Price-Driven Coordination of Distributed Model Predictive Controllers: A Bi-Level Optimization Approach

by

Bardia Hassanzadeh

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

Doctor of Philosophy

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

Abstract

Chemical and petrochemical plants typically integrate a number of geographically distributed operating units, which are physically linked through energy and material streams or inherently coupled via plant-wide constraints. The main drawback of the current decentralized control system is that it fails to consider the interrelations between subsystems, which could usually result in poor performance or even loss of closed-loop stability. Such concerns have motivated various control strategies to tackle these problems. One possibility is to replace the whole network with a centralized control structure. Despite the potential benefits, this renovation would require significant capital cost, increase maintenance costs, and reduce fault tolerance. Another practical approach is a distributed control that aims to improve the performance of a currently installed decentralized network. Distributed model predictive control (DMPC) methods are divide into two general categories: non-coordinated and coordinated schemes. Coordinated DMPC (CDMPC) networks, which consist of distributed controllers and a coordinator, are able to attain an overall optimal solution over a wide range of conditions. The focus of this thesis is to develop on-line strategies for CDMPC systems and overcome existing issues with global convergence and stability of closed-loop systems, under price-driven CDMPC concept. In particular, the main contributions are developing two novel information flow mechanisms for CDMPC of nonlinear systems and proposing a new solution method for CDMPC of linear systems, via a bi-level optimization framework.

Preface

The Materials presented in the current thesis are part of the research project under supervision of Dr. J. Fraser Forbes and Dr. Jinfeng Liu, which has been funded by Natural Sciences and Engineering Research Council (NSERC) of Canada.

Chapter 2 of this work is in preparation for submission to the *International Journal of Control*.

Chapter 3 of this work is in preparation for submission to the journal of *Computers & Chemical Engineering*.

Chapter 4 of this work is in preparation for submission to the *Journal of Process Control*.

It should be mentioned that the format of this thesis is paper-based and there is some repetition, in the materials and methods sections of each chapter.

To my family and my grandmother, for their love, encouragement and support.

Acknowledgements

I am extremely grateful to my supervisors, Dr. J. Fraser Forbes and Dr. Jinfeng Liu for their appreciable inspiration, support and patience throughout my PhD studies. Their invaluable kindness, guidance and comments allowed me to develop my character and knowledge. They have been great supervisors and friends. It was a great pleasure to have the opportunity to work under their supervision.

I would especially like to thank Mr. Hallas Pakravesh, Dr. Padideh Mohseni, Ms. Shuning Li, Dr. Natalia Marcos, Dr. Javad Abdollahi, Dr. Sepideh Kasiri, and Dr. Farnaz Mani for their great help and encouragement.

Last but not least, I would like to gratefully acknowledge the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support.

Contents

List of Tables

List of Figures

List of Symbols

α	an estimate of closed-loop sub-optimality degree, chapter 3
α	step size, chapters 2,4
$\bar{\mathcal{L}}$	Lagrange function of transformed single-layer problem, chapter 4
$\bar{g}$	discretized set of nonlinear equations in the DAE system, chapter 3
$\bar{P}_{LQR}$	weighting matrix of terminal cost of the centralized LQR, chapter 4
$\bar{Z}$	concatenated vector of optimization variables of local MPCs, chapter 4
Δ	trust region criterion, chapter 3
ℓ	number of integration stages for the IRK method, chapter 3
ϵ	a predefined tolerance, chapter 2
ϵ_0	a predefined tolerance, chapter 3
η	perturbation parameter
Γ	coefficient matrix of manipulated input variables in state-space system, chapter 2
Γ	manipulated input variables coefficient matrix (discrete-time system), chapter 4
Γ_c	manipulated input variables coefficient matrix (continuous-time system), chapter 4
$\hat{u}$	concatenated vector of manipulated inputs inside controller
$\hat{v}$	predicted value of interacting variables
$\hat{x}$	concatenated vector of states inside controller

$\hat{x}_a$ concatenated vector of auxiliary state variables inside local MPCs, chapter 3

λ Lagrange multiplier of the inequality constraints

λ Lagrange multiplier of the centralized LQR at time k, chapter 4

λ^{eq} Lagrange multiplier of equality constraints in the lower-level problem, chapter 3

λ^{ineq} Lagrange multiplier of inequality constraints in the lower-level problem, chapter 3

λ_{LP} Lagrange multipliers associated with inequality constraints of the LP problem, chapter 4

$\mathcal{C}$ convex set of constraints in local MPCs, chapter 4

$\mathcal{C}$ convex set of constraints over manipulated input variables, chapter 2

$\mathcal{E}$ constraint violation of the penalty problem, chapter 3

$\mathcal{F}_{LP}$ Lagrange dual function of the LP problem, chapter 4

$\mathcal{H}$ Hessian matrix of $\mathcal{L}$, chapter 2

$\mathcal{J}$ Jacobian of $\mathcal{L}$, chapter 2

$\mathcal{J}_G$ index set of inequality constraints in the lower-level problem, chapter 4

$\mathcal{J}_H$ index set of equality constraints in the lower-level problem, chapter 4

$\mathcal{K}$ IRK internal integrator variable, chapter 3

$\mathcal{L}$ Lagrange function of the lower-level problem, chapters 3,4

$\mathcal{L}$ overall Lagrange function of local MPCs, chapter 2

$\mathcal{L}_{LP}$ Lagrange function of the LP problem, chapter 4

$\mathcal{L}_{QP}$ Lagrange function of the QP problem, chapter 4

$\mathcal{M}$ merit function, chapter 2

$\mathcal{S}$ set of feasible solutions of the lower-level problem, chapter 4

μ barrier weighting function, chapter 2

μ Lagrange multiplier of the exact penalty function, chapter 3

ν Lagrange multiplier associated with equality constraint, chapter 4

ν_{LP} Lagrange multipliers associated with equality constraints of the LP problem, chapter 4

ν_{QP} Lagrange multipliers associated with equality constraints of the QP problem, chapter 4

Ω logarithmic barrier function, chapter 2

$\Omega(p)$ feasible solution set of the lower-level problem for a fixed value of p, chapter 3

Φ coefficient matrix of local optimization variables inside MPCs, chapter 2

Φ exact penalty function, chapter 3

ϕ concatenated vector function of discretized ODE system, chapter 3

Π CHKS smoothing function

Ψ coefficient matrix of optimization variables inside local NMPCs, chapter 3

Ψ coefficient matrix of state variables in state-space system, chapter 2

Ψ state variables coefficient matrix (discrete-time system), chapter 4

Ψ_c state variables coefficient matrix (continuous-time system), chapter 4

τ IRK integration step-size, chapter 3

Θ coefficient matrix for the interacting constraints

$\tilde{\Phi}$ approximation of Φ, chapter 3

Υ weighting matrix of local optimization variables

ϖ_ρ Region of attraction, defined by a level set of Lyapunov function with magnitude ρ, chapter 2

ϑ Lagrange multiplier of the equality constraints, chapter 2

A coefficient matrix of equality constraints, chapter 4

a IRK internal parameter, chapter 3

A^{ineq} coefficient matrix of linear inequality constrains, chapter 4

b IRK internal parameter, chapter 3

b RHS of equality constraints, chapter 4

b^{ineq} RHS of linear inequality constraints, chapter 4

c IRK internal parameter, chapter 3

C^{eq} concatenated vector function of equality constraints in the transformed problem, chapter 3

C^{ineq} concatenated vector function of inequality constraints in the transformed problem, chapter 3

d feasible direction of price vector, chapter 4

E concatenated vector of error between the predicted and captured interaction, chapter 2

e concatenated vector of error between the predicted and plant-wide interaction

F objective function of the upper-level problem, chapter 4

f ODE nonlinear function in the DAE system, chapter 3

f ODE nonlinear function, chapter 2

f objective function of the lower-level problem, chapter 4

F^A approximate discretization of f, chapter 2

F^E exact discretization of f, chapter 2

G concatenated vector function of discretized DAE system, chapter 3

G concatenated vector function of discretized ODEs in the DAE system, chapter 3

G concatenated vector function of inequality constraints, chapter 4

g nonlinear set of equations in the DAE system, chapter 3

g vector of convex functions that contains mixed constraints, chapter 4

G^{eq} concatenated vector function of equality constraints in the lower-level problem, chapter 3

G^{eq} concatenated vector function of equality constraints, chapter 2

g^{eq} RHS of equality constraints, chapter 2

G^{ineq} concatenated vector function of inequality constraints in the lower-level problem, chapter 3

G^{ineq} concatenated vector function of inequality constraints, chapter 2

g^{ineq} RHS of inequality constraints, chapter 2

G_{act} vector set of active inequality constraints, chapter 4

H concatenated vector function of equality constraints, chapter 4

h coefficient matrix of plant-wide interaction, chapter 3

h integration step for the discrete-time approximation, chapter 2

H_p length of prediction horizon, chapter 2

H_u length of control horizon, chapter 2

H_{LQR} Hamiltonian of the centralized LQR, chapter 4

J objective function of a monolithic MPC problem, chapter 2

J_C objective function of the centralized problem, chapter 2

J_D objective function of CDNMPC, chapter 3

J_P objective function of plant-wide MPC

J_{DC} objective function of the decentralized problem

J_D objective function of CDMPC, chapters 2,4

J_{LQR} objective function of the centralized LQR, chapter 4

k current sampling time

K_{LQR} state feedback gain of the centralized LQR, chapter 4

l^C stage cost of the centralized problem

l^D stage cost of CDMPC, chapter 4

l^D stage cost of CDNMPC problem, chapter 3

lb concatenated vector of lower bounds, chapter 3

lb_u lower bound of manipulated input variables, chapter 2

M a real positive-definite matrix, chapter 4

m number of sub-systems

N length of prediction (control) horizon

n a typical sampled time

n_u length of manipulated input variable vector, chapter 2

n_x length of state variable vector, chapter 2

N_{LQR} finite time of the centralized LQR, chapter 4

P weighting matrix of terminal cost, chapter 2

p price vector

P_{LQR} Ricatti variable of the centralized LQR, chapter 4

Q weighting matrix of state variables

Q_{LQR} weighting matrix of state variables, chapter 4

R weighting matrix of manipulated input variables

R_{LQR} weighting matrix of manipulated input variables, chapter 4

s iteration number, chapters 2,4

s slack variable in the lower-level problem, chapter 3

T sampling period, chapter 2

t time

t_k current sampling time, chapter 2

T_s sampling time, chapter 4

U concatenated vector of calculated manipulated input variables

u concatenated vector of manipulated input variables

U_{sp} concatenated vector of set-points for manipulated input variables, chapter 3

ub concatenated vector of upper bounds, chapter 3

ub_u upper bound of manipulated input variables, chapter 2

V concatenated vector of predicted values of interacting variables

v concatenated vector of interaction variables, chapter 3

V^C Lyapunov function of centralized MPC, chapter 4

V^C Lyapunov function of centralized NMPC, chapter 3

V^D Lyapunov function of CDMPC, chapter 4

V^D Lyapunov function of CDNMPC problem, chapter 3

V_h Lyapunov function, chapter 2

V_{LQR} Lyapunov funtion of the centralized LQR, chapter 4

W positive semi-definite part of F, chapter 3

w feasible direction of lower-level problem optimization variables, chapter 4

w slack variable, chapter 3

X concatenated vector of predicted state trajectories

x concatenated vector of state variables

x_0 initial value for state variables, chapter 3

X_a concatenated vector of predicted auxiliary state trajectories, chapter 3

x_a concatenated vector of auxiliary state variables, chapter 3

X_{sp} concatenated vector of set-points for state variables, chapter 3

X_{sp} concatenated vector of set-points, chapter 4

Y a real non-singular matrix, chapter 4

Y concatenated vector of optimization variables, chapter 3

Z concatenated vector of local optimization variables (excluding interacting variables), chapter 4

Z concatenated vector of local optimization variables inside MPCs, chapter 2

Z concatenated vector of optimization variables inside local NMPCs, chapter 3

Chapter 1

Introduction

Chemical and petrochemical plants integrate a number of operating units that are connected via material and energy flows, and they are usually geographically separated. These relations can be interpreted as information flow or dynamical constraints between subsystems. Regulations, process safety and environmental responsibilities, together with expectations for profitability, have given rise to more complexities in process design and control. Such complexities may intensify the interactions between subsystems by adding new process units, energy recovery operations or material recycle streams. Thus, optimal plant-wide strategies should be deployed to achieve improved performance, while ensuring safe and responsible plant operation. Three common strategies are decentralized, centralized and distributed. Among these, distributed control has received considerable attention, since it takes advantage of decentralized control's flexibility and has the potential to attain centralized control performance. In the context of model predictive control (MPC), the distributed strategy that pertains to this work is coordinated distributed MPC (CDMPC). This powerful approach is capable of achieving maximum plant-wide performance, by taking into account interrelations between subsystems with a small modification applied to the current decentralized network that adds a coordination level to subsystems. To date, a number of research studies has been dedicated to CDMPC [19, 66, 76, 57]; however, a detailed study on CDMPC of nonlinear systems

with global convergence proofs and closed-loop stability properties had not been proposed prior to this study. In addition, this research focuses on improving the currently developed CDMPC schemes for linear systems and suggests a new solution strategy to ensure global convergence and stability.

To clarify terms and definitions, centralized, decentralized and distributed networks are briefly described; next, the discussion is narrowed down to CDMPC and the challenges that define scope of this thesis.

Decentralized Control Network

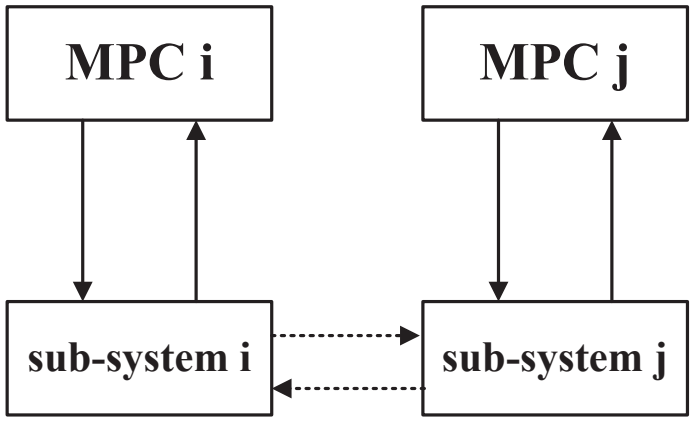


Figure 1.1: Decentralized MPC network architecture

A decentralized structure is the most common approach for control of many large-scale applications. A typical plant with two subsystems under decentralized control is presented in Figure 1.1. As shown in Figure 1.1, a separate controller is assigned to each subsystem, and decisions are made based on local information gathered. In this scheme, each controller can be designed and tuned independently. Note that in decentralized control, the effect of interactions between subsystems is ignored. As a results, a sub-optimal solution to the plant-wide problem is usually obtained. This could result in

losing closed-loop stability [100, 96, 93, 23]. In general, there are three solutions to these problems: 1) switching to a centralized scheme; or 2) modifying the current network and applying distributed approaches; and 3) redesigning of the decentralized system to ensure stability, typically at a performance cost.

Centralized Control Network

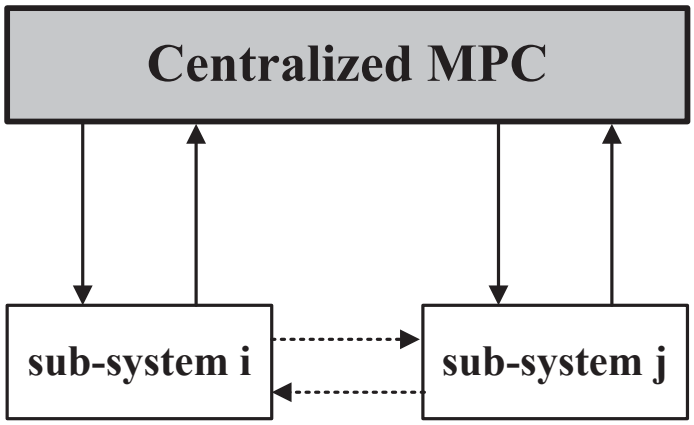


Figure 1.2: Centralized MPC network architecture

In a centralized control network, effects of interactions between subsystems are considered in the computation of control actions. An illustration of this scheme is shown in Figure 1.2, where a monolithic MPC controls a plant composed of two subsystems. In theory, the optimal solution obtained by this schemes reflects the maximum achievable plant-wide performance. Implementation of centralized structure requires that the currently installed decentralized controllers be decommissioned and a centralized controller is implemented, complete with appropriate information flows from the operating units to the control center. Despite the potential benefits, switching to centralized scheme requires substantial investment of time and money. Any benefits may be neglected by increased maintenance costs, loss of fault tolerance and diminished resilience. Consider

that centralized controllers must solve a large-scale problem, which demands sophisticated optimization algorithms and high computational power. Thus, the capital cost of a centralized control system can be substantial. Furthermore, any small change in a local unit, due to repairs, or during emergency shutdown incidents, would result in recasting of the monolithic controller.

Distributed Control Network

The technical issues mentioned in decentralized and centralized schemes can be overcome, to a great extent, by distributed control techniques. The basic idea is to take advantage of flexibility in the decentralized design and attain a performance improvement that approaches the centralized counterpart. Two different schools of thought exist in distributed control design , which can be differentiated in terms of information exchange between controllers and the approach that is taken to reaching a consensus on the control actions to be taken. These predictive control strategies can be divided into two categories: non-coordinated and coordinated distributed MPC.

Non-coordinated Distributed Control Structure

In the context of non-coordinated distributed MPC, local controllers include interaction models and exchange information to achieve an improved performance. A typical representation of a non-coordinated DMPC is depicted in Figure 1.3. Within this scope, a classification can be made based on communication design of network: fully-connected networks where all controllers exchange information, and partially-connected networks where only a limited number of controllers send and receive information. Further, if this transmission is limited to one time per sampling time the algorithm is called non-iterative [15, 31]; and, when it is transmitted more than once per sampling time the

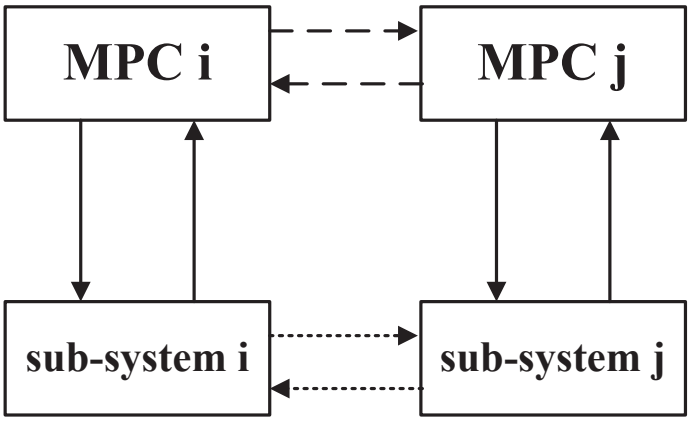


Figure 1.3: Non-coordinated distributed MPC network architecture

algorithm is called iterative [58, 69].

Among iterative approaches, cooperative distributed MPC schemes have enjoyed more interest by researches. Unlike non-cooperative iterative approaches that tend to find a Nash equilibrium [15, 58], cooperative schemes tend to reach a Pareto optimal [101, 102]. The very first versions of cooperative MPC was proposed by [100], mainly for networked linear MPCs with convex subproblems. In [100], and later in [93], convergence and closed-loop stability for premature algorithm termination were studied for the proposed methods. These ideas were extended to non-convex problems with classes of nonlinearity within the prediction models [95]; however, global convergence to centralized optimal solution was not guaranteed.

Application of Lyapunov MPC in cooperative DMPC of nonlinear systems was investigated by [60]. One immediate property of such networks is that local controllers are responsible for ensuring the closed-loop stability; nevertheless, achieving the global optimal solution of the centralized performance was still an issue due to the non-convexity of subproblems. Another cooperative DMPC scheme was proposed by [90], based on sensitivity information exchange, as an attempt to achieve centralized performance of non-

linear systems using successive linear approximations of the subproblems. Nonetheless, feasibility of the derived solutions and stability of the closed-loop network has remained an open question. An enthusiastic reader is referred to [88, 23] , and references therein, for a comprehensive review of research developed in this area.

Coordinated Distributed Control Structure

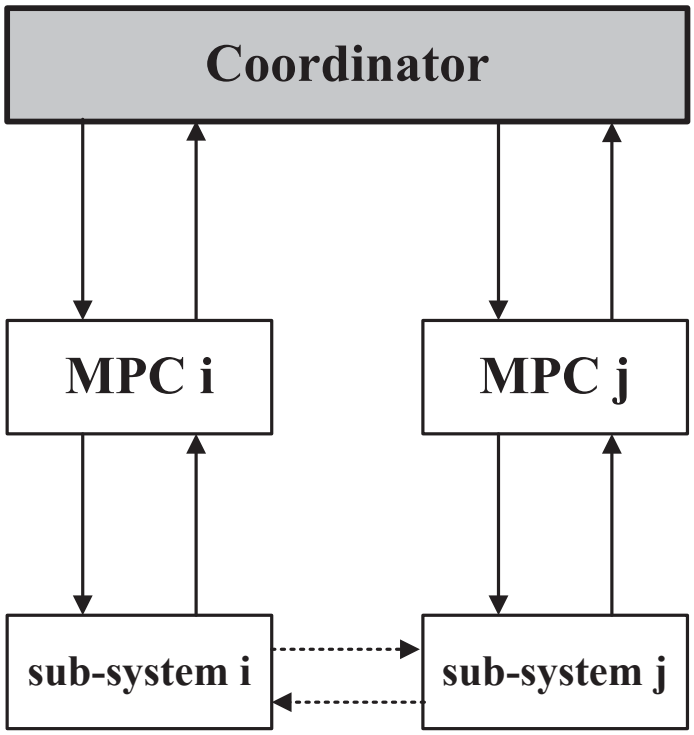


Figure 1.4: Coordinated distributed MPC network architecture

An alternative approach, as shown in Figure 1.4 to the schemes outlined in previous section is coordinated distributed control. In fact, such schemes belong to a general class of two-level (hierarchical) control systems, in which the upper level coordinates decisions making process of local controllers at the lower-level. The general idea, behind

two-level MPC control, is that each local controller, in the network, solves a suitable optimization problem and constantly communicates with the coordinator to satisfy plant-wide constraints. This iterative procedure continues until the error of prediction stays within a predefined limit [88]. An extensive study on mathematical theory of hierarchical systems may be found in [71].

Specifically, in CDMPC, the main concept is to achieve the centralized MPC performance and guarantee closed-loop stability properties of any intermediate solution produced by the algorithm. The first feasibility study of applying Dantzig-Wolfe decomposition on coordination of MPC networks was proposed by [20, 19]. Accordingly, [21] developed an efficient price-adjustment algorithm for steady-state target calculation of distributed MPC systems based on Newton's method. In this iterative procedure, sensitivity analysis and active-set identification techniques were used as a means of communication between local controllers and the coordination level to achieve the centralized performance. This method was extended by [67, 66] to plant-wide coordination of single-rate and dual-rate linear MPC systems. Also, a state feedback scheme for prediction driven coordination of linear-quadratic control systems, based on interaction prediction principle [71], was proposed by [66].

Convergence of iterative price-driven and prediction-driven CDMPC algorithms for time invariant linear systems was studied by [76]. Based on work of [81], a pseudo-model coordination scheme was proposed by [76] for linearly constrained decentralized MPC systems. In addition, application of price-driven, prediction driven and pseudo-model CDMPC schemes to chance-constrained systems for dealing with uncertain plant-wide disturbances in prediction models was studied by [76].

The first attempt at coordination of nonlinear dynamical systems using the pseudo-model CDMPC was developed by [76] . The performance was an identical match to the centralized MPC derived by linearizing the nonlinear model around a nominal operating

point. In a parallel research, computational and convergence properties of iterative price-driven CDMPC and prediction-driven CDMPC algorithms for linear systems were studied by [57]. This analysis was made for dynamical systems subject to equality constraints dictated by interaction models and predictions models. The study of such properties for systems subject to inequality constraints remained as an open area of research.

1.1 Research Scope

To date, the existing body of research on various CDMPC algorithms [20, 19, 21, 67, 66, 76, 57] has been generally devoted to linearly constrained dynamical systems. Although a pseudo-model coordination scheme [76] was proposed for handling non-linear systems, the performance was compared to a centralized MPC based on linear time invariant appropriate model around the desired operating condition. Thus, in order to capture the maximum plant-wide performance, nonlinearity of plant model has to be considered within the structure of CDMPC. For price-driven CDMPC of linear systems, Newton-based nested approach has been taken to solve the optimization problem, which could result in poor convergence properties [76].

This work contributes to developing a systematic method of coordination for nonlinear interconnected systems. In particular, two novel coordination structures are proposed to improve performance of an existing decentralized predictive control system for nonlinear systems. Moreover, two new solution strategies are proposed to design the CDMPC structure for linear dynamical systems, to compensate for performance loss in Newton-based algorithms. This thesis uses a bi-level optimization approach to CDMPC network, based on the concept of price-driven coordination. Mathematically, the coordination strategies in this research, are based on dual decomposition of the plant-wide predictive control problem into a proper separable structure. Starting from a current decentralized

network, the necessary modification to local controllers are applied. Then, the coordination level and the information flow in the network is implemented. The required modification can be summarized as an appropriate relaxation of interaction equations into the objective function of local controllers. Also, the coordinator is synthesized to establish a proper plant-wide decision making process to compensate for interaction model violations inside local prediction models.

In this study, it is assumed that local MPCs are single-rate synchronous controllers, and all required measurements are available at the selected rate. The performance of proposed CDMPC approaches are evaluated against the corresponding centralized MPC benchmark and the existing decentralized MPC network. For the purpose of this work, all prediction models are derived via proper discretization methods. The computed receding horizon actions are applied to the continuous-time plant. Essential conditions to ensure global convergence of the proposed algorithms as well as closed-loop stability of the system under the CDMPC schemes are investigated. This thesis is a collection of three candidate journal submissions, where the first two are dedicated to nonlinear systems and the last paper is devoted to linear dynamical systems.

Chapter 2 proposed a nested bilevel optimization strategy to design a price-driven CDMPC scheme for a nonlinear plant, which is operated by a decentralized control system. The prediction model is derived through successive linearization of continuous-time plant model around the current operating condition, at each sampling time. This paper proposes an interior-point approach that develops a CDMPC structure that stabilizes open-loop stable nonlinear systems. In fact, this extension to [20, 19, 21, 67, 66, 76, 57] eliminates the need to identify the correct set of active constraints at the end of each coordination cycle. In the proposed CDMPC scheme, the coordinator receives information from the distributed subsystem MPCs and sends prices to the distributed controllers to coordinate their actions in an iterative fashion to achieve the performance and stability

of the corresponding centralized MPC benchmark. Another highlight of this chapter is the investigation of sufficient conditions to ensure convergence and stability of the proposed algorithm. In particular, the scenario when the algorithm is stopped prior to convergence to the desired optimal solution is studied, and the closed-loop stability is guaranteed under certain conditions.

