4

9.3 Terminal state constraint FC-MPC

A terminal state constraint FC-MPC framework is now described. The terminal state constraint employed by each MPC forces the decentralized states at the end of the control horizon to be at the origin. Let Assumptions 9.1 and 9.4 be satisfied. Closed-loop stability is achieved through the use of a terminal decentralized state constraint and a suitable terminal penalty for the evolution of the interaction model states.

Let $\widetilde{A}_i = \mathrm{diag}(\widehat{A}_i, A_{ii})$, in which $\widehat{A}_i = \mathrm{diag}(A_{i1}, \ldots, A_{i(i-1)}, A_{i(i+1)}, \ldots, A_{iM})$. There exists a unitary transformation $\mathbb{J}_i$ satisfying $\widetilde{A}_i = \mathbb{J}_i A_i \mathbb{J}_i$. Assumption 9.1 implies $\widehat{A}_i$ is a stable matrix. Let $\widetilde{Q}_i = \mathbb{J}_i Q_i \mathbb{J}_i$ and let $\widehat{Q}_i$ be a square, principal submatrix of $\widetilde{Q}_i$ consisting of the first $n_i - n_{ii}$ rows and columns of $\widetilde{Q}_i$. Since $\widetilde{Q}_i \geq 0$, $\widehat{Q}_i \geq 0$. Invariance of detectability under a similarity transformation (Lemma 5.1, p. 90) and stability of $\widehat{A}_i$ gives $(\widehat{A}_i, \widehat{Q}_i)$ is detectable. Let $\widehat{S}_i$ be the solution to the Lyapunov equation $\widehat{A}_i' \widehat{S}_i \widehat{A}_i - \widehat{S}_i = -\widehat{Q}_i$. Define $\widetilde{S}_i = \mathrm{diag}(\widehat{S}_i, 0_{n_{ii}})$ and let $S_i = \mathbb{J}_i \widetilde{S}_i \mathbb{J}_i$. The state and input penalties over the control horizon are now defined.

Assumption 9.5. Let Assumption 9.2 hold and let

$$\begin{aligned}\mathbb{Q}_i &= \text{diag}\Big(Q_i(1),\ldots,Q_i(N-1),S_i\Big)\\ \mathbb{R}_i &= \text{diag}\Big(R_i(0),R_i(1),\ldots,R_i(N-1)\Big)\end{aligned},\ i\in\mathbb{I}_M$$

Consider a control horizon N. The terminal state constraint FC-MPC optimization problem, with $\phi_i(\cdot)$ defined in Equation (4.5), can be written as

$\mathcal{P}_i^c$: **Terminal state constraint FC-MPC**

$$\min_{\overline{\boldsymbol{u}}_i(k;N)} \quad \frac{1}{2}\overline{\boldsymbol{u}}_i(k;N)'\mathfrak{R}_i\overline{\boldsymbol{u}}_i(k;N)+\left(\boldsymbol{r}_i(\mu(k))+\sum_{j\neq i}\mathcal{H}_{ij}\overline{\boldsymbol{u}}_j^{p-1}(k;N)\right)'\overline{\boldsymbol{u}}_i(k;N) \tag{9.17a}$$

subject to

$$u_i(t|k)\in\Omega_i,\ \ k\leq t\leq k+N-1 \tag{9.17b}$$

$$x_{ii}(k+N|k)=0 \tag{9.17c}$$

$$\begin{aligned}\mathfrak{R}_i &= w_i\big(\mathbb{R}_i+E_{ii}{}'\mathbb{Q}_iE_{ii}\big)+\sum_{j\neq i}^{M}w_jE_{ji}{}'\mathbb{Q}_jE_{ji}\\ \mathcal{H}_{ij} &= \sum_{l=1}^{M}w_lE_{li}{}'\mathbb{Q}_lE_{lj}\\ \boldsymbol{r}_i(\mu(k)) &= w_iE_{ii}{}'\mathbb{Q}_if_ix_i(k)+\sum_{j\neq i}^{M}w_jE_{ji}{}'\mathbb{Q}_jf_jx_j(k)\end{aligned}$$

An algorithm for terminal state constraint FC-MPC is described below.

Algorithm 9.3 (**Terminal state constraint FC-MPC)**. Given :

$\overline{\boldsymbol{u}}_i^0, x_i(k), \mathbb{Q}_i, \mathbb{R}_i,\ \forall\, i\in\mathbb{I}_M, p_{\max}(k)\geq 0$ and $\epsilon>0$

$p \leftarrow 1, \kappa_i \leftarrow \Gamma\epsilon, \Gamma \gg 1$

while $\kappa_i > \epsilon$ for some $i \in \mathbb{I}_M$ and $p \leq p_{\max}(k)$

do $\forall\ i \in \mathbb{I}_M$

$\overline{\boldsymbol{u}}_i^{*(p)} \in \arg \mathcal{P}_i^c$ (9.17)

$\overline{\boldsymbol{u}}_i^p \leftarrow w_i \overline{\boldsymbol{u}}_i^{*(p)} + (1 - w_i)\, \overline{\boldsymbol{u}}_i^{p-1}$

$\kappa_i \leftarrow \|\overline{\boldsymbol{u}}_i^p - \overline{\boldsymbol{u}}_i^{p-1}\|$

Transmit $\overline{\boldsymbol{u}}_i^p$ to each interconnected subsystem $j \in \mathbb{I}_M,\ j \neq i$

end (do)

$\overline{\boldsymbol{x}}_i^p \leftarrow \overline{\boldsymbol{x}}_i^p \left(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p; \mu(k)\right),\ \forall\, i \in \mathbb{I}_M$

$p \leftarrow p + 1$

end (while)

Remark 9.3. To initialize terminal state constraint FC-MPC, each MPC solves a QP to determine a feasible initial trajectory. The initialization QP solved by each subsystem is similar to Equation (9.8) with the terminal set constraint in Equation (9.8b) replaced by the terminal point constraint $x_{ii}(N) = \mathcal{C}_N(A_{ii}, B_{ii})\overline{\boldsymbol{u}}_i + A_{ii}^N x_{ii} = 0$. Define

$$\mathbb{S}_i = \{x_{ii} \,|\, \exists\, \overline{\boldsymbol{u}}_i \in \mathcal{U}_i \text{ such that } x_{ii}(N) = 0\} \qquad \text{steerable set}$$

For $x_{ii} \in \mathbb{S}_i,\ i \in \mathbb{I}_M$, the initialization QP is feasible. Let

$$\mathbb{D}_{R_i} = \mathbb{R}^{n_{i1}} \times \cdots \times \mathbb{S}_i \times \cdots \mathbb{R}^{n_{iM}},\ i \in \mathbb{I}_M \qquad \text{domain of regulator}$$

The domain of the controller is given by

$$\mathbb{D}_C = \{\mu \,|\, \mu^+ \in \mathbb{D}_C, x_i \in \mathbb{D}_{R_i},\ i \in \mathbb{I}_M\} \qquad \text{domain of controller}$$

The set $\mathbb{D}_C$ is positively invariant. For $\mu(0) \in \mathbb{D}_C$, convexity of Ω_i and Algorithm 9.3 guarantee feasibility of $\mathcal{P}_i^c$ for all $i \in \mathbb{I}_M$ and all $p(k) > 0, k \geq 0$.

Remark 9.4. At time k, let Algorithm 9.3 be terminated after $p(k)$ iterates. For each subsystem $i \in \mathbb{I}_M$ the input injected is $u_i^{p(k)}(\mu(k), 0)$. The domain of attraction for the nominal closed-loop system is the set $\mathbb{D}_C$.

Remark 9.5. For each subsystem $i \in \mathbb{I}_M$, let Assumptions 9.1, 9.4 and 9.5 hold. The nominal closed-loop system $x_i(k+1) = A_i x_i(k) + B_i u_i^{p(k)}(\mu(k), 0) + \sum_{j \neq i}^{M} W_{ij} u_j^{p(k)}(\mu(k), 0),\ i \in \mathbb{I}_M$, is exponentially stable for all $\mu(0) \in \mathbb{D}_C$ and all $p(k) > 0$.

Remark 9.6. For both nonsingular and singular A_{cm}, there exists a finite time T after which the closed-loop system evolves in a closed and bounded (hence compact) subset of $\mathbb{R}^n$.

9.4 Examples

9.4.1 Distillation column of Ogunnaike and Ray (1994)

We revisit the distillation column of (Ogunnaike and Ray, 1994, p. 813) described in Section 4.7.1, p. 54. Regulator parameters and constraints are given in Table 9.1. Here, we evaluate the performance of terminal penalty FC-MPC (FC-MPC (tp)) and terminal control FC-MPC (FC-MPC (tc)). The performance of either formulation is compared against centralized constrained LQR (CLQR) (Scokaert and Rawlings, 1998; Sznaier and Damborg, 1990). The closed-

loop performance of CLQR, FC-MPC (tp) and FC-MPC(tc) when the temperature setpoints for trays 21 and 7 are altered by $-0.5^{\circ}C$ and $0.5^{\circ}C$ respectively is shown in Figures 9.1 and 9.2. The terminal control formulation of Section 9.2.4 is used. For terminal control FC-MPC, an initial control horizon $N_0 = 15$ is used. The initialization QP for MPC 2 is infeasible for this value of N_0. Subsequently, N_0 is increased and feasible initialization trajectories for both MPCs are obtained for $N_0 = 19$.

Table 9.1: Distillation column model of Ogunnaike and Ray (1994). Bound constraints on inputs L and V. Regulator parameters for MPCs.

$$\text{-0.8} \leq \text{V} \leq \text{0.8}$$
$$\text{-0.8} \leq \text{L} \leq \text{0.8}$$

$Q_{y_1} = 50$	$Q_{y_2} = 50$
$R_1 = 1$	$R_2 = 1$
$\varepsilon_1 = 10^{-2}$	$\varepsilon_2 = 10^{-2}$

A closed-loop performance comparison for the different MPCs is given in Table 9.2. Both FC-MPC (tp, 5 iterates) and FC-MPC (tc, 5 iterates) perform poorly in comparison to CLQR. FC-MPC (tc) incurs a lower control cost compared to FC-MPC (tp) however. If FC-MPC (tc) and FC-MPC (tp) are terminated after 20 iterates, the performance loss (relative to CLQR performance) incurred under FC-MPC (tp) is nearly 50% greater than FC-MPC (tc).

Table 9.2: Closed-loop performance comparison of CLQR, FC-MPC (tp) and FC-MPC (tc).

	Λ_{cost}	$\Delta\Lambda_{\text{cost}}\%$
CLQR	9.74	$--$
FC-MPC (tp, 5 iterates)	32.51	234
FC-MPC (tc, 5 iterates)	25	156.4
FC-MPC (tp, 20 iterates)	13.13	34.8
FC-MPC (tc, 20 iterates)	11.4	17
FC-MPC (tp, 50 iterates)	9.83	0.9
FC-MPC (tc, 50 iterates)	9.77	0.36

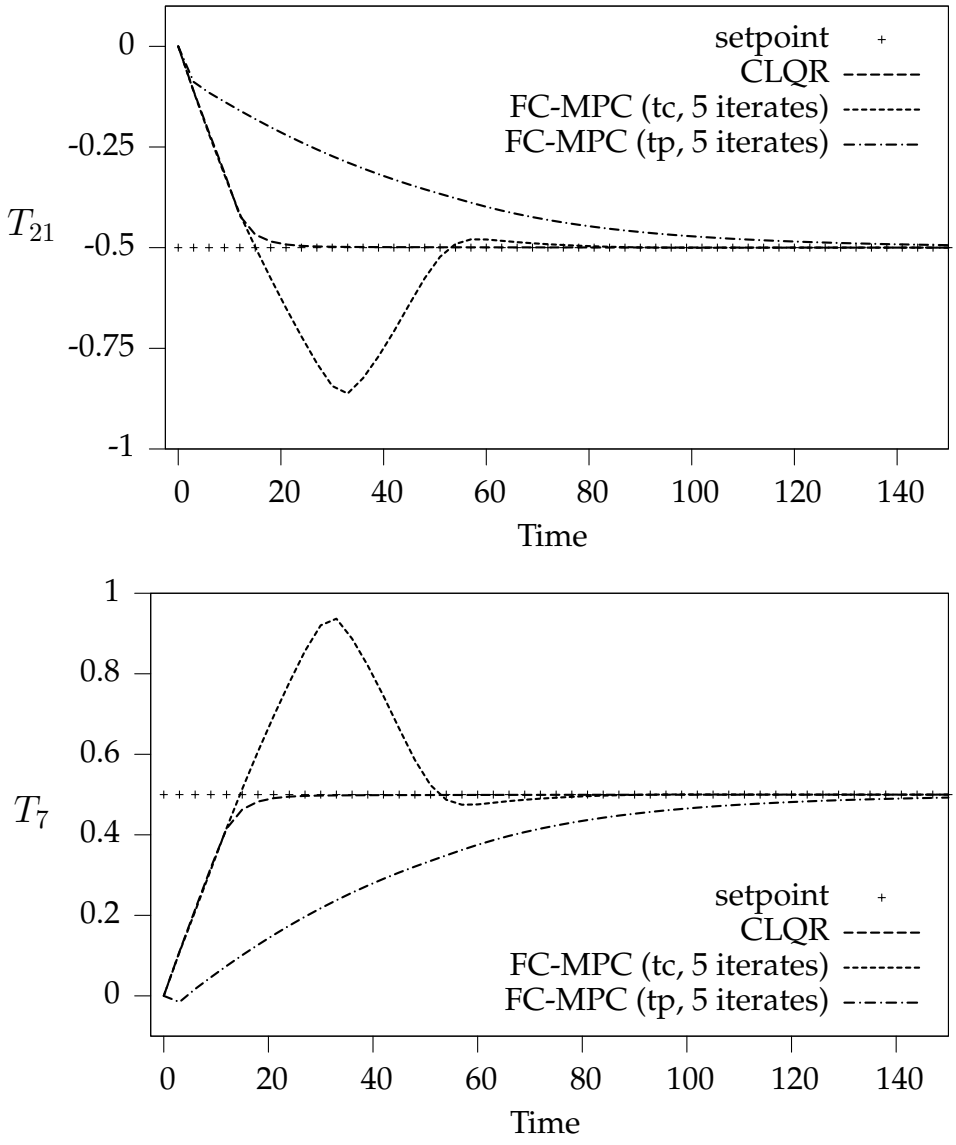

Figure 9.1: Setpoint tracking performance of CLQR, FC-MPC (tc) and FC-MPC (tp). Tray temperatures of the distillation column.

9.4.2 Unstable three subsystem network

We consider the plant consisting of three unstable subsystems, described in Section 4.7.3. Input constraints and controller parameters are given in Table 9.3. For each MPC, a control horizon $N = 10$ is used. The performance of terminal state constraint FC-MPC (FC-MPC (tsc)) and

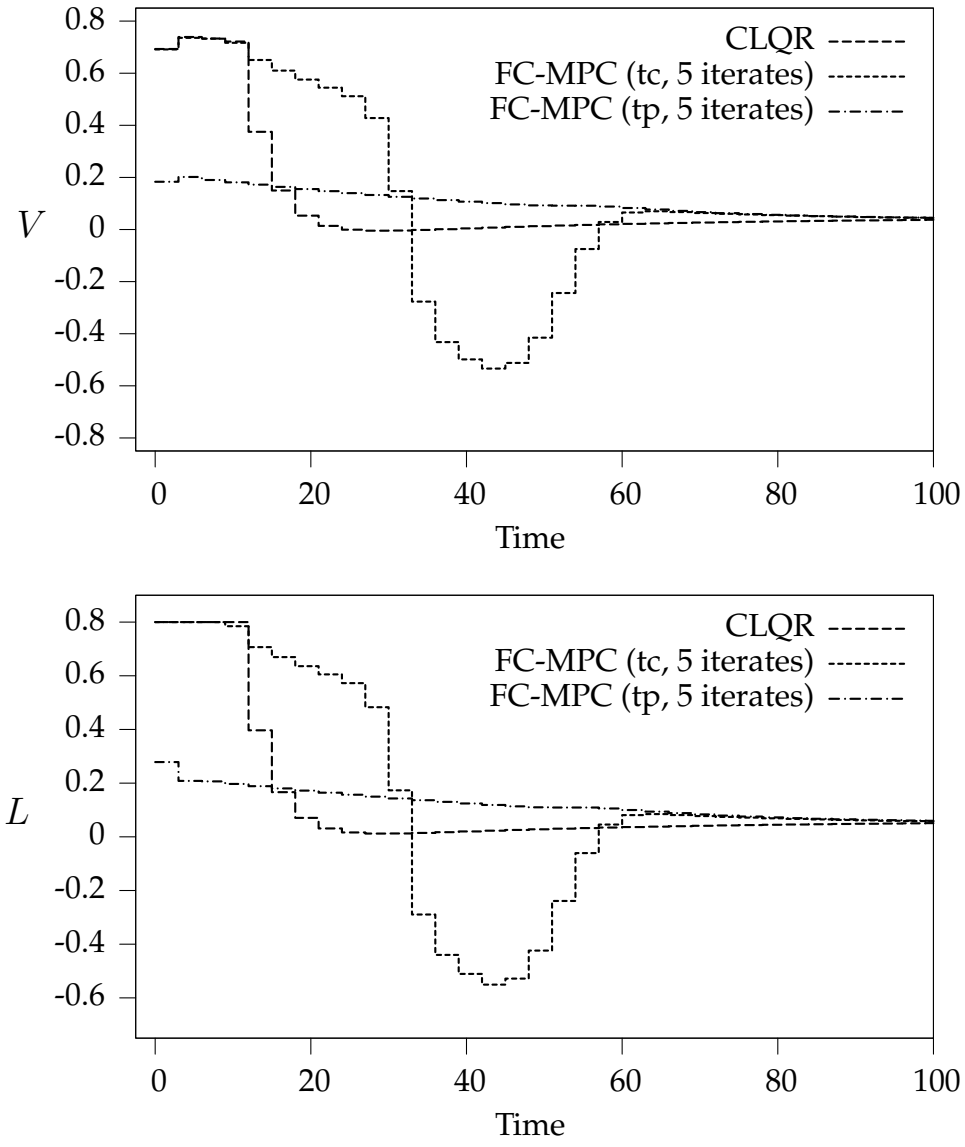

Figure 9.2: Setpoint tracking performance of CLQR, FC-MPC (tc) and FC-MPC (tp). Inputs (V and L) for the distillation column.

terminal penalty FC-MPC are compared against CLQR. Since each subsystem has an unstable decentralized mode, a terminal state constraint that forces the unstable mode to the origin at the end of the control horizon is necessary for terminal penalty FC-MPC (see Theorem 4.2, p. 53). For terminal constraint FC-MPC, all the decentralized modes are forced to be at the

origin at the end of the control horizon.

Table 9.3: Three subsystems, each with an unstable decentralized pole. Input constraints and regulator parameters.

$$\begin{array}{c} \text{-1} \leq u_1 \leq 1 \\ \text{-0.17} \leq u_2 \leq 0.17 \\ \text{-0.2} \leq u_3 \leq 0.2 \\ \text{-0.1} \leq u_4 \leq 0.1 \\ \text{-0.6} \leq u_5 \leq 0.6 \end{array}$$

$$\left| \begin{array}{c} Q_{y_1} = 25 \\ R_1 = 1 \\ \varepsilon_1 = 10^{-6} \end{array} \right\| \left. \begin{array}{c} Q_{y_2} = 25 \\ R_2 = 1 \\ \varepsilon_2 = 10^{-6} \end{array} \right\| \left. \begin{array}{c} Q_{y_3} = 25 \\ R_3 = 1 \\ \varepsilon_3 = 10^{-6} \end{array} \right|$$

A setpoint change of 0.5 and -0.5 is made to outputs y_1 and y_5 respectively. The performance of FC-MPC (tsc, 1 iterate), FC-MPC (tp, 1 iterate) and CLQR is shown in Figures 9.3 and 9.4. A closed-loop performance comparison for the different MPCs is given in Table 9.4. The performance loss (compared to CLQR) with FC-MPC (tsc) terminated after just 1 iterate is $\sim 17\%$. If 5 iterates per sampling interval are possible, the performance loss drops to about 2%.

Table 9.4: Closed-loop performance comparison of CLQR, FC-MPC (tp) and FC-MPC (tsc).

	Λ_{cost}	$\Delta\Lambda_{\text{cost}}\%$
CLQR	0.43	$--$
FC-MPC (tp, 1 iterate)	0.46	8.4
FC-MPC (tsc, 1 iterate)	0.5	16.7
FC-MPC (tp, 5 iterates)	0.426	0.2
FC-MPC (tsc, 5 iterates)	0.432	1.57

9.5 Discussion and conclusions

For DCLQR without an explicit terminal set constraint and with N increased online, the computational overhead is in the determination of an N that drives the system state to $\mathcal{O}_\infty$. The efficiency of this terminal control FC-MPC algorithm depends on the choice of N_0; a judicious

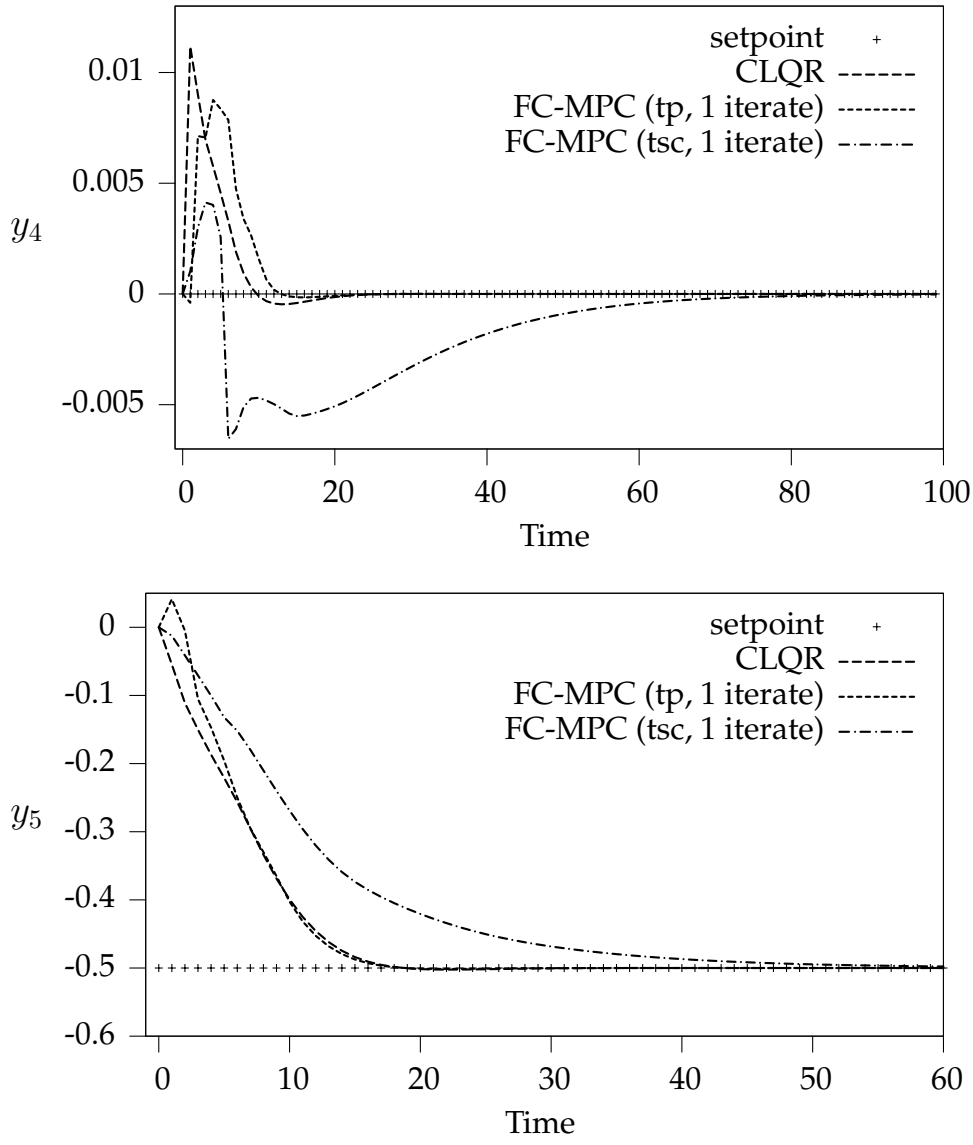

Figure 9.3: Three subsystem example. Each subsystem has an unstable decentralized pole. Performance comparison of CLQR, FC-MPC (tsc) and FC-MPC (tp). Outputs y_4 and y_5.

choice of N_0 and an effective heuristic for increasing N improves algorithmic efficiency. As recommended in Scokaert and Rawlings (1998), one possible choice is to increase N geometrically. The quantities Π_{cm}, K_{cm} and $\mathcal{O}_\infty$ are determined using a centralized calculation. Computation of Π_{cm}, K_{cm} and $\mathcal{O}_\infty$ is performed offline however. The quantities Π_{cm} and K_{cm} need to be re-

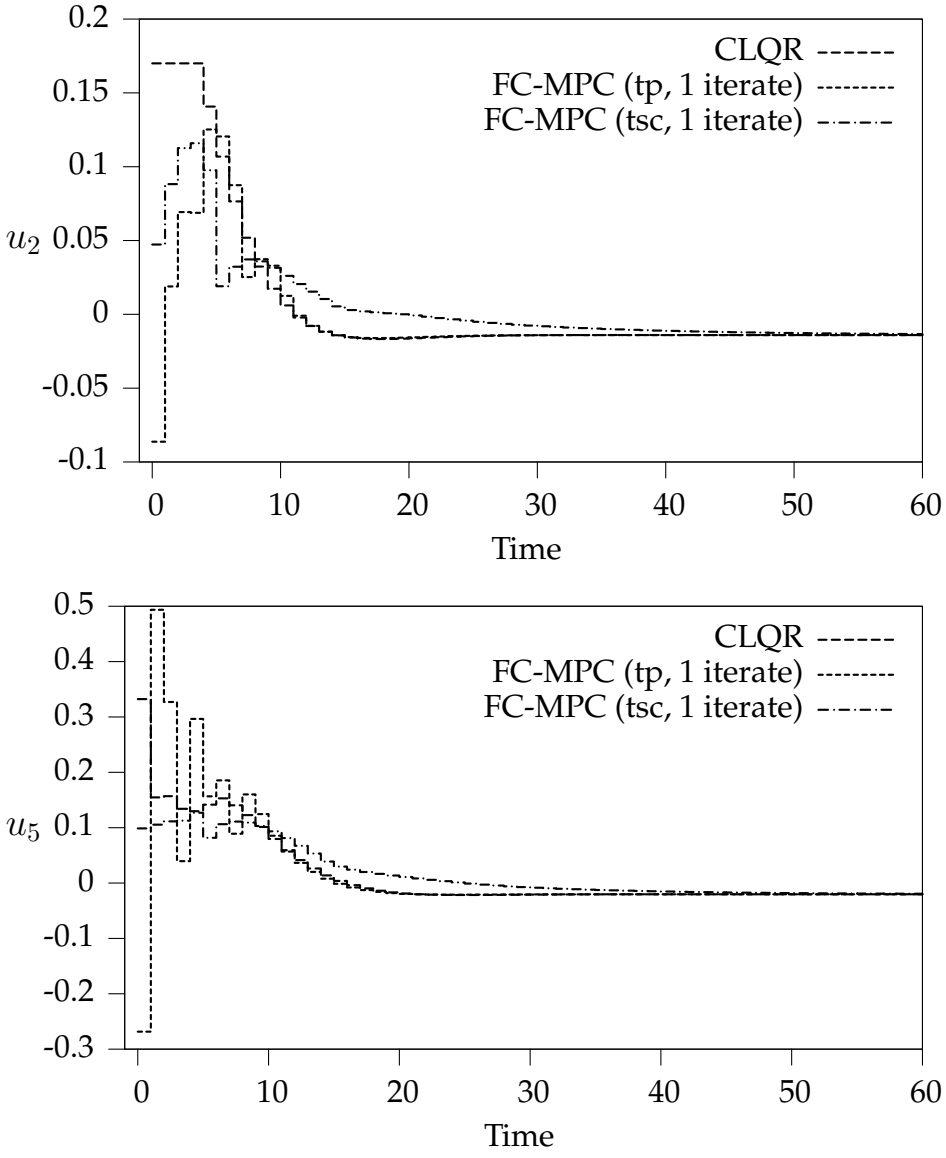

Figure 9.4: Three subsystem example. Each subsystem has an unstable decentralized pole. Performance comparison of CLQR, FC-MPC (tsc) and FC-MPC (tp). Inputs u_2 and u_5.

calculated only if the regulator parameters are altered and/or the system model changes. The computation of $K_{\rm cm}$ and $\Pi_{\rm cm}$ can be parallelized using techniques available in the literature for parallel solution of the discrete Riccati equation (Lainiotis, 1975; Lainiotis et al., 1996). The set $\mathcal{O}_\infty$ needs to be recomputed everytime a setpoint change is planned and/or the system

model, constraints are altered. The overhead associated with determination of a suitable N that ensures feasibility of the unconstrained feedback law and computation of $\mathcal{O}_\infty$ are issues not confined to distributed MPC. They are key concerns in centralized MPC as well.

In certain special cases, stabilizing (suboptimal) decentralized feedback laws may exist for each subsystem in a neighborhood of the origin. Such situations typically arise when the interactions among the subsystems are sufficiently weak (Sandell-Jr. et al., 1978). This class of problems can be treated as a special case. Let $K_d = \mathrm{diag}(K_{d_1}, \ldots, K_{d_M})$ in which $u_i = K_{d_i} x_i$ is the local feedback law for subsystem $i \in \mathbb{I}_M$ for $x_i \in \Lambda_i$ and Λ_i is closed, convex and encloses the origin. The terminal penalty Π_{cm} is obtained as the solution to the Lyapunov solution

$$(A_{\mathrm{cm}} - B_{\mathrm{cm}} K_d)' \Pi_{\mathrm{cm}} (A_{\mathrm{cm}} - B_{\mathrm{cm}} K_d) - \Pi_{\mathrm{cm}} = -(\mathcal{Q} + K_d' \mathcal{R} K_d)$$

Substituting $K_d = K_{\mathrm{cm}}$ in the equation above, we recover Equation (9.3a).

The main difference between Algorithm 9.2 and Algorithm 9.4 is that in the former case, we circumvent the need to search online for a suitable N that steers $\xi_i(\cdot; N)$ inside $\mathcal{O}_\infty^i$ by restricting the allowable initial states to a suitable positively invariant set. This restriction also simplifies the algorithm significantly. The main disadvantage of explicitly including the terminal set constraint in the FC-MPC optimization (Algorithm 9.1) is that the resulting controller response may be excessively aggressive, especially for small N.

9.6 Appendix

9.6.1 Proof for Lemma 9.2

Proof. From the Hautus lemma for stabilizability of the pair (A, B) (Sontag, 1998), we have (A, B) is stabilizable if and only if

$$\operatorname{rank}\begin{bmatrix}\lambda I - A & B\end{bmatrix} = n,\ \forall\ |\lambda| \geq 1 \tag{9.18}$$

$(\mathbb{A}, \mathbb{B})$ **stabilizable** $\implies$ $(\mathcal{A}, \mathcal{B})$ **stabilizable.** Consider $|\lambda| \geq 1$. We have,

$$\begin{bmatrix}\lambda I - \mathcal{A} & 0 & \mathcal{B} \\ 0 & \lambda I - \mathcal{A}_s & \mathcal{B}_s\end{bmatrix}$$

has $n + n_s$ independent rows. Therefore, $[\lambda I - \mathcal{A},\ 0,\ \mathcal{B}]$ has n independent rows, which implies $[\lambda I - \mathcal{A},\ \mathcal{B}]$ has n independent rows. Hence, $(\mathcal{A}, \mathcal{B})$ is stabilizable.

$(\mathcal{A}, \mathcal{B})$ **stabilizable** $\implies$ $(\mathbb{A}, \mathbb{B})$ **stabilizable.** Consider $|\lambda| \geq 1$. We have $[\lambda I - \mathcal{A},\ \mathcal{B}]$ has n independent rows, which implies $[\lambda I - \mathcal{A},\ 0,\ \mathcal{B}]$ has n independent rows. Since $\mathcal{A}_s$ is stable, $\lambda I - \mathcal{A}_s$ has n_s independent rows. Due to the position of the zeros,

$$\begin{bmatrix}\lambda I - \mathcal{A} & 0 & \mathcal{B} \\ 0 & \lambda I - \mathcal{A}_s & \mathcal{B}_s\end{bmatrix}$$

has $n + n_s$ independent rows *i.e.*, $(\mathbb{A}, \mathbb{B})$ is stabilizable. □

9.6.2 Proof for Lemma 9.4

Proof. Since the level set

$$S_0 = \{(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M) \mid \Psi(\overline{\boldsymbol{u}}_1, \ldots, \overline{\boldsymbol{u}}_M; \mu(k)) \leq \Psi(\overline{\boldsymbol{u}}_1^0, \ldots, \overline{\boldsymbol{u}}_M^0; \mu(k))\}$$

is closed and bounded (hence compact), a limit point for Algorithm 9.1 exists. We know that $(\overline{\boldsymbol{u}}_1^*, \ldots, \overline{\boldsymbol{u}}_M^*)$ is the unique solution for the centralized MPC optimization problem (Equation (9.11)). Let $\Psi^* = \Psi(\overline{\boldsymbol{u}}_1^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k))$. Assume that the sequence $(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p)$, generated by Algorithm 9.1, converges to a feasible subset of the nonoptimal level set

$$S_\infty = \{(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M) \mid \Psi(\overline{\boldsymbol{u}}_1, \ldots, \overline{\boldsymbol{u}}_M; \mu(k)) = \Psi^\infty = \Phi^\infty\}$$

Since $\Psi(\cdot)$ is strictly convex and by assumption of nonoptimality $\Psi^\infty > \Psi^*$. Let $(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty) \in S_\infty$ be generated by Algorithm 9.1 for p large. To establish convergence of Algorithm 9.1 to a point rather than a limit set, we assume the contrary and show a contradiction. Suppose that Algorithm 9.1 does not converge to a point. Our assumption here implies that there exists $(\overline{\boldsymbol{v}}_1, \ldots, \overline{\boldsymbol{v}}_M) \in S_\infty$ generated by the subsequent iterate of Algorithm 9.1 with $(\overline{\boldsymbol{v}}_1, \ldots, \overline{\boldsymbol{v}}_M) \neq (\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty)$.

