University of Alberta

Coordinated-Distributed Optimal Control of Large-Scale Linear Dynamic Systems

by

Natalia Iris Marcos

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Chemical Engineering

Department of Chemical and Materials Engineering

Spring 2012
Edmonton, Alberta

Examining Committee

Dr. J. Fraser Forbes

Dr. Martin Guay

Dr. William R. Cluett

Dr. Sergiy A. Vorobyov

Dr. Sirish L. Shah

This thesis is dedicated to...

Adrián

Abstract

Since the late 1970s, the design of chemical processes has evolved towards highly integrated operations that can increase plant efficiency. Advances in the design of chemical processes include energy and mass integration and use of recycle streams, among others. Although such plant-wide integration offers substantial opportunities to improve the performance of the entire plant, it often results in large-scale systems that are challenging to control.

In the context of industrial processes, the term 'large-scale system' is used to describe the plant, which includes a number of processing units (subsystems) that are linked to each other as a result of shared or interacting variables. Optimal control of large-scale systems is a non-trivial task. Distributed control, which has attracted increasing attention in recent years, is seen as a promising new strategy for control of large-scale systems. This thesis focuses on a class of distributed control schemes, which is referred to as *coordinated-distributed* control. The coordinated-distributed control schemes proposed in this work use a coordinator that exchanges information with local controllers. This exchange of information allows the coordinator to iteratively adjust (coordinate) the optimal control problem for each of the local subsystems to drive their solutions toward the plant-wide optimal performance operations.

In this work, coordinated-distributed control schemes are formulated for dynamic linear systems that can be locally controlled by linear quadratic controllers or model predictive controllers. Two distinct methods are used for coordination of the local controllers: the *prediction-driven* coordination method and the *price-driven* coordination method. A common characteristic in both coordination methods is the

computation of a price vector. The price vector is a key element in the coordination of local controllers. It is updated iteratively by the coordinator to achieve the desired plant-wide optimal performance.

Most coordinated-distributed control schemes proposed in this work assume the same execution rate for the local controllers. Extensions for coordination of local controllers that are executed at different rates are also explored. In this part of the work, different strategies for dual-rate coordination are discussed and the strategy that is seen to provide the most performance improvement is analyzed in detail.

Several simulation examples, including benchmark processes obtained from the literature, are used to demonstrate the effectiveness of the proposed coordinated-distributed control schemes and their viability for dynamic interacting systems.

The proposed coordinated-distributed control strategies have significant potential for the (re)design of industrial control systems. The coordinated-distributed control schemes do not require a radical new configuration of the decentralized controllers. They can be constructed with minor modifications to the existing decentralized control systems.

Acknowledgements

I would like to express my gratitude to my supervisors Dr. J. Fraser Forbes and Dr. Martin Guay. I would like to thank Dr. Forbes for his excellent supervision, his kindness and for always being available whenever I needed help. I would also like to thank Dr. Guay for his supervision, his continued support and for always challenging me to do better. I would like to thank both of them for providing me the opportunity to participate in various international conferences.

I would like to thank Dr. William R. Cluett, Dr. Sergiy A. Vorobyov and Dr. Sirish L. Shah for accepting to be members of my thesis committee and for their efforts in reviewing and evaluating this thesis. I would also like to express my gratitude to Dr. Ruoyu Cheng and Dr. Anes Dallagi for the technical discussions which greatly benefited my research.

I am grateful for the financial support during my study at University of Alberta provided by Natural Sciences and Engineering Research Council of Canada and Alberta Innovates Technology Futures. I would like to extend my gratitude to the staff at the Department of Chemicals and Materials Engineering at University of Alberta. Many thanks go to Lily Laser and Marion Pritchard for their help and kindness.

Appreciation also goes out to the good friends I made along the way: Padideh Ghafoor Mohseni, Leily Mohammadi, Amir Alizadeh Moghadam, Hector De la Hoz Siegler, Xinguang Shao, Fei Qi and Hailei Jiang.

I would like to thank my family in Argentina for their constant love and encouragement. Most importantly of all, I would like to thank my husband Adrián for his invaluable support during all these years. Adrián, you have been my greatest source of strength. I could never have done this without you.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Optimal Control of Large-Scale Systems

The need for more efficient process operations, which includes energy integration and mass recycles, has lead to large-scale systems that can be highly interconnected. Large-scale process systems are composed of a number of processing units (subsystems) that are linked to each other as a result of shared or interacting variables. Optimal control of large-scale systems is a challenging problem. The challenge arises from the dependency (interactions) that exists among the different subsystems in the plant. On one hand, controlling the entire large-scale system, in an effort to consider the interactions between the subsystems, results in an intractable control problem. On the other hand, controlling the subsystems independently results in disregard of the interactions, which leads to suboptimal plant-wide operation.

There exist three approaches for optimal control of such integrated large-scale processes: *decentralized control*, *centralized control* and *distributed control*. These approaches are described in detail below. Among the three approaches, *distributed control* is seen as a promising control strategy for large-scale systems because it can overcome some of the limitations of both decentralized and centralized control. The potential benefits of distributed control is what motivated this research work. In

particular, this thesis focuses on a class of distributed control, which we refer to as *coordinated-distributed* control.

Decentralized Control

Control of large-scale systems such as industrial processes is typically performed in a completely *decentralized* fashion. An illustration of decentralized control scheme for a plant with two subsystems[1] is shown in Figure 1.1. It can be observed in Figure 1.1 that the decentralized control scheme uses individual controllers for each subsystem in the plant. In decentralized control, the operations of the different processing units

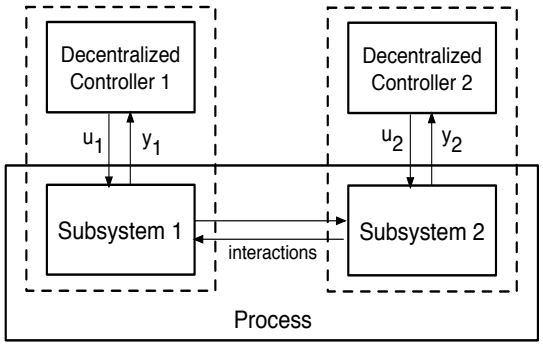

Figure 1.1: Illustration of decentralized control hierarchy for a plant with two subsystems

are optimized locally, without regard to the effects on the other processing units in the plant. Thus, decentralized control is an ideal control framework when there exist no interactions between the processing units in the plant. Since interactions between the processing units are common, decentralized control often results in suboptimal plant-wide performance and it can potentially lead to stability problems (Sun and El-Farra, 2008; Lu, 2003; Rawlings and Stewart, 2008).

[1]For the purpose of illustration and without loss of generality, a plant with two subsystems are used for the schematics in this chapter. Nevertheless, the coordinated-distributed control schemes proposed in this thesis can be applied to plants with more than two subsystems.

Centralized Control

Optimal plant-wide operations can be achieved if *centralized* control is implemented. A centralized control scheme is formulated as a monolithic control problem, where no real distinction is made among processing units. An illustration of centralized control scheme is shown in Figure 1.2. It can be observed in Figure 1.2 that centralized control relies on a single controller for the entire process. While centralized control can lead

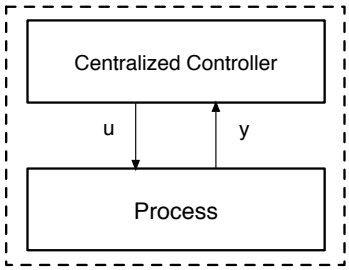

Figure 1.2: Illustration of centralized control hierarchy

to significantly improved plant-wide performance, it presents some disadvantages, which make centralized control unsuitable for optimal control of industrial processes (Lu, 2003). Some of the disadvantages of centralized control are lack of resiliency against equipment failure, lack of flexibility in terms of operation, and the use of a single centralized controller, which is difficult to tune and maintain.

Since centralized control is not generally preferred by practitioners and decentralized control might not achieve the entire plant-wide optimal performance, there is a need for development of a new control strategy for optimal control of large-scale systems. This new control strategy is referred to as *distributed control*.

Distributed Control

The distributed control scheme can be seen as an alternative to decentralized and centralized control schemes. Distributed control gained significant popularity in the last decade. Recently, a three-year project on hierarchical and distributed model predictive control of large-scale systems was organized in collaboration with researchers primarily from European universities (*http://www.ict-hd-mpc.eu/*, 2011).

One of the outcomes of this European project is a compilation of reports that include a thorough literature review of hierarchical and distributed model predictive controllers.

One observation worth noting is that distributed control schemes are available in the literature in a wide range of approaches according to the different communication topologies or control formulations considered (Scattolini, 2009). Nevertheless, all the distributed control schemes proposed in the literature shared one common characteristic. In the distributed control scheme, it is assumed that a certain amount of information is exchanged between the local controllers. The information exchange is required so that each local controller is aware of the other controllers' future actions and they can negotiate these future actions until the controllers achieve an agreement or a general consensus.

The distributed control schemes proposed in the literature can be classified into different categories. One classification criteria depends on the form of the objective function to be optimized by each local controller. If the distributed controller optimizes a local objective function and disregards the objectives of the other controllers in the plant, the distributed control scheme consists of "independent" algorithms (Scattolini, 2009; Li *et al.*, 2005). It is discussed in (Venkat *et al.*, 2006) that such independent distributed control formulation does not always guarantee closed-loop stability. The "independent" distributed control formulation can be improved when each local optimization problem is adjusted to include, up to a certain extent, the objectives of the other controllers in the plant. This improved distributed control scheme is a cooperating control scheme and it is referred in the literature as "cooperative" distributed control or "cooperation-based" distributed control (Stewart *et al.*, 2010; Venkat *et al.*, 2006).

Another classification criteria depends on the method used to communicate with the local distributed controllers. This method differs whether a coordinator is included or not in the distributed control scheme. When there is no coordinator incorporated in the distributed control scheme, the local distributed controllers exchange information among themselves to reach an agreement or consensus (see Figure 1.3a)). Examples of distributed control schemes that do not include a coordinator are described in (Dunbar and Murray, 2006; Venkat *et al.*, 2005; Venkat

et al., 2006; Jia and Krogh, 2002; Liu *et al.*, 2009). When there is a coordinator included in the distributed control structure, the information exchange takes place between the local distributed controllers and the coordinator. The control actions of the local distributed controllers are adjusted iteratively by the coordinator to reach a global objective (see Figure 1.3b)). The coordinator is generally designed on a different level with respect to the local distributed controllers and for that reason, this control scheme is also known as hierarchical distributed control. Coordination schemes for distributed control are given in (Katebi and Johnson, 1997; Aske *et al.*, 2008; Negenborn *et al.*, 2008; Cheng *et al.*, 2007; Cheng *et al.*, 2008). The *coordinated-distributed* control schemes proposed in this thesis correspond to one class of hierarchical distributed control schemes. The hierarchy of the coordinated-distributed control schemes proposed in this work is shown in Figure 1.3b).

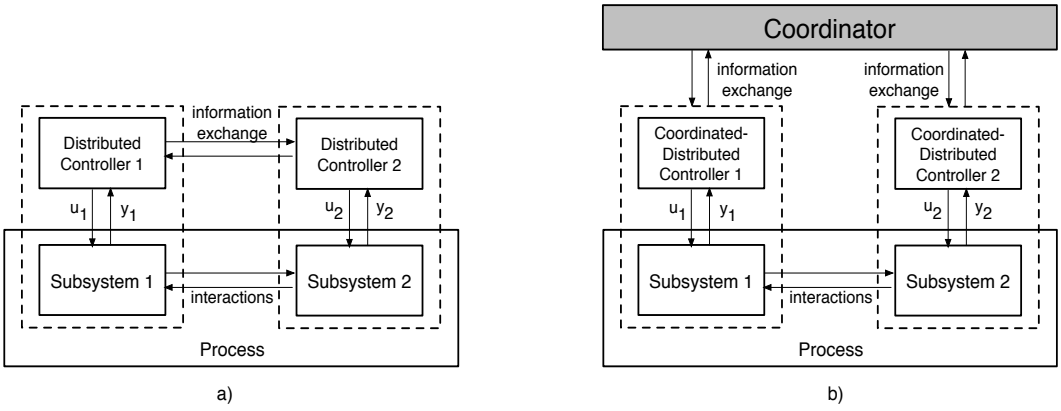

Figure 1.3: Illustration of distributed control hierarchy for a plant with two subsystems: a) without a coordinator; b) with the use of a coordinator

1.2 Research Scope and Contributions

This thesis is intended to make a contribution in the area of distributed control. In particular, this thesis concentrates on the development of coordinated-distributed control schemes. The main objective of the proposed control schemes is to optimize and control the dynamic behaviour of large-scale linear systems that present the following characteristics:

- the process systems are geographically distributed, often over large areas;
- there exist substantial interactions among processing units or subsystems arising, for example, from multi-step processing, energy integration and recycle streams;
- the processing units or subsystems may present different dimensions; that is, the number of process variables and manipulated variables can vary from subsystem to subsystem.

With regards to process modelling and control, it is assumed that:

- any plant-model mismatch is negligible;
- all the plant states are available for measurement. The construction of observers is not pursued in this thesis, although it is conceptually possible;
- the scope and tuning of each local controller are provided. In addition, it is assumed that the decentralized controllers existing in the plant (previous to the design of the coordinated-distributed controllers) provide stable operations for the entire process. In this thesis, the coordinated-distributed controllers are designed using the decentralized controllers already operating in the plant;
- the plant-wide performance objective is the sum of the performance objectives from the decentralized controllers. Furthermore, it is assumed that the performance objectives of the decentralized controllers do not compete with each other.
- For most of the coordinated-distributed control schemes proposed in this thesis, it is assumed that the control calculations are performed by the coordinated-distributed controllers at the same time. This means that all the coordinated-distributed controllers in the plant are executed with the same control rate. This assumption is required for the coordinated-distributed control schemes proposed in Chapters 2, 3 and 4. Nevertheless, this assumption is relaxed in Chapter 5, where a coordinated-distributed control scheme is proposed for local controllers that are executed at two different control rates.

In general, the development of distributed control schemes includes the following steps:

- decomposition of the large-scale control problem into smaller subproblems that are easier to solve. Decomposition of the large-scale problem also involves a decomposition of the entire process model into smaller processes or subsystems;
- design and tuning of each local controller;
- coordination of the local controllers to achieve a desired performance (for coordination-based distributed control). Alternatively, a method that allows cooperation between local controllers can be implemented (for cooperation-based distributed control).

The notion of "decomposition" in mathematical programming was explored in the early 60's by Dantzig and Wolfe (1960). Recently, Larsson and Skogestad (2000) described different methods of decomposing high-dimensional problems. These methods are classified as horizontal decomposition and hierarchical decomposition. Horizontal decomposition is based on process units, while the hierarchical decomposition considers process structure, control objectives and time scales (Larsson and Skogestad, 2000). Additional analysis on decomposition methods can be found in (Samyudia *et al.*, 1994; Walid *et al.*, 2010). The different decomposition methods are not pursued in this thesis[2]. Instead, coordination methods are considered in this thesis to adjust the solutions of the local controllers so that the centralized optimal performance can be achieved. In particular, two coordination methods are used in this work: the *price-driven* coordination method and the *prediction-driven* coordination method.

In the price-driven coordination method, the coordinator computes a price vector using a price-driven coordination algorithm and information provided by the local controllers. The price vector is then used in the optimal control problem of each subsystem and it is updated iteratively until a stopping criteria has been reached.

[2]In this work, it is assumed that the model of the subsystems (including the interactions) are given, regardless of the decomposition method used to determine the variables associated to each subsystem.

The *prediction-driven* method also uses a price vector as part of the prediction-driven coordination algorithm. In addition to the price vector, the coordinator predicts the state variables of each subsystem according to the entire plant process model. Both the price vector and the predicted states provided by the coordinator are used in the optimal control problem of each subsystem and are updated iteratively until a stopping criteria has been reached.

Finally, the main contributions of this thesis can be summarized as follows:

- coordinated-distributed control schemes are proposed to achieve the centralized optimal performance operations at all times. That is, the state and input trajectories obtained with the coordinated-distributed controllers exactly match the state and input trajectories obtained with the theoretical centralized controller at all times (provided that the coordination algorithm is allowed to iterate until its convergence is achieved). Therefore, the performance of the coordinated-distributed controllers is equal to the performance of the centralized controller;

- coordinated-distributed control schemes are proposed for a wide range of linear systems. Moreover, the coordinated-distributed control schemes proposed in Chapters 2, 3 and 4 can be used to control subsystems that are coupled through the control inputs and the states, while some distributed control frameworks available in the literature (e.g.: Morosan *et al.* (2011), Zafra-Cabeza *et al.* (2011), Maestre *et al.* (2011)) are suitable for subsystems that are coupled uniquely through the control inputs;

- the proposed coordinated-distributed control schemes are developed considering minimal modifications to the existing decentralized automation systems;

- a convergence analysis is provided for the prediction-driven and the price-driven coordination algorithms;

- stability analyses of closed-loop system under coordinated-distributed model predictive control are provided (for local controllers executed at the same control rate);

- a dual-rate coordinated-distributed control scheme is provided for local controllers executed at two different control rates. A method for extending the dual-rate coordinated-distributed control scheme to multi-rate coordinated-distributed control scheme is also proposed.

1.3 Thesis Outline

This thesis is structured as follows. Chapter 2 presents a state-feedback coordinated-distributed linear quadratic (CDLQ) control scheme for large-scale continuous-time, linear systems. The local CDLQ controllers are coordinated by means of the prediction-driven coordination algorithm. The convergence properties of the prediction-driven coordination algorithm are discussed. The performance of the CDLQ control scheme and the performance of the prediction-driven coordination algorithm presented in Chapter 2 are illustrated through computer simulations using two evaporator systems obtained from the literature.

The coordination principles developed in Chapter 2, which include the role of the coordinator and the local controllers, the communication flow between them and the iterative procedure required in the coordination algorithm, are extended for the development of a state-feedback coordinated-distributed model predictive control (CDMPC) scheme in Chapter 3. The CDMPC controllers constructed in Chapter 3 are suitable for unconstrained linear systems, whose dynamics are represented by discrete-time state-space models. Convergence of the prediction-driven coordination algorithm is shown along with a stability analysis of the closed-loop system under CDMPC control. The effectiveness of the CDMPC control scheme is also shown using a simulation example.

One of the reasons for the success of model predictive controllers is their capability to operate closer to constraints. Typically, process industries need to deal with constraints as a result of environmental and safety restrictions, product specifications and physical limitations in process equipments, among others. Therefore, limits on the state variables, input variables and input moves are incorporated in the formulation of the CDMPC controllers in Chapter 4. The price-driven coordination

method is used to coordinate the CDMPC controllers. The price-driven coordination method is used in Chapter 4, as an attempt to find a method that could potentially speed up coordination process. The price update mechanism in the price-driven coordination method is developed using Newton's method and a sensitivity analysis technique. An analysis of the CDMPC performance properties is also provided in this chapter. Furthermore, it is shown in Chapter 4 that the price-driven coordination algorithm is not restricted to processes whose dynamics are represented by state-space models. The price-driven coordination algorithm can also be used for processes whose dynamics are represented by finite step-response models. The performance of the CDMPC controllers using state-space models and finite step-response models are studied using two benchmark process systems: an evaporator process and a fluid catalytic cracking process obtained from the literature.

As an extension of the price-driven method developed in Chapter 4, Chapter 5 deals with the coordination of CDMPC controllers that are executed at two different control rates. Three distinct strategies are discussed in Chapter 5 for coordination of dual-rate CDMPC controllers. Among the three strategies, one is seen to have the most potential for performance improvement, thus it is further studied and developed. Using the selected strategy, a dual-rate CDMPC control scheme is formulated based on ideas derived from the *lifting technique*. The performance of the dual-rate CDMPC control scheme is tested through computer simulations. A method for extending the dual-rate CDMPC control scheme to multi-rate CDMPC control scheme is proposed towards the end of the chapter.

Finally, Chapter 6 summarizes the main results of this thesis and outlines directions for future research work.

1.4 Thesis Conventions

This section summarizes the conventions and definitions adopted throughout this thesis.

As described in Section 1.1, the term **large-scale system** refers to the entire system or entire process to be controlled. In an industrial process, the large-

scale system consists of the entire **plant-wide system**. From the point of view of modelling, the entire plant-wide system includes the local process units and the interactions between them.

When the model of the entire process is decomposed into smaller, local process models; these local process models are referred to as **subsystems**. If a distributed control scheme is used to control the subsystems, then all the interactions or only the most significant interactions are considered as part of the subsystems' models. If the decentralized control scheme is used to control the subsystems, then the interactions are not included as part of the subsystems' models.

The **centralized control problem** consists of optimizing an objective function subject to the entire plant-wide process model and all process operating constraints. When the centralized control problem is decomposed into smaller, local control problems, each local control problem is referred to as the **subproblem**.

The terms **optimal plant-wide operations** and **centralized optimal solution** are used interchangeably throughout this thesis to refer to the optimal solution (i.e., optimal control inputs and optimal state trajectories) obtained with a single monolithic centralized controller.

With regards to data sampling and control calculations performed by the local controllers, the following terms are used in this thesis:

Sampling instant: specific time at which a continuous-time signal is sampled (see Figure 1.4) (Chen and Francis, 1995).

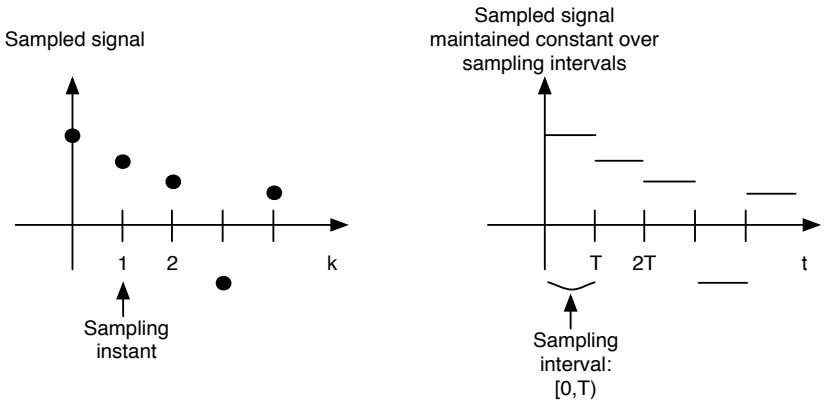

Figure 1.4: Illustration of sampling instants and sampling intervals

Sampling period: length of time between two consecutive sampling instants. It is denoted by the symbol 'T'.

Sampling intervals: set of times: $[0, T)$, $[T, 2T)$, $[2T, 3T)$, ... (see Figure 1.4).

Control instant: instant in time at which the control calculations are performed. It is assumed that the control instants are the same as the sampling instants.

Control interval: period between two consecutive control calculations. It is assumed to be the same as the sampling interval.

Finally, any other convention or term used in this thesis is explained in the corresponding chapter as needed.

2

Coordinated-Distributed LQ Control

This chapter presents a state-feedback coordinated-distributed linear quadratic (CDLQ) control scheme for large-scale continuous-time, linear dynamic systems. The proposed CDLQ control scheme incorporates a coordinator that manages the control actions in each local CDLQ controller to achieve the centralized optimal operation. The coordinator in CDLQ control scheme uses a coordination algorithm that is derived from the *Interaction Prediction Principle* (Mesarovic *et al.*, 1970). The convergence properties of the prediction-driven coordination algorithm are discussed. Furthermore, a quadratic *prediction error term* is implemented in the local CDLQ control problems to ensure convergence of the prediction-driven coordination algorithm. Finally, two illustrative examples are used to show the effectiveness of the CDLQ controllers and the behaviour of the prediction-driven coordination algorithm[1].

[1]Parts of this chapter are published in (Dallagi, A., N.I. Marcos and J. F. Forbes, 2008).

2.1 Background

The standard linear quadratic (LQ) control is the cornerstone of modern control theory. A number of advantageous properties are attributed to the standard LQ controllers. Firstly, the controller is optimal with respect to a meaningful performance functional. Secondly, the control input in the LQ regulator is formulated as a state-feedback control law that can stabilize processes that are open-loop unstable. Thirdly, the LQ optimal controller is linear in the state variables, which is a convenient feature for performance analysis and implementation purposes (Anderson and Moore, 1971; Lewis and Syrmos, 1995). These reasons have motivated the formulation of a CDLQ control scheme, which is the work presented in this chapter.

In this work, we propose a CDLQ control scheme, where the coordinator uses a prediction-driven coordination algorithm. The prediction-driven coordination algorithm is derived from the *Interaction Prediction Principle* (Mesarovic *et al.*, 1970). The *Interaction Prediction Principle* is very efficient for coordination of linear dynamic systems and it is used in (Cohen, 1977) to coordinate LQ regulators.

The main contributions of this chapter can be summarized as follows:

- In this chapter, state-feedback CDLQ tracking controllers are formulated for large-scale continuous-time, linear systems. We extended the work published in (Cohen, 1977) for CDLQ regulators. The CDLQ tracking controllers proposed here are a generalization of the CDLQ regulators presented in (Cohen, 1977). While the CDLQ regulators in (Cohen, 1977) can only drive the desired state variables to the origin, the CDLQ tracking controllers formulated in this chapter can track any desired trajectory (or steer controlled variables to their set-points) in an optimal fashion.

- The trade-off between the CDLQ controllers' performance and the coordination algorithm performance is discussed. In some cases, these two performance criteria tend to compete with each other. In some CDLQ control problems, tuning the weighting matrices to achieve zero-offset from the set-points, limits the ability of the coordination algorithm to converge to the centralized optimal solution. A quadratic *prediction error term*, referred in the literature as

'*proximal term*' (Rockafellar, 1976; Chen and Teboulle, 1994), is incorporated in the objective function of each subproblem to ensure both convergence of the coordination algorithm and zero-offset from the set-points. The *proximal term* was suggested by Cohen (1977) for CDLQ regulators and it is also included in this chapter for the CDLQ tracking control problem.

- Finally, the effect of a tuning parameter γ in the *proximal term* is studied and tested through computer simulations on a benchmark process system.

The CDLQ tracking control scheme proposed in this chapter has significant benefits for large-scale interacting systems. The CDLQ tracking controllers achieve the plant-wide (centralized) optimal performance, while providing a certain degree of flexibility and autonomy.

2.1.1 Optimal Control Theory

Before presenting the CDLQ control problems considered in this chapter, we introduce the background required to solve a general optimal control problem based on *calculus of variations* theory (Naidu, 2003).

Let us consider the finite-time optimal control problem, where we want to minimize the general objective function:

$$\mathcal{J} = V_f(x(t_f), t_f) + \int_{t_0}^{t_f} V(x(t), u(t), t)dt, \tag{2.1}$$

for the entire system or plant described by:

$$\dot{x}(t) = f(x(t), u(t), t), \tag{2.2}$$

with the following boundary conditions:

$$x(t_0) = x^{init} \text{ is fixed, } \; t = t_f \text{ is fixed and } \; x(t_f) \text{ is free.} \tag{2.3}$$

Optimal control problem (2.1)-(2.3) can be solved using *calculus of variations* with a Hamiltonian formalism (Naidu, 2003). Let us transform optimal control problem (2.1)-(2.2) by defining the Hamiltonian function as follows:

$$\mathcal{H}(x(t), u(t), \lambda(t), t) = V(x(t), u(t), t) + \lambda^T(t) f(x(t), u(t), t), \tag{2.4}$$

where the vector $\lambda(t)$ in (2.4) is the Lagrange multiplier vector.

Optimizing problem (2.1)-(2.3) is equivalent to optimizing problem (2.4) subject to the boundary conditions (2.3). The optimality conditions for problem (2.4) with boundary conditions (2.3) are referred to as the *Pontryagin* optimality conditions and are given by (Naidu, 2003):

$$\left.\frac{\partial \mathcal{H}}{\partial \lambda}\right|_{*} = \dot{x}^{*}(t), \qquad \text{state equation,} \tag{2.5}$$

$$\left.\frac{\partial \mathcal{H}}{\partial x}\right|_{*} = -\dot{\lambda}^{*}(t), \qquad \text{costate equation,} \tag{2.6}$$

$$\left.\frac{\partial \mathcal{H}}{\partial u}\right|_{*} = 0, \qquad \text{control equation,} \tag{2.7}$$

with the initial and final conditions:

$$x(t_0) = x^{init} \quad \text{and} \quad \left.\frac{\partial V_f}{\partial x(t_f)}\right|_{*} = \lambda^{*}(t_f). \tag{2.8}$$

The symbol '$*$' in equations (2.5)-(2.8) indicates the optimal value for the decision variables.

To summarize, optimal control problem (2.1)-(2.3) can be solved by following steps (1)-(5) described below (Naidu, 2003):

1. define the Hamiltonian function;
2. obtain the optimal control $u^{*}(t)$ from equation (2.7);
3. assume a form for the Lagrange multiplier $\lambda(t)$ in terms of the state variables $x(t)$;
4. solve the costate equation (2.6) and state equation (2.5), with initial and final conditions (2.8);
5. obtain the closed-loop optimal control $u^{*}(t)$.

2.1.2 Linear Quadratic Control

The linear quadratic control problem can be solved using the Hamiltonian approach described in Section 2.1.1. The term 'linear' in linear quadratic control refers to

the fact that the system to be controlled is a linear system. The term 'quadratic' indicates that the objective function in the control problem is a quadratic function of the decision variables (Ray, 1981).

Let us consider the following LQ tracking control problem for the entire plant:

$$\min_{x,u} \quad \mathcal{J}_{cen} = \frac{1}{2}\int_{t_0}^{t_f} \left((y_{sp}-y)^T Q(y_{sp}-y) + u^T R u\right) dt \tag{2.9}$$

subject to :

$$\begin{cases} \dot{x} = Ax + Bu, & \text{(2.10a)} \\ x(t_0) = x^{init}, & \text{(2.10b)} \\ y = Cx, & \text{(2.10c)} \end{cases}$$

where $x \in R^n$ and $u \in R^q$ denote the vector of state variables and input variables (or manipulated variables), respectively. The vector $y \in R^m$ denotes the controlled variables. The vector y_{sp} represents the vector of desired trajectories or set-points to be tracked in the entire process. In (2.10a)-(2.10c), the pair (A, B) is assumed to be controllable and the pair (A, C) is assumed to be observable. As discussed in Chapter 1, the construction of observers is not pursued in this thesis. Therefore, all the state variables are assumed to be measured; that is, $C = I_n$, where I_n is the identity matrix of dimensions $n \times n$.

In the objective function (2.9), the weighting matrix Q is symmetric positive semi-definite and the weighting matrix R is symmetric positive definite. The objective function (2.9) represents a trade-off between the control effort and the deviation of the controlled variables with respect to their desired value y_{sp}. Equation (2.10c) can be used to write the objective function (2.9) in terms of the state variables, leading to the following centralized LQ tracking control problem:

$$\min_{x,u} \quad \mathcal{J}_{cen} = \frac{1}{2}\int_{t_0}^{t_f} \left((y_{sp}-Cx)^T Q(y_{sp}-Cx) + u^T R u\right) dt \tag{2.11}$$

subject to :

$$\begin{cases} \dot{x} = Ax + Bu, \\ x(t_0) = x^{init}. \end{cases} \tag{2.12}$$

The Pontryagin optimality conditions (2.5)-(2.8) applied to the centralized control

problem (2.11)-(2.12) results in the following system of equations:

$$\begin{cases} \dot{x} = Ax + Bu, \\ x(t_0) = x^{init}, \end{cases} \tag{2.13}$$

$$\begin{cases} \dot{\lambda} = -A^T\lambda + C^TQ(y_{sp} - Cx), \\ \lambda(t_f) = 0, \end{cases} \tag{2.14}$$

$$0 = Ru + B^T\lambda. \tag{2.15}$$

2.2 Problem Description

A monolithic centralized LQ tracking controller can be obtained by solving optimal control problem (2.11)-(2.12). Due to the disadvantages of the centralized control scheme (see Section 1.1), a single centralized LQ tracking controller is not designed in this work. Instead, the centralized control problem (2.11)-(2.12) is partitioned into smaller subproblems that are easier to solve. It is proposed in this work the use of local CDLQ controllers that together produce the same optimal solution as the one obtained with the theoretical centralized controller.

To design the CDLQ controllers, it is assumed that the process model for the entire plant, equations (2.10a)-(2.10c), is partitioned into N subsystems. That is, it is assumed that the vector of state variables, input variables and controlled variables are partitioned into N components: $x = [x_1^T, \ldots, x_N^T]^T$ with $x_i \in \Re^{n_i}$; $u = [u_1^T, \ldots, u_N^T]^T$ with $u_i \in \Re^{q_i}$ and $y = [y_1^T, \ldots, y_N^T]^T$ with $y_i \in \Re^{n_i}$. Then, the dynamics of each subsystem i ($i = 1, \ldots, N$) are represented by the following state-space model[2]:

$$\begin{aligned} \dot{x}_i &= A_{ii}x_i + B_{ii}u_i + v_i, \\ x_i(t_0) &= x_i^{init}, \\ y_i &= C_{ii}x_i, \\ v_i &= \sum_{j \neq i} \left(A_{ij}x_j + B_{ij}u_j \right). \end{aligned} \tag{2.16}$$

[2]It is assumed that the state variables x_i, the input variables u_i and the controlled variables y_i are the variables associated to the local decentralized controller i existing in the plant, previous to the implementation of the CDLQ control scheme.

The matrices A_{ii}, B_{ii} and C_{ii} in (2.16) represent the local dynamics of subsystem i, for $i = 1, \ldots, N$. In this work, it is assumed that the pair (A_{ii}, B_{ii}) is controllable and $C_{ij} = 0$. The variables v_i in (2.16) are interacting variables and they represent the effect of the states x_j and the inputs u_j on the local subsystem i, where $j \neq i$.

It can be noted that the total number of state variables in the plant is $n = \sum_{i=1}^{N} n_i$ and the total number of input variables in the plant is $q = \sum_{i=1}^{N} q_i$.

2.3 Coordinated-Distributed LQ Control

In this section, the local CDLQ controllers are designed. The first step in designing the local CDLQ controllers is to formulate the optimization problem for the subsystems. The following optimization problem is proposed for each subsystem:

$$\begin{aligned}
\min_{x_i, u_i} \quad & \mathcal{J}_i = \frac{1}{2} \int_{t_0}^{t_f} \big((y_{i_sp} - C_{ii}x_i)^T Q_i (y_{i_sp} - C_{ii}x_i) + u_i^T R_i u_i \big) dt + \\
& \int_{t_0}^{t_f} \Big(\sum_{j \neq i} p_j^{s^T} \big(A_{ji}x_i + B_{ji}u_i \big) \Big) dt \\
\text{subject to :} \quad & \\
& \begin{cases} \dot{x}_i = A_{ii}x_i + B_{ii}u_i + v_i, \\ x_i(t_0) = x_i^{init}, \\ v_i = \sum_{j \neq i} \left(A_{ij}x_j^s + B_{ij}u_j^s \right). \end{cases}
\end{aligned} \tag{2.17}$$

The weighting matrix Q_i is symmetric positive semidefinite and the weighting matrix R_i is symmetric positive definite. The objective function in optimal control problem (2.17) consists of two main terms. The first term represents a trade-off between the local control effort and the deviation of the local controlled variables with respect to their desired value y_{i_sp}. The second term, '$\int_{t_0}^{t_f} \Big(\sum_{j \neq i} p_j^{s^T} \big(A_{ji}x_i + B_{ji}u_i \big) \Big) dt$', is included in the objective function $\mathcal{J}_i$ for coordination of the local CDLQ controllers. This second term is required so that the coordinator can manage the optimal solutions in the local subsystems and drive them towards the centralized optimal solution. The vector p_j^s, for $j \neq i$, is a price vector and it is computed by a coordinator based on the price update algorithm described in Section 2.3.2.

The superscript 's' used in optimal control problem (2.17) denotes the iteration step in the coordination algorithm. It is appended to the vectors p_j, x_j and u_j, for

$j \neq i$, to indicate that these vectors are computed by a coordinator at iteration step 's' (see Sections 2.3.2 and 2.3.3).

2.3.1 CDLQ Controllers

The CDLQ tracking controllers for each subsystem can be designed by following the procedure described in Section 2.1.1 for a general optimal control problem.

A Hamiltonian for each optimal control problem (2.17) is defined as follows:

$$\mathcal{H}_i = \frac{1}{2}\big((y_{i\_sp} - C_{ii}x_i)^T Q_i (y_{i\_sp} - C_{ii}x_i) + u_i^T R_i u_i\big) + \sum_{j\neq i} p_j^{s^T}\big(A_{ji}x_i + B_{ji}u_i\big) + \lambda_i^T (A_{ii}x_i + B_{ii}u_i + v_i). \tag{2.18}$$

Then, the Pontryagin optimality conditions for each subproblem become:

$$\frac{\partial \mathcal{H}_i}{\partial \lambda_i} = \dot{x}_i, \quad \Longrightarrow \quad \dot{x}_i = A_{ii}x_i + B_{ii}u_i + v_i, \tag{2.19}$$

$$\frac{\partial \mathcal{H}_i}{\partial x_i} = -\dot{\lambda}_i, \quad \Longrightarrow \quad \dot{\lambda}_i = -A_{ii}^T\lambda_i + C_{ii}^T Q_i (y_{i\_sp} - C_{ii}x_i) - \sum_{j\neq i} A_{ji}^T p_j^s, \tag{2.20}$$

$$\frac{\partial \mathcal{H}_i}{\partial u_i} = 0, \quad \Longrightarrow \quad 0 = R_i u_i + B_{ii}^T \lambda_i + \sum_{j\neq i} B_{ji}^T p_j^s, \tag{2.21}$$

with initial and final conditions:

$$x_i(t_0) = x_i^{init} \quad \text{and} \quad \lambda_i(t_f) = 0.$$

The control input can be obtained from equation (2.21) as:

$$u_i = -R_i^{-1}\Big(B_{ii}^T\lambda_i + \sum_{j\neq i} B_{ji}^T p_j^s\Big). \tag{2.22}$$

Next, the Lagrange multiplier is defined for each subsystem as follows:

$$\lambda_i = P_i x_i + r_i + z_i, \tag{2.23}$$

where the matrix P_i and the vectors r_i and z_i are yet to be determined. The vector r_i is included in equation (2.23) to let the controller track the desired trajectory y_{i_sp}. The vector z_i is included in equation (2.23) to let the coordinator drive the optimal

solutions of the subsystems towards the centralized optimal solution.
Equation (2.23) can be then differentiated to give:

$$\dot{\lambda}_i = \dot{P}_i x_i + P_i \dot{x}_i + \dot{r}_i + \dot{z}_i. \tag{2.24}$$

Equation (2.23) can be used to eliminate the vector λ_i from equations (2.22) and (2.20). Then, equations (2.19), (2.20), (2.22) and (2.24) can be arranged to obtain the following differential Riccati equation:

$$\begin{cases} \dot{P}_i = -P_i A_{ii} - A_{ii}^T P_i + P_i B_{ii} R_i^{-1} B_{ii}^T P_i - C_{ii}^T Q_i C_{ii}, \\ P_i(t_f) = 0, \end{cases} \tag{2.25}$$

and the following differential equations:

$$\begin{cases} \dot{r}_i = \left(P_i B_{ii} R_i^{-1} B_{ii}^T - A_{ii}^T\right) r_i + C_{ii}^T Q_i y_{i\_sp}, \\ r_i(t_f) = 0, \end{cases} \tag{2.26}$$

$$\begin{cases} \dot{z}_i = \left(P_i B_{ii} R_i^{-1} B_{ii}^T - A_{ii}^T\right) z_i - P_i \left(\sum_{j \neq i} A_{ij} x_j^s + B_{ij} u_j^s\right) + \\ \qquad P_i B_{ii} R_i^{-1} \left(\sum_{j \neq i} B_{ji}^T p_j^s\right) - \sum_{j \neq i} A_{ji}^T p_j^s, \\ z_i(t_f) = 0. \end{cases} \tag{2.27}$$

The control input for subsystem i can be obtained by substituting equation (2.23) into equation (2.22) as follows:

$$u_i = -R_i^{-1}\Big(B_{ii}^T (P_i x_i + r_i + z_i) + \sum_{j \neq i} B_{ji}^T p_j^s\Big). \tag{2.28}$$

In control law (2.28), the matrix P_i, and the vectors r_i and z_i are obtained by solving equations (2.25), (2.26) and (2.27), respectively; the vector p_j^s, for $j \neq i$, is provided by the coordinator (see Section 2.3.2).

Remark 2.3.1 *The affine state feedback control* (2.28) *consists of four terms. The first term includes the linear state feedback; the second term in* (2.28) *includes the vector r_i that is required by the local CDLQ controller to track the desired trajectory y_{i_sp}; the last two terms include the vectors z_i and p_j^s and they are required for coordination. Whereas the term that includes the vector r_i is required in the CDLQ tracking controllers, it is not necessary in the CDLQ regulators. Thus, by setting $y_{i\_sp} = 0$, the CDLQ regulators in (Cohen, 1977) can be obtained.*

2.3.2 Prediction-Driven Coordination Algorithm

In this section, the prediction-driven coordination algorithm is presented. The prediction-driven algorithm includes two types of calculations, which are performed by the coordinator at each iteration step 's'. In the first type of calculations, the coordinator predicts the subsystems' state variables using the entire plant process model. In the second type of calculations the coordinator updates the price vector. The price vector is required by the subsystems to solve their own optimal control problems.

Prediction of the state variables:
The subsystems' state variables are predicted by the coordinator according to the entire plant process model:

$$\begin{cases} \dot{x}^s = Ax^s + Bu^s, \\ x^s(t_0) = x^{init}, \end{cases} \tag{2.29}$$

where the input vector is defined as $u^s = [u_1^{s^T}, u_2^{s^T}, \ldots, u_N^{s^T}]^T$ and it is equal to the optimal values calculated by each CDLQ controller; that is, $u_i^s = u_i$, for $i = 1, ..., N$. The vector x^s in (2.29) is defined as $x^s = [x_1^{s^T}, x_2^{s^T}, \ldots, x_N^{s^T}]^T$, where x_i^s is the value predicted by the coordinator for the local state variables x_i, for $i = 1, ..., N$.