In Chapter 3, the problem of designing a CDMPC for an existing network of decentralized nonlinear model predictive controllers (NMPC) is addressed, which extends the ideas introduced in [89]. The nonlinear plant is assumed to be governed by dynamic algebraic equations (DAEs) with separable interconnections in the form of equality constraints. Despite linear systems, a limited body of research is dedicated to coordinated distributed NMPC (CDNMPC); and, the existing solution strategies are not suitable for on-line purposes. To date, the solution methods [89] have either been based on steepest ascent approaches or off-line dynamic optimization procedures, which could lead to poor convergence or demand high computational resources. Another approach to coordination of nonlinear MPC systems was proposed by [68], based on the difference between the total shared resources available and demanded by the local NMPC controllers. In this design, the coordination process is handled by manipulating the prices of every common resource by a controller with the aim of satisfying the plant-wide constraints in the shared resources. The main shortcomings of this approach is that global convergence to an overall optimal solution and stability of closed-loop system is not ensured, plus the scope is limited to binding constraints pertaining to shared resources. To overcome these issues, a novel structure of information flow between the coordinator and distributed NMPCs is proposed, via bi-level nonlinear optimization. This multi-level structure consists of the coordinator lying in the upper level and the distributed NMPC controllers lying in the lower level. Under certain preconditions, the bi-level optimization problem is transformed into a relaxed single level problem, and the resulting optimization is solved via

a trust-region technique. Previous designs of coordinated MPC or NMPC [89, 76] were based on finding a fixed value for the prediction horizon of the local predictive controllers, which were prone to unstable closed-loop behavior. This chapter illustrates a method to stabilize plant dynamics with open-loop unstable behavior by using an adaptive posteriori approach to determine the minimum required prediction horizon for local controllers. In addition, the proposed CDNMPC is guaranteed to be globally convergent to a unique optimal plant-wide trajectory, when a set of conditions are satisfied.

Two novel algorithms to coordinate distributed model predictive control (MPC) systems are developed in Chapter 4. Similar to Chapter 3, bi-level optimization techniques are deployed to solve the price-driven CDMPC problem. The mechanisms are implemented for two distinctive scenarios based on the presence of inequality constraint in the local MPCs. The first scheme is an analytic approach to find the closed-form solution of plant-wide problem, when all the constraints inside the structure of local controllers remain active all the time. The second algorithm is an iterative approach to find a local optimal solution to the general CDMPC problem via method of feasible directions. Unlike nested price-driven CDMPC approaches [20, 19, 21, 67, 66, 76, 57] , the proposed CDMPC schemes in this Chapter are globally convergent and are capable of stabilizing unstable open-loop dynamics, when a set predefined conditions hold in the network. Similar to Chapter 3, an on-line approach is used to ensure stability of overall system. This dynamic optimization approach enables the controlled network to stabilize unstable open-loop dynamical systems without any modifications applied to the native CDMPC formulation.

Finally, Chapter 5 summarizes and concludes the material covered in thesis. In addition, some research challenges in the design and development of coordinated distributed control systems are suggested.

1.2 Terms and Definitions

This section is intended to provide clarification and proper definitions for some of the common terms used throughout this thesis.

Lipschitz Function: [79] A continuous function $f : \mathbb{R}^q \to \mathbb{R}^r$ is said to be Lipschitz on some set $\mathcal{N} \subset \mathbb{R}^q$, if there exists a strictly positive constant L such that:

$$\big|\big|f(x_1) - f(x_2)\big|\big| \leq L||x_1 - x_2|| \tag{1.1}$$

Exact Discretization: Discretization is a procedure of obtaining a discrete-time model from a continuous-time system, under a suitable sample and hold scheme. Exact discretization obtains models that are exact and match the analytical solution of the continuous-time system. Consider a continuous-time LTI system, governed by the following differential equation:

$$\frac{dy(t)}{dt} + \frac{1}{\tau} = \frac{K}{\tau}u(t) \tag{1.2}$$

where: τ is the time constant of the first-order system; y denotes the output variable; K denotes the gain; and u denotes the input variable of the system. Here, it is assumed that u is held constant within the time period: $T := [k\Delta t, (k+1)\Delta t]$, for $k \geq 0$.

Performing the Laplace transformation on (1.2) within the time period: $0 \leq t \leq \Delta t$, the following is derived:

$$sY(s) - Y(0) = -\frac{1}{\tau}Y(s) + \frac{K}{\tau s}u(0) \tag{1.3a}$$

using the inverse Laplace transform, the analytic form of solution is obtained as:

$$y(t) = \exp(-\frac{t}{\tau})y(0) + K\big(1 - \exp(-\frac{t}{\tau})\big)u(0) \tag{1.3b}$$

in other words:

$$\begin{aligned} y(\Delta t) =&\ \exp(-\frac{\Delta t}{\tau})y(0) + K\big(1 - \exp(-\frac{\Delta t}{\tau})\big)u(0) && (1.3c)\\ y(2\Delta t) =&\ \exp(-\frac{\Delta t}{\tau})y(\Delta t) + K\big(1 - \exp(-\frac{\Delta t}{\tau})\big)u(\Delta t) && (1.3d)\\ &\vdots && (1.3e)\\ y(n\Delta t) =&\ \exp(-\frac{\Delta t}{\tau})y((n-1)\Delta t) + K\big(1 - \exp(-\frac{\Delta t}{\tau})\big)u((n-1)\Delta t) && (1.3f) \end{aligned}$$

Note that, this discretization is exact and the corresponding closed-form can be written as:

$$y_k = \exp(-\frac{\Delta t}{\tau})y_{k-1} + K\big(1 - \exp(-\frac{\Delta t}{\tau})\big)u_{k-1} \tag{1.4}$$

Global Convergence: Consider an optimization algorithm $\mathcal{A}$ on set $\mathcal{Z}$, which is initialized from an arbitrary point z_0, generates an infinite sequence $\{z_s\}_{s=0}^{\infty}$. If a solution set $\mathcal{S}$ and a real-valued function $\mathcal{D}(z)$ exists for the algorithm such that:

$$\begin{aligned} &(a)\ \text{all } z_s \text{ are contained in a compact subset of } \mathcal{Z}\\ &(b)\begin{cases} i)\text{if } z_s \notin \mathcal{S}, \text{ then } \mathcal{D}(z_{s+1}) < \mathcal{D}(z_s)\\ ii)\text{if } z_s \in \mathcal{S}, \text{ then } \mathcal{D}(z_{s+1}) \leq \mathcal{D}(z_s) \end{cases}\\ &(c)\ \text{the mapping of } \mathcal{A} \text{ is closed at all points outside } \mathcal{S} \end{aligned}$$

then the limit of any convergent subsequence of $\{z_s\}_{s=0}^{\infty}$ is a solution point [7]. Further-

more, if $\mathcal{A}$ is uniformly compact, closed and strictly monotone on $\mathcal{Z}$, any subsequence generated by $\mathcal{A}$ is convergent [73]. In other words, global convergence refers to a property of an optimization algorithm that ensures converging to a unique optimal solution of a problem from arbitrary starting points.

Plant-wide Optimal Solution: The term plant-wide control refers to controlling the plant by a network of controllers or a monolithic centralized controller. Similarly, optimal plant-wide solution refers to optimal performance of a centralized controller. MPC stands for Model Predictive Control/Controller. The term CD(N)MPC stands for coordinated distributed (nonlinear) MPC. Throughout this thesis, the optimal solution to the centralized problem (or the CD(N)MPC problem) is used interchangeably with the local optimum solution to the plant-wide (N)MPC problem, which satisfies a set of predefined criteria and feasibility conditions.

Chapter 2

Distributed Model Predictive Control of Nonlinear Systems Based on Price-Driven Coordination

Here a nonlinear plant is considered, which is operated by a decentralized control system. This existing control system ignores the interactions between sub-systems which often results in uncaptured plant-wide performance. The focus of this chapter is on the design of a distributed model predictive control (DMPC) network to stabilize plants governed by nonlinear dynamics, using successively linearized internal models. In such a design, interactions between the sub-systems should be considered in order to exceed the performance of the current decentralized DMPC scheme. In particular, a coordinated DMPC (CDMPC) schemes is proposed in which a coordination layer is added to the existing network, while minor modifications are applied to the local MPC controllers. In this chapter, a novel interior-point approach to design price-driven CDMPC structures is presented for stabilization of open-loop stable nonlinear systems. In the proposed CDMPC scheme, the coordinator receives information from the distributed subsystem MPCs and sends prices to the distributed controllers to coordinate their actions in an iterative fashion to achieve the performance and stability of a hypothetical centralized MPC for the entire plant. Sufficient conditions ensuring convergence and stability of the

proposed CDMPC are derived and the situation when the algorithm is stopped before convergence is investigated. The performance of the proposed approach is illustrated with a chemical process example.

2.1 Introduction

Large-scale processes are common occurrences in modern chemical and petrochemical industries for increased process efficiency. Typically, a large-scale process is composed of tightly integrated processes that interact via material and energy flows. Traditionally, centralized and decentralized control are the two primary frameworks for the control of large-scale systems. While the decentralized control is easy to implement, it may lead to degradation of plant-wide performance or even loss of closed-loop stability since in decentralized control the interactions between subsystems are in general neglected (see, for example, [86, 8, 96] and references therein). On the other hand, the centralized control is expected to give the best performance; however, it may become too complicated to implement as the size of the control problem grows. Moreover, a centralized control for a large-scale process may require substantial maintenance effort and is not favorable from a fault tolerance point of view [63, 77].

Motivated by the above observations, in recent years, significant efforts have been devoted to the development of distributed model predictive control (DMPC) for the control of large-scale systems. In a DMPC scheme, distributed local model predictive controllers (MPC) communicate and exchange information with each other or a coordinator to coordinate their actions; please see [15, 84, 88, 23, 22] for reviews of results in this area. As a strategy between decentralized and centralized control frameworks, DMPC is able to achieve improved plant-wide performance while preserving the flexibility of decentralized control such as low maintenance costs, unique fault tolerance properties and resilience

in design and operation. One classification divides DMPC schemes into two categories: non-coordinated DMPC and coordinated DMPC.

In the context of non-coordinated distributed MPC, local controllers include interaction models and exchange information to achieve an improved performance. Within this scope, a classification can be made based on communication design of network: fully-connected networks where all controllers exchange information, and partially-connected networks where only a limited number of controllers send and receive information. Some important recent work in this area includes cooperative DMPC of linear systems [84, 94], Lyapunov-based sequential and iterative DMPC of nonlinear systems [61, 60], robust DMPC of linear systems [3], DMPC of linear systems coupled through the inputs [64] and dissipativity-based DMPC of linear systems [98].

In contrast, inside a Coordinated DMPC (CDMPC) scheme, distributed MPCs communicate with a coordinator to achieve improved performance. Different methods have been developed for CDMPC including the price-driven approach (i.e., [20, 21, 67, 76, 57, 68]), the primal decomposition approach [35, 25], the prediction-driven approach (i.e., [66, 76]), and the pseudo-Model Coodination approach [76]. The main differences between these coordination methods appear in the ways that the interactions between subsystems are addressed [76]. Pertaining to nonlinear systems, there exists a limited amount of research dedicated to design of coordinated distributed control for nonlinear systems. In [76], a pseudo-model coordination method was proposed, which uses exact linearization of interconnected ODE systems. Coordination of non-linear networks governed by open-loop stable DAE systems are also investigated by [89] and [68]; however, closed-loop stability of coordinated control networks has not been addressed. Another challenge is the situation when the coordination scheme terminates prior to convergence. This could occur when there is insufficient time to complete the required calculations to converge to a solution.

In this work, design of CDMPC for nonlinear systems based on the price-driven coordination approach is considered. In the price-driven coordination, the coordinator calculates "prices" for local MPCs, based on information received from them, in order to coordinate their actions. It has been proven that the price-driven CDMPC can converge to the corresponding centralized control system, when the coordinator and distributed controllers are allowed to iterate a sufficient number of times at each sampling time for linear systems [66, 20, 21, 67, 76]. In this chapter, an interior-point based CDMPC (CDMPC) scheme is proposed, based on the results of [66] and [76], which focuses on a class of interacting nonlinear systems. In the proposed approach, an interior-point based model predictive controller (MPC) is designed for each subsystem and a price-driven scheme coordinates the actions of the distributed subsystem controllers to achieve centralized optimal performance and stability of the entire system. The proposed scheme is developed based on the assumption that a hypothetical centralized MPC, which uses successive linearization of the nonlinear system at every sampling time, can be designed to stabilize the closed-loop nonlinear system. A key idea behind this scheme is relating the hierarchical dual decomposition of the hypothetical centralized controller with a minor modification applied to the existing network of decentralized controllers. The modification to the decentralized MPC controllers is equivalent to relaxing interaction equations into the objective function of local controllers by assigning a price vector as a penalization coefficient. In this bilevel optimization problem the coordinator, which constructs the upper-level problem, receives information from the distributed subsystem MPCs, which belong to the lower-level problem. During coordination cycles between the coordinator and the plan-wide control system, the price value is sent to the distributed controllers to coordinate their actions in an iterative fashion and achieve the performance and stability of the centralized MPC. Sufficient conditions for ensuring the stability of the proposed CDMPC are derived. Note that convergence of the algorithm

and closed-loop stability are considered when the system is stopped prematurely prior to convergence. The performance of the proposed CDMPC is illustrated using a benzene alkylation process.

2.1.1 System Description

In this chapter, a class of nonlinear systems composed of m interconnected open-loop stable subsystems is considered. Each of the subsystems can be described by the following state-space model:

$$\dot{x}_i(t) = f_i(x, u) \tag{2.1}$$

where $i = 1, \ldots, m$, $x_i(t) \in \mathbb{R}^{n_{x_i}}$ denotes the vector of state variables of subsystem i, $x = [x_1^T \cdots x_i^T \cdots x_m^T]^T \in \mathbb{R}^{n_x}$ denotes the state of the entire nonlinear system, and $u = [u_1^T \cdots u_i^T \cdots u_m^T]^T \in \mathbb{R}^{n_u}$ is the vector of control inputs with u_i denoting the vector of control inputs associated with subsystem i. It is assumed that $u_i \in \mathcal{C}_i \subset \mathbb{R}^{n_{u_i}}$ with $\mathcal{C}_i$ being a nonempty convex set defined as follows:

$$\mathcal{C}_i \triangleq \{u_i \in \mathbb{R}^{n_{u,i}} |\ A_i^{ineq} u_i \leq b_i^{ineq}\} \tag{2.2}$$

where $i = 1, \ldots, m$. Note that in the remainder of this chapter, without the loss of generality, consider $\mathcal{C}_i$ to include only simple bound on control actions:

$$\mathcal{C}_i \triangleq \{u_i \in \mathbb{R}^{n_{u,i}} |\ lb_{u,i} \leq u_i \leq ub_{u,i}\} \tag{2.3}$$

with $lb_{u,i}$ being the lower bound and $ub_{u,i}$ being the upper bound over the control actions of subsystem 'i'. Note that, no bounds are considered over state variables of the

subsystems. This is mainly due to the methodology used in this chapter, which is based on successive linearization of the nonlinear system. This would limit the study, specially, when the CDMPC algorithm is stopped prematurely prior to convergence, so that the feasibility of nonlinear system may not be guaranteed. Here, it is assumed that the subsystems are interconnected through states and manipulated variables, i.e. the manipulated input variable of one sub-system might affect other sub-systems as an internal variable in the models. The dynamics of the entire system can be described as follows:

$$\dot{x}(t) = f(x, u) \tag{2.4}$$

where $f = [f_1^T \cdots f_i^T \cdots f_m^T]^T$. It is assumed that f is a twice differentiable Lipschitz vector function. The states of the m subsystems, x_i $(i = 1, \ldots, m)$, are assumed to be sampled synchronously at time instants $t_k = k\Delta t$ with $k = 0, 1, \ldots$. Note that in the remainder of this chapter, k is used to denote t_k in the discrete-time model. Based on the Lipschitz properties of $f(x(t), u(t))$, there is a unique exact discrete time nonlinear model [91] for system (2.4) expressed as:

$$x(k+1) = F_h^E(x(k), u(k)) \tag{2.5}$$

where: h is integration step for the discrete-time approximation. It is assumed that the corresponding sampling period T is fixed and h can be chosen independent of T arbitrarily. Although F_h^E may exist, the usual approach is to use an approximate discrete time system instead:

$$x(k+1) = F_h^A(x(k), u(k)) \tag{2.6}$$

which can be derived using explicit Runge-Kutta methods [44]. Note that numerical integration accuracy of the continuous-time model (2.4) depends upon the step-size 'h',

as does the accuracy of discrete time model.

Assumption 1 *[Model Consistency [78]] For a sampling period T, there exist an upper bound on the integration step-size $h \leq h^*$ and $\chi \in \mathcal{K}_\infty$ [1] such that $F_h^E \subseteq F_h^A + \chi(T)$. In other words, it is assumed that for a given sampling period T, the discretization method of the continuous-time model (2.4) should ensure that as the integration step-size $h \to 0$ the approximate model (2.6) converges to the exact model (2.5).*

Using a first-order Taylor expansion of (2.6) around the current measured state (x_k) and input (u_k) yields:

$$\hat{x}(k+1) = \Psi(k)\hat{x}(k) + \Gamma(k)\hat{u}(k) \tag{2.7}$$

where $\tilde{x}$ and $\tilde{u}$ are deviation variables. Additionally, the values of the matrices Ψ, Γ are determined at each sampling time as:

$$\Psi(k) := \frac{\partial F_h^A}{\partial x}|_{x=x_k, u=u_k} \tag{2.8a}$$

$$\Gamma(k) := \frac{\partial F_h^A}{\partial u}|_{x=x_k, u=u_k} \tag{2.8b}$$

In this work, it is assumed that all elements of Jacobian matrices $\Psi(k)$ and $\Gamma(k)$ are uniformly bounded for the integration step $h \in (0, h^*)$. At each control interval, the prediction model is constructed from $\Psi(k)$ and $\Gamma(k)$, as a linear time invariant state-space model along the prediction horizon. Thus, control actions are calculated using the successively linearized approximate model (2.8) and these actions are applied to the continuous plant (2.4).

[1] A continuous function $\chi : [0, \infty) \longmapsto [0, \infty)$ is said to belong to class $\mathcal{K}_\infty$, if it is strictly increasing; $\chi(0) = 0$ and $\chi(r) \to \infty$ as $r \to \infty$ [53].

2.1.2 Structure of the Centralized MPC

In this section, a monolithic (centralized) MPC is designed based on the successively linearized model (2.7) that satisfies the input constraints, for all x inside a compact set S containing the origin. The stable steady-state operating point of the system is transformed to the origin in order to have a unified definition of S. Note that, in the successive linearization approach, predicted trajectories inside controllers are constructed based on time-invariant linear models at the beginning of each sampling period. This could limit the scope of this study to nonlinear systems with low to moderate degrees of nonlinearity; so that, the dynamic behavior of the system can be captured, with acceptable accuracy, within the chosen prediction horizon.

Remark 2 *[91] A generic type finite-time MPC, based on the successively linearized approximate model (2.8) at time instant k, can be formulated as:*

$$\min_{X,U} J = \frac{1}{2}\Big(X(k)^T Q X(k) + U(k)^T R U(k)\Big) + \frac{1}{T}\hat{x}(k+H_p|k)^T P \hat{x}(k+H_p|k) \tag{2.9a}$$

$$s.t.\ \hat{x}(k+l+1|k) = \Psi(k)\hat{x}(k+l|k) + \Gamma(k)\hat{u}(k+l|k), \tag{2.9b}$$

$$\hat{u}(k+l|k) = \hat{u}(k+H_u|k),$$

$$for\ H_u \leq l \leq H_p - 1 \tag{2.9c}$$

$$\hat{u}(k+l|k) \in \mathcal{C} \tag{2.9d}$$

for $l = 0, \cdots, H_p$; where: H_p is the prediction horizon and H_u is the control horizon; and, $\hat{x}$ and $\hat{u}$ are states and manipulated input variables inside the controller, respectively; and $\mathcal{C} = \bigcup_{i=1,\cdots,m} \mathcal{C}_i$. Additionally, $X(k) = [\hat{x}(k+1|k)^T, \cdots, \hat{x}(k+H_p|k)^T]^T$ is the vector of the predicted state trajectory; $U(k) = [\hat{u}(k|k)^T, \cdots, \hat{u}(k+H_p-1|k)^T]^T$ is the vector of the calculated manipulated variable moves; Q is a positive definite block-diagonal weighting matrix for the states (i.e., $Q = diag\{Q_{ii}\}$); R is a positive definite block-

diagonal weighting matrix for the manipulated variables of the overall system (i.e., $R = diag\{R_{ii}\}$); and P is a positive definite block-diagonal weighting matrix for the terminal cost of the overall system (i.e., $P = diag\{P_{ii}\}$). In other words, the optimization problem (2.9) is meant to ensure [91]:

$$\hat{x}(k + H_p|k) \in \{x|x^T Px \leq \delta\} \tag{2.10}$$

for some $\delta > 0$; where: $0 \in \{x|x^T Px \leq \delta\}$.

Consider the approximate discrete-time model (2.6) and the generic MPC controller (2.9). Denote $\Delta V(x(k)) = V(x(k)) - V(x(k-1))$, where: $V(x(k)) = TJ(\hat{x}^*(k), \hat{u}^*(k))$ and $(\hat{x}^*(k), \hat{u}^*(k))$ the optimum trajectory of (2.9). Then $V(x(k)) > 0$ is the Lyapunov function of the closed-loop system, i.e. $\Delta V(x(k)) \leq 0$, if [91]:

$$\begin{aligned} &F_h^A(\hat{x}^*(k + H_p - 1|k), \hat{u}^*(k + H_p - 1|k))^T \times P \times F_h^A(\hat{x}^*(k + H_p - 1|k), \hat{u}^*(k + H_p - 1|k)) \\ &< \hat{x}^*(k + H_p - 1|k)^T P \hat{x}^*(k + H_p - 1|k) \end{aligned} \tag{2.11}$$

In the remainder of this chapter, it is assumed that the continuous time system (2.4) is open-loop stable. This implies that, the exact and the approximate discrete-time models, (2.5) and (2.6), are also open-loop stable. Accordingly, the generic MPC formulation (2.9) can be simplified to:

$$\min_{X,U} J = \frac{1}{2}\Big(X(k)^T Q X(k) + U(k)^T R U(k)\Big) \tag{2.12a}$$

$$\text{s.t. } \hat{x}(k + l + 1|k) = \Psi(k)\hat{x}(k + l|k) + \Gamma(k)\hat{u}(k + l|k), \tag{2.12b}$$

$$\hat{u}(k + l|k) = \hat{u}(k + H_u|k),$$

$$\text{for } H_u \leq l \leq H_p - 1 \tag{2.12c}$$

$$\hat{u}(k + l|k) \in \mathcal{C} \tag{2.12d}$$

if the prediction horizon H_p is chosen large enough to ensure closed-loop stability of (2.6). Throughout this chapter, it is assumed that a MPC controller (2.12) can be designed, such that: for a sufficient length of prediction horizon, the closed-loop system (2.6) is locally practically stable [85], without the terminal cost term included into the objective function.

Denote $V_h(x)$ the Lyapunov function for the closed-loop system under the centralized MPC scheme (2.12). Then, the following assumption is an immediate conclusion from the converse Lyapunov theorem stated in [78, 91]:

Assumption 3 *[Lyapunov stability certificates [78]] Given T, a family of continuous Lyapunov functions V_h, $(\alpha_1, \alpha_2) \in \mathcal{K}_\infty$, a positive-definite function α_3, strictly positive real numbers (σ_1, σ_2) with $\sigma_2 \leq D$ and a positive real number c_1, there exists $h^* > 0$ such that for $0 < h < h^*$ and $x \in \mathcal{B}_D$ [2]:*

$$\alpha_1(||x||) \leq V_h(x) \leq \alpha_2(||x||) \tag{2.13a}$$

$$V_h(F_h^A(x,u)) - V_h(x) \leq -T\alpha_3(||x||) \tag{2.13b}$$

and for all $x_a, x_b \in \mathcal{B}_D - \mathcal{B}_{\sigma_2}$ with $||x_a - x_b|| \leq c_1$:

$$\Big|\Big|V_h(x_a) - V_h(x_b)\Big|\Big| \leq \sigma_1 \tag{2.13c}$$

then, (2.6) is (T, D) stable with a continuous Lyapunov function V_h.

In order to proceed, the centralized MPC (2.12) problem is reformulated with an interior-point method. In the remainder, consider a level set of $V_h(x) = \rho$; then, an estimate for the region of attraction of the closed-loop system under the control of the centralized MPC (2.12) can be defined as: $\varpi_\rho = \{x \in S | V_h(x) < \rho\}$.

[2] Here, set $\mathcal{B}_\sigma$ is defined as $\mathcal{B}_\sigma = \{x \in R^n | 0 \leq ||x|| \leq \sigma\}$

In this chapter, an interior-point approach is used to design the centralized MPC controller. Using the interior-point method, unlike the active-set approach [66], it is not required to identify the correct set of active constraints. The finite-time centralized MPC formulation based on the successively linearized approximate model (2.8), at time instant k, is:

$$\min_{X,U} J_C = \frac{1}{2}\Big(X(k)^T Q X(k) + U(k)^T R U(k)\Big) + \mu\Omega_U(U(k)) \tag{2.14a}$$

$$\text{s.t. } \hat{x}(k+l+1|k) = \Psi(k)\hat{x}(k+l|k) + \Gamma(k)\hat{u}(k+l|k), \tag{2.14b}$$

$$\hat{u}(k+l|k) = \hat{u}(k+H_u|k),$$

$$\text{for } H_u \leq l \leq H_p - 1 \tag{2.14c}$$

for $l = 0, \cdots, H_p$. In (2.14), μ is the barrier weighting parameter and Ω_U is the logarithmic barrier function, which is defined as follows:

$$\Omega_U(U(k)) = \sum_{j=1}^{n_u} -\left(\sum_{l=1}^{H_u} \ln\big(ub_{u,j} - u_j(k+l|k)\big) + \sum_{j=1}^{n_u \times H_u} \ln\big(u_j(k+l|k) - lb_{u,j}\big)\right) \tag{2.14d}$$

In the remainder of this chapter, denote the optimal trajectory of the successively linearized centralized (SLC) problem (2.14), at time instant 'k', $OTSLC(k)$. In order to achieve local practical stability of the exact system (2.5) via the SLC-MPC controller (2.14), model consistency, existence of Lyapunov stability certificates based on the approximate model (2.6) and a bounded set of control actions must hold true. This can be stated in the following Lemma:

Lemma 4 *[91] Suppose model consistency, Assumption 1, and Lyapunov stability certificates, Assumption 3, hold true for a bounded set of control actions derived from the*

centralized MPC (2.14) based on successive linearization of the approximate model (2.6). Then, there exists $\beta \in \mathcal{KL}$ [3] *and* $D_1 \in (0, \Delta]$*; and for any* $\sigma > 0$ *there exists* $h^* > 0$ *such that: for all* $x_0 \in \{x \in R^n | 0 \leq ||x|| \leq D_1\}$ *and* $h \in (0, h^*)$*, the transient solution to the exact discrete-time closed-loop system* $\Phi_k^E(x_0, k)$ *satisfies:*

$$||\Phi_k^E(x_0, k)|| \leq \beta(||x_0||, kT) + \sigma \tag{2.15}$$

which implies that the closed-loop system of exact discrete-time system (or equivalently, the continuous-time system) is locally practically stable around the origin.