Consider the following optimization problem for each $i \in \mathbb{I}_M$

$$\overline{\boldsymbol{z}}_i^\infty = \arg\min_{\overline{\boldsymbol{u}}_i} \ \Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_{i-1}^\infty, \overline{\boldsymbol{u}}_i, \overline{\boldsymbol{u}}_{i+1}^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k)) \tag{9.19a}$$

$$\overline{\boldsymbol{u}}_i \in \mathcal{U}_i \tag{9.19b}$$

By assumption, there exists at least one i for which $\overline{\boldsymbol{z}}_i^\infty \neq \overline{\boldsymbol{u}}_i^\infty$. By definition, $\overline{\boldsymbol{v}}_i = w_i\overline{\boldsymbol{z}}_i^\infty + (1-w_i)\overline{\boldsymbol{u}}_i^\infty$, $\forall\, i \in \mathbb{I}_M$. Since $(\overline{\boldsymbol{v}}_1, \ldots, \overline{\boldsymbol{v}}_M) \in S_\infty$, $\Psi(\overline{\boldsymbol{v}}_1, \ldots, \overline{\boldsymbol{v}}_M; x(k)) = \Phi^\infty$. Using convexity of $\Psi(\cdot)$, we have

$$\begin{aligned}
\Psi^\infty = \Psi(\overline{\boldsymbol{v}}_1, \ldots, \overline{\boldsymbol{v}}_M; \mu(k)) &= \Psi(w_1\overline{\boldsymbol{z}}_1^\infty + (1-w_1)\overline{\boldsymbol{u}}_1^\infty, \ldots, w_M\overline{\boldsymbol{z}}_M^\infty + (1-w_M)\overline{\boldsymbol{u}}_M^\infty; \mu(k)) \\
&< w_1\Psi(\overline{\boldsymbol{z}}_1^\infty, \overline{\boldsymbol{u}}_2^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; x(k)) + \ldots \\
&\qquad \ldots + w_M\Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_{M-1}^\infty, \overline{\boldsymbol{z}}_M^\infty; x(k)) \\
&< w_1\Psi^\infty + \ldots w_M\Psi^\infty \\
&= \Psi^\infty = \Phi^\infty
\end{aligned}$$

in which the strict inequality follows from $\overline{\boldsymbol{z}}_i^\infty \neq \overline{\boldsymbol{u}}_i^\infty$ for at least one $i \in \mathbb{I}_M$. Hence, a contradiction. Suppose now that $(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p) \to (\overline{\boldsymbol{u}}_1^\infty, \overline{\boldsymbol{u}}_2^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty) \neq (\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*)$. From uniqueness of the optimizer, $\Psi(\overline{\boldsymbol{u}}_1^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k)) < \Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k))$. Since $(\overline{\boldsymbol{u}}_1^p, \ldots, \overline{\boldsymbol{u}}_M^p)$, generated using Algorithm 9.1, converges to $(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty)$, we have for each $i \in \mathbb{I}_M$ that

$$\overline{\boldsymbol{u}}_i^\infty = \arg\min_{\overline{\boldsymbol{u}}_i} \ \Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_{i-1}^\infty, \overline{\boldsymbol{u}}_i, \overline{\boldsymbol{u}}_{i+1}^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k)) \tag{9.20a}$$

$$\overline{\boldsymbol{u}}_i \in \mathcal{U}_i \tag{9.20b}$$

From Lemma 9.1,

$$\nabla_{\overline{\boldsymbol{u}}_j}\Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k))'(\overline{\boldsymbol{u}}_j^* - \overline{\boldsymbol{u}}_j^\infty) \geq 0, \ \forall\, j \in \mathbb{I}_M$$

Define $\Delta\overline{\boldsymbol{u}}_j = \overline{\boldsymbol{u}}_j^* - \overline{\boldsymbol{u}}_j^\infty$. We have, from our assumption $(\overline{\boldsymbol{u}}_1^\infty, \overline{\boldsymbol{u}}_2^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty) \neq (\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*)$,

that $\Delta\overline{\boldsymbol{u}}_i \neq 0$ for at least one index $i \in \mathbb{I}_M$. A second order Taylor series expansion around $(\overline{\boldsymbol{u}}_1^\infty, \overline{\boldsymbol{u}}_2^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty)$ gives

$$\begin{aligned}\Psi(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k)) &= \Psi(\overline{\boldsymbol{u}}_1^\infty + \Delta\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2^\infty + \Delta\overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M^\infty + \Delta\overline{\boldsymbol{u}}_M; \mu(k)) \\ &= \Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k)) + \underbrace{\sum_{j=1}^{M} \nabla_{\overline{\boldsymbol{u}}_j} \Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k))' \Delta\overline{\boldsymbol{u}}_j}_{\geq 0, \text{ Lemma 9.1}} \\ &\quad + \underbrace{\frac{1}{2} \begin{bmatrix} \Delta\overline{\boldsymbol{u}}_1 \\ \vdots \\ \Delta\overline{\boldsymbol{u}}_M \end{bmatrix}' \nabla^2 \Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k)) \begin{bmatrix} \Delta\overline{\boldsymbol{u}}_1 \\ \vdots \\ \Delta\overline{\boldsymbol{u}}_M \end{bmatrix}}_{\geq 0, \text{ since } \Psi(\cdot) \text{ p.d. quadratic}} \end{aligned} \tag{9.21}$$

Using Equation (9.21) and optimality of $(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*)$ gives

$$\begin{aligned}\Psi(\overline{\boldsymbol{u}}_1^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k)) &= \Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k)) + \beta(\Delta\overline{\boldsymbol{u}}_1, \ldots, \Delta\overline{\boldsymbol{u}}_M) \\ &\leq \Psi(\overline{\boldsymbol{u}}_1^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k)) \end{aligned} \tag{9.22}$$

in which $\beta(\cdot)$ is a positive definite function (from Equation (9.21)). We have from Equation (9.22) that $\beta(\cdot) \leq 0$, which implies $\beta(\Delta\overline{\boldsymbol{u}}_1, \ldots, \Delta\overline{\boldsymbol{u}}_M) = 0$.

It follows, therefore, that $\overline{\boldsymbol{u}}_j^\infty = \overline{\boldsymbol{u}}_j^*, \ \forall\ j \in \mathbb{I}_M$. Hence, $\Psi(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k)) = \Psi(\overline{\boldsymbol{u}}_1^\infty, \overline{\boldsymbol{u}}_2^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k))$.

□

Alternate proof. **Claim:** If $\Psi(\overline{\boldsymbol{u}}_1^p, \ldots, \overline{\boldsymbol{u}}_M^p; \mu(k)) \to \Psi(\overline{\boldsymbol{u}}^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k))$, $\overline{\boldsymbol{u}}_j^p \to \overline{\boldsymbol{u}}_j^*, \ \forall\ j \in \mathbb{I}_M$ as $p \to \infty$.

Let L_* be the level set

$$L_* = \left\{ (\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M) \mid \Psi(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M; \mu(k)) \leq \Psi(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k)) \right\} \tag{9.23}$$

Optimality of $\Psi(\overline{\boldsymbol{u}}_1^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k))$ and strict convexity of $\Psi(\cdot)$ gives $L_* = \{(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*)\}$.

Define L_p to be the level set

$$L_p = \left\{ (\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M) \mid \Psi(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M; \mu(k)) \leq \Psi(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p; \mu(k)) \right\}$$

By construction, $(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p) \in L_p$. Since $\Psi(\cdot)$ is a nonincreasing function of the iteration number p (Lemma 9.3), and is bounded below (hence convergent), we have

$$L_{p+1} \subseteq L_p \subseteq L_{p-1} \subseteq \ldots\ldots\ldots \subseteq L_0 \subseteq S_0$$

In the limit as $p \to \infty$, and since $\Psi(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k)) = \Psi(\overline{\boldsymbol{u}}_1^\infty, \overline{\boldsymbol{u}}_2^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty; \mu(k))$ (from the first part of the lemma), we can write

$$\begin{aligned} L_\infty &= \{(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M) \mid \Psi(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M; \mu(k)) \leq \Psi(\overline{\boldsymbol{u}}_1^\infty, \overline{\boldsymbol{u}}_2^\infty \ldots., \overline{\boldsymbol{u}}_M^\infty; \mu(k))\} \\ &= \{(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M) \mid \Psi(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M; \mu(k)) \leq \Psi(\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*; \mu(k))\} \\ &= L_* \end{aligned} \tag{9.24}$$

It follows by construction of the level sets that $(\overline{\boldsymbol{u}}_1^\infty, \overline{\boldsymbol{u}}_2^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty) \in L_\infty$. Using the definition of L_* (Equation (9.23)) and from Equation (9.24), we have $(\overline{\boldsymbol{u}}_1^\infty, \overline{\boldsymbol{u}}_2^\infty, \ldots, \overline{\boldsymbol{u}}_M^\infty) = (\overline{\boldsymbol{u}}_1^*, \overline{\boldsymbol{u}}_2^*, \ldots, \overline{\boldsymbol{u}}_M^*)$, which proves the second part of the lemma. □

9.6.3 DCLQR with N increased online (without terminal set constraint)

A subsystem-based algorithm to ensure feasibility of the terminal control law without explicitly enforcing the terminal set constraint, and in which N is increased online, is described below.

Algorithm 9.4 (**DCLQR (without set constraint))**.

Given: $x_i(k), \mathbb{Q}_i \geq 0, \mathbb{R}_i > 0, \ \forall\, i \in \mathbb{I}_M$

$p_{\max}(k) \geq 0, \epsilon > 0$ and $p \leftarrow 1$

1. Choose a finite horizon N_0. $N \leftarrow N_0$.
2. **for** $i \in \mathbb{I}_M$

   Compute $\overline{\boldsymbol{u}}_i^0(k;N) \in \arg \mathcal{L}_i^{\mathrm{N}}(\mu(k))$

   Construct $\boldsymbol{u}_i^0(k)$ (Remark 9.7)

   Transmit $\boldsymbol{u}_i^0(k)$ to each subsystem $j \in \mathbb{I}_M, \ j \neq i$.

   end (for)
3. $\rho_i \leftarrow \Gamma\epsilon, \Gamma \gg 1$
4. **while** $\rho_i > \epsilon$ for some $i \in \mathbb{I}_M$ and $p \leq p_{\max}$

   (a) **do** $\forall\, i \in \mathbb{I}_M$

   $\overline{\boldsymbol{u}}_i^{*(p)}(k;N) \in \arg \mathcal{P}_i^N$, (see Equation (9.12))

   Calculate $\xi_i^p(k;N) = \boldsymbol{G}_i(N)\overline{\boldsymbol{u}}_i^{*(p)}(k;N) + \boldsymbol{g}_i(\mu(k);N)$

   if $\xi_i^p(k;N) \notin \mathcal{O}_\infty^i$

   (i) Increase N

   (ii) Extract $\overline{\boldsymbol{u}}_i^{p-1}(k;N)$ from $\boldsymbol{u}_i^{p-1}(k), \ \forall\, i \in \mathbb{I}_M$

   (iii) Goto step 4.

end (if)

Calculate $\overline{\boldsymbol{x}}_i^{*(p)} \leftarrow \overline{\boldsymbol{x}}_i(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_i^{*(p)}, \ldots, \boldsymbol{u}_M^{p-1}; x_i(k))$

end (do)

(b) **for** each $i \in \mathbb{I}_M$

$\overline{\boldsymbol{u}}_i^p(k;N) = w_i \overline{\boldsymbol{u}}_i^{*(p)}(k;N) + (1-w_i)\,\overline{\boldsymbol{u}}_i^{p-1}(k;N)$

$\overline{\boldsymbol{x}}_i^p(k;N) = w_i \overline{\boldsymbol{x}}_i^{*(p)}(k;N) + (1-w_i)\,\overline{\boldsymbol{x}}_i^{p-1}(k;N)$

$\rho_i = \|\overline{\boldsymbol{u}}_i^p(k;N) - \overline{\boldsymbol{u}}_i^{p-1}(k;N)\|$

Transmit $\overline{\boldsymbol{u}}_i^p$ to each subsystem $j \in \mathbb{I}_M,\ j \neq i$.

Construct $\boldsymbol{u}_i^p(k)$ (Remark 9.7).

end (for)

$p \leftarrow p+1$

end (while)

In the nominal case, Step 2 in Algorithm 9.4 has to be solved only once at $k = 0$. The shifted input trajectories (Equation (9.9)) are feasible for all $k > 0$. Also, the initialization in Step 2 is used for $\mu(k) \notin \mathcal{O}_\infty$. If $\mu(k) \in \mathcal{O}_\infty$, Remark 9.7 can be used directly to generate $\boldsymbol{u}_i^0(k),\ \forall\, i \in \mathbb{I}_M$.

Remark 9.7. For a collection of finite input trajectories $\overline{\boldsymbol{u}}_1(k;N), \ldots, \overline{\boldsymbol{u}}_M(k;N)$ such that $x_{\mathrm{cm}}(k+N|k) = [x_1(k+N|k)', x_2(k+N|k)', \ldots, x_M(k+N|k)']' \in \mathcal{O}_\infty$, the corresponding collection of infinite subsystem input trajectories $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M$ can be constructed using the positive invariance of $\mathcal{O}_\infty$ for the system $x_{\mathrm{cm}}(k+1) = (A_{\mathrm{cm}} + B_{\mathrm{cm}}K_{\mathrm{cm}})x_{\mathrm{cm}}(k)$ (Gilbert and Tan, 1991).

The infinite input trajectory for subsystem i, $\boldsymbol{u}_i(k)$, is obtained as

$$\boldsymbol{u}_i(k) = \left[\overline{\boldsymbol{u}}_i(k;N)', u_i(k+N|k)', u_i(k+N+1|k)', \ldots\ldots\right], \ \forall\, i \in \mathbb{I}_M \tag{9.25}$$

in which $u_i(k+N+j|k) = \mathcal{V}_i{}' \left[K_{\rm cm}(A_{\rm cm}+B_{\rm cm}K_{\rm cm})^j x_{\rm cm}(k+N|k)\right],\ 0 \leq j$. The matrix $\mathcal{V}_i = [0, \ldots, I_{m_i}, \ldots]'$ is defined such that $u_i = \mathcal{V}_i{}' u$. The corresponding infinite horizon cost function value is given by $\Phi(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M; \mu(k))$. By construction, the infinite horizon cost function value is also equal to $\Psi(\overline{\boldsymbol{u}}_1(k;N), \overline{\boldsymbol{u}}_2(k;N), \ldots, \overline{\boldsymbol{u}}_M(k;N); \mu(k))$. For implementation, a maximum allowable value of N *i.e.*, $N_{\max}$ is selected a priori and the infinite horizon input trajectories $\boldsymbol{u}_i, i \in \mathbb{I}_M$ are constructed up to this value $N_{\max}$.

Properties

For Algorithm 9.4, monotonicity of the cost function with iteration number and optimality at convergence can be established using arguments identical to those used in Lemmas 9.3 and 9.4 respectively.

Lemma 9.6. *Consider the DCLQR framework employing Algorithm 9.4. Let Assumptions 9.1 and 9.4 hold. For each subsystem $i \in \mathbb{I}_M$ there exists a finite $N \leftarrow N(\mu(k)) \in \mathbb{I}_+$ such that $\xi_i^p(k;N) \in \mathcal{O}_\infty^i$ (Step 4(a), Algorithm 9.4).*

Proof. From Lemma 9.2, the pair $(\mathcal{A}_i, \mathcal{B}_i)$, in which $\mathcal{A}_i = \mathrm{diag}(A_{1i}, \ldots, A_{Mi})$, $\mathcal{B}_i = [B_{1i}{}', \ldots, B_{Mi}{}']'$ is stabilizable. Assume $\nexists\, N \leftarrow N(\mu(k)) \in \mathbb{I}_+$, N finite such that

$$\xi_i^p(k;N) = \boldsymbol{G}_i(N)\overline{\boldsymbol{u}}_i^{*(p)}(k;N) + \boldsymbol{g}_i(\mu(k);N) \in \mathcal{O}_\infty^i.$$

There exists, therefore, $\kappa_i \in \mathbb{I}_+$ such that $\|\xi_i^p(k;N)\| > \kappa_i,\ \forall\, N \in \mathbb{I}_+$.

At time $k = 0$, let $\mu(0) \in \mathbb{X}_{N_0}, N_0 \in \mathbb{I}_+$ such that the initialization QP (Equation (9.8) is feasible for all $i \in \mathbb{I}_M$. The cost function

$$\Phi(\boldsymbol{u}_1^0, \ldots, \boldsymbol{u}_M^0; \mu(0)) = \Psi(\overline{\boldsymbol{u}}_1^0(0;N_0), \ldots, \overline{\boldsymbol{u}}_M(0;N_0); \mu(0)) = \Phi_0$$

is, therefore, finite. Since $\|\xi_i(0;N)\| > \delta,\ \forall\, N$, there exists $N_* \in \mathbb{I}_+$ such that

$$\Psi(\overline{\boldsymbol{u}}_1^0(0;N_*), \ldots, \overline{\boldsymbol{u}}_i^{*(1)}(0;N_*), \ldots, \overline{\boldsymbol{u}}_M(0;N_*); \mu(0)) > \Psi_0,\ \forall\, i \in \mathbb{I}_M.$$

The relationship above contradicts optimality of $\overline{\boldsymbol{u}}_i^{*(1)}(\cdot)$ for each $i \in \mathbb{I}_M$. Hence, there exists $N \in \mathbb{I}_+$, N finite such that $\xi_i^1(0;N) \in \mathcal{O}_\infty^i,\ \forall\, i \in \mathbb{I}_M$. Subsequently, $\Psi(\overline{\boldsymbol{u}}_1^0(0;N), \ldots, \overline{\boldsymbol{u}}_i^{*(1)}(0;N), \ldots, \overline{\boldsymbol{u}}_M(0;N); \mu(0))$ is finite. From Lemma 9.3, $\Psi(\overline{\boldsymbol{u}}_1^1(0;N), \ldots, \overline{\boldsymbol{u}}_M^1(0;N); \mu(0))$ is finite, and by convexity $\eta_i^1(0;N) = \boldsymbol{G}_i(N)\overline{\boldsymbol{u}}_i^p(k;N) + \boldsymbol{g}_i(\mu(k);N) \in \mathcal{O}_\infty^i$. Repeating the sequence of arguments presented above establishes the existence of an N for each iterate p such that $\eta_i^p(0;N) \in \mathcal{O}_\infty^i,\ \forall\, i \in \mathbb{I}_M$. At times $k > 0$, either the shifted input trajectory (nominal case) or the QP (Equation (9.8)) is used for initialization. In either case, it can be established that for $\mu(k) \in \mathbb{X}_{N_k}, N_k \in \mathbb{I}_+$, $\Phi(\boldsymbol{u}_1^0, \ldots, \boldsymbol{u}_M^0; \mu(k)) = \Psi(\overline{\boldsymbol{u}}_1^0(k;N_k), \ldots, \overline{\boldsymbol{u}}_M^0(k;N_k); \mu(k))$ is finite. The remainder of the proof follows the arguments presented above. □

9.6.4 Proof for Lemma 9.5

Proof. Suppose $x_{\text{cm}}(k) \notin \mathcal{O}_\infty$ for all $k \in \mathbb{I}_+$. Because $0 \in \text{int}(\mathcal{O}_\infty)$, there exists some $\kappa \in \mathbb{I}_+$ such that $\sum_i \|x_i(k)\|^2 > \kappa$ for all k. From Equation (9.16),

$$\begin{aligned}
\Delta\Phi(k+1) &= \Phi(\boldsymbol{u}_1^{p(k+1)}, \ldots, \boldsymbol{u}_M^{p(k+1)}; \mu(k+1)) - \Phi(\boldsymbol{u}_1^{p(k)}, \ldots, \boldsymbol{u}_M^{p(k)}; \mu(k)) \\
&= J^{p(k+1)}(\mu(k+1)) - J^{p(k)}(\mu(k)) \\
&\leq -\sum_{i=1}^{M} w_i L_i(x_i(k), u_i^{p(k)}(k)) \\
&\leq -\sum_{i=1}^{M} w_i L_i(x_i(k), 0) \\
&\leq -\rho, \quad \forall\, k \in \mathbb{I}_+
\end{aligned} \tag{9.26}$$

in which $\rho = \kappa \min_{\,i \in \mathbb{I}_M} w_i \lambda_{\min}(Q_i) > 0$. Summing over times $k = 0, 1, 2, \ldots, T$ gives

$$\begin{aligned}
\Phi(\boldsymbol{u}_1^{p(T)}, \ldots, \boldsymbol{u}_M^{p(T)}; \mu(T)) &\leq \Phi(\boldsymbol{u}_1^{p(0)}, \ldots, \boldsymbol{u}_M^{p(0)}; \mu(0)) - T\rho \\
&\leq \Phi(\boldsymbol{u}_1^{0}, \ldots, \boldsymbol{u}_M^{0}; \mu(0)) - T\rho \text{ (using Lemma 9.3)}
\end{aligned}$$

Since $\Phi(\boldsymbol{u}_1^0, \ldots, \boldsymbol{u}_M^0; \mu(0))$ is finite (initialization), here exists $T_0 \in \mathbb{I}_+$ such that

$$\Phi(\boldsymbol{u}_1^0, \ldots, \boldsymbol{u}_M^0; \mu(0)) - T_0\rho = 0.$$

For $T > T_0$, we have $\Phi(\boldsymbol{u}_1^{p(T)}, \ldots, \boldsymbol{u}_M^{p(T)}; \mu(T)) < 0$, which contradicts $\Phi(\cdot) \geq 0$, thereby establishing the lemma. □

Chapter 10

Distributed MPC Strategies with Application to Power System Automatic Generation Control [1]

Most interconnected power systems rely on automatic generation control (AGC) for controlling system frequency and tie-line interchange (Wood and Wollenberg, 1996). These objectives are achieved by regulating the real power output of generators throughout the system. To cope with the expansive nature of power systems, a distributed control structure has been adopted for AGC. Also, various limits must be taken into account, including restrictions on the amount and rate of generator power deviation. AGC therefore provides a very relevant example for illustrating the performance of distributed MPC in a power system setting.

Flexible AC transmission system (FACTS) devices allow control of the real power flow over selected paths through a transmission network (Hingorani and Gyugyi, 2000). As trans-

[1] **Portions of this chapter appear in Venkat, Hiskens, Rawlings, and Wright (2006a) and in Venkat, Hiskens, Rawlings, and Wright (2006d).**

mission systems become more heavily loaded, such controllability offers economic benefits (Krogh and Kokotovic, 1984) . However FACTS controls must be coordinated with each other, and with other power system controls, including AGC. Distributed MPC offers an effective means of achieving such coordination, whilst alleviating the organizational and computational burden associated with centralized control.

This chapter is organized as follows. In Section 10.1, a brief description of the different modeling frameworks suitable for power networks is presented. In Section 10.2, a description of the different MPC based systemwide control frameworks is provided. A simple example that illustrates the unreliability of communication-based MPC is presented. An implementable algorithm for terminal penalty-based distributed MPC is described in Section 10.3. Properties of this distributed MPC algorithm and closed-loop properties of the resulting distributed controller are established subsequently. Three examples are presented to assess the performance of the terminal penalty-based distributed MPC framework. Two useful extensions of the proposed distributed MPC framework are described in Section 10.6. An algorithm for terminal control-based distributed MPC is described in Section 10.6.3. Two examples are presented to illustrate the efficacy of the terminal control-based distributed MPC framework. Conclusions of this study are provided in Section 10.7.

10.1 Models

Distributed MPC relies on decomposing the overall system model into appropriate subsystem models. A system comprised of M interconnected subsystems will be used to establish these concepts.

Centralized model. The overall system model is represented as a discrete LTI model of the form

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k)$$

in which k denotes discrete time and

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ A_{i1} & A_{i2} & \dots & A_{iM} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \dots & A_{MM} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ B_{i1} & B_{i2} & \dots & B_{iM} \\ \vdots & \vdots & \ddots & \vdots \\ B_{M1} & B_{M2} & \dots & B_{MM} \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & 0 & \dots & 0 \\ 0 & C_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & C_{MM} \end{bmatrix} \qquad u = \begin{bmatrix} u_1{}' & u_2{}' & \dots & u_M{}' \end{bmatrix}' \in \mathbb{R}^m$$

$$x = \begin{bmatrix} x_1{}' & x_2{}' & \dots & x_M{}' \end{bmatrix}' \in \mathbb{R}^n \quad y = \begin{bmatrix} y_1{}' & y_2{}' & \dots & y_M{}' \end{bmatrix}' \in \mathbb{R}^z.$$

For each subsystem $i \in \mathbb{I}_M$, (u_i, x_i, y_i) represents the subsystem input, state and output respectively. The centralized model pair (A, B) is assumed to be stabilizable and (A, C) is detectable [2].

[2] In the applications considered here, local measurements are typically a subset of subsystem states. The structure selected for the C matrix reflects this observation. A general C matrix may be used, but impacts possible choices for distributed estimation techniques (Venkat, Hiskens, Rawlings, and Wright, 2006b).

Decentralized model. In the decentralized modeling framework, it is assumed that the interaction between the subsystems is negligible. Subsequently, the effect of the external subsystems on the local subsystem is ignored in this modeling framework. The decentralized model for subsystem $i \in \mathbb{I}_M$ is

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k)$$

$$y_i(k) = C_{ii}x_i(k)$$

Partitioned model (PM). The PM for subsystem i combines the effect of the local subsystem variables and the effect of the states and inputs of the interconnected subsystems. The PM for subsystem i is obtained by considering the relevant partition of the centralized model and can be explicitly written as

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{j \neq i}(A_{ij}x_j(k) + B_{ij}u_j(k)) \tag{10.1a}$$

$$y_i(k) = C_{ii}x_i(k) \tag{10.1b}$$

10.2 MPC frameworks for systemwide control

The set of admissible controls for subsystem $i, \Omega_i \subseteq \mathbb{R}^{m_i}$ is assumed to be a nonempty, compact, convex set with the origin in its interior. The set of admissible controls for the whole plant Ω is defined to be the Cartesian product of the admissible control sets $\Omega_i, \forall\, i \in \mathbb{I}_M$. The *stage cost* at stage $t \geq k$ along the prediction horizon and the *cost function* $\phi_i(\cdot)$ for subsystem $i \in \mathbb{I}_M$ are defined in Equations (4.3) and (4.5) with each x_i now denoting the states in the PM

(Equation (10.1) for subsystem $i \in \mathbb{I}_M$. For any system, the constrained stabilizable set (also termed Null controllable domain) $\mathcal{X}$ is the set of all initial states $x \subseteq \mathbb{R}^n$ that can be steered to the origin by applying a sequence of admissible controls (see (Sznaier and Damborg, 1990, Definition 2)). It is assumed throughout that the initial system state vector $x(0) \in \mathcal{X}$, in which $\mathcal{X}$ denotes the constrained stabilizable set for the overall system. A feasible solution to the corresponding optimization problem, therefore, exists.

Four MPC based systemwide control frameworks are described below. The difference between the optimization problems described here and those in Section 4.3, p. 32 is in the modeling framework used (see Section 10.1). In each MPC framework, the controller is defined by implementing the first input in the solution to the corresponding optimization problem.

Centralized MPC. In the centralized MPC framework, the MPC for the overall system solves the following optimization problem

$$\min_{\boldsymbol{x},\boldsymbol{u}} \quad \phi\left(\boldsymbol{x},\boldsymbol{u};x(k)\right) = \sum_i w_i \phi_i\left(\boldsymbol{x}_i,\boldsymbol{u}_i;x_i(k)\right)$$

subject to

$$x(l+1|k) = Ax(l|k) + Bu(l|k), \;\; k \leq l$$

$$u_i(l|k) \in \Omega_i, \;\; k \leq l, \quad i \in \mathbb{I}_M$$

where $w_i > 0$,$\sum w_i = 1$.

For any system, centralized MPC achieves the best attainable performance (Pareto optimal) as the effect of interconnections among subsystems are accounted for exactly. Furthermore, any conflicts among controller objectives are resolved optimally.

Decentralized MPC. In the decentralized MPC framework, the following optimization problem is solved by each controller

$$\min_{\boldsymbol{x}_i, \boldsymbol{u}_i} \quad \phi_i \left(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i(k) \right)$$

subject to

$$x_i(l+1|k) = A_{ii} x_i(l|k) + B_{ii} u_i(l|k), \;\; k \leq l$$

$$u_i(l|k) \in \Omega_i, \;\; k \leq l$$

Each decentralized MPC solves an optimization problem to minimize its (local) cost function. The effects of the interconnected subsystems are assumed to be negligible and are ignored. In many situations, however, the above assumption is not valid and leads to reduced control performance.

Communication-based MPC. For communication-based MPC [3], the optimal state-input trajectory $(\boldsymbol{x}_i^p, \boldsymbol{u}_i^p)$ for subsystem $i,\ i \in \mathbb{I}_M$ at iterate p is obtained as the solution to the optimiza-

[3] Similar strategies have been proposed by Camponogara et al. (2002); Jia and Krogh (2001)

tion problem

$$\min_{\boldsymbol{x}_i, \boldsymbol{u}_i} \quad \phi_i\left(\boldsymbol{x}_i, \boldsymbol{u}_i; x_i(k)\right)$$

subject to

$$x_i(l+1|k) = A_{ii}x_i(l|k) + B_{ii}u_i(l|k) + \sum_{j \neq i}[A_{ij}x_j^{p-1}(l|k) + B_{ij}u_j^{p-1}(l|k)],\ k \leq l$$

$$u_i(l|k) \in \Omega_i,\ k \leq l$$

As described in Section 4.3, each communication-based MPC utilizes the objective function for that subsystem only. For each subsystem i at iteration p, only that subsystem input sequence $\boldsymbol{u}_i$ is optimized and updated. The other subsystems' inputs remain at $\boldsymbol{u}_j^{p-1}$, $\forall\, j \in \mathbb{I}_M, j \neq i$. If the communication-based iterates converge, then at convergence, the Nash equilibrium (NE) is achieved.

Instability under communication-based MPC. Figure 10.1 illustrates nonconvergence of communication-based MPC for a two subsystem case. For initial values of inputs at the origin and in the absence of input constraints, the sequence of communication-based iterates diverges to infinity. For a compact feasible region (the box in Figure 10.1), the sequence of communication-based iterates is trapped at the boundary of the feasible region (Point 5). For this system, the NE is at point n.

Feasible cooperation-based MPC (FC-MPC). To arrive at a reliable distributed MPC framework, we need to ensure that the subsystems' MPCs cooperate, rather than compete, with each other in achieving systemwide objectives. The local controller objective $\phi_i(\cdot)$ is replaced by an

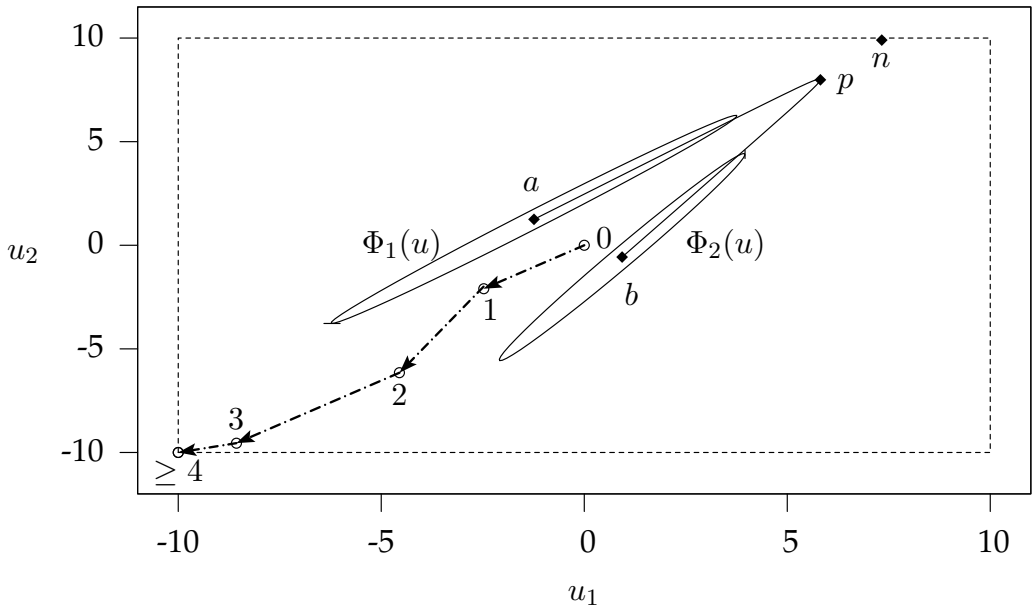

Figure 10.1: A Nash equilibrium exists. Communication-based iterates do not converge to the Nash equilibrium however.

objective that measures the systemwide impact of local control actions. The simplest choice for such an objective is a strict convex combination of the controller objectives $i.e.,\ \phi(\cdot) = \sum_i w_i \phi_i(\cdot),\ w_i > 0,\ \sum_i w_i = 1$.