Price update:
To coordinate the CDLQ controllers, the coordinator calculates a full price vector λ_{coor}^s. The full price vector λ_{coor}^s is computed by the coordinator by integrating:

$$\begin{cases} \dot{\lambda}_{coor}^s = -A^T\lambda_{coor}^s + C^TQ(y_{sp} - Cx^s), \\ \lambda_{coor}^s(t_f) = 0. \end{cases} \tag{2.30}$$

The elements in the full price vector are arranged as: $\lambda_{coor}^s = [p_1^{s^T}, p_2^{s^T}, ..., p_N^{s^T}]^T$, where p_i^s has dimensions $n_i \times 1$, for $i = 1, \ldots, N$. The elements p_j^s (for $j \neq i$) are used in the optimization problem of each subsystem i (for $i = 1, ..., N$).

Remark 2.3.2 *It can be observed that differential equation* (2.30) *resembles differential equation* (2.14)*. Thus, the full price vector* λ_{coor}^s *is an approximation of*

the Lagrange multiplier vector for the centralized control problem (2.11)-(2.12). *When the prediction-driven coordination algorithm converges, the inputs* u^s *are equal to the centralized optimal inputs* u^* *(see Theorem 2.4.1), the predicted states* x^s *are equal to centralized optimal states* x^* *and the full price vector* λ^s_{coor} *is equal to the centralized Lagrange multiplier vector* λ^*.

2.3.3 Implementation of the CDLQ Controllers

In this section, the implementation of the CDLQ controllers is described. In order to implement the CDLQ controllers, some calculations can be performed off-line while some online computation is required. The calculations that can be computed off-line are the ones that do not depend on the state variables, input variables or prices. For example, given the fixed time t_f and the vector y_{i_sp}, the differential Riccati equation (2.25) and the differential equation (2.26) can be integrated for each subsystem off-line to obtain the matrix P_i and the vector r_i, respectively. The calculations that need to be computed on-line involve the variables that are changed by the coordination algorithm.

In the prediction-driven coordination algorithm, the coordinator exchanges information with the local CDLQ controllers. As a result of this information exchange, the coordinator iteratively adjusts the price vector λ^s_{coor} to drive the optimal solution of the subsystems to the centralized optimal solution. The information exchanged on-line between the coordinator and the local CDLQ controllers is shown in Figure 2.1.

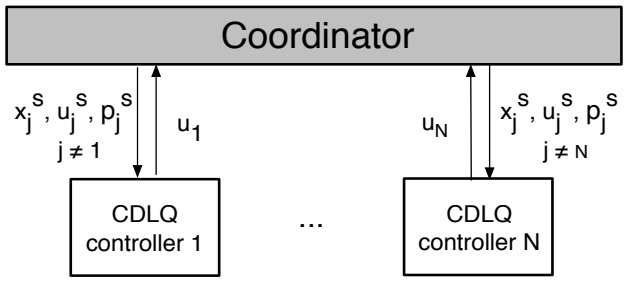

Figure 2.1: Communication between the coordinator and the CDLQ controllers

The implementation of the CDLQ controllers is carried out according to the following steps:

1. **Initialization ($s = 0$):** The coordinator provides initial predictions for the inputs u_i^0, for $i = 1, \ldots, N$. For initialization, the elements of the vector u^0 (with $u^0 = [u_1^{0^T}, u_2^{0^T}, \ldots, u_N^{0^T}]^T$) can be set to zero. Next, the coordinator predicts the state vector $x^0 = [x_1^{0^T}, x_2^{0^T}, \ldots, x_N^{0^T}]^T$ based on the initial inputs u^0, and the entire process model (2.29). Using the predicted state vector x^0 and price update algorithm (2.30), the coordinator calculates an initial price vector λ_{coor}^0, with $\lambda_{coor}^0 = [p_1^{0^T}, p_2^{0^T}, ..., p_N^{0^T}]^T$. The coordinator sends the predicted state variables x_j^0, the inputs u_j^0 and the prices p_j^0, with $j \neq i$, to each local CDLQ controller[3].

2. **Calculation of control inputs u_i:** Each CDLQ controller computes the vector z_i (see equation (2.27)) and control inputs u_i (see equation (2.28)) based on the matrix P_i and the vector r_i calculated off-line, and the predicted states, inputs and prices provided by the coordinator. The control inputs calculated by each CDLQ controller, u_i, are communicated back to the coordinator, for $i = 1, \ldots, N$.

3. **Price update:** The iteration counter 's' is incremented. The coordinator collects the information from each CDLQ controller and calculates a new prediction for the state variables, according to the entire process model (2.29) and with $u_i^s = u_i$, for $i = 1, \ldots, N$. Then, the coordinator determines the new price vector λ_{coor}^s, and transmits the predicted states x_j^s, the inputs u_j^s and the prices p_j^s, with $j \neq i$, to each CDLQ controller i.

4. **Iteration until convergence:** Steps (2)-(3) are repeated until the prediction-driven coordination algorithm is terminated. The prediction-driven coordination algorithm is terminated once $\|u^{s+1} - u^s\| \leq \epsilon$, where ϵ is a specified error tolerance.

[3]Alternatively, the prediction-driven algorithm can be initialized using $\lambda_{coor}^0 = 0$, $x^0 = 0$ and $u^0 = 0$. This is recommended for open-loop unstable systems.

5. **Implementation of the control inputs:** Once the prediction-driven coordination algorithm is terminated, each CDLQ controller implements their optimal control input u_i, for $i = 1, \ldots, N$.

2.4 Coordination Algorithm Convergence Analysis

In the optimal control problems considered in this chapter, the CDLQ controllers are required to achieve zero-offset from the set-points within a given time horizon. Nevertheless, tuning the CDLQ controllers to achieve a desired performance can sometimes conflict with the coordinator's behaviour. In some cases, tuning the weighting matrices to achieve zero-offset limits the ability of the coordination algorithm to converge to the centralized optimal solution. In this section, the convergence properties of the coordination algorithm are analyzed.

Theorem 2.4.1 is presented in this section to describe the convergence behaviour of the prediction-driven coordination algorithm. A convergence condition is provided in Theorem 2.4.1. This condition can be computed off-line to ensure that the coordination algorithm converges for the selected controller tunings. The objective functions $\mathcal{J}_{cen}(u)$ and $\Phi(u, v)$ are required to compute the convergence condition. They are defined in the following.

The centralized objective function $\mathcal{J}_{cen}$ is defined in the centralized control problem (2.11)-(2.12). By writing the state variables in the centralized control problem as $x = \mathbb{S}(u)$, the centralized objective function in problem (2.11)-(2.12) can be expressed as a function of the input variables as follows:

$$\mathcal{J}_{cen}(u) = \frac{1}{2}\int_{t_0}^{t_f} \Big(\big(y_{sp} - C\mathbb{S}(u)\big)^T Q \big(y_{sp} - C\mathbb{S}(u)\big) + u^T R u \Big) dt. \tag{2.31}$$

The objective function Φ is defined as:

$$\Phi = \sum_{i=1}^{N} \Phi_i, \tag{2.32}$$

where Φ_i is given as:

$$\Phi_i = \frac{1}{2}\int_{t_0}^{t_f} \big((y_{i\_sp} - C_{ii}x_i)^T Q_i (y_{i\_sp} - C_{ii}x_i) + u_i^T R_i u_i \big) dt.$$

It can be observed that Φ_i is the term in the objective function $\mathcal{J}_i$ (2.17), which is not affected by the price vector.

The state variables in subproblem (2.17) can be expressed as: $x_i = S_i(u_i, v_i)$, for $i = 1, ..., N$. Then, the objective function Φ_i is written in terms of the control inputs as follows:

$$\Phi_i(u_i, v_i) = \frac{1}{2}\int_{t_0}^{t_f} \big((y_{i\_sp} - C_{ii}S_i(u_i, v_i))^T Q_i(y_{i\_sp} - C_{ii}S_i(u_i, v_i)) + u_i^T R_i u_i\big)dt. \quad (2.33)$$

From (2.32) and (2.33), the objective function Φ can be expressed as a function of the input variables as follows:

$$\Phi(u, v) = \frac{1}{2}\sum_{i=1}^{N}\int_{t_0}^{t_f} \big((y_{i\_sp} - C_{ii}S_i(u_i, v_i))^T Q_i(y_{i\_sp} - C_{ii}S_i(u_i, v_i)) + u_i^T R_i u_i\big)dt. \quad (2.34)$$

Finally, the objective functions $\mathcal{J}_{cen}(u)$ in (2.31) and $\Phi(u, v)$ in (2.34) are differentiated to obtain:

$$\Psi = D^2\mathcal{J}_{cen}(u), \quad (2.35)$$

$$\Omega = diag(\Omega_i), \quad (2.36)$$

$$\text{with:} \quad \Omega_i = D^2\Phi(u_i), \quad \text{for } i = 1, ..., N. \quad (2.37)$$

where $\mathcal{J}_{cen}$ and Φ are assumed to be twice Fréchet differentiable and the input u belongs to a Hilbert space. The derivatives $D^2\mathcal{J}_{cen}(u)$ in (2.35) and $D^2\Phi(u_i)$ in (2.37) denote the second Fréchet derivatives of the functionals $\mathcal{J}_{cen}$ and Φ, respectively[4]. The operators Ψ in (2.35) and Ω in (2.36) are used to compute the convergence condition C1 in Theorem 2.4.1.

Theorem 2.4.1 *Consider that the prediction-driven coordination algorithm* (2.29)-(2.30) *is used to coordinate CDLQ controllers, and let us assume that:*

C1 : $W = \left(\Omega - \frac{\Psi}{2}\right)$ *is coercive*[5].

[4]The Fréchet derivative of a functional G at u_0, denoted by $DG(u_0)$, is defined as: $lim_{||h_1||\to 0}(||G(u_0 + h_1) - G(u_0) - DG(u_0)h_1||/||h_1||) = 0$. By defining $DG(u)h_1 = H(u)h_1$, the second Fréchet derivative is given by: $lim_{||h_2||\to 0}(||H(u_0+h_2) - H(u_0) - D^2G(u_0)(h_1, h_2)||/||h_2||) = 0$ (Guay, 1996).

[5]K is a coercive operator iff $\exists\ c > 0$ and $\forall x,\ \langle Kx, x\rangle \geq c\,\|x\|^2$.

Then:

- *the objective function $\mathcal{J}_{cen}(u^s)$ decreases monotonically as the number of iterations 's' increases;*
- *as the number of iterations 's' $\rightarrow \infty$, the control inputs u^s converge to the centralized optimal control inputs u^*. Thus, the solutions obtained with the CDLQ controllers are equal to the centralized optimal solution.*

Proof: Theorem 2.4.1 was presented and proved in (Cohen, 1977) assuming that the objective functions $\mathcal{J}_{cen}$ and Φ are twice Fréchet differentiable and convex in u. The proof presented in (Cohen, 1977) can be followed to prove the convergence behaviour of the prediction-driven coordination algorithm when it is used to coordinate CDLQ tracking controllers.

Remark 2.4.1 *Condition C1 in Theorem 2.4.1 can be computed off-line and it depends on the properties of the entire plant, as well as the weighting matrices; that is, $\Psi = \Psi(A, B, C, Q, R)$ and $\Omega_i = \Omega_i(A_{ii}, B_{ii}, C_{ii}, Q_i, R_i)$.*

If condition C1 in Theorem 2.4.1 is not satisfied for the proposed coordinated-distributed control formulation, this problem can be overcome by modifying the objective function $\mathcal{J}_i$ in (2.17)*. The modification can be accomplished by including the following quadratic 'prediction error term' in the objective function $\mathcal{J}_i$ of each subsystem:*

$$\frac{1}{2}\int_{t_0}^{t_f} \big((x_i - x_i^s)^T \gamma M_i (x_i - x_i^s) + (u_i - u_i^s)^T \gamma R_i (u_i - u_i^s)\big) dt, \tag{2.38}$$

where γ is a positive scalar and M_i is a symmetric positive definite matrix.

Quadratic terms similar to the integrand in (2.38) *have been used in the static optimization literature and they are referred to as the 'proximal term' (Rockafellar, 1976; Chen and Teboulle, 1994). The term* (2.38) *was also proposed in (Cohen, 1977) to ensure that the optimal solutions obtained with the local linear quadratic regulators achieve the centralized optimal solution.*

2.4.1 Proximal Method to Ensure Convergence of the Coordination Algorithm

When the term (2.38) is included in subproblem (2.17), the modified objective function becomes:

$$\begin{aligned}\mathcal{J}_i \quad = \quad & \frac{1}{2}\int_{t_0}^{t_f}\big((y_{i\_sp}-C_{ii}x_i)^T Q_i(y_{i\_sp}-C_{ii}x_i)+u_i^T R_i u_i\big)dt+ \\ & \int_{t_0}^{t_f}\Big(\sum_{j\neq i} p_j^{s^T}\big(A_{ji}x_i+B_{ji}u_i\big)\Big)dt+ \\ & \frac{1}{2}\int_{t_0}^{t_f}\big((x_i-x_i^s)^T\gamma M_i(x_i-x_i^s)+(u_i-u_i^s)^T\gamma R_i(u_i-u_i^s)\big)dt.\end{aligned} \tag{2.39}$$

The term (2.38) in modified objective function (2.39) has the purpose of penalizing the deviation of the optimal solutions calculated by the CDLQ controllers from the solutions predicted by the coordinator at each iteration step. As a result, the solutions calculated by the local CDLQ controllers and the ones predicted by the coordinator evolve in the same direction, which ensures convergence of the coordinated-distributed optimal solutions to the centralized optimal solution. The effect of the parameter γ on the prediction-driven coordination algorithm is discussed in Section 2.5.2 along with a method to tune the parameter γ.

CDLQ Control with Modified Objective Function

When the term (2.38) is included in the optimization problem of each subsystem, the CDLQ controllers as well as the information exchanged between the CDLQ controllers and the coordinator need to be adjusted. When subproblem (2.17) is solved using the modified objective function (2.39), it leads to the following differential Riccati equation:

$$\begin{cases}\dot{P}_i = -P_iA_{ii}-A_{ii}^TP_i+P_iB_{ii}\big((1+\gamma)R_i\big)^{-1}B_{ii}^TP_i-(C_{ii}^TQ_iC_{ii}+\gamma M_i),\\ P_i(t_f)=0,\end{cases} \tag{2.40}$$

the following differential equations:

$$\begin{cases} \dot{r}_i = \Big(P_i B_{ii}\big((1+\gamma)R_i\big)^{-1} B_{ii}^T - A_{ii}^T\Big)\, r_i + C_{ii}^T Q_i y_{i\_sp}, \\ r_i(t_f) = 0, \end{cases} \tag{2.41}$$

$$\begin{cases} \dot{z}_i = \Big(P_i B_{ii}\big((1+\gamma)R_i\big)^{-1} B_{ii}^T - A_{ii}^T\Big)\, z_i - P_i \left(\sum_{j\neq i} A_{ij} x_j^s + B_{ij} u_j^s\right) + \\ P_i B_{ii}\big((1+\gamma)R_i\big)^{-1} \left(-\gamma R_i u_i^s + \sum_{j\neq i} B_{ji}^T p_j^s\right) - \sum_{j\neq i} A_{ji}^T p_j^s + \gamma M_i x_i^s, \\ z_i(t_f) = 0. \end{cases} \tag{2.42}$$

and the following control input:

$$u_i = -\big((1+\gamma)R_i\big)^{-1}\Big(B_{ii}^T(P_i x_i + r_i + z_i) + \sum_{j\neq i} B_{ji}^T p_j^s - \gamma R_i u_i^s\Big). \tag{2.43}$$

The steps in the implementation of the CDLQ controllers (see Section 2.3.3) need to be adjusted as follows:

- The calculations performed off-line by each CDLQ controller involve integrating the differential Riccati equation (2.40) and the differential equation (2.41).
- With respect to the information exchanged between the coordinator and the CDLQ controllers, the coordinator needs to communicate x_i^s and u_i^s to each CDLQ controller i, apart from x_j^s and u_j^s and p_j^s (see steps 1 and 3 of the implementation of CDMPC controllers, Section 2.3.3).
- In step 2 of the implementation of the CDLQ controllers, the vector z_i can be obtained by integrating differential equation (2.42) and the inputs computed by each CDLQ controller, u_i, are given by (2.43).

2.5 Illustrative Examples

In this section, simulation studies are performed on two benchmark process systems to demonstrate the performance of the CDLQ controllers and the performance of the prediction-driven algorithm. The first illustrative example considers the evaporator process system described in (Fisher and Seborg, 1976) and it is used to show the performance of the CDLQ tracking controllers. The second illustrative example considers a simplified model of the evaporator process and it is used to show the effect of the proximal term parameter γ on the prediction-driven coordination algorithm.

2.5.1 Illustrative Example I

In this illustrative example, we consider the evaporator process system described in (Fisher and Seborg, 1976). A diagram of the evaporator process is shown in Figure 2.2.

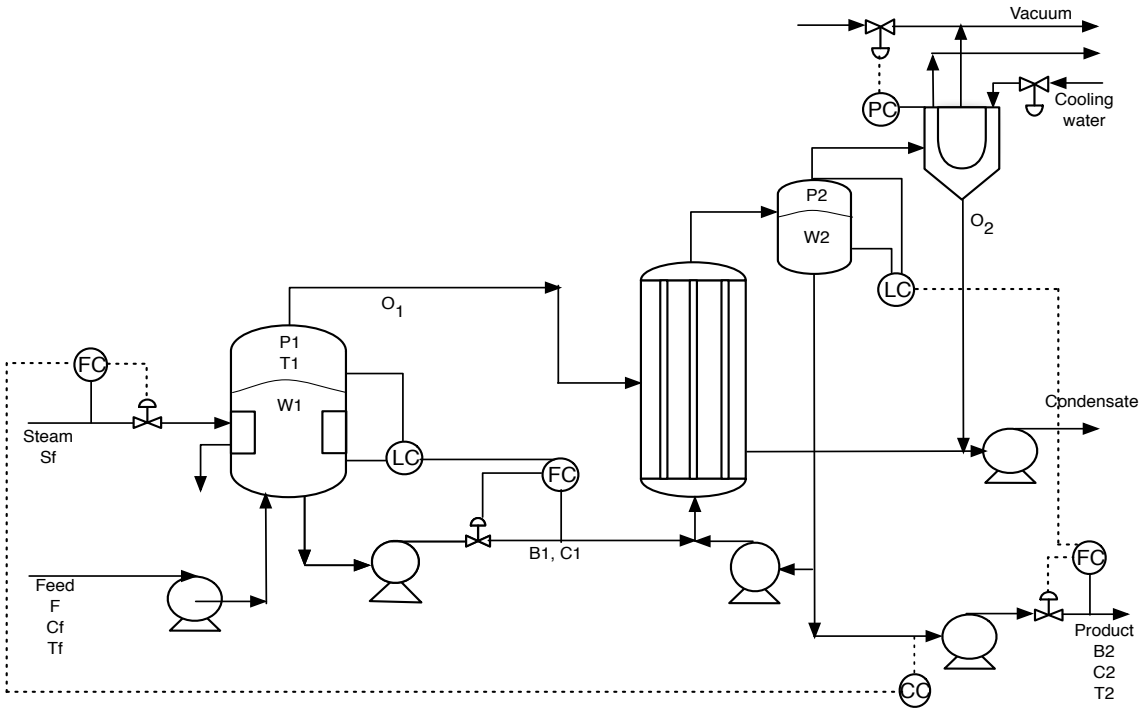

Figure 2.2: Evaporator process

The evaporator process includes five state variables, three input variables and three disturbance variables. The variables in the evaporator process are given in Table 2.1.

Set-point tracking performance is evaluated in this illustrative example. For the simulations performed in this section, it is assumed that all the state variables are measured and there are no disturbances entering the evaporator process. The dynamics of the evaporator are represented by the state-space model (2.10a)-(2.10c)

Table 2.1: Evaporator variables

State variables	Input variables	Disturbance variables
Solution holdup '$W1$' $[lb]$	Bottom flow rate '$B1$' $[lb/min]$	Feed flow rate 'F' $[lb/min]$
Solute concentration '$C1$' $[wt\%]$	Steam flow rate 'Sf' $[lb/min]$	Solute concentration 'Cf' $[wt\ \%]$
Liquid enthalpy '$h1$' $[Btu/lb]$	Bottom flow rate '$B2$' $[lb/min]$	Liquid enthalpy '$h1$' $[Btu/lb]$
Solution holdup '$W2$' $[lb]$		
Solute concentration '$C2$' $[wt\%]$		

with:

$$x = [W1,\ C1,\ h1,\ W2,\ C2]^T, \quad u = [B1,\ Sf,\ B2]^T, \quad y = x,$$

$$A = \begin{bmatrix} 0 & -0.00156 & -0.1711 & 0 & 0 \\ 0 & -0.1419 & 0.1711 & 0 & 0 \\ 0 & -0.00875 & -1.102 & 0 & 0 \\ 0 & -0.00128 & -0.1489 & 0 & 0.00013 \\ 0 & 0.0605 & 0.1489 & 0 & -0.0591 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.143 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.392 & 0 \\ 0.108 & 0 & -0.0592 \\ 0.0486 & 0 & 0 \end{bmatrix}, \qquad C = I_5.$$

It is also assumed that the evaporator process consists of two subsystems. The parameters used in this simulation study are listed in Table 2.2.

Table 2.2: Parameters used in illustrative example I

	Subsystem 1	Subsystem 2
State variables	$x_1 = [W1,\ C1]^T$	$x_2 = [h1,\ W2,\ C2]^T$
Input variables	$u_1 = B1$	$u_2 = [Sf,\ B2]^T$
Controlled variables	$y_1 = x_1$	$y_2 = x_2$
Initial conditions	$x_1(0) = [0,\ 0]^T$	$x_2(0) = [0,\ 0,\ 0]^T$
Weighting matrices	$Q_1 = diag(2, 0)$	$Q_2 = diag(0, 2, 2)$
	$R_1 = 2$	$R_2 = diag(5, 5)$

A set-point change of magnitude 1.0 in controlled variable '$W2$' was performed at the initial time, while the set-points for the remaining controlled variables were left at the origin; that is, $y_{1\_sp} = [0, 0]^T$ and $y_{2\_sp} = [0, 1, 0]^T$. Figure 2.3 - Figure 2.5 show the simulation results based on 100 minutes simulation. The simulation results are shown in deviation variable form.

Figure 2.3 shows the trajectories of the controlled variables '$W1$', '$W2$' and '$C2$', and the input variables obtained with the CDLQ controllers. It can be observed in Figure 2.3 that the CDLQ controllers drive the controlled variables '$W1$', '$W2$' and '$C2$' to their desired set-points. It can also be observed in Figure 2.3 that the trajectories obtained with the CDLQ controllers are equal to the ones obtained with a centralized controller[6]. Therefore, the performance of the CDLQ controllers achieve the performance of the centralized controller. Convergence of the CDLQ solutions to the centralized optimal solution is also shown in Figure 2.4.

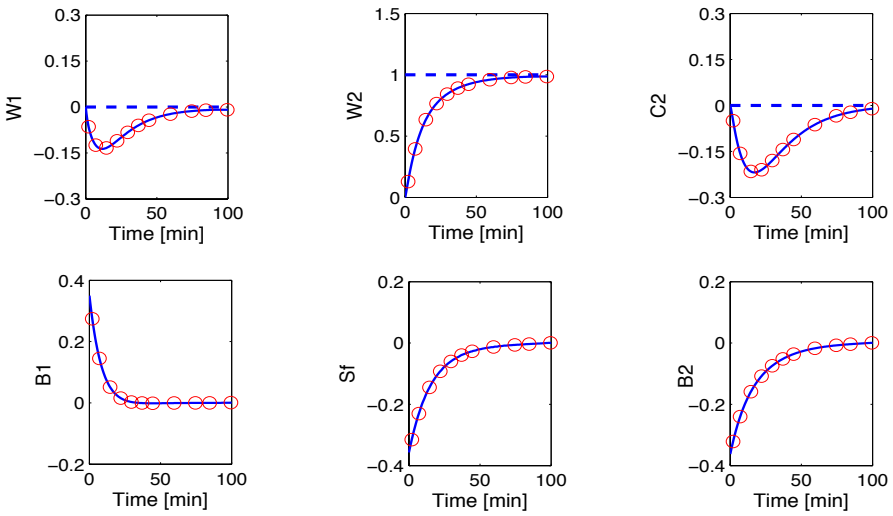

Figure 2.3: Trajectories of the controlled variables '$W1$', '$W2$' and '$C2$', and the input variables obtained with the CDLQ controllers (solid line), and with a centralized controller (circles); desired set-points (dashed line)

[6]The simulations obtained with a centralized controller were performed with the same initial conditions used for the CDLQ controllers and with the simulation parameters provided in Table 2.2. Nevertheless, the centralized controller was designed by solving the centralized control problem (2.11)-(2.12).

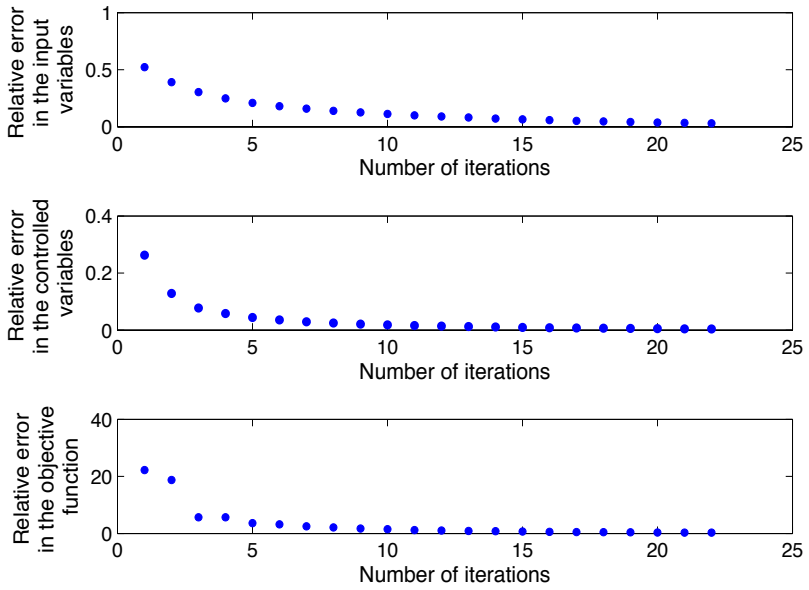

Figure 2.4: Relative errors in the input variables, in the controlled variables and in the objective function

Figure 2.4 shows the relative errors in the input variables, in the controlled variables and in the objective function through the iterations. These relative errors are calculated as: $(\|u^s - u^*\| / \|u^*\|)$ for the input variables, $(\|y^s - y^*\| / \|y^*\|)$ for the controlled variables and $(\|\mathcal{J}_{cen}(u^s) - \mathcal{J}_{cen}(u^*)\| / \|\mathcal{J}_{cen}(u^*)\|)$ for the objective function. The centralized optimal solutions are denoted with the superscript (*). It can be observed that the errors decrease as the number of iterations increase. Moreover, it can be observed that the prediction-driven coordination algorithm converges within 20 iterations. This indicates that, at convergence, the performance of the CDLQ controllers resulted in the centralized optimal performance.

Finally, Figure 2.5 shows the prices of the prediction-driven coordination algorithm at convergence. The number of components in the price vectors p_i^s are equal to the number of state variables x_i in each subsystem i. Therefore, the vector p_1^s has two components ($p_{1\_1}^s$ and $p_{1\_2}^s$), as shown in Figure 2.5a) and vector p_2^s has three components ($p_{2\_1}^s$, $p_{2\_2}^s$ and $p_{2\_3}^s$), as shown in Figure 2.5b). Both price vectors p_1^s and p_2^s converge to the origin at the end of the simulation time, as given by the final

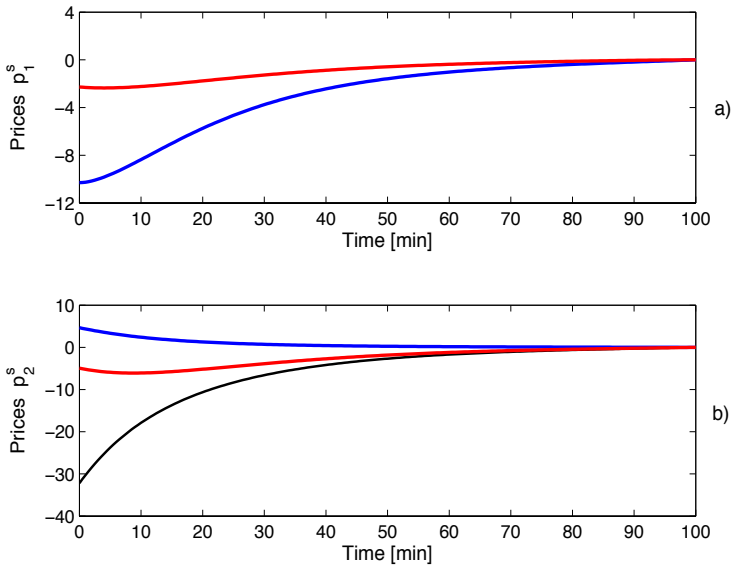

Figure 2.5: Prices p_1^s and p_2^s: a) $p_{1\_1}^s$ (blue) and $p_{1\_2}^s$ (red); b) $p_{2\_1}^s$ (blue), $p_{2\_2}^s$ (black) and $p_{2\_3}^s$ (red)

condition in the price update mechanism (2.30).

2.5.2 Illustrative Example II

To show the effect of the proximal term parameter γ on the prediction-driven coordination algorithm, a simulation experiment was performed using a simplified model of the evaporator process[7]. The simplified model of the evaporator process is given in (Newell and Lee, 1989) and it includes three input variables (product flow rate '$F2$', steam pressure '$P100$', cooling water flowrate '$F200$'), three state variables to be controlled (separator level '$L2$', product composition '$X2$', and operating pressure '$P2$') and five disturbance variables (circulating flow rate '$F3$', feed flow rate '$F1$', feed concentration '$X1$', feed temperature '$T1$' and cooling water inlet tmperature '$T200$'). In this illustrative example, it is assumed that there are no disturbances entering the evaporator process. Thus, the dynamics of the evaporator with no disturbances are

[7]The simulation results in illustrative example II have been presented and published in (Dallagi, A., N.I. Marcos and J. F. Forbes, 2008).

represented by state-space model (2.10a)-(2.10c) with:

$$
\begin{aligned}
&x = [L2,\ X2,\ P2]^T, \quad u = [F2,\ P100,\ F200]^T, \quad y = x, \\
&A = \begin{bmatrix} 0 & 0.10445 & 0.37935 \\ 0 & -0.1 & 0 \\ 0 & -0.10340 \times 10^{-1} & -0.54738 \times 10^{-1} \end{bmatrix}, \\
&B = \begin{bmatrix} -0.1 & 0.37266 & 0 \\ -0.1 & 0 & 0 \\ 0 & 0.36914 \times 10^{-1} & -0.75272 \times 10^{-2} \end{bmatrix}, \qquad C = I_3.
\end{aligned}
\tag{2.44}
$$

It is assumed that the evaporator process consists of two subsystems. The parameters used in this simulation study are listed in Table 2.3.

Table 2.3: Parameters used in illustrative example II

	Subsystem 1	Subsystem 2
State variables	$x_1 = [L2,\ X2]^T$	$x_2 = P2$
Input variables	$u_1 = [F2,\ P100]^T$	$u_2 = F200$
Controlled variables	$y_1 = x_1$	$y_2 = x_2$
Weighting matrices	$Q_1 = diag(1,1),\ R_1 = diag(100,100)$	$Q_2 = 1,\ R_2 = 100$

In this illustrative example, the regulation problem is considered; thus, $y_{1\_sp} = [0,\ 0]^T$, $y_{2\_sp} = 0$. The initial conditions for the state variables are: $x_1(0) = [1,\ 1]^T$ and $x_2(0) = 1$. The proximal term (see equation (2.38)) was included in the objective function of each subsystem to ensure convergence of the prediction-driven algorithm. The effect of the parameter γ on the coordinated-distributed control problems was studied for the evaporator process (Newell and Lee, 1989) and the simulation results are shown in Table 2.4.

For the given weighting matrices Q, R and simulation time t_f, there is a set of parameters $\gamma \geq \gamma_0$, which ensures convergence of the prediction-driven coordination algorithm. Any positive value for γ smaller than γ_0 will not satisfy condition C1 in Theorem 2.4.1[8]. A simple method for tuning the parameter γ is by 'trail and error'. A small value for γ can be selected initially and convergence condition C1 should

[8]Each objective function Φ_i in (2.33) needs to be modified to include the proximal term (2.38).

be checked for this initial value of γ. If condition C1 is not satisfied, the value of γ should be increased until a value of γ that satisfies condition C1 is found.

It is shown in Table 2.4 that for $\gamma \leq \gamma_0 = 0.5$ the coordination algorithm diverges. Furthermore, it can be observed in Table 2.4 that as the value of γ increases, the prediction-driven coordination algorithm requires more iterations to converge[9].

Table 2.4: Effect of the parameter γ on the prediction-driven coordination algorithm

Parameter γ	Number of Iterations
< 0.5	algorithm diverges
1	29
2	45
5	85
10	145

The effect of the parameter γ on the prediction-driven coordination algorithm can be interpreted by analyzing the objective function (2.39). It was discussed in Section 2.4 that the term (2.38) in the objective function (2.39) has the purpose of penalizing the deviation of the optimal solutions calculated by the CDLQ controllers from the solutions predicted by the coordinator at each iteration step. As the parameter γ increases, the CDLQ controllers will compute their optimal solutions with small deviations with respect to the predictions performed by the coordinator. Therefore, as γ increases, the coordination algorithm becomes more sluggish and it requires more iterations to converge to the centralized optimal solution. This effect should be taken into account when choosing the value for the parameter γ.

2.6 Summary

In this chapter, a CDLQ control scheme is proposed and developed for large-scale continuous-time, linear dynamic systems. This work extends the control scheme presented in (Cohen, 1977) to formulate CDLQ controllers that steer controlled variables to their desired set-points. In the CDLQ control scheme, local CDLQ controllers are coordinated using a prediction-driven coordination algorithm.

[9]The simulation results in Table 2.4 are taken from (Dallagi, A., N.I. Marcos and J. F. Forbes, 2008).

Convergence properties of the prediction-driven coordination algorithm are studied, and a convergence condition for the coordination algorithm is provided. If the convergence condition is satisfied, the following results are obtained:

- convergence of the prediction-driven coordination algorithm is guaranteed;
- the objective function $\mathcal{J}_{cen}(u^s)$ decreases monotonically as the number of iterations 's' increases;
- as the number of iterations 's' $\rightarrow \infty$, the control inputs u^s converge to the centralized optimal control inputs u^*. Therefore, the performance of the CDLQ controllers is equal to the centralized controller's performance.

If the convergence condition is not satisfied, there exists a remedy for this problem. A proximal term can be incorporated in the objective functions of each subsystem to ensure convergence of the coordination algorithm. One of the tuning parameters in the proximal term is the constant scalar γ. It is shown with a benchmark process system that there exist a minimum value γ_0, such that for any $\gamma \geq \gamma_0$ the convergence condition is satisfied. The effect of γ on the coordination algorithm is also discussed. It can be observed from the simulation results in Section 2.5.2 that as the value of γ is increased above γ_0, the coordination algorithm is slowed down. Understanding this effect is beneficial in order to tune the parameter γ.

Overall, the results and discussions presented in this chapter indicate that CDLQ control is a promising strategy for control of interacting dynamic systems. The CDLQ control scheme can lead to the centralized optimal operations and still maintain a flexible control framework. For example, the proposed CDLQ control framework can switch from the coordinated-distributed scheme to the decentralized control scheme, if needed. This can be done by setting the prices p_i and the interacting variables x_i^s and u_i^s to be equal to zero, and by interrupting the communications between the coordinator and the local CDLQ controllers. Similarly, the local CDLQ controllers can be individually removed from or added to the coordination loop.

3

Coordinated-Distributed MPC via Prediction-Driven Coordination

This chapter presents a state-feedback coordinated-distributed model predictive control (CDMPC) scheme for discrete-time, unconstrained linear dynamic systems. In the proposed CDMPC control scheme, local CDMPC controllers are coordinated via the prediction-driven method. The prediction-driven coordination algorithm proposed in this chapter is obtained by adapting the coordination algorithm presented in Chapter 2. Convergence of the coordination algorithm is shown along with a stability analysis of the closed-loop system under CDMPC control. Finally, a simulation example is used to illustrate the effectiveness of the proposed coordinated-distributed control scheme.

3.1 Model Predictive Control: Background

Model predictive control (MPC) is a model based optimal control technique that has been successfully implemented in process industries and has received a considerable

amount of interest among academic researchers for more than three decades. Model predictive control techniques use a model of the process to predict the future plant behaviour. The aim of the MPC control calculations is to determine a sequence of control moves so that the predicted process variables achieve their desired values or set-points in an optimal manner (Garcia *et al.*, 1989; Rossiter, 2005; Maciejowski, 2002). To determine the optimal control moves, an objective function is optimized over a specified time horizon. The objective function in MPC generally takes the form of a quadratic function that penalizes the control efforts and the difference between the output trajectories from their desired set-points.

The basic idea of MPC is shown in Figure 3.1 for a single-input single-output process (Seborg, 1989). Four key variables are shown in Figure 3.1: the measured output variable y, the predicted output variable $\hat{y}$, the past input u applied to the process and the calculated input variable $\hat{u}$. At the current sampling instant 'k', the MPC controller computes a set of H_u values of the input $\{\hat{u}(k+l-1|k),\ l = 1, 2, .., H_u\}$. The input is held constant after the H_u control moves. The inputs are calculated so that a set of H_p predicted outputs $\hat{y} = \{\hat{y}(k+l|k),\ l = 1, 2, .., H_p\}$ reaches the set-point in an optimal manner. The number of control moves H_u is referred to as the control horizon and the number of predictions H_p is referred to as the prediction horizon (Seborg, 1989).

One characteristic of an MPC control strategy is its receding horizon nature. Although a sequence of H_u control moves is calculated at each sampling instant, only the first control move of the calculated sequence is applied to the process, while the rest of the calculated control moves are discarded. At the next sampling instant, the optimization is repeated with current measurements to calculate a new sequence of control moves, but again only the first calculated control move is implemented. This procedure is repeated at every sampling instant (Camacho and Bordons, 1999; Maciejowski, 2002).

The benefits of MPC technology are well proven. The main advantages of MPC can be summarized as follows (Maciejowski, 2002; Qin and Badgwell, 2003):

- it handles multivariable control problems;

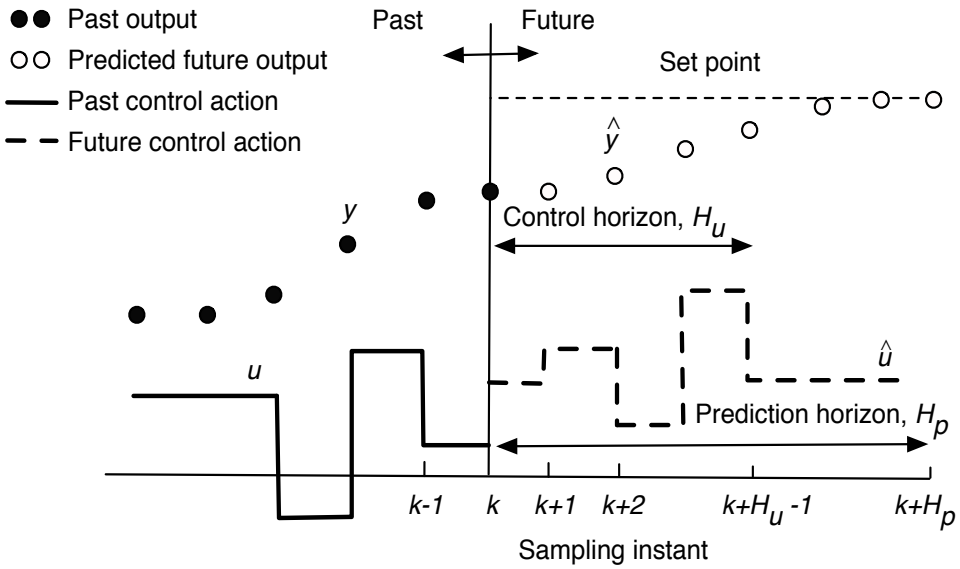

Figure 3.1: Model predictive control strategy

- it drives some process variables to their desired set-points, while maintaining other process variables within specified ranges;
- it allows operation closer to constraints, which frequently leads to more profitable operation;
- it can take actuator limitations into account.

Although MPC has been widely accepted, its application is often limited to single process units or a single piece of processing equipment. Plant-wide applications of MPC are not common due to their lack of flexibility and lack of resiliency. The development of schemes that use MPC controllers to achieve plant-wide optimal performance is highly desirable. Such development can help process industries improve plant-wide performance operations significantly, increase economic benefits, and, most importantly, ensure environmental and safety conditions. One control scheme that uses MPC technology and can lead to plant-wide optimal performance is proposed in this chapter and it is referred to as coordinated-distributed model predictive control (CDMPC). An illustration of the CDMPC control hierarchy for a plant with two subsystems is shown in Figure 3.2.

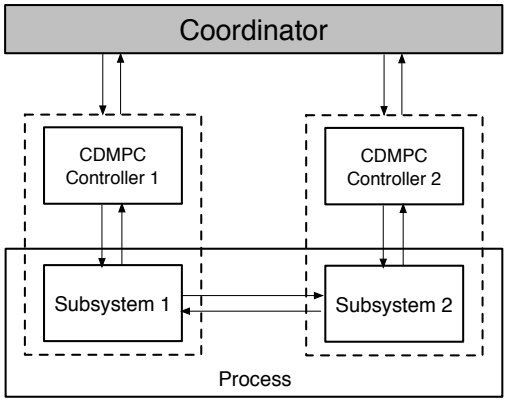

Figure 3.2: Illustration of CDMPC control hierarchy for a plant with two subsystems

The CDMPC control scheme is constructed using a coordinator and with minor modifications to the standard local MPC controllers. A prediction-driven coordination method is used in the proposed CDMPC control scheme. In the prediction-driven method, the coordinator predicts future values for the subsystems' state variables. In addition, the coordinator computes a price vector, which is required for the coordination of the CDMPC controllers. Note that in this chapter, the price vector is computed as an approximation of the Lagrange multipliers of the centralized optimization problem. Examples of hierarchical control schemes using local Lagrange multipliers as prices can be found in (Cohen, 1977; Jamshidi, 1983) for linear quadratic regulators and in (Negenborn *et al.*, 2008) for multi-agent MPC controllers applied to transportation networks.