In the following sections, the idea of price-driven CDMPC approach for linear systems is extended to the control of nonlinear systems based on successive linearization of (2.6) at each sampling time. The centralized MPC will provide performance benchmarks for evaluation of the CDMPC for nonlinear systems.

2.2 The Proposed Coordination Algorithm

In this section, a price-driven CDMPC algorithm is proposed for nonlinear systems, based on successive linearization. The main idea is to form a multilevel network consisting of a coordination level interacting with an existing network of MPC controllers, in order to achieve a higher performance compared to the fully decentralized scheme. In this method, distributed controllers are modified to account for interactions inside their objective functions using a price vector. The information flow between the coordinator and the local controllers provides each MPC with the optimal price vector and as a result the optimum plant-wide trajectory. This structure is depicted in Figure 2.1, in which: p

[3] A continuous function $\varrho : [0, a) \times [0, \infty) \longmapsto [0, \infty)$ is said to belong to class $\mathcal{KL}$, if it belongs to class $\mathcal{K}$ in its first argument and decreasing in its second argument such that: $\varrho(., s) \to 0$ as $s \to \infty$ [53].

denotes the price vector; Θ_i denotes the local interaction matrix of sub-system i; and Z_i is the optimal trajectory of local MPCs with respect to (w.r.t.) the current price value.

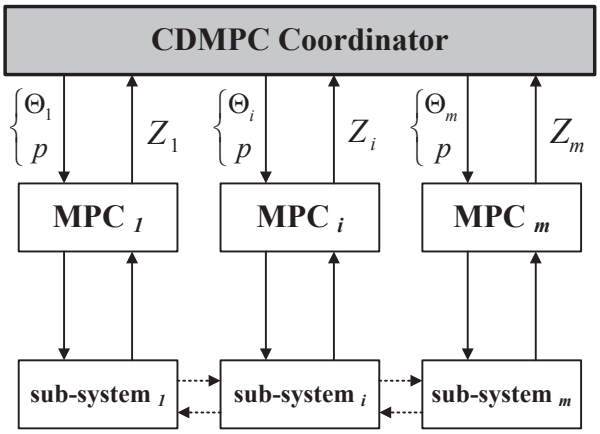


Figure 2.1: Architecture and information flow of the proposed CDMPC.

For each subsystem, the MPC is formulated based on successively linearized subsystem model. Specifically, for subsystem i, $i = 1, \ldots, m$, the prediction model used in the design of the subsystem MPC at time instant k takes the following form:

$$\hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k) + \sum_{j\neq i}(1-\beta)\Psi_{ij}\hat{x}_j(k|k) + \hat{v}_i(k+l|k) \tag{2.16a}$$

$$\hat{x}_i(k|k) = x_i(k) \tag{2.16b}$$

with:

$$\beta = \begin{cases} 0 & l = 0 \\ 1 & l = 1, \cdots, H_p - 1 \end{cases} \tag{2.16c}$$

where: Ψ_{ii} and Γ_{ii} denote coefficient matrices for linearized system description of subsystem i, namely the ith block diagonal elements of the plant-wide matrices Ψ and

Γ given in (2.8); and $\hat{v}_i$ is defined as the interacting or linking variable that contains unknown interaction information between different subsystems. Note that the term $\hat{v}_i$ in (2.16) characterizes the interaction of subsystem 'i' with other subsystems. We define the interaction error for subsystem 'i' as:

$$e_i(k+l|k) \triangleq \hat{v}_i(k+l|k) - \sum_{j \neq i}^{m} \left(\beta \Psi_{ij} \hat{x}_j(k+l|k) + \Gamma_{ij} \hat{u}_j(k+l|k))\right) \tag{2.17}$$

for $l = 0, \cdots, H_p - 1$. The coordinator finds a price for subsystem i, such that the interaction term $\hat{v}_i$ determined by the price, ensures that $e_i(k+l|k) = 0$ and the CDMPC approaches the performance of the centralized MPC trajectory, e.g. $OTSLC(k)$. The overall interaction error over the prediction horizon can be described as follows:

$$E(k|k) \triangleq \begin{bmatrix} E_1(k|k) \\ \vdots \\ E_m(k|k) \end{bmatrix} \tag{2.18a}$$

where:

$$E_i = \begin{bmatrix} e_i(k|k) \\ e_i(k+1|k) \\ \vdots \\ e_i(k+H_p-1|k) \end{bmatrix} \tag{2.18b}$$

According to (2.17), $E_i(k|k)$ can be written as:

$$E_i(k|k) = \begin{bmatrix} -\Gamma_{1i}\hat{u}_i(k|k) \\ -\Big[\Psi_{1i}\hat{x}_i(k+1|k) + \Gamma_{1i}\hat{u}_i(k+1|k)\Big] \\ \vdots \\ -\Big[\Psi_{1i}\hat{x}_i(k+H_p-1|k) + \Gamma_{1i}\hat{u}_i(k+H_u-1|k)\Big] \\ \hline \vdots \\ \hline \hat{v}_i(k|k) \\ \hat{v}_i(k+1|k) \\ \vdots \\ \hat{v}_i(k+H_p-1|k) \\ \hline \vdots \\ \hline -\Gamma_{mi}\hat{u}_i(k|k) \\ -\Big[\Psi_{mi}\hat{x}_i(k+1|k) + \Gamma_{mi}\hat{u}_i(k+1|k)\Big] \\ \vdots \\ -\Big[\Psi_{mi}\hat{x}_i(k+H_p-1|k) + \Gamma_{mi}\hat{u}_i(k+H_u-1|k)\Big] \end{bmatrix} \tag{2.19a}$$

$$= \Theta_i(k) \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix} \tag{2.19b}$$

where: $X_i(k) \triangleq [\hat{x}_i(k+1|k)^T, \cdots, \hat{x}_i(k+H_p|k)^T]^T$ is the vector containing predictions for state variables; $U_i(k) \triangleq [\hat{u}_i(k+1|k)^T, \cdots, \hat{u}_i(k+H_p|k)^T]^T$ is the vector containing predictions for manipulated input variables; $V_i(k) \triangleq [\hat{v}_i(k+1|k)^T, \cdots, \hat{v}_i(k+H_p|k)^T]^T$ is the vector containing predictions for linking variables; and $\Theta_i(k)$ is the coefficient matrix

for the linking constraints defined as:

$$\Theta_i(k) = \left[\theta_{1,i}^T, \quad \cdots, \quad \theta_{i,i}^T, \quad \cdots, \quad \theta_{m,i}^T\right]^T \tag{2.20a}$$

where:

$$\theta_{j,i} = \begin{cases} \left[0_{H_p n_{x_i} \times H_p n_{x_i}}, 0_{H_p n_{x_i} \times H_u n_{u_i}}, I\right] & \text{for } j = i \\ \left[\theta_{\Psi_{j,i}}, \theta_{\Gamma_{j,i}}, 0_{H_p n_{x_j} \times H_p n_{x_i}}\right] & \text{for } j \neq i \end{cases} \tag{2.20b}$$

with '0' being zero matrices of appropriate dimensions, I being an identity matrix of size $H_p n_{x_i} \times H_p n_{x_i}$, as well as $\theta_{\Psi_{j,i}}$ and $\theta_{\Gamma_{j,i}}$ being $H_p n_{x_j} \times H_p n_{x_i}$ and $H_p n_{x_j} \times H_u n_{u_i}$ matrices, respectively, and defined as:

$$\theta_{\Psi_{j,i}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ -\Psi_{ji} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & -\Psi_{ji} & 0 \end{bmatrix}, \theta_{\Gamma_{j,i}} = \begin{bmatrix} -\Gamma_{ji} & 0 & \cdots & 0 \\ 0 & -\Gamma_{ji} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\Gamma_{ji} \end{bmatrix} \tag{2.20c}$$

Equivalently, matrix Θ_i can be written in terms of an augmented matrix of coefficients for X_i, U_i, and V_i as follows:

$$\Theta_i = \left[\begin{array}{c|c|c} \theta_{\Psi_{1,i}} & \theta_{\Gamma_{1,i}} & 0 \\ \vdots & \vdots & \vdots \\ 0_{H_p n_{x_i} \times H_p n_{x_i}} & 0_{H_p n_{x_i} \times H_p n_{u_i}} & -I_{H_p n_{x_i} \times H_p n_{x_i}} \\ \vdots & \vdots & \vdots \\ \theta_{\Psi_{m,i}} & \theta_{\Gamma_{m,i}} & 0 \end{array}\right] \tag{2.21a}$$

$$= [\Theta_{X_i}, \Theta_{U_i}, \Theta_{V_i}] \tag{2.21b}$$

and the overall interaction error E, defined in (2.18), can be written as:

$$E(k) = \sum_{i=1}^{m} \Theta_i(k) \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix} \tag{2.22}$$

Based on the above, the optimization problem (2.14) can be re-written as follows:

$$\min_{X,U,V} J_P = \sum_{i=1}^{m} J_{P_i} \tag{2.23a}$$

$$\text{s.t. } \hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k) + \sum_{j\neq i}(1-\beta)\Psi_{ij}\hat{x}_j(k|k) + \hat{v}_i(k+l|k)$$

$$\text{for } l = 0, \ldots, H_p - 1 \tag{2.23b}$$

$$\sum_{i=1}^{m} \Theta_i \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix} = 0 \tag{2.23c}$$

$$\hat{u}_i(k+l|k) = \hat{u}_i(k+H_u|k)$$

$$\text{for } H_u \leq l \leq H_p - 1 \tag{2.23d}$$

where:

$$J_{P_i} = \frac{1}{2}\Big((X_i(k))^T Q_{ii}(X_i(k)) + U_i(k)^T R_{ii} U_i(k)\Big) + \mu\Omega_{U_i}(U_i(k)) \tag{2.23e}$$

$$\Omega_{U_i}(U_i(k)) = \sum_{j=1}^{n_{u_i}} -\left(\sum_{l=1}^{H_u} \ln\left(ub_{u_i}(j) - u_i^j(k+l|k)\right) + \sum_{j=1}^{H_u} \ln\left(u_i^j(k+l|k) - ub_{u_i}(j)\right) \right) \tag{2.23f}$$

Optimization problem (2.23) is the main core for the design of the CDMPC system. Note that the cost function and constraints are separable by subsystem. Relaxing con-

straint (2.23c), so that it is also separable in terms of subsystems yields:

$$\min_{X,U,V} J_D = \sum_{i=1}^{m} J_{D_i} \tag{2.24a}$$

$$\text{s.t. } \hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k) + \sum_{j\neq i}(1-\beta)\Psi_{ij}\hat{x}_j(k|k) + \hat{v}_i(k+l|k)$$

$$\text{for } l = 0, \ldots, H_p - 1 \tag{2.24b}$$

$$\hat{u}_i(k+l|k) = \hat{u}_i(H_u|k)$$

$$\text{for } H_u \leq l \leq H_p - 1 \tag{2.24c}$$

where:

$$J_{D_i} = \frac{1}{2}\Big((X_i(k))^T Q_{ii}(X_i(k)) + U_i(k)^T R_{ii} U_i(k)\Big) + \mu\Omega_{U_i}(U(k)) + p^T \Theta_i \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix} \tag{2.24d}$$

In (2.24), "p" is a price vector provided by the coordinator and is used to drive the distributed system to the plant-wide optimal operation, see Section 2.2.1 for more details. Optimization problem (2.24) is separable and in the proposed CDMPC, the formulation of local MPC i, $i = 1, \ldots, m$, is as follows:

$$\min_{X_i,U_i,V_i} J_{D_i} = \frac{1}{2}\Big((X_i(k))^T Q_{ii}(X_i(k)) + U_i(k)^T R_{ii} U_i(k)\Big) + \Omega_{U_i}(U(k)) + p^T \Theta_i \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix} \tag{2.25a}$$

$$\text{s.t. } \hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k) + \sum_{j\neq i}(1-\beta)\Psi_{ij}\hat{x}_j(k|k) + \hat{v}_i(k+l|k)$$

$$\text{for } l = 0, \ldots, H_p - 1 \tag{2.25b}$$

$$\hat{u}_i(k+l|k) = \hat{u}_i(H_u|k)$$

$$\text{for } H_u \leq l \leq H_p - 1 \tag{2.25c}$$

2.2.1 Price-Driven Coordinator Formulation

In order to simplify the description of the coordinator design, the overall CDMPC formulation (2.24) is re-written in a more compact form:

$$\min_{Z} \; J_D = \sum_{i=1}^{m} \bar{J}_{D_i} \tag{2.26a}$$

$$\text{s.t. } G_i^{eq} Z_i(k) = g_i^{eq} \tag{2.26b}$$

$$G_i^{ineq} Z_i(k) \leq g_i^{ineq} \tag{2.26c}$$

with J_{D_i} rewritten as:

$$\bar{J}_{D_i} = \Big(\frac{1}{2}\big((Z_i(k))^T \Upsilon_i(Z_i(k)) + \Phi_i^T Z_i(k) + p^T \Theta_i Z_i(k)\Big) \tag{2.26d}$$

and G_i^{eq} defined as:

$$G_i^{eq} = [G_{i,X_i}^{eq}, G_{i,U_i}^{eq}, G_{i,V_i}^{eq}] \tag{2.26e}$$

$$G_{i,X_i}^{eq} = \begin{bmatrix} I_{n_{x_i}} & 0 & \cdots & 0 \\ -\Psi_{ii} & I_{n_{x_i}} & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & -\Psi_{ii} & I_{n_{x_i}} \end{bmatrix}, \tag{2.26f}$$

$$G_{i,U_i}^{eq} = \begin{bmatrix} -\Gamma_{ii} & 0 & \cdots & 0 \\ 0 & -\Gamma_{ii} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\Gamma_{ii} \end{bmatrix}, \tag{2.26g}$$

$$G_{i,V_i}^{eq} = I_{H_p n_{x_i} \times H_p n_{x_i}} \tag{2.26h}$$

$$g_i^{eq} = \begin{bmatrix} \sum_{j \neq i} (1-\beta) \Psi_{ij} \hat{x}_j(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.26i}$$

In (2.26), $Z_i(k)$ represents the vector of decision variables, namely:

$$Z_i(k) = [X_i(k)^T, U_i(k)^T, V_i(k)^T]^T \tag{2.27}$$

The inequalities within the MPC problem (2.26) can be relaxed, using barrier functions, to:

$$\min_{Z} \; J_D = \sum_{i=1}^{m} \bar{J}_{D_i} \tag{2.28a}$$

$$\text{s.t. } G_i^{eq} Z_i(k) = g_i^{eq} \tag{2.28b}$$

where:

$$\bar{J}_{D_i} = \Big(\frac{1}{2} \big((Z_i(k))^T \Upsilon_i (Z_i(k)) + \mu_i \Omega_{Z,i}(Z_i(k)) + p^T \Theta_i Z_i(k) \Big) \tag{2.28c}$$

In (2.28), Ω_Z is the corresponding logarithmic barrier function for subsystem i:

$$\Omega_{Z,i} = \begin{cases} 0 & X_i(k), V_i(k) \\ \Omega_{U_i}(U_i(k)) & U_i(k) \end{cases} \tag{2.29a}$$

with:

$$\Omega_{U_i}(U_i(k)) = - \sum_{j=1}^{n_{u_i} \times H_u} \ln \left(g_{i,U}^{ineq}(j) - G_{i,U}^{ineq}(j,:) \times U_i \right) \tag{2.29b}$$

In (2.28), the price vector p can be interpreted as the Lagrange multipliers associated with interaction equality constraints. This vector is determined by the coordinator at each coordination cycle, based on the information provided by the local MPC controllers. In order to find a plant-wide solution to the distributed system (2.25), the primal-dual optimization problem (2.30) is solved:

$$\max_{p} \; J_D \tag{2.30a}$$

$$\min_{Z} J_D = \sum_{i=1}^{m} \bar{J}_{D_i} \tag{2.30b}$$

$$\text{s.t. } G_i^{eq} Z_i(k) = g_i^{eq} \tag{2.30c}$$

where: $Z = [X^T, U^T, V^T]^T$. Denote $\mathcal{L}$ the overall Lagrangian of the local sub-problems (2.28); and let $\mathcal{L}_i$ defined as:

$$\mathcal{L}_i(Z_i, \vartheta_i, \mu_i, p) = \frac{1}{2} Z_i(k)^T \Upsilon_i Z_i(k) + p^T \Theta_i Z_i(k) + \vartheta_i^T \big(G_i^{eq} Z_i(k) - g_i^{eq}\big) + \mu_i \Omega_{Z,i}(Z_i(k)) \tag{2.31}$$

with ϑ_i being the Lagrange multiplier of the equality constraints for sub-system "i". In Barrier methods, it is common to express the first-order optimality conditions via the modified KKT conditions and solve the resulting set of equations with a Newton's method variant [12]. Correspondingly, the modified KKT conditions of the primal problem are:

$$\nabla_{Z_i} \mathcal{L}_i(Z_i, \vartheta_i, \lambda_i, p) = \Upsilon_i Z_i(k) + \Theta_i^T p + G_i^{eq^T} \vartheta_i + \sum_{j=1}^{m_{u_i} \times H_u} \lambda_i(j) \big(G_i^{ineq}(j,:)\big)^T = 0 \tag{2.32a}$$

$$\mathcal{R}_{1,i}(Z_i, p) = G_i^{eq} Z_i(k) - g_i^{eq} = 0 \tag{2.32b}$$

$$\mathcal{R}_{2,i}(Z_i, p) = diag(\lambda_i(j))\zeta_i - \mu_i \rho = 0 \tag{2.32c}$$

where:

$$\zeta_i(j) = G_i^{ineq}(j,:) Z_i(k) - g_i^{ineq}(j) \tag{2.32d}$$

$$\rho = [1, \cdots, 1]^T \tag{2.32e}$$

and $\lambda_i(j)$ can be interpreted as the Lagrange multiplier associated with the 'j'th inequality constraint of sub-system 'i'. The use of $\lambda_i(j)$ simplifies the notation and can always be eliminated with the relation defined in (2.32c). It is ensured that the parameter μ_i is monotonically decreasing to zero at each iteration, e.g. by a factor of 10 to 20 [12]. As $\mu_i \to 0$ the KKT conditions (2.32) converge to that of the original problem (2.26).

On the other hand, the dual problem is an unconstrained optimization problem w.r.t

the price vector "p". Denote the optimal value of Z at the current value of p as $Z^*(k)$, then according to (2.30), the Lagrangian of the unconstrained problem can be written as:

$$\bar{\mathcal{L}}(p, Z^*(k)) = -J_D(p, Z^*(k)) \tag{2.33a}$$

based on the convex nature of the lower level problem, at Z_i^* the equality constraint (2.32b) is always satisfied, i.e. $G_i^{eq} Z_i^*(k) - g_i^{eq} = 0$ and $\mu_i = 0$. Therefore, penalizing the equality constraint with ϑ_i and adding $-\mu_i \Omega_{Z,i}(Z_i^*(k))$ to the right-hand-side of (2.33a) would result in:

$$\bar{\mathcal{L}}(p, Z^*(k)) = -\, J_D(p, Z^*(p)) - \vartheta_i^T \big(G_i^{eq} Z_i^*(k) - g_i^{eq}\big) - \mu_i \Omega_{Z,i}(Z_i^*(k)) \tag{2.33b}$$

in other words, at $Z^*(k)$:

$$\bar{\mathcal{L}}(p, Z^*) = -\mathcal{L}(p, Z^*) \tag{2.33c}$$

The first-order optimality condition for the dual problem can be written as:

$$\nabla_p \bar{\mathcal{L}}(p, Z^*) = -\left(\Upsilon Z^*(k) + \Theta^T p + G^{eq^T} \vartheta + \sum_{j=1}^{m_u \times H_u} \lambda(j) \big(G^{ineq}(j)\big)^T\right) \frac{dZ}{dp} - \Theta Z^* = 0 \tag{2.34a}$$

but $\Upsilon Z^*(k) + \Theta^T p + G^{eq^T} \vartheta + \sum_{j=1}^{m_u \times H_u} \lambda(j) \big(G^{ineq}(j)\big)^T = 0$, since $\nabla_Z \mathcal{L} = 0$. This is equivalent to say:

$$\nabla_p \bar{\mathcal{L}}(p, Z^*) = \; - \Theta Z^* = 0 \tag{2.34b}$$

in other words, the KKT condition of the dual problem is:

$$\Theta Z^* = 0. \tag{2.34c}$$

Hence, an efficient price-updating method, such as Newton's method [4], can be used to achieve the plant-wide performance for the coordination problem defined in (2.28):

$$p(k+1) = p(k) - \alpha \mathcal{H}^{-1}(k)\mathcal{J}(k), \tag{2.35}$$

where $\mathcal{J}(k)$ is the Jacobian and $\mathcal{H}(k)$ is the Hessian matrix of $\mathcal{L}$, and α is the step size in Newton's method that is determined using line search method given in [19]. The Jacobian is calculated by:

$$\mathcal{J}(k) = \Theta Z^* = \sum_{i}^{m} \Theta_i Z_i^*(k) \tag{2.36}$$

Then the Jacobian is equal to the overall interaction error vector '$E(k)$', defined in (2.19a). Accordingly, the Hessian matrix $\mathcal{H}(k)$ can be calculated as:

$$\mathcal{H}(k) = \frac{d\mathcal{J}(k)}{dp(k)} = \frac{dE(k)}{dp(k)} = \sum_{i}^{m} \Theta_i(k) \nabla_p Z_i^*(k) \tag{2.37}$$

Thus, the coordinator calculates $\Theta_i(k)$ at each sampling time and is provided with the sensitivity matrix of the decision variables $Z_i(k)$ with respect to changes in the price

[4] Gradient-based methods and Quasi-Newton methods can also be deployed [76].

vector information. The problem of determining the sensitivities can be written as:

$$\Gamma_i \begin{bmatrix} \nabla_p Z_i(k) \\ \nabla_p \vartheta_i(k) \\ \nabla_p \lambda_i \end{bmatrix} = \begin{bmatrix} -\Theta_i^T \\ 0 \\ 0 \end{bmatrix} \tag{2.38a}$$

where:

$$\Gamma_i = \begin{bmatrix} \Upsilon_i & G_i^{eq^T} & G_i^{ineq^T} \\ G_i^{eq} & 0 & 0 \\ diag(\lambda_i(j)) G_i^{ineq} & 0 & diag\big(\zeta_i(j)\big) \end{bmatrix}. \tag{2.38b}$$

The system of equations (2.38) is derived from second-order optimality conditions of the primal CDMPC problem. The second-order optimality conditions are:

$$\nabla^2_{Z_i(k),Z_i(k)} \mathcal{L}_i dZ_i(k) + \nabla^2_{Z_i(k),p^s} \mathcal{L}_i dp^s + \nabla_{Z_i(k)} \mathcal{R}_{1,i}^T d\vartheta_i + G_i^{ineq^T} d\lambda_i = 0 \tag{2.39a}$$

$$\nabla_{Z_i(k)} \mathcal{R}_{1,i} dZ_i(k) + \nabla_{p^s} \mathcal{R}_{1,i} dp^s = 0 \tag{2.39b}$$

$$\nabla_{Z_i(k)} \mathcal{R}_{2,i} dZ_i(k) + \nabla_{p^s} \mathcal{R}_{2,i} dp^s + \nabla_{\lambda_i} \mathcal{R}_{2,i} dp^s = 0 \tag{2.39c}$$

where 's' denotes the iteration number of the coordination algorithm. The implementation of the CDMPC algorithm at sampling time k can be summarized as illustrated in Algorithm 1:

Algorithm 1: the proposed successive linearization CDMPC algorithm

Input: coordination cycle counter $s = 0$, price vector $p = 0$, stopping criterion $\epsilon > 0$, $Z_i^0(k)$;
Local MPCs: Calculate $\Psi(k)$ and $\Gamma(k)$ using (2.8);
Coordinator: Send the value of p^s to the local MPCs;
Coordinator: Calculate the overall interaction error $E(k)$ using (2.22);
while $||E(k)|| > \epsilon$ **do**
  Local MPCs: Solve problem (2.24) using the interior-point method and update $\bar{Z}_i^s(k)$;
  Local MPCs: Calculate sensitivity matrices $\mathcal{J}(k)$ and $\mathcal{H}(k)$ using (2.36) and (2.37);
  Local MPCs: Send $\mathcal{J}(k)$ and $\mathcal{H}(k)$ to the coordinator;
  Coordinator: Calculate the price vector p^s using (2.35);
  Coordinator: Send p^s to the local MPCs;
  Coordinator: Calculate the overall interaction error $E(k)$ using (2.22)
  $s = s + 1$;
Local MPCs: Apply the receding horizon action to the plant;
Output: $\bar{Z}_i^s(k)$, and p^s;

Proposition 5 *At time instant 'k', the proposed CDMPC scheme (2.30) will converge to a local optimal solution of problem (2.23) using the price update scheme (2.35).*

Proof. Define the merit function $\mathcal{M}$ for the CDMPC problem (2.30) as:

$$\mathcal{M} = \mathcal{L}^* - \mathcal{L}(p) \tag{2.40}$$

where $\mathcal{L}^*$ is the optimum value of the dual problem objective function $\mathcal{L}$. In fact, $\mathcal{M} \geq 0$ and is zero iff $p = p^*$. Consider the changes of $\mathcal{M}$ w.r.t. the CDMPC cycle 's':

$$\nabla_s \mathcal{M} = \Big(\frac{d\mathcal{M}}{dp}\Big)^T \frac{dp}{ds} \tag{2.41}$$

$$= \Big(-\frac{d\mathcal{L}(p)}{dp}\Big)^T \frac{dp}{ds} \tag{2.42}$$

which reduces to:

$$\nabla_s \mathcal{M} = -\mathcal{J}^T \frac{dp}{ds} \tag{2.43}$$

Furthermore, the rate of change of 'p' w.r.t. the iteration number 's' is directly proportional to the gradient ascent direction $\mathcal{J}$ in price update scheme (2.35). Thus, the rate of change of the merit function $\mathcal{M}$ w.r.t. the iteration number 's' is:

$$\nabla_s \mathcal{M} = -\delta \mathcal{J}^T \mathcal{J},\ \delta \geq 0 \tag{2.44}$$

which means $\nabla_s \mathcal{M} \leq 0$. Therefore, the proposed CDMPC scheme (2.30) converges to the corresponding (local) optimal plant-wide solution (2.23) at time instant 'k' with the price update scheme (2.35). ■

2.3 Stability Analysis

In this section, the nominal stability of the CDMPC algorithm is discussed. The analysis is based on the premise that the coordination algorithm has not converged to $OTSLC(k)$, i.e. $0 < ||E(k)|| \leq \epsilon$.