For notational convenience, we drop the k dependence of $\overline{\boldsymbol{x}}_i(k), \overline{\boldsymbol{u}}_i(k), i \in \mathbb{I}_M$. It is shown in Appendix 10.8.1 that each $\overline{\boldsymbol{x}}_i$ can be expressed as

$$\overline{\boldsymbol{x}}_i = E_{ii}\overline{\boldsymbol{u}}_i + f_{ii}x_i(k) + \sum_{j \neq i}[E_{ij}\overline{\boldsymbol{u}}_j + f_{ij}x_j(k)]. \tag{10.4}$$

We consider open-loop stable systems here. Extensions of the distributed MPC methodology to handle large, open-loop unstable systems are described in Sections 10.6.2 and 10.6.3.

For open-loop stable systems, the FC-MPC optimization problem for subsystem i, denoted $\mathcal{F}_i$, is defined as

$$\mathcal{F}_i \triangleq \min_{\boldsymbol{u}_i} \quad \sum_{r=1}^{M} w_r \Phi_r \left(\boldsymbol{u}_1^{p-1}, \ldots, \boldsymbol{u}_{i-1}^{p-1}, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}^{p-1}, \ldots, \boldsymbol{u}_M^{p-1}; x_r(k) \right) \tag{10.5a}$$

subject to

$$u_i(t|k) \in \Omega_i, \quad k \leq t \leq k+N-1 \tag{10.5b}$$

$$u_i(t|k) = 0, \quad k+N \leq t \tag{10.5c}$$

The infinite horizon input trajectory $\boldsymbol{u}_i$ is obtained by augmenting $\overline{\boldsymbol{u}}_i$ with the input sequence $u_i(t|k) = 0, k+N \leq t$. The infinite horizon state trajectory $\boldsymbol{x}_i$ is derived from $\overline{\boldsymbol{x}}_i$ by propagating the terminal state $x_i(k+N|k)$ using Equation (10.1) and $u_i(t|k) = 0, k+N \leq t, \forall\, i \in \mathbb{I}_M$. The cost function $\Phi_i(\cdot)$ is obtained by eliminating the state trajectory $\boldsymbol{x}_i$ from Equation (4.5) using Equation (10.4) and the input, state parameterization described above. The solution to the optimization problem $\mathcal{F}_i$ is denoted by $\boldsymbol{u}_i^{*(p)}$. By definition,

$$\boldsymbol{u}_i^{*(p)} = [u_i^{*(p)}(k|k)', u_i^{*(p)}(k+1|k)', \ldots\ldots]' \text{ and}$$

$$\overline{\boldsymbol{u}}_i^{*(p)} = [u_i^{*(p)}(k|k)', u_i^{*(p)}(k+1|k)', \ldots, u_i^{*(p)}(k+N-1|k)']'$$

10.3 Terminal penalty FC-MPC

10.3.1 Optimization

For $\phi_i(\cdot)$ defined in Equation (4.5), the FC-MPC optimization problem (Equation (10.5)), for each subsystem $i \in \mathbb{I}_M$, can be written as

$$\mathcal{F}_i \triangleq \min_{\overline{\boldsymbol{u}}_i} \quad \frac{1}{2}\overline{\boldsymbol{u}}_i{}'\mathfrak{R}_i\overline{\boldsymbol{u}}_i + \left(\boldsymbol{r}_i(x(k)) + \sum_{j\neq i}\mathcal{H}_{ij}\overline{\boldsymbol{u}}_j^{p-1}\right)'\overline{\boldsymbol{u}}_i \tag{10.6a}$$

subject to

$$u_i(t|k) \in \Omega_i, \;\; k \leq t \leq k+N-1 \tag{10.6b}$$

in which

$$\begin{aligned}
\mathfrak{R}_i &= \mathbb{R}_i + \sum_{j=1}^{M} E_{ji}{}'\mathbb{Q}_j E_{ji} + \sum_{j=1}^{M} E_{ji}{}' \sum_{l\neq j} \mathbb{T}_{jl} E_{li} \\
\mathbb{Q}_i &= \text{diag}\left(w_i Q_i(1), \ldots, w_i Q_i(N-1), P_{ii}\right) \\
\mathbb{T}_{ij} &= \text{diag}\left(0, \ldots, 0, P_{ij}\right) \\
\mathbb{R}_i &= \text{diag}\left(w_i R_i(0), w_i R_i(1), \ldots, w_i R_i(N-1)\right) \\
\boldsymbol{r}_i(x(k)) &= \sum_{j=1}^{M} E_{ji}{}'\,\mathbb{Q}_j\,\boldsymbol{g}_j(x(k)) + \sum_{j=1}^{M} E_{ji}{}' \sum_{l\neq j} \mathbb{T}_{jl}\boldsymbol{g}_l(x(k))
\end{aligned}$$

$$\mathcal{H}_{ij} = \sum_{l=1}^{M} E_{li}{}'\mathbb{Q}_l E_{lj} + \sum_{l=1}^{M} E_{li}{}' \sum_{s\neq l} \mathbb{T}_{ls} E_{sj} \qquad\qquad \boldsymbol{g}_i(x(k)) = \sum_{j=1}^{M} f_{ij} x_j(k)$$

and

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & \dots & P_{1M} \\ P_{21} & P_{22} & \dots & \dots & P_{2M} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ P_{M1} & P_{M2} & \dots & \dots & P_{MM} \end{bmatrix} \tag{10.7}$$

is a suitable terminal penalty matrix. Restricting attention (for now) to open-loop stable systems simplifies the choice of P. For each $i \in \mathbb{I}_M$, let $Q_i(0) = Q_i(1) = \ldots = Q_i(N-1) = Q_i$. The terminal penalty P can be obtained as the solution to the centralized Lyapunov equation

$$A' \, P \, A - P = -\mathcal{Q} \tag{10.8}$$

in which $\mathcal{Q} = \text{diag}(w_1 Q_1, w_2 Q_2, \ldots, w_M Q_M)$. The centralized Lyapunov equation (Equation (10.8)) is solved offline. The solution to Equation (10.8), P, has to be recomputed if the subsystems' models and/or cost functions are altered.

10.3.2 Algorithm and properties

Algorithm 4.1 (p. 43) solving the optimization problem of Equation (10.6) is used for FC-MPC. The state trajectory for subsystem i generated by the input trajectories $\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M$ and initial state z is represented as $\overline{\boldsymbol{x}}_i(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \ldots, \overline{\boldsymbol{u}}_M; z)$. At each iterate p in Algorithm 4.1, the state trajectory for subsystem $i \in \mathbb{I}_M$ can be calculated as $\overline{\boldsymbol{x}}_i^p(\overline{\boldsymbol{u}}_1^p, \overline{\boldsymbol{u}}_2^p, \ldots, \overline{\boldsymbol{u}}_M^p; x(k))$. The infinite horizon input and state trajectories $(\boldsymbol{x}_i^p, \boldsymbol{u}_i^p)$ can be obtained following the discussion in

Section 10.2. Denote the cooperation-based cost function after p iterates by

$$\Phi(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; x(k)) = \sum_{r=1}^{M} w_r \Phi_r \left(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; x_r(k)\right).$$

Properties for Algorithm 4.1 established in Section 4.5 apply here (with each $\mu(k)$ replaced by $x(k)$).

10.3.3 Distributed MPC control law

At time k, let the FC-MPC algorithm (Algorithm 4.1) be terminated after $p(k)$ iterates, with

$$\boldsymbol{u}_i^{p(k)}(x(k)) = \left[u_i^{p(k)}(x(k),0)', u_i^{p(k)}(x(k),1)', \ldots\right]', \quad \forall\, i \in \mathbb{I}_M \tag{10.9}$$

representing the solution to Algorithm 4.1 after $p(k)$ cooperation-based iterates. The distributed MPC control law is obtained through a receding horizon implementation of optimal control whereby the input applied to subsystem i is $u_i^{p(k)}(x(k),0)$.

10.3.4 Feasibility of FC-MPC optimizations

Since $x(0) \in \mathcal{X}$, there exists a set of feasible, open-loop input trajectories $(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M)$ such that $x_i(k) \to 0,\ \forall\, i \in \mathbb{I}_M$ and k sufficiently large. Convexity of $\Omega_i, \forall\, i \in \mathbb{I}_M$ and Algorithm 4.1 guarantee that given a feasible input sequence at time $k = 0$, a feasible input sequence exists for all future times. One trivial choice for a feasible input sequence at $k = 0$ is $u_i(k+l|k) = 0,\ l \geq 0,\ \forall\, i \in \mathbb{I}_M$. This choice follows from our assumption that each Ω_i is nonempty and $0 \in \text{int}(\Omega_i)$. Existence of a feasible input sequence for each subsystem i at $k = 0$ ensures that

the FC-MPC optimization problem (Equations (10.5) and (10.6)) has a solution for each $i \in \mathbb{I}_M$ and all $k \geq 0$.

10.3.5 Initialization

At discrete time $k+1$, define $\forall\ i \in \mathbb{I}_M$

$$\boldsymbol{u}_i^0\,(\cdot)\,' = \left[u_i^{p(k)}(x(k),1)', u_i^{p(k)}(x(k),2)', \ldots, u_i^{p(k)}(x(k),N-1)', 0, 0, \ldots\right] \tag{10.10}$$

It follows that $\boldsymbol{u}_1^0(k+1), \boldsymbol{u}_2^0(k+1), \ldots, \boldsymbol{u}_M^0(k+1)$ constitute feasible subsystem input trajectories with an associated cost function $\Phi\left(\boldsymbol{u}_1^0(k+1), \boldsymbol{u}_2^0(k+1), \ldots, \boldsymbol{u}_M^0(k+1); x(k+1)\right)$.

10.3.6 Nominal closed-loop stability

Given the set of initial subsystem states $x_i(0),\ \forall\ i\ \in\ \mathbb{I}_M$. Define $\widetilde{J}_N(x(0))$ to be the value of the cooperation-based cost function with the set of zero input trajectories $u_i(k+j|k)\ =\ 0,\ j\ \geq\ 0, \forall\ i\ \in\ \mathbb{I}_M$. At time k, let $J_N^0(x(k))$ represent the value of the cooperation-based cost function with the input trajectory initialization described in Equation (10.10). For notational convenience we drop the function dependence of the generated state trajectories and write $\boldsymbol{x}_i \equiv \boldsymbol{x}_i(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M; z),\ \forall\ i\ \in\ \mathbb{I}_M$. The value of the cooperation-based cost function after $p(k)$ iterates is denoted by $J_N^{p(k)}(x(k))$. Thus,

$$J_N^{p(k)}(x(k)) = \sum_{i=1}^{M} w_i \phi_i \left(\boldsymbol{x}_i^{p(k)}, \boldsymbol{u}_i^{p(k)}; x(k)\right) \tag{10.11a}$$

$$= \sum_{i=1}^{M} w_i \sum_{j=0}^{\infty} L_i \left(x_i^{p(k)}(k+j|k), u_i^{p(k)}(k+j|k)\right) \tag{10.11b}$$

At $k = 0$, we have using Lemma 4.4, p. 44 (with $\mu(k)$ replaced by $x(k)$) that

$$J_N^{p(0)}(x(0)) \leq J_N^0(x(0)) = \widetilde{J}_N(x(0)).$$

It follows from Equation (10.10) and Lemma 4.4 that

$$0 \leq J_N^{p(k)}(x(k)) \leq J_N^0(x(k)) = J_N^{p(k-1)}(x(k-1)) - \sum_{i=1}^{M} w_i L_i(x_i(k-1), u_i^{p(k-1)}(k-1)),\ \forall\, k > 0 \tag{10.12}$$

Using the above relationship recursively from time k to time 0 gives

$$J_N^{p(k)}(x(k)) \leq \widetilde{J}_N(x(0)) - \sum_{j=0}^{k-1} \sum_{i=1}^{M} w_i L_i(x_i(j), u_i^{p(j)}(j)) \leq \widetilde{J}_N(x(0)), \tag{10.13}$$

From Equation (10.11), we have $\frac{1}{2}\lambda_{\min}(\mathcal{Q})\|x(k)\|^2 \leq J_N^{p(k)}(x(k))$. Using Equation (10.13), gives $J_N^{p(k)}(x(k)) \leq \widetilde{J}_N(x(0)) = \frac{1}{2}x(0)'Px(0) \leq \frac{1}{2}\lambda_{\max}(P)\|x(0)\|^2$. From the previous two cost relationships, we obtain $\|x(k)\| \leq \sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(\mathcal{Q})}}\|x(0)\|$, which shows that the closed-loop system is Lyapunov stable (Vidyasagar, 1993, p. 265). In fact, using the cost convergence relationship (Equation (10.12)) the closed-loop system is also attractive, which proves asymptotic stability under the distributed MPC control law.

Lemmas 4.4 and 4.5 can be used to establish the following (stronger) exponential closed-loop stability result.

Theorem 10.1. *Given Algorithm 4.1 using the distributed MPC optimization problem of Equation* (10.6) *with* $N \geq 1$. *In Algorithm 4.1, let* $0 < p_{\max}(k) \leq p^* < \infty, \forall\, k \geq 0$. *If A is stable, P is obtained*

from Equation (10.8)*, and*

$$Q_i(0) = Q_i(1) = \ldots = Q_i(N-1) = Q_i > 0$$

$$R_i(0) = R_i(1) = \cdots = R_i(N-1) = R_i > 0$$

$$\forall\, i \in \mathbb{I}_M$$

then the origin is an exponentially stable equilibrium for the closed-loop system

$$x(k+1) = Ax(k) + Bu(x(k))$$

in which

$$u(x(k)) = \left[u_1^{p(k)}(x(k),0)', \ldots, u_M^{p(k)}(x(k),0)'\right]'$$

for all $x(k) \in \mathbb{R}^n$ *and all* $p(k) = 1, 2, \ldots, p_{\max}(k)$*.*

A proof is given in Appendix 10.8.1.

Remark 10.1. If $(A, \mathcal{Q}^{\frac{1}{2}})$ is detectable, then the weaker requirement $Q_i \geq 0$, $R_i > 0$, $\forall\, i \in \mathbb{I}_M$ is sufficient to ensure exponential stability of the closed-loop system under the distributed MPC control law.

10.4 Power system terminology and control area model

For the purposes of AGC, power systems are decomposed into control areas, with tie-lines providing interconnections between areas (Wood and Wollenberg, 1996). Each area typically consists of numerous generators and loads. It is common, though, for all generators in an area

to be lumped as a single equivalent generator, and likewise for loads. That model is adopted in all subsequent examples. Some basic power systems terminology is provided in Table 10.1. The notation Δ is used to indicate a deviation from steady state. For example, $\Delta\omega$ represents a deviation in the angular frequency from its nominal operating value (60 Hz.).

Table 10.1: Basic power systems terminology.

ω	:	angular frequency of rotating mass
δ	:	phase angle of rotating mass
M^a	:	Angular momentum
D	:	$\frac{\text{percent change in load}}{\text{percent change in frequency}}$
P_{mech}	:	mechanical power
P_{L}	:	nonfrequency sensitive load
T_{CH}	:	charging time constant (prime mover)
P_{v}	:	steam valve position
P_{ref}	:	load reference setpoint
R^f	:	$\frac{\text{percent change in frequency}}{\text{percent change in unit output}}$
T_{G}	:	governor time constant
P_{tie}^{ij}	:	tie-line power flow between areas i and j
T_{ij}	:	tie-line (between areas i and j) stiffness coefficient
K_{ij}	:	FACTS device coefficient (regulating impedance between areas i and j)

Consider any control area $i \in \mathbb{I}_M$, interconnected to control area $j, j \neq i$ through a tie line. A simplified model for such a control area i is given in (10.14).

Area i

$$\frac{d\Delta\omega_i}{dt} + \frac{1}{M_i^a} D_i \Delta\omega_i + \frac{1}{M_i^a} \Delta P_{\text{tie}}^{ij} - \frac{1}{M_i^a} \Delta P_{\text{mech}_i} = -\frac{1}{M_i^a} \Delta P_{\text{L}_i} \tag{10.14a}$$

$$\frac{d\Delta P_{\text{mech}_i}}{dt} + \frac{1}{T_{\text{CH}_i}} \Delta P_{\text{mech}_i} - \frac{1}{T_{\text{CH}_i}} \Delta P_{\text{v}_i} = 0 \tag{10.14b}$$

$$\frac{d\Delta P_{\text{v}_i}}{dt} + \frac{1}{T_{\text{G}_i}} \Delta P_{\text{v}_i} - \frac{1}{T_{\text{G}_i}} \Delta P_{\text{ref}_i} + \frac{1}{R_i^f T_{\text{G}_i}} \Delta\omega_i = 0 \tag{10.14c}$$

tie-line power flow between areas i and j

$$\frac{d\Delta P_{\text{tie}}^{ij}}{dt} = T_{ij}\left(\Delta\omega_i - \Delta\omega_j\right) \tag{10.14d}$$

$$\Delta P_{\text{tie}}^{ji} = -\Delta P_{\text{tie}}^{ij} \tag{10.14e}$$

10.5 Examples

Performance comparison The cumulative stage cost Λ is used as an index for comparing the performance of different MPC frameworks. Define

$$\Lambda = \frac{1}{t}\sum_{k=0}^{t-1}\sum_{i=1}^{M} L_i\left(x_i(k), u_i(k)\right). \tag{10.15}$$

where t is the simulation horizon.

10.5.1 Two area power system network

An example with two control areas interconnected through a tie line is considered. The controller parameters and constraints are given in Table 10.2. A control horizon $N = 15$ is used for each MPC. The controlled variable (CV) for area 1 is the frequency deviation $\Delta\omega_1$ and the CV for area 2 is the deviation in the tie-line power flow between the two control areas $\Delta P_{\text{tie}}^{12}$. From the control area model (Equation (10.14)), if $\Delta\omega_1 \to 0$ and $\Delta P_{\text{tie}}^{12} \to 0$ then $\Delta\omega_2 \to 0$.

For a 25% load increase in area 2, the load disturbance rejection performance of the FC-MPC formulation is evaluated and compared against the performance of centralized MPC (cent-MPC), communication-based MPC (comm-MPC) and standard automatic generation control (AGC) with anti-reset windup. The load reference setpoint in each area is constrained

between ± 0.3. In practice, a large load change, such as the one considered above, would result in curtailment of AGC and initiation of emergency control measures such as load shedding. The purpose of this exaggerated load disturbance is to illustrate the influence of input constraints on the different control frameworks.

The relative performance of standard AGC, cent-MPC and FC-MPC (terminated after 1 iterate) rejecting the load disturbance in area 2 is depicted in Figure 10.2. The closed-loop trajectory of the FC-MPC controller, obtained by terminating Algorithm 4.1 after 1 iterate, is almost indistinguishable from the closed-loop trajectory of cent-MPC. Standard AGC performs nearly as well as cent-MPC and FC-MPC in driving the local frequency changes to zero. Under standard AGC, however, the system takes in excess of 400 seconds to drive the deviational tie-line power flow to zero. With the cent-MPC or the FC-MPC framework, the tie-line power flow disturbance is rejected in about 100 seconds. A closed-loop performance comparison of the different control frameworks is given in Table 10.3. The comm-MPC framework stabilizes the system but incurs a control cost that is nearly 18% greater than that incurred by FC-MPC (1 iterate). If 5 iterates per sampling interval are allowed, the performance of FC-MPC is almost identical to that of cent-MPC.

Notice from Figure 10.2 that the initial response of AGC is to increase generation in both areas. This causes a large deviation in the tie-line power flow. On the other hand under comm-MPC and FC-MPC, MPC-1 initially reduces area 1 generation and MPC-2 orders a large increase in area 2 generation (the area where the load disturbance occurred). This strategy enables a much more rapid restoration of tie-line power flow.

Table 10.2: Model parameters and input constraints for the two area power network model (Example 10.5.1).

D_1 =	2	D_2 =	2.75
R_1^f =	0.03	R_2^f =	0.07
M_1^a =	3.5	M_2^a =	4.0
T_{CH_1} =	50	T_{CH_2} =	10
T_{G_1} =	40	T_{G_2} =	25
Q_1 =	$\mathrm{diag}(1000, 0, 0)$	Q_2 =	$\mathrm{diag}(0, 0, 0, 1000)$
R_1 =	1	R_2 =	1
T_{12} =	7.54	Δ_{samp} =	0.1 sec

$-0.3 \leq \Delta P_{\mathrm{ref}_1} \leq 0.3$
$-0.3 \leq \Delta P_{\mathrm{ref}_2} \leq 0.3$

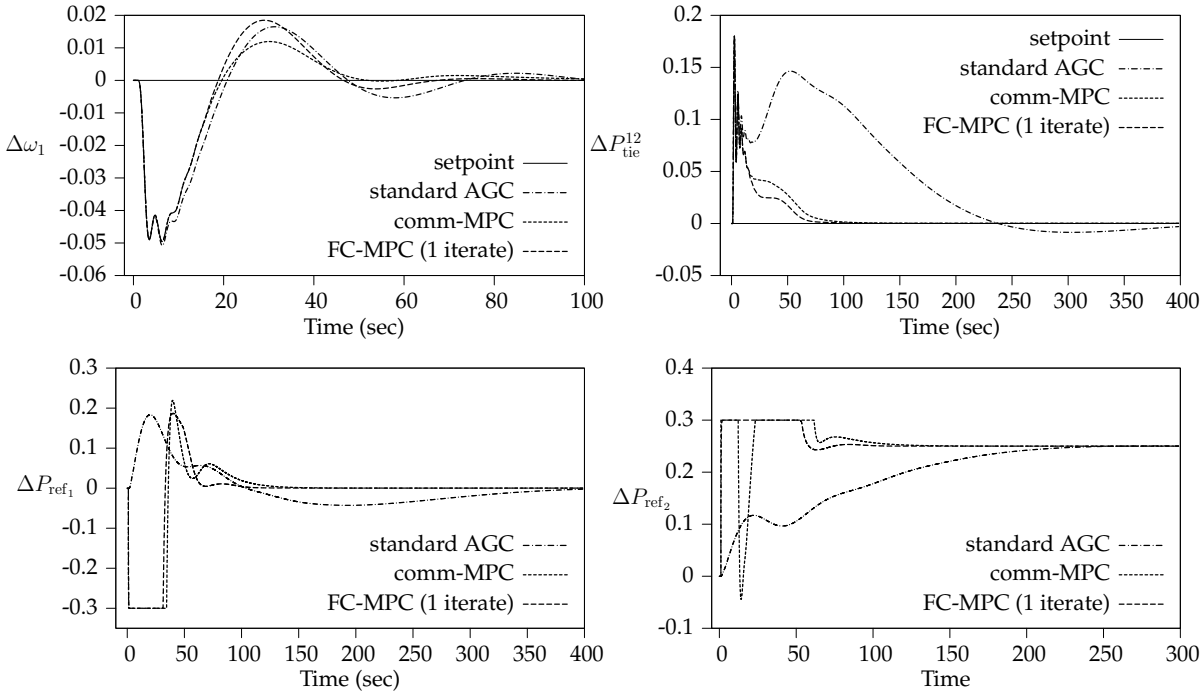

Figure 10.2: Performance of different control frameworks rejecting a load disturbance in area 2. Change in frequency $\Delta\omega_1$, tie-line power flow $\Delta P_{\mathrm{tie}}^{12}$ and load reference setpoints $\Delta P_{\mathrm{ref}_1}, \Delta P_{\mathrm{ref}_2}$.

10.5.2 Four area power system network

We revisit the four area power system described in Chapter 3, Section 3.2. The relative performance of cent-MPC, comm-MPC and FC-MPC is analyzed for a 25% load increase in area 2 and

Table 10.3: Performance of different control formulations w.r.t. cent-MPC, $\Delta\Lambda\% = \frac{\Lambda_{\text{config}} - \Lambda_{\text{cent}}}{\Lambda_{\text{cent}}} \times 100$.

	Λ	$\Delta\Lambda\%$
standard AGC	39.26	189
comm-MPC	15.82	18.26
FC-MPC (1 iterate)	13.42	0.26
FC-MPC (5 iterates)	~ 13.38	~ 0
cent-MPC	13.38	–

a simultaneous 25% load drop in area 3. This load disturbance occurs at 5 sec. For each MPC, we choose a prediction horizon of $N = 20$. In the comm-MPC and FC-MPC formulations, the load reference setpoint (ΔP_{ref_i}) in each area is manipulated to reject the load disturbance and drive the change in local frequencies ($\Delta\omega_i$) and tie-line power flows ($\Delta P_{\text{tie}}^{ij}$) to zero. In the cent-MPC framework, a single MPC manipulates all four ΔP_{ref_i}. The load reference setpoint for each area is constrained between ± 0.5.

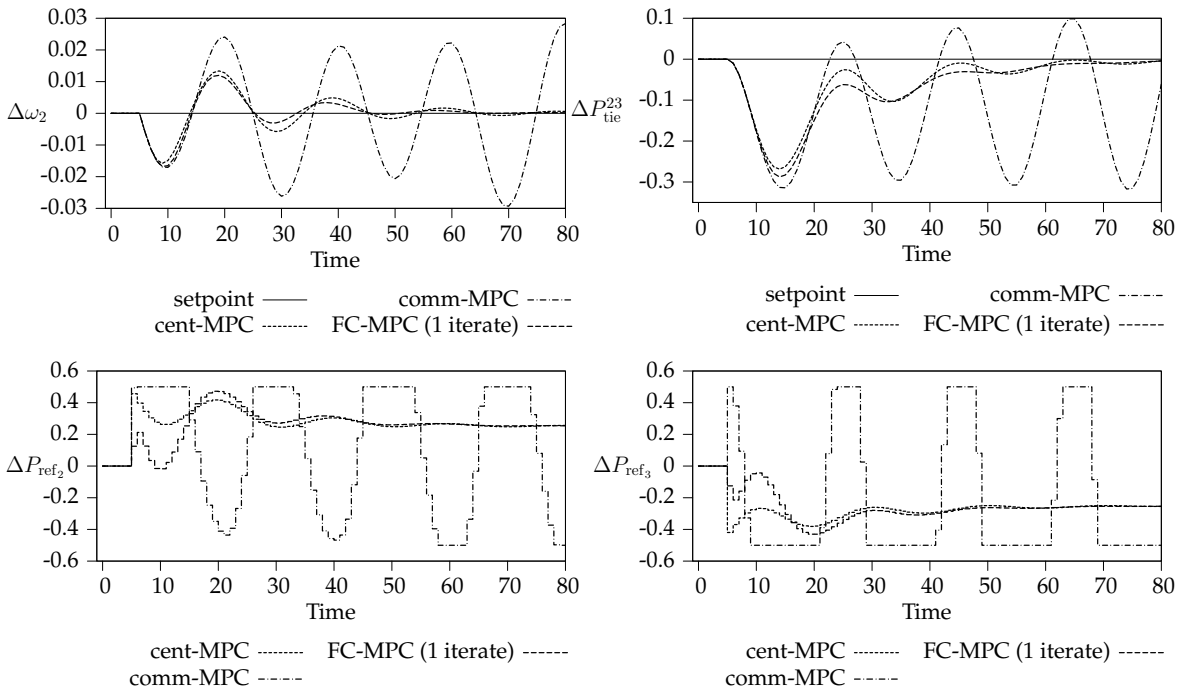

Figure 10.3: Performance of different control frameworks rejecting a load disturbance in areas 2 and 3. Change in frequency $\Delta\omega_2$, tie-line power flow $\Delta P_{\text{tie}}^{23}$ and load reference setpoints $\Delta P_{\text{ref}_2}, \Delta P_{\text{ref}_3}$.

Table 10.4: Performance of different MPC frameworks relative to cent-MPC, $\Delta\Lambda\% = \frac{\Lambda_{\text{config}} - \Lambda_{\text{cent}}}{\Lambda_{\text{cent}}} \times 100$.

	$\Lambda \times 10^{-2}$	$\Delta\Lambda\%$
cent-MPC	7.6	–
comm-MPC	$\uparrow \infty$	$\uparrow \infty$
FC-MPC (1 iterate)	9.6	26
FC-MPC (5 iterates)	7.87	3.7

The performance of cent-MPC, comm-MPC and FC-MPC (1 iterate) are shown in Figure 10.3. Only $\Delta\omega_2$ and $\Delta P_{\text{tie}}^{23}$ are shown as the frequency and tie-line power flow deviations in the other areas display similar qualitative behavior. Likewise, only ΔP_{ref_2} and ΔP_{ref_3} are shown as other load reference setpoints behave similarly. The control costs are given in Table 10.4. As seen in Section 3.2, the power system network is unstable under comm-MPC. The closed-loop performance of the FC-MPC formulation, terminated after just 1 iterate, is within 26% of cent-MPC performance. If the FC-MPC algorithm is terminated after 5 iterates, the performance of FC-MPC is within 4% of cent-MPC performance. By allowing the cooperation-based iterative process to converge, the closed-loop performance of FC-MPC can be driven to within any pre-specified tolerance of cent-MPC performance.

10.5.3 Two area power system with FACTS device

In this example, we revisit the two area network considered in Section 10.5.1. In this case though, a FACTS device is employed by area 1 to manipulate the effective impedance of the tie line and control power flow between the two interconnected control areas. The control area model follows from Equation (10.14). In order to incorporate the FACTS device, Equa-

tion (10.14a) in area 1 is replaced by

$$\frac{d\Delta\delta_{12}}{dt} = (\Delta\omega_1 - \Delta\omega_2)$$
$$\frac{d\Delta\omega_1}{dt} = -\frac{1}{M_1^a}D_1\Delta\omega_1 - \frac{1}{M_1^a}T_{12}\Delta\delta_{12} + \frac{1}{M_1^a}K_{12}\Delta X_{12} + \frac{1}{M_1^a}\Delta P_{\mathrm{mech}_1} - \frac{1}{M_1^a}\Delta P_{\mathrm{L}_1}$$

and in area 2 by

$$\frac{d\Delta\omega_2}{dt} = -\frac{1}{M_2^a}D_2\Delta\omega_2 + \frac{1}{M_2^a}T_{12}\Delta\delta_{12} - \frac{1}{M_2^a}K_{12}\Delta X_{12} + \frac{1}{M_2^a}\Delta P_{\mathrm{mech}_2} - \frac{1}{M_2^a}\Delta P_{\mathrm{L}_2}$$

where ΔX_{12} is the impedence deviation induced by the FACTS device. The tie-line power flow deviation becomes

$$\Delta P_{\mathrm{tie}}^{12} = -\Delta P_{\mathrm{tie}}^{21} = T_{12}\Delta\delta_{12} - K_{12}\Delta X_{12}$$

Notice that if $\Delta X_{12} = 0$, the model reverts to Equation (10.14). Controller parameters and constraints are given in Table 10.5. The MPC for area 1 manipulates $\Delta P_{\mathrm{ref}_1}$ and ΔX_{12} to drive $\Delta\omega_1$ and the relative phase difference $\Delta\delta_{12} = \Delta\delta_1 - \Delta\delta_2$ to zero. The MPC for area 2 manipulates $\Delta P_{\mathrm{ref}_2}$ to drive $\Delta\omega_2$ to zero.

The relative performance of cent-MPC, comm-MPC and FC-MPC rejecting a simultaneous 25% increase in the load of areas 1 and 2 is investigated. The closed-loop performance of the different MPC frameworks is shown in Figure 10.4. The associated control costs are given in Table 10.6. The performance of FC-MPC (1 iterate) is within 28% of cent-MPC performance. The performance of comm-MPC, on the other hand, is highly oscillatory and significantly worse than that of FC-MPC (1 iterate). While comm-MPC is stabilizing, the system

Table 10.5: Model parameters and input constraints for the two area power network model. FACTS device operated by area 1.

D_1	= 3	D_2	= 0.275
R_1^f	= 0.03	R_2^f	= 0.07
M_1^a	= 4	M_2^a	= 40
T_{CH_1}	= 5	T_{CH_2}	= 10
T_{G_1}	= 4	T_{G_2}	= 25
T_{12}	= 2.54	K_{12}	= 1.95
Q_1	= $\mathrm{diag}(100, 0, 0, 100)$	Q_2	= $\mathrm{diag}(100, 0, 0)$
R_1	= 1	R_2	= 1
N	= 15	Δ_{samp}	= 1 sec

-0.3$\leq \Delta P_{\mathrm{ref}_1} \leq$0.3
-0.1$\leq \Delta X_{12} \leq$0.1
-0.3$\leq \Delta P_{\mathrm{ref}_2} \leq$0.3

Table 10.6: Performance of different MPC frameworks relative to cent-MPC, $\Delta\Lambda\% = \frac{\Lambda_{\mathrm{config}} - \Lambda_{\mathrm{cent}}}{\Lambda_{\mathrm{cent}}} \times 100$.

	$\Lambda \times 10^{-2}$	$\Delta\Lambda\%$
cent-MPC	3.06	–
comm-MPC	9.53	211
FC-MPC (1 iterate)	3.92	28
FC-MPC (5 iterates)	3.13	2.3

takes nearly 400 sec to reject the load disturbance. With FC-MPC (1 iterate), the load disturbance is rejected in less than 80 sec. If 5 iterates per sampling interval are possible, the FC-MPC framework achieves performance that is within 2.5% of cent-MPC performance.