The main contributions of this chapter can be summarized as follows:

- First, a novel state-feedback CDMPC control scheme is formulated for discrete-time, linear unconstrained dynamic systems. The prediction-driven method used in this chapter is an extension of the method presented in Chapter 2 to discrete-time dynamic systems.

- Then, a convergence analysis for the prediction-driven coordination algorithm is provided. In the convergence analysis, the behaviour of the coordination algorithm is shown and convergence of the coordinated-distributed optimal solutions to the centralized optimal solution is proven.

- Finally, closed-loop stability under CDMPC control is shown to be guaranteed even when the coordination algorithm is stopped before the algorithm convergence is achieved. To the best of our knowledge, this is the first study where stability of the closed-loop system under coordinated-distributed control has been proven. While the bulk of the work in the area of distributed control considers cooperative distributed control schemes, and stability of the closed-loop system using cooperative distributed controllers has been proven (e.g., Venkat *et al.* (2006)), no work has been done to prove stability of the closed-loop system when distributed controllers communicate with a coordinator.

3.2 Problem Description

Let us consider the entire large-scale system or plant, which is adequately modeled by a linear discrete-time state-space representation:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ x(0) &= x^{init}, \end{aligned} \tag{3.1}$$

where $x(k) \in \Re^n$ and $u(k) \in \Re^q$ denote the vector of state and input variables at time step 'k', respectively. A typical approach for control of a large-scale system such as (3.1) is to partition the entire process model into smaller subsystems and control each subsystem using a local decentralized controller. If the entire process (3.1) is partitioned into N subsystems, the dynamics for each subsystem i can be represented by:

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{j\neq i}\big(A_{ij}x_j(k) + B_{ij}u_j(k)\big), \\ x_i(0) &= x_i^{init}, \end{aligned} \tag{3.2}$$

where $x_i(k) \in \Re^{n_i}$ and $u_i(k) \in \Re^{q_i}$. The matrices A_{ii} and B_{ii} in (3.2) represent the local subsystem dynamics, for $i = 1, \ldots, N$. It is assumed that the pair (A_{ii}, B_{ii}) is controllable. The term '$\sum_{j\neq i}\big(A_{ij}x_j(k) + B_{ij}u_j(k)\big)$' in (3.2) represents the effect of the interactions; that is, the effect of the state variables and the input variables from subsystems j (with $j \neq i$) on the local subsystem i. It can be noted that the total number of state variables in the plant is $n = \sum_{i=1}^{N} n_i$ and the total number of input variables in the plant is $q = \sum_{i=1}^{N} q_i$.

The CDMPC controllers proposed in this work can be constructed with minor modifications to the local decentralized controllers. Thus, a decentralized MPC control formulation for subsystem (3.2) is presented in Section 3.3 as an initial step for the development of the CDMPC controllers in Section 3.4.

3.3 Unconstrained Decentralized MPC Control

Let us assume that the true dynamics of subsystem i (for $i = 1, \ldots, N$) are represented by model (3.2). For the decentralized control formulation, it is further assumed that:

- the length of the prediction horizon (H_p) is the same for all the subsystems. This is also assumed for the control horizon (H_u), with $H_u \leq H_p$;
- the same control intervals are used in all the MPC controllers in the plant;
- the optimal values for the variables x_i and u_i are calculated by the local MPC controller i;
- the decentralized controllers existing in the plant (previous to the design of the coordinated-distributed controllers) provide stable operations for the entire process;
- at time 'k', all the states $x_i(k)$ are measured for $i = 1, \ldots, N$.

In the decentralized control scheme considered in this work, it is assumed that all of the information available at time 'k' can be used to predict the future values for the local state trajectories. Current measurements of the state variables $x_j(k)$ and past values of the input variables $u_j(k-1)$, for $j \neq i$, are used as feedforward information by the local MPC controller i. The feedforward action is included in the decentralized MPC control scheme in an effort to minimize the unavoidable losses of performance associated with the decentralized approach.

Based on the assumptions previously stated, the variables predicted by decentralized MPC controller i along the prediction and control horizons are given

by:

$$\begin{cases} \hat{x}_i(k+l+1|k) = A_{ii}\hat{x}_i(k+l|k) + B_{ii}\hat{u}_i(k+l|k) + \\ \qquad\qquad \sum_{j\neq i}\big(\beta_i A_{ij}x_j(k) + B_{ij}u_j(k-1)\big), \quad \text{for } l = 0, ..., H_p - 1, \\ \hat{x}_i(k|k) = x_i(k), \\ \beta_i = 1, \quad \text{for } l = 0, \\ \beta_i = 0, \quad \text{for } l = 1, ..., H_p - 1, \end{cases} \tag{3.3}$$

with:

$$\begin{cases} \hat{u}_i(k+l|k) = \sum_{\nu=0}^{l} \Delta\hat{u}_i(k+\nu|k) + u_i(k-1), \quad \text{for } l = 0, ..., H_p - 1, \\ \Delta\hat{u}_i(k+l|k) = 0, \qquad H_u \leq l \leq H_p - 1. \end{cases} \tag{3.4}$$

Process model (3.3) and (3.4) can be arranged in a matrix form for the entire prediction and control horizons as follows:

$$G_{ii}Z_i(k) = g_i, \tag{3.5}$$

$$Z_i(k) = [X_i(k)^T, \Delta U_i(k)^T]^T, \tag{3.6}$$

where the vectors of predicted states and predicted input moves are defined as:

$$X_i(k) = [\hat{x}_i(k+1|k)^T, \hat{x}_i(k+2|k)^T, ..., \hat{x}_i(k+H_p|k)^T]^T,$$
$$\Delta U_i(k) = [\Delta\hat{u}_i(k|k)^T, \Delta\hat{u}_i(k+1|k)^T, ..., \Delta\hat{u}_i(k+H_u-1|k)^T]^T.$$

The matrix G_{ii} is defined as:

$$G_{ii} = \Bigg[\underbrace{\begin{array}{ccccc} I_{n_i} & 0 & 0 & \dots & 0 \\ -A_{ii} & I_{n_i} & 0 & \dots & 0 \\ 0 & \dots & & \ddots & \vdots \\ 0 & \dots & 0 & -A_{ii} & I_{n_i} \end{array}}_{G_{A\_ii}} \Bigg| \underbrace{\begin{array}{ccccc} -B_{ii} & 0 & 0 & \dots & 0 \\ -B_{ii} & -B_{ii} & 0 & \dots & 0 \\ \dots & \dots & & \ddots & \vdots \\ -B_{ii} & -B_{ii} & \dots & -B_{ii} & -B_{ii} \end{array}}_{G_{B\_ii}} \Bigg], \tag{3.7}$$

where the dimensions of G_{A_ii} and G_{B_ii} are $((H_p \cdot n_i) \times (H_p \cdot n_i))$ and $((H_p \cdot n_i) \times (H_u \cdot q_i))$, respectively. The matrix I_{n_i} in (3.7) is the identity matrix of dimensions $n_i \times n_i$. The vector g_i in equation (3.5) is defined as:

$$g_i = \left.\begin{bmatrix} \sum_{j=1}^{N} A_{ij}x_j(k) + B_{ij}u_j(k-1) \\ \sum_{j=1}^{N} B_{ij}u_j(k-1) \\ \dots \\ \sum_{j=1}^{N} B_{ij}u_j(k-1) \end{bmatrix}\right\} \begin{array}{l} \text{repeated} \\ (H_p - 1) \text{ times,} \end{array} \tag{3.8}$$

and it includes measured data, such as the effect of current state variables and past input variables from the plant.

Assuming that the vector of desired state values to be tracked in each subsystem (X_{i_sp}) is provided, then the decentralized MPC control problem can be formulated for each subsystem i as follows:

$$\begin{aligned} &\min_{X_i, \Delta U_i} \ \Phi_i = \frac{1}{2}\Big(\big(X_{i\_sp} - X_i(k)\big)^T \mathcal{Q}_i \big(X_{i\_sp} - X_i(k)\big) + \Delta U_i(k)^T \mathcal{R}_i \Delta U_i(k)\Big) \\ &\text{subject to :} \\ &\qquad G_{ii} Z_i(k) = g_i. \end{aligned} \tag{3.9}$$

The matrices $\mathcal{Q}_i$ and $\mathcal{R}_i$ in (3.9) are defined as $\mathcal{Q}_i = diag(Q_i(1), Q_i(2), \ldots, Q_i(H_p))$ and $\mathcal{R}_i = diag(R_i(0), R_i(1), \ldots, R_i(H_u - 1))$, where the weighting matrices $Q_i(.) \in \Re_+^{n_i \times n_i}$ and $R_i(.) \in \Re_+^{q_i \times q_i}$ are positive definite matrices with their off-diagonal elements equal to zero.

3.4 Unconstrained Coordinated-Distributed MPC Control

In this section, the CDMPC control scheme is formulated. In Section 3.4.1, the process models used in the decentralized MPC controllers are modified to incorporate the effect of the interactions between the subsystems. In Section 3.4.2, the decentralized objective function is augmented by an extra term, which includes a price and is required for coordination of the subsystems. The price update mechanism is presented in Section 3.4.3 along with the implementation of the CDMPC controllers in Section 3.4.4.

3.4.1 Process Models used in the CDMPC Controllers

In order to design the CDMPC controllers, equations (3.3) and (3.5) need to be modified. Process model (3.3) is modified to include the interactions between

subsystems as follows:

$$\begin{cases} \hat{x}_i(k+l+1|k) = A_{ii}\hat{x}_i(k+l|k) + B_{ii}\hat{u}_i(k+l|k) + \\ \qquad\qquad \sum_{j\neq i}\big(\beta_i A_{ij}x_j(k) + B_{ij}u_j(k-1)\big) + \hat{v}_i(k+l|k), \\ \qquad\qquad \text{for } l = 0, ..., H_p - 1, \\ \hat{x}_i(k|k) = x_i(k), \\ \beta_i = 1, \quad \text{for } l = 0, \\ \beta_i = 0, \quad \text{for } l = 1, ..., H_p - 1, \end{cases} \tag{3.10}$$

where $\hat{u}_i(k+l|k)$ is defined as in equation (3.4). The variables $\hat{v}_i$ in (3.10) represent the effect of predicted $\hat{x}_j^s$ and $\Delta\hat{u}_j^s$ on the local subsystem i, where $j \neq i$. The variables $\hat{v}_i$ are not decision variables in the local optimization problem of each subsystem i and they are given by:

$$\begin{cases} \hat{v}_i(k+l|k) = \sum_{j\neq i}\beta_i A_{ij}\hat{x}_j^s(k+l|k) + \\ \qquad\qquad \sum_{j\neq i} B_{ij}\big(\sum_{\nu=0}^{l}\Delta\hat{u}_j^s(k+\nu|k)\big), \quad \text{for } l = 0, ..., H_p - 1, \\ \beta_i = 0, \quad \text{for } l = 0, \\ \beta_i = 1, \quad \text{for } l = 1, ..., H_p - 1, \\ \Delta\hat{u}_j^s(k+l|k) = 0, \qquad H_u \leq l \leq H_p - 1. \end{cases} \tag{3.11}$$

The superscript 's' denotes the iteration step in the coordination algorithm as the algorithm converges to a solution during a control interval. It is appended to $\hat{x}_j$ and $\Delta\hat{u}_j$ to indicate that these variables are predicted by a coordinator at iteration step 's' (see Section 3.4.3 and Section 3.4.4). Finally, the model for the variables in local subsystem i, including the predicted effect of the interactions from the other subsystems, can be obtained from (3.10) and (3.11) as follows:

$$G_{ii}Z_i(k) = g_i - V_i(k),$$

where $Z_i(k)$, G_{ii} and g_i are defined in (3.6), (3.7) and (3.8), respectively and

$$V_i(k) = \sum_{j\neq i} G_{ij}Z_j^s(k), \tag{3.12}$$

$$Z_j^s(k) = [X_j^s(k)^T, \Delta U_j^s(k)^T]^T. \tag{3.13}$$

The state variables predicted by the coordinator are given by:

$$X_j^s(k) = [\hat{x}_j^s(k+1|k)^T, \hat{x}_j^s(k+2|k)^T, ..., \hat{x}_j^s(k+H_p|k)^T]^T,$$

and the vector of input control moves $\Delta U_j^s(k)$ is given by:

$$\Delta U_j^s(k) = [\Delta \hat{u}_j^s(k|k)^T, \Delta \hat{u}_j^s(k+1|k)^T, ..., \Delta \hat{u}_j^s(k+H_u-1|k)^T]^T.$$

The matrix G_{ij} is defined as:

$$G_{ij} = \left[\underbrace{\begin{array}{ccccc} 0 & 0 & 0 & \dots & 0 \\ -A_{ij} & 0 & 0 & \dots & 0 \\ 0 & \dots & & \ddots & \vdots \\ 0 & \dots & 0 & -A_{ij} & 0 \end{array}}_{G_{A\_ij}} \vdots \underbrace{\begin{array}{ccccc} -B_{ij} & 0 & 0 & \dots & 0 \\ -B_{ij} & -B_{ij} & 0 & \dots & 0 \\ \dots & \dots & & \ddots & \vdots \\ -B_{ij} & -B_{ij} & \dots & -B_{ij} & -B_{ij} \end{array}}_{G_{B\_ij}} \right], \quad (3.14)$$

where the dimensions of G_{A_ij} and G_{B_ij} are $((H_p \cdot n_i) \times (H_p \cdot n_j))$ and $((H_p \cdot n_i) \times (H_u \cdot q_j))$, respectively.

3.4.2 Formulation of CDMPC Controllers

The CDMPC controller for each subsystem i $(i = 1, ..., N)$ is formulated as follows:

$$\begin{aligned} &\min_{X_i, \Delta U_i} \ \mathcal{J}_i = \Phi_i + p_i^{s^T} \Theta_i [X_i(k)^T, \Delta U_i(k)^T]^T \\ &\text{subject to :} \\ &\qquad G_{ii} Z_i(k) = g_i - V_i(k), \end{aligned} \quad (3.15)$$

where the objective function Φ_i is defined in (3.9). The objective function in optimal control problem (3.15) differs from the objective function in (3.9) by the term '$p_i^{s^T} \Theta_i [X_i(k)^T, \Delta U_i(k)^T]^T$'. This term is only a minor modification to the objective function of decentralized control problem (3.9) and it is needed to coordinate the optimal solutions of the local subsystems in order to achieve the centralized optimal solution (see proof of Theorem 3.5.1). The elements of the price vector 'p_i^s' are computed by a coordinator based on the price update mechanism described in Section 3.4.3.

The matrix Θ_i in (3.15) takes into account the effect of the local decision variables

on the other subsystems. For example, for $i = 1$, $i = 2$ and $i = N$, Θ_i becomes:

$$\Theta_1 = \begin{bmatrix} 0 \\ \hline G_{21} \\ \hline \cdots \\ \hline G_{N1} \end{bmatrix}, \qquad \Theta_2 = \begin{bmatrix} G_{12} \\ \hline 0 \\ \hline \cdots \\ \hline G_{N2} \end{bmatrix} \quad \text{and} \quad \Theta_N = \begin{bmatrix} G_{1N} \\ \hline G_{2N} \\ \hline \cdots \\ \hline 0 \end{bmatrix}. \tag{3.16}$$

The matrices G_{ji} in (3.16), with $j \neq i$, can be constructed by inverting the subscripts 'j' and 'i' in equation (3.14). Finally, optimal control problem (3.15) can be reformulated in a compact form for each subsystem i as follows:

$$\begin{aligned} &\min_{Z_i} \ \mathcal{J}_i = \frac{1}{2}\Big(Z_i(k)^T \Upsilon_i Z_i(k)\Big) + \Big(\phi_i^T + p_i^{s^T}\Theta_i\Big) Z_i(k) \\ &\text{subject to :} \\ &\qquad G_{ii} Z_i(k) = g_i - V_i(k), \end{aligned} \tag{3.17}$$

where $\Upsilon_i = diag(\mathcal{Q}_i, \mathcal{R}_i)$ and $\phi_i^T = -[X_{i\_sp}^T \mathcal{Q}_i, \ \underbrace{0, ..., 0}_{1\times(H_u \times q_i)}]$.

3.4.3 Prediction-Driven Coordination Algorithm

In this section, a prediction-driven algorithm is developed to coordinate local CDMPC controllers. The prediction-driven algorithm formulated in this section is adapted from the prediction-driven coordination algorithm derived in Chapter 2. As discussed in Chapter 2, the prediction-driven algorithm includes two types of calculations, which are performed by the coordinator at each iteration step 's'. In the first type of calculations, the coordinator predicts the subsystems' state variables using the entire plant process model. In the second type of calculations, the coordinator updates the price vector. The price vector is required by the subsystems to solve their own optimal control problems.

Prediction of the state variables:

The states $X_i^s(k)$ for the subsystems are predicted by the coordinator according to

the mathematical model for the entire process:

$$\begin{bmatrix} G_{A\_11} & \dots & G_{A\_1N} \\ G_{A\_21} & \dots & G_{A\_2N} \\ \dots & \ddots & \vdots \\ G_{A\_N1} & \dots & G_{A\_NN} \end{bmatrix} \begin{bmatrix} X_1^s(k) \\ X_2^s(k) \\ \dots \\ X_N^s(k) \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \dots \\ g_N \end{bmatrix} - \begin{bmatrix} G_{B\_11} & \dots & G_{B\_1N} \\ G_{B\_21} & \dots & G_{B\_2N} \\ \dots & \ddots & \vdots \\ G_{B\_N1} & \dots & G_{B\_NN} \end{bmatrix} \begin{bmatrix} \Delta U_1^s(k) \\ \Delta U_2^s(k) \\ \dots \\ \Delta U_N^s(k) \end{bmatrix}, \quad (3.18)$$

where the control moves $\Delta U_i^s(k)$ in (3.18) are equal to the optimal control moves $\Delta U_i(k)$ calculated by each CDMPC controller i, for $i = 1, ..., N$. The matrices G_{A_ii}, G_{B_ii}, G_{A_ij} and G_{B_ij} in (3.18) are defined in equations (3.7) and (3.14).

Price update:

To coordinate the control problems for the subsystems, a coordinator calculates the full price vector λ_{coor}^s. The price vector p_i^s required in optimization problem (3.17) is a subset of the full price vector λ_{coor}^s. The price vector λ_{coor}^s is calculated by solving the set of linear equations:

$$G^T \lambda_{coor}^s = -\Big(\Upsilon Z_{coor}^s(k) + \phi\Big), \quad (3.19)$$

provided that the entire system has been modeled to ensure that the matrix G is full rank. Equation (3.19) is an approximation of equation (3.24) and it is required in the coordination algorithm to ensure that the centralized optimal solution is achieved (see proof of Theorem 3.5.1).

Matrix G in equation (3.19) is constructed by arranging the model coefficients of all the subsystems, as follows:

$$G = \begin{bmatrix} G_{11} & G_{12} & \dots & G_{1N} \\ G_{21} & G_{22} & \dots & G_{2N} \\ \dots & \vdots & \ddots & \vdots \\ G_{N1} & G_{N2} & \dots & G_{NN} \end{bmatrix}.$$

The matrix Υ and the vector ϕ in (3.19) are denoted as $\Upsilon = diag(\Upsilon_1, \Upsilon_2, ..., \Upsilon_N)$ and $\phi = [\phi_1^T, \phi_2^T, \dots, \phi_N^T]^T$, respectively. The vector $Z_{coor}^s(k)$ in (3.19) is defined as:

$$Z_{coor}^s(k) = \left[X_1^s(k)^T, \Delta U_1^s(k)^T, X_2^s(k)^T, \Delta U_2^s(k)^T, ..., X_N^s(k)^T, \Delta U_N^s(k)^T\right]^T. \quad (3.20)$$

The coordinator's tasks at each iteration step 's' (for $s \geq 1$) can be summarized as follows[1]:

1. the coordinator predicts the value of the state variables $X_i^s(k)$ according to equation (3.18) and the input moves calculated by the subsystems ($\Delta U_i^s(k) = \Delta U_i(k)$), for $i = 1, ..., N$;

2. the coordinator constructs the vector $Z_{coor}^s(k)$ using the predicted states $X_i^s(k)$ and the input moves $\Delta U_i^s(k)$, for $i = 1, ..., N$;

3. the coordinator computes the full price vector λ_{coor}^s by solving the set of linear equations (3.19). The vector λ_{coor}^s is arranged as $\lambda_{coor}^s = [\lambda_1^{s^T}, \lambda_2^{s^T}, ..., \lambda_N^{s^T}]^T$, where the elements λ_i^s have dimensions $((H_p \cdot n_i) \times 1)$, for $i = 1, \ldots, N$;

4. the coordinator calculates the price vector p_i^s for each subsystem i, for $i = 1, ..., N$. The price p_i^s is then communicated to each subsystem, where it is used to solve the local optimization problem (3.17). The price p_i^s required in the local optimization problem (3.17) includes all the elements of the price vector λ^s, except the components λ_i^s. For example, for $i = 1$, $i = 2$ and $i = N$, p_i^s becomes:

$$\begin{aligned} p_1^s &= [0^T, \lambda_2^{s^T}, ..., \lambda_N^{s^T}]^T, \\ p_2^s &= [\lambda_1^{s^T}, 0^T, ..., \lambda_N^{s^T}]^T, \text{ and} \\ p_N^s &= [\lambda_1^{s^T}, \lambda_2^{s^T}, ..., 0^T]^T. \end{aligned} \tag{3.21}$$

Steps 1 to 4, described above, include all the calculations carried out by the coordinator at each iteration step in order to update the price vector p_i^s for each subsystem. In Section 3.4.4, all the steps in the coordination algorithm are described along with the implementation of the CDMPC controllers.

3.4.4 Implementation of the CDMPC Controllers

As described in Section 3.1, one of the features of the MPC control is its receding horizon approach. For the implementation of a standard MPC controller, a sequence

[1]Step $s = 0$ is the initialization step in the prediction-driven algorithm and it is described in Section 3.4.4.

of H_u control input moves is calculated at each sampling instant by optimizing an objective criterion over a specified time horizon. Only the first control input move of the calculated sequence is applied to the process. Then, a new sequence is calculated at the next sampling instant, after new measurements become available; only the first control input move is implemented. This procedure is repeated at each sampling instant (Camacho and Bordons, 1999; Maciejowski, 2002).

In CDMPC control, the implementation steps of standard MPC are kept, but the coordinator imposes an additional step. During a single control interval and before the first calculated control input move is implemented, the coordinator exchanges information with the local CDMPC controllers. During this data exchange, the coordinator iteratively adjusts the price vector to drive the calculated optimal solution of the subsystems to the centralized optimal solution. The information exchanged between the coordinator and the local CDMPC controllers is shown in Figure 3.3. The implementation of the CDMPC controllers is carried out according to the following steps:

1. **Initialization**[2] **($s = 0$):** The coordinator sets an initial value for the vector $Z^0_{coor}(k)$ (see equation (3.20)) and for the full price vector λ^0_{coor}. For initialization, the elements of the vectors $Z^0_{coor}(k)$ and λ^0_{coor} can be set to zero. The coordinator sends the predicted variables $Z^0_j(k)$, with $j \neq i$ (see equation (3.13)), and the corresponding price vector p^0_i (see equation (3.21)) to each local CDMPC controller.

2. **Optimization performed by each local CDMPC controller:** Based on the information provided by the coordinator, each local CDMPC controller calculates the interacting variables $V_i(k)$ (see equation (3.12))[3], and solves the local optimization problem (3.17). The optimal control input moves

[2]Initializing the prediction-driven algorithm with $Z^0_{coor}(k) = 0$ and $\lambda^0_{coor} = 0$ allows the solutions obtained with the CDMPC controllers at iteration step s=0 to be equal to the solutions obtained by the decentralized controllers. Recall that the existing decentralized controllers are assumed to provide stable operations for the entire process.

[3]Alternatively, the interacting variables $V_i(k)$ could be computed by the coordinator, rather than being computed by each local CDMPC controller. If the variables $V_i(k)$ were computed by the coordinator, model maintenance would be easier. Nevertheless, it would increase the computational load of the coordinator.

calculated by each CDMPC controller i, $\Delta U_i(k)$, are communicated back to the coordinator.

3. **Price update:** The iteration counter 's' is incremented. The coordinator collects the control input moves $\Delta U_i(k)$ calculated by each CDMPC controller. Then, the coordinator predicts the states $X_i^s(k)$ for each subsystem based on the control input moves $\Delta U_i^s(k) = \Delta U_i(k)$, with $i = 1, \ldots, N$, and using the entire process model (3.18). Next, the coordinator constructs the vector $Z_{coor}^s(k)$ (see equation (3.20)). Using $Z_{coor}^s(k)$ and price update mechanism (3.19), the coordinator calculates the full price vector λ_{coor}^s. The coordinator sends the predicted variables $Z_j^s(k)$ with $j \neq i$ (see equation (3.13)), and the corresponding p_i^s to each local CDMPC controller.

4. **Iteration until convergence:** Steps (2)-(3) are repeated until the prediction-driven coordination algorithm is terminated. The prediction-driven coordination algorithm is terminated when $\|\Delta U^{s+1}(k) - \Delta U^s(k)\| \leq \epsilon$, where ϵ is a specified error tolerance and the vector $\Delta U^s(k)$ is defined as $\Delta U^s(k) = \left[\Delta U_1^s(k)^T, \Delta U_2^s(k)^T, ..., \Delta U_N^s(k)^T\right]^T$.

5. **Implementation of the first calculated control move:** Once the prediction-driven coordination algorithm is terminated, the first calculated control input moves $\Delta \hat{u}_i(k|k)$ are implemented in each subsystem i, while the rest of the calculated control input moves are discarded. The iteration counter 's' in the coordination algorithm is reset to zero and the coordinated optimization problems (steps (1)-(4)) are initiated again for the next receding horizon.

3.5 Coordination Algorithm Convergence Analysis

In this section, a convergence analysis for the proposed prediction-driven coordination algorithm is presented. Theorem 3.5.1 shows that at convergence, the solutions

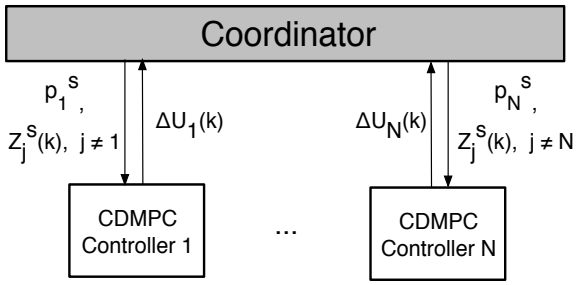

Figure 3.3: Communication between the coordinator and the local CDMPC controllers

obtained with the CDMPC controllers are equivalent to the centralized optimal solution. Theorem 3.5.2 describes the behaviour of the prediction-driven method as the coordination algorithm evolves throughout the iterations.

3.5.1 Coordination Algorithm Accuracy

In this section, the ability of the CDMPC controllers to achieve the centralized optimal solution is shown. This can be accomplished by first defining the centralized control problem. The centralized control problem can be formulated as:

$$\begin{aligned} &\min_{Z} \ \mathcal{J}_{cen} = \sum_{i=1}^{N} \Phi_i \\ &\text{subject to :} \\ &\qquad GZ(k) = g, \end{aligned} \tag{3.22}$$

where Φ_i is defined in (3.9). The vector g is defined as $g = [g_1^T, g_2^T, \ldots, g_N^T]^T$ and the elements of the vector $Z(k)$ are arranged in the same order as the elements in $Z^s_{coor}(k)$ (see equation (3.20)).

Theorem 3.5.1 *Consider that the entire process system* (3.1) *is controlled by* N *coordinated-distributed MPC controllers* (3.17) *that are coordinated using the prediction-driven coordination algorithm* (3.18)-(3.19). *When* $\|\Delta U^{s+1}(k) - \Delta U^s(k)\| \to 0$, *then the optimal solutions obtained with the CDMPC controllers converge to the centralized optimal solution.*

Proof:

Centralized control problem:

For the purpose of this proof, and to compare the centralized optimal solution with the solutions obtained with the CDMPC controllers, let us re-write centralized control problem (3.22) as follows:

$$\begin{aligned} &\min_{Z} \ \mathcal{J}_{cen} = \frac{1}{2}\Big(Z(k)^T \Upsilon Z(k)\Big) + \phi^T Z(k) \\ &\text{subject to :} \\ &\qquad GZ(k) = g. \end{aligned} \tag{3.23}$$

The Lagrangian $\mathcal{L}_{cen}$ for the centralized control problem (3.23) can be formulated as:

$$\mathcal{L}_{cen} = \frac{1}{2}\Big(Z(k)^T \Upsilon Z(k)\Big) + \phi^T Z(k) + \lambda^T \big(GZ(k) - g\big).$$

Then, the first-order optimality conditions can be written as:

$$\Upsilon Z^*(k) + \phi + G^T \lambda^* = 0, \tag{3.24}$$

$$GZ^*(k) - g = 0, \tag{3.25}$$

where $Z^*(k)$ and λ^* represent the optimal values for $Z(k)$ and λ, respectively.
In equation (3.24), the matrix G can be written as: $G = \bar{G} + \bar{\Theta}$, with $\bar{G} = diag(G_{11}, G_{22}, \ldots, G_{NN})$ and $\bar{\Theta} = [\Theta_1 \nvdash \Theta_2 \nvdash \ldots \nvdash \Theta_N]$. Then, equation (3.24)-(3.25) can be re-written as:

$$\Upsilon Z^*(k) + \phi + \bar{G}^T \lambda^* + \bar{\Theta}^T \lambda^* = 0, \tag{3.26}$$

$$\bar{G} Z^*(k) + \bar{\Theta} Z^*(k) - g = 0. \tag{3.27}$$

Coordinated-distributed control problems:

Next, let us calculate the first-order optimality conditions for the subproblems. The Lagrangian $\mathcal{L}_i$ for subproblem (3.17), for $i = 1, \ldots, N$, is:

$$\mathcal{L}_i = \frac{1}{2}\Big(Z_i(k)^T \Upsilon_i Z_i(k)\Big) + \Big(\phi_i^T + p_i^{s^T} \Theta_i\Big) Z_i(k) + \lambda_i^T \big(G_{ii} Z_i(k) + V_i(k) - g_i\big).$$

Therefore, the first-order optimality conditions for subproblem i becomes:

$$\Upsilon_i Z_i(k) + \phi_i + \Theta_i^T p_i^s + G_{ii}^T \lambda_i = 0, \tag{3.28}$$

$$G_{ii} Z_i(k) + V_i(k) - g_i = 0. \tag{3.29}$$

The first-order optimality conditions (3.28)-(3.29) can be written for all the subproblems as:

$$\Upsilon Z_{CDMPC}(k) + \phi + \bar{G}^T \lambda_{CDMPC} + \bar{\Theta}^T \lambda^s_{coor} = 0, \tag{3.30}$$

$$\bar{G} Z_{CDMPC}(k) + \bar{\Theta} Z^s_{coor}(k) - g = 0, \tag{3.31}$$

where $Z_{CDMPC}(k) = \left[Z_1(k)^T, Z_2(k)^T, \ldots, Z_N(k)^T\right]^T$ and $\lambda_{CDMPC} = \left[\lambda_1^T, \lambda_2^T, \ldots, \lambda_N^T\right]^T$.

The coordination algorithm converges once $\|\Delta U^{s+1}(k) - \Delta U^s(k)\|$ has been driven to zero. Due to the structure of the prediction-driven algorithm (3.18)-(3.19) and optimal control problem (3.17), when the coordination algorithm converges, the variables predicted by the coordinator are equal to the centralized optimal solutions. That is: $\lambda^s_{coor} = \lambda^*$ and $Z^s_{coor}(k) = Z^*(k)$. Then, equations (3.30)-(3.31) become:

$$\Upsilon Z_{CDMPC}(k) + \phi + \bar{G}^T \lambda_{CDMPC} + \bar{\Theta}^T \lambda^* = 0, \tag{3.32}$$

$$\bar{G} Z_{CDMPC}(k) + \bar{\Theta} Z^*(k) - g = 0. \tag{3.33}$$

It can be shown from equations (3.26)-(3.27) and (3.32)-(3.33) that $Z_{CDMPC}(k) = Z^*(k)$ and $\lambda_{CDMPC} = \lambda^*$, provided that the entire system has been modeled to ensure that the column-rank$(\bar{G}^T)$ = dim(λ^*). As a result, the optimal solutions obtained with the CDMPC controllers achieve the optimal solutions obtained with a centralized controller. This proves that, in order to achieve the centralized optimal solution, the term '$p_i^{s^T} \Theta_i Z_i(k)$' is required in the optimal control problems of the subsystems, using p_i^s for the price vector in the local control problems, and λ^s_{coor} as the full price vector.

□

3.5.2 Coordination Algorithm Behaviour

Theorem 3.5.2 is presented in this section to describe the behaviour of the prediction-driven coordination algorithm throughout the iterations. A convergence condition is provided in Theorem 3.5.2 and it can be computed off-line to verify whether the coordination algorithm will converge for the selected controller tunings. The objective functions $\Phi(\Delta U(k), V(k))$ and $\mathcal{J}_{cen}(\Delta U(k))$ are required to compute the convergence condition in Theorem 3.5.2, and they are defined as shown below.

The objective function Φ is defined as:

$$\Phi = \sum_{i=1}^{N} \Phi_i, \tag{3.34}$$

where Φ_i is given in (3.9). The predicted states in subproblem (3.17) for the specified prediction and control horizons can be expressed as: $X_i(k) = S_i(\Delta U_i(k), V_i(k))$, for $i = 1, ..., N$. Then, the objective function Φ_i is written in terms of the control input moves as follows:

$$\begin{aligned}\Phi_i(\Delta U_i(k), V_i(k)) = \tfrac{1}{2}\Big(\big(X_{i\_sp} - S_i(\Delta U_i(k), V_i(k))\big)^T \mathcal{Q}_i \big(X_{i\_sp} - S_i(\Delta U_i(k), V_i(k))\big) + \\ \Delta U_i(k)^T \mathcal{R}_i \Delta U_i(k)\Big).\end{aligned} \tag{3.35}$$

From (3.34) and (3.35), the objective function Φ can be expressed as a function of the control input moves as follows:

$$\begin{aligned}\Phi(\Delta U(k), V(k)) = \tfrac{1}{2}\sum_{i=1}^{N} \Big(\big(X_{i\_sp} - S_i(\Delta U_i(k), V_i(k))\big)^T \mathcal{Q}_i \big(X_{i\_sp} - S_i(\Delta U_i(k), V_i(k))\big) + \\ \Delta U_i(k)^T \mathcal{R}_i \Delta U_i(k)\Big).\end{aligned} \tag{3.36}$$

Let $X_{_{CDMPC}}(k)$ be the vector that contains all the predicted state variables $X_i(k)$ from the distributed control problem (3.17), for $i = 1, ..., N$. That is, $X_{_{CDMPC}}(k) = [X_1(k)^T, X_2(k)^T, \ldots, X_N(k)^T]^T$. Then, the vector $X_{_{CDMPC}}(k)$ can be written as: $X_{_{CDMPC}}(k) = S(\Delta U(k), V(k))$, with $\Delta U(k) = \big[\Delta U_1(k)^T, \Delta U_2(k)^T, ..., \Delta U_N(k)^T\big]^T$ and $V(k) = \big[V_1(k)^T, V_2(k)^T, ..., V_N(k)^T\big]^T$. Let K be the mapping from the set of $\Delta U(k)$ to the set of interacting variables $V(k)$, that is $V(k) = K(\Delta U(k))$. Then, the vector of all predicted states can be written as:

$$X_{_{CDMPC}}(k) = S(\Delta U(k), K(\Delta U(k))). \tag{3.37}$$

Let us, also, express the predicted states for centralized control problem (3.22) as a function of the control input moves for the specified prediction and control horizons, as follows:

$$X(k) = \mathbb{S}(\Delta U(k)), \tag{3.38}$$

where the elements of the vector $X(k)$ are arranged in the same order as the elements of the vector $X_{_{CDMPC}}(k)$. Next, the centralized objective function $\mathcal{J}_{cen}$ in (3.22) can be expressed in terms of the control input moves as:

$$\mathcal{J}_{cen}(\Delta U(k)) = \tfrac{1}{2}\Big(\big(X_{sp} - \mathbb{S}(\Delta U(k))\big)^T \mathcal{Q} \big(X_{sp} - \mathbb{S}(\Delta U(k))\big) + \Delta U(k)^T \mathcal{R} \Delta U(k)\Big), \tag{3.39}$$

where $X_{sp} = [X_{1\_sp}^T, X_{2\_sp}^T, \ldots, X_{N\_sp}^T]^T$, $\mathcal{Q} = diag(\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_N)$ and $\mathcal{R} = diag(\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_N)$. Finally, the objective functions $\mathcal{J}_{cen}(\Delta U(k))$ in (3.39) and $\Phi(\Delta U(k), V(k))$ in (3.36) are differentiated to obtain:

$$\Psi = \frac{d^2 \mathcal{J}_{cen}}{d\Delta U(k)^2}, \tag{3.40}$$

$$\Omega = diag(\Omega_i), \tag{3.41}$$

$$\text{with:} \quad \Omega_i = \frac{\partial^2 \Phi}{\partial \Delta U_i(k)^2}, \quad \text{for } i = 1, ..., N.$$

The matrices Ψ in (3.40) and Ω in (3.41) are used to compute the convergence condition C1 in Theorem 3.5.2.

Theorem 3.5.2 *Consider that the prediction-driven coordination algorithm* (3.18)-(3.19) *is used to coordinate CDMPC controllers, and let us assume that:*

C1 : $W = \left(\Omega - \frac{\Psi}{2}\right)$ *is positive definite.*

Then:

- *the objective function* $\mathcal{J}_{cen}(\Delta U^s(k))$ *decreases monotonically as the number of iterations 's' increases;*
- *as the number of iterations 's'* $\to \infty$*, the control input moves* $\Delta U^s(k)$ *converge to the centralized optimal control input moves* $\Delta U^*(k)$*.*

Proof[4]**:**

It can be shown from equations (3.37) and (3.38) that for any given $\Delta U(k)$:

$$\mathbb{S}(\Delta U(K)) = S(\Delta U(k), K(\Delta U(k))). \tag{3.42}$$

Then, from equations (3.36), (3.39) and (3.42), we obtain[5]:

$$\mathcal{J}_{cen}(\Delta U(k)) = \Phi(\Delta U(k), V(k)), \tag{3.43}$$

[4]Theorem 3.5.2 is developed in this chapter as an extension of the convergence analysis provided in (Cohen, 1977) for coordination of continuous-time linear quadratic regulators. The proof for Theorem 3.5.2 could be inferred from (Cohen, 1977). Nevertheless, all the steps in the proof are included in this chapter because some of the intermediate results in the proof are required for the stability analysis presented in Section 3.6.2.

[5]To compute the centralized objective function $\mathcal{J}_{cen}(\Delta U(k))$ in (3.43) and the CDMPC objective function $\mathcal{J}_i(\Delta U_i(k))$ in (3.45), the reduced space of their decision variables is considered.

and:

$$d\mathcal{J}_{cen}/d\Delta U(k) = \partial\Phi/\partial\Delta U(k) + (\partial\Phi/\partial V(k))(dV(k)/d\Delta U(k)). \tag{3.44}$$

The objective function $\mathcal{J}_i$ for each CDMPC control problem (3.15) can be written as:

$$\begin{cases} \mathcal{J}_i(\Delta U_i(k)) & = \Phi_i(\Delta U_i(k), V_i(k)) + \Lambda_i(\Delta U^s(k))\Delta U_i(k), \\ \Lambda_i(\Delta U^s(k)) & = (\partial\Phi/\partial V(k))(\partial V(k)/\partial\Delta U_i(k))|_{\Delta U^s(k)}, \end{cases} \tag{3.45}$$

where $\Lambda_i(\Delta U^s(k))\Delta U_i(k) = p_i^{s^T}\Theta_i[X_i(k)^T, \Delta U_i(k)^T]^T$ for optimization problem (3.15). Note that for a given $\Delta U^s(k)$, the optimal solution for each subproblem with objective function (3.45) is $\Delta U_i(k) = \Delta U_i^{s+1}(k)$. That is, $d\mathcal{J}_i/d\Delta U_i(k)|_{\Delta U_i^{s+1}(k)} = 0$.

Next, let us perform a Taylor series expansion[6] of the centralized objective function $\mathcal{J}_{cen}(\Delta U(k))$ in (3.39) around the centralized optimum $\Delta U^*(k)$. By taking $\mathcal{J}_{cen}^{'}(\Delta U(k)) = d\mathcal{J}_{cen}/d\Delta U(k)$, we obtain:

$$(\mathcal{J}_{cen}^{'}(\Delta U(k)))^T = \Psi(\Delta U(k) - \Delta U^*(k)), \tag{3.46}$$

where $\Psi = d^2\mathcal{J}_{cen}/d\Delta U(k)^2$. We can then evaluate $\mathcal{J}_{cen}^{'}(\Delta U(k))$ at $\Delta U(k)$ for iteration step 's' to obtain:

$$(\mathcal{J}_{cen}^{'}(\Delta U^s(k)))^T = \Psi(\Delta U^s(k) - \Delta U^*(k)). \tag{3.47}$$

Let us perform a Taylor series expansion of the subsystems' objective function $\mathcal{J}_i$ around optimum $\Delta U_i^{s+1}(k)$. Then, by taking $\mathcal{J}_i^{'}(\Delta U_i(k)) = d\mathcal{J}_i/d\Delta U_i(k)$, we get:

$$(\mathcal{J}_i^{'}(\Delta U_i(k)))^T = \frac{d^2\mathcal{J}_i}{d\Delta U_i(k)^2}(\Delta U_i(k) - \Delta U_i^{s+1}(k)). \tag{3.48}$$

It can be observed from (3.45) that $d^2\mathcal{J}_i/d\Delta U_i(k)^2 = d^2\Phi_i/d\Delta U_i(k)^2$. Let us define $\Omega_i = d^2\Phi_i/d\Delta U_i(k)^2 = \partial^2\Phi/\partial\Delta U_i(k)^2$. Then, equation (3.48) becomes:

$$(\mathcal{J}_i^{'}(\Delta U_i(k)))^T = \Omega_i(\Delta U_i(k) - \Delta U_i^{s+1}(k)).$$

Next, we can evaluate $\mathcal{J}_i^{'}(\Delta U_i(k))$ at $\Delta U_i(k)$ for iteration step 's' to obtain:

$$(\mathcal{J}_i^{'}(\Delta U_i^s(k)))^T = \Omega_i(\Delta U_i^s(k) - \Delta U_i^{s+1}(k)). \tag{3.49}$$

[6]In this section, the Taylor series expansions include up to the quadratic term because they are used to approximate quadratic objective functions. The remainder term in the Taylor series expansion is neglected.