Proposition 6 *Consider the successively linearized process system (2.7) controlled by the CDMPC algorithm (2.28) is stopped prematurely at time instant 'k' with* $||\Delta E^s(k)|| \leq \epsilon$. *Let* μ_i *be monotonically decreasing to zero, i.e.* $\mu_i \longrightarrow 0$, *and* $\{U_i(k)^s\}$ *remains strictly feasible [40]. Then the norm of difference between the calculated control action* $U_D(k)$ *and the control action* $U_C(k)$, *corresponding to* $OTSLC(k)$, *remains bounded, namely* $||U_C(k) - U_D(k)|| \leq d_{U_{C,D}}(\epsilon)$.

Proof. Consider the first-order optimality condition for the subproblem 'i' given in (2.32) where coefficient matrices Θ_i, G_i^{eq} and G_i^{ineq} are:

$$\Theta_i = [\Theta_{i,X_i}, \Theta_{i,U_i}, I_{i,V_i}] \tag{2.45a}$$

$$G_i^{eq} = [G_{i,X_i}^{eq}, G_{i,U_i}^{eq}, -I_{i,V_i}] \tag{2.45b}$$

$$G_i^{ineq} = [0_{i,X_i}, G_{i,U_i}^{ineq}, 0_{i,V_i}] \tag{2.45c}$$

where I denotes a square identity matrix and 0 denotes a square zero matrix. This results in the set of equations:

$$Q_{ii}X_i^s + \Theta_{i,X_i}^T p^s + G_{i,X_i}^{eq^T}\vartheta_i^s = 0 \tag{2.46a}$$

$$R_{ii}U_i^s + \Theta_i^T p^s + G_{i,U_i}^{eq^T}\vartheta_i^s + \sum_{j=1}^{m_{u_i}\times H_u} \lambda_i(j)\big(G_{i,U_i}^{ineq}(j,:)\big)^T = 0 \tag{2.46b}$$

$$p_i^s - \vartheta_i^s = 0 \tag{2.46c}$$

$$V_i^s - G_{i,X_i}^{eq^T}X_i^s - G_{i,U_i}^{eq^T}U_i^s + g_i^{eq} = 0 \tag{2.46d}$$

where $p_i^s = I_{i,V_i} \times p^s$. Consequently, the overall lower-level problem can be written as:

$$QX^s + \Theta_X^T p^s + G_X^{eq^T}\vartheta^s = 0 \tag{2.47a}$$

$$RU^s + \Theta^T p^s + G_U^{eq^T}\vartheta^s + \sum_{j=1}^{m_u\times H_u} \lambda(j)\big(G_U^{ineq}(j)\big)^T = 0 \tag{2.47b}$$

$$p^s - \vartheta^s = 0 \tag{2.47c}$$

$$V^s - G_X^{eq^T}X^s - G_U^{eq^T}U^s + g^{eq} = 0 \tag{2.47d}$$

Hence, the price vector "p" and the Lagrange multiplier associated with the equality constraints "ϑ" are equal, according to 2.47. Therefore X_i^s, U_i^s and V_i^s can be directly

calculated at the optimum value of the local MPC controllers as:

$$X_i^s = -Q_{ii}^{-1}(\Theta_{i,X_i}^T + G_{i,X}^{eq^T}\Theta_{i,I_i}^T)p^s \tag{2.48a}$$

$$U_i^s = -R_{ii}^{-1}\Big((\Theta_{i,U_i}^T + G_{i,U_i}^{eq^T}\Theta_{i,I_i}^T)p^s + \sum_{j=1}^{m_{u_i}\times H_u} \lambda_i(j)\big(G_{i,U_i}^{ineq}(j,:)\big)^T\Big) \tag{2.48b}$$

$$V_i^s = G_{i,X_i}^{eq^T}X_i^s + G_{i,U_i}^{eq^T}U_i^s - g_i^{eq}. \tag{2.48c}$$

Note that $\lambda_i(j)$ is bounded since $\mu_i \longrightarrow 0$ and U_i^s is strictly feasible. Moreover, as the local MPC problem converges to the optimal value, μ_i becomes zero while all of the inequality constraints remain inactive. Thereby, according to (2.32) all the Lagrange multipliers will become zero when local MPCs converge, i.e. $\lambda_i \to 0$, hence:

$$X_i^s = -Q_{ii}^{-1}(\Theta_{i,X_i}^T + G_{i,X}^{eq^T}\Theta_{i,I_i}^T)p^s \tag{2.49a}$$

$$U_i^s = -R_{ii}^{-1}\big(\Theta_{i,U_i}^T + G_{i,U_i}^{eq^T}\Theta_{i,I_i}^T\big)p^s \tag{2.49b}$$

$$V_i^s = G_{i,X_i}^{eq^T}X_i^s + G_{i,U_i}^{eq^T}U_i^s - g_i^{eq}. \tag{2.49c}$$

Now consider the CDMPC algorithm is stopped premature, namely, the stopping criterion $||\Delta E^s(k)|| = \Big|\Big| \sum_{i=1}^m \Theta_i(k)Z_i^s(k)\Big|\Big| \leq \epsilon$, then based on (2.48) it can be shown that:

$$\begin{aligned} ||\Delta E^s(k)|| &= \Big|\Big| \sum_{i=1}^m \Theta_i(k)[X_i^s(k), U_i^s(k), V_i^s(k)]^T\Big|\Big| \leq \epsilon \\ &= \big|\big|\Pi \times p^s + \Xi\big|\big| \leq \epsilon \end{aligned} \tag{2.50a}$$

where:

$$\Pi = \sum_{i=1}^{m} \Bigg[\Theta_{i,X_i}\big(- Q_{ii}^{-1}(\Theta_{i,X_i}^T + G_{i,X}^{eq^T}\Theta_{i,I_i}^T)\big), \Theta_{i,U_i}\big(- R_{ii}^{-1}(\Theta_{i,U_i}^T + G_{i,U_i}^{eq^T}\Theta_{i,I_i}^T)\big), \\ G_{i,X_i}^{eq^T}\big(- Q_{ii}^{-1}(\Theta_{i,X_i}^T + G_{i,X}^{eq^T}\Theta_{i,I_i}^T)\big) + G_{i,U_i}^{eq^T}\big(- R_{ii}^{-1}(\Theta_{i,U_i}^T + G_{i,U_i}^{eq^T}\Theta_{i,I_i}^T)\big) \Bigg] \tag{2.50b}$$

$$\Xi = \sum_{i=1}^{m} \Big[0, 0, -g_i^{eq}\Big]^T. \tag{2.50c}$$

Using the triangular inequality $||a|| - ||b|| \leq ||a + b||$, (2.50a) yields:

$$||\Pi|| \times ||p^s|| - ||\Xi|| \leq ||\Pi \times p^s + \Xi|| \leq \epsilon \tag{2.51a}$$

or:

$$||p^s|| \leq d_{p^s} = \frac{\epsilon + ||\Xi||}{||\Pi||}. \tag{2.51b}$$

Therefore:

$$||X_i^s|| \leq d_{X_i^s} = ||Q_{ii}^{-1}(\Theta_{i,X_i}^T + G_{i,X}^{eq^T}\Theta_{i,I_i}^T)|| \times dp^s \tag{2.52a}$$

$$||U_i^s|| \leq d_{U_i^s} = ||R_{ii}^{-1}(\Theta_{i,U_i}^T + G_{i,U_i}^{eq^T}\Theta_{i,I_i}^T)|| \times dp^s \tag{2.52b}$$

$$||V_i^s|| \leq d_{V_i^s} = ||G_{i,X_i}|| \times ||X_i^s|| + ||G_{i,U_i}|| \times ||U_i^s|| + ||g_i^{eq}|| \tag{2.52c}$$

or, $Z_i^s(k) = [{X_i^s}^T(k), {U_i^s}^T(k), {V_i^s}^T(k)]^T$ is bounded. This bound is given by:

$$\big|\big|Z_i^s(k)\big|\big| \leq d_{Z_i^s} = \sqrt{d_{X_i^s}^2 + d_{U_i^s}^2 + d_{V_i^s}^2} \tag{2.53}$$

In other words, the upper bound of the nested vector of control actions $U_D = [{U_1^s}^T, \cdots, {U_m^s}^T]$ derived from CDMPC can be stated as function of ϵ based on (2.51b) and (2.52b),

namely:

$$||U_D(k)|| \leq d_{U_D}(\epsilon) = \sqrt{\sum_{i=1}^{m} d_{U_{D,i}}^2(\epsilon)} \tag{2.54a}$$

where:

$$d_{U_{D,i}}(\epsilon) = ||R_{ii}^{-1}(\Theta_{i,U_i}^T + G_{i,U_i}^{eq^T}\Theta_{i,I_i}^T)|| \times \frac{\epsilon + ||\Xi||}{||\Pi||}. \tag{2.54b}$$

Note that, setting $\epsilon = 0$ in (2.51b) is equivalent to an upper bound on the control action at time instant 'k', corresponding to $OTSLC(k)$, using (2.52b), if $\{Z_i^s(k)\}$ is a convergent subsequence to $OTSLC(k)$:

$$||U_C(k)|| \leq d_{U_C} = \sqrt{\sum_{i=1}^{m} d_{U_{D,i}}^2(0)} \tag{2.55a}$$

or:

$$||U_C(k)|| \leq d_{U_C} = d_{U_D}(0) \tag{2.55b}$$

Finally, if the CDMPC algorithm stops prematurely, namely at some $\epsilon \geq 0$, the upper bound on the norm of difference between the nested vector of control actions derived from this algorithm, $U_D(k)$, and the optimal $U_C(k)$ corresponding to $OTSLC(k)$, will be bounded by:

$$||U_C(k) - U_D(k)|| \leq ||U_C(k)|| + ||U_D(k)|| \leq d_{U_{C,D}}(\epsilon) \tag{2.56a}$$

where:

$$d_{U_{C,D}}(\epsilon) = d_{U_D}(\epsilon) + d_{U_D}(0) \tag{2.56b}$$

■

The following assumption is required to find a bound on the deviation of the exact state trajectory w.r.t. the difference between $OTSLC(k)$ and the sub-optimal CDMPC solution, derived from the approximate state trajectory.

Assumption 7 *Both F_h^E and F_h^A are Lipschitz functions.*

Based on Assumption 7, the absolute deviation between the two approximate trajectories of centralized and CDMPC can be derived as in Proposition 8.

Proposition 8 *Denote x_C the trajectory of the system under the control of the centralized MPC, and denote x_D the trajectory of the system under the control of the proposed CDMPC. Consider the following state trajectories:*

$$x_C(k+1) = F_h^A(x_C(k), u_C(k)) \tag{2.57a}$$

$$x_D(k+1) = F_h^A(x_D(k), u_D(k)) \tag{2.57b}$$

with states $\Big(x_C(k) = x_D(k), x_C(k+1), x_D(k+1)\Big) \in \varpi_\rho$, and there exists some non-zero γ such that: $\lim_{h\to 0}\gamma = 0$. Then $e^E(k+1) = F_h^E(x_C(k), u_C(k)) - F_h^E(x_D(k), u_D(k))$ is bounded:

$$||e^E(k+1)|| \leq d^A_{X_{C,D}}(k+1) \tag{2.58}$$

Proof. Define the error vector at time k as $e(k) = x_C(k) - x_D(k)$. According to Assumption 1:

$$F_h^E(x_C, u_C) \subseteq F_h^A(x_C, u_C) + \chi(T) \tag{2.59a}$$

$$F_h^E(x_D, u_D) \subseteq F_h^A(x_D, u_D) + \chi(T) \tag{2.59b}$$

differentiating (2.59b) from (2.59a) results the following rule for the sets e^E and e^A:

$$e^E \subseteq e^A \tag{2.59c}$$

which implies that there is an upper bound on the difference between e^E and e^A at any specific time, i.e. $k+1$:

$$||e^E(k+1) - e^A(k+1)|| \leq \gamma \tag{2.59d}$$

Using the triangular inequality $||a|| - ||b|| \leq ||a+b||$ yields:

$$||e^E(k+1)|| - ||e^A(k+1)|| \leq ||e^E(k+1) - e^A(k+1)|| \leq \gamma \tag{2.59e}$$

or

$$||e^E(k+1)|| \leq ||e^A(k+1)|| + \gamma \tag{2.59f}$$

Since F_h^A is a Lipschitz function, i.e. $\left|\left|F_h^A\big(x'(k),u'(k)\big)-F_h^A\big(x(k),u(k)\big)\right|\right| \leq L_X^A||x'(k)-x(k)|| + L_U^A||u'(k)-u(k)||$, an upper bound on $e^A(k+1)$ [5] can be derived as:

$$||e^A(k+1)|| \leq L_X^A||e^A(k)|| + L_U^A d_{U_{C,D}}(k) \tag{2.61}$$

Hence, with $||e^A(k)||$ being zero:

$$||e^E(k+1)|| \leq d^A_{X_{C,D}}(k+1) \tag{2.62a}$$

where:

$$d^A_{X_{C,D}}(k+1) = L_U^A d_{U_{C,D}}(k) + \gamma. \tag{2.62b}$$

■

Theorem 9 *Consider the successively linearized process model (2.7) is controlled by the CDMPC algorithm (2.28). Assume that the following conditions are satisfied:*

C1: all of the sub-systems are open-loop stable;

C2: matrices Q_{ii} and R_{ii} are constant positive definite and fixed;

C3: Assumptions 1 and 3 hold;

C4: Propositions 6 and 8 apply;

C5: Consider $\Big(x_C(k)=x_D(k), x_C(k+1), x_D(k+1)\Big) \in \varpi_\rho$;

Then, the entire closed-loop system is locally practically stable.

[5] The error vector of the approximate discrete-time system can be defined as:

$$e^A(k+1) = x_C(k+1) - x_D(k+1) \tag{2.60a}$$
$$= F_h^A(x_C(k),u_C(k)) - F_h^A(x_D(k),u_D(k)) \tag{2.60b}$$

Proof. Consider the Lyapunov function of the closed-loop system under the centralized controller (2.5) $V_h(x)$ along the trajectory of the successively linearized process system (2.7) under the CDMPC controller (2.28) at two consecutive sampling periods 'k' and 'k+1', namely $V_h(x_D(k))$ and $V_h(x_D(k+1))$. In order to show that this Lyapunov function is decreasing along the CDMPC algorithm, it is required to verify that $V_h(x_D(k+1)) - V_h(x_D(k)) \leq 0$. Adding and subtracting $\Big[V_h(x_C(k+1)) - V_h(x_C(k))\Big]$ to the left hand side would result in:

$$V_h(x_D(k+1)) - V_h(x_D(k)) = V_h(x_D(k+1)) - V_h(x_D(k)) + \Big[V_h(x_C(k+1)) - V_h(x_C(k))\Big] - \Big[V_h(x_C(k+1)) - V_h(x_C(k))\Big] \tag{2.63}$$

with $x_C(k)$ being equal to $x_D(k)$, $\Big(V_h(x_C(k) - V_h(x_D(k)\Big)$ vanishes and (2.63) becomes:

$$V_h(x_D(k+1)) - V_h(x_D(k)) = \Big[V_h(x_C(k+1)) - V_h(x_C(k))\Big] - \Big[V_h(x_C(k+1)) - V_h(x_D(k+1))\Big] \tag{2.64}$$

According to Assumption 3: property (2.13b), equation (2.64) can be written as follows:

$$V_h(x_D(k+1)) - V_h(x_D(k)) \leq -T\alpha_3(||x||) - \Big[V_h(x_C(k+1)) - V_h(x_D(k+1))\Big] \tag{2.65}$$

From Proposition 8, $||x_C(k+1) - x_D(k+1)||$ is bounded, namely $||x_C(k+1) - x_D(k+1)|| \leq d^A_{xC,D}(k+1)$. Thus, based on property (2.13c) of Assumption 3, there exists $h^* > 0$ such that: for $h \in (0, h^*)$ and $x_C(k+1), x_D(k+1) \in \mathcal{B}_D - \mathcal{B}_{\sigma_2}$, the following holds:

$$\Big|\Big|V_h(x_C(k+1)) - V_h(x_D(k+1))\Big|\Big| \leq \sigma_1 \tag{2.66}$$

which implies:

$$V_h(x_D(k+1)) - V_h(x_D(k)) \leq -T\alpha_3(||x_D(k)||) + \sigma_1. \tag{2.67}$$

Let $\alpha_3(||x||) = c||x_D(k)||^2$, for strictly positive real numbers (c, T^*) and $T \in (0, T^*)$, such that:

$$T\frac{c}{2}||x_D(k)||^2 \geq \sigma_1; \tag{2.68}$$

Therefore, the following holds:

$$V_h(x_D(k+1)) - V_h(x_D(k)) \leq -T\frac{c}{2}||x_D(k)||^2 \tag{2.69}$$

which guarantees that $V_h(x_D(k+1)) \leq V_h(x_D(k))$. Due to the fact that V_h is decreasing along x_D, the Lyapunov function of the centralized closed-loop system can be considered as the Lyapunov function of the CDMPC closed-loop system (2.6); thus, according to Lemma 4, the entire closed-loop system (2.5) is locally practically stable around the origin. ■

2.4 Alkylation Process Case Study

The proposed CDMPC is applied to a benzene alkylation example [22]. The process consists of four continuous stirred tank reactors (CSTRs) and a flash tank separator, as shown in Figure 2.2. Pure benzene is fed via stream F_1 and pure ethylene is fed via streams F_2, F_4 and F_6. Two catalytic reactions take place in CSTR-1, CSTR-2 and CSTR-3. In reaction 1, benzene (A) reacts with ethylene (B) and produces the desired product ethylbenzene (C); in reaction 2, ethylbenzene reacts with ethylene to form a

byproduct, $1,3$-diethylbenzene (D). In the flash tank separator, most of benzene is separated, flows overhead and recycled back to the plant. A portion of the recycle stream F_{r1} is fed to CSTR-4 and the other portion F_{r2} is fed back to CSTR-1 together with an additional feed stream F_{10}, which contains $1,3$-diethylbenzene. Moreover, reaction 2 and reaction 3, in which $1,3$-diethylbenzene reacts with benzene to produce ethylbenzene, take place in CSTR-4. Finally, it is assumed that throughout the process, all the materials are in liquid phase and their molar volumes are constant [22]. The nonlinear state-space model of the process is described in equations (2.70)-(2.74).

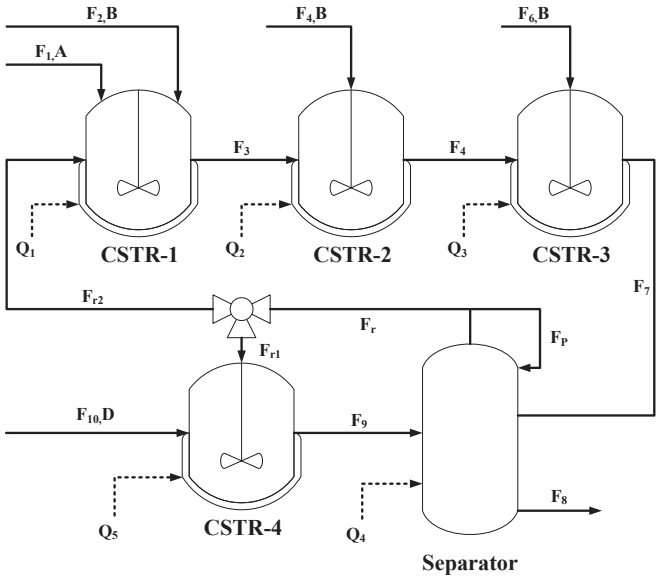


Figure 2.2: Process flow diagram of alkylation of benzene

CSTR-1:

$$\frac{dC_{A1}}{dt} = \frac{F_1 C_{A0} + F_{r2} C_{Ar} - F_3 C_{A1}}{V_1} - r_1(T_1, C_{A1}, C_{B1}) \tag{2.70a}$$

$$\frac{dC_{B1}}{dt} = \frac{F_2 C_{B0} + F_{r2} C_{Br} - F_3 C_{B1}}{V_1} - r_1(T_1, C_{A1}, C_{B1}) - r_2(T_1, C_{A1}, C_{B1}) \tag{2.70b}$$

$$\frac{dC_{C1}}{dt} = \frac{F_{r2} C_{Cr} - F_3 C_{C1}}{V_1} + r_1(T_1, C_{A1}, C_{B1}) - r_2(T_1, C_{B1}, C_{C1}) \tag{2.70c}$$

$$\frac{dC_{D1}}{dt} = \frac{F_{r2} C_{Dr} - F_3 C_{D1}}{V_1} + r_2(T_1, C_{B1}, C_{C1}) \tag{2.70d}$$

$$\frac{dT_1}{dt} = \frac{Q_1 + F_1 C_{A0} H_A(T_{A0}) + F_2 C_{B0} H_B(T_{B0})}{\Sigma_{i=A}^{D} C_{i1} C_{pi} V_1} + \frac{\Sigma_{i=A}^{D}\big(F_{r2} C_{ir} H_i(T_4) - F_3 C_{i1} H_i(T_1)\big)}{\Sigma_{i=A}^{D} C_{i1} C_{pi} V_1} - \frac{\Delta H_{r1} r_1(T_1, C_{A1}, C_{B1})}{\Sigma_{i=A}^{D} C_{i1} C_{pi}} - \frac{\Delta H_{r2} r_2(T_1, C_{B1}, C_{C1})}{\Sigma_{i=A}^{D} C_{i1} C_{pi}} \tag{2.70e}$$

CSTR-2:

$$\frac{dC_{A2}}{dt} = \frac{F_3 C_{A1} + F_5 C_{A2}}{V_2} - r_1(T_2, C_{A2}, C_{B2}) \tag{2.71a}$$

$$\frac{dC_{B2}}{dt} = \frac{F_3 C_{B1} + F_4 C_{B0} - F_5 C_{B2}}{V_2} - r_1(T_2, C_{A2}, C_{B2}) - r_2(T_2, C_{B2}, C_{C2}) \tag{2.71b}$$

$$\frac{dC_{C2}}{dt} = \frac{F_3 C_{C1} - F_5 C_{C2}}{V_2} + r_1(T_2, C_{A2}, C_{B2}) - r_2(T_2, C_{B2}, C_{C2}) \tag{2.71c}$$

$$\frac{dC_{D2}}{dt} = \frac{F_3 C_{D1} - F_5 C_{CR2}}{V_2} + r_2(T_2, C_{B2}, C_{C2}) \tag{2.71d}$$

$$\frac{dT_2}{dt} = \frac{Q_2 + F_4 C_{B0} H_B(T_{B0})}{\Sigma_{i=A}^{D} C_{i2} C_{pi} V_2} + \frac{\Sigma_{i=A}^{D}\big(F_3 C_{i1} H_i(T_1) - F_5 C_{i2} H_i(T_2)\big)}{\Sigma_{i=A}^{D} C_{i2} C_{pi} V_2} - \frac{\Delta H_{r1} r_1(T_2, C_{A2}, C_{B2})}{\Sigma_{i=A}^{D} C_{i2} C_{pi}} - \frac{\Delta H_{r2} r_2(T_2, C_{A2}, C_{B2})}{\Sigma_{i=A}^{D} C_{i2} C_{pi}} \tag{2.71e}$$

CSTR-3:

$$\frac{dC_{A3}}{dt} = \frac{F_5 C_{A2} + F_7 C_{A3}}{V_3} - r_1(T_3, C_{A3}, C_{B3}) \tag{2.72a}$$

$$\frac{dC_{B3}}{dt} = \frac{F_5 C_{B2} + F_6 C_{B0} - F_7 C_{B3}}{V_3} - r_1(T_3, C_{A3}, C_{B3}) - r_2(T_3, C_{B3}, C_{C3}) \tag{2.72b}$$

$$\frac{dC_{C3}}{dt} = \frac{F_5 C_{C2} - F_7 C_{C3}}{V_3} + r_1(T_3, C_{A3}, C_{B3}) - r_2(T_3, C_{B3}, C_{C3}) \tag{2.72c}$$

$$\frac{dC_{D3}}{dt} = \frac{F_5 C_{D2} - F_7 C_{D3}}{V_3} + r_2(T_3, C_{B3}, C_{C3}) \tag{2.72d}$$

$$\frac{dT_3}{dt} = \frac{Q_3 + F_6 C_{B0} H_B(T_{B0})}{\Sigma_{i=A}^{D} C_{i3} C_{pi} V_3} + \frac{\Sigma_{i=A}^{D}\big(F_5 C_{i2} H_i(T_2) - F_7 C_{i3} H_i(T_3)\big)}{\Sigma_{i=A}^{D} C_{i3} C_{pi} V_3} - \frac{\Delta H_{r1} r_1(T_3, C_{A3}, C_{B3})}{\Sigma_{i=A}^{D} C_{i3} C_{pi}} - \frac{\Delta H_{r2} r_2(T_3, C_{B3}, C_{C3})}{\Sigma_{i=A}^{D} C_{i3} C_{pi}} \tag{2.72e}$$

Separator:

$$\frac{dC_{A4}}{dt} = \frac{F_7 C_{A3} + F_9 C_{A5} - F_r C_{Ar} - F_8 C_{A4}}{V_4} \tag{2.73a}$$

$$\frac{dC_{B4}}{dt} = \frac{F_7 C_{B3} + F_9 C_{B5} - F_r C_{Br} - F_8 C_{B4}}{V_4} \tag{2.73b}$$

$$\frac{dC_{C4}}{dt} = \frac{F_7 C_{C3} - F_9 C_{C5} - F_r C_{Cr} - F_8 C_{C4}}{V_4} \tag{2.73c}$$

$$\frac{dC_{D4}}{dt} = \frac{F_7 C_{D3} - F_9 C_{D5} - F_r C_{Dr} - F_8 C_{D4}}{V_4} \tag{2.73d}$$

$$\begin{aligned}\frac{dT_4}{dt} =& \frac{Q_4 + \Sigma_{i=A}^{D}\big(F_7 C_{i3} H_i(T_3) + F_9 C_{i5} H_i(T_5)\big)}{\Sigma_{i=A}^{D} C_{i4} C_{pi} V_4} \\ &+ \frac{\Sigma_{i=A}^{D}\big(-M_i H_i(T_4) - F_8 C_{i4} H_i(T_4) - M_i H_{vap}\big)}{\Sigma_{i=A}^{D} C_{i4} C_{pi} V_4}\end{aligned} \tag{2.73e}$$