10.6 Extensions

10.6.1 Penalty and constraints on the rate of change of input

We consider the stage cost defined in Equation (4.15) (p. 68). To convert to the stage cost of Equation (4.15) to the standard form (Equation (4.3)), we augment $x_i(k)$ with the subsystem input $u_i(k-1)$ obtained at time $k-1$ (Muske and Rawlings, 1993). The stage cost can be

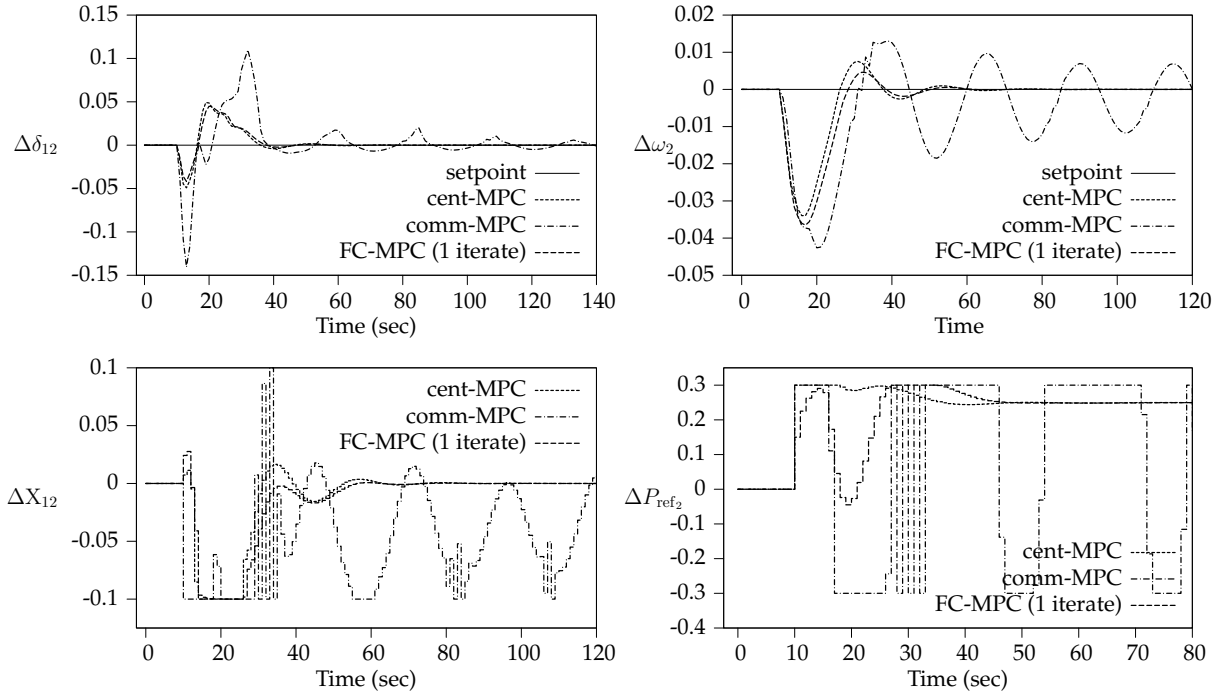

Figure 10.4: Performance of different control frameworks rejecting a load disturbance in area 2. Change in relative phase difference $\Delta\delta_{12}$, frequency $\Delta\omega_2$, tie-line impedence ΔX_{12} due to the FACTS device and load reference setpoint ΔP_{ref_2}.

re-written as

$$L_i(z_i(k), u_i(k)) = \frac{1}{2}\left[z_i(k)'\widetilde{Q}_i z_i(k) + u_i(k)'\widetilde{R}_i u_i(k) + 2z_i(k)'\widetilde{M}_i u_i(k)\right] \tag{10.16}$$

in which

$$z_i(k) = \begin{bmatrix} x_i(k) \\ u_i(k-1) \end{bmatrix} \qquad \widetilde{Q}_i = \begin{bmatrix} Q_i & \\ & S_i \end{bmatrix}$$

$$\widetilde{R}_i = R_i + S_i \qquad \widetilde{M}_i = \begin{bmatrix} 0 \\ -S_i \end{bmatrix}$$

The augmented PM for subsystem $i \in \mathbb{I}_M$ is

$$z_i(k+1) = \widetilde{A}_{ii} z_i(k) + \widetilde{B}_{ii} u_i(k) + \sum_{j \neq i} \left[\widetilde{A}_{ij} z_j(k) + \widetilde{B}_{ij} u_j(k) \right] \tag{10.17}$$

in which

$$\widetilde{A}_{ij} = \begin{bmatrix} A_{ij} & 0 \\ 0 & 0 \end{bmatrix}, \ \forall \, i, j \in \mathbb{I}_M$$

$$\widetilde{B}_{ii} = \begin{bmatrix} B_{ii} \\ I \end{bmatrix}, \quad \widetilde{B}_{ij} = \begin{bmatrix} B_{ij} \\ 0 \end{bmatrix}, \ \forall \, i, j \in \mathbb{I}_M, j \neq i$$

The cost function for subsystem i is defined as

$$\phi_i(\boldsymbol{z}_i, \boldsymbol{u}_i; x(k)) = \sum_{j=k}^{\infty} L_i(z_i(j|k), u_i(j|k)) \tag{10.18}$$

The constraints on the rate of change of input for each subsystem $i \in \mathbb{I}_M$ can, therefore, be expressed as

$$\overline{\boldsymbol{\Delta} \mathbf{u}_i}^{\min} \leq \mathcal{D}_i \, \overline{\boldsymbol{u}}_i \leq \overline{\boldsymbol{\Delta} \mathbf{u}_i}^{\max} \tag{10.19a}$$

in which

$$\overline{\boldsymbol{\Delta}\mathbf{u}_i}^{\min} = \begin{bmatrix} \Delta u_i^{\min} - u_i(k-1) \\ \Delta u_i^{\min} \\ \vdots \\ \Delta u_i^{\min} \end{bmatrix} \qquad \overline{\boldsymbol{\Delta}\mathbf{u}_i}^{\max} = \begin{bmatrix} \Delta u_i^{\max} - u_i(k-1) \\ \Delta u_i^{\max} \\ \vdots \\ \Delta u_i^{\max} \end{bmatrix} \tag{10.19b}$$

$$\mathcal{D}_i = \begin{bmatrix} I & & & & \\ -I & I & & & \\ & -I & I & & \\ & & & \ddots & \\ & & & -I & I \end{bmatrix} \tag{10.19c}$$

Following the model manipulation described in Appendix 10.8.1, with each (A_{ij}, B_{ij}) pair replaced by the corresponding $(\widetilde{A}_{ij}, \widetilde{B}_{ij})$ pair (from the augmented PM in Equation (10.17)), gives

$$\overline{\boldsymbol{z}}_i = E_{ii}\overline{\boldsymbol{u}}_i + f_{ii}z_i(k) + \sum_{j\neq i}[E_{ij}\overline{\boldsymbol{u}}_j + f_{ij}z_j(k)], \quad \forall\, i \in \mathbb{I}_M \tag{10.20}$$

in which $\overline{\boldsymbol{z}}_i = [z_i(k+1|k)', \ldots, z_i(k+N|k)']'$. Similar to Section 10.2, the augmented state z_i in Equation (10.18) can be eliminated using Equation(10.20). The cost function $\phi_i(\cdot)$ can therefore be re-written as a function $\Phi_i(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M; z(k))$ where $z = [z_1', z_2', \ldots, z_M']'$. For $\phi_i(\cdot)$ defined

in Equation (10.18), the FC-MPC optimization problem for subsystem i is

$$\overline{\boldsymbol{u}}_i^{*(p)} \in \arg\min_{\overline{\boldsymbol{u}}_i} \quad \frac{1}{2}\overline{\boldsymbol{u}}_i{}'\mathfrak{R}_i\overline{\boldsymbol{u}}_i + \left(\boldsymbol{r}_i(z(k)) + \sum_{j\neq i}^{M}\mathcal{H}_{ij}\overline{\boldsymbol{u}}_j^{p-1}\right)'\overline{\boldsymbol{u}}_i \tag{10.21a}$$

subject to

$$u_i(j|k) \in \Omega_i, \;\; k \leq j \leq k+N-1 \tag{10.21b}$$

$$\overline{\boldsymbol{\Delta}\mathbf{u}}_i^{\min} \leq \mathcal{D}_i\,\overline{\boldsymbol{u}}_i \leq \overline{\boldsymbol{\Delta}\mathbf{u}}_i^{\max} \tag{10.21c}$$

in which

$$\begin{aligned}
\mathfrak{R}_i &= \left(\mathbb{R}_i + E_{ii}{}'\mathbb{Q}_iE_{ii} + 2E_{ii}{}'\mathbb{M}_i\right) + \sum_{j\neq i}^{M}E_{ji}{}'\mathbb{Q}_jE_{ji} + \sum_{j=1}^{M}E_{ji}{}'\sum_{l\neq j}\mathbb{T}_{jl}E_{li}\\
\mathcal{H}_{ij} &= \sum_{l=1}^{M}E_{li}{}'\mathbb{Q}_lE_{lj} + \mathbb{M}_i{}'E_{ij} + E_{ji}{}'\mathbb{M}_j + \sum_{l=1}^{M}E_{li}{}'\sum_{s\neq l}\mathbb{T}_{ls}E_{sj}\\
\boldsymbol{r}_i(z(k)) &= \left(E_{ii}{}'\mathbb{Q}_i\boldsymbol{g}_i(z(k)) + \mathbb{M}_i{}'\boldsymbol{g}_i(z(k)) + \boldsymbol{p}_iz_i(k)\right)\\
&\quad + \sum_{j\neq i}^{M}E_{ji}{}'\mathbb{Q}_j\boldsymbol{g}_j(z(k)) + \sum_{j=1}^{M}E_{ji}{}'\sum_{l\neq j}\mathbb{T}_{jl}\boldsymbol{g}_l(z(k))\\
\mathbb{Q}_i &= \text{diag}\left(w_i\widetilde{Q}_i(1),\ldots,w_i\widetilde{Q}_i(N-1),\widetilde{P}_{ii}\right)\\
\mathbb{T}_{ij} &= \text{diag}\left(0,\ldots,0,\widetilde{P}_{ij}\right)\\
\mathbb{R}_i &= \text{diag}\left(w_i\widetilde{R}_i(0),w_i\widetilde{R}_i(1),\ldots,w_i\widetilde{R}_i(N-1)\right)\\
\boldsymbol{p}_i{}' &= \begin{bmatrix}w_i\widetilde{M}_i & 0 & \ldots & 0\end{bmatrix}
\end{aligned}$$

$$\mathbb{M}_i = \begin{pmatrix} 0 & w_i\widetilde{M}_i & & & \\ & 0 & w_i\widetilde{M}_i & & \\ & & 0 & \ddots & \\ & & & \ddots & w_i\widetilde{M}_i \\ 0 & 0 & \dots & \dots & 0 \end{pmatrix}$$

The terminal penalty $\widetilde{P}$ is obtained as the solution to the centralized Lyapunov equation (Equation (10.8)) with each A_{ij}, Q_i replaced by $\widetilde{A}_{ij}, \widetilde{Q}_i$ respectively $\forall\, i, j \in \mathbb{I}_M$.

10.6.2 Unstable systems

In the development of the proposed distributed MPC framework, it was convenient to assume that the system is open-loop stable. That assumption can be relaxed however. For any real matrix $A \in \mathbb{R}^{n\times n}$ the **Schur decomposition** (Golub and Van Loan, 1996, p. 341) is defined as

$$A = \begin{bmatrix} U_s & U_u \end{bmatrix} \begin{bmatrix} A_s & A_{12} \\ 0 & A_u \end{bmatrix} \begin{bmatrix} U_s' \\ U_u' \end{bmatrix} \tag{10.22}$$

in which $U = \begin{bmatrix} U_s & U_u \end{bmatrix}$ is a real and orthogonal $n \times n$ matrix, the eigenvalues of A_s are strictly inside the unit circle, and the eigenvalues of A_u are on or outside the unit circle. Let ${U_u}' = [{U_{u_1}}', {U_{u_2}}', \dots, {U_{u_M}}']$.

Define ${\mathcal{T}_i}' = [0, 0, \dots, I]$ such that $x_i(k+N|k) = {\mathcal{T}_i}'\overline{\boldsymbol{x}}_i(k)$. To ensure closed-loop stability while dealing with open-loop unstable systems, a terminal state constraint that forces the unstable modes to be at the origin at the end of the control horizon is necessary. The control horizon must satisfy $N \geq \alpha$, in which α is the number of unstable modes.

For each subsystem $i \in \mathbb{I}_M$ at time k, the terminal state constraint can be written as

$$\begin{aligned} U_u{}'x(k+N|k) &= \sum_i U_{u_i}{}'x_i(k+N|k) \\ &= \sum_i \left(\mathcal{T}_i U_{u_i}\right)'\overline{\boldsymbol{x}}_i(k) \\ &= 0 \end{aligned} \tag{10.23}$$

From Equations (10.23) and (10.4), the terminal state constraint can be re-written as a coupled input constraint of the form

$$\mathcal{J}_1\overline{\boldsymbol{u}}_1 + \mathcal{J}_2\overline{\boldsymbol{u}}_2 + \ldots + \mathcal{J}_M\overline{\boldsymbol{u}}_M = -\mathbf{c}(x(k)) \tag{10.24a}$$

in which

$$\mathcal{J}_i = \sum_{j=1}^{M}\left(\mathcal{T}_j U_{u_j}\right)'E_{ji} \qquad \mathbf{c}(x(k)) = \sum_{j=1}^{M}\left(\mathcal{T}_j U_{u_j}\right)'\boldsymbol{g}_j(x(k)) \tag{10.24b}$$

$$\forall\, i \in \mathbb{I}_M$$

Using the definitions in Equation (10.6), the FC-MPC optimization problem for each $i \in \mathbb{I}_M$ is

$$\mathcal{F}_i^{\text{unstb}} \triangleq \min_{\overline{\boldsymbol{u}}_i} \quad \frac{1}{2}\overline{\boldsymbol{u}}_i{}'\mathfrak{R}_i\overline{\boldsymbol{u}}_i + \left(\boldsymbol{r}_i(x(k)) + \sum_{j\neq i}\mathcal{H}_{ij}\overline{\boldsymbol{u}}_j^{p-1}\right)'\overline{\boldsymbol{u}}_i \tag{10.25a}$$

subject to

$$u_i(t|k) \in \Omega_i, \;\; k \leq t \leq k+N-1 \tag{10.25b}$$

$$\mathcal{J}_i\overline{\boldsymbol{u}}_i + \sum_{j\neq i}^{M}\mathcal{J}_j\overline{\boldsymbol{u}}_j^{p-1} = -\mathbf{c}(x(k)) \tag{10.25c}$$

The optimization problem of Equation (10.25) is solved within the framework of Algorithm 4.1. To initialize Algorithm 4.1, a simple quadratic program is solved to compute subsystem input trajectories that satisfy the constraints in Equation (10.25) for each subsystem. To ensure feasibility of the end constraint (Equation (10.25c)), it is assumed that the initial state $x(0) \in \mathcal{X}_N$, the N-step stabilizable set for the system. Since $\mathcal{X}_N \subseteq \mathcal{X}$, the system is constrained stabilizable. It follows from Algorithm 4.1, Section 10.3.3 and Section 10.3.5 that $\mathcal{X}_N$ is a positively invariant set for the nominal closed-loop system, which ensures that the optimization problem (Equation (10.25)) is feasible for each subsystem $i \in \mathbb{I}_M$ for all $k \geq 0$ and any $p(k) > 0$. It can be shown that all iterates generated by Algorithm 4.1 are systemwide feasible, the cooperation-based cost function $\Phi(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; x(k))$ is a nonincreasing function of the iteration number p, and the sequence of cooperation-based iterates is convergent [4]. An important distinction, which arises due to the presence of the coupled input constraint (Equation (10.25c)), is that the limit points of Algorithm 4.1 (now solving optimization problem of Equation (10.25) instead) are no longer necessarily optimal (see Section 4.9.2 for examples). The distributed MPC control law based on any intermediate iterate is feasible and closed-loop stable, but may not achieve optimal (centralized) performance at convergence of the iterates.

10.6.3 Terminal control FC-MPC

The distributed LQR framework presented in Chapter 9 is used for terminal control FC-MPC. The modeling framework described in Section 10.1 is employed. The motivation for terminal control FC-MPC is to achieve infinite horizon optimal (centralized, constrained LQR) performance at convergence using finite values of N. For brevity, we omit details of this framework

[4] The proof is identical to that presented for Lemma 4.4 (p. 44) and is, therefore, omitted.

here. An algorithm for terminal control FC-MPC employing the modeling framework of Section 10.1 is presented in Venkat, Hiskens, Rawlings, and Wright (2006c). Two examples for terminal control FC-MPC are provided below.

Two area power system with FACTS device

We revisit the two area power system considered in Section 10.5.3. A 28% load increase affects area 1 at time 10 sec and simultaneously, an identical load disturbance affects area 2. The controller parameters are $R_1 = \mathrm{diag}(1,1), R_2 = 1, Q_1 = \mathrm{diag}(10,0,0,10), Q_2 = \mathrm{diag}(10,0,0)$, $\Delta_{\mathrm{samp}} = 2$ sec. The controlled variables (CVs) for the MPC in area 1 are $\Delta\omega_1$ and $\Delta\delta_{12}$. The CV for the MPC in area 2 is $\Delta\omega_2$. In this case, we evaluate the load disturbance rejection performance of terminal control FC-MPC (FC-MPC (tc)) and compare it against the performance of terminal penalty FC-MPC (FC-MPC (tp)) and centralized constrained LQR (CLQR).

The relative performance of FC-MPC(tc), FC-MPC(tp) and CLQR rejecting the described load disturbance is shown in Figure 10.5. For terminal control FC-MPC employing Algorithm 9.4, an initial control horizon length (N_0) of 20 is selected. This choice of N is sufficient to steer the dummy state vectors $\zeta_i(\cdot), \forall\ i \in \mathbb{I}_M$ to $\mathcal{O}_\infty$ throughout the period where the effect of the load disturbance persists. The terminal penalty FC-MPC employs Algorithm 4.1 (Section 10.3) .

Due to an increase in load in both control areas, the MPCs (in areas 1 and 2) order an increase in generation. In Figure 10.5, the transient tie-line power flow and frequency deviations under FC-MPC (tc, 1 iterate) are almost identical to the infinite horizon optimal CLQR performance. The incurred control costs are given in Table 10.7. FC-MPC (tc, 1 iterate) achieves a performance improvement of about $\sim 16\%$ compared to FC-MPC (tp, 1 iterate). If 5 iterates

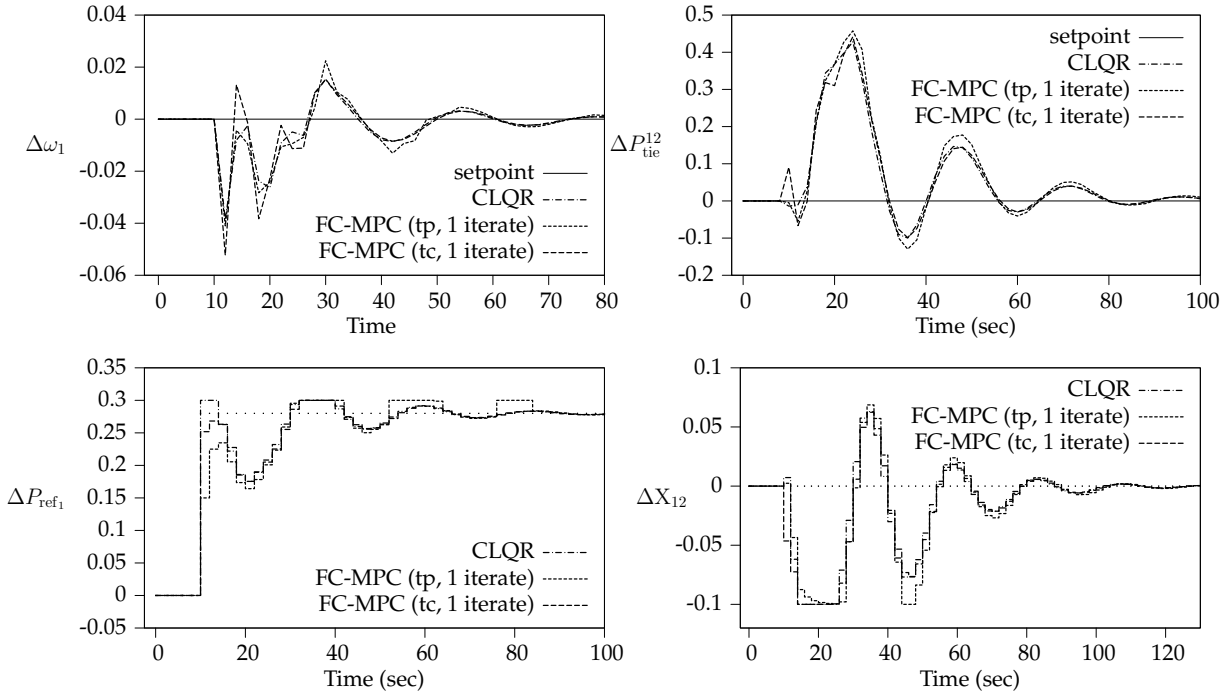

Figure 10.5: Comparison of load disturbance rejection performance of terminal control FC-MPC, terminal penalty FC-MPC and CLQR. Change in frequency $\Delta\omega_1$, tie-line power flow $\Delta P_{\text{tie}}^{12}$, load reference setpoints ΔP_{ref_1} and ΔP_{ref_2}.

per sampling interval are permissible, the disturbance rejection performance of FC-MPC (tc, 5 iterates) is within 0.5% of CLQR performance. The performance loss incurred under FC-MPC (tp, 5 iterates), relative to CLQR performance, is about 13%, which is significantly higher than the performance loss incurred with FC-MPC (tc, 5 iterates).

Table 10.7: Performance of different control formulations relative to centralized constrained LQR (CLQR), $\Delta\Lambda\% = \frac{\Lambda_{\text{config}} - \Lambda_{\text{cent}}}{\Lambda_{\text{cent}}} \times 100$.

	$\Lambda \times 10^{-3}$	$\Delta\Lambda\%$
CLQR	1.77	
FC-MPC (tp, 1 iterate)	2.21	25
FC-MPC (tc, 1 iterate)	1.93	9.2
FC-MPC (tp, 5 iterates)	2	12.9
FC-MPC (tc, 5 iterates)	1.774	< 0.2

Unstable four area power network

Consider the four area power network described in Section 10.5.2. In this case though, $M_4^a = 40$ to force the system to be open-loop unstable. The regulator parameters are specified in Table 10.8. The sampling interval $\Delta_{\text{samp}} = 2$ sec. At time 10 sec, the load in area 2 increases by 15% and simultaneously, the load in area 3 decreases by 15%. The load disturbance rejection performance of terminal control FC-MPC (FC-MPC(tc)) is investigated and compared to the performance of the benchmark CLQR.

Figure 10.6 depicts the disturbance rejection performance of FC-MPC (tc) and CLQR. Only quantities relating to area 2 are shown as variables in other areas displayed similar qualitative behavior. The associated control costs are given in Table 10.9. For terminal control FC-MPC terminated after 1 iterate, the load disturbance rejection performance is within 13% of CLQR performance. If 5 iterates per sampling interval are possible, the incurred performance loss drops to $< 1.5\%$.

Table 10.8: Regulator parameters for unstable four area power network.

$Q_1 = \text{diag}(50, 0, 0)$	$R_1 = 1$
$Q_2 = \text{diag}(50, 0, 0, 50)$	$R_2 = 1$
$Q_3 = \text{diag}(50, 0, 0, 50)$	$R_3 = 1$
$Q_4 = \text{diag}(50, 0, 0, 50)$	$R_4 = 1$

Table 10.9: Performance of terminal control FC-MPC relative to centralized constrained LQR (CLQR), $\Delta\Lambda\% = \frac{\Lambda_{\text{config}} - \Lambda_{\text{cent}}}{\Lambda_{\text{cent}}} \times 100$.

	$\Lambda \times 10^{-2}$	$\Delta\Lambda\%$
CLQR	4.91	
FC-MPC (tc, 1 iterate)	5.52	12.4
FC-MPC (tc, 5 iterates)	4.97	1.2

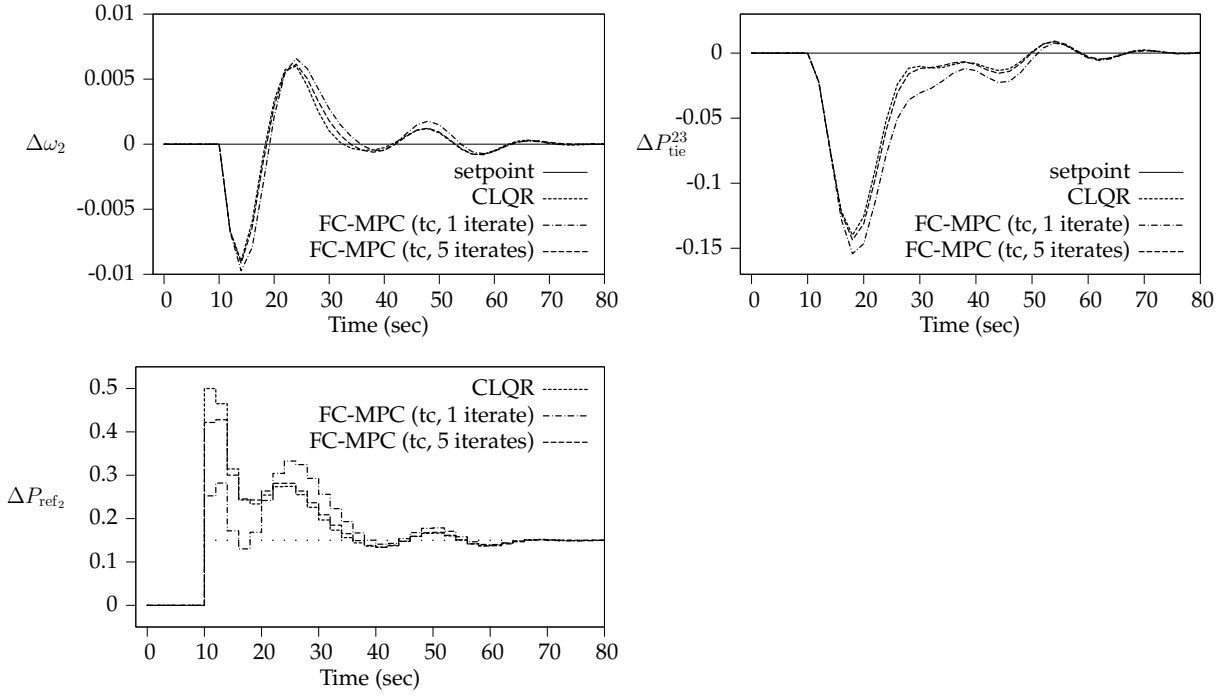

Figure 10.6: Performance of FC-MPC (tc) and CLQR, rejecting a load disturbance in areas 2 and 3. Change in local frequency $\Delta\omega_2$, tie-line power flow $\Delta P_{\text{tie}}^{23}$ and load reference setpoint ΔP_{ref_2}.

10.7 Discussion and conclusions

Centralized MPC is not well suited for control of large-scale, geographically expansive systems such as power systems. However, performance benefits obtained with centralized MPC can be realized through distributed MPC strategies. Distributed MPC strategies for power systems rely on decomposition of the overall system into interconnected subsystems, and iterative optimization and exchange of information between these subsystems. An MPC optimization problem is solved within each subsystem, using local measurements and the latest available external information (from the previous iterate). Feasible cooperation-based MPC (FC-MPC) precludes the possibility of parochial controller behavior by forcing the MPCs to cooperate towards attaining systemwide objectives. A terminal penalty version of FC-MPC was initially

established. The solution obtained at convergence of the FC-MPC algorithm is identical to the centralized MPC solution (and therefore, Pareto optimal). In addition, the FC-MPC algorithm can be terminated prior to convergence without compromising feasibility or closed-loop stability of the resulting distributed controller. This feature allows the practitioner to terminate the algorithm at the end of the sampling interval, even if convergence is not achieved. A terminal control FC-MPC framework, which achieves infinite horizon optimal performance at convergence, has also been considered. For small values of N, the performance of terminal control FC-MPC is superior to that of terminal penalty FC-MPC.

Examples were presented to illustrate the applicability and effectiveness of the proposed distributed MPC framework for automatic generation control (AGC). First, a two area network was considered. Both communication-based MPC and cooperation-based MPC outperformed AGC due to their ability to handle process constraints. The controller defined by terminating Algorithm 4.1 after 5 iterates achieves performance that is almost identical to centralized MPC. Next, the performance of the different MPC frameworks are evaluated for a four area network. For this case, communication-based MPC leads to closed-loop instability. FC-MPC (1 iterate) stabilizes the system and achieves performance that is within 26% of centralized MPC performance. The two area network considered earlier, with an additional FACTS device to control tie line impedence, is examined subsequently. Communication-based MPC stabilizes the system but gives unacceptable closed-loop performance. The FC-MPC framework is shown to allow coordination of FACTS controls with AGC. The controller defined by terminating Algorithm 4.1 after just 1 iterate gives an $\sim 190\%$ improvement in performance compared to communication-based MPC. For this case, therefore, the cooperative aspect of FC-MPC was very important for achieving acceptable response. Next, the two area network with

FACTS device was used to compare the performance of terminal penalty FC-MPC and terminal control FC-MPC. As expected, terminal control FC-MPC outperforms terminal penalty FC-MPC for short horizon lengths. Finally, the performance of terminal control FC-MPC is evaluated on an unstable four area network. FC-MPC (tc, 5 iterates) achieves performance that is within 1.5% of the centralized constrained LQR performance.

10.8 Appendix

10.8.1 Model Manipulation

To ensure strict feasibility of the FC-MPC algorithm, it is convenient to eliminate the states $\overline{\boldsymbol{x}}_i$, $i \in \mathbb{I}_M$ using the PM (10.1). Propagating the model for each subsystem through the control horizon N gives

$$\overline{\boldsymbol{x}}_i = \overline{E}_{ii}\overline{\boldsymbol{u}}_i + \overline{f}_{ii}x_i(k) + \sum_{j\neq i}[\overline{E}_{ij}\overline{\boldsymbol{u}}_j + \overline{g}_{ij}\overline{\boldsymbol{x}}_j + \overline{f}_{ij}x_j(k)]$$

$$\forall\, i \in \mathbb{I}_M \tag{10.26}$$

in which

$$\overline{E}_{ij} = \begin{bmatrix} B_{ij} & 0 & \dots & \dots & 0 \\ A_{ii}B_{ij} & B_{ij} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{ii}^{N-1}B_{ij} & \dots & \dots & \dots & B_{ij} \end{bmatrix} \qquad \overline{f}_{ij} = \begin{bmatrix} A_{ij} \\ A_{ii}A_{ij} \\ \vdots \\ A_{ii}^{N-1}A_{ij} \end{bmatrix}$$

$$\overline{g}_{ij} = \begin{bmatrix} 0 & 0 & \dots & \dots & 0 \\ A_{ij} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A_{ii}^{N-2}A_{ij} & A_{ii}^{N-3}A_{ij} & \dots & \dots & 0 \end{bmatrix}.$$

Combining the models in (10.26), $\forall\ i\ = 1, 2, \dots, M$, gives the following system of equations

$$\mathcal{A}\widetilde{\boldsymbol{x}} = \mathcal{E}\widetilde{\boldsymbol{u}} + \mathcal{G}x(k) \tag{10.27}$$

in which

$$\mathcal{G} = \begin{bmatrix} \overline{f}_{11} & \overline{f}_{12} & \dots & \overline{f}_{1M} \\ \overline{f}_{21} & \overline{f}_{22} & \dots & \overline{f}_{2M} \\ \ddots & \ddots & \ddots & \ddots \\ \overline{f}_{M1} & \dots & \dots & \overline{f}_{MM} \end{bmatrix} \quad \mathcal{E} = \begin{bmatrix} \overline{E}_{11} & \overline{E}_{12} & \dots & \overline{E}_{1M} \\ \overline{E}_{21} & \overline{E}_{22} & \dots & \overline{E}_{2M} \\ \ddots & \ddots & \ddots & \ddots \\ \overline{E}_{M1} & \dots & \dots & \overline{E}_{MM} \end{bmatrix}$$

(10.28)

$$\mathcal{A} = \begin{bmatrix} I & -\overline{g}_{12} & \dots & -\overline{g}_{1M} \\ -\overline{g}_{21} & I & \dots & -\overline{g}_{2M} \\ \ddots & \ddots & \ddots & \ddots \\ -\overline{g}_{M1} & \dots & \dots & I \end{bmatrix} \quad \widetilde{\boldsymbol{x}} = \begin{bmatrix} \overline{\boldsymbol{x}}_1 \\ \overline{\boldsymbol{x}}_2 \\ \vdots \\ \overline{\boldsymbol{x}}_M \end{bmatrix} \quad \widetilde{\boldsymbol{u}} = \begin{bmatrix} \overline{\boldsymbol{u}}_1 \\ \overline{\boldsymbol{u}}_2 \\ \vdots \\ \overline{\boldsymbol{u}}_M \end{bmatrix} \tag{10.29}$$

Since the system is LTI, a solution to the system (10.27) exists for each permissible RHS. Matrix $\mathcal{A}$ is therefore invertible and consequently, we can write for each $i \in \mathbb{I}_M$

$$\overline{\boldsymbol{x}}_i = E_{ii}\overline{\boldsymbol{u}}_i + f_{ii}x_i(k) + \sum_{j \neq i}[E_{ij}\overline{\boldsymbol{u}}_j + f_{ij}x_j(k)]. \tag{10.30}$$

in which E_{ij} and $f_{ij},\ \forall\ j = 1, 2, \dots, M$ denote the appropriate partitions of $\mathcal{A}^{-1}\mathcal{E}$ and $\mathcal{A}^{-1}\mathcal{G}$ respectively.

Lemma 10.1. *Let the input constraints in Equation (10.6) be specified in terms of a collection of linear inequalities. Consider the closed ball $B_\varepsilon(0)$, in which $\varepsilon > 0$ is chosen such that the input constraints in each FC-MPC optimization problem (Equation (10.6)) are inactive for all $x \in B_\varepsilon(0)$. The distributed MPC control law defined by the FC-MPC formulation of Theorem 10.1 is a Lipschitz continuous function of x for all $x \in B_\varepsilon(0)$.*

The proof is identical to the proof for Lemma 4.7 (p. 79) with each μ replaced by x.