Let us express objective function $\mathcal{J}_{CDMPC}$ as the objective function that includes the contributions from all the subsystems. That is, $\mathcal{J}_{CDMPC} = \sum_{i=1}^{N} \mathcal{J}_i$. Then, using equation (3.49) for $i = 1, ..., N$, we can calculate $\mathcal{J}'_{CDMPC}(\Delta U^s(k)))$ as follows:

$$(\mathcal{J}'_{CDMPC}(\Delta U^s(k)))^T = \Omega(\Delta U^s(k) - \Delta U^{s+1}(k)), \tag{3.50}$$

where $\Omega = diag(\Omega_1, \Omega_2, ..., \Omega_N)$. Then, from equations (3.44), (3.45), (3.47) and (3.50), we get:

$$\Delta U^s(k) - \Delta U^{s+1}(k) = \Omega^{-1}\Psi(\Delta U^s(k) - \Delta U^*(k)). \tag{3.51}$$

Finally, let us perform a Taylor series expansion of the centralized objective function $\mathcal{J}_{cen}(\Delta U(k))$ around control input move $\Delta U^{s+1}(k)$, and let us use equation (3.46) evaluated at $\Delta U^{s+1}(k)$ and equation (3.51) to obtain:

$$\mathcal{J}_{cen}(\Delta U^s(k)) - \mathcal{J}_{cen}(\Delta U^{s+1}(k)) = (\Delta U^s(k) - \Delta U^*(k))^T \Psi \Omega^{-1} \Big(\Omega - \tfrac{\Psi}{2}\Big) \Omega^{-1} \Psi (\Delta U^s(k) - \Delta U^*(k)). \tag{3.52}$$

If $W = (\Omega - \frac{\Psi}{2})$ is positive definite, then $\mathcal{J}_{cen}(\Delta U^s(k))$ decreases monotonically with the number of iterations 's' for $\Delta U^s(k) \neq \Delta U^*(k)$. In addition,

$$\mathcal{J}_{cen}(\Delta U^s(k)) - \mathcal{J}_{cen}(\Delta U^{s+1}(k)) \geq \sigma_{min}(W) \|\Omega^{-1}\Psi(\Delta U^s(k) - \Delta U^*(k))\|^2, \tag{3.53}$$

where $\sigma_{min}(W) > 0$ is the minimum eigenvalue of the positive definite matrix W. Then, as $s \to \infty$, $\Delta U^s(k) \to \Delta U^*(k)$. Furthermore, as $s \to \infty$, $\mathcal{J}_{cen}(\Delta U^\infty(k)) = \mathcal{J}_{cen}(\Delta U^*(k))$ and inequality (3.53) has an upper bound given by:

$$\mathcal{J}_{cen}(\Delta U^s(k)) - \mathcal{J}_{cen}(\Delta U^*(k)) \geq \mathcal{J}_{cen}(\Delta U^s(k)) - \mathcal{J}_{cen}(\Delta U^{s+1}(k)).$$

□

Remark 3.5.1 *Condition C1 in Theorem 3.5.2 can be computed off-line and it depends on the properties of the entire plant, as well as weighting matrices; that is,* $\Psi = \Psi(A, B, \mathcal{Q}, \mathcal{R})$ *and* $\Omega_i = \Omega_i(A_{ii}, B_{ii}, \mathcal{Q}_i, \mathcal{R}_i)$.

If condition C1 in Theorem 3.5.2 is not satisfied for the proposed coordinated-distributed control formulation, this can be overcome by modifying the objective

function $\mathcal{J}_i$ in (3.17)*. The modification can be accomplished by including the following quadratic 'prediction error term' in the objective function $\mathcal{J}_i$ of each subsystem:*

$$\tfrac{1}{2}\Big((X_i^s(k)-X_i(k))^T\gamma\mathcal{Q}_i(X_i^s(k)-X_i(k))+(\Delta U_i^s(k)-\Delta U_i(k))^T\gamma\mathcal{R}_i(\Delta U_i^s(k)-\Delta U_i(k))\Big), \quad (3.54)$$

where γ is a positive scalar.

As discussed in Chapter 2, quadratic terms similar to (3.54) *have been used in the static optimization literature and they are referred to as the 'proximal term' (Rockafellar, 1976; Chen and Teboulle, 1994). A term similar to* (3.54) *was also proposed in (Cohen, 1977) to ensure that the optimal solutions obtained with the local linear quadratic regulators achieve the centralized optimal solution. The term* (3.54) *has the same effect in the prediction-driven coordination algorithm as discussed in Chapter 2 and the method for tuning the parameter γ described in Chapter 2 also applies for the parameter γ in the proximal term* (3.54)*.*

When the proximal term (3.54) is included in the optimization problem of each subsystem, the information exchanged between the CDMPC controllers and the coordinator needs to be adjusted. If the term (3.54) is included in the objective function (3.17), the coordinator needs to communicate $Z_i^s(k)$ to each subsystem apart from $Z_j^s(k)$ and p_i^s (see steps 1 and 3 of the implementation of CDMPC controllers, Section 3.4.4).

3.6 Stability Analysis

In this section, the stability of the nominal closed-loop system under state-feedback coordinated-distributed control is studied. The term nominal system refers to the case when there are no disturbances affecting the process, and the process dynamics are perfectly represented by the process model. For simplicity and without loss of generality, the regulation control problem will be used, where the set-point $X_{i\_sp}=0$, for $i=1,\ldots,N$, and the state variables in each subsystem i are driven to the origin.

Two distinct cases are considered for the stability analysis. The first case shows the stability of the closed-loop system when the coordination algorithm is allowed to iterate until its convergence is achieved (Section 3.6.1). The second case studies

the stability of the closed-loop system when the coordination algorithm is stopped before it converges and the optimal solution obtained with the CDMPC controllers is confined to a set that belongs to a neighborhood of the centralized optimal solution (Section 3.6.2).

3.6.1 Stability upon Convergence of the Coordination Algorithm

In this section, the stability of the entire closed-loop system under state-feedback coordinated-distributed control is studied for the case when the prediction-driven coordination algorithm is allowed to iterate within each control interval until convergence is achieved.

First, an infinite (or sufficiently long) prediction horizon is considered for the formulation of the CDMPC controllers. Then, it is shown that for subsystems with a particular form of interactions, the formulation of the CDMPC controllers using infinite prediction horizon can be written as a finite horizon problem with a terminal penalty.

3.6.1.1 Infinite prediction horizon

Here, the stability of the entire closed-loop system under state-feedback coordinated-distributed control is shown for the CDMPC controllers constructed using an infinite prediction horizon. An infinite prediction horizon implies a prediction horizon sufficiently long to ensure that the optimal solutions calculated by the local CDMPC controllers do not change from one sampling instant to the next one. This idea invokes *Bellman's principle of optimality* (Bellman, 1957), which states that the "tail" of any optimal trajectory is itself the optimal trajectory from its starting point (Maciejowski, 2002).

Theorem 3.6.1 *Consider that the entire process system* (3.1) *is controlled by* N *coordinated-distributed MPC controllers* (3.17) *that are coordinated using the prediction-driven coordination algorithm* (3.18)-(3.19). *Let us assume that the following conditions are satisfied:*

C1 : A is stable;

C2 : weighting matrices $Q_i(.) = Q_i$ and $R_i(.) = R_i$ are positive definite and constant throughout the prediction and control horizons, for $i = 1, \ldots N$;

C3 : the control interval is sufficiently long to ensure that the prediction-driven coordination algorithm converges.

Then, the entire closed-loop system under CDMPC control is asymptotically stable.

Proof: As defined in Section 3.5.2, the objective function $\mathcal{J}_{CDMPC}$ includes the contributions from all the subsystems. That is, $\mathcal{J}_{CDMPC} = \sum_{i=1}^{N} \mathcal{J}_i$. Let $\mathcal{J}^*_{CDMPC}(k)$ be the optimal value of objective function $\mathcal{J}_{CDMPC}(k)$ at time 'k'. To prove stability under coordinated-distributed control, it is sufficient to show that $\mathcal{J}^*_{CDMPC}(k)$ is a Lyapunov function for the entire closed-loop system.

If conditions C1, C2 and C3 in Theorem 3.6.1 are satisfied then, within each control interval, the optimal solutions obtained with the coordinated-distributed control scheme converge to the centralized optimal solution. Considering the nominal case, the objective function $\mathcal{J}^*_{CDMPC}(k)$ can be written for the regulatory control problem as:

$$\mathcal{J}^*_{CDMPC}(k) = \frac{1}{2} \sum_{i=1}^{N} \Big(\sum_{l=1}^{\infty} ||\hat{x}^*_i(k+l|k)||^2_{Q_i} + \sum_{l=1}^{H_u} ||\Delta\hat{u}^*_i(k+l-1|k)||^2_{R_i} \Big), \quad (3.55)$$

where $\hat{x}^*_i$ and $\Delta\hat{u}^*_i$ denote the optimal values for the predicted state variables $\hat{x}_i$ and predicted control moves $\Delta\hat{u}_i$, for $i = 1, ..., N$.

At time '$k+1$' the new optimization problem is solved for the CDMPC control scheme, with initial condition $x_i(k+1)$, for $i = 1, ..., N$. The nominal process models are considered in this stability study; thus, $x_i(k+1) = \hat{x}^*_i(k+1|k)$, for $i = 1, ..., N$. At time '$k+1$', we get:

$$\begin{aligned} &\mathcal{J}^*_{CDMPC}(k+1) \leq \mathcal{J}^*_{CDMPC}(k) - \frac{1}{2} \sum_{i=1}^{N} \big(||x_i(k+1)||^2_{Q_i} + ||\Delta u_i(k)||^2_{R_i} \big), \\ &\mathcal{J}^*_{CDMPC}(k+1) < \mathcal{J}^*_{CDMPC}(k). \end{aligned} \quad (3.56)$$

It can be observed from (3.55) that $\mathcal{J}^*_{CDMPC}(k) \geq 0$, and $\mathcal{J}^*_{CDMPC}(k) = 0$ only if $(\hat{x}^*_i, \Delta u^*_i) = (0, 0)$ for $i = 1, ..., N$. Moreover, from equation (3.56), we get:

$\mathcal{J}^*_{CDMPC}(k+1) - \mathcal{J}^*_{CDMPC}(k) < 0$ for $(\hat{x}^*_i, \Delta u^*_i) \neq (0,0)$ for $i = 1, ..., N$. Thus, $\mathcal{J}^*_{CDMPC}(k)$ is a Lyapunov function for the entire closed-loop system, which shows in turn that the nominal closed-loop system under state-feedback CDMPC control is asymptotically stable.

□

3.6.1.2 Finite horizons

Here, it is shown that the CDMPC formulation using an infinite prediction horizon in Section 3.6.1 can be relaxed for subsystems that are coupled only through the control inputs. This means that the interacting variables $\hat{v}_i(k+l|k)$ in equation (3.11) are defined as: $\hat{v}_i(k+l|k) = \sum_{j \neq i} \left(B_{ij} \sum_{\nu=0}^{l} \Delta \hat{u}^s_j(k+\nu|k) \right)$, for $l = 0, .., H_p - 1$, with $\Delta \hat{u}^s_j(k+l|k) = 0$ for $H_u \leq l \leq H_p - 1$.

Theorem 3.6.2 *Consider that the entire process system* (3.1) *is controlled by* N *coordinated-distributed MPC controllers* (3.17) *that are coordinated using the prediction-driven coordination algorithm* (3.18)-(3.19). *Let us assume that the following conditions are satisfied:*

C1 : matrix $A = diag(A_{11}, A_{22}, \ldots, A_{NN})$, *where* A_{ii} *is stable for* $i = 1, \ldots, N$;

C2 : weighting matrices $Q_i(.) = Q_i$ *and* $R_i(.) = R_i$ *are positive definite and constant throughout the prediction and control horizons, for* $i = 1, \ldots N$;

C3 : the control interval is sufficiently long to ensure that the prediction-coordination algorithm converges.

Then, the entire closed-loop system under CDMPC control is asymptotically stable.

Proof: If conditions C1, C2 and C3 in Theorem 3.6.2 are satisfied then, within each control interval, the optimal solutions obtained with the coordinated-distributed control scheme converge to the centralized optimal solution. Furthermore, equation

(3.55), can be re-written as:

$$\begin{aligned}\mathcal{J}^*_{CDMPC}(k) =& \frac{1}{2}\sum_{i=1}^{N}\Big(\sum_{l=H_u}^{\infty}||\hat{x}^*_i(k+l|k)||^2_{Q_i} + \sum_{l=1}^{H_u-1}||\hat{x}^*_i(k+l|k)||^2_{Q_i}\Big)+ \\ & \frac{1}{2}\sum_{i=1}^{N}\Big(\sum_{l=1}^{H_u}||\Delta\hat{u}^*_i(k+l-1|k)||^2_{R_i}\Big).\end{aligned} \tag{3.57}$$

For the regulatory control problem, $\hat{u}_i(k+l-1|k) = 0$ for $l \geq H_u$. As a result, equation (3.57) can be simplified as:

$$\begin{aligned}\mathcal{J}^*_{CDMPC}(k) =& \frac{1}{2}\sum_{i=1}^{N}\Big(||\hat{x}^*_i(k+H_u|k)||^2_{\overline{Q}_i} + \sum_{l=1}^{H_u-1}||\hat{x}^*_i(k+l|k)||^2_{Q_i}\Big)+ \\ & \frac{1}{2}\sum_{i=1}^{N}\Big(\sum_{l=1}^{H_u}||\Delta\hat{u}^*_i(k+l-1|k)||^2_{R_i}\Big),\end{aligned} \tag{3.58}$$

where $\overline{Q}_i$ is the terminal penalty for each subsystem and it is the solution to the Lyapunov equation $A^T_{ii}\overline{Q}_i A_{ii} - \overline{Q}_i = -Q_i$ for $i = 1, \ldots, N$. The rest of the proof follows as in Section 3.6.1.1 with $\mathcal{J}^*_{CDMPC}(k)$ given by equation (3.58).

□

Remark 3.6.1 *If some (or all) of the local subsystems in condition C1, Theorem 3.6.2 are open-loop unstable, the closed-loop stability under coordinated-distributed control can also be shown using a finite prediction horizon as in Section 3.6.1.2. Nevertheless, the following additional conditions need to be taken into account when formulating the local optimization problems for the unstable subsystems:*

- *a terminal equality constraint for the unstable modes needs to be included in the optimal control problem of the unstable subsystems. The reason for including these equality constraints is to force the unstable modes to be zero at the end of the control horizon;*

- *terminal penalty $\overline{Q}_i$ for each unstable subsystem needs to be calculated based on the Lyapunov equation, which includes only the stable modes in the local unstable subsystem.*

The terminal equality constraint for the unstable modes and the terminal penalty can be calculated as follows:

- *First, the matrix A_{ii} in the unstable subsystems needs to be written in terms of the stable and unstable modes by using, for example, an eigenvalue-eigenvector decomposition (Maciejowski, 2002): $A_{ii} = E_i P_i E_i^{-1}$, where: $E_i = [E_{i\_u}, E_{i\_s}]$; $P_i = diag(P_{i\_u}, P_{i\_s})$; and $E_i^{-1} = \tilde{E}_i$ with:*

$$\tilde{E}_i = \begin{bmatrix} \tilde{E}_{i\_u} \\ \tilde{E}_{i\_s} \end{bmatrix};$$

- *Then, the unstable modes (ξ_{i_u}) and stable modes (ξ_{i_s}) for the local subsystems can be obtained from the transformation: $\xi_i = E_i^{-1} x_i$, where $\xi_i = [\xi_{i\_u}^T, \xi_{i\_s}^T]^T$. The terminal equality constraints on the unstable modes are given by: $\hat{\xi}_{i\_u}(k + H_u|k) = \tilde{E}_{i\_u} \cdot \hat{x}_i(k + H_u|k) = 0$.*

- *Finally, the terminal penalty $\overline{Q}_i$ is calculated as: $\overline{Q}_i = \tilde{E}_{i\_s}^T \Sigma_i \tilde{E}_{i\_s}$, where Σ_i is the solution of the Lyapunov equation: $\Sigma_i - P_{i\_s}^T \Sigma_i P_{i\_s} = E_{i\_s}^T Q_i E_{i\_s}$.*

3.6.2 Stability upon Premature Termination of the Coordination Algorithm

Premature termination of the coordination algorithm occurs when the coordination algorithm is stopped at iteration step 's' before convergence is achieved within each control interval. Premature termination of the coordination algorithm could be the result, for example, of poorly chosen termination criteria or selection of an insufficiently long control interval. Since the coordination algorithm is prematurely terminated, the number of iteration steps will probably differ between the different control intervals. In this section, the superscript 's_k' is used to denote the iteration step associated with time 'k'.

Theorem 3.6.3 *Consider that the entire process system* (3.1) *is controlled by N coordinated-distributed MPC controllers* (3.17) *that are coordinated using the prediction-driven coordination algorithm* (3.18)-(3.19). *Let us assume that the following conditions are satisfied:*

C1 : depending on the form of the process interactions, the weighting matrices $Q_i(.)$ and $R_i(.)$, as well as the prediction horizon H_p are selected as described in Sections 3.6.1.1 and 3.6.1.2;

C2 : the prediction-driven coordination algorithm is stopped before it converges and the solutions obtained with the CDMPC controllers are confined to a set which belongs to a neighborhood of the centralized optimal solution (δ_k);

C3 : the bound δ_k does not increase as the time 'k' increases.

Then, the entire closed-loop system under CDMPC control is asymptotically stable.

Proof: To show stability of the entire closed-loop system under coordinated-distributed MPC control, we propose $\mathcal{J}_{cen}(\Delta U^{s_k}(k))$ defined in (3.39) as a Lyapunov function candidate, and we rely on the convergence properties of the prediction-driven coordination algorithm. It is shown in Section 3.5.2 that the objective function $\mathcal{J}_{cen}(\Delta U^{s}(k))$ decreases as the number of iterations increase. Moreover, from equations (3.51) and (3.52), we get:

$$\begin{aligned}
\mathcal{J}_{cen}(\Delta U^{s_k}(k)) - \mathcal{J}_{cen}(\Delta U^{s_k+1}(k)) &= (\Delta U^{s_k}(k) - \Delta U^{s_k+1}(k)))^T W (\Delta U^{s_k}(k) - \Delta U^{s_k+1}(k)), \\
&= \|(\Delta U^{s_k}(k) - \Delta U^{s_k+1}(k))\|_W^2, \\
\mathcal{J}_{cen}(\Delta U^{s_k+1}(k)) - \mathcal{J}_{cen}(\Delta U^{s_k+2}(k)) &= \|(\Delta U^{s_k+1}(k) - \Delta U^{s_k+2}(k))\|_W^2, \\
&\cdots \\
\mathcal{J}_{cen}(\Delta U^{sf_k-1}(k)) - \mathcal{J}_{cen}(\Delta U^{sf_k}(k)) &= \|(\Delta U^{sf_k-1}(k) - \Delta U^{sf_k}(k))\|_W^2.
\end{aligned} \tag{3.59}$$

By adding the objective functions in (3.59) and taking iteration step $sf_k \to \infty$, we can calculate $\mathcal{J}_{cen}(\Delta U^{s}(k)) - \mathcal{J}_{cen}(\Delta U^{*}(k))$ at time 'k' as:

$$\mathcal{J}_{cen}(\Delta U^{s_k}(k)) - \mathcal{J}_{cen}(\Delta U^{*}(k)) = \textstyle\sum_{m=s_k}^{\infty} \|(\Delta U^{m}(k) - \Delta U^{m+1}(k))\|_W^2. \tag{3.60}$$

Let $\delta_k = \sum_{m=s_k}^{\infty} \|(\Delta U^{m}(k) - \Delta U^{m+1}(k))\|_W^2$, which denotes the difference between $\mathcal{J}_{cen}(\Delta U^{s_k}(k))$ and the optimal centralized objective function at time 'k'. Then, equation (3.60) becomes:

$$\mathcal{J}_{cen}(\Delta U^{s_k}(k)) - \mathcal{J}_{cen}(\Delta U^{*}(k)) = \delta_k. \tag{3.61}$$

Similarly, at time '$k+1$', we get:

$$\mathcal{J}_{cen}(\Delta U^{s_{k+1}}(k+1)) - \mathcal{J}_{cen}(\Delta U^{*}(k+1)) = \delta_{k+1}. \tag{3.62}$$

From equation (3.61) and (3.62), we obtain:

$$\mathcal{J}_{cen}(\Delta U^{s_{k+1}}(k+1)) - \mathcal{J}_{cen}(\Delta U^{s_k}(k)) = \mathcal{J}_{cen}(\Delta U^{*}(k+1)) - \mathcal{J}_{cen}(\Delta U^{*}(k)) + \delta_{k+1} - \delta_k. \tag{3.63}$$

If condition C1 in Theorem 3.6.3 is satisfied, then it can be shown that:

$$\mathcal{J}_{cen}(\Delta U^{*}(k+1)) - \mathcal{J}_{cen}(\Delta U^{*}(k)) < 0. \tag{3.64}$$

Moreover, from condition C3 in Theorem 3.6.3 there exists an iteration step 's_{k+1}' at time '$k+1$', such that:

$$\delta_{k+1} - \delta_k \leq 0. \tag{3.65}$$

Then, it follows from equations (3.63)-(3.65) that:

$$\mathcal{J}_{cen}(\Delta U^{s_{k+1}}(k+1)) - \mathcal{J}_{cen}(\Delta U^{s_k}(k)) < 0. \tag{3.66}$$

It can be observed from equations (3.39) and (3.66) that $\mathcal{J}_{cen}(\Delta U^{s_k}(k))$ is a Lyapunov function for the entire closed-loop system, which shows in turn that the nominal closed-loop system under state-feedback CDMPC control is asymptotically stable. Therefore, if the prediction-driven coordination algorithm is stopped before convergence is achieved, asymptotic stability of the closed-loop system is guaranteed (provided that the optimal solutions obtained with the CDMPC controllers are bounded by a set which belongs to a neighborhood of the centralized optimal solution and this bound does not increase as the time 'k' increases).

□

3.7 Illustrative Example

In this section, a numerical example is presented to show the effectiveness of the proposed coordinated-distributed control scheme. The dynamics of the entire process

are given by equation (3.1), where the dimensions of the state vector x and input vector u are $n = q = 15$. The matrix A is given by:

$$A = \tag{3.67}$$

$$\begin{bmatrix}
0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 \\
0 & 0.4 & 0 & 0 & 0 & 0 & 1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.1 & 0 & 0 & 0 & 0 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.35 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
-0.1 & 0 & 0 & 0 & 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.01 & 0 & 0 & 0 & 0 & 0 & 0.35 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.45 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2 & -0.5 & 0 & 0 & 0 \\
0 & -1.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.32 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.8 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.5 & 0 & 0 & 0 & 0 & 0 & 0 & -0.65
\end{bmatrix},$$

and the B matrix is given by:

$$B = \tag{3.68}$$

$$\begin{bmatrix}
0.25 & 0 & 0 & 0 & 0 & 0.0625 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.0875 & 0 & 0 & 0 & 0 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.025 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.55 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 \\
0.0375 & 0 & 0 & 0 & 0 & -0.025 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0025 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & -0.075 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.175 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0875 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.075 & 0.1875 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.025 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.0625 & 0 & 0 & 0 & 0 & 0 & 0 & -0.125 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0875 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & -0.075
\end{bmatrix}.$$

In this example, the process is considered to consist of three interacting subsystems. The first subsystem includes the first 5 state variables and the first 5 input variables from the process model; the second subsystem includes the next 5 state variables and the next 5 input variables from the process model; and finally, the third subsystem includes the last 5 state variables and last 5 input variables from the process model. To be able to identify the individual states and inputs in each subsystem, a subscript is appended to each element of the state and input vectors, respectively. For example for subsystem 1, the 5 components of the state vector x_1 are defined as: $x_1 = [x_{1\_1}, x_{1\_2}, x_{1\_3}, x_{1\_4}, x_{1\_5}]^T$ and the 5 components of the input vector u_1 are defined as: $u_1 = [u_{1\_1}, u_{1\_2}, u_{1\_3}, u_{1\_4}, u_{1\_5}]^T$. Similarly for subsystems 2 and 3.

The following parameters are used in the simulation study: weighting matrices $Q_1(l+1) = diag(5,1,1,1,1)$ for $l = 0, ..., H_p - 1$, $Q_i(l+1) = I_5$ for $i = 2, 3$ and $l = 0, ..., H_p - 1$; and $R_i(\eta) = 20 \cdot I_5$, for $i = 1, 2, 3$ and $\eta = 0, ..., H_u - 1$. The prediction

horizon H_p was selected as 20 and the control horizon H_u was selected as 5. To show the performance of the CDMPC controllers, a set-point change of magnitude 1.0 in state variable x_{1_1} was simulated at time $t = 0$, while the targets for the remaining state variables were kept at the origin[7]. The simulations were performed in Matlab 7.6, in an Intel Core Duo machine with 2 GHz of processor speed and 2 GB of RAM memory. The results of the computer simulation are presented in Figure 3.4 and Figure 3.5.

A comparison between the performance obtained with the CDMPC controllers and the performance obtained with a centralized controller was carried out for this simulation example. For ease of presentation, only the first two state trajectories and the first two input trajectories in each subsystem are shown in Figure 3.4. It can be observed in Figure 3.4 that the trajectories obtained with CDMPC controllers are equal to the trajectories obtained with a centralized controller. Therefore, the CDMPC controllers achieve the performance of a centralized controller. For the states and inputs not shown in Figure 3.4, the same result is observed. That is, the trajectories obtained with the CDMPC controllers exactly match the trajectories obtained with a centralized controller.

Convergence of the solutions obtained with the CDMPC controllers to the centralized solutions can also be observed in Figure 3.5. Figure 3.5 shows the percentage error in the overall objective function[8] for the first control interval (without loss of generality). It can be observed from Figure 3.5 that the error in the overall objective function decreases monotonically. Moreover, the error in the objective function is kept below a specified tolerance error (10^{-3}) after 7 iterations of the coordination algorithm. This indicates that, at convergence, the performance of the subsystems resulted in the centralized optimal performance.

The computational time was calculated for the CDMPC controllers and the centralized controller for the first control interval. The computational load for the CDMPC control scheme involved the time required for the coordinator's calculations

[7]The same simulation study was performed using a centralized controller and decentralized controllers.

[8]The percentage error in overall objective function is calculate as: $\big(|\mathcal{J}_{cen}(\Delta U^s(k)) - \mathcal{J}_{cen}(\Delta U^*(k))|/\mathcal{J}_{cen}(\Delta U^*(k)) \cdot 100\%\big)$.

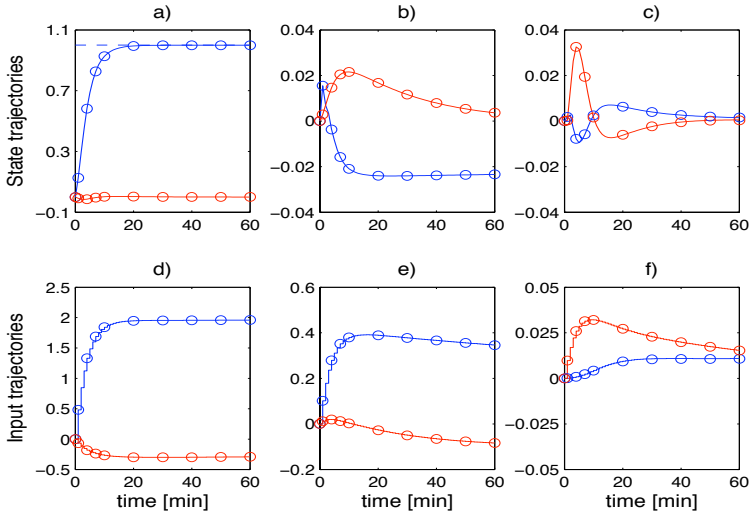

Figure 3.4: Comparison between CDMPC trajectories (solid line) and centralized trajectories (circles). a) Set-point for x_{1_1} (dashed line), states x_{1_1} (blue) and x_{1_2} (red); b) states x_{2_1} (blue) and x_{2_2} (red); c) states x_{3_1} (blue) and x_{3_2} (red); d) inputs u_{1_1} (blue) and u_{1_2} (red); e) inputs u_{2_1} (blue) and u_{2_2} (red); f) inputs u_{3_1} (blue) and u_{3_2} (red)

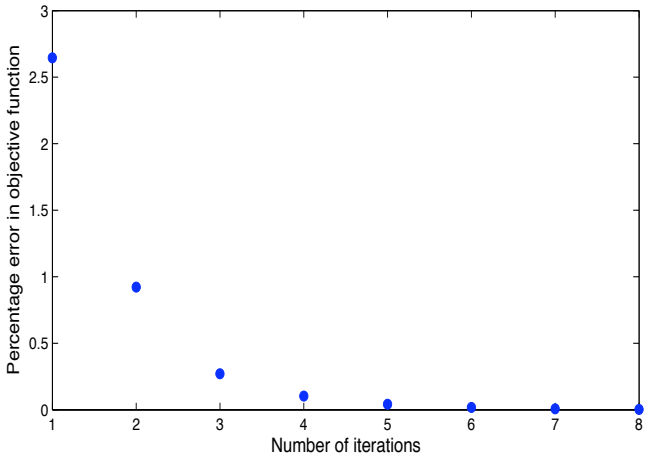

Figure 3.5: Percentage of error in objective function for the first control interval

and the time required to solve the optimization problem for the three subsystems (including all the iterations until the coordination algorithm converged within the first control interval). The computation load resulted in 0.91[s] for the centralized controller, and 1.53 [s] for the CDMPC controllers and the coordinator's calculations. It can be observed that the computation load is comparable for both control schemes. Nevertheless, it should be noted that the optimization problem for the three subsystems were solved in series with only one processor. A faster computational time for the CDMPC control problems is expected if they are solved in parallel computing.

Next, a comparison between the performance obtained with the CDMPC controllers and the performance obtained with decentralized controllers was carried out for this simulation example. The decentralized control formulation for each subsystem is given by equation (3.9). The improvement of the CDMPC controllers with respect to the decentralized controllers can be quantified using the following performance index:

$$\Pi = \frac{|\mathcal{J}_{dec} - \mathcal{J}_{CDMPC}|}{\mathcal{J}_{CDMPC}} \times 100\%, \tag{3.69}$$

where

$$\mathcal{J}_{(.)} = \frac{1}{2} \sum_{i=1}^{N} \sum_{k=0}^{t_f-1} \Big(||x_{i\_sp} - x_i(k+1)||^2_{Q_i} + ||\Delta u_i(k)||^2_{R_i} \Big),$$

and t_f is the simulation time. For the simulation studies performed on this illustrative example, the performance index Π indicates that the CDMPC controllers outperform the fully decentralized controllers by approximately 6.82% (based on a 60 $[min]$ closed-loop simulation).

Finally, different decompositions were simulated and the performance index (3.69) was calculated. The decomposition strategies simulated involved partitioning the entire process (3.67)-(3.68) into different subsystems and performing a set-point change of magnitude 1.0 in state variable x_{1_1} at time $t = 0$. The simulation results are shown in Table 3.1. It can be observed in Table 3.1 that the optimal solutions obtained with the CDMPC controllers are equal to the centralized solution regardless of the decomposition strategies selected. As discussed in Section 3.5.1, this shows the ability of the coordination algorithm in reaching the centralized optimal solution. Moreover, it can be observed in Table 3.1 that different decomposition strategies

result in different performance indexes with respect to the decentralized controllers. Due to the effect of the interacting variables, distinct decomposition strategies can lead to different levels of decentralized performance, which leads to different degrees of improvement between the decentralized MPC controllers and the CDMPC controllers.

Table 3.1: Performance index for different decomposition strategies

Decomp. strategy	Subsystems	Decomp. of state vector x	Decomp. of input vector u	$\frac{\lvert\mathcal{J}_{cen}-\mathcal{J}_{CDMPC}\rvert}{\mathcal{J}_{CDMPC}}\cdot 100\%$	$\frac{\lvert\mathcal{J}_{dec}-\mathcal{J}_{CDMPC}\rvert}{\mathcal{J}_{CDMPC}}\cdot 100\%$
1	1	first 2 states	first 2 inputs	0	6.79
	2	last 13 states	last 13 inputs		
2	1	first 6 states	first 6 inputs	0	3.98
	2	next 3 states	next 3 inputs		
	3	last 6 states	last 6 inputs		
3	1	first 3 states	first 3 inputs	0	6.82
	2	next 3 states	next 3 inputs		
	3	next 4 states	next 4 inputs		
	4	last 5 states	last 5 inputs		
4	1	first 9 states	first 9 inputs	0	1.82
	2	next 2 states	next 2 inputs		
	3	next 2 states	next 2 inputs		
	4	next state	next input		
	5	last state	last input		

3.8 Summary

In this chapter, a state-feedback CDMPC control scheme is proposed for discrete-time, linear unconstrained systems. It is shown that the CDMPC controllers are built by modifying existing decentralized MPC controllers. This allows the CDMPC controllers to maintain some of the desired properties of the decentralized MPC controllers such as autonomy and resiliency against equipment failure or shut-downs. In addition, if the communication between the coordinator and the CDMPC controllers is stopped, the CDMPC controllers transition to the original MPC decentralized controllers.

Convergence properties of the prediction-driven coordination algorithm are discussed in this chapter. It is shown that when the convergence condition is satisfied, the proposed prediction-driven algorithm improves monotonically with the number of iterations. That is, each iteration in the coordination algorithm produces a better solution than the previous iteration step. Furthermore, it is shown that when the coordination algorithm converges, the CDMPC solutions are equal to the plant-wide optimal solution.

Stability of the closed loop system under CDMPC control is also analyzed in this chapter. Two distinct cases are considered in the stability analysis. The first stability case considers CDMPC controllers with sufficiently long control interval which allows the coordination algorithm to iterate until convergence. The second case shows the stability of the closed-loop system when the coordination algorithm is prematurely stopped. In both cases, stability of the closed-loop system under CDMPC control is guaranteed provided that controller tuning conditions and coordination performance conditions are satisfied.

4

Coordinated-Distributed MPC via Price-Driven Coordination

This chapter presents a coordinated-distributed model predictive control (CDMPC) scheme for discrete-time, constrained linear dynamic systems. A *price-driven* method is used to coordinate the local CDMPC controllers in order to achieve the optimal plant-wide operations. In the price-driven method, the price vector is updated using Newton's method along with a sensitivity analysis technique. The flexibility of the CDMPC scheme is shown in the sense that CDMPC controllers can be implemented using state-space models as well as step-response models. An analysis of the CDMPC scheme's performance properties is also provided. Finally, the performance of the CDMPC controllers is shown using two benchmark process systems: an evaporator process and a fluid catalytic cracking process[1].

[1]Parts of this chapter are published in (Marcos *et al.*, 2008) and in (Marcos *et al.*, 2009).

4.1 Background

In Chapter 3, a CDMPC control scheme was developed for large-scale unconstrained systems, in which local CDMPC controllers are coordinated using the prediction-driven method. In this chapter, a CDMPC control scheme is formulated to include constraints on the state (or output) variables and the input (manipulated) variables. In addition, a new coordination method is employed. The price-driven coordination method is used in this chapter as an attempt to develop a price-update mechanism that can speed up the coordination process. As opposed to the prediction-driven method (see Chapters 2 and 3), the price-driven method does not require the coordinator to predict the state variables in the entire system, resulting in a faster price-update algorithm.

In the price-driven method considered in this chapter, a price vector 'p' is computed by the coordinator and updated based on Newton's method. A coordination method using a price was proposed in (Jose and Ungar, 1998; Jose and Ungar, 2000) to solve algebraic optimization problems such as resource allocation or auction problems. In (Jose and Ungar, 2000), a general large-scale optimization problem of the form:

$$\begin{aligned} \max_{z_1,\ldots,z_N} \quad & \sum_{i}^{N} \Phi_i(z_i) \\ \text{subject to:} \quad & \sum_{i}^{N} R_i(z_i) \leq \bar{R}, \qquad z_i \in \Omega_i \end{aligned} \tag{4.1}$$

is considered, where $z_i \in \Re^{n_i}$ is the vector of decision variables, Φ_i is the objective function for subproblem i, R_i is the vector of resource demands for subsystem i (for $i = 1, ..., N$), and the vector $\bar{R}$ represents the availability of shared resources. The large-scale optimization problem (4.1) can be decomposed into N subproblems:

$$\max_{z_i \in \Omega_i} \Phi_i(z_i) - (p + qR_i(z_i))^\top R_i(z_i). \tag{4.2}$$

In subproblem (4.2), 'p' represents the price vector, and the variable q is a nonnegative scalar that can be assumed to be zero for quadratic programming problems. It is shown in (Jose and Ungar, 2000) that, when the subproblems (4.2) are based on concave objective functions and compact convex feasible sets, they can be coordinated

successfully. Moreover, at equilibrium, the following condition is satisfied:

$$\Delta R(p,q) = \sum_i R_i(p,q) - \bar{R} \leq 0,$$

with $p^T(\Delta R(p,q)) = 0$. The price-driven method proposed in (Jose and Ungar, 1998; Jose and Ungar, 2000) was adapted and implemented in (Cheng *et al.*, 2007) to solve MPC steady-state target calculation problems. In this chapter, the price-driven method described in (Cheng *et al.*, 2007) is used to solve MPC dynamic optimization problems.

The main contributions of this chapter can be summarized as follows:

- First, a state-feedback CDMPC control scheme is formulated for constrained process systems whose dynamics are modeled in a state-space representation. The price-driven algorithm considered in this chapter uses Newton's method as a price update mechanism and it has been adapted from the price-driven algorithm presented in (Cheng *et al.*, 2007) for coordination of algebraic optimization problems.

- Then, the performance properties of the CDMPC scheme are evaluated. The performance of the price-driven algorithm is studied, followed by a discussion of the stability of the closed-loop system under CDMPC control.

- Finally, the price-driven coordination algorithm developed using state-space models is extended so that it can be used for processes whose dynamics are represented by finite step-response models. This shows that the proposed price-driven coordination algorithm is flexible, since it can be implemented using state-space models as well as input-output models obtained from step-test data.

4.2 Problem Description

The model used to describe the dynamics of the entire process and the models used by the decentralized MPC controllers have been previously presented in Chapter 3. These models are described again in Sections 4.2 and 4.3 for ease of presentation of the CDMPC problems solved in this chapter.

It is assumed that the large-scale system can be modeled by a linear discrete-time state-space representation:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ x(0) &= x^{init}, \end{aligned} \tag{4.3}$$

where $x(k) \in \Re^n$ and $u(k) \in \Re^q$ denote the vector of state and input variables at time step 'k', respectively.

As described in Chapter 3, a practical approach consists of using decentralized controllers to control each subsystem in the plant. The dynamics of each subsystem i are assumed to be represented by the following state-space model:

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{j \neq i} \big(A_{ij}x_j(k) + B_{ij}u_j(k)\big), \\ x_i(0) &= x_i^{init}, \end{aligned} \tag{4.4}$$

where $x_i(k) \in \Re^{n_i}$ and $u_i(k) \in \Re^{q_i}$. The matrices A_{ii} and B_{ii} in (4.4) represent the local subsystem dynamics, for $i = 1, \ldots, N$. The term '$\sum_{j \neq i} \big(A_{ij}x_j(k) + B_{ij}u_j(k)\big)$' in (4.4) represents the effect of the interactions. It can be noted that the total number of state variables in the plant is $n = \sum_{i=1}^{N} n_i$ and the total number of input variables in the plant is $q = \sum_{i=1}^{N} q_i$.

The CDMPC controllers can be built on existing decentralized MPC controllers. A decentralized MPC control formulation for subsystem (4.4) is presented in Section 4.3, followed by the development of the CDMPC controllers in Section 4.4.

4.3 Decentralized MPC Control

In the decentralized MPC, the variables predicted by the local MPC controller i along the prediction horizon H_p and control horizon H_u are given by:

$$\begin{cases} \hat{x}_i(k+l+1|k) = A_{ii}\hat{x}_i(k+l|k) + B_{ii}\hat{u}_i(k+l|k) + \\ \qquad\qquad \sum_{j \neq i} \big(\beta_i A_{ij}x_j(k) + B_{ij}u_j(k-1)\big), \quad \text{for } l = 0, ..., H_p - 1, \\ \hat{x}_i(k|k) = x_i(k), \\ \beta_i = 1, \quad \text{for } l = 0, \\ \beta_i = 0, \quad \text{for } l = 1, ..., H_p - 1, \end{cases} \tag{4.5}$$

with:

$$\begin{cases} \hat{u}_i(k+l|k) = \sum_{\nu=0}^{l} \Delta\hat{u}_i(k+\nu|k) + u_i(k-1), & \text{for } l = 0, ..., H_p - 1, \\ \Delta\hat{u}_i(k+l|k) = 0, \qquad H_u \leq l \leq H_p - 1. \end{cases} \tag{4.6}$$

All the assumptions described in Chapter 3 with respect to the decentralized MPC controllers (see Section 3.3) are applicable to the decentralized MPC controllers considered in this chapter.