CSTR-4:

$$\frac{dC_{A5}}{dt} = \frac{F_{r1} C_{Ar} + F_9 C_{A5}}{V_5} - r_3(T_5, C_{A5}, C_{D5}) \tag{2.74a}$$

$$\frac{dC_{B5}}{dt} = \frac{F_{r1} C_{Br} + F_9 C_{B5}}{V_5} - r_2(T_3, C_{B5}, C_{C5}) \tag{2.74b}$$

$$\frac{dC_{C5}}{dt} = \frac{F_{r1} C_{Cr} - F_9 C_{C5}}{V_5} - r_2(T_5, C_{B5}, C_{C5} + 2r_3(T_5, C_{A5}, C_{D5}) \tag{2.74c}$$

$$\frac{dC_{D5}}{dt} = \frac{F_{r1} C_{Dr} - F_{10} C_{D0} - F_9 C_{D5}}{V_5} + r_2(T_5, C_{B5}, C_{C5}) - r_3(T_5, C_{A5}, C_{D5}) \tag{2.74d}$$

$$\begin{aligned}\frac{dT_5}{dt} =& \frac{Q_5 + F_{10} C_{D0} H_D(T_{D0})}{\Sigma_{i=A}^{D} C_{i5} C_{pi} V_5} + \frac{\Sigma_{i=A}^{D}\big(F_{r1} C_{ir} H_i(T_4) - F_9 C_{i5} H_i(T_5)\big)}{\Sigma_{i=A}^{D} C_{i5} C_{pi} V_5} \\ &- \frac{\Delta H_{r2} r_2(T_5, C_{B5}, C_{C5})}{\Sigma_{i=A}^{D} C_{i5} C_{pi}} - \frac{\Delta H_{r3} r_3(T_5, C_{A5}, C_{D5})}{\Sigma_{i=A}^{D} C_{i5} C_{pi}}\end{aligned} \tag{2.74e}$$

where r_1, r_2 and r_3 are reaction rates defined as:

$$r_1(T, C_A, CB) = 0.084 e^{\frac{-9502}{RT}} C_A^{0.32} C_B^{1.5} \tag{2.75a}$$

$$r_2(T, C_B, CC) = \frac{0.085 e^{\frac{-20643}{RT}} C_B^{2.5} C_C^{0.5}}{1 + K_{EB2} C_D} \tag{2.75b}$$

$$r_3(T, C_A, CD) = \frac{66.1 e^{\frac{-61280}{RT}} C_A^{1.0218} C_D}{1 + K_{EB3} C_A} \tag{2.75c}$$

where $K_{EB2} = 0.152 e^{\frac{-3933}{RT}}$ and $K_{EB3} = 0.490 e^{\frac{-50870}{RT}}$. The heat capacities $C_{pi=A,B,C,D}$ are assumed to be constant and the molar enthalpies have are linearly related to temperature:

$$H_i = H_{iref} + C_{pi}(T - T_{ref}). \tag{2.76}$$

Furthermore, the relative volatity of each species inside the flash tank separator is assumed to be a linear function of vessel temperature in the operating range of the flash tank:

$$\alpha_A = 0.0449 T_4 + 10 \tag{2.77a}$$

$$\alpha_B = 0.0260 T_4 + 10 \tag{2.77b}$$

$$\alpha_C = 0.0065 T_4 + 10 \tag{2.77c}$$

$$\alpha_D = 0.0058 T_4 + 10. \tag{2.77d}$$

The molar flow rates of the overhead reactants, $M_{i=A,B,C,D}$, can be defined as in equation (2.78):

$$M_i = k \frac{\alpha (F_7 C_{i5}) \Sigma_{j=A,B,C,D} (F_7 C_{j3} + F_9 C_{j5})}{\Sigma_{j=A,B,C,D} \alpha_j (F_7 C_{j3} + F_9 C_{j5})} \tag{2.78}$$

where k is the fraction of condensed overhead flow recycled to the reactors. Based on M_i, the concentration of the reactants in the recycle stream can be calculated:

$$C_{ir} = \frac{M_i}{\Sigma_{j=A,B,C,D} M_i / \rho_{j0}} \tag{2.79}$$

where $\rho_{j0=A,B,C,D}$ are the molar densities of pure reactants. The vapor is condensed in the overhead, and a portion of the condensed liquid is redirected back to separator to maintain the flow rate of the recycled stream at a fixed value. Also, the temperature of the condensed liquid is assumed to be the same as the temperature of the vessel. Furthermore, the definition and values of process parameters used in the process model are given in Tables 2.1 and 2.2 [22].

Table 2.1: Alkylation process parameters

Process Parameter	Description
C_{A1},C_{B1},C_{C1},C_{D1}	Concentration of species in CSTR-1
C_{A2},C_{B2},C_{C2},C_{D2}	Concentration of species in CSTR-2
C_{A3},C_{B3},C_{C3},C_{D3}	Concentration of species in CSTR-3
C_{A4},C_{B4},C_{C4},C_{D4}	Concentration of species in Separator
C_{A5},C_{B5},C_{C5},C_{D5}	Concentration of species in CSTR-4
C_{Ar},C_{Br},C_{Cr},C_{Dr}	Concentration of species in F_r, F_{r1}, F_{r2}
T_1,T_2,T_3,T_4,T_5	Temperature in each vessel
T_{ref}	Reference Temperature
F_3,F_5,F_7,F_8,F_9	Effluent flow rates from each vessel
F_1,F_2,F_4,F_6,F_{10}	Feed flow rates to each vessel
F_r, F_{r1}, F_{r2}	Recycle flow rates
H_{vapA},H_{vapB},H_{vapC},H_{vapD}	Enthalpies of vaporization
H_{Aref},H_{Bref},H_{Cref},H_{Dref}	Enthalpies of vaporization
ΔH_{r1},ΔH_{r2},ΔH_{r3}	Heat of reactions
V_1,V_2,V_3,V_4,V_5	Volume of each vessel
Q_1,Q_2,Q_3,Q_4,Q_5	External heat/coolant input to each vessel
C_{pA},C_{pA},C_{pA},C_{pA}	Heat capacity at liquid phase
α_A,α_B,α_C,α_D	Relative volatilities
$\rho A0$,$\rho B0$,$\rho C0$,$\rho D0$	Molar densities of pure species
T_{A0},T_{B0},T_{D0}	Feed temperature of pure species
k	Fraction of overhead flow recycled to the reactors

The steady-state values of manipulated input variables and state variables are shown in Tables 2.3 and 2.5 [22]. Moreover, the acceptable operating range of the manipulated variable moves are shown in Table 2.4 [22].

In the design of the proposed CDMPC, the process is divided into two subsystems. The first subsystem includes CSTR-1 and CSTR-2. The second subsystem includes CSTR-3, the separator and CSTR-4. Consequently, the following subsystem decomposi-

Table 2.2: Alkylation process parameter values

Parameter	Value	Parameter	Value
F_1	$7.1 \times 10^{-3}[m^3/s]$	F_r	$0.012[m^3/s]$
F_2	$8.697 \times 10^{-4}[m^3/s]$	F_{r1}	$0.006[m^3/s]$
F_{r21}	$0.006[m^3/s]$	V_1	$1[m^3]$
F_{10}	$2.31 \times 10^{-3}[m^3/s]$	V_2	$1[m^3]$
H_{vapA}	$3.073 \times 10^4[J/mole]$	V_3	$1[m^3]$
H_{vapB}	$1.35 \times 10^4[J/mole]$	V_4	$3[m^3]$
H_{vapC}	$4.226 \times 10^4[J/mole]$	V_5	$1[m^3]$
H_{vapD}	$4.55 \times 10^4[J/mole]$	C_{pA}	$184.6[J/moleK]$
H_{Aref}	$7.44 \times 10^4[J/mole]$	H_{Bref}	$5.91 \times 10^4[J/mole]$
H_{Cref}	$2.02 \times 10^4[J/mole]$	H_{Dref}	$-2.89 \times 10^4[J/mole]$
ΔH_{r1}	$-1.536 \times 10^5[J/mole]$	C_{pB}	$59.1[J/moleK]$
ΔH_{r2}	$-1.118 \times 10^5[J/mole]$	C_{pC}	$247[J/moleK]$
ΔH_{r3}	$4.141 \times 10^5[J/mole]$	C_{pD}	$301.3[J/moleK]$
C_{A0}	$1.126 \times 10^4[mole/m^3]$	T_{ref}	$450[K]$
C_{B0}	$2.028 \times 10^4[mole/m^3]$	T_{A0}	$473[K]$
C_{C0}	$8174[mole/m^3]$	T_{B0}	$473[K]$
C_{D0}	$6485[mole/m^3]$	T_{D0}	$473[K]$
k	0.8		

Table 2.3: Alkylation process steady-state values manipulated input variables

Input	Value	Input	Value
$Q_{1,s}$	$-4.4 \times 10^6[J/s]$	$Q_{2,s}$	$-4.6 \times 10^6[J/s]$
$Q_{3,s}$	$-4.7 \times 10^6[J/s]$	$Q_{4,s}$	$-9.2 \times 10^6[J/s]$
$Q_{5,s}$	$5.9 \times 10^6[J/s]$	$F_{4,s}$	$8.697 \times 10^{-4}[m^3/s]$
$F_{4,s}$	$8.697 \times 10^{-4}[m^3/s]$		

Table 2.4: The operating range of the manipulated variables

Input	Operating Range	Input	Operating Range
Q_1	$(\pm 7.5 \times 10^5 + Q_{1,s})[J/s]$	Q_2	$(\pm 5 \times 10^5 + Q_{2,s})[J/s]$
Q_3	$(\pm 5 \times 10^5 + Q_{3,s})[J/s]$	Q_4	$(\pm 6 \times 10^5 + Q_{4,s})[J/s]$
Q_5	$(\pm 5 \times 10^5 + Q_{5,s})[J/s]$	F_4	$(\pm 4.93 \times 10^{-5} + F_{4,s})[m^3/s]$
F_6	$(\pm 4.93 \times 10^{-5} + F_{6,s})[m^3/s]$		

tion of state and input variables is defined as:

$$\begin{cases} X_1 = [C_{A1}, C_{B1}, C_{C1}, C_{D1}, T_1, C_{A2}, C_{B2}, C_{C2}, C_{D2}, T_2] \\ U_1 = [Q_1, Q_2, F_4] \end{cases} \tag{2.80a}$$

Table 2.5: Alkylation process steady-state values of the state variables

State	Value	State	Value
C_{A1}	$9.101 \times 10^3[mole/m^3]$	C_{A2}	$7.548 \times 10^3[mole/m^3]$
C_{B1}	$22.15[mole/m^3]$	C_{B2}	$23.46[mole/m^3]$
C_{C1}	$1.120 \times 10^3[mole/m^3]$	C_{C2}	$1.908 \times 10^3[mole/m^3]$
C_{D1}	$2.120 \times 10^2[mole/m^3]$	C_{D2}	$3.731 \times 10^2[mole/m^3]$
T_1	$4.772 \times 10^2[K]$	T_2	$4.77 \times 10^3[K]$
C_{A3}	$6.163 \times 10^3[mole/m^3]$	C_{A4}	$1.723 \times 10^3[mole/m^3]$
C_{B3}	$24.84[mole/m^3]$	C_{B4}	$13.67[mole/m^3]$
C_{C3}	$2.616 \times 10^3[mole/m^3]$	C_{C4}	$5.473 \times 10^3[mole/m^3]$
C_{D3}	$5.058 \times 10^2[mole/m^3]$	C_{D4}	$7.044 \times 10^2[mole/m^3]$
T_3	$4.735 \times 10^2[K]$	T_4	$4.706 \times 10^2[K]$
C_{A5}	$5.747 \times 10^3[mole/m^3]$	C_{D5}	$1.537 \times 10^2[mole/m^3]$
C_{B1}	$3.995[mole/m^3]$	T_5	$4.783 \times 10^2[K]$
C_{C5}	$3.830 \times 10^3[mole/m^3]$		

and:

$$\begin{cases} X_2 = & [C_{A3}, C_{B3}, C_{C3}, C_{D3}, T_3, C_{A4}, C_{B4}, C_{C4}, C_{D4}, T_4, C_{A5}, C_{B5}, C_{C5}, C_{D5}, T_5] \\ U_2 = & [Q_3, Q_4, Q_5, F_6] \end{cases} \tag{2.80b}$$

In order to have a high performance the closed-loop system, without the loss of generality, the prediction horizon H_p and the control horizon H_u are taken to be 15 times the sampling period. The integration time is $h = 0.3[sec]$ and the sampling period is $T = 30[sec]$. In order to improve solver performance, the decision variables of optimization are scaled to be between 0.1 and 1. The weighting matrices Q and R are $10^2 \times I_{25}$ and $10^{-4} \times I_7$, respectively[6]. The initial conditions of this system are shown in Table 2.6 [22].

Similar to the CDMPC and the centralized schemes, the decentralized control actions are also applied to the continuous time plant. This scheme ignores the interactions between subsystems, and the overall objective function of the decentralized controlled

[6]Here, 'I' represents an identity matrix of appropriate dimension

Table 2.6: Alkylation process Initial values of the state variables

State	Value	State	Value
C_{A1}	$9.112 \times 10^3[mole/m^3]$	C_{A2}	$7.557 \times 10^3[mole/m^3]$
C_{B1}	$25.09[mole/m^3]$	C_{B2}	$27.16[mole/m^3]$
C_{C1}	$1.113 \times 10^3[mole/m^3]$	C_{C2}	$1.905 \times 10^3[mole/m^3]$
C_{D1}	$2.186 \times 10^2[mole/m^3]$	C_{D2}	$3.695 \times 10^2[mole/m^3]$
T_1	$4.430 \times 10^2[K]$	T_2	$4.371 \times 10^3[K]$
C_{A3}	$6.170 \times 10^3[mole/m^3]$	C_{A4}	$1.800 \times 10^3[mole/m^3]$
C_{B3}	$29.45[mole/m^3]$	C_{B4}	$16.35[mole/m^3]$
C_{C3}	$2.617 \times 10^3[mole/m^3]$	C_{C4}	$5.321 \times 10^3[mole/m^3]$
C_{D3}	$5.001 \times 10^2[mole/m^3]$	C_{D4}	$7.790 \times 10^2[mole/m^3]$
T_3	$4.284 \times 10^2[K]$	T_4	$4.331 \times 10^2[K]$
C_{A5}	$5.889 \times 10^3[mole/m^3]$	C_{D5}	$2.790 \times 10^2[mole/m^3]$
C_{B1}	$5.733[mole/m^3]$	T_5	$4.576 \times 10^2[K]$
C_{C5}	$3.566 \times 10^3[mole/m^3]$		

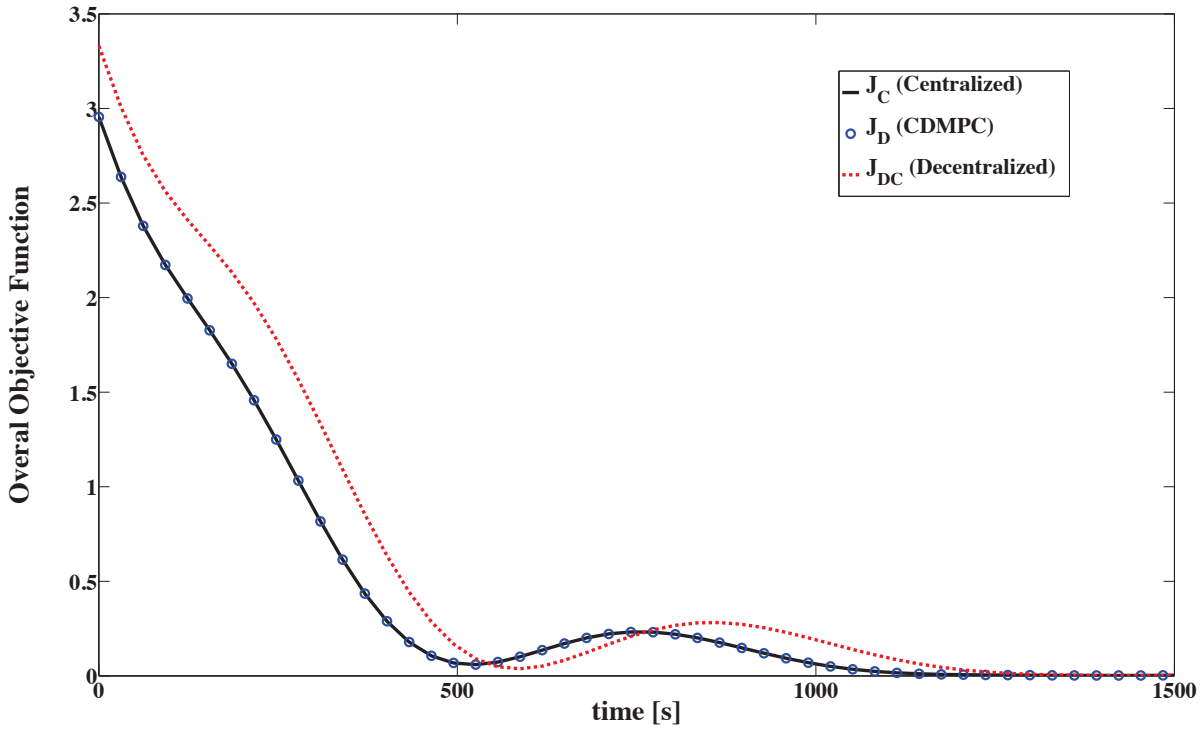


Figure 2.3: Comparison of the overall objective function for the centralized, coordinated and decentralized control schemes, based on scaled decision variables

network J_{DC} is defined as:

$$\min_{X,U} \; J_{DC} = \sum_{i=1}^{m} J_{DC_i} \tag{2.81a}$$

$$\text{s.t. } \hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k)$$

$$\text{for } l = 0, \ldots, H_p - 1 \tag{2.81b}$$

$$\hat{u}_i(k+l|k) = \hat{u}_i(H_u|k)$$

$$\text{for } H_u \leq l \leq H_p - 1 \tag{2.81c}$$

where:

$$J_{DC_i} = \frac{1}{2}\Big((X_i(k))^T Q_{ii}(X_i(k)) + U_i(k)^T R_{ii} U_i(k)\Big) + \Omega_{U_i}(U(k)) \tag{2.81d}$$

The simulation studies in this section were performed on a **2.3 GHz Intel® Core-i7 3610QM** processor with **8 GB** of RAM. For the numerical calculations, **MATLAB ® 2012a** environment is used. In Figure 2.3, the trends of the overall objective functions[7] of the CDMPC design J_D and decentralized J_{DC} are compared to the centralized objective function J_C. Note that these results are obtained via applying the sampled-data control actions to the actual continuous plant. As can be seen, CDMPC can efficiently track $OTSLC(k)$. The result obtained by decentralized MPC shows that the decentralized MPC is not able to achieve the performance of $OTSLC(k)$, i.e. the mean square error between the decentralized MPC and the centralized MPC scaled objective functions is at least two orders of magnitude less than that of between the CDMPC and the centralized MPC, over the whole simulation time.

In addition, temperature profiles of CSTR-1, CSTR-2, CSTR-3, and CSTR-4 in Figure 2.4 show that the state trajectories obtained form the CDMPC algorithm track the

[7]This plot is based on the scaled decision variables.

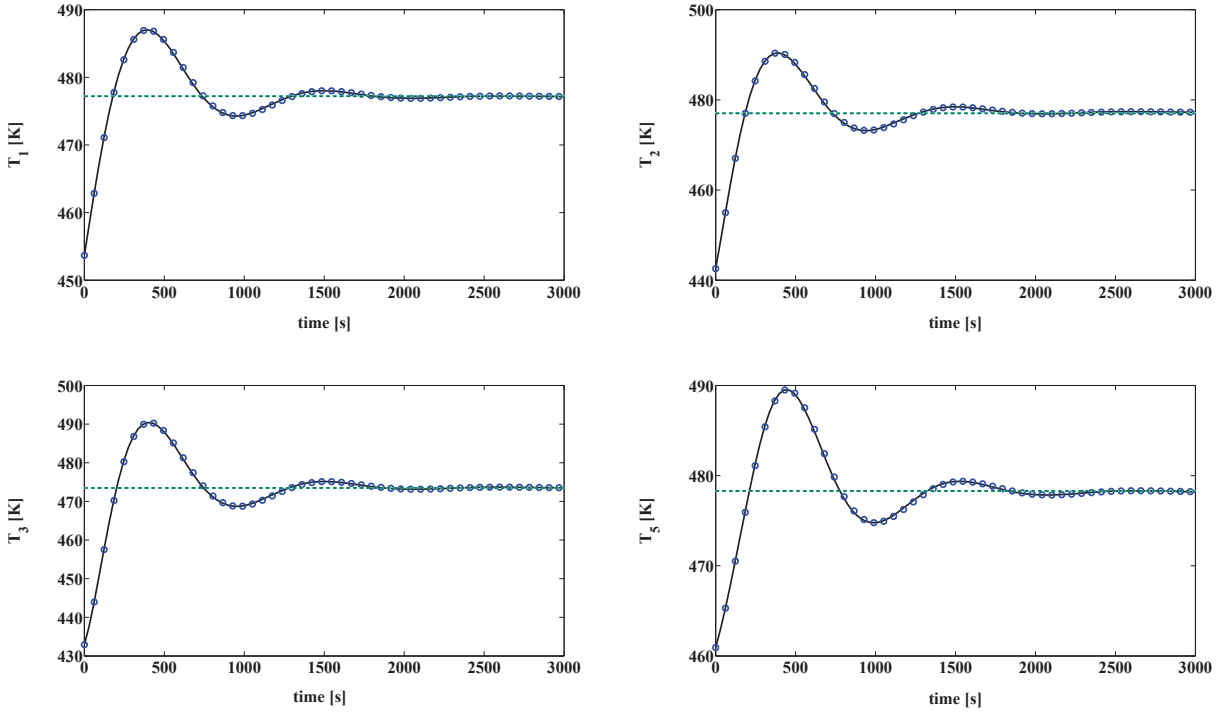


Figure 2.4: Temperature profiles in CSTR-1, CSTR-2, CSTR-3, and CSTR-4; solid lines (—): centralized scheme, circles (∘): CDMPC scheme, dotted lines (· · ·): steady-state values

$OTSLC(k)$ perfectly. Control action inputs of sub-systems 1 and 2 are also shown in Figures 2.5 and 2.6, respectively. These also demonstrate that the CDMPC is capable of achieving the optimum centralized input trajectories, corresponding to $OTSLC(k)$, entirely.

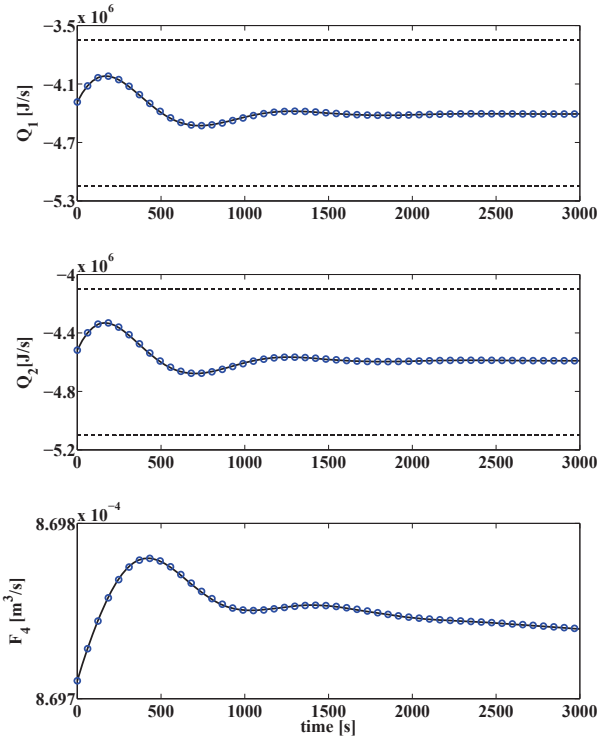


Figure 2.5: Control action inputs of sub-system 1: CSTR-1 and CSTR-2; solid lines (—): centralized scheme, circles (∘): CDMPC scheme, dotted lines (· · ·): upper and lower bounds

2.5 Conclusion

In this chapter, a nested bilevel optimization approach for price-driven coordination of distributed MPC networks is proposed for nonlinear systems. Sufficient conditions for the stability of CDMPC are derived for premature convergence of the algorithm. In the proposed scheme, an interior-point approach is presented to coordinate the actions of the distributed controllers based on the price-driven coordination algorithm. Assuming that a centralized MPC based on successive linearization can be designed to stabilize the entire system, sufficient conditions were derived which ensures that the proposed CDMPC guarantees the closed-loop stability, even when the algorithm is stopped prematurely, i.e. prior to convergence.