Proof of Theorem 10.1. Since $\mathcal{Q} > 0$ and A is stable, $P > 0$ (Sontag, 1998). The constrained stabilizable set $\mathcal{X}$ for the system is $\mathbb{R}^n$. To prove exponential stability, we use the value function $J_N^{p(k)}(x(k))$ as a candidate Lyapunov function. We need to show (Vidyasagar, 1993, p. 267) that

there exists constants $a, b, c > 0$ such that

$$a\|x(k)\|^2 \leq J_N^p(x(k)) \leq b\|x(k)\|^2 \tag{10.31a}$$

$$\Delta J_N^p(x(k)) \leq -c\|x(k)\|^2 \tag{10.31b}$$

in which $\Delta J_N^{p(k)}(x(k)) = J_N^{p(k+1)}(x(k+1)) - J_N^{p(k)}(x(k))$.

Let $\varepsilon > 0$ be chosen such that the input constraints remain inactive for $x \in B_\varepsilon(0)$. Such an ε exists because the origin is Lyapunov stable and $0 \in \text{int}(\Omega_1 \times \ldots \Omega_M)$. Since $\Omega_i, \forall\, i \in \mathbb{I}_M$ is compact, there exists $\sigma > 0$ such that $\|\overline{\boldsymbol{u}}_i\| \leq \sigma$. For any x satisfying $\|x\| > \varepsilon$, $\|\overline{\boldsymbol{u}}_i\| < \frac{\sigma}{\varepsilon}\|x\|, \forall i \in \mathbb{I}_M$. For $x \in B_\varepsilon(0)$, we have from Lemma 10.1 that $\overline{\boldsymbol{u}}_i^{p(k)}(x)$ is a Lipschitz continuous function of x. There exists, therefore, a constant $\rho > 0$ such that $\|\overline{\boldsymbol{u}}_i^{p(k)}(x)\| \leq \rho\|x\|$, $\forall\, 0 < p(k) \leq p^*$. Define $K_u = \max\left(\frac{\sigma}{\varepsilon}, \rho\right)^2$, in which $K_u > 0$ and independent of x. The above definition gives $\|u_i^{p(k)}(x, j)\| \leq \sqrt{K_u}\|x\|,\ \forall\, i \in \mathbb{I}_M$ and all $0 < p \leq p^*$. For $j \geq 0$, define $u(x(k), j) = [u_1^{p(k)}(x(k), j)', \ldots, u_M^{p(k)}(x(k), j)']'$. By definition, $u(x(k), k) \equiv u(x(k))$. We have

$$\|u(x(k), j)\| = \sqrt{\sum_{i=1}^{M} \|u_i^{p(k)}(x(k), j)\|^2} \leq \sqrt{K_u M}\|x(k)\|.$$

Similarly, define $x(k + j|k) = [x_1^{p(k)}(k + j|k)', \ldots, x_M^{p(k)}(k + j|k)']',\ \forall\, j \geq 0$. By definition $x(k|k) \equiv x(k)$. Since A is stable, there exists $\bar{c} > 0$ such that $\|A^j\| \leq \bar{c}\lambda^j$ (Horn and Johnson,

1985, Corollary 5.6.13, p. 199), in which $\lambda_{\max}(A) \leq \lambda < 1$. Hence,

$$\begin{aligned}
\|x(k+j|k)\| &\leq \|A^j\|\|x(k)\| + \sum_{l=0}^{j-1} \|A^{j-1-l}\|\|B\|\|u(x(k),l)\| \\
&\leq \overline{c}\lambda^j\|x(k)\| + \sum_{l=0}^{j-1} \overline{c}\lambda^{j-1-l}\|B\|\|u(x(k),l)\| \\
&\leq \overline{c}\left(1 + \frac{\|B\|}{1-\lambda}\sqrt{MK_u}\right)\|x(k)\|, \ \forall\, j > 0, \\
&\text{(since } \sum_{l=0}^{j} \lambda^l \leq \sum_{l=0}^{\infty} \lambda^l = \frac{1}{1-\lambda}, \ \forall\, j \geq 0).
\end{aligned}$$

Let $\mathcal{R} = \text{diag}(w_1R_1, w_2R_2, \ldots, w_MR_M)$ and $\Gamma = \left[\overline{c}\left(1 + \frac{\|B\|}{1-\lambda}\sqrt{MK_u}\right)\right]^2$.

$$\begin{aligned}
J_N^{p(k)}(x(k)) &= \frac{1}{2}\sum_{i=1}^{M} w_i \sum_{j=0}^{\infty}\left[\|x_i^{p(k)}(k+j|k)\|_{Q_i}^2 + \|u_i^{p(k)}(x(k),j)\|_{R_i}^2\right] \\
&= \frac{1}{2}\sum_{j=0}^{N-1}\left[x(k+j|k)'\mathcal{Q}x(k+j|k) + u(x(k),j)'\mathcal{R}u(x(k),j)\right] \\
&\qquad + \frac{1}{2}x(k+N|k)'Px(k+N|k) \\
&\leq \frac{1}{2}\Bigg[\sum_{j=0}^{N-1}\left(\lambda_{\max}(\mathcal{Q})\|x(k+j|k)\|^2 + \lambda_{\max}(\mathcal{R})\|u(x(k),j)\|^2\right) \\
&\qquad + \lambda_{\max}(P)\|x(k+N|k)\|^2\Bigg] \\
&\leq \frac{1}{2}\left[N\lambda_{\max}(\mathcal{Q})\Gamma + N\lambda_{\max}(\mathcal{R})K_uM + \lambda_{\max}(P)\Gamma\right]\|x(k)\|^2 \\
&\leq b\|x(k)\|^2
\end{aligned}$$

in which $0 < \frac{1}{2}\left[N\lambda_{\max}(\mathcal{Q})\Gamma + N\lambda_{\max}(\mathcal{R})K_u M + \lambda_{\max}(P)\Gamma\right] \leq b$. Also, $\frac{1}{2}\lambda_{\min}(\mathcal{Q})\|x(k)\|^2 \leq J_N^{p(k)}(x(k))$ Furthermore,

$$\begin{aligned} J_N^{p(k+1)}(x(k+1)) - J_N^{p(k)}(x(k)) &\leq J_N^0(x(k+1)) - J_N^{p(k)}(x(k)) \\ &= -\sum_{i=1}^{M} w_i L_i\left(x_i(k), u_i^{p(k)}(x(k), 0)\right) \\ &\leq -\sum_{i=1}^{M} w_i L_i\left(x_i(k), 0\right) \\ &= -\frac{1}{2}x(k)'\mathcal{Q}x(k) \\ &\leq -\frac{1}{2}\lambda_{\min}(\mathcal{Q})\|x(k)\|^2 \end{aligned} \tag{10.32}$$

which proves the theorem. □

Chapter 11

Asynchronous feedback for distributed MPC.

In previous chapters, it was assumed that all subsystems are sampled at an uniform rate. In many situations, time constants and subsequently sampling rates for different subsystems may vary significantly. Furthermore, measurements for certain subsystems may be sampled slower than others. Low measurement sampling rates may be due to technological limitations and/or cost of measurement. It is well known that control at the slowest sampling rate may result in significant loss in achievable control performance. In this chapter, an asynchronous feedback distributed MPC framework that allows subsystems' MPCs to operate at different sampling rates is described. The goal of asynchronous feedback is to enable faster sampled MPCs to inject their control actions into the plant while slower sampled MPCs compute their desired control action. For simplicity, we restrict our analysis here to systems with two sampling rates.

This chapter is organized as follows. Modeling aspects and notation for asynchronous feedback FC-MPC is described in Section 11.1. In Section 11.2, we tackle the problem of asyn-

chronous feedback in FC-MPC. Algorithms are described for both fast sampled MPCs and slow sampled MPCs. The asynchronous feedback control law is defined in Section 11.3. Some implementation issues are considered subsequently and an illustrative case study is presented. Nominal closed-loop stability for the asynchronous feedback distributed MPC control law is established in Section 11.4. An example is presented in Section 11.5 to demonstrate the benefits of employing asynchronous feedback distributed MPC. Finally, the contributions of this chapter are summarized in Section 11.6.

11.1 Models and groups

Consider the following LTI continuous time model for each subsystem $i \in \mathbb{I}_M$

$$\frac{dx_{ii}}{dt} = A^c_{ii}x_{ii} + B^c_{ii}u_i \tag{11.1a}$$

$$\frac{dx_{ij}}{dt} = A^c_{ij}x_{ii} + B^c_{ij}u_j,\ \forall\, j \in \mathbb{I}_M, j \neq i \tag{11.1b}$$

$$y_i = C_{ii}x_{ii} + \sum_{j\neq i} C_{ij}x_{ij} \tag{11.1c}$$

in which t is continuous time. Let $\mathbb{I}$ represent the set of integers and define $\mathbb{I}_+$ to be the set of positive integers. Each subsystem $i \in \mathbb{I}_M$ is assumed to belong to either the group of MPCs sampled at the *fast* rate, $\mathcal{J}_{\text{fast}}$, or the group of MPCs sampled at the slow rate, $\mathcal{J}_{\text{slow}}$. Let the sampling rate for each subsystem $i \in \mathcal{J}_{\text{fast}}$ be t_f. The sampling rate for each subsystem $i \in \mathcal{J}_{\text{slow}}$ is T, where $T = \sigma t_f, \sigma \in \mathbb{I}_+$. Let $\mathcal{J}_{\text{fast}}$, $\mathcal{J}_{\text{slow}}$ consist of M_1, M_2 MPCs, respectively, with $M_1 + M_2 = M$. Let (WLOG) $\mathcal{J}_{\text{fast}} = \{1, 2, \ldots, M_1\}$ and $\mathcal{J}_{\text{slow}} = \{M_1 + 1, M_1 + 2, \ldots, M\}$. A discrete time realization of the continuous time subsystem model (Equation (11.1)) for sub-

system $i \in \mathbb{I}_M$, at the (faster) sampling rate t_f is (Kailath, 1980)

$$x_{ii}((s+1)t_f) = A_{ii}x(st_f) + B_{ii}u_i(st_f) \tag{11.2a}$$

$$x_{ij}((s+1)t_f) = A_{ij}x(st_f) + B_{ij}u_i(st_f),\ \forall\, j \in \mathbb{I}_M, j \neq i \tag{11.2b}$$

$$y_i(st_f) = C_{ii}x_{ii}(st_f) + \sum_{j \neq i} C_{ij}x_{ij}(st_f) \tag{11.2c}$$

in which

$$A_{ij} = \exp[A_{ij}t_f] \qquad B = \int_0^{t_f} \exp[A_{ij}(t_f - \tau)]B_{ij}\, d\tau,\ \forall\, j \in \mathbb{I}_M$$

The composite model (CM) $(A_i, B_i, \{W_{ij}\}, C_i)$ for subsystem $i \in \mathbb{I}_M$ is given by Equation (4.1) (p. 27). The overall plant CM is obtained by collecting these individual subsystem CMs and is given by Equation (4.2) (p. 28).

During each sampling interval $[(\beta-1)t_f, \beta t_f)$, $\beta \in \mathbb{I}_+$, MPCs in $\mathcal{J}_{\text{fast}}$ perform optimizations in parallel (one for each MPC) and exchange input trajectories. Each such optimization and exchange of information between MPCs in $\mathcal{J}_{\text{fast}}$ is termed an *inner iterate*. Let $q_f \in \mathbb{I}_+$ inner iterates be performed during each sampling interval (duration t_f) for MPCs in $\mathcal{J}_{\text{fast}}$.

During each $[kT, (k+1)T), k \geq 0, k \in \mathbb{I}$, each MPC $i \in \mathcal{J}_{\text{slow}}$ performs an optimization, and at time $(k+1)T$, transmits calculated input trajectories to all MPCs $j \in \mathbb{I}_M, j \neq i$. At $(k+1)T$, each MPC $i \in \mathcal{J}_{\text{slow}}$ also receives latest input trajectories from MPCs in $\mathcal{J}_{\text{fast}}$. Information transfer across the groups occurs only during synchronization *i.e.*, at each kT. Each synchronization between MPCs in the two groups is also called an *outer iterate* and occurs periodically at the rate T (sampling rate of $\mathcal{J}_{\text{slow}}$).

Interaction model information requirement. To enable asynchronous feedback, we make the following assumption:

Assumption 11.1. Each MPC $i \in \mathcal{J}_{\text{fast}}$ has explicit knowledge of the set of interaction models $(A_{ji}, B_{ji}),\ j \in \mathcal{J}_{\text{slow}}$.

The interaction model (A_{ji}, B_{ji}) in Assumption 11.1 describes the effect of control action initiated by MPC $i \in \mathcal{J}_{\text{fast}}$ on subsystem $j \in \mathcal{J}_{\text{slow}}$. Assumption 11.1 allows MPC $i \in \mathcal{J}_{\text{fast}}$ to assess the impact of its control moves on subsystem $j \in \mathcal{J}_{\text{slow}}$ without requiring any information exchange between the two groups during the sampling interval $[kT, (k+1)T)$. The advantage of this approach will be highlighted in the sequel.

At time kT, define

$$\boldsymbol{u}_i(kT) = [u_i(kT|kT)', u_i(kT + t_f|kT)', u_i(kT + 2t_f|kT)', \ldots\ldots]'$$

to be the predicted input trajectory for subsystem i. For notational simplicity, we write $k \leftarrow kT$ and $\alpha \leftarrow \alpha t_f,\ \forall\, \alpha \in \mathbb{I}_+, k \in \mathbb{I}$ (see Figure 11.1) . At time $kT + \alpha t_f$, the predicted input for subsystem i at time $kT + (\alpha + s)t_f, s \geq 0, s \in \mathbb{I}$ is written as $u_i([k, \alpha], s)$. The predicted input trajectory for $i \in \mathcal{J}_{\text{fast}}$ at time $kT + \delta t_f,\ \delta \in \mathbb{I}_+$ is defined as

$$\boldsymbol{u}_i([k, \delta]) = [u_i([k, \delta], 0)', u_i([k, \delta], 1)', u_i([k, \delta], 2)', \ldots\ldots]'$$

The corresponding shifted input trajectory at time $kT + (\delta + s)t_f$ is denoted as

$$\boldsymbol{u}_i([k, \delta], s) = [u_i([k, \delta], s)', u_i([k, \delta], s + 1)', \ldots\ldots]'.$$

Assumption 11.2. A zero-order hold (ZOH) is employed for each MPC $i \in \mathbb{I}_M$.

From Assumption 11.2, we have for each $i \in \mathcal{J}_{\text{slow}}$ that

$$u_i([k,0]) = u_i([k,0],1) = \ldots = u_i([k,0],\sigma-1),\ \forall\, k \geq 0.$$

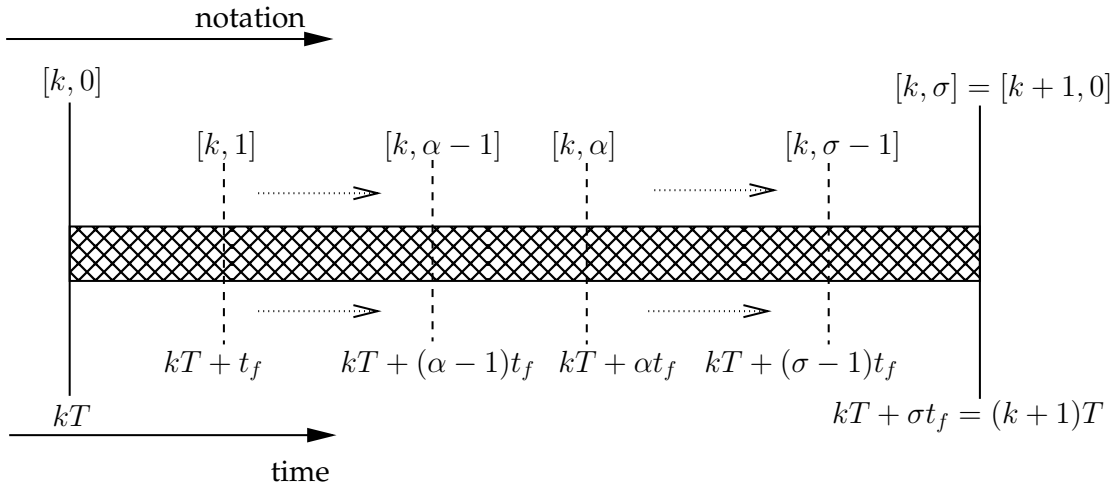

Figure 11.1: Time scales for asynchronous feedback distributed MPC.

For subsystem i, let $x_i([k,0];[k-1,0])$ denote an estimate of the subsystem state at time kT using measurements up to and including time $(k-1)T$. If the measurement at time kT is available, the estimate is represented as $x_i([k,0]) \equiv x_i([k,0];[k,0])$. Similarly, the estimate of the states of subsystem i at time $kT + \alpha t_f$ given measurements up to and including time $kT + \beta t_f,\ \beta \leq \alpha$ is denoted as $x_i([k,\alpha];[k,\beta])$. Define $\mu_{\text{fast}} = [x_1, \ldots, x_{M_1}]$, $\mu_{\text{slow}} = [x_{M_1+1}, \ldots, x_M]$ to be the set of subsystem states in $\mathcal{J}_{\text{fast}}$ and $\mathcal{J}_{\text{slow}}$ respectively. The notation $\mu([k,\alpha];[\beta_1,\beta_2]) = [\mu_{\text{fast}}([k,\alpha];[k,\beta_1]),\ \mu_{\text{slow}}([k,\alpha];[k,\beta_2])]$. For convenience, we write $\mu([k,0]) \equiv \mu([k,0];[0,0])$. At time t, let $u_i^{\text{inj}}(t)$ denote the input injected into subsystem $i \in \mathbb{I}_M$. The input injected into the plant $u(t) = \text{vec}(u_1^{\text{inj}}(t), u_2^{\text{inj}}(t), \ldots, u_M^{\text{inj}}(t))$.

Define $\pi_{\text{fast}} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{M_1}]$ and $\pi_{\text{slow}} = [\boldsymbol{u}_{M_1+1}, \ldots, \boldsymbol{u}_M]$. By definition

$$\Phi(\pi^a_{\text{fast}}, \pi^b_{\text{slow}}; \mu(\cdot)) = \Phi([\boldsymbol{u}^a_1, \ldots, \boldsymbol{u}^a_{M_1}], [\boldsymbol{u}^b_{M_1+1}, \ldots, \boldsymbol{u}^b_M]; \mu(\cdot))$$

For each subsystem i, a parameter $\delta_i > 0,\ \sum_i \delta_i = 1$ must be selected for the inner iterates. One suitable choice is $\delta_i = w_i,\ i \in \mathbb{I}_M$, the relative weight/importance assigned to each subsystem. To allow periodic synchronization of the two asynchronous groups of MPCs, parameters γ_f, γ_s must be chosen for each group such that $\gamma_f, \gamma_s > 0,\ \gamma_f + \gamma_s = 1$. One simple choice for γ_f, γ_s is

$$\gamma_f = \frac{\sum_{i \in \mathcal{J}_{\text{fast}}} w_i}{\sum_i w_i} = \sum_{i \in \mathcal{J}_{\text{fast}}} w_i,\ \gamma_s = \frac{\sum_{i \in \mathcal{J}_{\text{slow}}} w_i}{\sum_i w_i} = \sum_{i \in \mathcal{J}_{\text{slow}}} w_i$$

11.2 FC-MPC optimization for asynchronous feedback

At time kT, let the set of initial subsystem states be $\mu([k,0])$. The initial input trajectory for subsystem $i \in \mathbb{I}_M$, $\boldsymbol{u}^k_i([k,0])$, is relayed to all subsystems' MPCs. The calculation of this initial input trajectory will be described in the sequel. The predicted state trajectory for subsystem $i \in \mathbb{I}_M$ due to the set of input trajectories $\boldsymbol{u}^k_1([k,0]), \ldots, \boldsymbol{u}^k_M([k,0])$ and system state $\mu([k,0])$ is $\widehat{\boldsymbol{x}}_i(\mu([k,0])) \leftarrow \boldsymbol{x}_i(\boldsymbol{u}^k_1([k,0]), \ldots, \boldsymbol{u}^k_M([k,0]); \mu([k,0]))$, where $\widehat{\boldsymbol{x}}_i(\mu([k,0])) = [\widehat{x}_i([k,0];[k,0])', \widehat{x}_i([k,1];[k,0])', \widehat{x}_i([k,2];[k,0])', \ldots]'$. This predicted subsystem state trajectory $\widehat{\boldsymbol{x}}_i(\mu([k,0])$ is broadcast all subsystems' MPCs. A forecast of the initial system state trajectory is known, therefore, to all MPCs $i \in \mathbb{I}_M$. During a sampling interval, each MPC computes an optimal input trajectory originating from the predicted system state at the end of the current sampling interval.

Slow MPCs (group $\mathcal{J}_{\text{slow}}$). Consider the sampling interval $[kT,(k+1)T)$, where $(k+1)T \equiv kT+\sigma t_f$. Each MPC $i \in \mathcal{J}_{\text{slow}}$ solves the following optimization problem

$$\boldsymbol{z}_i^1([k+1,0]) \in \arg\min_{\boldsymbol{z}_i} \quad \sum_{r=1}^{M} w_r \Phi_r \Big(\pi_{\text{fast}}^k([k,0],\sigma), \big[\boldsymbol{u}_{M_1+1}^k([k,0],\sigma),\ldots,\boldsymbol{z}_i,\ldots,\boldsymbol{u}_M^k([k,0],\sigma)\big], \; ; \; \widehat{\mu}([k,\sigma];[0,0]) \Big)$$

subject to

$$z_i(l) \le \Omega_i, \; 0 \le l \le N-1$$

$$z_i(l) = 0, \; N \le l$$

in which $\widehat{\mu} = [\widehat{x}_1,\ldots,\widehat{x}_M]$, $\boldsymbol{z}_i = [z_i(0)', z_i(1)', \ldots]'$. Let $\overline{\boldsymbol{u}}_i = [u_i(0)',\ldots,u_i(N-1)']'$, $\overline{\boldsymbol{z}}_i = [z_i(0)',\ldots,z_i(N-1)']'$. For $\phi_i(\cdot)$ for each $i \in \mathbb{I}_M$ given by Equation (4.5) (p. 34) and $\Phi_i(\cdot)$ obtained by eliminating the CM states x_i from $\phi_i(\cdot)$ using the subsystem CM (see p. 39), the FC-MPC optimization problem for subsystem $i \in \mathcal{J}_{\text{slow}}$ is

$$\overline{\boldsymbol{z}}_i^1([k+1,0]) \in \arg\min_{\overline{\boldsymbol{u}}_i} \quad \frac{1}{2}\overline{\boldsymbol{u}}_i'\mathfrak{R}_i\overline{\boldsymbol{u}}_i + \left(r_i([k,\sigma]) + \sum_{j\neq i}^{M} \mathcal{H}_{ij}\overline{\boldsymbol{v}}_j \right)' \overline{\boldsymbol{u}}_j + \text{constant} \tag{11.3a}$$

$$\text{subject to} \quad \overline{\boldsymbol{u}}_i \in \mathcal{U}_i, \tag{11.3b}$$

in which $\mathcal{U}_i = \Omega_i \times \ldots \times \Omega_i \in \mathbb{R}^{m_i N}$,

$$\mathbb{Q}_i = \text{diag}(Q_i(1),\ldots,Q_i(N-1),\overline{Q}_i), \; \mathbb{R}_i = \text{diag}(R_i(0),\ldots,R_i(N-1)),$$

$$\mathfrak{R}_i = w_i\mathbb{R}_i + w_i{E_{ii}}'\mathbb{Q}_i E_{ii} + \sum_{l\neq i}^{M} w_l{E_{li}}'\mathbb{Q}_l E_{li} \qquad \overline{\boldsymbol{v}}_j = \overline{\boldsymbol{u}}_j^k([k,0],\sigma),\ \forall\, j \in \mathbb{I}_M$$

$$\omega_j(k,\sigma) = \widehat{x}_j([k,\sigma];[k,0]),\ \forall\, j \in \mathbb{I}_M \qquad \mathcal{H}_{ij} = \sum_{l=1}^{M} w_l{E_{li}}'\mathbb{Q}_l E_{lj}$$

$$r_i([k,\sigma]) = w_i{E_{ii}}'\mathbb{Q}_i f_i\omega_i(k,\sigma) + \sum_{l\neq i}^{M} w_l{E_{li}}'\mathbb{Q}_l f_l\omega_l(k,\sigma)$$

with $\overline{Q}_j$ denoting an appropriately chosen terminal penalty. For stable systems, the terminal penalty is calculated using Theorem 4.1 (p. 49). For systems with unstable decentralized modes, a suitable terminal penalty is obtained using Theorem 4.2 (p. 53). The terminal decentralized state constraint described in Theorem 4.2 is necessary for closed-loop stability. By definition, $\boldsymbol{u}_i = [\overline{\boldsymbol{u}}_i', 0, 0, \ldots]$. The following algorithm is employed by the MPCs in $\mathcal{J}_{\text{slow}}$.

Algorithm 11.1. (Slow MPCs) Given $\overline{\boldsymbol{u}}_i^0([0,0]), \mathbb{Q}_i \geq 0, \mathbb{R}_i > 0,\ \forall\, i \in \mathcal{J}_{\text{slow}},$ and

$k, t \leftarrow 0,\ t_{\text{sim}} \leftarrow \Gamma, \Gamma \gg 0$

while $t \leq t_{\text{sim}}$

do $\forall\, i \in \mathcal{J}_{\text{slow}}$ during sampling interval $t \in \big[kT, (k+1)\,T\,\big)$

Determine $\overline{\boldsymbol{z}}_i^1([k+1,0])$ from Equation (11.3)

$\overline{\boldsymbol{u}}_i^{k+1}([k+1,0]) = \gamma_s\overline{\boldsymbol{z}}_i^1([k+1,0]) + (1-\gamma_s)\,\overline{\boldsymbol{u}}_i^k([k,0],\sigma)$

end (do)

if $t = (k+1)T$

Transmit $\overline{\boldsymbol{u}}_i^{k+1}([k+1,0]),\ \forall\, i \in \mathcal{J}_{\text{slow}}$ to each interconnected subsystem $j \in \mathbb{I}_M, j \neq i$

end (if)

$k \leftarrow k+1$

end (while)

Fast MPCs (group $\mathcal{J}_{\text{fast}}$). Consider the sampling interval $[kT+(\alpha-1)t_f, kT+\alpha t_f)$, in which $1 \leq \alpha \leq \sigma, \alpha \in \mathbb{I}_+$. During this sampling interval, each MPC $i \in \mathcal{J}_{\text{fast}}$ solves the following optimization problem to compute the optimal input trajectory at time $kT+\alpha t_f$.

$$\begin{aligned}\boldsymbol{\zeta}_i^{\beta}([k,\alpha]) \in \ \arg\ \min_{\boldsymbol{\zeta}_i} \quad & \sum_{r=1}^{M} w_r \Phi_r \Big(\big[\boldsymbol{z}_1^{\beta-1}([k,\alpha]), \ldots, \boldsymbol{\zeta}_i, \ldots, \boldsymbol{z}_{M_1}^{\beta-1}([k,\alpha])\big], \\ & \qquad \pi_{\text{slow}}^{k}([k,0],\alpha)\ ;\ \widetilde{\mu}([k,\alpha];[\alpha-1,0]) \Big) \\ \text{subject to} \quad & \zeta_i(l) \leq \Omega_i,\ 0 \leq l \leq N-1 \\ & \zeta_i(l) = 0,\ N \leq l\end{aligned}$$

in which $\boldsymbol{\zeta}_i = [\zeta_i(0)', \zeta_i(1)', \ldots]'$ and $\widetilde{\mu}$ denotes the set of subsystem states obtained if MPCs in $\mathcal{J}_{\text{fast}}$ are allowed to inject their calculated inputs without accounting for asynchronous feedback. The set of states $\widetilde{\mu}$ is generated by setting $\gamma_f = 1$ in the asynchronous feedback law (see Section 11.3.1, p. 284). Details for calculating $\widetilde{\mu}$ are given in p. 280 and p. 285. In the FC-MPC optimization problem for subsystem $i \in \mathcal{J}_{\text{fast}}$, the assumed input trajectories for subsystem $j \in \mathcal{J}_{\text{slow}}$ are obtained by shifting forward to time $kT+\alpha t_f$, the input trajectories calculated at time kT (synchronization iterate k). The input trajectories for subsystem $j \in \mathcal{J}_{\text{fast}}, j \neq i$ are held constant at values obtained at iterate $\beta-1$. The FC-MPC optimization for $i \in \mathcal{J}_{\text{slow}}$ is

$$\overline{\boldsymbol{\zeta}}_i^{\beta}([k,\alpha]) \in \ \arg\min_{\overline{\boldsymbol{u}}_i} \quad \frac{1}{2}\overline{\boldsymbol{u}}_i{}'\mathfrak{R}_i\overline{\boldsymbol{u}}_i + \left(r_i([k,\alpha]) + \sum_{j\neq i}^{M} \mathcal{H}_{ij}\overline{\boldsymbol{v}}_j\right)' \overline{\boldsymbol{u}}_j + \text{constant} \tag{11.4a}$$

$$\text{subject to} \quad \overline{\boldsymbol{u}}_i \in \mathcal{U}_i, \tag{11.4b}$$

in which $r_i([k,\alpha]) = w_i E_{ii}{}'\mathbb{Q}_i f_i \omega_i(k,\alpha) + \sum_{l\neq i}^{M} w_l E_{li}{}'\mathbb{Q}_l f_l \omega_l(k,\alpha)$,

$$\overline{\boldsymbol{v}}_j = \begin{cases} \overline{\boldsymbol{z}}_j^{\beta-1}([k,\alpha]), \text{ if } j \in \mathcal{J}_{\text{fast}}, \\ \overline{\boldsymbol{u}}_j^{k}([k,0],\alpha), \text{ if } j \in \mathcal{J}_{\text{slow}}. \end{cases} \qquad \omega_j(k,\alpha) = \begin{cases} \widetilde{x}_j([k,\alpha];[k,\alpha-1]), \text{ if } j \in \mathcal{J}_{\text{fast}}, \\ \widetilde{x}_j([k,\alpha];[k,0]), \text{ if } j \in \mathcal{J}_{\text{slow}}. \end{cases} \tag{11.5}$$

During the sampling interval $[kT + (\alpha-1)t_f, kT + \alpha t_f),\ 1 \leq \alpha \leq \sigma$, the following algorithm is employed by MPCs in $\mathcal{J}_{\text{fast}}$ to determine $\overline{\boldsymbol{z}}_i^{q_f}([k,\alpha]), \forall\, i \in \mathcal{J}_{\text{fast}}$.

Algorithm 11.2. (Fast-Inner) Given $\overline{\boldsymbol{z}}_i^{0}([k,\alpha]), q_f > 0$ and $\epsilon > 0$

$\beta \leftarrow 1, \kappa_i \leftarrow \Gamma\epsilon, \Gamma \gg 1$

while $\kappa_i > \epsilon$ for some $i \in \mathbb{I}_M$ and $\beta \leq q_f$

do $\forall\, i \in \mathcal{J}_{\text{fast}}$

Determine $\overline{\boldsymbol{\zeta}}_i^{\beta}$ from Equation (11.4)

$\overline{\boldsymbol{z}}_i^{\beta}([k,\alpha]) = w_i \overline{\boldsymbol{\zeta}}_i^{\beta}([k,\alpha]) + (1-w_i)\,\overline{\boldsymbol{z}}_i^{\beta-1}([k,\alpha])$

Transmit $\overline{\boldsymbol{z}}_i^{\beta}(\cdot)$ to each interconnected subsystem $j \neq i, j \in \mathcal{J}_{\text{fast}}$

$\kappa_i = \|\overline{\boldsymbol{z}}_i^{\beta} - \overline{\boldsymbol{z}}_i^{\beta-1}\|$

end (do)

$\beta \leftarrow \beta + 1$

end (while)

In Equation (11.5), $\omega_i([k,\alpha])$ is obtained as follows: For $i \in \mathcal{J}_{\text{fast}}$, $\widetilde{x}_i([k,\alpha];[k,\alpha-1])$ is calculated using the model equation

$$\widetilde{x}_i([k,\alpha];[k,\alpha-1]) = A_i\widetilde{x}_i([k,\alpha-1]) + B_i z_i^{q_f}([k,\alpha-1],0) + \sum_{j\in\mathcal{J}_{\text{fast}},j\neq i} W_{ij} z_j^{q_f}([k,\alpha-1],0) + \sum_{j\in\mathcal{J}_{\text{slow}}} W_{ij} u_j^k([k,0],\alpha-1) \quad (11.6)$$

with $\widetilde{x}_i([k,0]) = x_i(k) = \widehat{x}_i([k,0])$. The subsystem state $\widetilde{x}_i([k,\alpha]), i \in \mathcal{J}_{\text{fast}}$ is obtained when each MPC in $i \in \mathcal{J}_{\text{fast}}$ is allowed to inject $z_i^{q_f}([k,\alpha-1])$ without correcting for asynchronous feedback [1]. At time $kT+(\alpha-1)t_f$, the prior state $x_i([k,\alpha-1];[k,\alpha-2])$ is updated using the new measurement $y_i([k,\alpha-1])$. The updated state estimate $x_i([k,\alpha-1])$ is used to calculate $\widetilde{x}_i([k,\alpha-1])$. Details of this calculation are provided in Section 11.3.2.