Typically, process industries need to deal with constraints on the process variables, input variables and input moves. The limitations on the subsystems' variables are expressed as inequality constraints in each decentralized control problem, as follows:

$$\begin{cases} x_{i\_min} \leq \hat{x}_i(k+l+1|k) \leq x_{i\_max}, \\ u_{i\_min} \leq \hat{u}_i(k+l|k) \leq u_{i\_max}, \\ \Delta u_{i\_min} \leq \Delta\hat{u}_i(k+\eta|k) \leq \Delta u_{i\_max}, \\ \text{for} \quad l = 0, \ldots, H_p - 1, \quad \text{and} \quad \eta = 0, \ldots, H_u - 1. \end{cases} \tag{4.7}$$

Assuming that the vector of desired set-points (X_{i_sp}) is provided, then the decentralized MPC control problem for each subsystem i (i=1,...,N) can be formulated as:

$$\begin{aligned} &\min_{X_i, \Delta U_i} \ \Phi_i = \frac{1}{2}\Big(\big(X_{i\_sp} - X_i(k)\big)^T \mathcal{Q}_i \big(X_{i\_sp} - X_i(k)\big) + \Delta U_i(k)^T \mathcal{R}_i \Delta U_i(k)\Big) \\ &\text{subject to :} \\ &\qquad \begin{cases} \text{Equality constraints } (4.5) - (4.6) \text{ and} \\ \text{Inequality constraints } (4.7), \end{cases} \end{aligned} \tag{4.8}$$

where the vectors $X_i(k)$ and $\Delta U_i(k)$, for $i = 1, ..., N$, are defined as:

$$\begin{aligned} X_i(k) &= [\hat{x}_i(k+1|k)^T, \hat{x}_i(k+2|k)^T, ..., \hat{x}_i(k+H_p|k)^T]^T, \\ \Delta U_i(k) &= [\Delta\hat{u}_i(k|k)^T, \Delta\hat{u}_i(k+1|k)^T, ..., \Delta\hat{u}_i(k+H_u-1|k)^T]^T. \end{aligned} \tag{4.9}$$

The matrices $\mathcal{Q}_i$ and $\mathcal{R}_i$ in (4.8) are defined as $\mathcal{Q}_i = diag(Q_i(1), Q_i(2), \ldots, Q_i(H_p))$ and $\mathcal{R}_i = diag(R_i(0), R_i(1), \ldots, R_i(H_u - 1))$, where the weighting matrices $Q_i(.) \in \Re_+^{n_i \times n_i}$ and $R_i(.) \in \Re_+^{q_i \times q_i}$ are positive definite matrices with their off-diagonal elements equal to zero.

4.4 Coordinated-Distributed MPC Control

In this section, the CDMPC control scheme is formulated. In Section 4.4.1, the process models used in the decentralized MPC controllers (see Section 4.3) are modified to

incorporate the effect of the interactions between the subsystems. In Section 4.4.2, the decentralized objective function is augmented by an extra term, which includes a price and is required for coordination of the subsystems.

4.4.1 Process Models used in the CDMPC Controllers

To design the CDMPC controllers, equation (4.5) needs to be modified. Process model (4.5) is modified to include the interactions between subsystems as follows:

$$\begin{cases} \hat{x}_i(k+l+1|k) = A_{ii}\hat{x}_i(k+l|k) + B_{ii}\hat{u}_i(k+l|k) + \\ \qquad\qquad \sum_{j\neq i}\left(\beta_i A_{ij}x_j(k) + B_{ij}u_j(k-1)\right) + \hat{v}_i(k+l|k), \\ \qquad\qquad \text{for } l = 0, ..., H_p - 1, \\ \hat{x}_i(k|k) = x_i(k), \\ \beta_i = 1, \quad \text{for } l = 0, \\ \beta_i = 0, \quad \text{for } l = 1, ..., H_p - 1, \end{cases} \tag{4.10}$$

where $\hat{u}_i(k+l|k)$ is defined as in equation (4.6). In the proposed control scheme, $\hat{v}_i$ are decision variables in the local CDMPC control problem[2]. It is shown in Section 4.4.3 and Section 4.4.4, that the coordinator is formulated to ensure that the following equation is satisfied for each subsystem i, for $i = 1, ..., N$:

$$\begin{cases} \Delta e_i(k+l|k) = \\ \hat{v}_i(k+l|k) - \Big(\sum_{j\neq i}\beta_i A_{ij}\hat{x}_j(k+l|k) + \sum_{j\neq i} B_{ij}\sum_{\nu=0}^{l}\Delta\hat{u}_j(k+\nu|k)\Big) = 0, \\ \text{for } l = 0, ..., H_p - 1, \\ \beta_i = 0, \quad \text{for } l = 0, \\ \beta_i = 1, \quad \text{for } l = 1, ..., H_p - 1, \\ \Delta\hat{u}_j(k+l|k) = 0, \qquad H_u \leq l \leq H_p - 1. \end{cases} \tag{4.11}$$

[2]It can be noted that in this chapter, the variables $\hat{v}_i$ are considered as decision variables in the local CDMPC control problem. In Chapter 3, the variables $\hat{v}_i$ are not decision variables in the local CDMPC control problems and they are calculated based on the data predicted by the coordinator.

4.4.2 Formulation of CDMPC Controllers

The CDMPC controller for each subsystem i (with $i = 1, ..., N$) is formulated as follows:

$$\begin{aligned} &\min_{X_i, \Delta U_i, V_i} \quad \mathcal{J}_i = \Phi_i + p^{s^T} \Theta_i [X_i(k)^T, \Delta U_i(k)^T, V_i(k)^T]^T \\ &\text{subject to :} \\ &\qquad \begin{cases} \text{Equality constraints (4.6), (4.10) and} \\ \text{Inequality constraints (4.7),} \end{cases} \end{aligned} \tag{4.12}$$

where the objective function Φ_i is defined in (4.8).

The vector of predicted states $X_i(k)$ and predicted input moves $\Delta U_i(k)$ are defined in (4.9). The vector $V_i(k)$ is defined as:

$$V_i(k) \quad = \quad [\hat{v}_i(k|k)^T, \hat{v}_i(k+1|k)^T, ..., \hat{v}_i(k+H_p-1|k)^T]^T.$$

The objective function in optimal control problem (4.12) differs from the objective function in (4.8) by the term '$p^{s^T} \Theta_i [X_i(k)^T, \Delta U_i(k)^T, V_i(k)^T]$'. The price vector '$p^s$' is computed by a coordinator, based on the price update mechanism described in Section 4.4.3. The matrix Θ_i in (4.12) is a coefficient matrix that takes into account the effect of the interactions between the subsystems. The matrix Θ_i is defined in Appendix 4.9.1.

For simplicity, the optimal control problem of each subsystem (4.12) can be reformulated in a compact form as:

$$\min_{Z_i} \ \mathcal{J}_i = \frac{1}{2}\Big(Z_i(k)^T \Upsilon_i Z_i(k)\Big) + \Big(\phi_i^T + p^{s^T} \Theta_i\Big) Z_i(k) \tag{4.13}$$

$$\text{subject to :} \quad \begin{cases} G_i^{eq} Z_i(k) = g_i^{eq}, \\ G_i^{ineq} Z_i(k) \leq g_i^{ineq}, \end{cases} \tag{4.14}$$

where the vector of decision variables $Z_i(k)$ is given by: $Z_i(k) = [X_i(k)^T, \Delta U_i(k)^T, V_i(k)^T]^T$.

The system of relations (4.14) can be obtained by arranging equations (4.6), (4.10) and inequalities (4.7) in a matrix form for the entire prediction horizon H_p and control horizon H_u. In (4.14), G_i^{eq} is the coefficient matrix for the decision variables in the

equality constraints (4.6) and (4.10). The vector g_i^{eq} includes known data, such as the effect of current measured state variables and past input variables from the plant. The matrix G_i^{ineq} is the coefficient matrix in the inequality constraints (4.7), and the vector g_i^{ineq} represents the upper and lower limits for the decision variables in the local subsystem i.

4.4.3 Price-Driven Coordination Algorithm

In this section, an efficient coordination method is proposed to ensure convergence of the coordinated-distributed optimal solutions to the centralized (plant-wide) optimal solution. Coordination of subproblems (4.13)-(4.14) to achieve the centralized optimal solution can be performed by using a price-update technique, such as Newton's method. Based on Newton's method, the price vector can be updated as follows (Cheng *et al.*, 2007):

$$\mathbb{J} \cdot p^{s+1} = \mathbb{J} \cdot p^s - \alpha \ \Delta E^s(k), \tag{4.15}$$

provided that the matrix $\mathbb{J}$ is full rank (see equation (4.18)).

In price update mechanism (4.15), the superscripts 's' and '$s+1$' denote two consecutive iteration steps and α is the step size in Newton's method. The task of the coordinator is to iteratively adjust the price p^s according to equation (4.15) until $\Delta E^s(k)$ is driven to zero. The interaction error vector $\Delta E^s(k)$ concatenates the error variables Δe_i in (4.11) for all the subsystems. That is, the vector $\Delta E^s(k)$ is arranged as:

$$\Delta E^s(k) = [\Delta e_1(k+1|k)^T, ..., \Delta e_1(k+H_p-1|k)^T, ..., \Delta e_N(k+1|k)^T, ..., \Delta e_N(k+H_p-1|k)^T]^T,$$

where $\Delta e_i(k+l|k)$, for $l = 0, ..., H_p - 1$, is defined in equation (4.11). The interaction error vector $\Delta E^s(k)$ can be written in terms of the decision variables $Z_i^s(k)$ as follows:

$$\Delta E^s(k) = \sum_{i=1}^{N} \Theta_i Z_i^s(k), \tag{4.16}$$

where $Z_i^s(k)$ are the solutions calculated by each CDMPC controller, $Z_i(k)$, at iteration step 's'.

The Jacobian matrix $\mathbb{J}$ in price update (4.15) is defined as:

$$\mathbb{J} = \frac{d\Delta E^s(k)}{dp^s}. \tag{4.17}$$

Then, from equations (4.16) and (4.17), the Jacobian matrix $\mathbb{J}$ becomes:

$$\mathbb{J} = \sum_{i=1}^{N} \Theta_i \frac{dZ_i^s(k)}{dp^s}. \tag{4.18}$$

It can be observed that the sensitivity matrix $dZ_i^s(k)/dp^s$, for $i = 1, ..., N$, is required to compute the Jacobian matrix $\mathbb{J}$ in equation (4.18). That is, in order to efficiently adjust the price vector, the coordinator should be aware of how the price affects the decision variables $Z_i(k)$ at each iteration step 's'.

A sensitivity analysis was proposed in (Wolbert *et al.*, 1994) for an algebraic optimization of a process flowsheet, and was extended in (Cheng *et al.*, 2007) for the MPC steady-state target calculation. This approach can be used to calculate $dZ_i^s(k)/dp^s$ and solve problem (4.18). The sensitivity matrix $dZ_i^s(k)/dp^s$ can be computed for each subsystem as follows:

First, the Lagrangian $\mathcal{L}_i$ can be formulated for optimization problem (4.13)-(4.14) as:

$$\begin{aligned}\mathcal{L}_i\Big(Z_i(k), \lambda_i, \mu_i, p^s\Big) =& \frac{1}{2}\Big(Z_i(k)^T \Upsilon_i Z_i(k)\Big) + \phi_i^T Z_i(k) + p^{s^T} \Theta_i Z_i(k) + \\ & \lambda_i^T \Big(G_i^{eq} Z_i(k) - g_i^{eq}\Big) + \mu_i^T \Big(G_i^{ineq} Z_i(k) - g_i^{ineq}\Big),\end{aligned}$$

where the vectors λ_i and μ_i denote the Lagrange multipliers for the equality and inequality constraints in (4.14), respectively. The first-order optimality conditions are:

$$\begin{aligned}&\nabla_{Z_i(k)} \mathcal{L}_i\Big(Z_i(k), \lambda_i, {}_{\mathcal{AC}}\mu_i, p^s\Big) = \\ &\qquad \Upsilon_i Z_i(k) + \phi_i + \Theta_i^T p^s + G_i^{eq^T} \lambda_i + {}_{\mathcal{AC}}G_i^{ineq^T} {}_{\mathcal{AC}}\mu_i = 0, \\ &F_i\Big(Z_i(k), p^s\Big) = G_i^{eq} Z_i(k) - g_i^{eq} = 0, \\ &{}_{\mathcal{AC}}f_i\Big(Z_i(k), p^s\Big) = {}_{\mathcal{AC}}G_i^{ineq} Z_i(k) - {}_{\mathcal{AC}}g_i^{ineq} = 0, \\ &{}_{\mathcal{IN}}f_i\Big(Z_i(k), p^s\Big) + {}_{\mathcal{IN}}\sigma_i = {}_{\mathcal{IN}}G_i^{ineq} Z_i(k) - {}_{\mathcal{IN}}g_i^{ineq} + {}_{\mathcal{IN}}\sigma_i = 0.\end{aligned} \tag{4.19}$$

In (4.19), the vector ${}_{\mathcal{IN}}\sigma_i$ represents the slack variables associated with the inactive inequality constraints. The subscript '$\mathcal{AC}$' indicates the inequality constraints that

are active, whereas the subscript '$\mathcal{IN}$' indicates the inequality constraints that are inactive. The optimality conditions (4.19) can be then differentiated as in (Cheng *et al.*, 2007), to obtain the following system of equations:

$$\begin{aligned}
&\nabla^2_{Z_i(k)Z_i(k)}\mathcal{L}_i dZ_i(k) + \nabla^2_{Z_i(k)p}\mathcal{L}_i dp^s + \nabla_{Z_i(k)}F_i^T d\lambda_i + \nabla_{Z_i(k)\,\mathcal{AC}} f_i^T d_{\mathcal{AC}}\mu_i = 0,\\
&\nabla_{Z_i(k)}F_i dZ_i(k) + \nabla_p F_i dp^s = 0,\\
&\nabla_{Z_i(k)\,\mathcal{AC}} f_i dZ_i(k) + \nabla_{p\,\mathcal{AC}} f_i dp^s = 0,\\
&\nabla_{Z_i(k)\,\mathcal{IN}} f_i dZ_i(k) + \nabla_{p\,\mathcal{IN}} f_i dp^s + d_{\mathcal{IN}}\sigma_i = 0.
\end{aligned}$$

Finally, the sensitivity matrix $dZ_i^s(k)/dp^s = \nabla_p Z_i(k)$ can be calculated by solving the following system of equations, provided that Γ_i is full rank:

$$\Gamma_i \begin{bmatrix} \nabla_p Z_i(k) \\ \nabla_p \lambda_i \\ \nabla_{p\,\mathcal{AC}}\mu_i \\ \nabla_{p\,\mathcal{IN}}\sigma_i \end{bmatrix} = - \begin{bmatrix} \Theta_i^T \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.20}$$

where

$$\Gamma_i = \begin{bmatrix} \Upsilon_i & G_i^{eq^T} & {}_{\mathcal{AC}}G_i^{ineq^T} & 0 \\ G_i^{eq} & 0 & 0 & 0 \\ {}_{\mathcal{AC}}G_i^{ineq} & 0 & 0 & 0 \\ {}_{\mathcal{IN}}G_i^{ineq} & 0 & 0 & I \end{bmatrix}, \tag{4.21}$$

and where $\nabla_p\lambda_i$, $\nabla_{p\,\mathcal{AC}}\mu_i$ and $\nabla_{p\,\mathcal{IN}}\sigma_i$ in (4.20) denote $d\lambda_i/dp^s$, $d_{\mathcal{AC}}\mu_i/dp^s$ and $d_{\mathcal{IN}}\sigma_i/dp^s$, respectively.

4.4.4 Implementation of the CDMPC Controllers

As described in Chapter 3, during a single control interval and before the first calculated control input move is implemented, the coordinator exchanges information with the local CDMPC controllers. During this data exchange, the coordinator iteratively adjusts the price vector to drive the calculated optimal solution of the subsystems to the centralized optimal solution.

The implementation of the CDMPC controllers are summarized in the following steps:

1. **Initialization (s=0):** The coordinator sets up an initial price vector p^s for the interacting variables ($\Theta_i Z_i(k),\ \forall i = 1, ..., N$). For initialization, the elements

of the vector p^s can be set to zero. The coordinator transfers the information to every local CDMPC controller.

2. **Optimization performed by each local CDMPC controller:** Based on the price provided by the coordinator, each CDMPC controller solves the local optimization problem (4.13)-(4.14) and calculates the sensitivity matrices according to equations (4.20)-(4.21). The calculated optimal solution $Z_i(k)$; the sensitivity information $\nabla_p Z_i(k)$, $\nabla_{p_{\mathcal{AC}}}\mu_i$ and $\nabla_{p_{\mathcal{IN}}}\sigma_i$; as well as the Lagrange multipliers ${}_{\mathcal{AC}}\mu_i$ and the slack variables ${}_{\mathcal{IN}}\sigma_i$ are communicated to the coordinator.

3. **Price update:** The iteration counter 's' is incremented. The coordinator gathers the information from each CDMPC controller. The coordinator calculates $\Delta E^s(k)$ using equation (4.16) and with $Z_i^s(k) = Z_i(k)$, for $i = 1, ...N$; and it calculates $\mathbb{J}$ using equation (4.18). Next, the coordinator determines the step size α, with $0 < \alpha \leq 1$, as the smallest positive value that causes an individual constraint change its activity (see Cheng *et al.* (2007) for a detailed description on how α is calculated based on the sensitivity information, the Lagrange multipliers ${}_{\mathcal{AC}}\mu_i$ and the slack variables ${}_{\mathcal{IN}}\sigma_i$ of the subsystems).
Then, the coordinator updates the price vector as per equation (4.15). The new price vector is informed to each local CDMPC controller.

4. **Iteration until convergence:** Steps (2)-(3) are repeated until the price-driven coordination algorithm is terminated. The price-driven coordination algorithm is terminated when $\|\Delta E^s(k)\| \leq \epsilon$, where ϵ is a specified error tolerance.

5. **Implementation of first calculated control action:** Once the price-driven coordination algorithm is terminated, the first calculated control input moves $\Delta\hat{u}_i(k|k)$ are implemented in each subsystem i, while the rest of the calculated control moves are discarded. The iteration counter 's' in the coordination algorithm is reset to zero and the coordinated optimization problems (steps (1)-(4)) are initiated again for the next receding horizon.

4.5 Analysis of the CDMPC Performance Properties

In this section, a discussion of the CDMPC performance properties is presented. An analysis of the price-driven coordination method is shown in Section 4.5.1, followed by a discussion of the stability of the closed-loop system under CDMPC control in Section 4.5.2.

4.5.1 Price-Driven Coordination Algorithm Analysis

The performance of the price-driven coordination algorithm is analyzed in this section. First, it is shown that when the coordination algorithm converges, the price vector is equivalent to the Lagrange multiplier vector for the equality constraints in the centralized control problem. Next, the behaviour of the coordination algorithm (as it evolves throughout the iterations) is shown with a small illustrative example.

Interpretation of the Price

Theorem 4.5.1 *Consider that the entire process system* (4.3) *is controlled by N coordinated-distributed MPC controllers* (4.13)-(4.14). *Suppose that the CDMPC controllers are coordinated using a price-driven coordination algorithm, where the price* p^s *is computed using price update mechanism* (4.15). *When* $\|\Delta E^s(k)\| \to 0$, *the price vector* p^s *is equivalent to the Lagrange multiplier vector* λ *for the equality constraints in the centralized control problem.*

Proof: For simplicity, and without loss of generality, the inequality constraints in the CDMPC control problems are not considered in this study. Theorem 4.5.1 can be proved by first defining the centralized MPC control problem for the entire process and calculating the first-order optimality conditions for the centralized MPC control problem. The first-order optimality conditions of the centralized MPC control problem are then compared to the first-order optimality conditions of the CDMPC control problems to show that at convergence $p^s = \lambda$.

Centralized MPC control problem:

The centralized MPC optimal control problem[3] can be written as:

$$\begin{aligned} &\min_{Z} \ \mathcal{J}_{cen} = \frac{1}{2}\Big(Z(k)^T \Upsilon Z(k)\Big) + \phi^T Z(k) \\ &\text{subject to :} \\ &\qquad G^{eq} Z(k) + \Delta E = g^{eq}, \end{aligned} \tag{4.22}$$

where the matrix Υ is defined as $\Upsilon = diag(\Upsilon_1, \Upsilon_2, ..., \Upsilon_N)$, the vector ϕ is defined as $\phi = [\phi_1^T, \phi_2^T, ..., \phi_N^T]^T$. The matrix G^{eq} in (4.22) is defined as: $G^{eq} = diag(G_{11}^{eq}, G_{22}^{eq}, ..., G_{NN}^{eq})$ and the vector g^{eq} is defined as: $g^{eq} = [g_1^{eq^T}, g_2^{eq^T}, ..., g_N^{eq^T}]^T$. The vector ΔE is defined in equation (4.16), but it can also be expressed as:

$$\Delta E = \bar{\Theta} Z(k), \tag{4.23}$$

where $\bar{\Theta} = [\Theta_1 \nmid \Theta_2 \nmid \ldots \nmid \Theta_N]$. From (4.22) and (4.23), the centralized MPC control problem can be written as:

$$\begin{aligned} &\min_{Z} \ \mathcal{J}_{cen} = \frac{1}{2}\Big(Z(k)^T \Upsilon Z(k)\Big) + \phi^T Z(k) \\ &\text{subject to :} \\ &\qquad G^{eq} Z(k) + \bar{\Theta} Z(k) = g^{eq}. \end{aligned} \tag{4.24}$$

The Lagrangian $\mathcal{L}_{cen}$ for the centralized control problem (4.24) can be formulated as:

$$\mathcal{L}_{cen} = \frac{1}{2}\Big(Z(k)^T \Upsilon Z(k)\Big) + \phi^T Z(k) + \lambda^T \big(G^{eq} Z(k) + \bar{\Theta} Z(k) - g^{eq}\big).$$

Then, the first-order optimality conditions can be written as:

$$\begin{aligned} &\Upsilon Z^*(k) + \phi + G^{eq^T} \lambda^* + \bar{\Theta}^T \lambda^* = 0, \\ &G^{eq} Z^*(k) + \bar{\Theta} Z^*(k) - g^{eq} = 0, \end{aligned} \tag{4.25}$$

where $Z^*(k)$ and λ^* represent the optimal values for $Z(k)$ and λ, respectively.

Coordinated-distributed MPC control problems:

Next, let us calculate the first-order optimality conditions for the subproblems[4]. The Lagrangian $\mathcal{L}_i$ for subproblem (4.13)-(4.14), for $i = 1, \ldots, N$, is:

$$\mathcal{L}_i = \frac{1}{2}\Big(Z_i(k)^T \Upsilon_i Z_i(k)\Big) + \Big(\phi_i^T + p^{s^T} \Theta_i\Big) Z_i(k) + \lambda_i^T \big(G_i^{eq} Z_i(k) - g_i^{eq}\big).$$

[3]The inequality constraints are not considered in this analysis.

[4]The inequality constraints are not considered in this analysis. Thus, in subproblem (4.13)-(4.14), it is assumed that $G_i^{ineq} = 0$ and $g_i^{ineq} = 0$.

Therefore, the first-order optimality conditions for subproblem i becomes:

$$\begin{aligned} &\Upsilon_i Z_i(k) + \phi_i + \Theta_i^T p^s + G_i^{eq^T} \lambda_i = 0, \\ &G_i^{eq} Z_i(k) - g_i^{eq} = 0. \end{aligned} \tag{4.26}$$

The first-order optimality conditions (4.26) can be written for all the subproblems as:

$$\begin{aligned} &\Upsilon Z_{_{CDMPC}}(k) + \phi + G^{eq^T} \lambda_{_{CDMPC}} + \bar{\Theta}^T p^s = 0, \\ &G^{eq} Z_{_{CDMPC}}(k) - g^{eq} = 0, \end{aligned} \tag{4.27}$$

where $Z_{_{CDMPC}}(k) = \left[Z_1(k)^T, Z_2(k)^T, \ldots, Z_N(k)^T\right]^T$ and $\lambda_{_{CDMPC}} = \left[\lambda_1^T, \lambda_2^T, \ldots, \lambda_N^T\right]^T$.

The price-driven coordination algorithm converges once $\|\Delta E^s(k)\| \to 0$, with $\Delta E = \bar{\Theta} Z_{_{CDMPC}}$, and the solutions obtained with the CDMPC controllers $Z_{_{CDMPC}}(k)$ are equal to the centralized optimal solution $Z^*(k)$. Then, it can be shown from equations (4.25) and (4.27) that $\lambda_{_{CDMPC}} = \lambda^*$ and $p^s = \lambda^*$, provided that the entire system has been modeled to ensure that the column-rank($[G^{eq^T} \ \bar{\Theta}^T]$) = dim($\lambda_{_{CDMPC}}$)+dim($p^s$). As a result, the price vector p^s is equal to the Lagrange multiplier vector λ^* for the equality constraints in the centralized control problem.

□

Coordination Algorithm Behaviour

As the coordination algorithm iterates in the price-driven coordination method, the coordinator is searching for the value of the price that will lead to the centralized optimal solution. To show the behaviour of the price in the CDMPC control scheme, let us consider a simple discrete-time system whose dynamics are represented by model (4.3), with:

$$A = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.3 \end{bmatrix}, B = \begin{bmatrix} 0.2 & 0.5 \\ 0 & 1 \end{bmatrix}, \tag{4.28}$$

and initial conditions $x(0) = (0, 0)^T$. It is assumed that process (4.28) consists of two subsystems. Subsystem 1 includes the first state and the first input; subsystem 2 includes the second state and the second input. The parameters used in this simulation example are: prediction horizon $H_p = 20$; control horizon $H_u = 1$ and weighting matrices $Q_1 = Q_2 = 10$ and $R_1 = R_2 = 2$; $|x_i| \leq 10$ and $|\Delta u_i| \leq 1$ for $i = 1, 2$. Here, we selected a simple system and a control horizon H_u equal to 1

for illustration purposes only, because under these conditions, the price reduces to a scalar value.

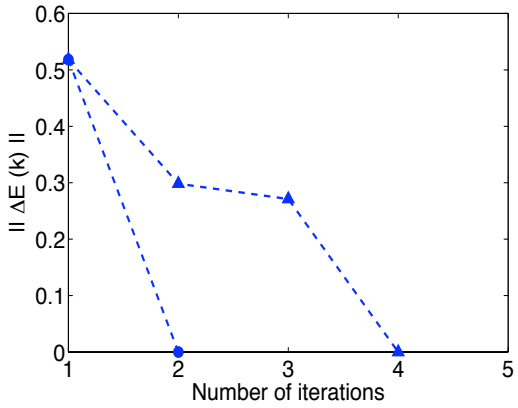

Figure 4.1: Price-driven algorithm convergence behaviour for system (4.28): unconstrained system (•); upper limit for $\Delta\hat{u}_1$ is active (▲).

Figure 4.1 shows the interaction error $||\Delta E^s(k)||$ as the coordination algorithm progresses within the first control interval. Overall, it can observed from Figure 4.1 that the proposed coordination method converges quickly. In particular, it can be observed in Figure 4.1 that the coordination algorithm converges at the second iteration for the unconstrained subsystems. When there is a change in the active constraint set of the subsystems, more iterations may be required to achieve convergence of the price-driven algorithm (within a control interval). This behaviour can be observed in Figure 4.1, where it is shown that the price-driven method converges within 4 iterations as the variable $\Delta\hat{u}_1$ became active. The fast convergence behaviour of the price-driven coordination method can also be observed for the illustrative examples in Sections 4.6 and 4.7.3.

4.5.2 Stability of the closed-loop system under CDMPC

The stability of the closed-loop system under CDMPC control can be proved as in Section 3.6.1 in Chapter 3. In particular, for the constrained CDMPC control scheme proposed in this chapter, the following conditions are assumed:

C1 : feasibility of the CDMPC optimization problems at the first iteration step,

$s = 0$, in the coordination algorithm (within each control interval). In general, feasibility in MPC implies that, for the current measured states, the constraints specified in the MPC optimization problem are satisfied. See comment below regarding infeasibility in MPC.

C2 : The control interval should be sufficiently long to ensure that the price-driven coordination algorithm converges.

If condition C1 is satisfied (within each control interval), then:

- feasibility of the CDMPC optimization problems at the first step in the coordination algorithm guarantees that the CDMPC objective functions can be minimized at each iteration step of the coordination algorithm. Thus, the CDMPC optimization problem is feasible within each control interval.

If conditions C1 and C2 are satisfied, then:

- within each control interval, the solutions obtained with the CDMPC controllers are equal to the optimal solution obtained by a centralized controller. Thus, the trajectories obtained with the CDMPC controllers are equal to the trajectories obtained with a centralized controller at all times. The stability of the closed-loop system under CDMPC control can be then proved as in Section 3.6.1.

Comment regarding infeasibility in MPC:

"Infeasibility" is defined as the inability to satisfy all the constraints simultaneously (Rossiter, 2005). Infeasibility is likely to arise in any constrained MPC application. Many situations can cause infeasibility in MPC. For example, a disturbance or a large set-point change can force process variables outside their limits, making the MPC algorithm unable to satisfy all the hard constraints. The first step in dealing with infeasibility in MPC is to identify its cause. By understanding the cause of infeasibility, one is in a position to propose strategies for dealing with it.

Many suggestions have been published in the literature to cope with infeasibility issues in MPC control (Camacho and Bordons, 1999). One suggestion is to drop (if possible) the constraints on the state variables during the initial portion of the

prediction horizon in order to make the optimization problem feasible. Another suggestion for dealing with infeasibility is to "soften" the hard constraints on the states variables. In this sense, the constrained control problems will be feasible because the state constraints are allowed to violate their limits to some extent (Camacho and Bordons, 1999; Maciejowski, 2002).

4.6 Illustrative Example I

To show that the effectiveness of the CDMPC controllers using state-space models, a simulation study was performed on the evaporator process system presented in (Newell and Lee, 1989). The continuous-time model used to represent the dynamics of the evaporator process is given in equations (2.44) in Chapter 2. To implement the CDMPC controllers, the process model (2.44) is discretized (as proposed by Newell and Lee (1989)) with a sampling period of 1 $[min]$, leading to discrete-time model (4.3) with:

$$
\begin{aligned}
& x = [L2,\ X2,\ P2]^T, \quad u = [F2,\ P100,\ F200]^T, \\
& A = \begin{bmatrix} 1 & 0.0975 & 0.3692 \\ 0 & 0.9048 & 0 \\ 0 & -0.0096 & 0.9467 \end{bmatrix}, \\
& B = \begin{bmatrix} -0.1050 & 0.3795 & -0.0014 \\ -0.0952 & 0 & 0 \\ 0.0005 & 0.0359 & -0.0073 \end{bmatrix}.
\end{aligned} \tag{4.29}
$$

It is assumed that the evaporator process (4.29) consists of two subsystems. The state and input variables considered in each subsystem are described in Table 2.3 (Chapter 2). The weighting matrices Q_i and R_i in Table 2.3 are also used in this illustrative example, with $Q_i = Q_i(l+1)$, for $l = 0, ..., H_p - 1$ and $R_i = R_i(\eta)$, for $\eta = 0, ..., H_u - 1$. The prediction horizon H_p was selected as 30, and the control horizon H_u was selected as 5. To initialize the coordination algorithm, all the elements in the price vector were set to zero.

To demonstrate the performance of the CDMPC controllers, a change of one unit in variable $L2$ was simulated at time $t = 0$, while the targets for variables $X2$ and $P2$ were kept at their nominal operating conditions. Figure 4.2 and Figure 4.4 show the results based on a 100 $[min]$ simulation with initial conditions $x_1(0) = [0, 0]^T$ and

$x_2(0) = 0$, and using model (4.29) for the true plant dynamics. The simulation results are shown in normalized deviation variables. Figure 4.2a) and Figure 4.2b) show the trajectory of the state variables ($L2$, $X2$ and $P2$) and the control inputs ($F2$, $P100$ and $F200$). It can be observed that the CDMPC controllers drive level $L2$ to the new target while keeping controlled variables $X2$ and $P2$ at their desired set-points.

The state and input trajectories shown in Figure 4.2a) and Figure 4.2b) match the trajectories obtained with a centralized controller. Convergence of the CDMPC optimal solutions to the centralized optimal solutions can be observed in Figures 4.3a) and 4.3b) where the errors of the predicted input moves ($||\Delta U_{CDMPC} - \Delta U_{cen}||$) and predicted states ($||X_{CDMPC} - X_{cen}||$) are shown for the first control interval. These errors in the predicted variables are calculated as the difference between the CDMPC optimal solutions and the optimal solutions calculated with a centralized MPC controller. In the numerical simulations performed for the evaporator process, the fast convergence behaviour shown in Figure 4.3 (2 iterations) was observed for the coordination algorithm at each control interval.

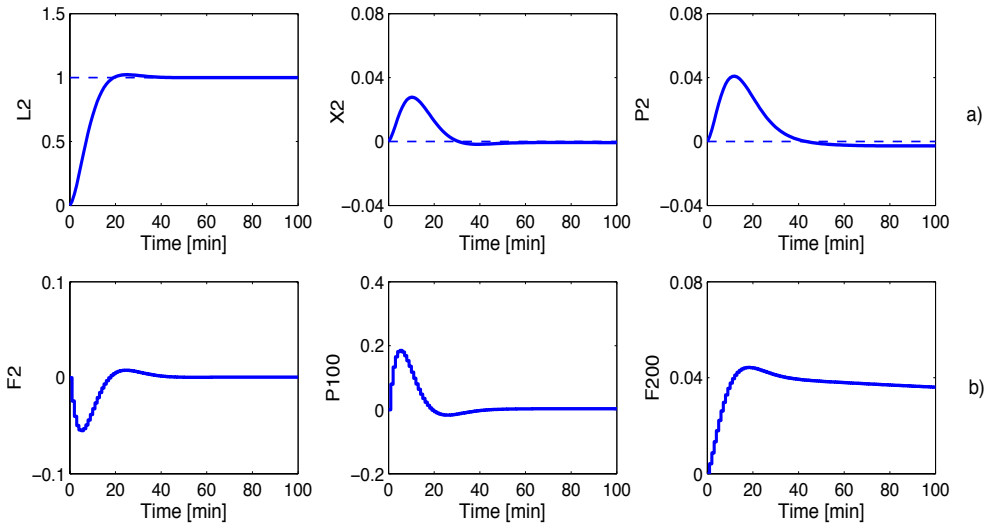

Figure 4.2: (a) Trajectories of state variables (solid line), targets for state variables (dashed line); (b)Trajectories of control inputs.

A comparison between the performance of the CDMPC controllers with the performance of decentralized controllers is shown in Figure 4.4. The decentralized

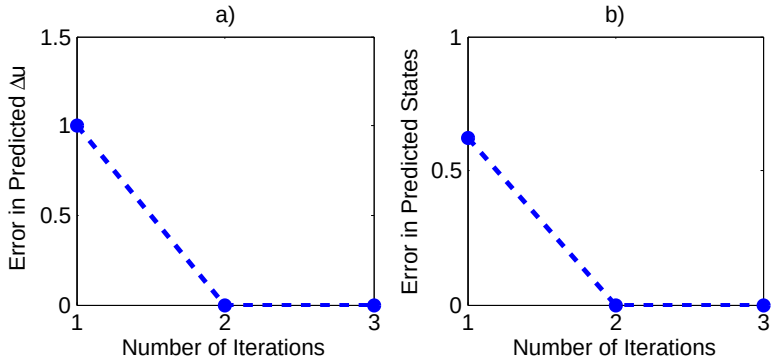

Figure 4.3: a) Error in predicted input moves ($||\Delta U_{CDMPC} - \Delta U_{cen}||$); b) Error in predicted states ($||X_{CDMPC} - X_{cen}||$).

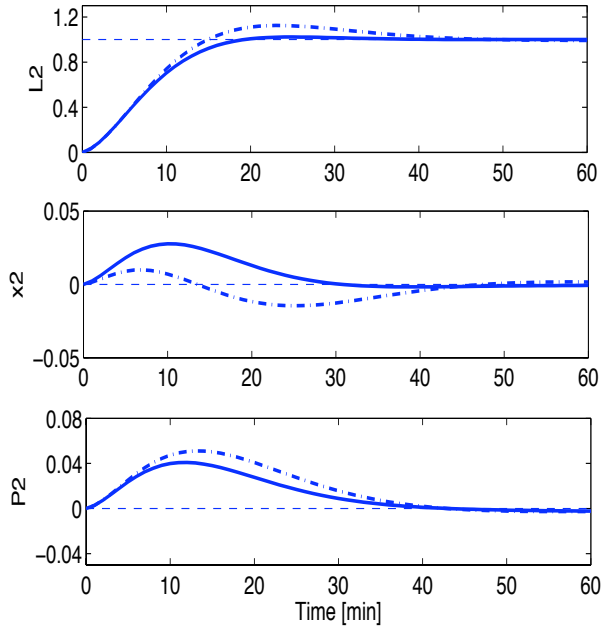

Figure 4.4: Comparison between trajectories obtained with CDMPC controller (solid line) and decentralized controllers (dash-dot line); targets for state variables (dashed line).

control formulation for each subsystem is given by equation (4.8). The simulations obtained with the decentralized controllers were performed with the same initial conditions and simulation parameters used for the CDMPC controllers. It can be observed in Figure 4.4 that the CDMPC controllers outperform the decentralized controllers. As shown in Figure 4.4, the trajectories obtained with the CDMPC

controllers are less oscillatory than the trajectories obtained with the decentralized controllers. The improvement in the performance is possible due to the information exchanged between the coordinator and the CDMPC controllers. Because of this information exchange, the coordinator is aware of the future predicted control moves in all the subsystems. Then, the coordinator assists the local CDMPC controllers in calculating the control moves that can compensate for the interactions between the subsystem, leading to a less oscillatory behaviour for the entire plant.

The improvement of the CDMPC controllers with respect to the decentralized controllers can be quantified using the following performance index:

$$\Pi = \frac{|\mathcal{J}_{dec} - \mathcal{J}_{CDMPC}|}{\mathcal{J}_{CDMPC}} \times 100\%, \tag{4.30}$$

where

$$\mathcal{J}_{(.)} = \frac{1}{2} \sum_{i=1}^{N} \sum_{k=0}^{t_f-1} \Big(||x_{i\_sp} - x_i(k+1)||^2_{Q_i} + ||\Delta u_i(k)||^2_{R_i} \Big), \tag{4.31}$$

and t_f is the simulation time. For the simulation studies performed with the evaporator process system, the performance index Π indicates that the CDMPC controllers outperform the decentralized controllers by approximately 2% (based on a 100 $[min]$ closed-loop simulation).

4.7 Extension: Formulation of CDMPC using Finite Step-Response Models

The CDMPC control scheme developed in Section 4.4 is extended here to use finite step-response models. This section shows that the proposed CDMPC control scheme is not limited only to state-space models, but it can also be implemented with other model representations, such as input-output models obtained from step-test data.

Consider that the process dynamics for each subsystem i, for $i = 1, ...N$, are represented by the following finite step-response coefficient model:

$$y_i(k+l) \;=\; \sum_{h=1}^{T-1} \sum_{w=1}^{N} S_{iw,h} \Delta u_w(k+l-h) + \sum_{w=1}^{N} S_{iw,T}\; u_w(k+l-T), \tag{4.32}$$

where $y_i(\cdot) \in \Re^{m_i}$ denote the output variables for subsystem i; $u_i(\cdot) \in \Re^{q_i}$ and $\Delta u_i(\cdot) \in \Re^{q_i}$ denote the input variables and input moves for subsystem i, respectively.

The matrix $S_{iw,h}$ in (4.32) denotes the h^{th} step-response coefficients for the model that relates y_i and Δu_w, for $i = 1, ...N$ and $w = 1, ...N$.

4.7.1 Decentralized MPC Control

The variables predicted by decentralized MPC controller i along the prediction horizon H_p and control horizon H_u are given by:

$$\hat{y}_i(k+l) = \sum_{h=1}^{l} S_{ii,h}\Delta\hat{u}_i(k+l-h) + \hat{y}_i^0(k+l) + \hat{d}_i(k+l|k), \quad \text{for } l = 1, ..., H_p, \quad (4.33)$$

with:

$$\begin{cases} \hat{d}_i(k+l|k) = y_i(k) - \hat{y}_i(k|k-1), & \text{for } l = 1, ..., H_p, \\ \Delta\hat{u}_i(k+l|k) = 0, & H_u \leq l \leq H_p. \end{cases} \quad (4.34)$$

The matrix $S_{ii,h}$ in (4.33) represents the step-response coefficients between the outputs and inputs in subsystem i for the h^{th} time step (for $h = 1, ..., H_p$). The vector $\hat{y}_i^0$ is the vector of predicted unforced responses and it accounts for past control actions (Seborg, 1989). The vector $\hat{d}_i(k+l|k)$ has been included in model (4.33) to correct, through feedback, for the discrepancies between the measured and predicted outputs. It is assumed that the difference between the measured and predicted outputs at time 'k' remains constant throughout the prediction horizon. The assumption of constant $\hat{d}_i(k+l|k)$ throughout the prediction horizon is one of the simplest assumption for estimation of disturbances and it has been used in the classical *Dynamic Matrix Control* (DMC) algorithms (Maciejowski (2002) and the references therein).