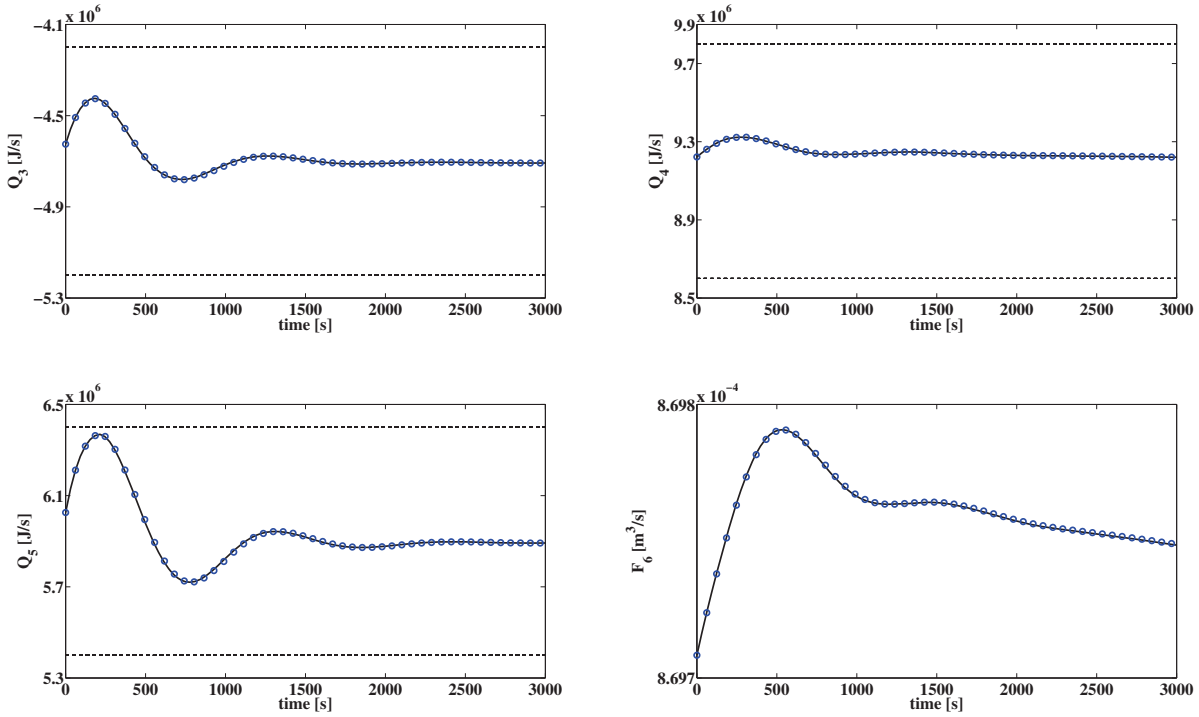


Figure 2.6: Control action inputs of sub-system 2: CSTR-3, the separator and CSTR-4; solid lines: centralized scheme, circles: CDMPC scheme, dotted lines: upper and lower bounds

From the simulation studies, it is demonstrated that the proposed CDMPC is able to track the performance of $OTSLC(k)$, nevertheless premature termination may introduce noticeably small deviations from the optimal trajectory. This scheme is limited to nonlinear systems that are open-loop stable and limitations of the successive linearization method. In the next chapter, the idea is extended to stabilize open-loop unstable dynamics. A novel bilevel formulation is proposed to coordinate distributed nonlinear model predictive control systems, without the limitations imposed by successive linearization.

Chapter 3

A Trust-Region Approach to Price-Driven Coordination of Networked Nonlinear Model Predictive Control Systems

In this chapter, an on-line solution strategy is presented to improve the plant-wide performance of an existing network of decentralized nonlinear model predictive controllers (NMPC). The nonlinear plant is assumed to be governed by differential-algebraic equations (DAEs) with separable interconnections in the form of equality constraints. For this reason, the problem of designing a price-driven coordination level for the network, to account for interactions between the distributed subsystems, is addressed. Nevertheless, the coordination scheme introduces minor modifications to the distributed controllers. In this work, a novel coordinated distributed NMPC (CDNMPC) algorithm is proposed. Note that, the price-driven CDNMPC scheme can be thought of as a bi-level nonlinear programming (BLNP) problem in which the coordinator is represented in the upper level and the subsystem NMPC network belongs to the lower level. In this technique, the coordinator exchanges information with the distributed control network to solve a transformed version of the bi-level optimization problem, namely a relaxed single level problem. The optimal plant-wide trajectory is found using a trust-region method. Ac-

cordingly, global convergence of the this algorithm, to a unique optimal solution of the BLNP problem, is ensured. Additionally, an adaptive prediction horizon module is implemented to find the minimum required horizon length to ensure stability of the finite-time dynamic optimization problem. The proposed method can be applied to nonlinear plants with open-loop unstable behavior. The effectiveness of the proposed CDNMPC algorithm is illustrated via the application to a two-CSTR process.

3.1 Preliminaries

Industrial chemical and petrochemical plants are typically complex, highly integrated nonlinear systems composed of geographically distributed processing units that share material and energy streams. It is a commonplace that optimal control methods such as model predictive controllers (MPCs) are widely used to ensure efficient operation. In general, performance and reliability are two key factors that determine the control strategy in the network. Traditionally, decentralized control systems are used to operate the plant, mainly due to the ease of implementation; however, such controlled networks ignore interactions between the subsystems and can result in degradation of plant-wide performance [86, 8, 96]. Alternatively, monolithic centralized controllers can take into account all of the interactions to obtain the highest performance for the entire plant. The main drawbacks for centralized control are capital cost due to replacement of the whole network with one controller, increased maintenance requirements, and poor fault tolerance [63, 77]. As a result, a considerable body of research is developed for distributed MPC (DMPC) strategies such as cooperative and coordinated methods, as an alternative to centralized MPC.

Cooperative distributed schemes achieve an improved plant-wide performance while preserving the flexibility of decentralized control. Specifically, in cooperative DMPC

schemes, local controllers communicate over the network to achieve a higher performance [15, 84, 88, 23, 22]. Most important cooperative schemes in the literature are robust implementation of DMPC for linear systems [3], DMPC of linear systems coupled through the inputs [64] and dissipativity-based DMPC of linear systems [98]. On the other hand, coordinated distributed control belongs to class of hierarchical multilevel systems [71, 72, 36], and are composed of two levels: local controllers and a coordinator. There is no communication between the local controllers, instead they communicate with the coordinator to achieve the improved plant-wide performance. Based on the way that interactions are addressed, different algorithms have been proposed for coordination of linear systems, including the price-driven approach (e.g., [20, 21, 67, 76, 57, 68]), primal decomposition DMPC [35, 25], and prediction-driven DMPC (e.g., [66, 76]).

In this chapter, the problem of designing CDNMPC scheme for nonlinear systems is addressed. To date, only a limited amount of research is dedicated to design CDNMPC systems. In [76], a pseudo-model coordination method is proposed via exact linearization of interconnected ODE systems. Coordination of nonlinear networks governed by open-loop stable DAE systems was investigated by [89] and [68]. In [89], price driven coordination based on iterative steepest ascent was shown exhibit poor convergence to the optimal plant-wide trajectory. They also proposed an off-line dynamic optimization method, where a sensitivity based interaction driven coordination is obtained by collocation methods. Moreover, another coordination approach was proposed for systems that share a common resource [68]. In this scheme, the coordinator acts as a single input single output (SISO) controller and is fed the error between the summation of resource demands by the local NMPCs and the total available shared resource in the network. In other words, the set point is the maximum availability of the shared resource and the output is the price vector.

The main scope of this study is to design a price-driven CDNMPC for on-line coor-

dination of nonlinear systems governed by DAEs, which may exhibit open-loop unstable dynamics. In addition, it is assumed the interconnections between the subsystems are separable and in the form of equality constraints. The method proposed by [76] is limited to exact linearization of ODEs and may fail to stabilize an unstable system for small prediction horizons. The interaction driven method proposed by [89] is an off-line dynamic optimization and is mainly developed for open-loop stable DAE systems. The scheme proposed by [68] is only applicable to open-loop stable DAE systems with shared resources.

In this work, a bi-level optimization approach to design a novel CDNMPC algorithm is proposed. It is shown that the proposed price-driven coordination is the hierarchical dual decomposition of a hypothetical centralized control problem. The structure can be thought of as a non-linear bi-level programming (NBLP) problem, in which the coordinator is the upper level decision maker and the local controllers belong to the lower level problem. In this design, a minor modification is applied to the existing decentralized NMPC controllers. The modification required in the subsystems is equivalent to relaxing local versions of the overall interaction constraint, and assigning a price vector to penalize the local violations of predicted interaction errors into the objective function of NMPCs. Methods for solving any NBLP may belong to one of four categories: enumeration (branch-and-bound), descent, evolutionary based, and penalty methods.

In the branch-and-bound approach, the lower-level problem is replaced by the equivalent Karush Kuhn-Tucker (KKT) system. This is the idea underlying the approaches of Edmunds and Bard [32], Al-Khayyal et al. [4] and more recently, Thoai et al. [97]. In [43], NBLP problems are addressed that involve twice differentiable nonlinear functions and the linear independence constraint qualification condition holds for their inner problem constraints. This approach gives a global solution based on the relaxation of the feasible region by convex underestimation. Global optimization of nonlinear BLP with

non-convex lower and upper problems and related dynamic optimization topics have been addressed by Mitsos et al [75, 74]. In evolutionary methods, the bi-level problem is transformed into a single level non-convex problem and then several evolutionary heuristic methods such as genetic algorithms [9] or particle swarm optimization are applied to find a global optimum.

Approaches based on descent method attempt to find local optimum of the NBLP problems. In this method, an implicit relation between the lower-level variable as a function of the upper-level variable is defined. The major issue is to find a decreasing feasible direction based on the gradient of the upper-level objective function. Several approximations of the gradient information are proposed by Kolstad and Lasdon [54] and Savard and Gauvin [87]. In this category, Falk and Liu [34] presented a bundle method where the decrease of the upper-level objective is adapted according to subgradient information of the lower-level problem.

The penalty method category also attempts to compute stationary point and local optimal solutions to NBLP problems [1, 2, 92, 24]. In [51] a double penalty method is deployed in both objective functions of upper and lower level problems, and the BLP problem is transformed into a single level optimization by replacing the lower-level problem with its corresponding stationary conditions. Case [16] transformed the nonlinear BLP into a single level optimization problem via replacing the lower-level optimization with the corresponding KKT conditions. The resulting problem is then solved with exact l_1-norm penalty methods and quadratic approximate models subject to trust-region constraints. More recently, Lin et al. [59] proposed a smoothing projected gradient algorithm to a BLP problem in which the lower level program is a non-convex minimization problem with a convex set constraint and the upper level program has a convex set constraint. Jiang et al. [52] proposed an augmented Lagrangian method to nonlinear BLP problems that are transformed to single level optimization problem via KKT conditions

and their complementarity constraints have been smoothed via Chen Harker Kanzow Smale (CHKS) smoothing function [33].

In this chapter, an exact penalty method is used to solve the CDNMPC problem. This BLNP problem is relaxed into a single level problem and solved using a trust-region approach to obtain the local optimal plant-wide trajectory, in an iterative procedure between the coordinator and subsystem controllers. Global convergence of the algorithm to unique local optimal solution is investigated under certain assumptions. In addition, the coordinator and local NMPCs perform an outer iteration to adaptively adjust the prediction horizon length required to ensure stability for a predefined sub-optimality criteria. Finally, the performance of the proposed CDNMPC is illustrated via an application to an unstable two-CSTR process and comparison to decentralized and centralized control schemes. This case study exhibits unstable behavior around one of its steady-state operating points and the control objective is to stabilize the system around this operating condition.

3.1.1 System Description

In this work, a class of nonlinear systems composed of m interconnected open-loop subsystems is considered. Each of the subsystems can be described by the following DAE model:

$$\dot{x}_i = f_i(x_i, x_{ai}, u_i, v_i) \tag{3.1a}$$

$$0 = g_i(x_i, x_{ai}, u_i) \tag{3.1b}$$

$$x_i(0) = x_{i,0} \tag{3.1c}$$

where $i = 1, \ldots, m$, $x_i(t) \in \mathbb{R}^{n_{x_i}}$ denotes the vector of state variables, $x_{ai}(t) \in \mathbb{R}^{n_{x_{ai}}}$ denotes the vector of auxiliary state variables[1], $u_i \in \mathbb{R}^{n_{u_i}}$ is the vector of control inputs, and $v_i \in \mathbb{R}^{n_{v_i}}$ denotes the vector of interaction variables associated with subsystem i. In this formulation, the interaction variable of sub-system 'i' is defined as:

$$v_i = \sum_{j \neq i} h_{i,j} \left[x_j^T, x_{aj}^T, u_j^T \right]^T \tag{3.2a}$$

where $h_{i,j}$ is a coefficient matrix that defines the relation between interaction variables of sub-system i with other sub-systems $j = 1, \cdots, i-1, i+1, \cdots, m$. In other words, it is assumed that the nonlinear interactions can be replaced by defining new auxiliary variables within the local models, so that the relation between v and (x, x_a, u) remains affine (3.2) as dictated by (3.2). Additionally, the states and the manipulated input variables are feasible if they respect the following set of bounds:

$$lb_i \leq \begin{bmatrix} x_i \\ x_{ai} \\ u_i \end{bmatrix} \leq ub_i \tag{3.3a}$$

where $i = 1, \ldots, m$, $lb_i = [lb_{x_i}^T, lb_{x_{ai}}^T, lb_{u_i}^T]^T$, and $ub_i = [ub_{x_i}^T, ub_{x_{ai}}^T, ub_{u_i}^T]^T$. The states of the m subsystems, x_i $(i = 1, \ldots, m)$, are assumed to be sampled synchronously at time instants t_k. The dynamics of the entire nonlinear system can be described as:

$$\dot{x} = f(x, x_a, u, v) \tag{3.4a}$$

$$0 = g(x, x_a, u) \tag{3.4b}$$

$$x(0) = x_0 \tag{3.4c}$$

[1]In this definition, auxiliary state variables depend implicitly on time; whereas, the time derivatives of the state variables appear explicitly in the ODE defining system dynamics

where: $x = [x_1^T \cdots x_i^T \cdots x_m^T]^T \in \mathbb{R}^{n_x}$ denotes the state variables vector, $x_a = [x_{a1}^T \cdots x_{ai}^T \cdots x_{am}^T]^T \in \mathbb{R}^{n_{x_a}}$ denotes the auxiliary state variables vector, $u = [u_1^T \cdots u_i^T \cdots u_m^T]^T$ are the manipulated input variables vector, and $v = [v_1, \ldots, v_i, \ldots, v_m]^T$ is the vector of interaction variables for the entire nonlinear system. In addition, $f = [f_1^T \cdots f_i^T \cdots f_m^T]^T$, $g = [g_1^T \cdots g_i^T \cdots g_m^T]^T$, and $x_0 = [x_{0_1}^T \cdots x_{0_i}^T \cdots x_{0_m}^T]^T$. Here, it is assumed that f and g are smooth Lipschitz functions. In addition, the differential-algebraic equation (DAE) system formulation defined in (3.1) is assumed to be index-1, which means that g is solvable for x_a. In other words, $g(x, x_a, u)$ can be written as $x_a = \bar{g}(x, u)$ from (3.4b). In this chapter, implicit Runge-Kutta (IRK) methods, such as Radau IIA formulation, are used due to their reliability in handling stiff problems [44]. Assuming that at t_k, the solution to (3.4b) exists, the nonlinear model of sub-system 'i' (3.1) can be discretized, at $t_k \leq t = t_k + c^l\tau \leq t_{k+1}$, by solving the following system of equations:

$$G_i = \begin{bmatrix} \mathcal{K}_{x,i}^l - f_i\Big(x_{i,t_k} + \tau \sum_{j=1}^{\ell} a_j^l \mathcal{K}_{x,i}^j, x_{a_i,t_k}^l + \tau \sum_{j=1}^{\ell} a_j^l \mathcal{K}_{x_A,i}^j, u_i^l, v_i^l\Big) \\ g_i\Big(x_{i,t_k} + \tau \sum_{j=1}^{\ell} a_j^l \mathcal{K}_{x,i}^j, x_{a_i,t_k}^l + \tau \sum_{j=1}^{\ell} a_j^l \mathcal{K}_{x_A,i}^j, u_i^l\Big) \end{bmatrix} = 0 \qquad (3.5a)$$

where: 'ℓ' denotes the number of stages in the IRK method, 'τ' denotes the integration step-size, and '$l = 1, \ldots, \ell$'; and, the state variables can be explicitly updated [82] for the next sampling period using:

$$x_{i,t_{k+1}} = \phi_i(x_{i,t_k}, x_{ai,t_k}, u_{i,t_k}, v_{i,t_k}) = x_{i,t_k} + \tau \sum_{l=1}^{\ell} b^l \mathcal{K}_{x,i}^l, \qquad (3.6a)$$

$$x_{a_i,t_{k+1}} = \bar{g}_i\Big(x_{i,t_{k+1}}, u_{i,t_k}\Big) \qquad (3.6b)$$

where: a_j^l, b^l and c^l are internal IRK coefficients [44].

3.1.2 Centralized NMPC

In this section, the centralized NMPC is defined to serve as the benchmark control approach in the development of the CDNMPC for nonlinear systems. The finite-time centralized NMPC formulation based on the discretized DAE model (3.6) at time instant t_k can be formulated as:

$$\min_{X,X_a,U} J_C = \sum_{i=1}^{m} J_{C_i} \tag{3.7a}$$

$$\text{s.t. } \hat{x}_i(t_{k+l+1}|t_k) = \phi_i\big(\hat{x}_i(t_{k+l}|t_k), \hat{x}_{a_i}(t_{k+l}|t_k), \hat{u}_i(t_{k+l}|t_k), \hat{v}_i(t_{k+l}|t_k)\big) \tag{3.7b}$$

$$\hat{x}_{a_i}(t_{k+l+1}|t_k) = \bar{g}_i\Big(\hat{x}_i(t_{k+l+1}|t_k), \hat{u}_i(t_{k+l}|t_k)\Big) \tag{3.7c}$$

$$\hat{v}_i(t_{k+l}|t_k) = \sum_{j\neq i} h_{i,j}[\hat{x}_j(t_{k+l}|t_k), \hat{x}_{a_j}(t_{k+l}|t_k), \hat{u}_j(t_{k+l}|t_k)]^T \tag{3.7d}$$

$$lb_i \leq \begin{bmatrix} x_i \\ x_{ai} \\ u_i \end{bmatrix} \leq ub_i \tag{3.7e}$$

where:

$$J_{C_i} = \big(X_i(t_k) - X_{i,sp}(t_k)\big)^T Q_{ii} \big(X_i(t_k) - X_{i,sp}(t_k)\big) + \big(U_i(t_k) - U_{i,sp}(t_k)\big)^T R_{ii} \big(U_i(t_k) - U_{i,sp}(t_k)\big) \tag{3.7f}$$

here: $\hat{x}$, $\hat{x_a}$, $\hat{u}$, and $\hat{v}$ are states, auxiliary states, manipulated input variables, and interaction variables inside the controller for $l = 0, \cdots, N$, respectively; and N is the prediction horizon[2]. Additionally: $X(t_k) = [\hat{x}(t_{k+1}|t_k)^T, \cdots, \hat{x}(t_{k+N}|t_k)^T]^T$ is the vector of the predicted dynamic state trajectory; $X_a(t_k) = [\hat{x}_a(t_{k+1}|t_k)^T, \cdots, \hat{x}(t_{k+N}|t_k)^T]^T$ is the vector of the predicted auxiliary state trajectory; $U(k) = [\hat{u}(t_k|t_k)^T, \cdots, \hat{u}(t_{k+N-1}|t_k)^T]^T$ is the vector of the calculated manipulated variable moves; Q is a positive definite block-diagonal weighting matrix for the dynamic states (i.e., $Q = diag\{Q_i\}$); and R is a positive definite block-diagonal weighting matrix for the manipulated variables of the overall system (i.e., $R = diag\{R_i\}$). Note that, in optimization problem (3.7), the vector of interaction variables is known over the whole prediction horizon, i.e. the centralized NMPC accounts for all the interactions between the sub-systems, as a part of the given model in the network.

3.1.3 Stability Analysis of Centralized NMPC

In this subsection, the idea stated in [80, 41, 42] is followed to develop a stability criteria for the centralized NMPC. Conventionally, a terminal region and/or terminal cost function is defined to ensure stability of centralized NMPC systems [17]. In this work, an adaptive horizon scheme is implemented to find the minimum prediction horizon required to stabilize the network. The main advantage of the adaptive horizon method is

[2]Here, for the sake of simplicity, it is assumed the control horizon and the prediction horizon of the system are the same. Although it might lead to a higher computation load, longer control horizons can improve the performance and compensate for package dropouts in networked control systems [80]

that the optimization problem inside NMPC is not altered and it can be extended to the CDNMPC problem in a straightforward manner. This criteria is applicable to NMPC schemes with admissible sets consisting of bound constraints and/or mixed states and input constraints [80].

Denote the optimum predicted state trajectory of the centralized NMPC optimization problem (3.7), for a prediction horizon N, as the open-loop control solution. Accordingly, the receding horizon control (RHC) action $u_{RHC}(N, x(n))$ is calculated from solving problem (3.7) with prediction horizon N and the initial value $x(n)$.

Define a dynamic programming value function [80] based on the RHC trajectory of the centralized NMPC as a function of a predefined stage cost l^C:

$$V_N^C\big(x(n)\big) = \sum_{n=0}^{N-1} l^C\Big(x(n), u_{RHC}(N-n, x(n))\Big) \tag{3.8a}$$

where the stage cost of the centralized trajectory is defined as:

$$\begin{aligned} l^C\Big(x(n), u_{RHC}(N-n, x(n))\Big) = \sum_{i=1}^{m} \Big(\big(x_i(n) - x_{i,sp}(t_k)\big)^T Q_{ii}(n) \big(x_i(n) - x_{i,sp}(t_k)\big) + \\ \big(u_{i,RHC}(N-n, x_i(n)) - u_{i,sp}(t_k)\big)^T R_{ii}(n) \big(u_{i,RHC}(N-n, x_i(n)) - u_{i,sp}(t_k)\big)\Big) \end{aligned} \tag{3.8b}$$

with Q_{ii} and R_{ii} defined over the prediction horizon as:

$$Q_{ii} = \begin{bmatrix} Q_{ii}(1) & & \\ & \ddots & \\ & & Q_{ii}(N) \end{bmatrix}, \quad R_{ii} = \begin{bmatrix} R_{ii}(1) & & \\ & \ddots & \\ & & R_{ii}(N) \end{bmatrix} \tag{3.8c}$$

Remark 10 *In order to obtain the value function V_N^C, the difference between open-loop control and receding horizon control (RHC) trajectories for a dynamic programming MPC*

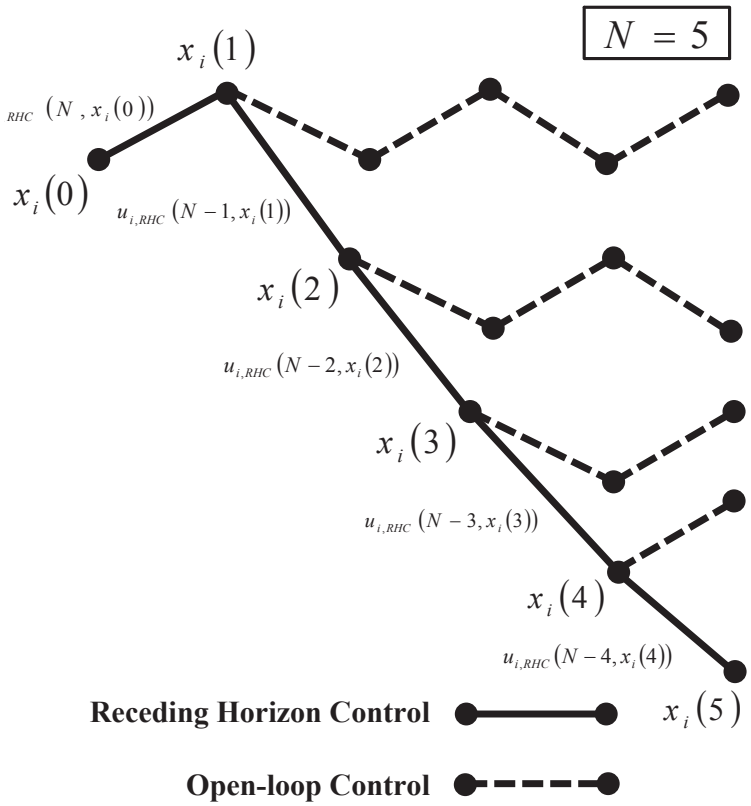


Figure 3.1: Receding horizon control (RHC) vs open-loop control trajectories for subsystem i with a typical prediction horizon length of $N = 5$.

problem, at sampling time t_k, has to be determined. Consider a sample problem, described in Figure 3.1, for a typical MPC problem with a prediction horizon $N = 5$. Firstly, an open-loop trajectory is calculated for $N = 5$ and the RHC action, applied to the process model, is used to solve the next open-loop problem for $N = 4$. This procedure continues until five RHC points are obtained and the value function V_N^C is calculated.

Therefore, in order to calculate V_N^C at the current sampling time t_k, namely $V_N^C\big(x(t_k)\big)$, one has to perform a multi-step calculation forward in time. The procedure to calculate (3.8) is listed in Algorithm 2:

Algorithm 2: The algorithm to calculate the finite-time value function (3.8)

Initialization: $n = 0$, prediction horizon N, number of sub-systems m, initial state $x(n) = x(t_k)$, Q_{ii}, R_{ii}, $V_N^C = 0$;

for $n := 0$ ***to*** $N - 1$ **do**

Perform the plant-wide optimization problem (3.7) with prediction horizon $N - n$ and initial value $x_i(0) = x_i(n)$;

Update the RHC control action move $u_{RHC}(N - n, x(n))$;

Calculate the centralized NMPC stage cost $l^C\Big(x(n), u_{RHC}(N - n, x(n))\Big)$;

$V_N^C = V_N^C + l^C\Big(x(n), u_{RHC}(N - n, x(n))\Big)$;

Apply $u_{RHC}(N - n, x(n))$ to the model and update $x_i(n + 1)$;

Output: the value function $V_N^C(x(n)) \leftarrow V_C^D$;

Lemma 11 *Consider the optimization problem (3.7) at sampling time t_k. Provided that the stage cost l^C is a positive definite function, and if there exists $\alpha \in [0, 1]$ such that:*

$$V_N^C(x(t_k)) - V_N^C(x(t_{k+1})) \geq \alpha l^C\Big(x(t_k), u_{RHC}(N, x(t_k))\Big) \tag{3.9}$$

then V_N^C is a Lyapunov function for the finite-time centralized NMPC system (3.7) and α is an estimate of the closed-loop sub-optimality degree of the system w.r.t. the corresponding infinite horizon NMPC problem, i.e. $\alpha \to 1$ as $N \to \infty$ [80].