Consider $j \in \mathcal{J}_{\text{slow}}$. During sampling interval $[kT,(k+1)T)$, we note that only interaction model states $\widetilde{x}_{ji}, i \in \mathcal{J}_{\text{fast}}$ deviate from their predicted state trajectory calculated at time kT. From Assumption 11.1, (A_{ji},B_{ji}), $\forall\, j \in \mathcal{J}_{\text{slow}}$ is known to MPC $i \in \mathcal{J}_{\text{fast}}$. Hence, $\widetilde{x}_{ji}$, $\forall\, j \in \mathcal{J}_{\text{slow}}$ is updated by MPC $i \in \mathcal{J}_{\text{fast}}$ using the interaction model equation

$$\widetilde{x}_{ji}([k,\alpha];[k,0]) = A_{ji}\widetilde{x}_{ji}([k,\alpha-1];[k,0]) + B_{ji} z_i^{q_f}([k,\alpha-1],0) \quad (11.7)$$

with $\widetilde{x}_{ji}([k,0]) = x_{ji}([k,0]) = \widehat{x}_{ji}([k,0])$. At each $kT+(\alpha-1)t_f$, $2 \le \alpha \le \sigma$, MPC $i \in \mathcal{J}_{\text{fast}}$ transmits calculated states $\widetilde{x}_i([k,\alpha];[k,\alpha-1])$ (Equation (11.6)) and $\widetilde{x}_{ji}([k,\alpha];[k,0])$, $j \in \mathcal{J}_{\text{slow}}$ (Equation (11.7)) to all MPCs $l \in \mathcal{J}_{\text{fast}}, l \neq i$. At this time, MPC $i \in \mathcal{J}_{\text{fast}}$ also receives $\widetilde{x}_l([k,\alpha];[k,\alpha-1])$ and $\widetilde{x}_{jl}([k,\alpha\ [k,0]), j \in \mathcal{J}_{\text{slow}}$ from each $l \in \mathcal{J}_{\text{fast}}, l \neq i$. The states $\widetilde{x}_{js}$ for

[1] In the asynchronous feedback control law (see Section 11.3.1), this corresponds to setting $\gamma_f = 1$.

$j, s \in \mathcal{J}_{\text{slow}}$ remain at the values predicted at time kT *i.e.,* $\widetilde{x}_{js}([k,\alpha];[k,0]) = \widehat{x}_{js}([k,\alpha];[k,0])$. Each MPC $i \in \mathcal{J}_{\text{fast}}$ can, therefore, reconstruct $\widetilde{\mu}([k,\alpha];[\alpha-1,0])$.

For $2 \leq \alpha \leq \sigma$, we define $\overline{\boldsymbol{z}}_i^0([k,\alpha]) = \overline{\boldsymbol{z}}_i^{q_f}([k,\alpha-1],1)$, a shifted version of the final input trajectory calculated during the sampling interval $[kT+(\alpha-2)t_f, kT+(\alpha-1)t_f)$. For $\alpha = 1$, Algorithm 11.2 is initialized as follows. During the sampling interval $[(k-1)T+(\sigma-1)t_f, kT)$, the input trajectory $\overline{\boldsymbol{z}}_i^{q_f}([k,0])\ (\equiv \overline{\boldsymbol{z}}_i^{q_f}([k-1,\sigma]))$, $\forall\, i \in \mathcal{J}_{\text{fast}}$ is calculated. The input trajectory $\overline{\boldsymbol{u}}_i^{k-1}([k-1,0])$, $i \in \mathcal{J}_{\text{fast}}$ was determined at the outer iterate $k-1$ at time $(k-1)T$ (see Algorithms 11.3 and 11.4). We define

$$\overline{\boldsymbol{u}}_i^k([k,0]) = \gamma_f \overline{\boldsymbol{z}}_i^{q_f}([k,0]) + (1-\gamma_f)\overline{\boldsymbol{u}}_i^{k-1}([k-1,0],\sigma).$$

For the sampling interval $[kT, kT+t_f)$, the input trajectory used for initializing Algorithm 11.2 is a shifted version of $\overline{\boldsymbol{u}}_i^k([k,0])$ and is defined as $\overline{\boldsymbol{z}}_i^0([k,1]) = \overline{\boldsymbol{u}}_i^k([k,0],1)$, $\forall\, i \in \mathcal{J}_{\text{fast}}$. This initialization strategy will prove useful to demonstrate nominal closed-loop stability under asynchronous feedback (Section 11.4). The following algorithm is used by the MPCs in $\mathcal{J}_{\text{fast}}$ during the sampling interval $[kT, (k+1)\,T\,), \forall\, k \geq 0$.

Algorithm 11.3. (Fast-Outer)

Given $k \geq 0, k \in \mathbb{I}_+, \overline{\boldsymbol{z}}_i^0([k,1]) = \overline{\boldsymbol{u}}_i^k([k,0],1)$, $\forall\, i \in \mathcal{J}_{\text{fast}}, \pi_{\text{slow}}^k, \alpha \leftarrow 1$.

while $\alpha \leq \sigma$

For sampling interval $[kT+(\alpha-1)t_f, kT+\alpha t_f)$

Execute Algorithm 11.2

$\overline{\boldsymbol{\Gamma}}_i([k,\alpha]) = \gamma_f \overline{\boldsymbol{z}}_i^{q_f}([k,\alpha]) + (1-\gamma_f)\,\overline{\boldsymbol{u}}_i^k([k,0],\alpha)$, $i \in \mathcal{J}_{\text{fast}}$

if $\alpha - \sigma < 0$

$\overline{\boldsymbol{z}}_i^0([k,\alpha+1]) = \overline{\boldsymbol{z}}_i^{q_f}([k,\alpha],1),\ \forall\, i \in \mathcal{J}_{\text{fast}}$

else if $\alpha - \sigma = 0$

$\overline{\boldsymbol{u}}_i^{k+1}([k+1,0]) = \overline{\boldsymbol{\Gamma}}_i([k,\sigma],1),\ \forall\, i \in \mathcal{J}_{\text{fast}}$

end (if)

end (for)

$\alpha \leftarrow \alpha + 1$

end (while)

In Algorithm 11.3, no information transfer across the two groups of MPCs is required. For all sampling intervals $[kT + (\alpha-1)t_f, kT + \alpha t_f)$, $1 \le \alpha \le \sigma - 1$, the information exchange occurs only among the MPCs in $\mathcal{J}_{\text{fast}}$ (see Algorithm 11.2). Let $\boldsymbol{z}_i^{\beta}(\cdot) = [\overline{\boldsymbol{z}}_i^{\beta}(\cdot)', 0, 0, \ldots]'$ and $\theta_{\text{fast}} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{M_1}]$. We have the following lemma.

Lemma 11.1. *The sequence of cost functions $\Phi(\theta_{\text{fast}}^{\beta}, \pi_{\text{slow}}^{k}; \mu(\cdot))$ is a nonincreasing function of the inner iteration number β.*

The proof is similar to the proof for Lemma 8.3 (p. 177), and is omitted for brevity.

Consider the sampling interval $[kT + (\alpha-1)t_f, kT + \alpha t_f)$. Using Lemma 11.1 for the nominal case gives,

$$\begin{aligned}
\Phi(\theta_{\text{fast}}^{q_f}([k,\alpha]), \pi_{\text{slow}}^{k}([k,0],\alpha); \mu([k,\alpha])) &\le \Phi(\theta_{\text{fast}}^{\beta}([k,\alpha]), \pi_{\text{slow}}^{k}([k,0],\alpha); \mu([k,\alpha])) \\
&\le \Phi(\theta_{\text{fast}}^{0}([k,\alpha]), \pi_{\text{slow}}^{k}([k,0],\alpha); \mu([k,\alpha])) \\
&= \Phi(\theta_{\text{fast}}^{q_f}([k,\alpha-1],1), \pi_{\text{slow}}^{k}([k,0],\alpha); \mu([k,\alpha])) \\
&= \Phi(\theta_{\text{fast}}^{q_f}([k,\alpha-1]), \pi_{\text{slow}}^{k}([k,0],\alpha-1); \mu([k,\alpha-1]) \\
&\quad - \mathbb{L}_{\alpha-1}(\mu([k,\alpha-1]), \widetilde{\Delta}_{\alpha-1}([k,\alpha-1])) \qquad (11.8)
\end{aligned}$$

in which $\widetilde{\Delta}_s = [\nu_1^{q_f}, \ldots, \nu_M^{q_f}]$,

$$\mathbb{L}_s(\mu([k,s]), \widetilde{\Delta}_s([k,s])) = \sum_{i=1}^{M} w_i L_i(x_i([k,s]), \nu_i^{q_f}([k,s])),$$

$\nu_i^{q_f}([k,s]) = z_i^{q_f}([k,s],0)$ if $i \in \mathcal{J}_{\text{fast}}$ and $\nu_i^{q_f}([k,s]) = u_i^k([k,0],s)$ if $i \in \mathcal{J}_{\text{slow}}$. Using Algorithms 11.2 and 11.3, we obtain the following algorithm for MPCs in $\mathcal{J}_{\text{fast}}$.

Algorithm 11.4. (Fast MPCs) Given $\overline{\boldsymbol{u}}_i^0([0,0]), \mathbb{Q}_i \geq 0, \mathbb{R}_i > 0, \ \forall\, i \in \mathcal{J}_{\text{fast}},$ and

$k, t \leftarrow 0, \ t_{\text{sim}} \leftarrow \Gamma, \Gamma \gg 0$

while $t \leq t_{\text{sim}}$

do $\forall\, i \in \mathcal{J}_{\text{fast}}$ during sampling interval $t \in \left[kT, (k+1)\,T\right)$

Execute Algorithm 11.3

end (do)

if $t = (k+1)T$

Transmit $\overline{\boldsymbol{u}}_i^{k+1}([k+1,0]), \ \forall\, i \in \mathcal{J}_{\text{fast}}$ to each interconnected subsystem $j \in \mathbb{I}_M, j \neq i$

end (if)

$k \leftarrow k+1$

end (while)

For initialization at time $k = 0$, we consider two cases. For stable systems, the zero input trajectory $\boldsymbol{u}_i^0([0,0]) = [0,0,\ldots], \forall\, i \in \mathbb{I}_M$ can be used to initialize Algorithms 11.1 and 11.4. Existence of a feasible input trajectory for each $i \in \mathbb{I}_M$ is assured for all future times (see Section 4.6.1, p. 48). For systems with unstable decentralized modes, the initialization procedure described in Section 4.6.2 is used. A quadratic program (QP) (see Equation (4.14), p. 51) is

solved by each MPC $i \in \mathbb{I}_M$ to determine a feasible initial input trajectory $\boldsymbol{u}_i^0([0,0])$. From Section 4.6.2 and Algorithms 11.1 to 11.4, it follows that feasibility of all the initialization QPs at $k=0$ guarantees existence of a feasible input trajectory for each $i \in \mathbb{I}_M$ at all future times.

11.3 Asynchronous feedback policies in FC-MPC

11.3.1 Asynchronous feedback control law

Let Assumption 11.2 be satisfied. The control law for each MPC $i \in \mathcal{J}_{\text{slow}}$ is defined as,

$$u_i^{\text{inj}}(t) = u_i^k([k,0],0),\ t \in [kT,(k+1)T)$$

For each MPC $i \in \mathcal{J}_{\text{fast}}$, the control law under asynchronous feedback is

$$u_i^{\text{inj}}(t) = \begin{cases} u_i^k([k,0],0),\ t \in [kT, kT+t_f), \\ \\ \Gamma_i([k,\alpha],0),\ t \in [kT+\alpha t_f, kT+(\alpha+1)t_f),\ 1 \leq \alpha \leq \sigma-1, \alpha \in \mathbb{I}_+ \end{cases}$$

in which $\Gamma_i([k,\alpha],0) = \gamma_f z_i^{q_f}([k,\alpha],0) + (1-\gamma_f)u_i^k([k,0],\alpha)$. Notice that to generate the states $\widetilde{\mu}(\cdot)$, we set $\gamma_f = 1$ at each t, and consequently, $u_i(t) = z_i^{q_f}([k,\alpha],0), t \in [kT+\alpha t_f, kT+(\alpha+1)t_f)$, $i \in \mathcal{J}_{\text{fast}}$ and $1 \leq \alpha \leq \sigma, \alpha \in \mathbb{I}_+$. Similarly to generate the states $\widehat{\mu}(\cdot)$, we choose $\gamma_f = 0$ at each t. For $i \in \mathcal{J}_{\text{fast}}$, the choice $\gamma_f = 0$ implies $u_i(t) = u_i^k([k,0],0),\ t \in [kT,(k+1)T)$.

11.3.2 Implementation

To implement asynchronous feedback FC-MPC, a procedure for estimating the state when measurements become available is necessary. The measurements for MPCs in $\mathcal{J}_{\text{fast}}$ are avail-

able every t_f time instants while the measurements for MPCs in $\mathcal{J}_{\text{slow}}$ are available only every kT time instants.

Fast MPCs. Consider the sampling interval $[kT+(\alpha-1)t_f, kT+\alpha t_f)$. At time $kT+(\alpha-1)t_f$, measurements $y_i([k,\alpha-1]),\ \forall\ i\ \in\ \mathcal{J}_{\text{fast}}$ become available. This new measurement at time $kT+(\alpha-1)t_f$ can be used to calculate the state estimate $x_i([k,\alpha-1]),\ \forall\ i\ \in\ \mathcal{J}_{\text{fast}}$. Using steady-state estimators with gain $\mathcal{L}_i,\ i \in \mathbb{I}_M$, we have

$$x_i([k,\alpha-1]) = x_i([k,\alpha-1];[k,\alpha-2]) + \mathcal{L}_i\left(y_i([k,\alpha-1] - C_i x_i([k,\alpha-1];[k,\alpha-2])\right),\ i \in \mathcal{J}_{\text{fast}}$$

An estimate of $\widetilde{x}_i([k,\alpha]),\ i \in \mathbb{I}_M$ is required to solve the FC-MPC optimization problem (Equation (11.4)) during the sampling interval $[kT+(\alpha-1)t_f, kT+\alpha t_f)$. For $i\ \in\ \mathcal{J}_{\text{fast}}$, the state estimate $\widetilde{x}_i([k,\alpha];[k,\alpha-1])$ is obtained as follows. A revised estimate (using latest measurement) of $\widetilde{x}_i([k,\alpha-1])$ is required first. The subsystem state $\widetilde{x}_i([k,\alpha-1])$ is obtained if all the MPCs in $\mathcal{J}_{\text{fast}}$ are allowed to choose $\gamma_f\ =\ 1$ for their respective asynchronous feedback laws. From the definition of the asynchronous feedback control law (Section 11.3.1), we have $x_i = \gamma_f\widetilde{x}_i + \gamma_s\widehat{x}_i,\ i \in \mathcal{J}_{\text{fast}}$. At time $kT+(\alpha-1)t_f$, $\widetilde{x}_i([k,\alpha-1];[k,\alpha-2])$ is known. The state estimate $\widetilde{x}_i([k,\alpha-1])$ is obtained using

$$\widetilde{x}_i([k,\alpha-1]) = \widetilde{x}_i([k,\alpha-1];[k,\alpha-2]) + \frac{1}{\gamma_f}\left[x_i([k,\alpha-1]) - x_i([k,\alpha-1];[k,\alpha-2])\right], \quad (11.9)$$

$\forall\, i \in \mathcal{J}_{\text{fast}}$. In the nominal case, Equation (11.9) reduces to $\widetilde{x}_i([k,\alpha-1]) = \widetilde{x}_i([k,\alpha-1];[k,\alpha-2]), i \in \mathcal{J}_{\text{fast}}$. It is known that MPCs in $\mathcal{J}_{\text{slow}}$ do not deviate from their policy $\overline{\boldsymbol{u}}_j^k([k,0]), j \in \mathcal{J}_{\text{slow}}$ (calculated at time kT), at least until time $(k+1)T$. Also, from Algorithm 11.2, $\overline{\boldsymbol{z}}_j^{q_f}([k,\alpha-$

$1]),\ \forall\, j \in \mathcal{J}_{\text{fast}},\ j \neq i$, calculated at the previous sampling interval, is known to MPC $i \in \mathcal{J}_{\text{fast}}$. The subsystem state $\widetilde{x}_i([k,\alpha];[k,\alpha-1]), i \in \mathcal{J}_{\text{fast}}$ is calculated using Equation (11.6). For MPC $i \in \mathcal{J}_{\text{fast}}$, the states $\widetilde{x}_{ji}([k,\alpha];[k,0]), j \in \mathcal{J}_{\text{slow}}$ are obtained using Equation (11.7). The states $\widetilde{x}_l([k,\alpha];[k,\alpha-1])$ and $\widetilde{x}_{jl}([k,\alpha];[k,0]),\ j \in \mathcal{J}_{\text{slow}}$ are transmitted by all MPCs $l \in \mathcal{J}_{\text{fast}}$ to MPC $i \in \mathcal{J}_{\text{fast}}$. We also note that $\widetilde{x}_{js}([k,\alpha];[k,0] = \widehat{x}_{js}([k,\alpha];[k,0]),\ j,s \in \mathcal{J}_{\text{slow}}$.

Slow MPCs. At time $(k+1)T$, measurements $y_j([k+1,0]), \forall j \in \mathcal{J}_{\text{slow}}$ become available. During the sampling interval $[kT,(k+1)T)$, MPCs in $\mathcal{J}_{\text{fast}}$ recalculate their predicted input trajectories and furthermore, introduce control actions into their respective subsystems. The control actions initiated by MPCs in $\mathcal{J}_{\text{fast}}$ during the sampling interval $[kT,(k+1)T)$ are unknown to MPCs in $\mathcal{J}_{\text{slow}}$. To obtain an estimate of the subsystem state $x_j([k+1,0])\ (\equiv x_j([k,\sigma])), j \in \mathcal{J}_{\text{slow}}$, we require an estimate of x_j prior to the measurement at $(k+1)T$. This prior state estimate must account for all control actions introduced by MPCs $i \in \mathcal{J}_{\text{fast}}$ during $[kT,(k+1)T)$. Since each MPC $i \in \mathcal{J}_{\text{fast}}$ maintains an estimate of $\widetilde{x}_{ji}, \forall j \in \mathcal{J}_{\text{slow}}$, the state $\widetilde{x}_{ji}([k,\sigma];[k,0]), \forall j \in \mathcal{J}_{\text{slow}}$ is known to MPC $i \in \mathcal{J}_{\text{fast}}$. At time $(k+1)T$, each MPC $i \in \mathcal{J}_{\text{fast}}$ transmits its estimate of $\widetilde{x}_{ji}([k,\sigma];[k,0])$ to MPC $j \in \mathcal{J}_{\text{slow}}$. Noting that $\widetilde{x}_{js}([k,\sigma];[k,0]) = \widehat{x}_{js}([k,\sigma];[k,0]),\ \forall\, s \in \mathcal{J}_{\text{slow}}$, the state $\widetilde{x}_j([k,\sigma];[k,0])$ can be reconstructed by each subsystem $j \in \mathcal{J}_{\text{slow}}$. From the definition of the asynchronous feedback control law, we have

$$x_j([k,\sigma];[k,0]) = \gamma_f \widetilde{x}_j([k,\sigma];[k,0]) + \gamma_s \widehat{x}_j([k,\sigma];[k,0]) \tag{11.10}$$

The RHS of Equation (11.10) is known; $x_j([k,\sigma];[k,0]), \forall j \in \mathcal{J}_{\text{slow}}$ can, therefore, be calculated. Using steady-state estimators with gain $\mathcal{L}_j, j \in \mathcal{J}_{\text{slow}}$, and in the absence of any additional

measurements, we have

$$x_j([k+1,0]) = x_j([k,\sigma];[k,0]) + \mathcal{L}_j\left(y_j([k+1,0]) - C_j x_j([k,\sigma];[k,0])\right),\ \forall\, j \in \mathcal{J}_{\text{slow}}$$

An alternative strategy to obtain $x_j([k,\sigma];[k,0])$ is as follows: At time $(k+1)T$, MPCs $i \in \mathcal{J}_{\text{fast}}$ transmit a history of all introduced control actions within the sampling interval $[kT,(k+1)T)$ to MPCs $j \in \mathcal{J}_{\text{slow}}$. The state $x_j([k,\sigma];[k,0])$ is then calculated using the recursion

$$\begin{aligned} x_j([k,\alpha];[k,0]) = A_j x_j([k,\alpha-1];[k,0]) + B_j u_j^{\text{inj}}(kT+(\alpha-1)t_f) \\ + \sum_{s\in\mathcal{J}_{\text{slow}},s\neq j} W_{js}u_s^{\text{inj}}(kT+(\alpha-1)t_f) + \sum_{l\in\mathcal{J}_{\text{fast}}} W_{jl}u_l^{\text{inj}}(kT+(\alpha-1)t_f) \end{aligned}$$

in which $1 \leq \alpha \leq \sigma$.

At time $(k+1)T$, the state estimate $x_i([k+1,0]),\ \forall\, i \in \mathbb{I}_M$ is relayed to all subsystems $j \in \mathbb{I}_M, j \neq i$. This sequence of operations is repeated during subsequent sampling intervals.

11.3.3 An illustrative case study

The procedure for asynchronous feedback is illustrated using a simple example in which $\mathcal{J}_{\text{fast}} = \{1,2\}$, $\mathcal{J}_{\text{slow}} = \{3\}$ and $\sigma = 3$. The notation $'\text{cost}(AEF)'$ represents the cost along the trajectory AEF. The various sampling intervals are examined.

$t \in [kT, kT+t_f)$: During this sampling interval, the input injected into each subsystem is

$$u_i^{\text{inj}}(t) = u_i^k([k,0],0),\ i = 1,2,3$$

MPCs 1 and 2 in $\mathcal{J}_{\text{fast}}$ determine $\boldsymbol{z}_i^{q_f}([k,1]),\ i=1,2$ such that

$$\Phi(\boldsymbol{z}_1^{q_f}([k,1]),\boldsymbol{z}_2^{q_f}([k,1]),\boldsymbol{u}_3^k([k,0],1);\widetilde{\mu}([k,1];[0,0]))$$
$$\leq \Phi(\boldsymbol{u}_1^k([k,0],1),\boldsymbol{u}_2^k([k,0],1),\boldsymbol{u}_3^k([k,0],1);\widetilde{\mu}([k,1];[0,0]))$$

in which $\widetilde{x}_i([k,1];[k,0])=\widehat{x}_i([k,1];[k,0])=x_i([k,1];[k,0]),\ i=1,2,3$. For the nominal case, we have $x_i([k,1];[k,0])=x_i([k,1]),\ i=1,2,3$. The cost relationship for the nominal system is

$$\Phi(\boldsymbol{z}_1^{q_f}([k,1]),\boldsymbol{z}_2^{q_f}([k,1]),\boldsymbol{u}_3^k([k,0],1);\widetilde{\mu}([k,1]))$$
$$\leq \Phi(\boldsymbol{u}_1^k([k,0],1),\boldsymbol{u}_2^k([k,0],1),\boldsymbol{u}_3^k([k,0],1);\widetilde{\mu}([k,1])) \quad (11.11)$$

In Figure 11.2, Equation (11.11) implies that $\text{cost}(EIX)\leq\text{cost}(BCJW)$, where BCJW is the shifted version of the initial trajectory obtained at time kT (ABCJW). During this sampling interval, the nominal closed-loop system moves from a to b in Figure 11.2.

$t\in[kT+t_f,kT+2t_f)$: The input injected into the plant during this sampling interval is

$$u_i^{\text{inj}}(t)=\gamma_f z_i^{q_f}([k,1],0)+\gamma_s u_i^k([k,0],1),\ i=1,2$$
$$u_3^{\text{inj}}(t)=u_3^k([k,0],1)$$

At time $kT+t_f$, measurements $y_i([k,1]),i=1,2$ are available. The state estimators for subsystems 1 and 2 use this new measurement to obtain updated estimates $x_i([k,1]),i=1,2$. MPCs 1 and 2 use Equation (11.9) with $\alpha=1$ to obtain $\widetilde{x}_i([k,1]),i=1,2$. The predicted state $\widetilde{x}_i([k,2];[k,1]),i=1,2$ is calculated using Equation (11.6). Interaction model (A_{31},B_{31}) is

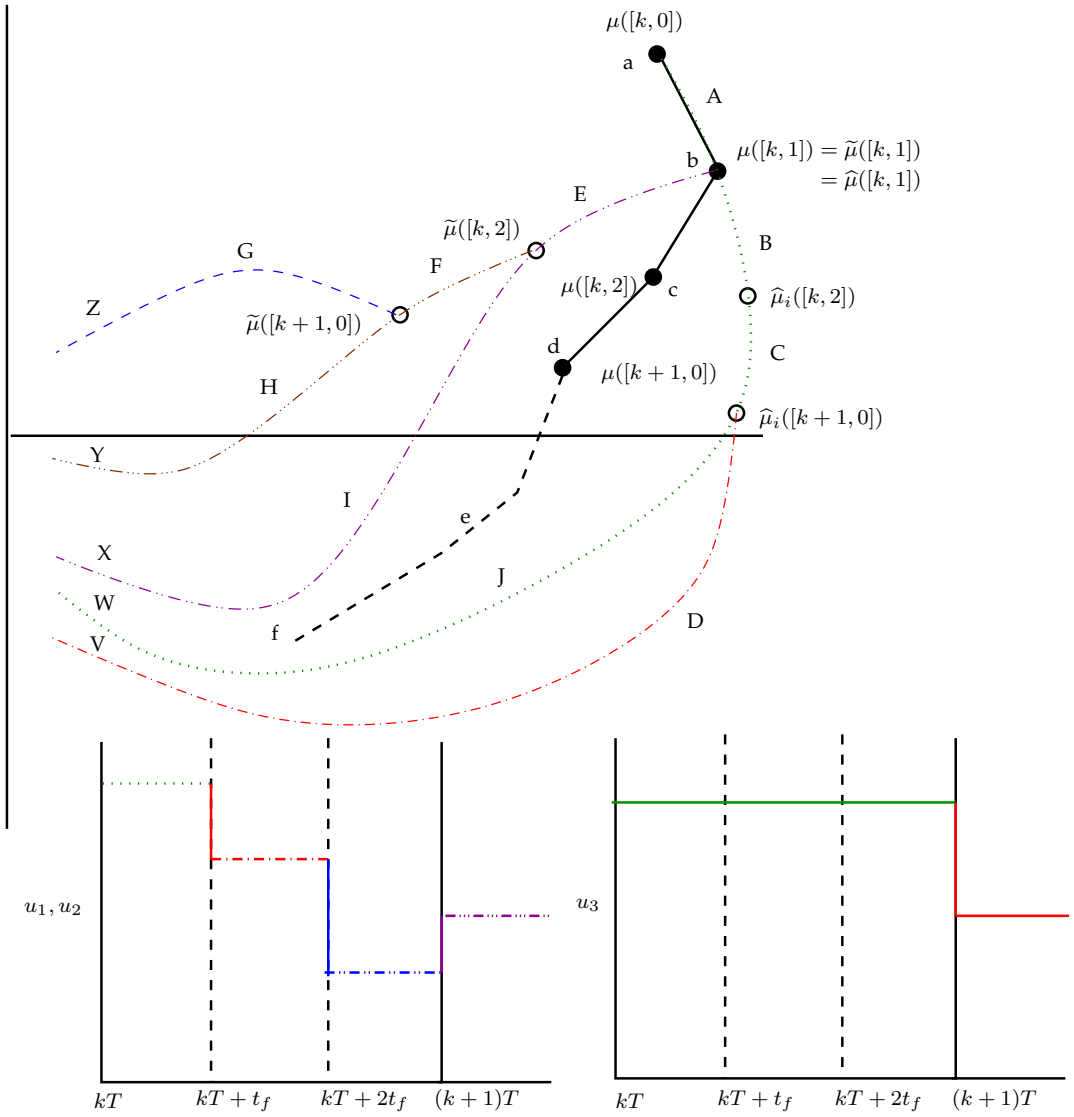

Figure 11.2: Nominal closed-loop state trajectories for asynchronous feedback FC-MPC.

known to MPC 1; $\widetilde{x}_{31}([k,2];[k,0])$ is calculated using Equation (11.7) with $\alpha = 1$. Similarly, using the interaction model (A_{32}, B_{32}) (known to MPC 2), $\widetilde{x}_{32}([k,2];[k,0])$ is calculated by MPC 2. MPC 1 transmits $\widetilde{x}_1([k,2];[k,1])$ and $\widetilde{x}_{31}([k,2];[k,0])$ to MPC 2 and receives $\widetilde{x}_2([k,2];[k,1])$ and $\widetilde{x}_{32}([k,2];[k,0])$ from MPC 2. Since, $\widetilde{x}_{33}([k,2];[k,0]) = \widehat{x}_{33}([k,2];[k,0])$ and $\widehat{\mu}([k,0])$ is known to all MPCs, both MPCs 1 and 2 can reconstruct $\widetilde{\mu}([k,2];[1,0])$.

During this sampling interval, MPCs 1 and 2 compute $\boldsymbol{z}_1^{q_f}([k,2]), \boldsymbol{z}_2^{q_f}([k,2])$ respectively such that

$$\Phi(\boldsymbol{z}_1^{q_f}([k,2]), \boldsymbol{z}_2^{q_f}([k,2]), \boldsymbol{u}_3^k([k,0],2); \widetilde{\mu}([k,2];[1,0])) \\ \leq \Phi(\boldsymbol{z}_1^{q_f}([k,1],1), \boldsymbol{z}_2^{q_f}([k,1],1), \boldsymbol{u}_3^k([k,0],2); \widetilde{\mu}([k,2];[1,0]))$$

For the nominal case, we have

$$\Phi(\boldsymbol{z}_1^{q_f}([k,2]), \boldsymbol{z}_2^{q_f}([k,2]), \boldsymbol{u}_3^k([k,0],2); \widetilde{\mu}([k,2])) \\ \leq \Phi(\boldsymbol{z}_1^{q_f}([k,1],1), \boldsymbol{z}_2^{q_f}([k,1],1), \boldsymbol{u}_3^k([k,0],2); \widetilde{\mu}([k,2])) \quad (11.12)$$

In Figure 11.2, Equation (11.12) implies that $\text{cost}(FHY) \leq \text{cost}(IX)$. The nominal closed-loop path for the system during this sampling interval is from b to c.

$t \in [kT + 2t_f, (k+1)T)$**:** The input injected into the plant during this sampling interval is

$$u_i^{\text{inj}}(t) = \gamma_f z_i^{q_f}([k,2],0) + \gamma_s u_i^k([k,0],2),\ i = 1,2$$
$$u_3^{\text{inj}}(t) = u_3^k([k,0],2)$$

At time $kT + 2t_f$, measurements $y_i([k,2]), i = 1,2$ are available. The state estimators for subsystems 1 and 2 use these new measurements to estimate the state $x_i([k,2]), i = 1,2$. An identical procedure to the one described for the earlier sampling interval is used to determine $\widetilde{x}_i([k+1,0];[k,2]), i = 1,2$ and $\widetilde{x}_3([k+1,0];[k,0])$. During this sampling interval, MPCs 1 and

2 calculate $\boldsymbol{z}_1^{q_f}([k+1,0])$ and $\boldsymbol{z}_2^{q_f}([k+1,0])$ respectively such that

$$\Phi(\boldsymbol{z}_1^{q_f}([k+1,0]), \boldsymbol{z}_2^{q_f}([k+1,0]), \boldsymbol{u}_3^k([k,0],3); \widetilde{\mu}([k,3];[2,0]))$$
$$\leq \Phi(\boldsymbol{z}_1^{q_f}([k,2],1), \boldsymbol{z}_2^{q_f}([k,2],1), \boldsymbol{u}_3^k([k,0],3); \widetilde{\mu}([k,3];[2,0]))$$

For the nominal case, the cost relationship above reduces to

$$\Phi(\boldsymbol{z}_1^{q_f}([k+1,0]), \boldsymbol{z}_2^{q_f}([k+1,0]), \boldsymbol{u}_3^k([k,0],3); \widetilde{\mu}([k,3]))$$
$$\leq \Phi(\boldsymbol{z}_1^{q_f}([k,2],1), \boldsymbol{z}_2^{q_f}([k,2],1), \boldsymbol{u}_3^k([k,0],3); \widetilde{\mu}([k,3])) \quad (11.13)$$

In Figure 11.2, we have from Equation (11.13) that the $\text{cost}(GZ) \leq \text{cost}(HY)$. During this sampling interval, MPC 3 completes computation of $\boldsymbol{z}_3^{q_f}([k+1,0])$ satisfying

$$\Phi(\boldsymbol{u}_1^k([k,0],3), \boldsymbol{u}_2^k([k,0],3), \boldsymbol{z}_3^{q_f}([k+1,0]); \widehat{\mu}([k,3];[0,0]))$$
$$\leq \Phi(\boldsymbol{u}_1^k([k,0],3), \boldsymbol{u}_2^k([k,0],3), \boldsymbol{u}_3^k([k,0],3); \widehat{\mu}([k,3];[0,0]))$$

For the nominal case, we have

$$\Phi(\boldsymbol{u}_1^k([k,0],3), \boldsymbol{u}_2^k([k,0],3), \boldsymbol{z}_3^{q_f}([k+1,0]); \widehat{\mu}([k,3]))$$
$$\leq \Phi(\boldsymbol{u}_1^k([k,0],3), \boldsymbol{u}_2^k([k,0],3), \boldsymbol{u}_3^k([k,0],3); \widehat{\mu}([k,3])) \quad (11.14)$$

From Equation (11.14), we have in Figure 11.2 that $\text{cost}(DV) \leq \text{cost}(JW)$. The nominal closed-loop path during this sampling interval is from c to d.