Typically, process industries need to deal with constraints on the process variables, input variables and input moves. The limitations on the subsystems' variables are expressed as inequality constraints in each decentralized control problem, as follows:

$$\begin{cases} y_{i\_min} \leq \hat{y}_i(k+l|k) \leq y_{i\_max}, \\ u_{i\_min} \leq \hat{u}_i(k+l|k) \leq u_{i\_max}, \\ \Delta u_{i\_min} \leq \Delta\hat{u}_i(k+\eta|k) \leq \Delta u_{i\_max}, \\ \text{for} \quad l = 1, \ldots, H_p, \quad \text{and} \quad \eta = 0, \ldots, H_u - 1. \end{cases} \quad (4.35)$$

Assuming that the vector of desired set-points (Y_{i_sp}) is provided, then the decentralized MPC control problem for each subsystem i (i=1,...,N) can be formulated

as:

$$\begin{aligned}
&\min_{Y_i,\Delta U_i} \ \Phi_i = \frac{1}{2}\Big(\big(Y_{i_sp} - Y_i(k)\big)^T \mathcal{Q}_i \big(Y_{i_sp} - Y_i(k)\big) + \Delta U_i(k)^T \mathcal{R}_i \Delta U_i(k)\Big) \\
&\text{subject to :} \\
&\qquad \begin{cases} \text{Equality constraints (4.33)} - \text{(4.34) and} \\ \text{Inequality constraints (4.35),} \end{cases}
\end{aligned} \tag{4.36}$$

where the vectors $Y_i(k)$ and $\Delta U_i(k)$, for $i = 1, ..., N$, are defined as:

$$\begin{aligned}
Y_i(k) &= [\hat{y}_i(k+1|k)^T, \hat{y}_i(k+2|k)^T, ..., \hat{y}_i(k+H_p|k)^T]^T, \\
\Delta U_i(k) &= [\Delta\hat{u}_i(k|k)^T, \Delta\hat{u}_i(k+1|k)^T, ..., \Delta\hat{u}_i(k+H_u-1|k)^T]^T.
\end{aligned} \tag{4.37}$$

The matrices $\mathcal{Q}_i$ and $\mathcal{R}_i$ in (4.36) are defined as $\mathcal{Q}_i = diag(Q_i(1), Q_i(2), \ldots, Q_i(H_p))$ and $\mathcal{R}_i = diag(R_i(0), R_i(1), \ldots, R_i(H_u - 1))$, where the weighting matrices $Q_i(.) \in \Re_+^{n_i \times n_i}$ and $R_i(.) \in \Re_+^{q_i \times q_i}$ are positive definite matrices with their off-diagonal elements equal to zero.

4.7.2 CDMPC Control

As described in Section 4.4, the CDMPC controllers can be designed by modifying the decentralized MPC controllers. Process model (4.33) is modified to include the interactions between subsystems as follows:

$$\begin{aligned}
\hat{y}_i(k+l) = &\sum_{h=1}^{l} S_{ii,h} \Delta\hat{u}_i(k+l-h) + \hat{y}_i^0(k+l) + \hat{d}_i(k+l|k) + \\
&\hat{v}_i(k+l|k), \qquad \text{for } l = 1, ..., H_p.
\end{aligned} \tag{4.38}$$

In the proposed control scheme, $\hat{v}_i$ are decision variables in the local CDMPC control problem. The coordinator in the price-driven method is formulated to ensure that equation (4.39) below is satisfied for each subsystem i, for $i = 1, ..., N$:

$$\begin{cases}
\Delta e_i(k+l|k) = \hat{v}_i(k+l|k) - \sum_{j\neq i} \big(\sum_{h=1}^{l} S_{ij,h} \Delta\hat{u}_j(k+l-h)\big) = 0, \\
\text{for } l = 1, ..., H_p, \\
\Delta\hat{u}_j(k+l|k) = 0, \qquad H_u \leq l \leq H_p,
\end{cases} \tag{4.39}$$

where the matrix $S_{ij,h}$ in (4.39) denotes the h^{th} step-response coefficients for the model that relates y_i and Δu_j, for $j \neq i$.

The CDMPC controller for each subsystem i (with $i = 1, ..., N$) is then formulated as follows:

$$\begin{aligned} &\min_{Y_i, \Delta U_i, V_i} \quad \mathcal{J}_i = \Phi_i + p^{s^T} \Theta_i [Y_i(k)^T, \Delta U_i(k)^T, V_i(k)^T]^T \\ &\text{subject to :} \\ &\qquad \begin{cases} \text{Equality constraints (4.34), (4.38) and} \\ \text{Inequality constraints (4.35),} \end{cases} \end{aligned} \tag{4.40}$$

where the objective function Φ_i is defined in (4.36). The vector of predicted outputs $Y_i(k)$ and predicted input moves $\Delta U_i(k)$ are defined in (4.37). The vector $V_i(k)$ in (4.40) is defined as: $V_i(k) = [\hat{v}_i(k|k)^T, \hat{v}_i(k+1|k)^T, ..., \hat{v}_i(k+H_p-1|k)^T]^T$.

Optimal control problem (4.40) can be written in a compact form as problem (4.13)-(4.14), where the vector of decision variables $Z_i(k)$ are defined here as: $Z_i(k) = [Y_i(k)^T, \Delta U_i(k)^T, V_i(k)^T]^T$. The price '$p^{s^T}$' is updated as described in Section 4.4.3. The CDMPC controllers using finite step-response models are implemented as described in Section 4.4.4.

4.7.3 Illustrative Example II

To show that the effectiveness of the CDMPC controllers using finite step-response models, a simulation study was performed on the fluid catalytic cracking (FCC) unit presented in (Grosdidier *et al.*, 1993), where a description of the FCC process, together with the limits for the controlled and manipulated variables, is provided. A diagram of the FCC system is shown in Figure 4.5.

In the FCC unit, gas oil is converted into hydrocarbons of shorter chains. The model of the FCC process and the models for the regulatory control loops are given in the Appendix 4.9.2 in Table 4.1 and Table 4.2, respectively. The continuous-time transfer function models include seven outputs and six inputs and were obtained through identification analysis of step-test data. The transfer function matrix for the entire FCC process can be obtained by multiplying each transfer function model in Table 4.1 by its corresponding input model in Table 4.2 (see Appendix 4.9.2), except for the transfer functions that relate output $y5$ with input $u5$ ($y5 - u5$) and output $y6$ with input $u5$ ($y6 - u5$), which do not require such multiplication (Grosdidier *et*

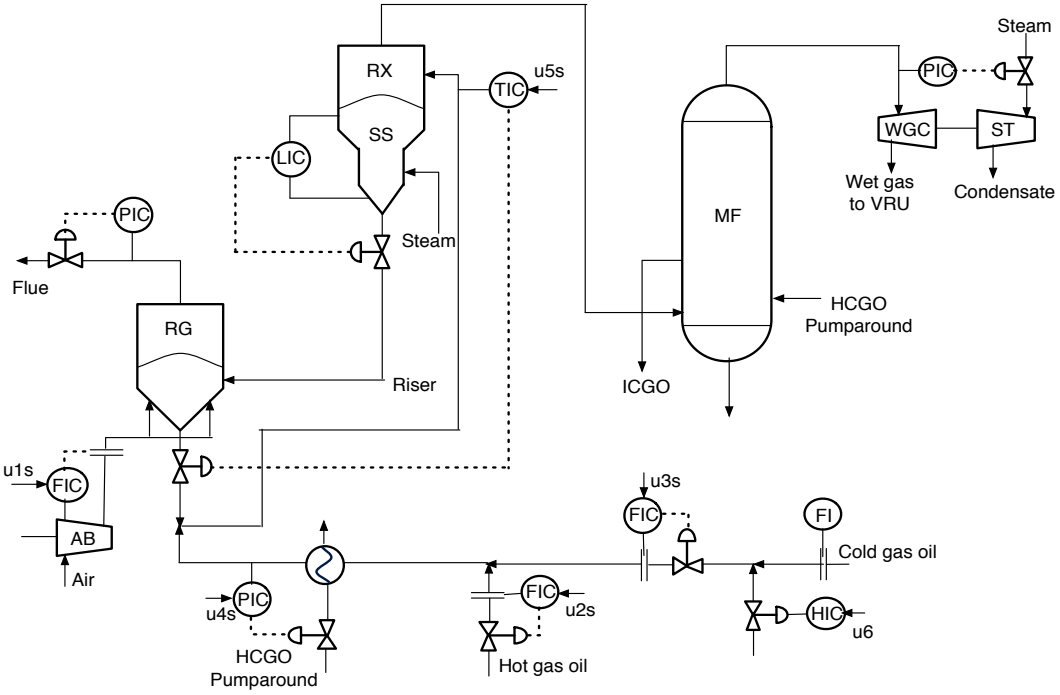

Figure 4.5: FCC process from (Grosdidier *et al.*, 1993)

al., 1993). The transfer functions for the output-input pairs $y5 - u5$ and $y6 - u5$ are used as given in Table 4.1.

To implement the CDMPC controllers, finite step-response models were obtained based on the process dynamics given in Table 4.1 and Table 4.2. These finite step-response models were also used to simulate the true plant dynamics. The sampling period used in the simulations was 1 $[min]$. This corresponds to the sampling period used in (Grosdidier *et al.*, 1993) for control of the FCC unit. It is assumed that the FCC process (4.29) consists of two subsystems. The first subsystem includes outputs $y1$ to $y3$ and inputs $u1$ to $u3$, while the second subsystem includes outputs $y4$ to $y7$ and inputs $u4$ to $u6$.

The following parameters are used in the simulation study: weighting matrices $Q_1(l) = diag\{5; 10; 5\}$ for subsystem 1 and $Q_2(l) = diag\{5; 5; 5; 0.001\}$ for subsystem 2, for $l = 1, ..., H_p$; $R_1(\eta) = diag\{100; 100; 100\}$ for subsystem 1 and $R_2(\eta) = diag\{100; 100; 100\}$ for subsystem 2, for $\eta = 0, ..., H_u - 1$. The prediction horizon H_p and the control horizon H_u considered for the computer simulations are 50 and

5, respectively. To initialize the coordination algorithm, all the elements in the price vector were set to zero.

A set-point change of magnitude 0.5 in output variable $y1$ was simulated at initial time $t = 0$, while the targets of the remaining outputs were kept at the origin. The results of the simulation are presented in Figure 4.6 and Figure 4.7. The closed-loop performance of the CDMPC controllers for subsystems 1 and 2 is shown in Figure 4.6, where the trajectories are plotted in deviation variables. It can be seen in Figure 4.6a) - Figure 4.6d) that the CDMPC controllers provide a good performance since output $y1$ achieves the new set-point and outputs $y2$ to $y7$ are stabilized at their new steady-state optimal values. Figure 4.7a) and Figure 4.7b) show the errors of the predicted

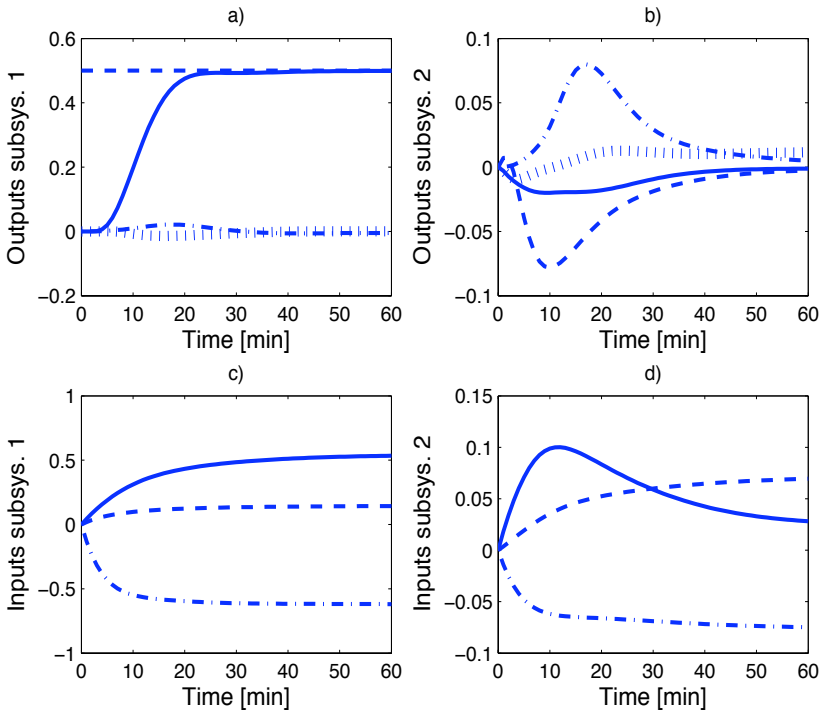

Figure 4.6: a) Outputs subsystem 1: set-point for $y1$ (dashed line), $y1$ (solid line), $y2$ (dotted line), $y3$ (dash-dot line); b) Outputs subsystem 2: $y4$ (dash-dot line), $y5$ (dashed line), $y6$ (solid line), $y7$ (dotted line); c) Inputs subsystem 1: $u1$ (dashed line), $u2$ (dash-dot line), $u3$ (solid line); d) Inputs subsystem 2: $u4$ (dash-dot line), $u5$ (solid line), $u6$ (dashed line).

input moves ($||\Delta U_{CDMPC} - \Delta U_{cen}||$) and predicted outputs ($||Y_{CDMPC} - Y_{cen}||$) for the optimization performed at the first sampling instant. These errors in the predicted

variables are calculated as the difference between the CDMPC optimal solutions and the optimal solutions calculated with a centralized MPC controller. It is shown in Figure 4.7a) and Figure 4.7b) that the solutions achieved with the CDMPC controllers converge to the centralized solution within 2 iterations. In the computer simulations performed for the FCC unit, the same fast convergence behaviour (2 iterations) was observed at each control interval.

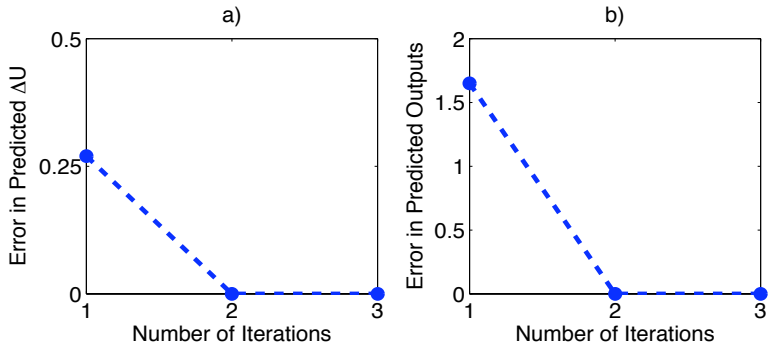

Figure 4.7: a) Error in predicted input moves ($||\Delta U_{CDMPC} - \Delta U_{cen}||$); b) Error in predicted outputs ($||Y_{CDMPC} - Y_{cen}||$).

For the simulation experiments performed with the FCC unit, the performance index Π, given by equation (4.30), indicates that the CDMPC controllers outperform the fully decentralized controllers by approximately 8.15% (based on a 60 $[min]$ closed-loop simulation).

4.8 Summary

In this chapter, a coordinated-distributed model predicted control scheme is proposed for discrete-time, linear constrained systems. The price-driven coordination algorithm is used to efficiently coordinate the local CDMPC controllers. Newton's method along with a sensitivity analysis technique are used to update the price vector in the price-driven coordination algorithm.

An analysis of the CDMPC scheme performance properties is provided in this chapter. Based on this analysis and the simulation experiments performed, the following results can be concluded:

- When the price-driven coordination algorithm converges:
  - the trajectories obtained with CDMPC controllers match the trajectories obtained with the centralized controller. Thus, the performance of the CDMPC controllers achieve the performance of the theoretical centralized controller.
  - the price vector is equivalent to the Lagrange multiplier vector for the equality constraints in the centralized control problem.
- The CDMPC controllers outperform the decentralized controllers because the coordinator is aware of the interacting variables among the subsystems and it can iteratively adjust the calculated optimal solutions of the local CDMPC controllers.
- A rapid convergence of the subsystems' operations towards the optimal plant-wide operations is achieved when using Newton-based approach as a coordination mechanism. In particular, when there is no change in the active set of the subproblems, the price-driven coordination algorithm converges within two iterations.
- The proposed CDMPC control scheme is not restricted only to processes whose dynamics are represented by state-space models. It is shown in this chapter that the CDMPC controllers can also be implemented using finite step-response models. This is a convenient property since many commercial MPC products still use step-response models in their formulation. Thus, the CDMPC controllers can be built directly on the existing decentralized MPC controllers, without the need to convert the step-response models into state-space models.

4.9 Appendix

4.9.1 Matrices Required to Update the Price Vector

The matrix Θ_i (for $i = 1, ..., N$) required in the price update mechanism and in the objective function of the subproblems is defined here for the CDMPC controllers

based on state-space models. For the CDMPC controllers that are formulated using finite step-response models, a different matrix Θ_i (which depends on the step-response coefficients) is required.

The matrix Θ_i (for $i = 1, ..., N$) includes the coefficients for the interacting variables. For example, for subsystems $i = 1$, $i = 2$ and $i = N$, Θ_i becomes:

$$\Theta_1 = \begin{bmatrix} G_{11} \\ \hdashline G_{21} \\ \hdashline \cdots \\ \hdashline G_{N1} \end{bmatrix}, \qquad \Theta_2 = \begin{bmatrix} G_{12} \\ \hdashline G_{22} \\ \hdashline \cdots \\ \hdashline G_{N2} \end{bmatrix} \quad \text{and} \quad \Theta_N = \begin{bmatrix} G_{1N} \\ \hdashline G_{2N} \\ \hdashline \cdots \\ \hdashline G_{NN} \end{bmatrix}. \tag{4.41}$$

The matrix G_{ii} is given by:

$$G_{ii} = [0,\ 0,\ I], \qquad \text{for} \quad i = 1, ..., N, \tag{4.42}$$

where the identity matrix, I, in equation (4.42) has dimensions $(H_p \cdot n_i) \times (H_p \cdot n_i)$. The matrices G_{ji} required in equation (4.41) are given by:

$$G_{ji} = [G_{A\_ji},\ G_{B\_ji},\ 0], \qquad \begin{array}{l} \text{for} \quad i = 1, ..., N, \\ \qquad j = 1, ..., N. \text{ with } j \neq i, \end{array} \tag{4.43}$$

with:

$$G_{A\_ji} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ -A_{ji} & 0 & 0 & \dots & 0 \\ 0 & \dots & & \ddots & \vdots \\ 0 & \dots & 0 & -A_{ji} & 0 \end{bmatrix}}_{(H_p \cdot n_i) \times (H_p \cdot n_j)}, \tag{4.44}$$

$$G_{B\_ji} = \underbrace{\begin{bmatrix} -B_{ji} & 0 & 0 & \dots & 0 \\ -B_{ji} & -B_{ji} & 0 & \dots & 0 \\ \dots & \dots & & \ddots & \vdots \\ -B_{ji} & -B_{ji} & \dots & -B_{ji} & -B_{ji} \end{bmatrix}}_{(H_p \cdot n_i) \times (H_u \cdot q_j)}. \tag{4.45}$$

4.9.2 Models Used in the Illustrative Example II

The model of the FCC process and the models for the regulatory control loops are shown in Table 4.1 and Table 4.2, respectively.

Table 4.1: FCC process models

	$u1$ [T/H]	$u2$ [M^3/H]	$u3$ [M^3/H]	$u4$ [^{o}C]	$u5$ [^{o}C]	$u6$ [%]
$y1$ [%]	$\frac{0.097(1.7s+1)e^{-2s}}{19s^2+6.5s+1}$	$\frac{-0.87e^{-2s}}{13s^2+4.9s+1}$	$\frac{-0.092(0.25s+1)e^{-3s}}{3.7s^2+4.7s+1}$	$\frac{0.026e^{-7s}}{12s+1}$	$\frac{-0.074(4.8s+1)}{9.3s^2+3.4s+1}$	$\frac{-(0.48s)e^{-12s}}{(6s+1)(8s+1)}$
$y2$ [^{o}C]	0	$\frac{0.55e^{-4s}}{27s^2+8.7s+1}$	$\frac{0.55e^{-4s}}{10s^2+4.9s+1}$	0	$\frac{0.74(1.7s+1)e^{-2s}}{11s^2+7.3s+1}$	$\frac{0.36e^{-11s}}{33s^2+6.5s+1}$
$y3$ [T/H]	0	$\frac{0.14e^{-11s}}{46s^2+8.5s+1}$	$\frac{0.14e^{-6s}}{46s^2+8.5s+1}$	0	$\frac{0.27(16s+1)}{53s^2+23s+1}$	$\frac{0.015(12s+1)e^{-9s}}{66s^2+27s+1}$
$y4$ [%]	0	$\frac{0.25e^{-11s}}{17s^2+7s+1}$	$\frac{0.25e^{-7s}}{3s+1}$	0	$\frac{0.70}{3s+1}$	$\frac{0.079(6.3s+1)e^{-10s}}{24s^2+12s+1}$
$y5$ [%]	0	$\frac{0.66e^{-s}}{2.5s+1}$	$\frac{0.66e^{-s}}{2.5s+1}$	$\frac{-0.9e^{-10s}}{6s+1}$	$\frac{1}{2s+1}$	$\frac{-0.54e^{-11s}}{9s+1}$
$y6$ [kPa]	0	$\frac{-0.84e^{-s}}{6.1s+1}$	$\frac{-0.90}{1.5s+1}$	$\frac{0.35e^{-10s}}{5s+1}$	$\frac{-(0.64s+1)}{13s^2+7s+1}$	$\frac{0.23(0.5s+1)e^{-14s}}{3.6s^2+11s+1}$
$y7$ [kPa]	0	$\frac{0.81}{6s+1}$	$\frac{0.90}{s+1}$	$\frac{-0.35e^{-10s}}{5s+1}$	0.80	$\frac{-0.26e^{-18s}}{7.1s+1}$

Table 4.2: Models between regulatory controller set-points $(u1s, ..., u6s)$ and inputs $(u1, ..., u6)$

$(u1s, u1)$	$(u2s, u2)$	$(u3s, u3)$	$(u4s, u4)$	$(u5s, u5)$	$(u6s, u6)$
$\frac{1}{(0.75s+1)(4.5s+1)}$	$\frac{1}{(s+1)}$	$\frac{1}{1.7s^2+2.1s+1}$	$\frac{(3.3s+1)e^{-s}}{40s^2+13s+1}$	$\frac{(0.64s+1)}{13s^2+7s+1}$	1

5

Dual-Rate Coordinated-Distributed MPC

This chapter addresses the problem of coordination of distributed MPC controllers that are executed at two different control rates (i.e., dual-rate CDMPC controllers). To the best of our knowledge, this is a first approach at solving dual-rate distributed control problems where a coordinator is involved in the distributed control scheme.

Three strategies are discussed to give an insight of different possible methods for coordination of dual-rate CDMPC controllers. Among these strategies, the one with most potential to improve the controllers' performance is further analyzed. The proposed dual-rate coordination algorithm is an extension of the price-driven coordination algorithm described in Chapter 4 for single-rate CDMPC controllers. A case study is used to illustrate the effectiveness of the proposed dual-rate CDMPC control scheme. Finally, a method for extending the dual-rate CDMPC control scheme to multi-rate CDMPC control scheme is proposed.

5.1 Terminology

Some terminology, required for the understanding of the dual-rate coordinated-distributed control problem addressed in this chapter, is defined in this section.

Sampling instant: specific time at which a continuous-time signal is sampled (see Figure 1.4, Chapter 1).

Sampling period: length of time between two consecutive sampling instants. It is denoted by the symbol 'T'.

Sampling intervals: set of times: $[0, T)$, $[T, 2T)$, $[2T, 3T)$, ... (see Figure 1.4, Chapter 1).

Control instant: instant in time at which the control calculations are performed. It is assumed that the control instants are the same as the sampling instants.

Control interval: period between two consecutive control calculations. It is assumed to be the same as the sampling interval.

Controller execution rate (or simply **control rate**): rate at which a controller performs its control calculations. For example, given two controllers: 'controller 1' and 'controller 2' that are executed every period T^1 and T^2, respectively. If $T^1 < T^2$, then over a fixed time frame, 'controller 2' is executed at a slower rate than 'controller 1'.

Computer clock: the individual MPC computer clock is used so that the MPC controller can perform its calculations at specific times. In this thesis, it is assumed that all the local controllers are synchronized across the plant by their computer clock.

5.2 Background

In previous chapters, it was assumed that the sampling period for all the measured variables across the plant was the same. It was also assumed that the local controllers in the plant executed their control calculations at the same control rate. In many plants, process units can have dynamics that span a range of time scales (e.g., processes with different time constants and settling times). Different sampling periods are generally used to sample the variables associated to process units with different dynamics. In addition, the controllers used to control process units with different

dynamics are also executed at different control rates. Figure 5.1 serves to explain this idea. The single-input single output process in 'subsystem 1' has a faster dynamic than the single-input single-output process in 'subsystem 2'. This can be concluded because the time constant and settling time in 'subsystem 1' are smaller than the time constant and settling time in 'subsystem 2'. The output in 'subsystem 1' is expected to be sampled with a sampling period shorter than the one used to sample the output in 'subsystem 2'. Because the length of the control intervals is assumed to be the same as the sampling period, 'controller 1' in Figure 5.1 is expected to be executed at a faster rate than 'controller 2'.

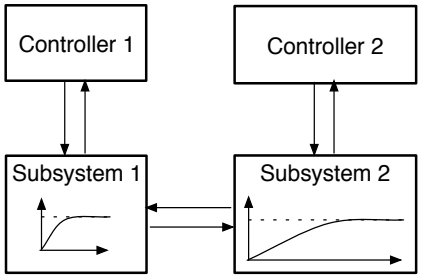

Figure 5.1: Schematic of two subsystems with different dynamics

Coordination of distributed controllers that are executed at different control rates is the topic of this chapter. We first explain the difference between 'multi-rate distributed control' and 'asynchronous distributed control', since the latter term has been used in the literature in the general sense to solve both type of problems.

Multi-rate distributed control involves communication between local distributed controllers that are executed at different control rates; however, the control calculations performed by the different distributed controllers are synchronized at some specific control instants. This implies that all control rates are integer multiples of some base rate, and that a clock time exists when all the local distributed controllers are synchronous. The concept of multi-rate control can be illustrated in Figure 5.2. In Figure 5.2, the control intervals for 'controller 1' are shorter than the control intervals for 'controller 2'. Nevertheless, both controllers are synchronized at certain control instants.

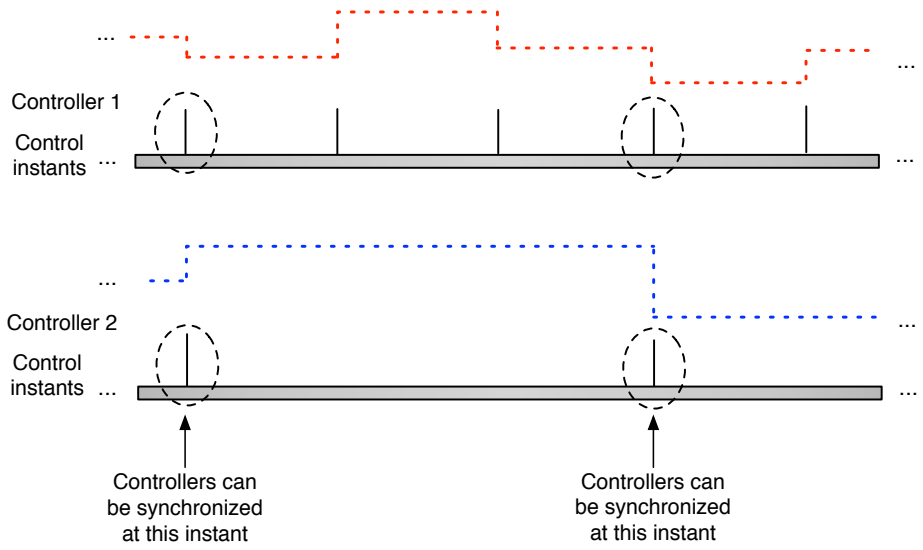

Figure 5.2: Control instants in multi-rate control. Control input moves are represented by dotted lines

The definition of asynchronous distributed control has been more broadly used in the literature. Asynchronous distributed control involves communication between distributed controllers that are not synchronized by their computer clock (e.g., Venkat (2006)). In addition, the asynchronous distributed control schemes can include distributed controllers that are executed at the same or different control rates (see Figure 5.3). In Figure 5.3a), 'controller 1' and 'controller 2' are executed at the same control rate, but they are not synchronized. In Figure 5.3b), 'controller 1' and 'controller 2' are executed at different control rates and different control instants.

Distributed control schemes including controllers that are executed at different control rates and controllers or agents with asynchronous communications have recently attracted the attention of many researchers (Nedić and Ozdaglar, 2007; Mehyar *et al.*, 2005; Androulakis and Reklaitis, 1999; Camponogara and Talukdar, 2007; Camponogara *et al.*, 2002). Liu *et al.* (2010) designed distributed model predictive controllers that take asynchronous and delayed measurements. Venkat (2006) proposed two asynchronous distributed MPC control schemes. In the first scheme, the frequency of information exchanged between the subsystems depends on the required computational time of the cooperative distributed MPC controllers.

In the second scheme, asynchronous feedback policies are used for cooperative distributed MPC controllers that are executed at two different rates. Camponogara and coworkers proposed some heuristics for asynchronous communication among distributed MPC controllers (Camponogara *et al.*, 2002; Talukdar and Camponogara, 2001). Two heuristics considered in (Camponogara *et al.*, 2002) include: a) estimation of neighboring agents' actions; and b) inclusion of some resource margins to make the constrained variables in the optimization problems more conservative. Nevertheless, to our knowledge, there is no work published regarding asynchronous distributed control or multi-rate distributed control where a coordinator communicates with the local controllers. The work presented in this chapter is a first approach at solving multi-rate coordinated-distributed control problems.

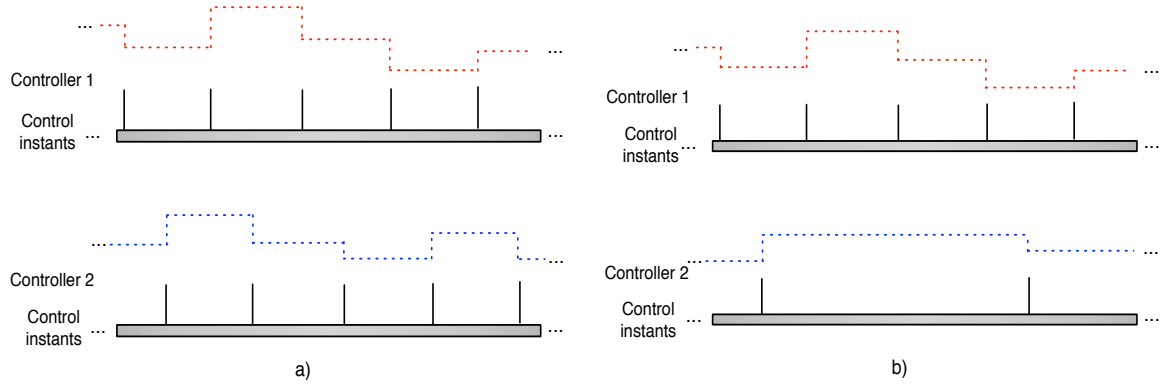

Figure 5.3: Control instants in asynchronous control: a) Same control rates, but different control instants; b) different control rates and control instants. Control input moves are represented by dotted lines

In this chapter, a method for coordination of CDMPC controllers that are executed at different control rates is presented. In particular, a dual-rate case is used for the examples and schematics. This allows us to simplify the notation and the presentation of the problem that needs to be solved.

In the dual-rate control problem, two classes of subsystems are considered. One class of subsystems is referred to as the 'fast' subsystems and it includes subsystems with faster dynamics. The other class of subsystems is referred to as the 'slow' subsystems and it includes subsystems with slower dynamics. To identify the variables

and coefficients in the subsystems with the distinct dynamics, the superscripts 'f' and 'sl' are appended to the fast subsystems and slow subsystems, respectively. In the fast subsystems, measurements of the process variables are available every fast sampling period T^f and the inputs are updated every fast period T^f, as well. In the slow subsystems, measurements of the process variables are available every slow sampling period T^{sl} and the inputs are also updated every slow period T^{sl}. For the dual-rate systems considered in this work, it is assumed that the slow sampling period is a multiple of the fast sampling period; that is, $T^{sl}/T^f = m$, where 'm' is a positive integer number. Moreover, over the period $[\tau T^{sl}, (\tau+1)T^{sl}]$, we have: $\tau T^{sl} < \tau T^{sl} + T^f < \cdots < \tau T^{sl} + (m-1)T^f < (\tau+1)T^{sl}$.

Three possible strategies for the dual-rate CDMPC control scheme are identified in this chapter. Among the three strategies, the one that requires fewer modifications to the existing decentralized controllers and, at the same time, can improve their performance is further investigated. The proposed dual-rate CDMPC control scheme is formulated based on ideas derived from the *lifting technique*. Finally, an extension of the dual-rate CDMPC control scheme to multi-rate CDMPC control scheme is presented along with a discussion of the challenges associated to the proposed multi-rate CDMPC control scheme.

5.3 Coordination of Dual-Rate CDMPC Controllers

Three different strategies (i.e., *strategies I, II* and *III*) are proposed in this section for coordination of dual-rate distributed controllers. Among the three strategies, only *strategy III* is further developed in detail and tested through computer simulations. The strategies and computer simulations considered in this chapter are for dual-rate controllers. An extension of dual-rate coordinated-distributed control schemes to multi-rate coordinated-distributed control schemes is possible using the proposed approach and it is explained for *strategy III* in Section 5.6.

5.3.1 Strategy I: Fast-Rate Coordination

In this strategy, a coordinator that operates at a fast rate is included in the distributed control scheme to coordinate the local CDMPC controllers. It is assumed that:

- the models that represent the dynamics of the fast and the slow subsystems are provided in discrete-time form based on the fast sampling period T^f;
- the control inputs, including the inputs from the slow subsystems are updated every fast period T^f.

The implementation of the coordinated-distributed control scheme using *strategy I* can be summarized as follows:

- all the controllers in the plant, including the controllers for the slow subsystems execute their control calculations at the fast rate (i.e., every fast period T^f);
- the coordinator is also executed every fast period T^f.

In order to coordinate all the controllers in the plant at the fast rate, the controllers need to be executed at the fast rate as well, including the controllers from the slow subsystems. One disadvantage of this strategy is that the inter-sample data (i.e., the data at times: $\tau T^{sl} + T^f, \ldots, \tau T^{sl} + (m-1)T^f$) is not available for the slow subsystems. Therefore, a data estimator is required to predict the unavailable inter-sample data in the slow subsystems (see Figure 5.4). The data predicted by the inter-sample data estimator can only be adjusted through feedback at the slow sampling instants. Thus, any mismatch between the predicted inter-sample data and the true values of the slow process variables will have an impact on the entire plant-wide performance. This becomes a more significant problem as the fast and slow sampling periods become considerably different (or $T^f \ll T^{sl}$). Since the plant-wide performance can deteriorate as a result of the slow subsystems' unavailable data at the inter-sample instants, this strategy is not further investigated in this work.

5.3.2 Strategy II: Slow-Rate Coordination

In this strategy, a coordinator that operates at a slow rate is included in the distributed control scheme to coordinate the local CDMPC controllers (see Figure

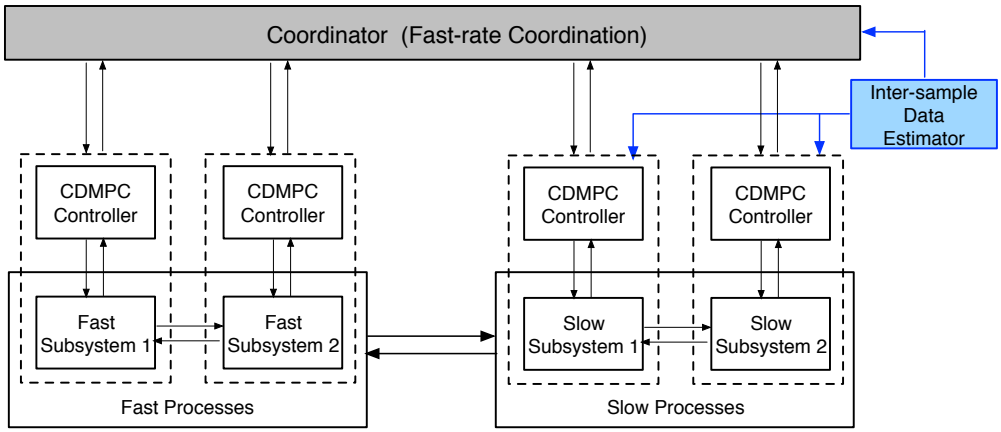

Figure 5.4: Hierarchy for *strategy I*: Fast-rate coordination

5.5). It is assumed that the models that represent the dynamics of the fast and the slow subsystems are provided in discrete-time form based on the slow sampling period T^{sl}. The implementation of the coordinated-distributed control scheme using *strategy II* can be summarized as follows:

- all the controllers in the plant, including the controllers for the fast subsystems execute their control calculations at the slow rate (i.e., every slow period T^{sl});
- the coordinator is also executed every slow period T^{sl}.

To coordinate all the controllers in the plant at the slow rate, the controllers need to be executed at the slow rate as well, including the controllers from the fast subsystems. One disadvantage of this strategy is that, while it is possible to measure the fast subsystems' outputs at the fast sampling instants, the controllers of the fast subsystems are executed every slow interval of length T^{sl}. This can have a negative impact on the plant-wide performance. It limits the ability to reject high frequency disturbances in the fast subsystems, which degrades the plant-wide performance. This problem becomes more significant as $T^{f} \ll T^{sl}$. When the controllers in the fast subsystems are executed at a slower rate, their effectiveness can be substantially reduced. This results from the fact that in between the slow sampling instants, the controller in each fast subsystem operates as an open-loop subsystem because no control action takes place. Thus, if the control interval T^{sl} is long for the dynamics of

the fast subsystems, the performance of the controllers in the fast subsystems clearly deteriorates, affecting the entire plant-wide performance.

Although *strategies I* and *II* can provide a feasible solution to the dual-rate coordinated-distributed control problem, they might not result in the best possible performance for the entire plant due to the selected optimization rate. A third strategy (*strategy III*) for solving dual-rate distributed control problems is presented in the next section. This strategy has the most potential to improve the entire plant-wide performance because it consists of a trade-off between *strategy I* and *II*.

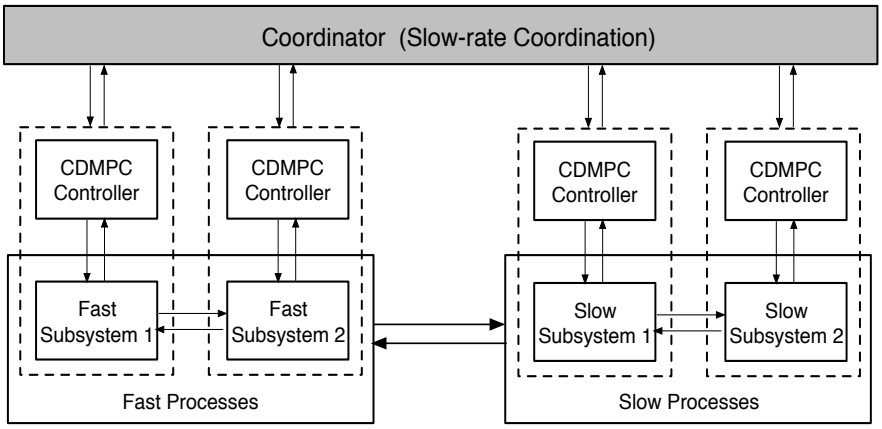

Figure 5.5: Hierarchy for *strategy II*: Slow-rate coordination

5.3.3 Strategy III: Dual-Rate Coordination

Strategy III is another alternative for solving dual-rate distributed control problems. This strategy takes advantage of both controller execution rates: the controllers in the fast subsystems perform their calculations at the fast rate, while the controllers in the slow subsystems perform their calculations at the slow rate. The hierarchy for *strategy III* can be represented by Figure 5.6, where the coordinator has dual coordination modes. Due to the interactions between the fast and the slow subsystems, the subsystems include mixed-rate variables. The *lifting technique* (see Section 5.4.1) is used in this work as a tool to obtain the mixed-rate models for the slow subsystems.

The implementation of the coordinated-distributed control scheme using *strategy III* can be summarized as follows:

- each controller in the plant is executed at its own control rate. That is, the controllers of the fast subsystems are executed every fast period T^f and the controllers of the slow subsystems are executed every slow period T^{sl};
- the coordinator is executed every fast period T^f and it has a multiple coordination modes that vary according to the current control instant. For example, every slow period T^{sl}, all the controllers in the plant are coordinated, including the controllers from the fast and the slow subsystems (see control instants: 'τT^{sl}' and '$(\tau+1)T^{sl}$' in Figure 5.7). Nevertheless, every fast period T^f and between two consecutive slow control instants, only the controllers from the fast subsystems are coordinated (see control instants: '$\tau T^{sl} + T^f$' and '$\tau T^{sl} + (m-1)T^f$' in Figure 5.7).

Among the three strategies, *strategy III* is seen as the most promising. It does not require the controllers to alter their execution rate because each controller optimizes the local subsystem according to their designed control intervals. Moreover, *strategy III* does not degrade the entire plant-wide performance because it takes advantage of the existing dual-rate control strategy. Since *strategy III* offers a potential improvement of the entire plant-wide performance, a mathematical framework for *strategy III* is derived in Section 5.4.2, and its performance is tested through computer simulations in Section 5.5.