Based on Lemma 11, an a posteriori algorithm to calculate α and determine the required prediction horizon 'N' can be presented to ensure closed-loop stability of the centralized NMPC problem (3.7). This procedure is:

Remark 12 *There has to be a trade-off between the values of the minimum required value of $\alpha_{\min}$ and the desired level of sub-optimality comparing to the infinite horizon*

Algorithm 3: A posteriori adaptive horizon algorithm for the NMPC problem (3.7)

Initialization: prediction horizon N, number of sub-systems m, initial state $x(n)$, Q_{ii}, R_{ii};
Calculate $V_N^C(x(n))$ using Algorithm 2 with $\big(N, m, x(n), Q_{ii}, R_{ii}\big)$;
Calculate $l^C\big(x(n), u_{RHC}(N, x(n))\big)$ using (3.8);
Apply $u_{RHC}(N, x(n))$ to the model and update $x(n+1)$;
Calculate $V_N^C(x(n+1))$ using Algorithm 2 with $\big(N, m, x(n+1), Q_{ii}, R_{ii}\big)$;
if $V_N^C(x(n)) \leq V_N^C(x(n+1))$ **then**
  Prompt: Solution may be unstable;
else
  Calculate $\alpha = \frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)}$;
  if $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)} \geq \alpha_{min}$ **then**
    Perform Horizon Shortening (Algorithm 4);
  else
    Perform Horizon Prolongation (Algorithm 5);
  end
end
Output: Apply the RHC action to the plant based on the accepted value of N;

trajectory. For values of $\alpha_{\min}$ *close to zero, fewer computations may be required but the closed-loop control performance may be sacrificed; on the other hand, as* $\alpha_{\min} \to 1$*, the computational cost will rise and the performance of the closed-loop systems will tend to that of the infinite horizon control.*

According to Algorithm 3, when the calculated $\alpha_N \geq \alpha_{\min}$, the prediction horizon is decreased as the algorithm converges the minimum amount of prediction horizon that satisfies (3.9). This procedure is called the horizon shortening strategy that is explained in Algorithm 4:

Algorithm 4: Horizon Shortening Algorithm for the NMPC problem (3.7)

Initialization: prediction horizon N, number of sub-systems m, initial state $x(k)$, Q_{ii}, R_{ii};

while $N \geq 2$ **do**

Save the current trajectory of the open-loop system;

$N = N - 1$;

Calculate $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)}$ with $\left(N, m, x(k), Q_{ii}, R_{ii}\right)$;

Save the shortened horizon trajectory of the open-loop system;

if $\alpha = \frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)} < \alpha_{min}$ **then**

$N = N + 1$;

Restore the current stored trajectory of the open-loop system;

STOP;

else

Save the shortened horizon trajectory as the current trajectory;

Output: Current trajectory of the system, and N.

When $\alpha_N < \alpha_{\min}$, the prediction horizon is increased to find the minimum N that satisfies $\alpha_N \geq \alpha_{\min}$. This procedure is listed in Algorithm 5:

Algorithm 5: Horizon Prolongation Algorithm for the NMPC problem (3.7)

Initialization: prediction horizon N, number of sub-systems m, initial state $x(k)$, Q_{ii}, R_{ii};

while $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)} < 1$ **do**

$N = N + 1$;

Calculate $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)}$ with $\left(N, m, x(k), Q_{ii}, R_{ii}\right)$;

Save the prolonged horizon trajectory as the current trajectory of the open-loop system;

Output: Current trajectory of the system, and N.

3.2 Proposed Coordination Algorithm

In this section, the proposed price-driven CDNMPC for the nonlinear DAE system (3.1) is presented. The algorithm is developed for the discretized model described in (3.6) and the receding horizon action is applied to continuous-time model (3.1). The main idea is to design a coordinator for an existing network of decentralized NMPCs, which will handle interactions between the sub-systems. This work extends the idea of [89] and [68] to design a new price-driven coordination scheme for constrained nonlinear DAE systems, with open-loop stable or unstable behavior. The proposed price-driven coordination is the hierarchical dual decomposition of the hypothetical centralized control problem defined in (3.7). In this approach the interaction constraints are relaxed by penalizing the local violations within the objective function of local NMPC controllers by a price vector. The dual decomposition formulation can be considered a bi-level nonlinear programming problem. Therefore, the coordinator converts the bi-level problem into a relaxed single level optimization problem. In addition, an on-line adaptive horizon selection is performed inside the coordinator to ensure stability of the closed-loop system; and, the local NMPC controllers update their internal prediction horizon length, accordingly.

Define the interaction error for subsystem 'i' based on (3.2) as:

$$e_i(t_{k+l}|t_k) \triangleq \hat{v}_i(t_{k+l}|t_k) - \sum_{j \neq i} h_{i,j}[\hat{x}_j(t_{k+l}|t_k), \hat{x}_{a_j}(t_{k+l}|t_k), \hat{u}_j(t_{k+l}|t_k)]^T \tag{3.10}$$

for $l = 0, \cdots, N-1$. A specific objective of the coordinator is to find a price for subsystem i such that the interaction term $\hat{v}_i$, which is subsequently calculated using the price, drives $e_i(t_{k+l}|t_k) = 0$. Define the overall interaction error over the prediction

horizon as:

$$E(t_k|t_k) \triangleq \begin{bmatrix} E_1(t_k|t_k) \\ \vdots \\ E_m(t_k|t_k) \end{bmatrix} \tag{3.11a}$$

where:

$$E_i = \begin{bmatrix} e_i(t_k|t_k) \\ e_i(t_{k+1}|t_k) \\ \vdots \\ e_i(t_{k+N-1}|t_k) \end{bmatrix} \tag{3.11b}$$

Therefore, according to (3.10), $E(t_k|t_k)$ can be written as:

$$E(t_k|t_k) = \left[\begin{array}{c} \hat{v}_1(t_k|t_k) - \sum_{j\neq 1} h_{1,j}[\hat{x}_j(t_k|t_k), \hat{x}_{a_j}(t_k|t_k), \hat{u}_j(t_k|t_k)]^T \\ \hat{v}_1(t_{k+1}|t_k) - \sum_{j\neq 1} h_{1,j}[\hat{x}_j(t_{k+1}|t_k), \hat{x}_{a_j}(t_{k+1}|t_k), \hat{u}_j(t_{k+1}|t_k)]^T \\ \vdots \\ \hat{v}_1(t_{k+N-1}|t_k) - \sum_{j\neq 1} h_{1,j}[\hat{x}_j(t_{k+N-1}|t_k), \hat{x}_{a_j}(t_{k+N-1}|t_k), \hat{u}_j(t_{k+N-1}|t_k)]^T \\ \hline \vdots \\ \hline \hat{v}_i(t_k|t_k) - \sum_{j\neq i} h_{i,j}[\hat{x}_j(t_k|t_k), \hat{x}_{a_j}(t_k|t_k), \hat{u}_j(t_k|t_k)]^T \\ \hat{v}_i(t_{k+1}|t_k) - \sum_{j\neq i} h_{i,j}[\hat{x}_j(t_{k+1}|t_k), \hat{x}_{a_j}(t_{k+1}|t_k), \hat{u}_j(t_{k+1}|t_k)]^T \\ \vdots \\ \hat{v}_i(t_{k+N-1}|t_k) - \sum_{j\neq i} h_{i,j}[\hat{x}_j(t_{k+N-1}|t_k), \hat{x}_{a_j}(t_{k+N-1}|t_k), \hat{u}_j(t_{k+N-1}|t_k)]^T \\ \hline \vdots \\ \hline \hat{v}_m(t_k|t_k) - \sum_{j\neq m} h_{m,j}[\hat{x}_j(t_k|t_k), \hat{x}_{a_j}(t_k|t_k), \hat{u}_j(t_k|t_k)]^T \\ \hat{v}_m(t_{k+1}|t_k) - \sum_{j\neq m} h_{m,j}[\hat{x}_j(t_{k+1}|t_k), \hat{x}_{a_j}(t_{k+1}|t_k), \hat{u}_j(t_{k+1}|t_k)]^T \\ \vdots \\ \hat{v}_m(t_{k+N-1}|t_k) - \sum_{j\neq m} h_{m,j}[\hat{x}_j(t_{k+N-1}|t_k), \hat{x}_{a_j}(t_{k+N-1}|t_k), \hat{u}_j(t_{k+N-1}|t_k)]^T \end{array}\right] \tag{3.12}$$

which can be rearranged into the following separable form:

$$E(t_k|t_k) = \begin{bmatrix} \hat{v}_1(t_k|t_k) \\ \hat{v}_1(t_{k+1}|t_k) \\ \vdots \\ \hat{v}_1(t_{k+N-1}|t_k) \\ \hline \vdots \\ \hline h_{i,1}[\hat{x}_1(t_k|t_k), \hat{x}_{a_1}(t_k|t_k), \hat{u}_1(t_k|t_k)]^T \\ h_{i,1}[\hat{x}_1(t_{k+1}|t_k), \hat{x}_{a_1}(t_{k+1}|t_k), \hat{u}_1(t_{k+1}|t_k)]^T \\ \vdots \\ h_{i,1}[\hat{x}_1(t_{k+N-1}|t_k), \hat{x}_{a_1}(t_{k+N-1}|t_k), \hat{u}_1(t_{k+N-1}|t_k)]^T \\ \hline \vdots \\ \hline h_{m,1}[\hat{x}_1(t_k|t_k), \hat{x}_{a_1}(t_k|t_k), \hat{u}_1(t_k|t_k)]^T \\ h_{m,1}[\hat{x}_1(t_{k+1}|t_k), \hat{x}_{a_1}(t_{k+1}|t_k), \hat{u}_1(t_{k+1}|t_k)]^T \\ \vdots \\ h_{m,1}[\hat{x}_1(k+N-1|k), \hat{x}_{a_1}(k+N-1|k), \hat{u}_1(t_{k+N-1}|t_k)]^T \end{bmatrix} + \cdots +$$

$$\begin{bmatrix} h_{1,m}[\hat{x}_m(t_k|t_k), \hat{x}_{a_m}(t_k|t_k), \hat{u}_m(t_k|t_k)]^T \\ h_{1,m}[\hat{x}_m(t_{k+1}|t_k), \hat{x}_{a_m}(t_{k+1}|t_k), \hat{u}_m(t_{k+1}|t_k)]^T \\ \vdots \\ h_{1,m}[\hat{x}_m(t_{k+N-1}|t_k), \hat{x}_{a_m}(t_{k+N-1}|t_k), \hat{u}_m(t_{k+N-1}|t_k)]^T \\ \hline \vdots \\ \hline h_{i,m}[\hat{x}_m(t_k|t_k), \hat{x}_{a_m}(t_k|t_k), \hat{u}_m(t_k|t_k)]^T \\ h_{i,m}[\hat{x}_m(t_{k+1}|t_k), \hat{x}_{a_m}(t_{k+1}|t_k), \hat{u}_m(t_{k+1}|t_k)]^T \\ \vdots \\ h_{i,m}[\hat{x}_m(t_{k+N-1}|t_k), \hat{x}_{a_m}(t_{k+N-1}|t_k), \hat{u}_m(t_{k+N-1}|t_k)]^T \\ \hline \vdots \\ \hline \hat{v}_m(t_k|t_k) \\ \hat{v}_m(t_{k+1}|t_k) \\ \vdots \\ \hat{v}_m(t_{k+N-1}|t_k)) \end{bmatrix} \tag{3.13a}$$

$$= \Theta_1 \begin{bmatrix} X_1(t_k) \\ X_{a1}(t_k) \\ U_1(t_k) \\ V_1(t_k) \end{bmatrix} + \cdots + \Theta_m \begin{bmatrix} X_m(t_k) \\ X_{am}(t_k) \\ U_m(t_k) \\ V_m(t_k) \end{bmatrix} \tag{3.13b}$$

$$= \sum_{i=1}^{m} \Theta_i \begin{bmatrix} X_i(t_k) \\ X_{ai}(t_k) \\ U_i(t_k) \\ V_i(t_k) \end{bmatrix} \tag{3.13c}$$

where: $V_i(t_k) \triangleq [\hat{v}_i(t_{k+1}|t_k)^T, \cdots, \hat{v}_i(t_{k+N}|t_k)^T]^T$ is the vector of predicted interaction

variables, and Θ_i is the interaction coefficient matrix of sub-system i. Therefore, the plant-wide optimization problem can be written based on the fact that the overall interaction error (3.13) is zero:

$$\min_{X,X_a,U} J_P = \sum_{i=1}^{m} J_{P_i} \tag{3.14a}$$

$$\text{s.t. } \hat{x}_i(t_{k+l+1}|t_k) = \phi_i\big(\hat{x}_i(t_{k+l}|t_k), \hat{x}_{a_i}(t_{k+l}|t_k), \hat{u}_i(t_{k+l}|t_k), \hat{v}_i(t_{k+l}|t_k)\big) \tag{3.14b}$$

$$\hat{x}_{a_i}(t_{k+l+1}|t_k) = \bar{g}_i\Big(\hat{x}_i(t_{k+l+1}|t_k), \hat{u}_i(t_{k+l}|t_k)\Big) \tag{3.14c}$$

$$E(t_k|t_k) = \sum_{i=1}^{m} \Theta_i \begin{bmatrix} X_i(t_k) \\ X_{ai}(t_k) \\ U_i(t_k) \\ V_i(t_k) \end{bmatrix} = 0 \tag{3.14d}$$

$$lb_i \leq \begin{bmatrix} X_i \\ X_{a_i} \\ U_i \end{bmatrix} \leq ub_i \tag{3.14e}$$

where:

$$\begin{aligned} J_{P_i} =& \big(X_i(t_k) - X_{i,sp}(t_k)\big)^T Q_{ii} \big(X_i(t_k) - X_{i,sp}(t_k)\big) + \big(U_i(t_k) \\ & - U_{i,sp}(t_k)\big)^T R_{ii} \big(U_i(t_k) - U_{i,sp}(t_k)\big) \end{aligned} \tag{3.14f}$$

This reformulation provides the basis for the CDNMPC design. Note that the cost function and constraints are separable by subsystem. Based on the separability of overall interaction error, the interaction constraint (3.14d) is relaxed by penalizing local interactions error within the modified NMPC objective functions with a price vector p:

$$\min_{X,X_a,U,V} J_D = \sum_{i=1}^{m} J_{D_i} \tag{3.15a}$$

$$\text{s.t. } \hat{x}_i(t_{k+1}|t_k) = \phi_i\big(\hat{x}_i(t_k|t_k), \hat{x}_{A_i}(t_k|t_k), \hat{u}_i(t_k|t_k), \hat{v}_i(t_k|t_k)\big) \tag{3.15b}$$

$$\hat{x}_{a_i}(t_{k+l+1}|t_k) = \bar{g}_i\Big(\hat{x}_i(t_{k+l+1}|t_k), \hat{u}_i(t_{k+l}|t_k)\Big) \tag{3.15c}$$

$$lb_i \leq \begin{bmatrix} X_i \\ X_{a_i} \\ U_i \end{bmatrix} \leq ub_i \tag{3.15d}$$

where:

$$J_{D_i} = \big(X_i(t_k) - X_{i,sp}(t_k)\big)^T Q_{ii}\big(X_i(t_k) - X_{i,sp}(t_k)\big) + \big(U_i(t_k) - U_{i,sp}(t_k)\big)^T R_{ii}\big(U_i(t_k) - U_{i,sp}(t_k)\big) + p^T \Theta_i [X_i(t_k), X_{A_i}(t_k), U_i(t_k), V_i(t_k)]^T \tag{3.15e}$$

The coordination level provides the optimum price vector for the local NMPC controllers by solving the following maximization problem over the distributed objective function:

$$\max_{p} \; J_D(X, X_a, U, V, p) \tag{3.15f}$$

Remark 13 *Define the decentralized NMPC as a control performance benchmark of the CDNMPC trajectory (3.15) vs the centralized trajectory (3.7):*

$$\min_{X,X_a,U} J_{DC} = \sum_{i=1}^{m} J_{DC_i} \tag{3.16a}$$

$$s.t.\ \hat{x}_i(t_{k+l+1}|t_k) = \phi_i\big(\hat{x}_i(t_{k+l}|t_k), \hat{x}_{a_i}(t_{k+l}|t_k), \hat{u}_i(t_{k+l}|t_k), \hat{v}_{i,ss}\big) \tag{3.16b}$$

$$\hat{x}_{a_i}(t_{k+l+1}|t_k) = \bar{g}_i\Big(\hat{x}_i(t_{k+l+1}|t_k), \hat{u}_i(t_{k+l}|t_k)\Big) \tag{3.16c}$$

$$v_{i,ss} = h_i[x_{i,ss}, x_{a_{i,ss}}, u_{i,ss}]^T \tag{3.16d}$$

$$lb_i \leq \begin{bmatrix} X_i \\ X_{a_i} \\ U_i \end{bmatrix} \leq ub_i \tag{3.16e}$$

where:

$$\begin{aligned} J_{DC_i} =& \big(X_i(t_k) - X_{i,sp}(t_k)\big)^T Q_{ii} \big(X_i(t_k) - X_{i,sp}(t_k)\big) + \\ & \big(U_i(t_k) - U_{i,sp}(t_k)\big)^T R_{ii} \big(U_i(t_k) - U_{i,sp}(t_k)\big) \end{aligned} \tag{3.16f}$$

and subscript 'ss' denotes the steady state value. The main feature of this formulation is that the interaction information between sub-systems, namely $v_{i,ss}$, is calculated at the desired steady-state condition. Unlike the decentralized NMPC, in centralized NMPC, the interaction information over the whole prediction horizon is available; but, in the CDN-MPC formulation the interaction information is an optimization variable over the whole prediction horizon. This would result in degradation of performance in the decentralized NMPC vs the centralized NMPC. In this paper, it is supposed that this control scheme exists in the plant and the CDNMPC strives to improve the performance of this controller network. Note that, the CDNMPC system is supposed to perform, at least, better than the decentralized NMPC and as close as possible to the centralized NMPC.

The CDNMPC formulation (3.15) can be summarized as the following bi-level optimization problem:

$$\max_{p} \; J_D(Z, p) \tag{3.17a}$$

$$\min_{Z} J_D = \sum_{i=1}^{m} J_{D_i}(Z_i, p) \tag{3.17b}$$

$$\text{s.t. } G_i^{eq}(Z_i) = 0 \tag{3.17c}$$

$$G_i(Z_i)^{ineq} \leq 0 \tag{3.17d}$$

where J_{D_i} is reformulated as:

$$J_{D_i} = \frac{1}{2} Z_i(t_k)^T \Upsilon_i Z_i(t_k) + \Psi_i Z_i(t_k) + p^T \Theta_i Z_i(t_k) \tag{3.17e}$$

and Z_i is defined as:

$$Z_i = \begin{bmatrix} X_i \\ X_{a_i} \\ U_i \\ V_i \end{bmatrix} \tag{3.17f}$$

Introducing non-negative slack vectors s_i for the inequality constraints of all subproblems '$i = 1, \ldots, m$', the associated Lagrangian functions $\mathcal{L}_i$ can be formulated as:

$$\mathcal{L}_i = J_{D_i} + \lambda_i^{eq^T} \left(G_i^{eq}(Z_i(t_k)) \right) + \lambda_i^{ineq^T} \left(G_i^{ineq}(Z_i(t_k)) + s_i \right) \tag{3.18}$$

The overall Lagrangian of the lower level problem can be written as:

$$\mathcal{L} = \sum_{i=1}^{m} \mathcal{L}_i(Z_i, p, \lambda_i^{eq}, \lambda_i^{ineq}, s_i) \tag{3.19}$$

Bi-level programming (BLP) problems with convex objective functions and constraints have been extensively discussed in [9, 29, 27]. There are several algorithms for Nonlinear or non-convex BLP problems, under various assumptions [27]. Herskovits et al. [46] proposed an algorithm for BLP problems with strictly convex lower level problems. In [5] the lower problem is allowed to have a nonlinear constraint, but it has to be convex for fixed values of the upper-level variable; also, the upper-level problem and the objective function of the lower-level problem are assumed to be convex. Colson et al. [26] proposed a trust-region solution method for nonlinear BLPs by linear approximations of the upper-level objective function and all constraints, as well as a quadratic approximation of the lower-level objective function. Clark et al. [24] proposed an active-set strategy for nonlinear BLP problems.

In this work, an exact penalty method is deployed to solve the CDNMPC problem. The lower-level problem is constructed by the distributed NMPCs and the upper-level problem tries to minimize the overall interaction error (3.13). Here two assumptions on the overall problem (3.17) are made:

Assumption 14 *To guarantee that there is at least one solution to the Nonlinear BLP problem (3.17), it is assumed that the feasible solution set of the lower-level problem based on the price vector $\Omega(p)$ is nonempty, uniformly compact.*

Assumption 15 *For every local optimum of the lower-level problem in (3.17), there exist a unique optimal price vector associated with the distributed NMPC network.*

Additionally, consider the following set of assumptions on the lower-level problem for a

fixed value of p to ensure the regularity of constraints holds at $Z \in (\Omega(p))$ [11]:

Assumption 16 *The strict complementarity slackness (SCS) property [11] holds at* $Z \in (\Omega(p))$ *w.r.t.* $(\lambda^{eq}, \lambda^{ineq})$[3].

Assumption 17 *The Mangasarian Fromowitz constraint qualification (MFCQ) [79] holds at* $Z \in (\Omega(p))$ *w.r.t.* $(\lambda^{eq}, \lambda^{ineq})$[4].

The BLP (3.17) is reduced into a single level optimization problem using the KKT conditions of the lower-level problem, as constraints of the upper-level problem. From Assumption 17, for a fixed value of p, every $Z \in (\Omega(p))$ must satisfy the KKT conditions of the lower-level problem:

$$\nabla_{Z_i}\mathcal{L}_i = \Upsilon_i Z_i(t_k) + \Psi_i^T + \Theta_i^T p + \nabla G_i^{eq}(Z_i(t_k))\lambda_i^{eq} + G_i^{ineq}(Z_i(t_k))\lambda_i^{ineq} = 0, \quad (3.21a)$$

$$G_i^{eq}(Z_i) = 0, \quad (3.21b)$$

$$G_i^{ineq}(Z_i) + s_i = 0, \quad (3.21c)$$

$$s_i^T \lambda_i^{ineq} = 0, \quad (3.21d)$$

$$s_i \geq 0,\ \lambda_i^{ineq} \geq 0 \quad (3.21e)$$

where s_i is a non-negative slack variable used to reformulate the inequalities into equality constraints as well as making the complementarity constraints linear. Furthermore, according to Assumption 17 the Lagrange multipliers are bounded [11].

The resulting formulation of (3.17) can also be regarded as a mathematical program

[3]SCS means: for all $\lambda_i^{ineq} > 0$, $G_i^{ineq}(Z_i(t_k))$=0 must hold.

[4]MFCQ holds if there exists a feasible direction d such that:

$$\begin{cases} d^T \nabla_Z G_i^{ineq}(Z_i(t_k)) < 0, & \text{when } G_i^{ineq}(Z_i(t_k)) \text{ is active} \\ d^T \nabla_Z G_i^{eq}(Z_i(t_k)) = 0 \end{cases} \quad (3.20)$$

and elements of $\nabla_Z G_i^{eq}(Z_i(t_k))$ are linearly independent.

with equilibrium constraints (MPEC):

$$\min_{p,Z_i,s_i,\lambda_i^{eq},\lambda_i^{ineq}} \quad -J_D(Z,p) \tag{3.22a}$$

$$\text{s.t. } \nabla_{Z_i}\mathcal{L}_i(Z_i,s_i)=0, \tag{3.22b}$$

$$G_i^{eq}(Z_i)=0, \tag{3.22c}$$

$$G_i^{ineq}(Z_i)+s_i=0, \tag{3.22d}$$

$$s_i^T\lambda_i^{ineq}=0, \tag{3.22e}$$

$$s_i\geq 0,\ \lambda_i^{ineq}\geq 0 \tag{3.22f}$$

This reformulation of the non-convex problem (3.17) is neither differentiable or regular [18], instead the following perturbation of (3.17), for a given vector $\eta=[\eta_i]$, is solved:

$$\min_{p,Z_i,s_i,\lambda_i^{eq},\lambda_i^{ineq}} \quad -J_D(Z,p) \tag{3.23a}$$

$$\text{s.t. } \nabla_{Z_i}\mathcal{L}_i(Z_i,s_i)=0, \tag{3.23b}$$

$$G_i^{eq}(Z_i)=0, \tag{3.23c}$$

$$G_i^{ineq}(Z_i)+s_i=0, \tag{3.23d}$$

$$s_i^T\lambda_i^{ineq}=\eta_i^2, \tag{3.23e}$$

$$s_i\geq 0,\ \lambda_i^{ineq}\geq 0 \tag{3.23f}$$

Starting from a positive value for η, this produces a set of smooth trajectories toward $\eta\rightarrow 0$ to find an approximate solution to (3.22), as stated in [52]. In this work, the nonlinear program with complementary constraints (3.21) is smoothed using CHKS smoothing

function [33] [5] Π defined as:

$$\Pi(a, b, \eta) = a + b - \sqrt{(a-b)^2 + 4\eta^2} \tag{3.25}$$

Remark 18 *For a given value of $\eta \geq 0$, $\Pi(s_i, \lambda_i^{ineq}, \eta) = 0$ is equivalent to the following condition:*

$$s_i \lambda_i^{ineq} = \eta_i^2 \tag{3.26}$$

as long as $s_i \geq 0,\ \lambda_i^{ineq} \geq 0$ hold.

Therefore, problem (3.23) is reformulated using the smoothing function (3.26) with $\Pi(s_i, \lambda_i^{ineq}, \eta) = 0$ into the following single level optimization problem:

$$\min_{p, Z_i, s_i, \lambda_i^{eq}, \lambda_i^{ineq}} \quad -J_D(Z, p) \tag{3.27a}$$

$$\text{s.t. } \nabla_{Z_i} \mathcal{L}_i(Z_i, s_i) = 0, \tag{3.27b}$$

$$G_i^{eq}(Z_i) = 0, \tag{3.27c}$$

$$G_i^{ineq}(Z_i) + s_i = 0, \tag{3.27d}$$

$$\Pi(s_i, \lambda_i^{ineq}, \eta) = 0, \tag{3.27e}$$

$$s_i \geq 0,\ \lambda_i^{ineq} \geq 0 \tag{3.27f}$$

Remark 19 *For a fixed value of η, while SCS (Assumption 16) and MFCQ (Assumption*

[5] As an alternative, a perturbed version of Fischer-Burmeister [37] can also be used :

$$\Pi(a, b, \eta) = a + b - \sqrt{a^2 + b^2 + 2\eta} \tag{3.24}$$

17) hold, problem (3.27) can overcome the difficulties in (3.22) due to smoothness and regularity issues [11, 52], and paves the way to solve (3.22) using exact penalty methods.

In this study, a local optimal solution to the nonlinear problem (3.27) is found, such that it is able to reduce the norm of the overall interaction error $\big|\big|E(t_k)\big|\big|$, defined in (3.13), within a sufficiently small predefined tolerance ϵ_0. Such a constraint to is then imposed into problem (3.27):

$$\min_{p,Z_i,s_i,\lambda_i^{eq},\lambda_i^{ineq}} \quad -J_D(Z,p) \tag{3.28a}$$

$$\text{s.t. } \nabla_{Z_i}\mathcal{L}_i(Z_i,s_i)=0, \tag{3.28b}$$

$$G_i^{eq}(Z_i)=0, \tag{3.28c}$$

$$G_i^{ineq}(Z_i)+s_i=0, \tag{3.28d}$$

$$\Pi(s_i,\lambda_i^{ineq},\eta)=0, \tag{3.28e}$$

$$-s_i\leq 0, -\lambda_i^{ineq}\leq 0 \tag{3.28f}$$

$$\big|\big|E(t_k)\big|\big|^2-\epsilon_0^2\leq 0 \tag{3.28g}$$

Note that, the feasible region of the transformed single-layer problem (3.27) might be expanded (compared to the original BLP problem) with stationary local optimum points of the lower-level problem, due to the KKT transformation. By adding $\big|\big|E(t_k)\big|\big|^2-\epsilon_0^2\leq 0$, the local search domain is more restricted and the corresponding optimal solution performs more efficient compared to the decentralized NMPC trajectory, since any feasible solution satisfies a predefined upper bound on the norm of overall interaction error. Moreover, the added constraint, is convex [6] and smooth; thus, it does not affect uniqueness of the solution.