At time $t = (k+1)T = kT + 3t_f$, an (outer) synchronization iterate is performed. Measurements $y_i([k+1,0]), i = 1,2,3$ are available. MPCs 1 and 2 use the corresponding measurement $y_i([k+1,0])$ and prior estimate $x_i([k+1,0];[k,\sigma-1])$ to calculate $x_i([k+1,0]), i = 1,2$. To determine $x_3([k+1,0])$, MPCs in $\mathcal{J}_{\text{fast}}$ transmit their calculated estimate of $\widetilde{x}_{3i}([k+1,0];[k,0]), i = 1,2$ to MPC 3. The prior estimate $x_3([k+1,0];[k,0])$ is then obtained using Equation (11.10). Using $y_3([k+1,0])$ and the estimate $x_3([k+1,0];[k,0])$, $x_3([k+1,0])$ is calculated. MPCs 1, 2 and 3 relay their respective state estimates $x_i([k+1,0]), i = 1,2,3$ to each other. From Algorithms 11.1 and 11.4, we have

$$\boldsymbol{u}_i^{k+1}([k+1,0]) = \gamma_f \boldsymbol{z}_i^{q_f}([k+1,0]) + \gamma_s \boldsymbol{u}_i^k([k,0],3),\ i = 1,2$$

$$\boldsymbol{u}_3^{k+1}([k+1,0]) = \gamma_s \boldsymbol{z}_3^{q_f}([k+1,0]) + \gamma_f \boldsymbol{u}_3^k([k,0],3)$$

An outline of a proof for nominal closed-loop stability under asynchronous feedback is now provided. A formal argument is presented in Section 11.4. In Figure 11.2, we have using arguments presented above that $\text{cost}(AEFGZ) \leq \text{cost}(ABCDJW)$ and $\text{cost}(ABCDV) \leq \text{cost}(ABCJW)$. From Figure 11.2, the cost to go along the initialized nominal closed-loop path at $(k+1)T$ *i.e.*, $\text{cost}(def)$ is equal to $\text{cost}(abcdef) - S(a) - S(b) - S(c)$ in which the notation $S(x)$ denote the net stage cost at point x and $S(x) \geq 0$. From the definition of the asynchronous feedback law, and using convexity we have $\text{cost}(abcdef) = \text{cost}(def) + S(a) + S(b) + S(c) \leq \gamma_f \text{cost}(AEFGZ) + \gamma_s \text{cost}(ABCDV) \leq \text{cost}(ABCJW)$. Hence,

$$\text{cost}(def) \leq \text{cost}(ABCJW) - [S(a) + S(b) + S(c)]$$

Nominal asymptotic stability follows.

11.4 Nominal closed-loop stability with asynchronous feedback policies

Define

$$\boldsymbol{w}_i([k,0]) = [u_i^k([k,0],0)', z_i^{q_f}([k,1],0)', \ldots, z_i^{q_f}([k,\sigma-1],0)', \boldsymbol{z}_i^{q_f}([k+1,0])']', \ \forall\, i \in \mathcal{J}_{\text{fast}}$$

and

$$\boldsymbol{w}_i([k,0]) = [u_i^k([k,0],0)', u_i^k([k,0],1)', \ldots, u_i^k([k,0],\sigma-1)', \boldsymbol{z}_i^{q_f}([k+1,0])']', \ \forall\, i \in \mathcal{J}_{\text{slow}}$$

From Assumption 11.2, we have $u_i^k([k,0],0) = u_i^k([k,0],1) = \ldots = u_i^k([k,0],\sigma-1), \ \forall\, i \in \mathcal{J}_{\text{slow}}$. Let

$$\boldsymbol{v}_i([k,0]) = [u_i^{\text{inj}}(kT)', u_i^{\text{inj}}(kT+t_f)', \ldots, u_i^{\text{inj}}(kT+(\sigma-1)t_f)', \boldsymbol{u}_i^{k+1}([k+1,0])']', \ \forall\, i \in \mathbb{I}_M$$

in which $u_i^{\text{inj}}(kT) = u_i^k([k,0],0)$. From the definition of the asynchronous feedback policy in Section 11.3, and from Algorithms 11.1 and 11.3, we have

$$\boldsymbol{v}_i([k,0]) = \gamma_f \boldsymbol{w}_i([k,0]) + \gamma_s \boldsymbol{u}_i^k([k,0]), \ \forall\, i \in \mathcal{J}_{\text{fast}}$$

and

$$\boldsymbol{v}_i([k,0]) = \gamma_s \boldsymbol{w}_i([k,0]) + \gamma_f \boldsymbol{u}_i^k([k,0]), \ \forall\, i \in \mathcal{J}_{\text{slow}}$$

With slight abuse of notation, we define

$$\xi^v_{\text{fast}}([k,0]) = [\boldsymbol{v}_1([k,0]), \ldots, \boldsymbol{v}_{M_1}([k,0])], \quad \xi^v_{\text{slow}}([k,0]) = [\boldsymbol{v}_{M_1+1}([k,0]), \ldots, \boldsymbol{v}_M([k,0])]$$

$$\xi^w_{\text{fast}}([k,0]) = [\boldsymbol{w}_1([k,0]), \ldots, \boldsymbol{w}_{M_1}([k,0])], \quad \xi^w_{\text{slow}}([k,0]) = [\boldsymbol{w}_{M_1+1}([k,0]), \ldots, \boldsymbol{w}_M([k,0])]$$

and write

$$\Phi([\boldsymbol{v}_1([k,0]), \ldots, \boldsymbol{v}_{M_1}([k,0])], [\boldsymbol{v}_{M_1+1}([k,0]), \ldots, \boldsymbol{v}_M([k,0])]; \mu([k,0])) \\ = \Phi(\xi^v_{\text{fast}}([k,0]), \xi^v_{\text{slow}}([k,0]); \mu([k,0])$$

Using convexity of $\Phi(\cdot)$ gives,

$$\Phi(\xi^v_{\text{fast}}([k,0]), \xi^v_{\text{slow}}([k,0]); \mu([k,0]) \leq \gamma_f \Phi(\xi^w_{\text{fast}}([k,0]), \pi^k_{\text{slow}}([k,0]; \mu([k,0])) \\ + \gamma_s \Phi(\pi^k_{\text{fast}}([k,0]), \xi^w_{\text{slow}}([k,0]); \mu([k,0]) \tag{11.15}$$

in which π_{fast} and π_{slow} are defined in Section 11.1 (p. 275). Consider the sampling interval $[kT, kT + t_f)$. From Algorithms 11.2 to 11.4, we have for group $\mathcal{J}_{\text{fast}}$ that,

$$\Phi(\theta^{q_f}_{\text{fast}}([k,1]), \pi^k_{\text{slow}}([k,0],1); \mu([k,1])) \leq \Phi(\pi^k_{\text{fast}}([k,0],1), \pi^k_{\text{slow}}([k,0],1); \mu([k,1])) \\ = \Phi(\pi^k_{\text{fast}}([k,0]), \pi^k_{\text{slow}}([k,0]); \mu([k,0]) \\ - \mathbb{L}_0(\mu([k,0]), \widetilde{\Delta}_0([k,0])) \tag{11.16}$$

in which θ_{fast} is defined in Section 11.2 (p. 282), $\widetilde{\Delta}_0([k,0]) = [u_1^k([k,0],0),\ldots,u_M^k([k,0],0)]$ and

$$\mathbb{L}_0(\mu([k,0],\widetilde{\Delta}_0([k,0])) = \sum_{i=1}^{M} w_i L_i(x_i([k,0]),u_i^k([k,0],0).$$

For the sampling interval $[kT+t_f, kT+2t_f)$, we have

$$\begin{aligned}
\Phi(\theta_{\text{fast}}^{q_f}([k,2]),\pi_{\text{slow}}^k([k,0],2);\mu([k,2])) &\leq \Phi(\theta_{\text{fast}}^k([k,1],1),\pi_{\text{slow}}^k([k,0],2);\mu([k,2])) \\
&= \Phi(\theta_{\text{fast}}^k([k,1]),\pi_{\text{slow}}^k([k,0],1);\mu([k,1]) \\
&\quad - \mathbb{L}_1(\mu([k,1]),\widetilde{\Delta}_1([k,1])) \\
&\leq \Phi(\pi_{\text{fast}}^k([k,0]),\pi_{\text{slow}}^k([k,0]);\mu([k,0]) \\
&\quad - \sum_{\alpha=0}^{1} \mathbb{L}_\alpha(\mu([k,\alpha]),\widetilde{\Delta}_\alpha([k,\alpha])) \\
&\quad \text{(from Equation (11.16))}
\end{aligned}$$

in which

$$\widetilde{\Delta}_1([k,1]) = [z_1^{q_f}([k,1],0),\ldots,z_{M_1}^{q_f}([k,1],0),u_{M_1+1}^k([k,0],1),\ldots,u_M^k([k,0],1)].$$

Proceeding recursively up to and including sampling interval $[kT+(\sigma-1)t_f,(k+1)T)$ gives

$$\begin{aligned}
\Phi(\theta_{\text{fast}}^{q_f}([k+1,0]),\pi_{\text{slow}}^k([k,0],\sigma);\mu([k+1,0])) &\leq \Phi(\pi_{\text{fast}}^k([k,0]),\pi_{\text{slow}}^k([k,0]);\mu([k,0]) \\
&\quad - \sum_{\alpha=0}^{\sigma-1} \mathbb{L}_\alpha(\mu([k,\alpha]),\widetilde{\Delta}_\alpha([k,\alpha])) \qquad (11.17)
\end{aligned}$$

in which

$$\widetilde{\Delta}_\alpha([k,\alpha]) = [z_1^{q_f}([k,\alpha],0),\ldots,z_{M_1}^{q_f}([k,\alpha],0),u_{M_1+1}^k([k,0],\alpha),\ldots,u_M^k([k,0],\alpha)].$$

Using the definition of $\boldsymbol{w}_i([k,0]),\ i\in\mathcal{J}_{\text{fast}}$ gives,

$$\Phi(\xi_{\text{fast}}^w([k,0]),\pi_{\text{slow}}^k([k,0]);\mu([k,0])) \le \Phi(\pi_{\text{fast}}^k([k,0]),\pi_{\text{slow}}^k([k,0]);\mu([k,0])) \tag{11.18}$$

Similarly for group $\mathcal{J}_{\text{slow}}$, we have

$$\begin{aligned}\Phi(\pi_{\text{fast}}^k([k,0],\sigma),\theta_{\text{slow}}^{q_f}([k+1,0]);\mu([k+1,0])) \le \Phi(\pi_{\text{fast}}^k([k,0]),\pi_{\text{slow}}^k([k,0]);\mu([k,0])\\ -\sum_{\alpha=0}^{\sigma-1}\mathbb{L}_\alpha(\mu([k,\alpha]),\widehat{\Delta}_\alpha([k,\alpha]))\end{aligned} \tag{11.19}$$

in which

$$\widehat{\Delta}_\alpha([k,\alpha]) = [u_1^k([k,0],\alpha),\ldots,u_M^k([k,0],\alpha)],\ 0\le\alpha\le\sigma-1$$

Hence,

$$\Phi(\pi_{\text{fast}}^k([k,0]),\xi_{\text{slow}}^w([k,0]);\mu([k,0])) \le \Phi(\pi_{\text{fast}}^k([k,0]),\pi_{\text{slow}}^k([k,0]);\mu([k,0])) \tag{11.20}$$

For the nominal case, we have from the definitions of $\boldsymbol{v}_i$ and $\boldsymbol{u}_i^{k+1}, i\in\mathbb{I}_M$ that,

$$\begin{aligned}\Phi(\xi_{\text{fast}}^v([k,0]),\xi_{\text{slow}}^v([k,0]);\mu([k,0])) = \Phi(\pi_{\text{fast}}^{k+1}([k+1,0]),\pi_{\text{slow}}^{k+1}([k+1,0]);\mu([k+1,0]))\\ +\sum_{\alpha=0}^{\sigma-1}\mathbb{L}_\alpha(\mu([k,\alpha]),\Delta_\alpha^v([k,\alpha]))\end{aligned} \tag{11.21}$$

in which

$$\Delta_{\alpha}^{v}([k,\alpha]) = [u_1^{\text{inj}}([k,\alpha]), \ldots, u_M^{\text{inj}}([k,\alpha])]$$

From Equations (11.15), (11.18), (11.20) and (11.21) we have,

$$0 \leq \Phi(\pi_{\text{fast}}^{k+1}([k+1,0]), \pi_{\text{slow}}^{k+1}([k+1,0]); \mu([k+1,0])) \leq \Phi(\pi_{\text{fast}}^{k}([k,0]), \pi_{\text{slow}}^{k}([k,0]); \mu([k,0])) - \sum_{\alpha=0}^{\sigma-1} \mathbb{L}_{\alpha}(\mu([k,\alpha]), \Delta_{\alpha}^{v}([k,\alpha])) \tag{11.22}$$

Equation (11.22) gives nominal asymptotic stability for the closed-loop system under the prescribed asynchronous feedback policy.

11.5 Example

A plant consisting of two CSTRs and a nonadiabatic flash is considered. A description of the plant is available in Chapter 4 (Section 4.7.2, p. 58). Plant parameters and regulator constraints are given in Tables 11.1 and 11.2 respectively. The open-loop time constants for the two CSTRs are much smaller than the open-loop time constant for the flash separator. A time scale separation is present therefore. The MPCs for the two CSTRs are assigned to $\mathcal{J}_{\text{fast}}$; the MPC for the flash is assigned to $\mathcal{J}_{\text{slow}}$. The sampling rate for MPCs in $\mathcal{J}_{\text{fast}}$ is 1.5 sec. The sampling rate for $\mathcal{J}_{\text{slow}}$ is 15 sec. Thus, $\sigma = 10$. Under asynchronous feedback, MPCs 1 and 2 (for the two CSTRs) may inject 10 control moves in the time MPC 3 injects one control move. A schematic of the plant with the group divisions is shown in Figure 11.3. For each unit, we select $w_i = 1/3,\ i = 1, 2, 3$. Hence, $\gamma_f = 2/3$ and $\gamma_s = 1/3$.

Table 11.1: Steady-state parameters. The operational steady state corresponds to maximum yield of B.

$\rho = 0.15$ Kg m^{-3}, $\rho_b = 10\rho$	$\alpha_A = 3.5$	$\alpha_B = 1.1$
$\alpha_C = 0.5$	$k_1^* = 0.334$ sec^{-1}	$k_2^* = 0.5$ sec^{-1}
$A_r = 3$ m^2	$A_m = 3$ m^2	$A_b = 30$ m^2
$F_0 = 2.667$ Kg sec^{-1}	$F_1 = 1.33$ Kg sec^{-1}	$D = 6$ Kg sec^{-1}
$F_p = 0.01D$	$T_0 = 313$ K	$T_d = 313$ K
$C_{p_r} = 25$ KJ (Kg K)$^{-1}$	$C_{p_b} = 2.5$ KJ (Kg K)$^{-1}$	$C_{p_m} = C_{p_r}$
$Q_{\mathrm{r}} = Q_{\mathrm{m}} = -25$KJ sec^{-1}	$Q_{\mathrm{b}} = 2.5$ KJ sec^{-1}	$x_{A_0} = 1$
$x_{B_0} = x_{C_0} = 0$	$x_{A_1} = 1$	$x_{B_1} = x_{C_1} = 0$
$\Delta H_1 = -40$ KJ Kg^{-1}	$\Delta H_2 = -500$ KJ Kg^{-1}	$\frac{E_1}{R} = \frac{E_2}{R} = 150$K
$k_r = 2.5$ Kg sec^{-1}m$^{-\frac{1}{2}}$	$k_m = 2.5$ Kg sec^{-1}m$^{-\frac{1}{2}}$	$k_b = 1.5$ Kg sec^{-1}m$^{-\frac{1}{2}}$

Table 11.2: Input constraints. The symbol Δ represents a deviation from the corresponding steady-state value.

$-0.15 \leq \Delta F_0 \leq 0.15$	$-0.15 \leq \Delta Q_r \leq 0.15$
$-0.15 \leq \Delta F_1 \leq 0.15$	$-0.15 \leq \Delta Q_r \leq 0.15$
$-0.15 \leq \Delta D \leq 0.15$	$-3 \leq \Delta Q_b \leq 3$

The manipulated variables (MVs) for CSTR-1 are the feed flowrate F_0 and the cooling duty Q_{r}. The measured variables are the level of liquid in the reactor H_{r}, the exit mass fractions of A and B *i.e.*, $x_{\mathrm{A_r}}, x_{\mathrm{B_r}}$ respectively and the reactor temperature T_{r}. The controlled variables (CVs) for CSTR-1 are H_{r} and T_{r}. The MVs for CSTR-2 are the feed flowrate F_1 and the reactor cooling load Q_{m}. The measured variables are the level H_{m}, the mass fractions of A and B $x_{\mathrm{A_m}}, x_{\mathrm{B_m}}$ at the outlet, and the reactor temperature T_{m} . The CVs are H_{m} and T_{m}. For the nonadiabatic flash, the MVs are the recycle flowrate D and the heat duty for the flash Q_{b}. The CVs are the holdup in the flash H_{b} and the temperature T_{b}. The measurements are $H_{\mathrm{b}}, T_{\mathrm{b}}$ and the product stream mass fractions $x_{\mathrm{A_b}}$, $x_{\mathrm{B_b}}$. For each MPC, a control horizon $N = 15$ is selected. The regulator penalty for each CV is 10; the penalty for each MV is 1.

The performance of the following MPC frameworks are examined: (i) centralized MPC operating at the slowest sampling rate (15 sec) (ii) FC-MPC (1 iterate) operating at the slowest

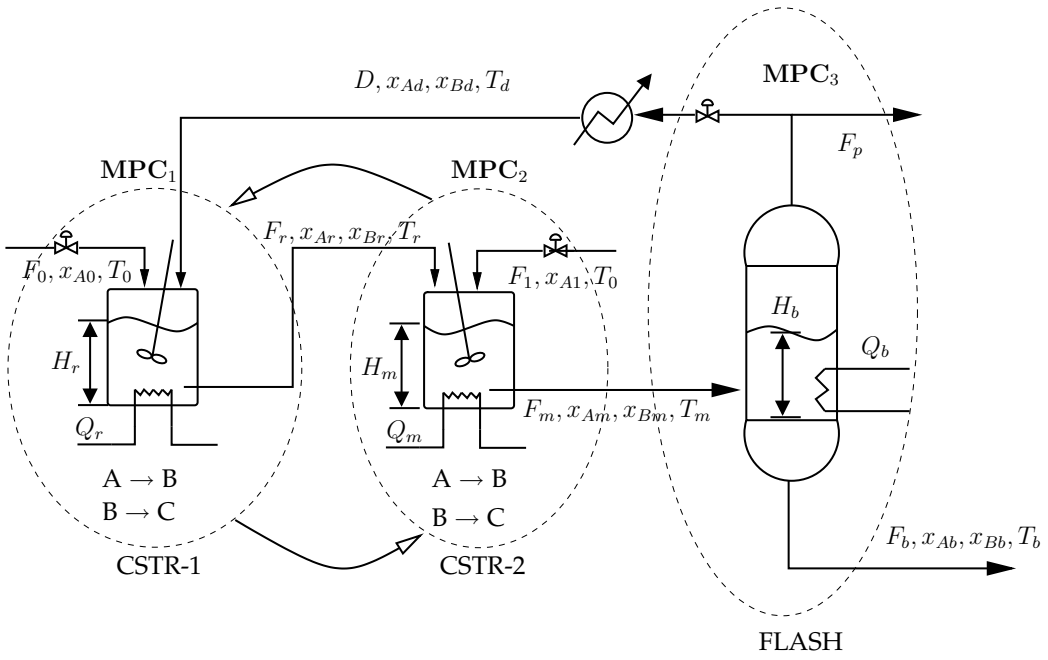

Figure 11.3: Two reactor chain followed by flash separator with recycle. MPCs for CSTRs 1 and 2 are assigned to group $\mathcal{J}_{\text{fast}}$. MPC 3 for the flash belongs to group $\mathcal{J}_{\text{slow}}$.

sampling rate (iii) asynchronous feedback FC-MPC with $q_f = 1$ (iv) asynchronous feedback FC-MPC with $q_f = 2$ (v) centralized MPC operating at the fastest sampling rate (1.5 sec). The performance of these MPCs is investigated when a setpoint change of $10^{\circ}C$ is made to liquid temperature T_b for the flash. The performance of the different MPCs is shown in Figures 11.4 and 11.5. Closed-loop control costs are given in Table 11.3.

Table 11.3: Closed-loop performance comparison of centralized MPC, FC-MPC and asynchronous feedback FC-MPC (AFFC-MPC). $\Delta\Lambda_{\text{cost}}$ calculated w.r.t performance of Cent-MPC (fast).

	Λ_{cost}	$\Delta\Lambda_{\text{cost}}\%$
Cent-MPC (fast)	13.3	0
Cent-MPC (slow)	32.1	141
FC-MPC (1 iterate, slow)	35.4	166
AFFC-MPC ($q_f = 1$)	21.6	61
AFFC-MPC ($q_f = 2$)	20.2	51

Asynchronous feedback FC-MPC with $q_f = 1$ outperforms centralized MPC operating

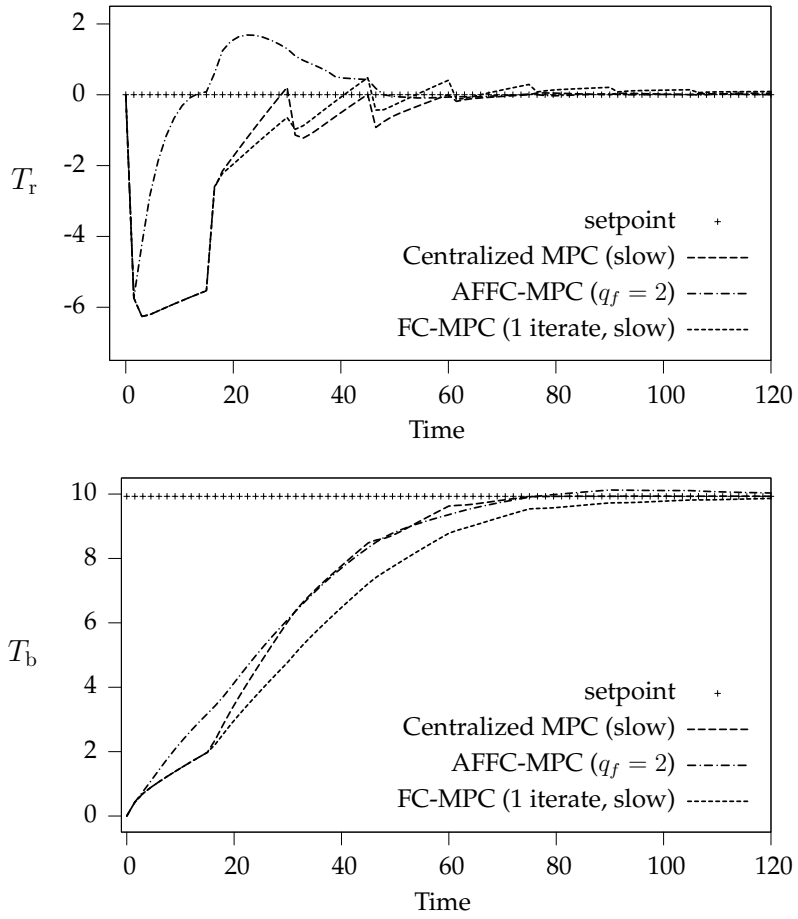

Figure 11.4: Setpoint tracking performance of centralized MPC, FC-MPC and asynchronous feedback FC-MPC (AFFC-MPC).

at the slowest sampling rate by about 33%. If $q_f = 2$, this performance improvement increases to about 37%. Centralized MPC at the fastest sampling rate outperforms AFFC-MPC ($q_f = 2$) by nearly 50%.

11.6 Discussion and conclusions

For any system, centralized MPC at the fastest sampling rate gives the best achievable performance. Implementing centralized MPC at the fastest sampling rate may not be feasible,

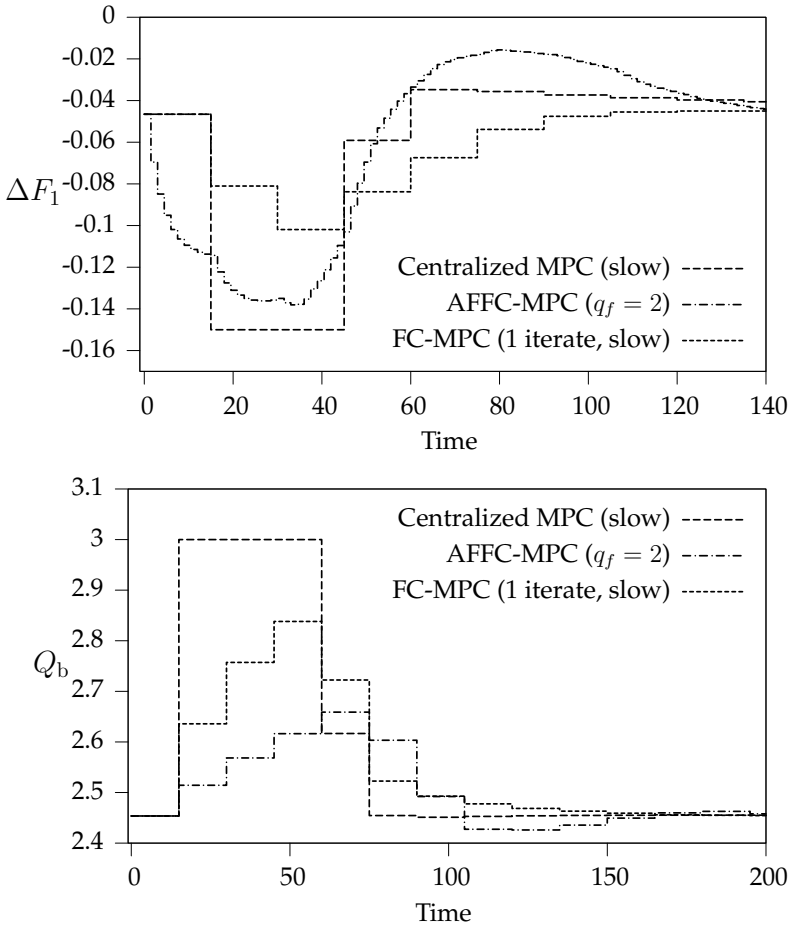

Figure 11.5: Setpoint tracking performance of centralized MPC, FC-MPC and asynchronous feedback FC-MPC (AFFC-MPC).

however, due to operational constraints such as unavailability of all process measurements at the fastest sampling rate. In such cases, asynchronous feedback FC-MPC presents an opportunity to obtain control performance that is superior to centralized MPC at the slowest sampling rate. A framework for asynchronous feedback distributed MPC was described in this chapter. A scenario with two sampling rates was considered. The subsystem sampling rates for a large, networked system are usually determined using the dominant time constant for each subsystem and frequency of available (local) measurements. Each MPC is assigned to either the group of fast MPCs $\mathcal{J}_{\text{fast}}$ or the group of slow MPCs $\mathcal{J}_{\text{slow}}$, based on the sampling rate for

the subsystem it controls. During each sampling interval $[kT + \alpha t_f, kT + (\alpha + 1)t_f)$ for $\mathcal{J}_{\text{fast}}$, each MPC in $\mathcal{J}_{\text{fast}}$ computes an optimal input trajectory from the predicted system state at the end of the current sampling interval. Information transfer occurs between MPCs in $\mathcal{J}_{\text{fast}}$ only. MPCs in $\mathcal{J}_{\text{fast}}$ utilize new input trajectories received from other MPCs in $\mathcal{J}_{\text{fast}}$ and reoptimize. Several such inner iterations may be performed by MPCs in $\mathcal{J}_{\text{fast}}$ during a sampling interval. A control law for asynchronous feedback distributed MPC was defined. This feedback law allows MPCs in $\mathcal{J}_{\text{fast}}$ to inject control actions into their respective subsystems without compromising nominal closed-loop stability. During a sampling interval for $\mathcal{J}_{\text{slow}}$, several control moves may be injected by MPCs in $\mathcal{J}_{\text{fast}}$. If all MPCs are sampled at the fast sampling rate, we set $\gamma_f = 1$ in Algorithm 11.3 and in the asynchronous feedback law (Section 11.3.1). In this limit, we revert to synchronous FC-MPC (at the fast sampling rate) described in Chapter 4. If all MPCs are sampled at the slow sampling rate, we set $\gamma_s = 1$ in Algorithm 11.1 and in the asynchronous feedback law. In this case, we revert to FC-MPC (Chapter 4) with $p(k) = 1$. Another interesting situation arises when the state estimators for the slow MPCs can function at the fastest sampling rate. In this case, Assumption 11.1 can be relaxed. Each MPC $i \in \mathcal{J}_{\text{fast}}$ instead, transmits its injected control action to the state estimator for each subsystem $j \in \mathbb{I}_M, j \neq i$.

In Chapter 7, partial cooperation FC-MPC was used to integrate the lower level flow controllers with the higher level MPC. Asynchronous feedback FC-MPC may be used in lieu of partial cooperation for vertical integration within a subsystem. The time scale separation present may be exploited for implementing asynchronous feedback FC-MPC. Flow controllers are typically sampled much faster than the higher level MPC. The flow controllers are assigned to $\mathcal{J}_{\text{fast}}$ while the higher level MPC is assigned to $\mathcal{J}_{\text{slow}}$. The advantages of asynchronous

feedback FC-MPC over partial cooperation are as follows:

- Asynchronous feedback FC-MPC guarantees nominal closed-loop stability. Partial cooperation does not guarantee closed-loop stability and is recommended only for cases where some of the interactions are significantly weaker than others.

- The performance with asynchronous feedback FC-MPC is generally better than the performance of centralized MPC at the slowest sampling rate, especially when the time scale separation is significant. Partial cooperation is, for most cases, suboptimal.

The main advantage of partial cooperation for vertical integration is the simplicity of the resultant controller network structure. The flow controllers do not communicate with each another under partial cooperation. In asynchronous feedback FC-MPC, the different flow controllers are required to communicate at the fast sampling rate. This additional communication requirement for asynchronous feedback FC-MPC may not be desirable in some cases.

Chapter 12

Concluding Remarks

In conclusion, the main contributions of this dissertation are summarized and suggestions for possible future work are provided.

12.1 Contributions

The focus of this dissertation was to develop a framework for distributed MPC with guaranteed stability and performance properties. A summary of the main contributions of this dissertation is provided below.

- Several distributed MPC techniques available in the literature are (pure) *communication* based strategies. We showed that modeling the interaction between subsystems and exchanging input trajectories among MPCs (communication) is insufficient to provide even closed-loop stability. A *cooperation*-based distributed MPC framework was proposed, in which the local objective of each subsystem-based MPC is modified to achieve systemwide control goals.

- Optimality conditions for the proposed cooperative distributed MPC framework were characterized and an algorithm for distributed MPC was presented. For any feasible initialization of the distributed MPC algorithm, all iterates generated are feasible and the resulting nominal closed-loop system is shown to be exponentially stable under intermediate termination. These attributes allow the practitioner to terminate the distributed MPC algorithm at an intermediate iterate, regardless of convergence. At convergence of the distributed MPC algorithm, optimal, centralized MPC performance is achieved.

- Two distributed state estimation strategies were developed for estimating subsystem states from local measurements. Exponential stability for the combined distributed estimator - distributed regulator assembly in the case of decaying estimate error was established for any intermediate termination of the distributed MPC algorithm. This perturbed exponential stability result was established without any constraint qualification requirements.

- A subsystem-based disturbance modeling framework was developed. Necessary and sufficient conditions for assessing the suitability of chosen local disturbance models were presented. A distributed target calculation algorithm that enables the calculation of steady-state targets at the subsystem level was described. All iterates generated by the distributed target calculation algorithm are feasible steady states. For large, networked systems, the number of measurements is typically chosen (for robustness and redundancy) to be greater than the number of manipulated variables. Selecting appropriate controlled variables is important for achieving offset-free control. Sufficient conditions for achieving offset-free control objectives with distributed MPC were provided. A maxi-

mal positively invariant stabilizable set for distributed MPC, with state estimation, target calculation and regulation, was described. This set contains all admissible system state, disturbance, estimate error and setpoint quadruples for which the target calculation for each subsystem is feasible, and the distributed MPC control law is stable.

- The concept of partial cooperation was introduced to achieve operational objectives and/or vertically integrate lower level flow control MPCs with the higher level MPC that supplies flow setpoints. The advantages of partial cooperation are the simplicity of the resulting controller network structure and reduction in communication among MPCs. The disadvantage of partial cooperation is that no stability guarantees are available, except in special cases.

- Simple extensions of the distributed MPC algorithm were used for distributed constrained LQR (DCLQR) and terminal state constraint distributed MPC. Two algorithms for DCLQR were developed. In the first algorithm, an explicit terminal set constraint was enforced to ensure feasibility of the terminal control law. In the second algorithm, the terminal set constraint is not enforced explicitly: rather, it remains implicit in the construction of a positively invariant set that restricts permissible initial states. Both DCLQR algorithms enable one to achieve infinite horizon optimal performance at convergence using finite values of N.

- The proposed distributed MPC algorithm was augmented to allow asynchronous operation among MPCs. First, a framework that enables asynchronous MPC optimizations and exchange of input trajectories was derived. This asynchronous optimization based distributed MPC algorithm allows one to integrate MPCs with varying computational

time requirements without requiring all MPCs to operate at the slowest computational rate. Feasibility, optimality and closed-loop stability under intermediate termination were established for this asynchronous distributed MPC algorithm. Next, a distributed MPC framework that allows asynchronous feedback was developed. Each MPC was assigned to either a fast group or a slow group of MPCs based on the sampling rate of the subsystem it controls. In this setup, MPCs in the fast group are allowed to inject their computed control actions at the faster sampling rate while MPCs in the slow group inject their inputs at the slower sampling rate. Nominal closed-loop stability under asynchronous feedback was established for any intermediate termination of the distributed MPC algorithm. Such an arrangement allows one to achieve performance superior to that of centralized MPC operating at the slower sampling rate.

12.2 Directions for Future Research

Some possible directions for future research are outlined below.

- Optimality properties for the FC-MPC algorithm, in the presence of constraints that couple inputs from different subsystems, need to be investigated further. For this case, as seen in Chapter 4, the FC-MPC algorithm need not converge to the optimal, centralized MPC solution. An obvious characterization is the following: If the coupled input constraints are inactive at the optimal, centralized MPC solution, the FC-MPC algorithm converges to the optimal solution. Evidently, this characterization has limited applicability. A more general characterization of optimality that handles situations where at least

one coupled input constraint is active at the optimal solution is required. Specifically, answers are sought for the following:

  - Are there scenarios for which the FC-MPC algorithm with coupled input constraints converges to the optimal solution?