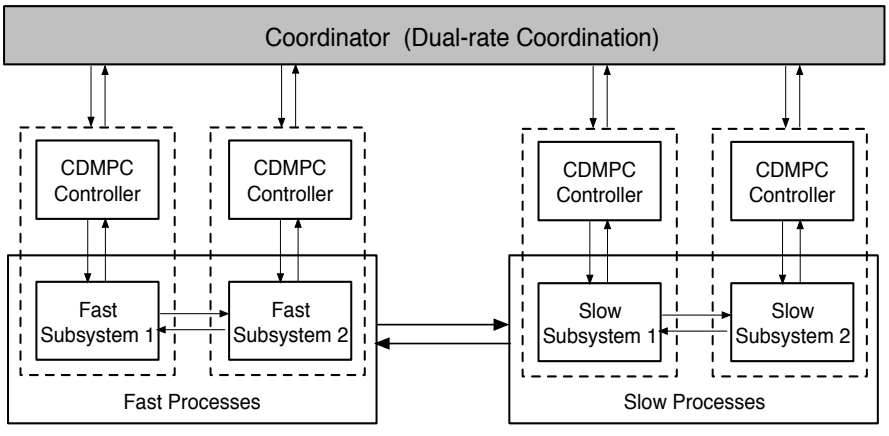

Figure 5.6: Hierarchy for *strategy III*: Dual-rate coordination

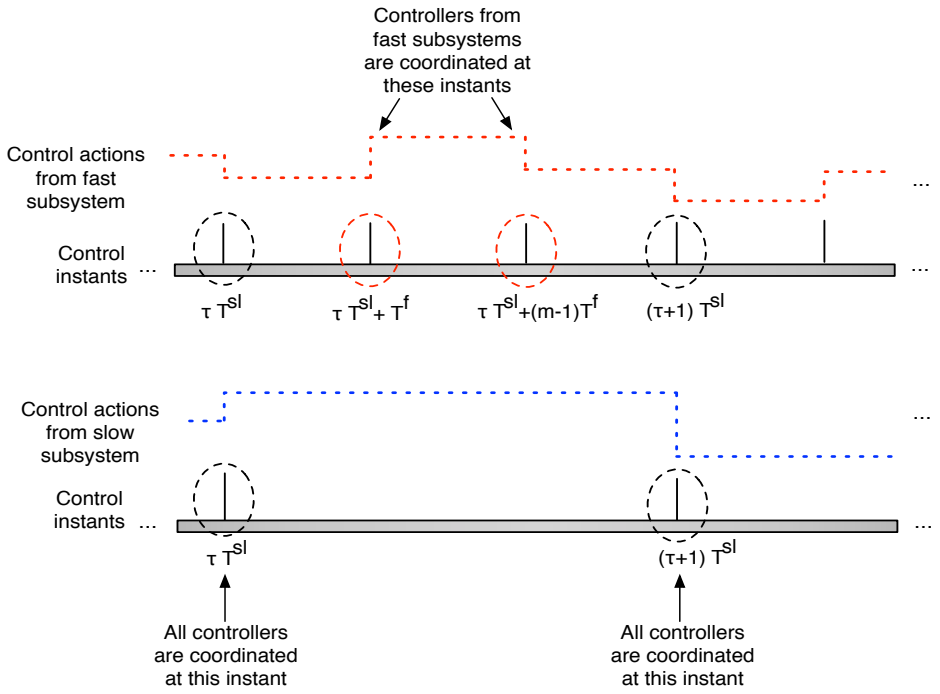

Figure 5.7: Dual coordination mode in *strategy III*

5.4 Dual-Rate CDMPC Control

In this section, the mathematical framework for the dual-rate distributed MPC controllers is derived based on *strategy III*. The *lifting technique*, which is used as a tool to obtain the models for the slow subsystems, is described in Section 5.4.1. The models for the slow and the fast subsystems are given in Section 5.4.2. The distributed control problem for each subsystem in the plant is formulated in Section 5.4.3. Finally, a price-driven algorithm used for coordination of the dual-rate CDMPC controllers is proposed in Section 5.4.4 along with the implementation of the dual-rate CDMPC controllers in Section 5.4.5.

5.4.1 Lifting Technique

One of the standard methods for solving multi-rate control problems is the so-called *lifting technique* (Chen and Francis, 1995; Sheng *et al.*, 2002; Rossiter *et al.*, 2005; Sheng *et al.*, 2001). The lifting technique consists of grouping (or 'lifting')

the fast-rate signals together to create slow-rate signals of higher dimension (Chen and Francis, 1995).

To understand the lifting technique, let us consider the single-input single-output sampled-data system shown in Figure 5.8. The system shown in Figure 5.8 is a dual-rate sampled-data system because the input updating instants and the output sampling instants occur at two different rates. The discrete-time input signal z is updated at a fast rate with period T^f and the output signal is sampled at a slow rate with period T^{sl}, where $T^{sl} = mT^f$. The fast-rate input signal is shown in Figure 5.8 with fast frequency dots and the slow-rate output signal is shown with slow frequency dots.

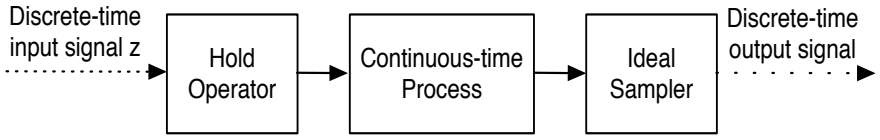

Figure 5.8: A single-input single-output dual-rate sampled-data system

Let the discrete-time input signal z be: $z = [z(0), z(1), z(2), \dots]$. For the positive integer m, the m-fold lifted signal $\underline{z}$ is defined as:

$$\underline{z} = \left\{ \begin{bmatrix} z(0) \\ z(1) \\ \vdots \\ z(m-1) \end{bmatrix}, \begin{bmatrix} z(m) \\ z(m+1) \\ \vdots \\ z(2m-1) \end{bmatrix}, \dots \right\}.$$

Note that the lifted signal is underlined and that the dimension of the lifted signal $\underline{z}$ equals m times the dimension of signal z (Chen and Francis, 1995). In this work, the lifting technique is used to derive a discrete-time state-space models for the slow subsystems from their continuous-time models (see Section 5.4.1).

Lifted Models for Systems with Slow Dynamics

Let us consider a continuous-time system modeled by the state-space representation as follows:

$$\dot{x}^{sl}(t) = A_c^{sl} x^{sl}(t) + B_c^{sl} u^{sl}(t) + D_c^{sl-f} u^f(t), \tag{5.1}$$

where the states $x^{sl} \in \Re^{n^{sl}}$, the inputs $u^{sl} \in \Re^{q^{sl}}$ and the inputs $u^{f} \in \Re^{q^{f}}$. As shown in Figure 5.8, dual-rate systems consists of systems where the input updating instants and the output sampling instants occur at two different rates. Let us assume that in system (5.1), the states x^{sl} are sampled every slow period T^{sl}, the inputs u^{sl} are updated every slow period T^{sl} and the inputs u^{f} are updated every fast period T^{f}. For the dual-rate system considered here, it is assumed that the slow sampling period is a multiple of the fast sampling period. That is, $T^{sl}/T^{f} = m$, where m is a positive integer. Thus, for the period T^{sl} shown in Figure 5.9, it is assumed that:

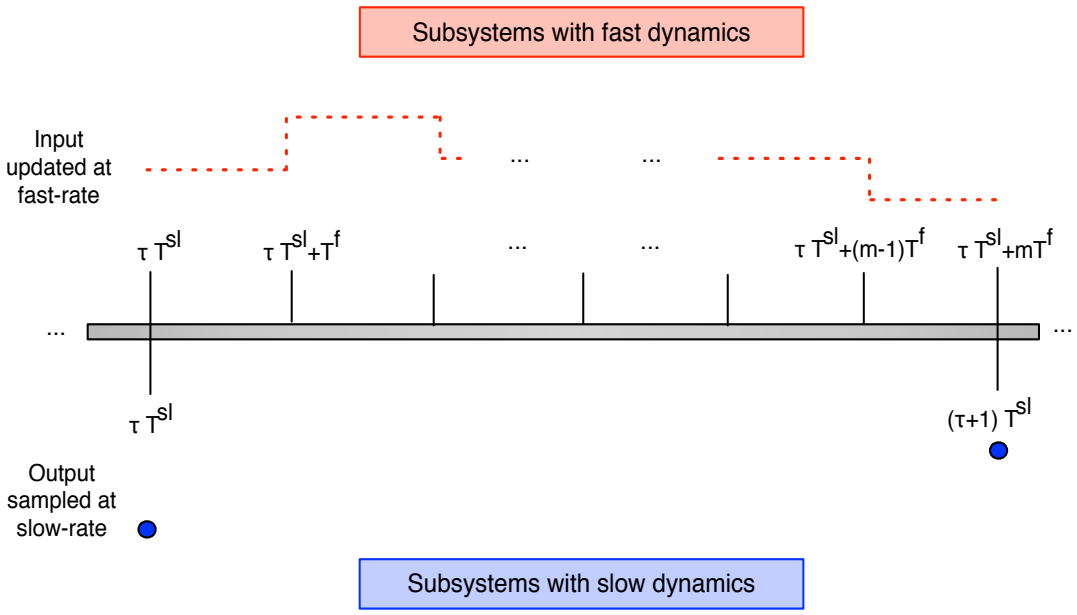

Figure 5.9: Example of output sampling and input update scheme for a dual-rate system

- over the interval $[\tau T^{sl}, (\tau+1)T^{sl})$, the control signals u^{f} are updated at the fast time instants: τT^{sl}, $\tau T^{sl} + T^{f}, \ldots, \tau T^{sl} + (m-1)T^{f}$;
- over the interval $[\tau T^{sl}, (\tau+1)T^{sl})$, the control signals u^{sl} are updated at the slow time instant τT^{sl} and the states (or outputs) are available at the slow time instant τT^{sl}.

Continuous-time model (5.1) can be discretized to give:

$$x^{sl}(\tau+1) \;=\; A^{sl}x^{sl}(\tau) + B^{sl}u^{sl}(\tau) + \underline{D}^{sl-f}\underline{u}^{f}(\tau), \tag{5.2}$$

where the matrices A^{sl} and B^{sl} in equation (5.2) are defined as:

$$A^{sl} = e^{A_c^{sl} \cdot T^{sl}}, \tag{5.3}$$

$$B^{sl} = \int_0^{T^{sl}} \left(e^{A_c^{sl} \cdot t} \right) B_c^{sl} \, dt. \tag{5.4}$$

The lifted matrix $\underline{D}^{sl-f}$ and lifted vector $\underline{u}^f(\tau)$ in equation (5.2) are obtained following the lifted modeling technique in (Sheng *et al.*, 2002) as:

$$\underline{D}^{sl-f} = [D^{sl-f}(0) \;\; D^{sl-f}(1) \;\; \ldots \;\; D^{sl-f}(m-1)],$$

$$D^{sl-f}(\sigma) = \int_{\left(1-\frac{\sigma+1}{m}\right)T^{sl}}^{(1-\frac{\sigma}{m})T^{sl}} \left(e^{A_c^{sl} \cdot t} \right) D_c^{sl-f} \, dt, \qquad \sigma = 0, ..., m-1,$$

$$\underline{u}^f(\tau) = \begin{bmatrix} u^f(\tau T^{sl}) \\ u^f(\tau T^{sl} + T^f) \\ \cdots \\ u^f(\tau T^{sl} + (m-1) \cdot T^f) \end{bmatrix}.$$

Equation (5.2) can be arranged in terms of the lifted input moves $\underline{\Delta u}^f(\tau)$ as follows:

$$x^{sl}(\tau+1) \;\; = \;\; A^{sl} x^{sl}(\tau) + B^{sl} u^{sl}(\tau) + \underline{\mathcal{D}}^{sl-f} \; \underline{\Delta u}^f(\tau) + \mathcal{D}^{sl-f}(0) u^f(\tau - 1/m), \tag{5.5}$$

where A^{sl} and B^{sl} are given by equations (5.3) and (5.4), respectively, and

$$\underline{\mathcal{D}}^{sl-f} = [\mathcal{D}^{sl-f}(0) \; \mathcal{D}^{sl-f}(1) \; \ldots \; \mathcal{D}^{sl-f}(m-1)],$$

$$\mathcal{D}^{sl-f}(\sigma) = \int_0^{\left(\frac{m-\sigma}{m}\right)T^{sl}} \left(e^{A_c^{sl} \cdot t} \right) D_c^{sl-f} \, dt, \qquad \sigma = 0, ..., m-1, \tag{5.6}$$

$$\underline{\Delta u}^f(\tau) = \begin{bmatrix} \Delta u^f(\tau T^{sl}) \\ \Delta u^f(\tau T^{sl} + T^f) \\ \cdots \\ \Delta u^f(\tau T^{sl} + (m-1) \cdot T^f) \end{bmatrix}.$$

5.4.2 Models for the Slow and Fast Subsystems

To formulate the dual-rate CDMPC controllers, it is assumed that the entire plant consists of two sets of subsystems: slow subsystems and fast subsystems.

With regards to the interactions, it is also assumed that the slow subsystems are coupled to the fast subsystems through the control inputs only. As discussed in

Section 5.3.1, the state variables in the slow subsystems are measured every slow sampling period T^{sl}. At the fast sampling instants $\tau T^{sl} + T^{f}, \ldots, \tau T^{sl} + (m-1)T^{f}$, the state variables in the slow subsystems are not available and they need to be estimated. Estimation of unavailable data is not pursued in this thesis. Therefore, in this chapter it is assumed that the interactions between the slow subsystems and the fast subsystems represent the effect of the input variables only.

Model dynamics for slow subsystems:

It is assumed that the continuous-time model (5.1) is available for the slow subsystems and it is discretized as shown in Section 5.4.1. The notation and the mixed-rate models used in the lifting technique can be applied to obtain the discrete-time model for all the slow subsystems as in equation (5.2). Next, model (5.2) can be partitioned into N^{sl} slow subsystems. The discrete-time process model for each slow subsystem is given by:

$$\begin{aligned} x_i^{sl}(\tau+1) &= A_{ii}^{sl} x_i^{sl}(\tau) + B_{ii}^{sl} u_i^{sl}(\tau) + \sum_{j \neq i} \left(A_{ij}^{sl} x_j^{sl}(\tau) + B_{ij}^{sl} u_j^{sl}(\tau) \right) + \\ &\qquad \sum_{w=1}^{N^f} \underline{D}_{iw}^{sl-f} \underline{u}_w^f(\tau), \qquad (5.7) \\ x_i^{sl}(0) &= x_i^{init\_sl}, \quad \text{for } i = 1, ..., N^{sl}, \end{aligned}$$

where N^{sl} is the total number of slow subsystems; N^f is the total number of fast subsystems; $x_i^{sl}(\tau) \in \Re^{n_i^{sl}}$ and $u_i^{sl}(\tau) \in \Re^{q_i^{sl}}$. The total number of state variables in all the slow subsystems is $n^{sl} = \sum_{i=1}^{N^{sl}} n_i^{sl}$, and the total number of input variables in all the slow subsystems is $q^{sl} = \sum_{i=1}^{N^{sl}} q_i^{sl}$.

The matrices A_{ii}^{sl} and B_{ii}^{sl} represent the local dynamics for the slow subsystem i (for $i = 1, ..., N^{sl}$), and the matrices A_{ij}^{sl} and B_{ij}^{sl} represent the effect of the slow subsystems j (with $j \neq i$) on the slow subsystem i. The lifted matrix $\underline{D}_{iw}^{sl-f}$ accounts for the effect of the lifted inputs $\underline{u}_w^f$ on the slow subsystem i, where $i = 1, ..., N^{sl}$ and $w = 1, ..., N^f$.

Model dynamics for fast subsystems:

It is assumed that the discrete-time model for each subsystem with fast dynamics is given by:

$$x_i^f(k+1) = A_{ii}^f x_i^f(k) + B_{ii}^f u_i^f(k) + \sum_{j \neq i} \left(A_{ij}^f x_j^f(k) + B_{ij}^f u_j^f(k) \right) + \sum_{w=1}^{N^{sl}} D_{iw}^{f-sl} u_w^{sl}(k), \tag{5.8}$$

$$x_i^f(0) = x_i^{init\_f}, \quad \text{for } i = 1, ..., N^f,$$

where $x_i^f(k) \in \Re^{n_i^f}$ and $u_i^f(k) \in \Re^{q_i^f}$. The total number of state variables in all the fast subsystems is $n^f = \sum_{i=1}^{N^f} n_i^f$, and the total number of input variables in all the fast subsystems is $q^f = \sum_{i=1}^{N^f} q_i^f$.

The matrices A_{ii}^f and B_{ii}^f represent the local dynamics for fast subsystem i (for $i = 1, ..., N^f$) and the matrices A_{ij}^f and B_{ij}^f represent the effect of the fast subsystems j (with $j \neq i$) on the fast subsystem i. The matrix D_{iw}^{f-sl} accounts for the effect of inputs u_w^{sl} on the fast subsystem i, where $i = 1, ..., N^f$ and $w = 1, ..., N^{sl}$.

It can be noted that the total number of slow and fast subsystems in the plant are equal to $N^{sl} + N^f$; the total number of state variables in the plant are equal to $n^{sl} + n^f$; and the total number of input variables in the plant are equal to $q^{sl} + q^f$.

5.4.3 Formulation of Dual-Rate CDMPC Controllers

In this section, the formulation of the CDMPC controllers is presented for the slow and the fast subsystems. The price-driven method described in Chapter 4 is extended to coordinate the dual-rate CDMPC controllers. In the price-driven coordination method, the price vector used in the local optimization problems is updated based on Newton's method.

Slow-rate CDMPC controllers

Consider the dynamics of the slow subsystems i (for $i = 1, ..., N^{sl}$) represented by model (5.7). The variables predicted by the CDMPC controller of each slow subsystem

i along the prediction horizon H_p^{sl} and control horizon H_u^{sl} are given by:

$$\begin{cases} \hat{x}_i^{sl}(\tau+l+1|\tau) = A_{ii}^{sl}\hat{x}_i^{sl}(\tau+l|\tau) + B_{ii}^{sl}\hat{u}_i^{sl}(\tau+l|\tau) + \hat{v}_i^{sl}(\tau+l|\tau) + \\ \qquad \sum_{j\neq i}\left(\beta_i^{sl}A_{ij}^{sl}x_j^{sl}(\tau) + B_{ij}^{sl}u_j^{sl}(\tau-1)\right) + \\ \qquad \sum_{w=1}^{N^f}\mathcal{D}_{iw}^{sl-f}(0)u_w^f(\tau-1/m), \qquad \text{for } l=0,..,H_p^{sl}-1, \\ \hat{x}_i^{sl}(\tau|\tau) = x_i^{sl}(\tau), \\ \beta_i^{sl} = 1, \quad \text{for } l=0, \\ \beta_i^{sl} = 0, \quad \text{for } l=1,..,H_p^{sl}-1, \end{cases} \tag{5.9}$$

with:

$$\begin{cases} \hat{u}_i^{sl}(\tau+l|\tau) = \sum_{\nu=0}^{l}\Delta\hat{u}_i^{sl}(\tau+\nu|\tau) + u_i^{sl}(\tau-1), \quad \text{for } l=0,...,H_p^{sl}-1, \\ \Delta\hat{u}_i^{sl}(\tau+l|\tau) = 0, \qquad H_u^{sl} \leq l \leq H_p^{sl}-1. \end{cases} \tag{5.10}$$

In the proposed control scheme, $\hat{v}_i^{sl}$ are decision variables in the CDMPC control problem of each slow subsystem i. The coordinator is formulated to ensure that the following equation (5.11) is satisfied for each slow subsystem i, for $i=1,...,N^{sl}$:

$$\begin{cases} \Delta e_i^{sl}(\tau+l|\tau) = 0 = \\ \hat{v}_i^{sl}(\tau+l|\tau) - \sum_{j\neq i}\left(\beta_i^{sl}A_{ij}^{sl}\hat{x}_j^{sl}(\tau+l|\tau) + B_{ij}^{sl}\sum_{\nu=0}^{l}\Delta\hat{u}_j^{sl}(\tau+\nu|\tau)\right) + \\ \qquad -\sum_{w=1}^{N^f}\beta_i^{sl}\mathcal{D}_{iw}^{sl-f}(0)\underline{I}_w^f\sum_{\nu=0}^{l-1}\underline{\Delta\hat{u}}_w^f(\tau+\nu|\tau) + \\ \qquad -\sum_{w=1}^{N^f}\underline{\mathcal{D}}_{iw}^{sl-f}\underline{\Delta\hat{u}}_w^f(\tau+l|\tau), \qquad \text{for } l=0,..,H_p^{sl}-1, \\ \beta_i^{sl} = 0, \quad \text{for } l=0, \\ \beta_i^{sl} = 1, \quad \text{for } l=1,..,H_p^{sl}-1, \end{cases} \tag{5.11}$$

with:

$$\underline{\Delta\hat{u}}_w^f(\tau+l|\tau) = \begin{bmatrix} \Delta\hat{u}_w^f(\tau+l|\tau) \\ \Delta\hat{u}_w^f(\tau+\frac{1}{m}+l|\tau) \\ \cdots \\ \Delta\hat{u}_w^f(\tau+\frac{m-1}{m}+l|\tau) \end{bmatrix}, \qquad \begin{array}{l} \text{for } \ w=1,..,N^f, \\ \qquad l=1,..,H_p^{sl}-1, \end{array}$$

$$\underline{I}_w^f = [\ \underbrace{I_{q_i^f}, I_{q_i^f}, ..., I_{q_i^f}}_{I_{q_i^f} \text{ repeated } m \text{ times}}\ ],$$

and where $I_{q_i^f}$ is the identity matrix of dimensions $q_i^f \times q_i^f$. The lifted matrix $\underline{\mathcal{D}}_{iw}^{sl-f}$ is defined as:

$$\underline{\mathcal{D}}_{iw}^{sl-f} = [\mathcal{D}_{iw}^{sl-f}(0), \mathcal{D}_{iw}^{sl-f}(1), ..., \mathcal{D}_{iw}^{sl-f}(m-1)]. \tag{5.12}$$

The elements $\mathcal{D}_{iw}^{sl-f}(\sigma)$ in equation (5.12), for $\sigma = 0, ..., m-1$, represent the effect of the input moves $\Delta\hat{u}_w^f$ from the fast subsystem w (for $w = 1, ..., N^f$) on the slow subsystems i (for $i = 1, .., N^{sl}$). The matrix $\mathcal{D}_{iw}^{sl-f}(\sigma)$ can be obtained by partitioning the matrix $\mathcal{D}^{sl-f}(\sigma)$ given in equation (5.6), for $\sigma = 0, ..., m-1$.

Typically, process industries need to deal with constraints on the process variables, input variables and input moves. The constraints on the variables of the subsystems can be the result of environmental and safety restrictions, product specifications and physical limitations in process equipment. The limitations on the slow subsystems' variables are expressed as inequality constraints in the optimization problem of each slow subsystem, as follows:

$$\begin{cases} x_{i\_min}^{sl} \leq \hat{x}_i^{sl}(\tau + l + 1|k) \leq x_{i\_max}^{sl}, \\ u_{i\_min}^{sl} \leq \hat{u}_i^{sl}(\tau + l|k) \leq x_{i\_max}^{sl}, \\ \Delta u_{i\_min}^{sl} \leq \Delta\hat{u}_i^{sl}(\tau + \nu|k) \leq \Delta u_{i\_max}^{sl}, \\ \text{for} \quad l = 0, ..., H_p^{sl} - 1, \quad \text{and} \quad \eta = 0, ..., H_u^{sl} - 1. \end{cases} \tag{5.13}$$

The optimal control problem of each slow subsystem encompasses the local process variables. Therefore, for each slow subsystem i (with $i = 1, ..., N^{sl}$), we propose the following optimization problem:

$$\min_{X_i^{sl}, \Delta U_i^{sl}, V_i^{sl}} \mathcal{J}_i^{sl} = \frac{1}{2}\Big(\big(X_{i\_sp}^{sl} - X_i^{sl}(\tau)\big)^T \mathcal{Q}_i^{sl} \big(X_{i\_sp}^{sl} - X_i^{sl}(\tau)\big) + \Delta U_i^{sl}(\tau)^T \mathcal{R}_i^{sl} \Delta U_i^{sl}(\tau) \Big) + \big(\Lambda_i^{sl}\big)^T \left[X_i^{sl}(\tau)^T, \Delta U_i^{sl}(\tau)^T, V_i^{sl}(\tau)^T\right]^T$$

$$\text{subject to :} \quad \begin{cases} \text{Equality constraints (5.9)} - \text{(5.10) and} \\ \text{Inequality constraints (5.13),} \end{cases} \tag{5.14}$$

where the vectors $X_i^{sl}(\tau)$, $\Delta U_i^{sl}(\tau)$ and $V_i^{sl}(\tau)$, for $i = 1, ..., N^{sl}$, are defined as:

$$\begin{aligned} X_i^{sl}(\tau) &= [\hat{x}_i^{sl}(\tau + 1|\tau)^T, \hat{x}_i^{sl}(\tau + 2|\tau)^T, ..., \hat{x}_i^{sl}(\tau + H_p^{sl}|\tau)^T]^T, \\ \Delta U_i^{sl}(\tau) &= [\Delta\hat{u}_i^{sl}(\tau|\tau)^T, \Delta\hat{u}_i^{sl}(\tau + 1|\tau)^T, ..., \Delta\hat{u}_i^{sl}(\tau + H_u^{sl} - 1|\tau)^T]^T, \\ V_i^{sl}(\tau) &= [\hat{v}_i^{sl}(\tau|\tau)^T, \hat{v}_i^{sl}(\tau + 1|\tau)^T, ..., \hat{v}_i^{sl}(\tau + H_p^{sl} - 1|\tau)^T]^T. \end{aligned}$$

In the objective function (5.14), the block-diagonal matrices $\mathcal{Q}_i^{sl}$ and $\mathcal{R}_i^{sl}$ are defined as:

$\mathcal{Q}_i^{sl} = diag(Q_i^{sl}(1), Q_i^{sl}(2), \ldots, Q_i^{sl}(H_p^{sl}))$ and $\mathcal{R}_i^{sl} = diag(R_i^{sl}(0), R_i^{sl}(1), \ldots, R_i^{sl}(H_u^{sl} - 1))$, where the weighting matrices $Q_i^{sl}(l+1)$ and $R_i^{sl}(\eta)$, for $l = 0, ..., H_p^{sl} - 1$ and $\eta = 0, ..., H_u^{sl} - 1$, are positive definite matrices of appropriate dimensions with their off-diagonal elements equal to zero. The vector Λ_i^{sl} is provided by the coordinator. The construction of the vector Λ^{sl} is shown in Section 5.4.4.

For simplicity, the optimal control problem (5.14) can be formulated in a compact form as:

$$\min_{Z_i^{sl}} \quad \mathcal{J}_i^{sl} = \frac{1}{2}\Big(Z_i^{sl}(\tau)^T \Upsilon_i^{sl} Z_i^{sl}(\tau)\Big) + \big(\phi_i^{sl}\big)^T Z_i^{sl}(\tau) + \big(\Lambda_i^{sl}\big)^T Z_i^{sl}(\tau) \tag{5.15}$$

subject to :

$$\begin{cases} G_i^{eq\_sl} Z_i^{sl}(\tau) = g_i^{eq\_sl}, \\ G_i^{ineq\_sl} Z_i^{sl}(\tau) \leq g_i^{ineq\_sl}. \end{cases} \tag{5.16}$$

The system of relations (5.16) can be obtained by arranging equations (5.9), (5.10) and inequalities (5.13) in a matrix form for the entire prediction horizon H_p^{sl} and control horizon H_u^{sl}. In (5.16), $G_i^{eq\_sl}$ is the coefficient matrix for the decision variables in the equality constraints (5.9)-(5.10) and, the vector $g_i^{eq\_sl}$ includes known data, such as the effect of current measured state variables and past input variables from the plant. The matrix $G_i^{ineq\_sl}$ is the coefficient matrix in the inequality constraints (5.13), and the vector $g_i^{ineq\_sl}$ represents the upper and lower limits for the decision variables in each slow subsystem i.

Fast-rate CDMPC controllers

Let us consider that the dynamics of the fast subsystems i (for $i = 1, ..., N^f$) are represented by model (5.8). The variables predicted by the CDMPC controller of each fast subsystem i along the prediction horizon H_p^f and control horizon H_u^f are given by:

$$\begin{cases} \hat{x}_i^f(k+l+1|k) = A_{ii}^f \hat{x}_i^f(k+l|k) + B_{ii}^f \hat{u}_i^f(k+l|k) + \hat{v}_i^f(k+l|k) + \\ \qquad\qquad \sum_{j \neq i} \big(\beta_i^f A_{ij}^f x_j^f(k) + B_{ij}^f u_j^f(k-1)\big) + \\ \qquad\qquad \sum_{w=1}^{N^{sl}} D_{iw}^{f-sl} u_w^{sl}(k-1), \qquad \text{for } l = 0, .., H_p^f - 1, \\ \hat{x}_i^f(k|k) = x_i^f(k), \\ \beta_i^f = 1, \quad \text{for } l = 0, \\ \beta_i^f = 0, \quad \text{for } l = 1, .., H_p^f - 1, \end{cases} \tag{5.17}$$

with:

$$\begin{cases} \hat{u}_i^f(k+l|k) = \sum_{\nu=0}^{l} \Delta\hat{u}_i^f(k+\nu|k) + u_i^f(k-1), & \text{for } l = 0, ..., H_p^f - 1, \\ \Delta\hat{u}_i^f(k+l|k) = 0, & H_u^f \leq l \leq H_p^f - 1. \end{cases} \tag{5.18}$$

In the proposed control scheme, $\hat{v}_i^f$ are decision variables in the CDMPC control problem of each fast subsystem i. The coordinator is formulated to ensure that the following equation (5.19) is satisfied for each fast subsystem i, for $i = 1, ..., N^f$:

$$\begin{cases} \Delta e_i^f(k+l|k) = 0 = \\ \hat{v}_i^f(k+l|k) - \sum_{j\neq i} \left(\beta_i^f A_{ij}^f \hat{x}_j^f(k+l|k) + B_{ij}^f \sum_{\nu=0}^{l} \Delta\hat{u}_j^f(k+\nu|k) \right) + \\ \qquad - D_{iw}^{f-sl} \sum_{\nu=0}^{l} \Delta\hat{u}_w^{sl}(\tau + \nu/m|\tau), \qquad \text{for } l = 0, ..., H_p^f - 1, \\ \beta_i^f = 0, \quad \text{for } l = 0, \\ \beta_i^f = 1, \quad \text{for } l = 1, .., H_p^f - 1. \end{cases} \tag{5.19}$$

It can be noted that the inputs $\hat{u}_w^{sl}$, for $w = 1, ..., N^{sl}$ are computed by the slow-rate CDMPC controllers and updated every control interval of length T^{sl}. A zero-order hold is assumed for the inputs $\hat{u}_w^{sl}$ within the slow control intervals; thus, if 'ν/m' in equation (5.19) is not an integer number, then $\Delta\hat{u}_w^{sl}(\tau + \nu/m|\tau) = 0$. In addition, the time counter τ in equation (5.19) is incremented every slow control interval of length T^{sl}.

As with the slow subsystems, there may exist some limitations on the fast subsystems' variables. These limitations are expressed as inequality constraints in the optimization problem of each fast subsystem, as follows:

$$\begin{cases} x_{i\_min}^f \leq \hat{x}_i^f(k+l+1|k) \leq x_{i\_max}^f, \\ u_{i\_min}^f \leq \hat{u}_i^f(k+l|k) \leq x_{i\_max}^f, \\ \Delta u_{i\_min}^f \leq \Delta\hat{u}_i^f(k+\eta|k) \leq \Delta u_{i\_max}^f, \\ \text{for} \quad l = 0, ..., H_p^f - 1, \quad \text{and} \quad \eta = 0, ..., H_u^f - 1. \end{cases} \tag{5.20}$$

The optimal control problem of each fast subsystem encompasses the local process variables. Therefore, for each fast subsystem i (with $i = 1, ..., N^f$), we propose the

following optimization problem:

$$\min_{X_i^f,\Delta U_i^f,V_i^f} \quad \mathcal{J}_i^f = \frac{1}{2}\Big(\big(X_{i\_sp}^f - X_i^f(k)\big)^T \mathcal{Q}_i^f \big(X_{i\_sp}^f - X_i^f(k)\big) + \Delta U_i^f(k)^T \mathcal{R}_i^f \Delta U_i^f(k)\Big) + \big(\Lambda_i^f\big)^T \left[X_i^f(k)^T, \Delta U_i^f(k)^T, V_i^f(k)^T\right]^T$$

subject to : (5.21)

$$\begin{cases} \text{Equality constraints (5.17)} - \text{(5.18) and} \\ \text{Inequality constraints (5.20),} \end{cases}$$

where the vectors $X_i^f(k)$, $\Delta U_i^f(k)$ and $V_i^f(k)$, for $i = 1, ..., N^f$, are defined as:

$$\begin{aligned} X_i^f(k) &= [\hat{x}_i^f(k+1|k)^T, \hat{x}_i^f(k+2|k)^T, ..., \hat{x}_i^f(k+H_p^f|k)^T]^T, \\ \Delta U_i^f(k) &= [\Delta\hat{u}_i^f(k|k)^T, \Delta\hat{u}_i^f(k+1|k)^T, ..., \Delta\hat{u}_i^f(k+H_u^f-1|k)^T]^T, \\ V_i^f(k) &= [\hat{v}_i^f(k|k)^T, \hat{v}_i^f(k+1|k)^T, ..., \hat{v}_i^f(k+H_p^f-1|k)^T]^T. \end{aligned}$$

In the objective function (5.21), the block-diagonal matrices $\mathcal{Q}_i^f$ and $\mathcal{R}_i^f$ are defined as: $\mathcal{Q}_i^f = diag(Q_i^f(1), Q_i^f(2), ..., Q_i^f(H_p^f))$ and $\mathcal{R}_i^f = diag(R_i^f(0), R_i^f(1), ..., R_i^f(H_u^f - 1))$, where the weighting matrices $Q_i^f(l+1)$ and $R_i^f(\eta)$, for $l = 0, ..., H_p^f - 1$ and $\eta = 0, ..., H_u^f - 1$, are positive definite matrices of appropriate dimensions with their off-diagonal elements equal to zero. The vector Λ_i^f is provided by the coordinator. The construction of the vector Λ^f is shown in Section 5.4.4.

For simplicity, the optimal control problem (5.21) can be formulated in a compact form as:

$$\min_{Z_i^f} \ \mathcal{J}_i^f = \frac{1}{2}\Big(Z_i^f(k)^T \Upsilon_i^f Z_i^f(k)\Big) + \big(\phi_i^f\big)^T Z_i^f(k) + \big(\Lambda_i^f\big)^T Z_i^f(k) \tag{5.22}$$

subject to :

$$\begin{cases} G_i^{eq\_f} Z_i^f(\tau) = g_i^{eq\_f}, \\ G_i^{ineq\_f} Z_i^f(\tau) \le g_i^{ineq\_f}. \end{cases} \tag{5.23}$$

System of relations (5.23) can be obtained by arranging equations (5.17), (5.18) and inequalities (5.20) in a matrix form for the entire prediction horizon H_p^f and control horizon H_u^f. In (5.23), $G_i^{eq\_f}$ is the coefficient matrix for the decision variables in the equality constraints (5.17)-(5.18) and, the vector $g_i^{eq\_f}$ includes known data, such as the effect of current measured state variables and past input variables from the plant.

The matrix $G_i^{ineq\text{-}f}$ is the coefficient matrix in the inequality constraints (5.20), and the vector $g_i^{ineq\text{-}f}$ represents the upper and lower limits for the decision variables in each fast subsystem i.

5.4.4 Price-Driven Coordination Algorithm

In this section, a price-driven coordination algorithm is presented to coordinate the dual-rate CDMPC controllers. The price vector is adjusted according to dual update modes that depend on the current control instant. The dual update modes are required in the price update algorithm because different sets of CDMPC controllers are coordinated at the different control instants. At control instants '$k = m\tau$', occurring every slow control interval, the CDMPC controllers from the fast subsystems and slow subsystems are executed. Thus, the coordinator updates the price vector to coordinate all the controllers in the plant. At control instants '$k \neq m\tau$', occurring every fast control interval, only the CDMPC controllers from the fast subsystems are executed. Therefore, the coordinator updates the price vector to coordinate only the fast-rate CDMPC controllers. For the problem formulation considered in this work, the price vector is updated according to the following dual-rate method:

$$p^{s+1} = p^s - \Xi_{mode}. \tag{5.24}$$

The superscripts 's' and '$s+1$' in (5.24) denote two consecutive iteration steps and the vector Ξ_{mode} is defined as:

$$\Xi_{mode} = \begin{cases} \alpha^{dual} \left(\mathbb{J}^{dual}\right)^{-1} \Delta E^s(k,\tau), & \text{for } k = m\tau, \quad (5.25a) \\ \alpha^{f} \left(\mathbb{J}^{f}\right)^{-1} \Delta E^s(k), & \text{for } k \neq m\tau. \quad (5.25b) \end{cases}$$

The task of the coordinator is to iteratively adjust the price p according to equation (5.24) until $\Delta E^s(k,\tau)$, for $k = m\tau$ (or $\Delta E^s(k)$, for $k \neq m\tau$) is driven to zero.
The positive parameters α^{dual} and α^{f} are the step sizes in the dual price update algorithm (5.24), where each step size can be calculated as described in Chapter 4.

The Jacobian matrices $\mathbb{J}^{dual}$ and $\mathbb{J}^{f}$ in (5.25a) and (5.25b) are defined as:

$$\mathbb{J}^{dual} = \sum_{i=1}^{N^f} \Theta_i^f \frac{dZ_i^{s^f}(k)}{dp^s} + \sum_{i=1}^{N^{sl}} \Theta_i^{sl} \frac{dZ_i^{s^{sl}}(\tau)}{dp^s}, \tag{5.26}$$

$$\mathbb{J}^{f} = \sum_{i=1}^{N^f} M^f \Theta_i^f \frac{dZ_i^{s^f}(k)}{dp^s}, \tag{5.27}$$

where the matrices Θ_i^f for $i = 1, ..., N^f$, and Θ_i^{sl} for $i = 1, ..., N^{sl}$ are given in Appendix 5.8.1 and the matrix M^f is defined as:

$$M^f = \Big[\; I \quad \underbrace{0}_{(H_p^f \cdot n^f)\times(H_p^{sl}\cdot n^{sl})} \; \Big]. \tag{5.28}$$

The interaction error vectors $\Delta E^s(k,\tau)$ and $\Delta E^s(k)$ in the price update algorithm are defined differently according to the control instants. For $k = m\tau$, $\Delta E^s(k,\tau)$ is defined as:

$$\Delta E^s(k,\tau) = \sum_{i=1}^{N^f} \Theta_i^f Z_i^{s^f}(k) + \sum_{i=1}^{N^{sl}} \Theta_i^{sl} Z_i^{s^{sl}}(\tau), \tag{5.29}$$

and for $k \neq m\tau$, $\Delta E^s(k)$ is defined as:

$$\Delta E^s(k) = \sum_{i=1}^{N^f} M^f \Theta_i^f Z_i^{s^f}(k). \tag{5.30}$$

In Chapter 4, the coordinator calculates the price vector p^s and sends that information to the local CDMPC controllers. To simplify the calculations performed by the local CDMPC controllers in the dual-rate distributed control scheme, the coordinator does not provide the price vector to each CDMPC controller, but it provides the vector Λ_i^f and Λ_i^{sl} to the fast-rate CDMPC controllers and the slow-rate CDMPC controllers, respectively. For the fast-rate CDMPC controllers, the vector Λ_i^f is computed by the coordinator as follows:

$$\left(\Lambda_i^f\right)^T = \begin{cases} \left(p^s\right)^T \Theta_i^f, & \text{for } k = m\tau \quad \text{and} \quad i = 1, ..., N^f, \quad (5.31a) \\ \left(p^s\right)^T M^f \Theta_i^f, & \text{for } k \neq m\tau \quad \text{and} \quad i = 1, ..., N^f. \quad (5.31b) \end{cases}$$

For the slow-rate CDMPC controllers, the vector Λ_i^{sl} is computed by the coordinator as follows:

$$\left(\Lambda_i^{sl}\right)^T = \left(p^s\right)^T \Theta_i^{sl}, \qquad \text{for } k = m\tau \quad \text{and} \quad i = 1, ..., N^{sl}. \tag{5.32}$$

In Section 5.4.5, the implementation of the dual-rate distributed controllers is described, including the steps of the coordination algorithm that take place within the different control intervals.

5.4.5 Implementation of Dual-Rate CDMPC Controllers

The implementation steps described in this section have been adapted from the implementation of the single-rate CDMPC controllers described in Chapter 4. The main difference between the implementation of the single-rate CDMPC controllers and the dual-rate CDMPC controllers is the dual mode of the coordination algorithm. In the dual coordination mode, the price vector is computed differently whether the fast-rate CDMPC controllers and the slow-rate CDMPC controllers are coordinated, or only the fast-rate CDMPC controllers are coordinated. The dual nature of the coordination algorithm is described below for control instants $k = m\tau$ and $k \neq m\tau$.

A. *Control instants $k = m\tau$:*

At control instants $k = m\tau$, the CDMPC controllers in the fast and the slow subsystems are executed. Therefore, all the CDMPC controllers in the plant need to be coordinated. At control instants $k = m\tau$, the coordination algorithm and the implementation of the dual-rate CDMPC controllers are summarized according to the following steps:

A.1 **Initialization (s=0):** The coordinator sets up an initial price vector p^s. For initialization, the elements of the vector p^s can be set to zero. The length of the price vector is equal to $H_p^f \cdot n^f + H_p^{sl} \cdot n^{sl}$. Next, the coordinator computes the vector Λ_i^f and Λ_i^{sl} according to equations (5.31a) and (5.32), and transfers the information to every local CDMPC controller.

A.2 **Optimization performed by each local CDMPC controller:** With the information provided by the coordinator, each CDMPC controller solves its own optimization problem. The slow-rate CDMPC controllers solve optimization problem (5.15)-(5.16) and the fast-rate CDMPC controllers solve optimization problem (5.22)-(5.23). The CDMPC controllers also calculate their sensitivity information as described in

Chapter 4. The calculated optimal solutions from the slow subsystems $Z_i^{sl}(\tau)$, for $i = 1, ..., N^{sl}$, and fast subsystems $Z_i^f(k)$, for $i = 1, ..., N^f$, as well as their sensitivity information are communicated to the coordinator.

A.3 **Price update:** The iteration counter 's' is incremented. The coordinator gathers the information from each CDMPC controller; it calculates $\Delta E^s(k, \tau)$ according to equation (5.29), and $\mathbb{J}^{dual}$ as given by equation (5.26). Next, the coordinator determines the step size α^{dual}, with $0 < \alpha^{dual} \leq 1$ for the dual-rate price update algorithm. Then, the coordinator updates the price vector p as per equation (5.24), with the vector Ξ_{mode} given by equation (5.25a), and it calculates the vectors Λ_i^f and Λ_i^{sl}, according to equations (5.31a) and (5.32). The vectors Λ_i^f and Λ_i^{sl}, are communicated to the corresponding CDMPC controllers.

A.4 **Iteration until convergence:** Steps (A.2)-(A.3) are repeated until the price-driven coordination algorithm is terminated. The price-driven coordination algorithm is terminated when $\|\Delta E^s(k, \tau)\| \leq \epsilon$, where ϵ is a specified error tolerance.

A.5 **Implementation of first calculated control action:** Once the price-driven coordination algorithm is terminated, the first calculated control moves are implemented in each subsystem, while the rest of the calculated control moves are discarded. Therefore, for the fast subsystems the first input moves $\Delta\hat{u}_w^f(\tau|\tau)$ in the lifted input vector are implemented, for $w = 1, ..., N^f$. In the slow subsystems, the input moves $\Delta\hat{u}_i^{sl}(\tau|\tau)$ are implemented, $i = 1, ..., N^{sl}$. Then, the iteration counter 's' in the coordination algorithm is reset to zero. The optimization problem (steps (A.1)-(A.4)) is solved every control instant $k = m\tau$.

B. *Control instants $k \neq m\tau$:*

At control instants $k \neq m\tau$, only the CDMPC controllers in the fast subsystems are executed and therefore coordinated. At instants $k \neq m\tau$, the coordination algorithm and the implementation of the fast-rate CDMPC controllers are

summarized according to the following steps:

B.1 **Initialization (s=0):** The coordinator sets up an initial price vector p^s. Here again, the elements of the vector p^s can be set to be zero, where the length of the price vector is equal to $H_p^f \cdot n^f$. Next, the coordinator computes the vector Λ_i^f for the fast subsystems according to equation (5.31b) and it transfers that information to every fast-rate CDMPC controller.

B.2 **Optimization performed by each fast-rate CDMPC controller:** Using the vector Λ_i^f provided by the coordinator, each fast-rate CDMPC controller solves its own optimization problem (5.22)-(5.23). The fast-rate CDMPC controllers also calculate their sensitivity information as described in Chapter 4. The calculated optimal solutions from the fast subsystems $Z_i^f(k)$, for $i = 1, ..., N^f$, as well as their sensitivity information are communicated to the coordinator.

B.3 **Price update:** The iteration counter 's' is incremented. The coordinator gathers the information from each fast-rate CDMPC controller; it calculates $\Delta E^s(k)$ according to equation (5.30), and $\mathbb{J}^f$ as given by equation (5.27). Next, the coordinator determines the step size α^f, with $0 < \alpha^f \leq 1$. Then, the coordinator updates the price vector p as per equation (5.24), with the vector Ξ_{mode} given by equation (5.25b), and it calculates the vector Λ_i^f for the fast subsystems, according to equation (5.31b). The vectors Λ_i^f is communicated to each fast-rate CDMPC controller.

B.4 **Iteration until convergence:** Steps (B.2)-(B.3) are repeated until the price-driven coordination algorithm is terminated. The price-driven coordination algorithm is terminated when $\|\Delta E^s(k)\| \leq \epsilon$.

B.5 **Implementation of first calculated control action:** Once the price-driven coordination algorithm is terminated, the first calculated control moves $\Delta\hat{u}_i^f(k|k)$ are implemented in each fast subsystem, while the rest of the calculated control moves are discarded. Then, the iteration counter 's'

in the coordination algorithm is reset to zero. The optimization problem (steps (B.1)-(B.4)) is solved every control instant $k \neq m\tau$.

5.5 Illustrative Example

In this section, an illustrative example is presented to demonstrate the performance of the dual-rate CDMPC controllers formulated with *strategy III*. For this illustrative example, two fast subsystems and two slow subsystems are considered. The model dynamics of the fast and the slow subsystems are given in Appendix 5.8.2. The sampling period for the fast and the slow subsystems are $T^f = 1\ [min]$ and $T^{sl} = 2\ [min]$, respectively. The parameters used in the simulation study are listed in Table 5.1.

Table 5.1: Parameters used in simulation study

Subsystems	Weighting matrices		Control and prediction horizons	
Fast subsystem 1	$Q_1^f(l+1) = diag(10,5,1,1,1)$, for $l = 0,...,H_p^f - 1$	$R_1^f(\eta) = diag(10,50,50,50,50)$, for $\eta = 0,...,H_u^f - 1$	$H_p^f = 30$	$H_u^f = 6$
Fast subsystem 2	$Q_2^f(l+1) = I_5$, for $l = 0,...,H_p^f - 1$	$R_2^f(\eta) = 50 \cdot I_5$, for $\eta = 0,...,H_u^f - 1$	$H_p^f = 30$	$H_u^f = 6$
Slow subsystem 1	$Q_1^{sl}(l+1) = diag(20,1,1,1,1)$, for $l = 0,...,H_p^{sl} - 1$	$R_1^{sl}(\eta) = diag(50,400,50,50,50)$, for $\eta = 0,...,H_u^{sl} - 1$	$H_p^{sl} = 30$	$H_u^{sl} = 6$
Slow subsystem 2	$Q_2^{sl}(l+1) = I_5$, for $l = 0,...,H_p^{sl} - 1$	$R_2^{sl}(\eta) = diag(400,50,50,50,50)$, for $\eta = 0,...,H_u^{sl} - 1$	$H_p^{sl} = 30$	$H_u^{sl} = 6$

The performance of the dual-rate CDMPC controllers was compared with the performance of four decentralized controllers and the performance of four CDMPC controllers that are coordinated by two single-rate coordinators (single-rate CDMPC control), as shown in Figure 5.10. It can be observed in Figure 5.10 that one of the coordinators operates at the fast rate to coordinate the CDMPC controllers for the fast subsystems, while the second coordinator operates at the slow rate to coordinate

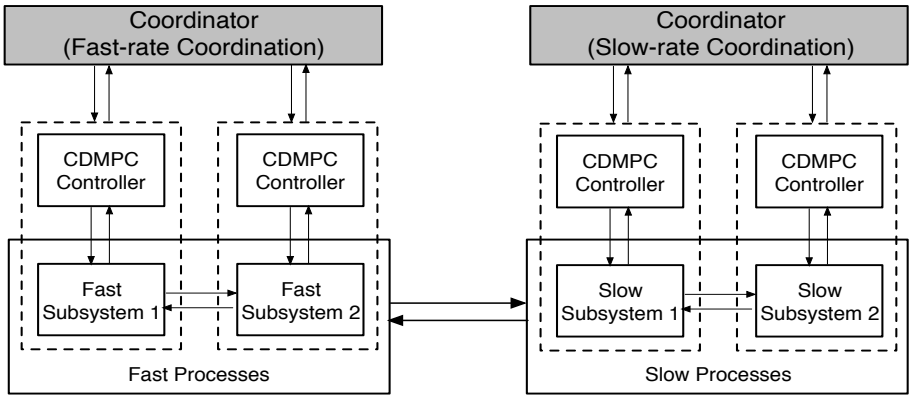

Figure 5.10: Hierarchy of CDMPC control scheme using two single-rate coordinators (single-rate CDMPC)

the CDMPC controllers for the slow subsystems.

A set-point change of magnitude 1.0 in the first state variable of the vector x_1^f and a set-point change of magnitude 0.5 in the first state variable of the vector x_1^{sl} were simulated at time $t = 0$, while the targets for the remaining state variables were kept at the origin. The results of the computer simulations are summarized in Table 5.2.

It can be observed in Table 5.2 that for the simulation studies carried out, the performance of the dual-rate CDMPC controllers using *strategy III* is superior to the performance of the decentralized controllers and also to the performance of the single-rate CDMPC controllers. The performance of the different control schemes in Table 5.2 are calculated as:

$$\begin{aligned}\mathcal{J}_{(.)} = \; & \frac{1}{2}\sum_{i=1}^{N^f}\sum_{k=0}^{t_f-1}\left(||x_{i\_sp}^f - x_i^f(k+1)||_{Q_i^f}^2 + ||\Delta u_i^f(k)||_{R_i^f}^2\right) + \\ & \frac{1}{2}\sum_{i=1}^{N^{sl}}\sum_{\tau=0}^{t_f-1}\left(||x_{i\_sp}^{sl} - x_i^{sl}(\tau+1)||_{Q_i^{sl}}^2 + ||\Delta u_i^{sl}(\tau)||_{R_i^{sl}}^2\right),\end{aligned} \tag{5.33}$$

where t_f is the simulation time. The performance improvement of the dual-rate CDMPC with respect to the other control schemes in Table 5.2 is given by the index Π [%].

In the dual-rate CDMPC, the predicted values for the interacting variables are taken into account in the local optimization problems. The interacting variables

Table 5.2: Control schemes applied to dual-rate subsystems: controllers' performance

Control schemes	$\mathcal{J}_{(\cdot)}$ calculated as in (5.33)	$\Pi = \frac{\lvert\mathcal{J}_{(\cdot)} - \mathcal{J}_{DR-CDMPC}\rvert}{\mathcal{J}_{DR-CDMPC}} \cdot 100\%$
Dual-rate CDMPC (*Strategy III*)	$\mathcal{J}_{DR-CDMPC} = 48.19$	—
Single-rate CDMPC (Figure 5.10)	$\mathcal{J}_{SR-CDMPC} = 50.83$	5.5
Decentralized MPC	$\mathcal{J}_{dec} = 85.16$	76.7

in the dual-rate CDMPC scheme include the effect of the interacting variables between subsystems optimized at the same rate and between subsystems optimized at different rates. In the decentralized control problems, the predicted values for all the interacting variables are neglected. Thus, a significant improvement of the dual-rate CDMPC over the decentralized controllers can be obtained. For this illustrative example, the performance improvement is approximately 76.7% (see index Π in Table 5.2).

In the single-rate CDMPC control scheme shown in Figure 5.10, the predicted values for the interacting variables between the fast subsystems and the slow subsystems are neglected. For the simulations performed in this illustrative example, the dual-rate CDMPC controllers outperformed the controllers in the single-rate CDMPC control scheme by approximately 5.5% (see index Π in Table 5.2). This value indicates that when the interactions between the fast subsystems and the slow subsystems are significant, considering these interactions in the subproblems helps improve the entire plant-wide performance.

5.6 Extension of Strategy III for Multi-Rate Coordination

The dual-rate coordination strategy proposed in *strategy III* can be extended for multi-rate coordination problems. It is assumed that there exist more than two sets

of controllers in the plant, where each set includes controllers that are executed at the same control rate. For example, assume there are three sets of controllers in the plant. In the first set, the controllers are executed every control interval of length T^{r1}. In the second set, the controllers are executed every control interval of length T^{r2}. In the third set, the controllers are executed every control interval of length T^{r3}. The superscripts '$r1$', '$r2$' and '$r3$' indicate different controller execution rates. Here again, at certain control instants some sets of controllers are synchronized as shown in Figure 5.11.

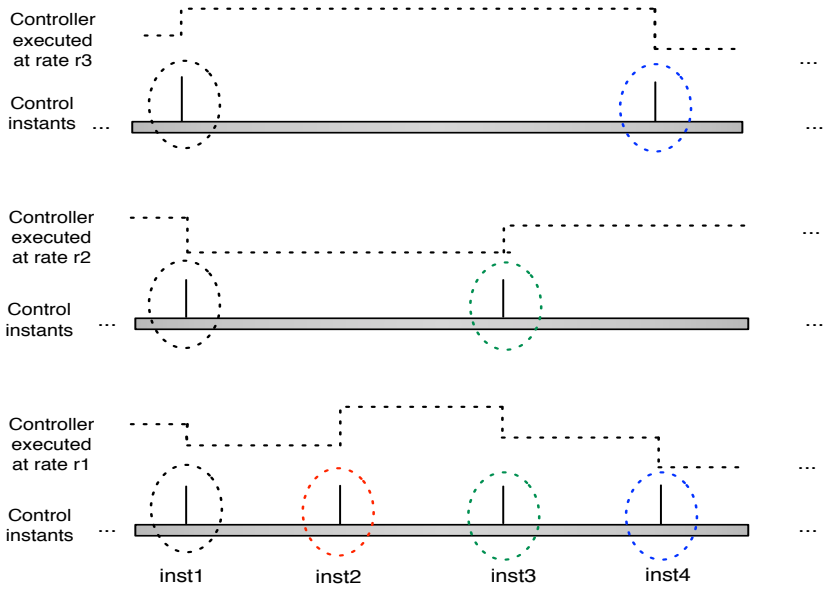

Figure 5.11: Multiple coordination mode for *strategy III*

In the multi-rate coordination, the price update algorithm has multiple coordination modes because at different control instants, the coordinator needs to coordinate a different set of controllers. According to Figure 5.11, at control instant 'inst1', the coordinator has to coordinate all the controllers in the plant; at control instant 'inst2', the coordinator has to coordinate the controllers that are executed at control rates '$r1$'; at control instant 'inst3', the coordinator has to coordinate the controllers that are executed at control rates '$r1$' and '$r2$' and at control instant 'inst3', the coordinator has to coordinate the controllers that are executed at control rates '$r1$' and '$r3$'.

One of the challenges of the proposed multi-rate coordination is the complexity of the coordination of multiple sets of controllers. There exists a trade-off between performance improvement and complexity of the coordinated-distributed control scheme. Therefore, the coordination strategy that is more suitable for the plant to be controlled, should be selected. For example, the options for coordination of multi-rate distributed controllers can include: a) designing a unique coordinator with multiple coordination modes for all the controllers in the plant, or b) designing coordinators with dual-rate coordination modes and a number of coordinators with single-rate modes, or c) any feasible combination between single-rate, dual-rate and multi-rate coordination. The possibilities for coordination increase with the number of controller execution rates, so one should opt for the coordination strategy that is more appropriate for the plant to be controlled.

5.7 Summary

In this chapter, coordination of dual-rate distributed controllers is investigated. This is a first attempt to solve multi-rate distributed control problems when a coordinator is used in the distributed control scheme.

Three strategies (*strategy I, II* and *III*) are discussed, where each strategy involved different coordination rates. *Strategy III*, which includes a dual-rate coordination mode, is studied in detail and tested using computer simulations. A case study is used to illustrate the effectiveness of the proposed dual-rate CDMPC control scheme. The performance of the dual-rate CDMPC controllers is compared to the performance of decentralized controllers and the performance of CDMPC controllers coordinated by two single-rate coordinators. Finally, an extension of *strategy III* for multi-rate coordination is presented. One of the challenges of multi-rate coordination is that the complexity of the coordination modes increases with the number of controller execution rates. In that case, the strategies for coordination of multi-rate distributed controllers can be adapted to include a combination of multi-rate and single-rate coordinators.

Another issue to investigate is the performance of the multi-rate CDMPC

controllers when the ratio between the fast and the slow sampling periods, m, increases. It is explained in Section 5.4.5 that at $k = m\tau$, the slow-rate CDMPC controllers and the fast-rate CDMPC controllers are coordinated. Then, the first calculated input moves from the slow and the fast subsystems are implemented. Within the slow control interval, the fast-rate CDMPC controllers are executed '$m-1$' more times. Therefore, the optimal input moves from the fast subsystems calculated at $k \neq m\tau$ (within the slow control interval) can be different from the optimal values calculated at $k = m\tau$[1]. As the ratio 'm' increases, the performance of the slow-rate CDMPC controllers is expected to worsen. The magnitude of the ratio 'm' represents an open research challenge. A further analysis on this issue is required to extend the current results to the design of multi-rate CDMPC control strategies.

5.8 Appendix

5.8.1 Matrices Required to Update the Price Vector

The matrices Θ_i^{sl} and Θ_i^{f} required in the price update mechanism and in the objective function of the subproblems are defined in this section. The matrix Θ_i^{sl} is the coefficient matrix for slow subsystem i, with $i = 1, ..., N^{sl}$. For example, for slow subsystems $i = 1$, $i = 2$ and $i = N$, Θ_i^{sl} becomes:

$$\Theta_1^{sl} = \begin{bmatrix} G_{11}^{f-sl} \\ \vdots \\ G_{N^f 1}^{f-sl} \\ G_{11}^{sl} \\ \vdots \\ G_{N^{sl} 1}^{s} \end{bmatrix}, \qquad \Theta_2^{sl} = \begin{bmatrix} G_{12}^{f-sl} \\ \vdots \\ G_{N^f 2}^{f-sl} \\ G_{12}^{sl} \\ \vdots \\ G_{N^{sl} 2}^{s} \end{bmatrix}, \qquad \Theta_{N^{sl}}^{sl} = \begin{bmatrix} G_{1N^{sl}}^{f-sl} \\ \vdots \\ G_{N^f N^{sl}}^{f-sl} \\ G_{1N^{sl}}^{sl} \\ \vdots \\ G_{N^{sl} N^{sl}}^{sl} \end{bmatrix}. \quad (5.34)$$

The matrix G_{ii}^{sl} in equation (5.34) is given by:

$$G_{ii}^{sl} = [0,\ 0,\ I], \qquad \text{for } \ i = 1, ..., , N^{sl}. \quad (5.35)$$

where the identity matrix, I, in equation (5.35) has dimensions $(H_p^{sl} \cdot n_i^{sl}) \times (H_p^{sl} \cdot n_i^{sl})$. The matrices G_{ji}^{sl} include the coefficients that represent the effect of the variables

[1]Only if infinite horizons and perfect models are considered in the optimization, the calculated input moves for the fast subsystems will be the same at subsequent optimizations.

$Z_i^{sl}(k)$ on the interacting variables $V_j^{sl}(k)$, for $j = 1, ..., , N^{sl}$ and $j \neq i$. The matrices G_{wi}^{f-sl} include the coefficients that represent the effect of the variables $Z_i^{sl}(k)$ on the interacting variables $V_w^f(k)$, for $w = 1, ..., N^f$.

Similarly, the matrix Θ_i^f is the coefficient matrix for fast subsystem i, with $i = 1, ..., N^f$. For example, for fast subsystems $i = 1$, $i = 2$ and $i = N^f$, Θ_i^f becomes:

$$\Theta_1^f = \begin{bmatrix} G_{11}^f \\ \vdots \\ G_{N^f 1}^f \\ G_{11}^{sl-f} \\ \vdots \\ G_{N^{sl}1}^{sl-f} \end{bmatrix}, \qquad \Theta_2^f = \begin{bmatrix} G_{12}^f \\ \vdots \\ G_{N^f 2}^f \\ G_{12}^{sl-f} \\ \vdots \\ G_{N^{sl}2}^{sl-f} \end{bmatrix}, \qquad \Theta_{N^f}^f = \begin{bmatrix} G_{1N^f}^f \\ \vdots \\ G_{N^f N^f}^f \\ G_{1N^f}^{sl-f} \\ \vdots \\ G_{N^{sl}N^f}^{sl-f} \end{bmatrix}. \quad (5.36)$$

The matrix G_{ii}^f in equation (5.36) is given by:

$$G_{ii}^f = [0,\ 0,\ I], \qquad \text{for } \ i = 1, ..., N^f. \quad (5.37)$$

where the identity matrix, I, in equation (5.37) has dimensions $(H_p^f \cdot n_i^f) \times (H_p^f \cdot n_i^f)$. The matrices G_{ji}^f include the coefficients that represent the effect of the variables $Z_i^f(k)$ on the interacting variables $V_j^f(k)$, for $j = 1, ..., N^f$ and $j \neq i$. The matrices G_{wi}^{sl-f} include the coefficients that represent the effect of the variables $Z_i^f(k)$ on the interacting variables $V_w^{sl}(k)$, for $w = 1, ..., N^{sl}$.

5.8.2 Models Used in the Illustrative Example

In this section, the process models for the illustrative example used in Section 5.5 are provided. The models for the slow and fast subsystems in the illustrative example are given in this section in continuous time. The continuous-time models are discretized and then partitioned into the corresponding subsystems to obtain the discrete-time models required for the formulation of the dual-rate CDMPC controllers.

Model dynamics for the slow subsystems:

The dynamics of the slow subsystems in continuous-time are given by equation (5.1)

with:

$$A_c^{sl} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & -1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 0 & 0 & 0 & 0.1 & 0.05 & 0 & 0 \\ 0 & 0 & 0.6 & -1.2 & 0 & 0 & 0 & -0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.35 & -1.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & -2.4 & 0 \\ 0 & 0 & -0.3 & -0.5 & 0 & 0 & 0 & 0 & 0 & -0.9000 \end{bmatrix},$$

$$B_c^{sl} = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0.65 & 0 & 0 & 0 & 0 & -0.25 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.2 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.3 & 1 & 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.75 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.5 \end{bmatrix}, \tag{5.38}$$

$$D_c^{sl-f} = \begin{bmatrix} -0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.005 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.002 & 0 & 0 & 0 & 0.005 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.015 \end{bmatrix},$$

where the vector of initial conditions $x^{sl}(0)$ for the slow subsystems is the zero vector of dimension $n^{sl} = 10$. The continuous-time model described by equations (5.38) is discretized as in Section 5.4.1. The discrete-time model for the slow subsystems is then partitioned into two slow subsystems: slow subsystem 1 and slow subsystem 2. The number of states and input variables in each slow subsystem are: $n_1^{sl} = 5$, $q_1^{sl} = 5$ for the slow subsystem 1; and $n_2^{sl} = 5$, $q_2^{sl} = 5$ for the slow subsystem 2. The slow subsystem 1 included the first five state variables of vector x^{sl} and the first five input variables of vector u^{sl}. The slow subsystem 2 included the last five state variables of vector x^{sl} and the last five input variables of vector u^f.

Model dynamics for the fast subsystems:

The dynamics of the fast subsystems in continuous-time are given by:

$$\dot{x}^f(t) = A_c^f x^f(t) + B_c^f u^f(t), \tag{5.39}$$

with:

$$A_c^f = \begin{bmatrix} -5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0.3 & 0 \\ 0 & -9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2.25 & 0 & 0 & 0 & 0 & -0.4 & 0 & 0 \\ 0 & 0 & 2.25 & -12.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.75 & -5.4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2.5 & 0 & 0 & 0 \\ -1.5 & 0 & 0 & 0 & 0 & 0 & -0.6 & -1.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -7.5 & 0 \\ 0 & 0.8 & 1.5 & 0 & 0 & 0 & 0 & 0 & 0 & -2.4 \end{bmatrix},$$

$$B_c^f = \begin{bmatrix} 0.8 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 1.6 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.6 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0.3 & -0.45 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 \end{bmatrix}, \quad (5.40)$$

where the vector of initial conditions $x^f(0)$ for the fast subsystems is the zero vector of dimension $n^f = 10$. The continuous-time model described by equations (5.39)-(5.40) is discretized at the fast sampling period T^f to give the discrete-time model for the fast subsystems. The discrete-time model is then partitioned into two fast subsystems: fast subsystem 1 and fast subsystem 2. The number of states and input variables in each fast subsystem are: $n_1^f = 5$, $q_1^f = 5$ for the fast subsystem 1; and $n_2^f = 5$, $q_2^f = 5$ for the fast subsystem 2. The fast subsystem 1 included the first five state variables of vector x^f and the first five input variables of vector u^f. The fast subsystem 2 included the last five state variables of vector x^f and the last five input variables of vector u^f.

6 Conclusions and Recommendations

6.1 Conclusions

In today's global and competitive market, process industries are looking for alternatives to increase their profits, while using the least amount of resources possible. Therefore, in order to compete, grow and satisfy environmental and safety restrictions, industrial companies seek for ways to improve their business operations by using, for example, new and advanced technologies and more efficient control strategies.

Distributed control has attracted increasing attention in recent years. It is considered a 'state of the art' control and it is seen as a promising strategy for control of large-scale systems. This thesis is concerned with the design of *coordinated-distributed* control schemes for optimal control of large-scale linear dynamic systems. This work is intended to make a contribution in the area of distributed control, which is currently one of the most active areas of process control. The coordinated-distributed control strategies proposed in this thesis benefit from using local controllers that can be coordinated to achieve optimal plant-wide performance. The three main advantages of the coordinated-distributed controllers developed in this thesis are:

- **guaranteed performance**. Convergence of the coordination algorithm allows the coordinated-distributed controllers to achieve the theoretical centralized optimal solution;

- **simplicity in the design of the coordinated-distributed controllers**. The coordinated-distributed controllers can be constructed with minor modifications to the existing decentralized controllers;

- **flexibility in the operation**. The coordinated-distributed control schemes can switch to the original decentralized control scheme, if needed. In addition, in case of equipment/communication failure or partial shutdowns, the coordinated-distributed controllers can be individually removed from the coordination loop and switched to the original decentralized control scheme.

The aforementioned properties of the coordinated-distributed controllers can be quite appealing to practitioners. A general practice in industry is to evaluate the potential return on investment of new technologies. The coordinated-distributed control schemes proposed in this thesis guarantee the centralized optimal performance, which is the "best" achievable performance for a given controller tuning. In addition, the coordinated-distributed control schemes proposed in this thesis do not require radical new configuration of the existing decentralized controllers. Thus, minor capital investment is required to implement the coordinated-distributed controllers.

The effectiveness of the coordinated-distributed control schemes and coordination algorithms are shown in this thesis using computer simulations for various illustrative examples, including benchmark processes obtained from the literature. Conclusions drawn from this research and recommendations for future work are provided below.

In Chapter 2, a state-feedback coordinated-distributed linear quadratic (CDLQ) control scheme is developed for large-scale continuous-time, linear dynamic systems. The local CDLQ controllers proposed in this chapter are coordinated using the prediction-driven method. Significant insight into coordination of local distributed controllers is gained from the development of the CDLQ control scheme and the analysis of the prediction-driven algorithm. Convergence properties of the prediction-driven coordination algorithm are studied. A trade-off between convergence of the

prediction-driven coordination algorithm and offset-free performance is identified and a remedy to tackle this issue is given. This remedy consists of including a proximal term in the objective function of each subproblem. The effect of the proximal term on the speed of convergence of the prediction-driven coordination algorithm is investigated. Furthermore, a method for tuning the parameter γ in the proximal term is proposed. The coordination principles developed in Chapter 2 served as the foundation for the design of the CDMPC controllers in Chapter 3.

In Chapter 3, a state-feedback coordinated-distributed model predictive control (CDMPC) scheme is proposed for unconstrained linear systems whose dynamics are represented by discrete-time state-space models. A prediction-driven algorithm is used for coordination of the local CDMPC controllers. Convergence properties of the prediction-driven coordination algorithm are shown. The proposed prediction-driven algorithm is shown to improve monotonically with the number of iterations until convergence to the centralized optimal solution is achieved. A stability analysis for the closed-loop system under CDMPC control is also provided. To the best of our knowledge, this is the first work that proves stability of the closed-loop system when a coordinator communicates with local CDMPC controllers.

Constraints handling is one of the desired features of the standard MPC controllers. Therefore, constraints are incorporated in the CDMPC control scheme developed in Chapter 4. In this chapter, the CDMPC controllers are developed using state-space models as well as finite step-response models. This is another convenient feature of the proposed CDMPC controllers since many commercial MPC products still use step-response models in their formulation. The price-driven method is used to coordinate the local CDMPC controllers. Newton's method and a sensitivity analysis technique are used to update the price in the price-driven method. A discussion of the price-driven algorithm performance is also given, followed by a study on the stability of the closed-loop system under CDMPC control.

It can be noted that both Chapter 3 and Chapter 4 deal with coordination of local CDMPC controllers. Nevertheless, different coordination methods are employed in these two chapters. By comparing the performance of both coordination methods, it can be observed that:

- The price-driven method in Chapter 4 exhibits fast convergence. In particular, when there is no change in the active set of the subproblems, the price-driven coordination algorithm converges within two iterations. As opposed to the price-driven method, the coordinator in the prediction-driven method predicts the state variables in the entire system in order to update the price vector. As a result, more iterations of information exchange between the coordinator and the local CDMPC controllers may be required until convergence of the prediction-driven algorithm is achieved.

- The full price vector in the prediction-driven method, λ^s_{coor}, is calculated as an approximation of the Lagrange multiplier vector of the centralized optimization problem, λ^*. The price vector in the price-driven method, p^s, is updated using Newton's method. When both coordination algorithms converge, their corresponding price vector is equivalent to the Lagrange multiplier vector of the centralized optimization problem used as performance benchmark. That is, in the prediction-driven method, λ^s_{coor} becomes λ^* and in the price-driven method, p^s becomes λ^*.

Finally, Chapter 5 addresses the problem of coordination of distributed MPC controllers that are executed at different control rates. The work proposed in Chapter 5 is a first attempt to solve multi-rate distributed control problems where a coordinator is involved in the distributed control scheme. In particular, three strategies are discussed to deal with coordination of local CDMPC controllers that are executed at two different control rates (dual-rate CDMPC controllers). A coordination algorithm for dual-rate CDMPC controllers is proposed and tested through computer simulations. The results obtained from the simulation experiments show that the proposed dual-rate CDMPC controllers outperform the existing decentralized controllers significantly. A method for extending the dual-rate CDMPC control scheme to multi-rate CDMPC control scheme is proposed towards the end of the chapter.

6.2 Recommendations for Future Work

A number of challenges remain in the development and application of coordinated-distributed control schemes. Some directions for future research work are outlined below:

- In most of the coordinated-distributed control schemes proposed in this thesis, it is assumed that full state measurements are available at every sampling instant[1]. It is often the case in process industry that some process variables cannot be measured on-line on frequent sampling instants (for example, as in the case of some product concentrations). It is even possible that no measurements are available at all for some process variables. In such situations, an observer should be constructed (if possible) to estimate the unmeasured variables (Ogunnaike and Ray, 1994; Goodwin *et al.*, 2001).

- In Chapter 3, it is assumed that the predicted variables $Z^0_{coor}(k)$ and full price vector λ^0_{coor} are set to zero for initialization of the prediction-driven coordination algorithm within each control interval (see Section 3.4.4). In order to reduce the number of iterations within each control interval, different possibilities for initialization of the prediction-driven algorithm can be studied. For example, one alternative for initialization of the prediction-driven algorithm is to use the control input moves calculated by the CDMPC controllers at the previous sampling instant. If the CDMPC calculations do not change significantly from one control interval to the next one, then a reduction in the number of iterations in the prediction-driven algorithm is expected.

- A complexity study for both the price-driven method and the prediction-driven method is recommended. A complexity study for the price-driven method is presented in (Cheng, 2007) for algebraic optimization problems. A similar complexity study can be performed for the dynamic optimization problems considered in this thesis in Chapter 4.

[1] In Chapter 4, a CDMPC control scheme is developed using input-output models obtained from step-test data. Thus, an observer is not required in this case.

With respect to the prediction-driven coordination method, it is shown in Chapter 2 and Chapter 3 that when the prediction-driven coordination algorithm converges, the solutions obtained with the coordinated-distributed controllers are equal to the centralized (or plant-wide) optimal solution. Convergence to the centralized solution is shown regardless of the decomposition strategy (and as long as a convergence condition is satisfied). Nevertheless, different decomposition strategies can have an impact on the speed of convergence of the prediction-driven coordination algorithm. It is recommended to perform a complexity study that includes evaluating the speed of convergence of the prediction-driven algorithm when it is used for:

- subsystems that have similar dimensions (such as similar number of states and inputs in each subsystem across the plant) vs. subsystems that have different dimensions across the plant;
- plants with small number of subsystems vs. plants with a considerably larger number of subsystems;
- subsystems with different number of interacting variables;
- subsystems with interactions as a result of recycle streams vs subsystems only unidirectional interactions as a result of serial operations.

To summarize, the structure and dimension of the subsystems and the interactions can be exploited to provide some recommendations that aim at reducing the speed of convergence of the prediction-driven coordination algorithm.

- In Chapter 5, a dual-rate coordinated-distributed control scheme is proposed. A method for extending the dual-rate CDMPC control scheme to multi-rate control scheme is also provided. Multi-rate coordinated-distributed control is an open research topic with great potential for future development. One of the assumptions of the dual-rate CDMPC control scheme proposed in Chapter 5 is the fact that all the local CDMPC controllers are synchronized across the plant by their computer clock. A multi-rate coordination algorithm to coordinate

local controllers that are executed asynchronously or with information delays is highly desirable.

Another assumption of the dual-rate CDMPC control scheme developed in Chapter 5 is the fact that subsystems controlled at different rates are coupled through the control input variables. That is, the control inputs from the slow subsystems affect the fast subsystems, and vice versa[2]. The control scheme proposed in Chapter 5 can be extended to include fast and slow subsystems coupled through the state variables, as well. In that case, the unavailable data (e.g., measurements of the slow subsystems' states at the fast sampling instants) needs to be estimated.

Finally, as described in Chapter 5, the positive integer 'm' represents the ratio between the slow sampling period T^{sl} and the fast sampling period T^{f} in the dual-rate CDMPC control scheme. As the ratio 'm' increases, the performance of the dual-rate CDMPC controllers is expected to degrade. Investigation of the effect of 'm' on the performance of the dual-rate CDMPC controllers should be undertaken. An analysis on this issue can benefit the design of the dual-rate CDMPC control strategies.

[2]In Chapter 5, only subsystems controlled at different rates are considered to be coupled through the control input variables. Subsystems controlled at the same rate are considered to be coupled through the control input variables as well as the state variables.

Bibliography

Anderson, B. D. O. and J. B. Moore (1971). *Linear Optimal Control*. Prentice Hall, Inc.. New Jersey.

Androulakis, I. P. and G. V. Reklaitis (1999). Approaches to asynchronous decentralized decision making. *Computers and Chemical Engineering* **23**, 341–355.

Aske, E. M. B., S. Strand and S. Skogestad (2008). Coordinator mpc for maximizing plant throughput. *Computers and Chemical Engineering* **32**, 195–204.

Bellman, R.E. (1957). *Dynamic Programming*. Princeton University Press. Princeton, New Jersey.

Camacho, E. F. and C. Bordons (1999). *Model Predictive Control*. Springer. Berlin.

Camponogara, E. and S. N. Talukdar (2007). Distributed model predictive control: Synchronous and asynchronous computation. *IEEE Transactions on Systems, Man. and Cybernetics - Part A: Systems and Humans* **37**(5), 732–745.

Camponogara, E., D. Jia, B. H. Krogh and S. Talukdar (2002). Distributed model predictive control. *IEEE Control Systems Magazine* pp. 44–52.

Chen, G. and M. Teboulle (1994). A proximal-based decomposition method for convex minimization problems. *Mathematical Programming* **64**, 81–101.

Chen, T. and B. Francis (1995). *Optimal Sampled-Data Control Systems*. Springer. London.

Cheng, R. (2007). Decomposition and Coordination of Large-Scale Operations Optimization. PhD thesis. University of Alberta. Edmonton, Canada.

Cheng, R., J. F. Forbes and W. S. Yip (2007). Price-driven coordination method for solving plant-wide mpc problems. *Journal of Process Control* **17**, 429–438.

Cheng, R., J. F. Forbes and W. S. Yip (2008). Dantzig-wolfe decomposition and plant-wide mpc coordination. *Computers and Chemical Engineering* **32**, 1507–1522.

Cohen, G. (1977). On an algorithm of decentralized optimal control. *Journal of Mathematical Analysis and Applications* **59**(2), 249–259.

Dallagi, A., N. I. Marcos and J. F. Forbes (2008). Coordination of decentralized large-scale process optimal control problems. *Proceedings of 17th World Congress of The International Federation of Automatic Control (IFAC)* pp. 9958–9963.

Dantzig, G. B and P. Wolfe (1960). Decomposition principle for linear programs. *Operations Research* **8**(1), 101–111.

Dunbar, W. B. and R. M. Murray (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica* **42**, 549–558.

Fisher, D. G. and D. E. Seborg (1976). *Multivariable computer control: A case study*. American Elsevier Publishing Company Inc.. New York.

Garcia, C. E., D. M. Prett and M. Morari (1989). Model predictive control: Theory and practice - a survey. *Automatica* **25**(3), 335–348.

Goodwin, G. C., S. F. Graebe and M. E. Salgado (2001). *Control System Design*. Prentice Hall. New Jersey.

Grosdidier, P., A. Mason, A. Aitolahti, P Heinonen and V. Vanhamäki (1993). Fcc unit reactor-regenerator control. *Computers and Chemical Engineering* **17**(2), 165–179.

Guay, M. (1996). Measurement of Nonlinearity in Chemical Process Control. PhD thesis. Queen's University. Kingston, Ontario, Canada.

http://www.ict-hd-mpc.eu/ (2011).

Jamshidi, M. (1983). *Large-scale systems: modeling and control*. North-Holland series in system science and engineering. New York.

Jia, D. and B. Krogh (2002). Min-max feedback model predictive control for distributed control with communication. In: *Proceedings of the American Control Conference*. pp. 4507–4512.

Jose, R. A. and L. H. Ungar (1998). Auction-driven coordination for plantwide optimization. In: *Proceedings of Foundations of Computer-Aided Process Operation (FOCAPO)*.

Jose, R. A. and L. H. Ungar (2000). Pricing interprocess streams using slack auctions. *AIChE Journal* **46**(3), 575–587.

Katebi, M. R. and M. A. Johnson (1997). Predictive control design for large-scale systems. *Automatica* **33**(3), 421–425.

Larsson, T. and S. Skogestad (2000). Plantwide control - a review and a new design procedure. *Modeling, Identification and Control* **21**(4), 209–240.

Lewis, F. L. and V. L. Syrmos (1995). *Optimal Control*. John Wiley & Sons, Inc.. New York.

Li, S., Y. Zhang and Q. Zhu (2005). Nash-optimization enhanced distributed model predictive control applied to the shell benchmark problem. *Information Sciences* **170**, 329–349.

Liu, J., D. Muñoz de la Peña and P.D. Christofides (2009). Distributed model predictive control of nonlinear process systems. *AIChE Journal* **55**(5), 1171–1184.

Liu, J., D. Muñoz de la Peña and P.D. Christofides (2010). Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements. *Automatica* **46**, 52–61.

Lu, J. Z. (2003). Challenging control problems and emerging technologies in enterprise optimization. *Control Engineering Practice* **11**, 847–858.

Maciejowski, J. M. (2002). *Predictive Control with Constraints*. Prentice Hall. Essex, England.

Maestre, J. M., D. Muñoz de la Peña, E. F. Camacho and T. Alamo (2011). Distributed model predictive control based on agent negotiation. *Journal of Process Control* **21**, 685–697.

Marcos, N. I., J. F. Forbes and M. Guay (2009). Coordination of distributed model predictive controllers for constrained dynamic processes. In: *International Symposium on Advanced Control of Chemical Processes (ADCHEM)*. Istanbul, Turkey, July 12-15, 2009. pp. 138–143.

Marcos, N. I., R. Cheng, J. F. Forbes and M. Guay (2008). Distributed model predictive control for large-scale dynamic systems. In: *Proceedings of the International Symposium on Advanced Control of Industrial Processes (ADCONIP)*. Jasper, Canada, May 4-6, 2008.

Mehyar, M., D. Spanos, J. Pongsajapan, S. H. Low and R. M. Murray (2005). Distributed averaging on asynchronous communication networks. *44th IEEE Conference on Decision and Control*.

Mesarovic, M. D., D. Macko and Y. Takahara (1970). *Theory of Hierarchical, Multilevel, Systems*. Academic Press. New York.

Morosan, P., R Bourdais, Dumur D. and J. Buisson (2011). A distributed mpc strategy based on benders' decomposition applied to multi-source multi-zone temperature regulation. *Journal of Process Control* **21**, 729–737.

Naidu, D. S. (2003). *Optimal Control Systems*. CRC Press. London.

Nedić, A. and A. Ozdaglar (2007). On the rate of convergence of distributed subgradient methods for multi-agent optimization. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. New Orleans, USA. pp. 4711–4716.

Negenborn, R.R., B. De Schutter and J. Hellendoorn (2008). Multi-agent model predictive control for transportation networks: serial versus parallel schemes. *Engineering Applications of Artificial Intelligence* **21**, 353–366.

Newell, R. B. and P. L. Lee (1989). *Applied process control: A case study*. Prentice Hall. Australia.

Ogunnaike, B. A. and W. H. Ray (1994). *Process Dynamics, Modeling and Control*. Oxford University Press. New York.

Qin, S. J. and T. A. Badgwell (2003). A survey of industrial model predictive control technology. *Control Engineering Practice* **11**, 733–764.

Rawlings, J. B. and B. T. Stewart (2008). Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control* **18**, 839–845.

Ray, W. H. (1981). *Advanced Process Control*. McGraw-Hill, Inc.

Rockafellar, R. T. (1976). Augmented lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research* **1**(2), 97–116.

Rossiter, J. A. (2005). *Model-based predictive control: A practical approach*. CRC Press. Florida.

Rossiter, J. A., J. Sheng, T. Chen and S. L. Shah (2005). Interpretations of and options in dual-rate predictive control. *Journal of Process Control* **15**, 135–148.

Samyudia, Y., P. L. Lee and I. T. Cameron (1994). A methodology for multi-unit control design. *Chemical Engineering Science* **49**(23), 3871–3882.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control - a review. *Journal of Process Control* **19**, 723–731.

Seborg, D. E. (1989). *Process dynamics and control*. Wiley. New York.

Sheng, J., T. Chen and S. L. Shah (2001). On stability robustness of dual-rate generalized predictive control systems. *Proceedings of the American Control Conference* pp. 3415–3420.

Sheng, J., T. Chen and S. L. Shah (2002). Generalized predictive control for non-uniformly sampled systems. *Journal of Process Control* **12**, 875–885.

Stewart, B. T., A. N. Venkat, J. B. Rawlings, S. J. Wright and G. Pannocchia (2010). Cooperative distributed model predictive control. *Systems & Control Letters* **59**, 460–469.

Sun, Y. and N. H. El-Farra (2008). Quasi-decentralized model-based networked control of process systems. *Computers and Chemical Engineering* **32**, 2016–2029.

Talukdar, S. and E. Camponogara (2001). Network control as distributed, dynamic game. In: *Proceedings of the 34th International Conference on System Sciences*.

Venkat, A. N. (2006). Distributed Model Predictive Control: Theory and Applications. PhD thesis. University of Wisconsin Madison.

Venkat, A. N., J. B. Rawlings and S. J. Wright (2005). Stability and optimality of distributed model predictive control. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. pp. 6680–6685.

Venkat, A. N., J. B. Rawlings and S. J. Wright (2006). Implementable distributed model predictive control with guaranteed performance properties. In: *Proceedings of the 2006 American Control Conference*. pp. 613–618.

Walid, A., H. Budman and A. Elkamel (2010). Selection of control structure for distributed model predictive control in the presence of model errors. *Journal of Process Control* **20**, 270–284.

Wolbert, D., X. Joulia, B. Koehret and L. T. Biegler (1994). Flowsheet optimization and optimal sensitivity analysis using analytical derivatives. *Computers and Chemical Engineering* **18**(11/12), 1083–1095.

Zafra-Cabeza, A., J. M. Maestre, Ridao M. A., E. F. Camacho and L. Sanchez (2011). A hierarchical distributed model predictive control approach to irrigation canals: A risk mitigation perspective. *Journal of Process Control* **21**, 787–799.