  - If so, can it be determined a priori if the FC-MPC algorithm will converge to an optimal (or suboptimal) solution?

- Reliable strategies are required for handling possible disruptions and delays in the communication of input trajectories among subsystems. It is postulated that the closed-loop system can be destabilized with incorrect input trajectory information due to information loss or delays. This conjecture needs verification. If decentralized MPC is stable, one may switch to decentralized MPC for stabilization and revert to cooperative distributed MPC when the communication among subsystems is back online. Using recent developments in control over networks (Baliga and Kumar, 2005; Casavola et al., 2006; Imer et al., 2004), it may be possible to develop more efficient and reliable strategies that handle situations where the information transfer among subsystems' MPCs is either delayed or disrupted.

- Handling uncertainty in the controller model remains a key issue that needs to be addressed. Interaction models are typically identified using closed-loop operating data. Typically, for small plant-model mismatch (using the identified models), the disturbance modeling framework described in Chapter 6 is sufficient to obtain good closed-loop performance. When the plant-model mismatch is more significant, robust distributed MPC

design may be necessary. With this issue in mind, a thorough investigation into robustness theory for distributed MPC needs to be undertaken. Establishing properties such as robust feasibility and stability in the distributed MPC setting could prove to be both useful and interesting. Construction of disturbance invariant sets (Kolmanovsky and Gilbert, 1998; Rakovic et al., 2004) for each subsystem could prove useful to establish robust stability.

- In Chapter 7, a partial cooperation strategy that helped reduce communication among subsystems was described. General stability guarantees for the partial cooperation framework are not available, however. In Chapters 8 and 11, communication among subsystems was reduced by enabling asynchronous operation. Techniques for further reducing communication among subsystems, without compromising closed-loop stability, should be investigated.

- To implement cooperative distributed MPC for systems with fast sampling rates, one may require techniques that allow a quick evaluation of the MPC optimization problem. The possibility of employing explicit MPC techniques (Bemporad and Filippi, 2003; Bemporad et al., 2002; Pannocchia, Rawlings, and Wright, 2006; Tondel et al., 2003) for distributed MPC should be investigated. One complication in distributed MPC is that the input trajectories for interconnected subsystems' MPCs are additional parameters for each MPC optimization problem. The dimensionality of the parameter space consequently, is much greater in distributed MPC.

- In Chapter 11, zero order holds were used for asynchronous feedback distributed MPC. A first order hold may be employed instead. For the same control performance, a first or-

der hold typically allows a larger sampling interval than a zero order hold; this feature is attractive as it provides time for further iterations. The underlying ideas for implementing first order holds in asynchronous feedback distributed MPC remain the same. The parameters in the optimization problems may vary however. Investigating the benefits and drawbacks of first order holds for asynchronous feedback distributed MPC could be an interesting research problem.

- The efficacy of the proposed distributed MPC framework is contingent on the quality of models used. While 'closed-loop identification' is a mature and well understood field, developing identification techniques tailored for MPC is a relatively recent research area. Improvements in techniques for closed-loop identification for distributed MPC will likely prove beneficial for practical implementation. Reliable integration of the algorithm used for closed-loop identification with the algorithm for distributed MPC may have significant impact in the process industry. The idea is to be able to re-identify the models when the control performance drops below some pre-assigned limits due to changes in the plant. Reliable '*adaptive*' distributed control remains one of the sternest challenges for the automatic control community.

Appendix A

Example parameters and model details

A.1 Four area power system

Table A.1: Model, regulator parameters and input constraints for four area power network of Figure 3.3.

D_1 = 3	D_2 =0.275
R_1^f =0.03	R_2^f = 0.07
M_1^a = 4	M_2^a = 40
T_{CH_1}= 5	T_{CH_2} = 10
T_{G_1} = 4	T_{G_2} = 25
D_3 = 2.0	D_4 = 2.75
R_3^f =0.04	R_4^f = 0.03
M_3^a = 35	M_4^a = 10
T_{CH_3}= 20	T_{CH_4} = 10
T_{G_3} = 15	T_{G_4} = 5
T_{12} =2.54	T_{23} = 1.5
T_{34} = 2.5	Δ_{samp} = 1 sec

-0.5$\leq\Delta P_{\mathrm{ref}_1}\leq$0.5	$Q_1 = \mathrm{diag}(5,0,0)$ $R_1 = 1$
-0.5$\leq\Delta P_{\mathrm{ref}_2}\leq$0.5	$Q_2 = \mathrm{diag}(5,0,0,5)$ $R_2 = 1$
-0.5$\leq\Delta P_{\mathrm{ref}_3}\leq$0.5	$Q_3 = \mathrm{diag}(5,0,0,5)$ $R_3 = 1$
-0.5$\leq\Delta P_{\mathrm{ref}_4}\leq$0.5	$Q_4 = \mathrm{diag}(5,0,0,5)$ $R_4 = 1$

Area	States	MVs	CVs
1	$\Delta\omega_1, \Delta P_{\mathrm{mech}_1}, \Delta P_{v_1}$	$\Delta P_{\mathrm{ref}_1}$	$\Delta\omega_1$
2	$\Delta\omega_2, \Delta P_{\mathrm{mech}_2}, \Delta P_{v_2}, \Delta P_{\mathrm{tie}}^{12}$	$\Delta P_{\mathrm{ref}_2}$	$\Delta\omega_2, \Delta P_{\mathrm{tie}}^{12}$
3	$\Delta\omega_3, \Delta P_{\mathrm{mech}_3}, \Delta P_{v_3}, \Delta P_{\mathrm{tie}}^{23}$	$\Delta P_{\mathrm{ref}_3}$	$\Delta\omega_3, \Delta P_{\mathrm{tie}}^{23}$
4	$\Delta\omega_4, \Delta P_{\mathrm{mech}_4}, \Delta P_{v_4}, \Delta P_{\mathrm{tie}}^{34}$	$\Delta P_{\mathrm{ref}_4}$	$\Delta\omega_4, \Delta P_{\mathrm{tie}}^{34}$

A.2 Distillation column control

Table A.2: Distillation column model.

$G_{11} = \frac{32.63}{(99.6s+1)(0.35s+1)}$	$G_{12} = \frac{-33.89}{(98.02s+1)(0.42s+1)}$
$G_{21} = \frac{34.84}{(110.5s+1)(0.03s+1)}$	$G_{12} = \frac{-18.85}{(75.43s+1)(0.3s+1)}$

$$\begin{bmatrix} T_{21} \\ T_7 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{32} & G_{22} \end{bmatrix} \begin{bmatrix} V \\ L \end{bmatrix}$$

A.3 Two reactor chain with flash separator

Table A.3: First principles model for the plant consisting of two CSTRs and a nonadiabatic flash. Part 1.

Reactor-1:

$$\frac{dH_r}{dt} = \frac{1}{\rho A_r}\left[F_0 + D - F_r\right]$$
$$\frac{dx_{A_r}}{dt} = \frac{1}{\rho A_r H_r}\left[F_0(x_{A_0} - x_{A_r}) + D(x_{A_d} - x_{A_r})\right] - k_{1_r}x_{A_r}$$
$$\frac{dx_{B_r}}{dt} = \frac{1}{\rho A_r H_r}\left[F_0(x_{B_0} - x_{B_r}) + D(x_{B_d} - x_{B_r})\right] + k_{1_r}x_{A_r} - k_{2_r}x_{B_r}$$
$$\frac{dT_r}{dt} = \frac{1}{\rho A_r H_r}\left[F_0(T_0 - T_r) + D(T_d - T_r)\right] - \frac{1}{C_p}\left[k_{1_r}x_{A_r}\Delta H_1 + k_{2_r}x_{B_r}\Delta H_2\right] + \frac{Q_r}{\rho A_r C_p H_r}$$

Reactor-2:

$$\frac{dH_m}{dt} = \frac{1}{\rho A_m}\left[F_r + F_1 - F_m\right]$$
$$\frac{dx_{A_m}}{dt} = \frac{1}{\rho A_m H_m}\left[F_r(x_{A_r} - x_{A_m}) + F_1(x_{A_1} - x_{A_m})\right] - k_{1_m}x_{A_m}$$
$$\frac{dx_{B_m}}{dt} = \frac{1}{\rho A_m H_m}\left[F_r(x_{B_r} - x_{B_m}) + F_1(x_{B_1} - x_{B_m})\right] + k_{1_m}x_{A_m} - k_{2_m}x_{B_m}$$
$$\frac{dT_m}{dt} = \frac{1}{\rho A_m H_m}\left[F_r(T_r - T_m) + F_1(T_0 - T_m)\right] - \frac{1}{C_p}\left[k_{1_m}x_{A_m}\Delta H_1 + k_{2_m}x_{B_m}\Delta H_2\right] + \frac{Q_m}{\rho A_m C_p H_m}$$

Nonadiabatic flash:

$$\frac{dH_b}{dt} = \frac{1}{\rho_b A_b}\left[F_m - F_b - D - F_p\right]$$
$$\frac{dx_{A_b}}{dt} = \frac{1}{\rho_b A_b H_b}\left[F_m(x_{A_m} - x_{A_b}) - (D + F_p)(x_{A_d} - x_{A_b})\right]$$
$$\frac{dx_{B_b}}{dt} = \frac{1}{\rho_b A_b H_b}\left[F_m(x_{B_m} - x_{B_b}) - (D + F_p)(x_{B_d} - x_{B_b})\right]$$
$$\frac{dT_b}{dt} = \frac{1}{\rho_b A_b H_b}\left[F_m(T_m - T_b)\right] + \frac{Q_b}{\rho A_b H_b C_{p_b}}$$

Table A.4: First principles model for the plant consisting of two CSTRs and a nonadiabatic flash. Part 2.

$$F_r = k_r\sqrt{H_r} \qquad F_m = k_m\sqrt{H_m} \qquad k_{1_r} = k_1^* \exp\left(\frac{-E_1}{RT_r}\right)$$

$$F_b = k_b\sqrt{H_b} \qquad x_{C_r} = 1 - x_{A_r} - x_{B_r} \qquad k_{2_r} = k_2^* \exp\left(\frac{-E_2}{RT_r}\right)$$

$$x_{C_m} = 1 - x_{A_m} - x_{B_m} \qquad x_{C_b} = 1 - x_{A_b} - x_{B_b} \qquad k_{1_m} = k_1^* \exp\left(\frac{-E_1}{RT_m}\right)$$

$$x_{A_d} = \frac{\alpha_A x_{A_b}}{\Sigma} \qquad x_{B_d} = \frac{\alpha_B x_{B_b}}{\Sigma} \qquad k_{2_r} = k_2^* \exp\left(\frac{-E_2}{RT_m}\right)$$

$$x_{C_d} = \frac{\alpha_C x_{C_b}}{\Sigma} \qquad \Sigma = \alpha_A x_{A_b} + \alpha_B x_{B_b} + \alpha_C x_{C_b}$$

Table A.5: Steady-state parameters for Example 4.7.2. The operational steady state corresponds to maximum yield of B.

$\rho = \rho_b = 0.15$ Kg m^{-3}	$\alpha_A = 3.5$	$\alpha_B = 1.1$
$\alpha_C = 0.5$	$k_1^* = 0.02$ sec^{-1}	$k_2^* = 0.018$ sec^{-1}
$A_r = 0.3$ m^2	$A_m = 3$ m^2	$A_b = 5$ m^2
$F_0 = 2.667$ Kg sec^{-1}	$F_1 = 1.067$ Kg sec^{-1}	$D = 30.74$ Kg sec^{-1}
$F_p = 0.01D$	$T_0 = 313$ K	$T_d = 313$ K
$C_p = C_{p_b} = 25$ KJ (Kg K)$^{-1}$	$Q_{\rm r} = Q_{\rm m} = Q_{\rm b} = -2.5$ KJ sec^{-1}	$x_{A_0} = 1$
$x_{B_0} = x_{C_0} = 0$	$x_{A_1} = 1$	$x_{B_1} = x_{C_1} = 0$
$\Delta H_1 = -40$ KJ Kg^{-1}	$\Delta H_2 = -50$ KJ Kg^{-1}	$\frac{E_1}{R} = \frac{E_2}{R} = 150$K
$k_r = 2.5$ Kg sec^{-1}m$^{-\frac{1}{2}}$	$k_m = 2.5$ Kg sec^{-1}m$^{-\frac{1}{2}}$	$k_b = 1.5$ Kg sec^{-1}m$^{-\frac{1}{2}}$

A.4 Unstable three subsystem network

Table A.6: Nominal plant model for Example 5 (Section 4.7.3). Three subsystems, each with an unstable decentralized pole. The symbols $y_{\rm I} = [y_1{}', y_2{}']'$, $y_{\rm II} = [y_3{}', y_4{}']'$, $y_{\rm III} = y_5$, $u_{\rm I} = [u_1{}', u_2{}']'$, $u_{\rm II} = [u_3{}', u_4{}']'$, $u_{\rm III} = u_5$.

$$
\begin{aligned}
G_{11} &= \begin{bmatrix} \dfrac{s-0.75}{(s+10)(s-0.01)} & \dfrac{0.5}{(s+11)(s+2.5)} \\ \dfrac{0.32}{(s+6.5)(s+5.85)} & \dfrac{1}{(s+3.75)(s+4.5)} \end{bmatrix} \\
G_{12} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
G_{13} &= \begin{bmatrix} \dfrac{s-5.5}{(s+2.5)(s+3.2)} \\ \dfrac{0.3}{(s+11)(s+27)} \end{bmatrix} \\
G_{21} &= \begin{bmatrix} \dfrac{s-0.3}{(s+6.9)(s+3.1)} & \dfrac{0.31}{(s+41)(s+34)} \\ \dfrac{-0.19}{(s+16)(s+5)} & \dfrac{0.67(s-1)}{(s+12)(s+7)} \end{bmatrix} \\
G_{22} &= \begin{bmatrix} \dfrac{s-0.5}{(s+20)(s+25)} & \dfrac{0.6}{(s+14)(s+15)} \\ \dfrac{-0.33}{(s+3.0)(s+3.1)} & \dfrac{s-1.5}{(s+20.2)(s-0.05)} \end{bmatrix} \\
G_{23} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
G_{31} &= \begin{bmatrix} 0 & 0 \end{bmatrix} \\
G_{32} &= \begin{bmatrix} \dfrac{0.9}{(s+17)(s+10.8)} & \dfrac{-0.45}{(s+26)(s+5.75)} \end{bmatrix} \\
G_{33} &= \begin{bmatrix} \dfrac{s-3}{(s+12)(s-0.01)} \end{bmatrix}
\end{aligned}
$$

$$
\begin{bmatrix} y_{\rm I} \\ y_{\rm II} \\ y_{\rm III} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix} \begin{bmatrix} u_{\rm I} \\ u_{\rm II} \\ u_{\rm III} \end{bmatrix}
$$

Bibliography

L. Acar and U. Ozguner. A completely decentralized suboptimal control strategy for moderately coupled interconnected systems. In *Proceedings of the American Control Conference*, volume 45, pages 1521–1524, 1988.

T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. SIAM, Philadelphia, 1999.

G. Baliga and P. R. Kumar. A middleware for control over networks. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, pages 482–487, Seville, Spain, December 2005.

R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

A. Bemporad and C. Filippi. Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. *IEEE Transactions on Automatic Control*, 117(1): 9–38, 2003.

A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

D. P. Bertsekas. *Dynamic Programming*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, second edition, 1999.

R. R. Bitmead and M. Gevers. Riccati difference and differential equations: Convergence, monotonicity and stability. In S. Bittanti, A. J. Laub, and J. C. Willems, editors, *The Riccati Equation*, chapter 10, pages 263–291. Springer-Verlag, 1991.

R. R. Bitmead, M. R. Gevers, I. R. Petersen, and R. J. Kaye. Monotonicity and stabilizability properties of solutions of the Riccati difference equation: Propositions, lemmas, theorems, fallacious conjectures and counterexamples. *Systems & Control Letters*, 5:309–315, 1985.

F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 1999.

E. Camacho and C. Bordons. *Model Predictive Control*. Springer, Berlin, second edition, 2004.

E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, pages 44–52, February 2002.

B. Carew and P. Bélanger. Identification of optimum filter steady-state gain for systems with unknown noise covariances. *IEEE Transactions on Automatic Control*, 18(6):582–587, 1973.

A. Casavola, M. Papini, and G. Franzé. Supervision of networked dynamical systems under coordination constraints. *IEEE Transactions on Automatic Control*, 51(3):421–437, March 2006.

S. W. Chan, G. C. Goodwin, and K. S. Sin. Convergence properties of the Riccati difference equation in optimal filtering of nonstabilizable systems. *IEEE Transactions on Automatic Control*, 29(2):110–118, 1984.

C. C. Chen and L. Shaw. On receding horizon control. *Automatica*, 16(3):349–352, 1982.

C.-T. Chen. *Linear System Theory and Design*. Oxford University Press, 3rd edition, 1999.

D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. *Systems & Control Letters*, 29:121–129, 1996.

J. Choi and W. H. Kwon. Continuity and exponential stability of mixed constrained model predictive control. *SIAM Journal on Control and Optimization*, 42(3):839–870, 2003.

A. Clemmens, T. Kacerek, B. Grawitz, and W. Schuurmans. Test cases for canal control algorithms. *ASCE Journal of Irrigation and Drainage Engineering.*, 124(1):23–30, 1998.

J. E. Cohen. Cooperation and self interest: Pareto-inefficiency of Nash equilibria in finite random games. *Proceedings of the National Academy of Sciences of the United States of America*, 95: 9724–9731, 1998.

H. Cui and E. Jacobsen. Performance limitations in decentralized control. *Journal of Process Control*, 12:485–494, 2002.

C. R. Cutler and B. L. Ramaker. Dynamic matrix control—a computer control algorithm. In *Proceedings of the Joint Automatic Control Conference*, 1980.

E. J. Davison and H. W. Smith. Pole assignment in linear time-invariant multivariable systems with constant disturbances. *Automatica*, 7:489–498, 1971.

C. E. de Souza, M. R. Gevers, and G. C. Goodwin. Riccati equation in optimal filtering of nonstabilizable systems having singular state transition matrices. *IEEE Transactions on Automatic Control*, 31(9):831–838, September 1986.

P. Dubey and J. Rogawski. Inefficiency of smooth market mechanisms. *J. Math. Econ.*, 19: 285–304, 1990.

W. B. Dunbar. A distributed receding horizon control algorithm for dynamically coupled nonlinear systems. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, pages 6673–6679, Seville, Spain, December 2005.

W. B. Dunbar. Distributed receding horizon control of coupled nonlinear oscillators: Theory and application, December 13-15 2006.

W. B. Dunbar and R. M. Murray. Distributed receding horizon control with application to multi-vehicle formation stabilization. *Automatica*, 2(4):549–558, 2006.

B. A. Francis and W. M. Wonham. The internal model principle of control theory. *Automatica*, 12:457–465, 1976.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

A. Garcia, M. Hubbard, and J. de Vries. Open channel transient flow control by discrete time LQR methods. *Automatica*, 28(2):255–264, 1992.

C. E. García and A. M. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 46:73–87, 1986.

C. E. García and D. M. Prett. Advances in industrial model-predictive control. In M. Morari and T. J. McAvoy, editors, *Chemical Process Control—CPC III*, pages 245–293, Amsterdam, 1986. Third International Conference on Chemical Process Control, Elsevier.

C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9): 1008–1020, September 1991.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, third edition, 1996.

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

P.-O. Gutman and M. Cwikel. An algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states. *IEEE Transactions on Automatic Control*, 32(3):251–254, 1987.

W. W. Hager. Lipschitz continuity for constrained processes. *SIAM Journal on Control and Optimization*, 17(3):321–338, 1979.

A. Halanay. Quelques questions de la théorie de la stabilité pour les systèmes aux différences finies. *Archive for Rational Mechanics and Analysis*, 12:150–154, 1963.

V. Havlena and J. Lu. A distributed automation framework for plant-wide control, optimisation, scheduling and planning. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

P. M. Hidalgo and C. B. Brosilow. Nonlinear model predictive control of styrene polymerization at unstable operating points. *Computers & Chemical Engineering*, 14(4/5):481–494, 1990.

N. G. Hingorani and L. Gyugyi. *Understanding FACTS*. IEEE Press, New York, NY, 2000.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

J. Hurt. Some stability theorems for ordinary difference equations. *SIAM Journal on Numerical Analysis*, 4(4):582–596, 1967.

O. C. Imer, S. Yuksel, and T. Basar. Optimal control of dynamical systems over unreliable communication links. In *IFAC Symposium on Nonlinear Control Systems, Stuttgart, Germany*, pages 1521–1524, 2004.

D. Jia and B. H. Krogh. Distributed model predictive control. In *Proceedings of the American Control Conference*, pages 2767–2772, Arlington, Virginia, June 2001.

D. Jia and B. 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.

J. Juang and M. Phan. Identification of system, observer, and controller from closed-loop experimental data. *Journal of Guidance, Control, and Dynamics*, 17(1):91–96, January-February 1994.

T. Kailath. *Linear Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

M. Katebi and M. Johnson. Predictive control design for large-scale systems. *Automatica*, 33: 421–425, 1997.

S. S. Keerthi and E. G. Gilbert. Optimal infinite–horizon control and the stabilization of linear discrete–time systems: State–control constraints and nonquadratic cost functions. *IEEE Transactions on Automatic Control*, 31(3):264–266, March 1986.

S. S. Keerthi and E. G. Gilbert. Computation of minimum-time feedback control laws for systems with state-control constraints. *IEEE Transactions on Automatic Control*, 32:432–435, 1987.

T. Keviczky, F. Borelli, and G. J. Balas. Stability analysis of decentralized RHC for decoupled systems. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, pages 1689–1694, Seville, Spain, December 2005.

D. L. Kleinman. An easy way to stabilize a linear constant system. *IEEE Transactions on Automatic Control*, 15(12):692, December 1970.

I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4(4):317–367, 1998.

B. H. Krogh and P. V. Kokotovic. Feedback control of overloaded networks. *IEEE Transactions on Automatic Control*, 29(8):704–711, 1984.

R. Kulhavý, J. Lu, and T. Samad. Emerging technologies for enterprise optimization in the process industries. In J. B. Rawlings, B. A. Ogunnaike, and J. W. Eaton, editors, *Chemical Process Control–VI: Sixth International Conference on Chemical Process Control*, pages 352–363, Tucson, Arizona, January 2001. AIChE Symposium Series, Volume 98, Number 326.

W. H. Kwon and A. E. Pearson. On feedback stabilization of time-varying discrete linear systems. *IEEE Transactions on Automatic Control*, 23(3):479–481, June 1978.

W. H. Kwon, A. M. Bruckstein, and T. Kailath. Stabilizing state-feedback design via the moving horizon method. *International Journal of Control*, 37(3):631–643, 1983.

D. G. Lainiotis. Discrete Riccati equation solutions: Partitioned algorithms. *IEEE Transactions on Automatic Control*, 20(8):555–556, 1975.

D. G. Lainiotis, K. Plataniotis, M. Papanikolaou, and P. Papaparaskeva. Discrete Riccati equation solutions: Distributed algorithms. *Mathematical Problems in Engineering*, 2:319–332, 1996.

S. Lakshminarayanan, G. Emoto, S. Ebara, K. Tomida, and S. Shah. Closed loop identification and control loop reconfiguration: an industrial case study. *Journal of Process Control*, 11: 587–599, 2001.

J. P. LaSalle. The stability of dynamical systems. In *Regional Conference Series in Applied Mathematics #25*. SIAM, 1976.

J. Lu. Multi-zone control under enterprise optimization: needs challenges, and requirements. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, volume 26 of *Progress in Systems and Control Theory*, pages 393–402, Basel, 2000. Birkhäuser.

J. Lunze. *Feedback Control of Large Scale Systems*. Prentice-Hall, London, U.K, 1992.

W. L. Luyben. Dynamics and control of recycle systems. 1. simple open-loop and closed-loop systems. *Industrial and Engineering Chemistry Research*, 32:466–475, 1993a.

W. L. Luyben. Dynamics and control of recycle systems. 2. comparison of alternative process designs. *Industrial and Engineering Chemistry Research*, 32:476–486, 1993b.

W. L. Luyben. Snowball effects in reactor/separator processes with recycle. *Industrial and Engineering Chemistry Research*, 33:299–305, 1994.

L. Magni and R. Scattolini. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*, 42(7):1231–1236, 2006.

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

T. Meadowcroft, G. Stephanopoulos, and C. Brosilow. The Modular Multivariable Controller: 1: Steady-state properties. *AIChE Journal*, 38(8):1254–1278, 1992.

E. S. Meadows. *Stability and Continuity of Nonlinear Model Predictive Control*. PhD thesis, The University of Texas at Austin, 1994.

R. Mehra. On the identification of variances and adaptive Kalman filtering. *IEEE Transactions on Automatic Control*, 15(12):175–184, 1970.

R. Mehra. Approaches to adaptive filtering. *IEEE Transactions on Automatic Control*, 17:903–908, 1972.

H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

M. Morari and J. H. Lee. Model predictive control: past, present and future. In *Proceedings of joint 6th international symposium on process systems engineering (PSE '97) and 30th European symposium on computer aided process systems engineering (ESCAPE 7)*, Trondheim, Norway, 1997.

N. Motee and B. Sayyar-Rodsari. Optimal partitioning in distributed model predictive control. In *Proceedings of the American Control Conference*, pages 5300–5305, Denver,Colorado, June 2003.

K. R. Muske and T. A. Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5):617–632, 2002.

K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

R. Neck and E. Dockner. Conflict and cooperation in a model of stabilization policies: A differential game approach. *Journal of Economic Dynamics and Control*, 11:153–158, 1987.

B. J. Odelson, M. R. Rajamani, and J. B. Rawlings. A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42(2):303–308, February 2006. URL `http://www.elsevier.com/locate/automatica`.

B. A. Ogunnaike and W. H. Ray. *Process Dynamics, Modeling, and Control*. Oxford University Press, New York, 1994.

G. Pannocchia and J. B. Rawlings. Disturbance models for offset-free MPC control. *AIChE Journal*, 49(2):426–437, 2002.

G. Pannocchia, N. Laachi, and J. B. Rawlings. A candidate to replace PID control: SISO constrained LQ control. *AIChE Journal*, 51(4):1171–1189, April 2005. URL `http://www3.interscience.wiley.com/cgi-bin/fulltext/109933728/PDFSTART`.

G. Pannocchia, J. B. Rawlings, and S. J. Wright. Fast, large-scale model predictive control by partial enumeration. Accepted for publication in *Automatica*, 2006.

M.-A. Poubelle, R. R. Bitmead, and M. R. Gevers. Fake algebraic Riccati techniques and stability. *IEEE Transactions on Automatic Control*, 33(4):379–381, April 1988.

S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne. Invariant approximations of robustly positively invariant sets for constrained linear discrete-time systems subject to bounded disturbances. Technical report, Department of Engineering, University of Cambridge, Cambridge, UK, January 2004. CUED/F-INFENG/TR.473.

J. B. Rawlings and K. R. Muske. Stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, October 1993.

J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428, 1978.

A. Richards and J. How. A decentralized algorithm for robust constrained model predictive control. In *Proceedings of the American Control Conference*, Boston, Massachusetts, June 2004.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, N.J., 1970.

L. P. Russo and B. W. Bequette. Operability of chemical reactors: multiplicity behavior of a jacketed styrene polymerization reactor. *Chemical Engineering Science*, 53(1):27–45, 1998.

M. Saif and Y. Guan. Decentralized state estimation in large-scale interconnected dynamical systems. *Automatica*, 28(1):215–219, 1992.

Y. Samyudia and K. Kadiman. Control design for recycled, multi unit processes. *Journal of Process Control*, 13:1–14, 2002.

N. R. Sandell-Jr., P. Varaiya, M. Athans, and M. Safonov. Survey of decentralized control methods for larger scale systems. *IEEE Transactions on Automatic Control*, 23(2):108–128, 1978.

S. Sawadogo, P. Malaterre, and P. Kosuth. Multivariable optimal control for on–demand operation of irrigation canals. *International Journal of Systems Science*, 26(1):161–178, 1995.

S. Sawadogo, R. Faye, and F. Mora-Camino. Decentralized adaptive control of multireach irrigation canals. *International Journal of Systems Science*, 32(10):1287–1296, 2001.

J. Schuurmans, O. Bosgra, and R. Brouwer. Open–channel flow model approximation for controller design. *Applied Mathematical Modelling*, 19:525–530, September 1995.

P. O. Scokaert, J. B. Rawlings, and E. S. Meadows. Discrete-time stability with perturbations: Application to model predictive control. *Automatica*, 33(3):463–470, 1997.

P. O. M. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 43(8):1163–1169, August 1998.

D. D. Siljak. *Decentralized Control of Complex Systems*. Academic Press, London, 1991.

F. Sims, D. Lainiotis, and D. Magill. Recursive algorithm for the calculation of the adaptive Kalman filter weighting coefficients. *IEEE Transactions on Automatic Control*, 14(2):215–218, 1969.

S. Skogestad and M. Morari. Implications of large RGA elements on control performance. *Industrial and Engineering Chemistry Research*, 26:2323–2330, 1987.

E. D. Sontag. *Mathematical Control Theory*. Springer-Verlag, New York, second edition, 1998.

M. K. Sundareshan. Decentralized observation in large-scale systems. *IEEE Transactions on Systems, Man and Cybernetics*, 7(12):863–867, 1977.

M. K. Sundareshan and R. M. Elbanna. Design of decentralized observation schemes for large-scale interconnected systems: Some new results. *Automatica*, 26(4):789–796, 1990.

M. K. Sundareshan and P. C. Huang. On the design of a decentralized observation scheme for large-scale systems. *IEEE Transactions on Automatic Control*, 29(3):274–276, 1984.

M. Sznaier and M. J. Damborg. Heuristically enhanced feedback control of constrained discrete-time linear systems. *Automatica*, 26(3):521–532, 1990.

S. Talukdar, L. Baerentzen, A. Gove, and P. de Souza. Asynchronous teams: Cooperation schemes for autonomous agents. To appear in *Journal of Heursitics* and available at http://www.ece.cmu.edu/ talukdar, 1996.

P. Tondel, T. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3):489–497, 2003.

U.S.-Canada Power System Outage Task Force. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. April 2004.

A. N. Venkat. Distributed model predictive control: Theory and applications. Ph.D. thesis, Department of Chemical and Biological Engineering, University of Wisconsin, 2006.

A. N. Venkat, J. B. Rawlings, and S. J. Wright. Distributed model predictive control of large-scale systems. Assessment and Future Directions of Nonlinear Model Predictive Control, Freudenstadt-Lauterbad, Germany, August 2005a.

A. N. Venkat, J. B. Rawlings, and S. 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 2005b.

A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed MPC strategies for Automatic Generation Control. In *Proceedings of the IFAC Symposium on Power Plants and Power Systems Control*, Kananaskis, Canada, June 25-28 2006a.

A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed output feedback MPC for power system control. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, California, December 13-15 2006b.

A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed MPC strategies with application to power system automatic generation control. Technical Report 2006–05, TWMCC, Department of Chemical Engineering, University of Wisconsin-Madison, July 2006c.

A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed MPC strategies with application to power system automatic generation control. Submitted for publication in *IEEE Transactions on Control Systems Technology*, August 2006d.

A. N. Venkat, J. B. Rawlings, and S. J. Wright. Implementable distributed model predictive control with guaranteed performance properties. In *Proceedings of the American Control Conference*, Minneapolis, Minnesota, June 14-16 2006e.

A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed, linear model predictive control. part 1: state feedback. Submitted to *Automatica*, October 2006f.

A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed, linear model predictive control. part 2: output feedback. Submitted to *Automatica*, October 2006g.

M. Verhaegen. Application of a subspace model identification technique to identify LTI systems operating in closed-loop. *Automatica*, 29(4):1027–1040, 1993.

M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 2nd edition, 1993.

N. Viswanadham and A. Ramakrishna. Decentralized estimation and control for interconnected systems. *Large Scale Systems*, 3:255–266, 1982.

A. J. Wood and B. F. Wollenberg. *Power Generation Operation and Control*. John Wiley & Sons, New York, NY, 1996.

R. E. Young, R. D. Bartusiak, and R. W. Fontaine. Evolution of an industrial nonlinear model predictive controller. In J. B. Rawlings, B. A. Ogunnaike, and J. 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.

G. Zhu and M. Henson. Model predictive control of interconnected linear and nonlinear processes. *Industrial and Engineering Chemistry Research*, 41:801–816, 2002.

G. Zhu, M. Henson, and B. Ogunnaike. A hybrid model predictive control strategy for nonlinear plant-wide control. *Journal of Process Control*, 10:449–458, 2000.

Vita

Aswin Venkat was born in Kochi (formerly Cochin), Kerala, India on March 2, 1979. In August 2001, he graduated from the Indian Institute of Technology (IIT), Mumbai (formerly Bombay) with a Bachelor of Technology degree in Chemical Engineering and a parallel Master of Technology degree in Process Systems Design and Engineering. Soon after he moved to Madison, WI, USA to purse graduate studies under the direction of James B. Rawlings in the Department of Chemical and Biological Engineering at the University of Wisconsin-Madison.

Permanent Address: 28/988 Indira Nagar,
Kadavanthra,
Kochi, India 682020

This dissertation was prepared with LaTeX 2$_\varepsilon$[1] by the author.

[1]This particular University of Wisconsin compliant style was carved from The University of Texas at Austin styles as written by Dinesh Das (LaTeX 2$_\varepsilon$), Khe–Sing The (LaTeX), and John Eaton (LaTeX). Knives and chisels wielded by John Campbell and Rock Matthews.