[6]This constraint is convex since $\nabla_Z\big(\big|\big|E(t_k)\big|\big|^2\big)=2\Theta^T\big(\Theta Z\big)$ and $\nabla^2_{Z,Z}\big(\big|\big|E(t_k)\big|\big|^2\big)=2\Theta^T\Theta\succ 0$ for any Θ, which is the second-order condition qualification of convex functions [12].

Theorem 20 *For any value of η, consider that the solution to (3.28) belongs to the set $\mathcal{S}_\eta$. Then for every $\hat{\eta}$ there exists a compact set $\mathcal{C}(\hat{\eta})$ such that $\mathcal{S}_\eta \subseteq \mathcal{C}(\hat{\eta})$ holds for every $\eta \in (0, \hat{\eta}]$.*

Proof. According to Assumption 15, for every solution that belongs to $\mathcal{S}_\eta$ and $\eta > 0$, there exist a unique bounded optimal price vector. Also, by the properties of the function Π in (3.25), if a feasible solution is found w.r.t the lower-level constraints and the added convex constraint $\left|\left|E(t_k)\right|\right|^2 - \epsilon_0^2 \leq 0$, then it would belong to a compact set according to Assumption 14.

As a result, since for the slack vector $s_i = -G_i^{ineq}(Z_i)$ holds, s_i also belongs to a compact set. Additionally, since MFCQ (Assumption 17) holds, the Lagrange multipliers λ_i^{eq} and λ_i^{ineq} would be bounded and belong to a compact set. This follows the conclusion that $\mathcal{S}_\eta \subseteq \mathcal{C}(\hat{\eta})$ holds for every $\eta \in (0, \hat{\eta}]$. ■

The conclusion of this section can be stated in Remark 21 on properties of the solution set $\mathcal{S}_\eta$ at the current sampling time 't_k' for a given decreasing sequence of η starting from $\eta = 1$.

Remark 21 *According to Theorem 20 and Remark 19, the solution set $\mathcal{S}_\eta(t_k)$ is a non-empty and compact set for all values of $0 \leq \eta \leq 1$ [33]. This produces a continuous central path [33, 52] towards the solution of problem (3.22), as a function of the decreasing sequence η starting from a value between $(0, 1)$.*

In this section, a trust-region exact penalty method is used to solve the smoothed series of transformed CDNMPC problems (3.28), for a predefined decreasing sequence of η. The method is called exact in the sense that, for a sufficiently large penalty parameter, local solutions of the corresponding nonlinear programming problem converge to local minimizers of a defined exact penalty function [38].

In order to simplify the notation in (3.28), define the concatenated vectors of system variables and Lagrange multipliers as: $Z = [Z_1^T, \ldots, Z_m^T]^T$, $S = [s_1^T, \ldots, s_m^T]^T$, $\Lambda^{eq} = [\lambda_1^{eq^T}, \ldots, \lambda_m^{eq^T}]^T$, $\Lambda^{ineq} = [\lambda_1^{ineq^T}, \ldots, \lambda_m^{ineq^T}]^T$, and $Y = [p^T, Z^T, S^T, \Lambda^{eq^T}, \Lambda^{ineq^T}]^T$. Now, the optimization problem (3.28) is presented in a simpler notation:

$$\min_{Y} \; F(Y, \eta) \tag{3.29a}$$

$$\text{s.t.} \quad C^{eq}(Y, \eta) = 0 \tag{3.29b}$$

$$C^{ineq}(Y, \eta) \leq 0 \tag{3.29c}$$

where:

$$F(Y, \eta) = \; - J_D(Z, p) \tag{3.29d}$$

$$C_i^{eq}(Y, \eta) = \begin{bmatrix} \nabla_{Z_i} \mathcal{L}_i(Z_i, s_i) \\ G_i^{eq}(Z_i) \\ G_i^{ineq}(Z_i) + s_i \\ \Pi(s_i, \lambda_i^{ineq}, \eta) \end{bmatrix}, \tag{3.29e}$$

$$C^{eq}(Y, \eta) = [C_1^{eq^T}(Y), \ldots, C_m^{eq^T}(Y)]^T, \tag{3.29f}$$

and:

$$C_i^{ineq}(Y, \eta) = \begin{bmatrix} -s_i \\ -\lambda_i^{ineq} \end{bmatrix}, \tag{3.29g}$$

$$C^{ineq}(Y, \eta) = \left[C_1^{ineq^T}(Y), \ldots, C_m^{ineq^T}(Y), \left|\left| E(t_k) \right|\right|^2 - \epsilon_0^2 \right]^T \tag{3.29h}$$

Lemma 22 *Suppose Y^* is a KKT point of the NLP defined in (3.29), then there exists μ^* equal to the maximum Lagrange multiplier of (3.29) such that Y^* is a local minimizer*

of the exact penalty function $\Phi(Y, \mu, \eta)$ *[79], defined as:*

$$\Phi(Y, \mu, \eta) = F(Y, \eta) + \mu\Big(\big|C^{eq}(Y, \eta)\big|_1 + \max\{0, C^{ineq}(Y, \eta)\}\Big) \tag{3.30}$$

or:

$$Y^* = \arg\min_Y \big\{\Phi(Y, \mu, \eta)\big\}, \qquad \forall \mu \geq \mu^* \tag{3.31}$$

Define a model $\tilde{\Phi}(Y, \mu, \eta)$ of $\Phi(Y, \mu, \eta)$ that is constructed by quadratic approximation of the objective function F, and affine approximations of constraints C^{eq} and C^{ineq}:

$$\begin{aligned}\tilde{\Phi}(d_Y, \mu, \eta) = F_k + \nabla_Y F_k^T d_Y + \frac{1}{2} d_Y^T W_k d_Y + \\ \mu\Big(\big|C_k^{eq} + \nabla_Y C^{eq} d_Y\big|_1 + \max\{0, C_k^{ineq} + \nabla_Y C_k^{ineq} d_Y\}\Big)\end{aligned} \tag{3.32a}$$

where:

$$F_k = F(Y_k, \eta) \tag{3.32b}$$

$$\nabla_Y F_k = \nabla_Y F(Y_k, \eta) \tag{3.32c}$$

$$C_k^{eq} = C^{eq}(Y_K, \eta) \tag{3.32d}$$

$$\nabla_Y C_k^{eq} = \nabla_Y C^{eq}(Y_K, \eta) \tag{3.32e}$$

$$C_k^{ineq} = C^{ineq}(Y_K, \eta) \tag{3.32f}$$

$$\nabla_Y C_k^{ineq} = \nabla_Y C^{ineq}(Y_K, \eta) \tag{3.32g}$$

$$d_Y = Y - Y_k \tag{3.32h}$$

Inside the model defined in (3.32), W_k is usually taken as a symmetric matrix that contains the second derivative information of the objective function F and constraints

C^{eq} and C^{ineq} [79].

Remark 23 *For the purpose of implementation, the idea of sequential convex programming (SCP) [49, 30, 83] is followed. A convex quadratic approximation of F is then used for the purpose of constructing the model $\tilde{\Phi}$. Thus, W_k in (3.32) is the positive-semidefinite part of $\nabla^2_{YY}F$* [7] *at iteration 't_k':*

$$W_k = (\nabla^2_{YY}F(Y_k, \eta))_+ \tag{3.33}$$

Although the terminal quadratic convergence properties of SQP methods could be lost, this convexification method is straightforward to implement and convex optimization techniques can be applied to solve the iterative sub-problems [49]. If $(\nabla^2_{YY}F(Y_k, \eta))_+ = 0$ the approximation falls into the SLQP category discussed in [79].

The approximate formulation of the NLP defined in (3.31) can now be written as an optimization problem using the penalty model (3.32) with a trust-region constraint on Y:

$$\min_{d_Y} \tilde{\Phi}(d_Y, \mu, \eta), \qquad \forall \mu \geq \mu^* \tag{3.34a}$$

$$\text{s.t. } \left|\left|d_Y\right|\right|_\infty \leq \Delta_k \tag{3.34b}$$

The constraint violations of optimization problem (3.34) can be written as:

$$\mathcal{E}(d_Y, \eta) = \left|C^{eq}_k + \nabla_Y C^{eq} d_Y\right|_1 + \max\{0, C^{ineq}_k + \nabla_Y C^{ineq}_k d_Y\} \tag{3.35}$$

[7] For the real symmetric matrix $\nabla^2_{YY}F$ there exists an eigenvalue decomposition $\nabla^2_{YY}F = ADA^T$. Changing D to D_+ by replacing each negative eigenvalue with 0, the corresponding "positive-semidefinite part" can be calculated by $(\nabla^2_{YY}F)_+ = AD_+A^T$.

Furthermore, the corresponding feasibility problem is formulated based on (3.35) as:

$$\min_{d_Y} \mathcal{E}(d_Y, \eta) \tag{3.36a}$$

$$\text{s.t. } \left|\left|d_Y\right|\right|_\infty \leq \Delta_k \tag{3.36b}$$

The objective function in (3.34) is still not smooth and linearization might lead to infeasibility. To overcome these problems and ensure the trust region constraint is always satisfied, slack variables w_1, w_2 and w_3 are introduced to solve an equivalent elastic-mode problem [79]:

$$[d_{Y,\mu}, w_{1,\mu}, w_{2,\mu}, w_{3,\mu}] = \arg \min_{d_Y, w_1, w_2, w_3} \Big\{ F_k + \nabla_Y F_k^T d_Y + \frac{1}{2} d_Y^T W_k d_Y + \mu \Big(\mathbf{1} \cdot (w_1 + w_2) + \mathbf{1} \cdot w_3 \Big) \Big\} \tag{3.37a}$$

$$\text{s.t.} \quad C_k^{eq} + \nabla_Y C^{eq} d_Y = w_1 - w_2 \tag{3.37b}$$

$$C_k^{ineq} + \nabla_Y C_k^{ineq} d_Y \leq w_3 \tag{3.37c}$$

$$w_1, w_2, w_3 \geq 0 \tag{3.37d}$$

$$\left|\left|d_Y\right|\right|_\infty \leq \Delta_k \tag{3.37e}$$

where: $\mathbf{1}$ represents a row vector defined as $\mathbf{1} = [1, \cdots, 1]$. Correspondingly, define the elastic version of the feasibility problem (3.36) using slack variables w_1, w_2 and w_3:

$$[d_{Y,f}, w_{1,f}, w_{2,f}, w_{3,f}] = \arg \min_{d_Y, w_1, w_2, w_3} \Big\{ \mathbf{1} \cdot (w_1 + w_2) + \mathbf{1} \cdot w_3 \Big\} \tag{3.38a}$$

$$\text{s.t. } C_k^{eq} + \nabla_Y C^{eq} d_Y = w_1 - w_2 \tag{3.38b}$$

$$C_k^{ineq} + \nabla_Y C_k^{ineq} d_Y \leq w_3 \tag{3.38c}$$

$$w_1, w_2, w_3 \geq 0 \tag{3.38d}$$

$$\left|\left|d_Y\right|\right|_\infty \leq \Delta_k \tag{3.38e}$$

It is crucial to update the penalty parameter 'μ' at each iteration of the algorithm since 'μ^*' is not known. The procedure to update 'μ', listed in Algorithm 6, is adapted from with the termination tolerance criteria '$\epsilon > 0$'. The importance of choosing a proper value for μ is to avoid an unnecessary imbalance in the merit function and at the same time being large enough to make sufficient progress in linearized feasibility at each iteration [79].

Algorithm 6: Update procedure for penalty parameter μ

Input: Y_k; penalty parameter μ_{k-1}; trust-region radius Δ_k; $\epsilon_1, \epsilon_2 \in (0, 1)$;

Local NMPCs: Send C_i^{eq}, $\nabla_Y C_i^{eq}$, C_i^{ineq}, $\nabla_Y C_i^{ineq}$ to the coordinator;

Coordinator: Build C^{eq}, $\nabla_Y C^{eq}$, C^{ineq}, and $\nabla_Y C^{ineq}$;

Coordinator - Local NMPCs: Compute $d_{Y,\mu}$ using (3.37) with $\mu = \mu_{k-1}$;

Coordinator: Compute $\mathcal{E}(d_{Y,\mu}, \eta)$ using (3.32);

if $\mathcal{E}(d_{Y,\mu}, \eta) = 0$ **then**

  $\mu = \mu_{k-1}$;

else

  Coordinator - Local NMPCs: Compute $d_{Y,f}$ using (3.38);

  Coordinator: Compute $\mathcal{E}(d_{Y,f}, \eta)$ using (3.32);

  if $\mathcal{E}(d_{Y,f}, \eta) = 0$ **then**

    Coordinator - Local NMPCs: Find $\mu > \mu_{k-1}$ such that: $\mathcal{E}(d_{Y,\mu}, \eta) = 0$;

  else

    Coordinator - Local NMPCs: Find $\mu \geq \mu_{k-1}$ such that:

    $\mathcal{E}(0, \eta) - \mathcal{E}(d_{Y,\mu}, \eta) \geq \epsilon_1[\mathcal{E}(0, \eta) - \mathcal{E}(d_{Y,f}, \eta)]$;

Coordinator - Local NMPCs: Increase μ such that:

$\tilde{\Phi}(0, \mu, \eta) - \tilde{\Phi}(d_{Y,\mu}, \mu, \eta) \geq \epsilon_2[\mathcal{E}(0, \eta) - \mathcal{E}(d_{Y,\mu}, \eta)]$;

Output: $\mu_k = \mu$, $Y = Y_k + d_{Y,\mu}$;

Algorithms based on merit functions may fail to converge rapidly because they reject

steps that make good progress toward a solution, known as the Maratos effect [65]. This undesirable phenomenon, can also be avoided by occasionally accepting steps that a decrease in the merit function is not observed. Second order correction [79] is one way to overcome such a problem, in which a quadratic approximations is added to the equality and inequality constraints in problems (3.37) and (3.38):

$$\hat{C}^{eq}(Y_k,\eta) = C^{eq}(Y_k,\eta) + \nabla_Y C^{eq}(Y_k,\eta)d_Y + \frac{1}{2}d_Y^T\nabla^2_{YY}C^{eq}(Y_k,\eta)d_Y \tag{3.39a}$$

$$\hat{C}^{ineq}(Y_k,\eta) = C^{ineq}(Y_k,\eta) + \nabla_Y C^{ineq}(Y_k,\eta)d_Y + \frac{1}{2}d_Y^T\nabla^2_{YY}C^{ineq}(Y_k,\eta)d_Y \tag{3.39b}$$

which yields:

$$[d_{Y,\mu}, w_{1,\mu}, w_{2,\mu}, w_{3,\mu}] = \arg\min_{d_Y, w_1, w_2, w_3} \Big\{ F_k + \nabla_Y F_k^T d_Y + \frac{1}{2}d_Y^T W_k d_Y + \mu\Big(\mathbf{1}\cdot(w_1+w_2) + \mathbf{1}\cdot w_3\Big)\Big\} \tag{3.40a}$$

$$\text{s.t.} \quad \hat{C}^{eq}(Y_k,\eta) = w_1 - w_2 \tag{3.40b}$$

$$\hat{C}^{ineq}(Y_k,\eta) \leq w_3 \tag{3.40c}$$

$$w_1, w_2, w_3 \geq 0 \tag{3.40d}$$

$$\big|\big|d_Y\big|\big|_\infty \leq \Delta_k \tag{3.40e}$$

Note that this correction is applied when the increase in the merit function '$\tilde{\Phi}(d_Y,\mu,\eta)$' is accompanied by an increase in the norm of the constraint violation criteria '$h(d_Y,\eta)$'.

Algorithm 7: The trust-region algorithm for solving problem (3.29)

Input: $Y_0; \mu_0; \Delta_0$; $\Delta_{\min}$; iteration number $k = 0$; $k_{\max}$; $\epsilon_0, \epsilon_1, \epsilon_2 \in (0, 1)$;

$\beta_{\min} \in (0, 1)$; $\gamma_{success} \geq 1$; $\gamma_{failure} \in (0, 1)$;

while $(\Delta_k \geq \Delta_{\min})$ ***and*** $(k \leq k_{\max})$ **do**

Local NMPCs: Send C_i^{eq}, $\nabla_Y C_i^{eq}$, C_i^{ineq}, $\nabla_Y C_i^{ineq}$ to the coordinator;

Coordinator: Build C^{eq}, $\nabla_Y C^{eq}$, C^{ineq}, and $\nabla_Y C^{ineq}$;

Coordinator - Local NMPCs: Compute $d_{Y,\mu}$ using (3.37) with μ_{k-1};

if $(\tilde{\Phi}(0, \mu, \eta) - \tilde{\Phi}(d_{Y,\mu}, \mu, \eta) \geq 0)$ ***and*** $(h(0, \eta) - h(d_{Y,\mu}, \eta) \geq 0)$ **then**

Local NMPCs: Send $\hat{C}_i^{eq}(Y_k, \eta)$, $\nabla_Y \hat{C}_i^{eq}(Y_k, \eta)$, $\hat{C}_i^{ineq}(Y_k, \eta)$, $\nabla_Y \hat{C}_i^{ineq}(Y_k, \eta)$ to the coordinator;

Coordinator: Build $\hat{C}^{eq}(Y_k, \eta)$, $\nabla_Y \hat{C}^{eq}(Y_k, \eta)$, $\hat{C}^{ineq}(Y_k, \eta)$, $\nabla_Y \hat{C}^{ineq}(Y_k, \eta)$;

Coordinator - Local NMPCs: Perform second-order correction: compute $d_{Y,\mu}$ using (3.40) with μ_{k-1};

Coordinator - Local NMPCs: Perform Algorithm 6 with $(Y_k, \mu_{k-1}, \Delta_k, \epsilon_1, \epsilon_2)$: update μ_k, $Y_k \leftarrow Y$;

Coordinator: Calculate $\beta_k = \frac{ared_k}{pred_k} = \frac{\Phi(0,\mu,\eta) - \Phi(d_{Y,\mu},\mu,\eta)}{\tilde{\Phi}(0,\mu,\eta) - \tilde{\Phi}(d_{Y,\mu},\mu,\eta)}$;

if $(\beta_k > \beta_{\min})$ **then**

Coordinator - Local NMPCs: $Y_{k+1} = Y_k + d_{Y,\mu}$;

Coordinator: $\Delta_{k+1} = \gamma_{success} \Delta_k$;

else

Coordinator - Local NMPCs: $Y_{k+1} = Y_k$;

Coordinator: $\Delta_{k+1} = \gamma_{failure} \Delta_k$;

Coordinator: $k = k + 1$;

Output: the optimum value of problem (3.29): Y^*;

A trust-region method to solve the NLP (3.29) for a fixed value of η is given in Algorithm

7 [8]. This procedure must be performed for any decreasing sequence of η_l, namely from a selected initial value of $0 \leq \eta_0 \leq 1$ until $\eta_l \leq \epsilon$ (the termination tolerance), in order to converge to the solution of the bi-level problem (3.17). The optimum value of this procedure, illustrated in Algorithm 8, is denoted by $Y_{opt} = [p_{opt}^T, Z_{opt}^T, S_{opt}^T, \Lambda_{opt}^{eq^T}, \Lambda_{opt}^{ineq^T}]^T$.

Algorithm 8: The overall algorithm for single-level problem (3.22)

Initialization: $\eta_0 \in (0,1)$, $\beta \in (0,1)$, Y_0, Δ_0, $\epsilon_0 \in (0,1)$, $\epsilon_1 \in (0,1)$, $\epsilon_2 \in (0,1)$, $\beta_{\min} \in (0,1)$, $\gamma_{success} \geq 1$, and $\gamma_{failure} \in (0,1)$, initial penalty μ_0, the minimum trust region radius $\Delta_{\min}$, iteration number $k = 0$, maximum iteration number $k_{\max}$, coordination cycle counter $c = 0$;

while $(\eta_0 \geq \epsilon)$ **do**

  Coordinator - Local NMPCs: Perform Algorithm 7 with $(Y_0, \mu_0, \Delta_0, \Delta_{\min}, k, k_{\max}, \epsilon_1, \epsilon_2, \varrho, \gamma)$: update $Y_0 \leftarrow Y^*$;

  $\eta_{c+1} = \beta \eta_c$;

  $c = c + 1$;

Output: the optimum value of problem (3.17): $Y_{opt} = Y_0$;

3.2.1 Convergence Analysis

For a fixed value of $0 < \eta < 1$, trust-region Algorithm 7 can be regarded as a variant of Sl1QP methods [38]. Global convergence of Sl1QP methods with the penalty update procedure listed in Algorithm 6 have been studied in [13, 14]. Here, the following Lemma is stated to describe the global convergence property of Sl1QP methods with finite number of penalty updates, and a bounded value of penalty parameter μ:

Lemma 24 *[13] Suppose for a fixed $0 < \eta < 1$, Algorithm 7 is applied to (3.29) generates a bounded sequence of iterates. If penalty parameter μ is bounded, then there is a*

[8] An active-set recognition phase via linear programming (LP) may also be added to this algorithm as suggested by [14].

solution point in the sequence of iterations which is either a KKT point of the nonlinear program (3.29) or a critical point of the constraints violation $h(d_Y, \eta)$ defined in (3.35).

Therefore, if Algorithm 7 produces a feasible solution, i.e., $h(d_Y, \eta) = 0$ or $Y_k + d_Y \in \mathcal{S}_\eta$, then it follows from Lemma 24 that it converges to a KKT point of the nonlinear program (3.29). Now, the global convergence of Algorithm 8 to a unique local optimal solution of (3.17) is stated.

Theorem 25 *Suppose SCS (Assumption 16) and MFCQ (Assumption 17) hold for the lower-level CDNMPC problem (3.17). Consider the smoothed nonlinear approximation of (3.17) defined in (3.29). Then:*

(i) *For $0 < \eta < 1$, the penalty model optimization problem (3.32) or its equivalent elastic-mode problem (3.37) has a unique solution.*

(ii) *For the sequence $\eta \to 0$, Algorithm 8 is globally convergent to a unique local optimal solution of (3.17).*

Proof. (i) Suppose Assumptions 16 and 17 hold for the lower-level CDNMPC problem (3.17), and consider Lemma 24 and Theorem 20. Then, for a fixed $0 < \eta < 1$ and every feasible solution of (3.29), this problem is locally regular [33] and thus has a unique local solution. Moreover, problem (3.37) can be reduced to an equivalent convex problem (or it is convex by Remark 23). Therefore, problem (3.37) has a unique optimal solution, because $d_Y = 0$ trivially satisfies $\left|\left|d_Y\right|\right|_\infty \leq \Delta_k$, $Y_k + d_Y \in \mathcal{S}_\eta$ and only a unique price vector is associated with each local optimum (Assumption 15).

(ii) Consider Theorem 20 and the central path defined by Remark 21, then for any value of the decreasing sequence $\eta \to 0$, a bounded solution $Y_k + d_Y \in \mathcal{S}_\eta$ can be found such that forms a compact set and converges to a local optimal solution of problem (3.22) [33] (or equivalently (3.17)). According to Lemma 24, the Sl1QP algorithm variant (Algorithm 7) is globally convergent. This concludes global convergence of Algorithm 8

to a unique local optimal solution of (3.17) with a unique value for the price vector (Assumption 15) is guaranteed, under conditions stated in Theorem 25. ■

3.2.2 Stability Analysis

In this section, a stability criterion is stated without terminal costs or constraints for the proposed CDNMPC scheme based on adaptive horizon techniques [80, 41, 42]. Here an a posteriori algorithm [80], similar to section 3.1.3, is adapted to construct a dynamic programming Lyapunov function for our CDNMPC problem.

Denote Z_{opt} as the open-loop control solution of the CDNMPC problem, and the receding horizon control (RHC) move as $u_{opt_{i,RHC}}(N, x_{opt_i}(n))$. This move is calculated by solving the problem (3.17) with prediction horizon length N and initial value $x_i(n)$.

Assuming that ϵ_0 is chosen sufficiently small, define a dynamic programming value function [80] for the CDNMPC scheme based on a predefined stage cost l^D:

$$V_N^D\big(x(n)\big) = \sum_{n=0}^{N-1} l^D\Big(x_{opt}(n), u_{opt_{RHC}}(N-n, x(n))\Big) \tag{3.41a}$$

where the stage cost of the centralized trajectory is defined as:

$$l^D\Big(x_{opt}(n), u_{opt_{RHC}}(N-n, x(n))\Big) = \textstyle\sum_{i=1}^{m} \big(x_{opt_i}(n) - x_{i,sp}(t_k)\big)^T Q_{ii}(n)\big(x_{opt_i}(n) - x_{i,sp}(t_k)\big)$$

$$+\big(u_{opt_{i,RHC}}(N-n, x_i(n)) - u_{i,sp}(t_k)\big)^T R_{ii}(n)\big(u_{opt_{i,RHC}}(N-n, x_i(n)) - u_{i,sp}(t_k)\big) \tag{3.41b}$$

Therefore, in order to calculate V_N^D at the current sampling time t_k, $V_N^D\big(x(t_k)\big)$, a multi-step calculation has to be performed forward in time. The procedure to calculate (3.41) is listed in Algorithm 9:

Algorithm 9: The algorithm to calculate the finite-time value function (3.41)

Initialization: prediction horizon N; number of sub-systems m; initial state $x_{opt}(n)$; Q_{ii}; R_{ii}; $V_N^D = 0$;

for $n := 0$ ***to*** $N-1$ **do**

Coordinator - Local NMPCs: Perform Algorithm 8 with $N-n$ and initial value $x_i(0) = x_{i,opt}(n)$;

Coordinator - Local NMPCs: Update the RHC control action move $u_{opt_{RHC}}(N-n, x_{opt}(n))$;

Coordinator: Calculate $l^D\Big(x_{opt}(n), uopt_{RHC}(N-n, x(n))\Big)$, and $V_N^D = V_N^D + l^D\Big(x_{opt}(n), u_{opt_{RHC}}(N-n, x(n))\Big)$ using (3.41);

Local NMPCs: Apply $u_{opt_{RHC}}(N-n, x(n))$ to the internal model;

Local NMPCs: Calculate $x_{opt}(n+1)$;

Output: the value function $V_N^D(x(n)) \leftarrow V_N^D$;

Remark 26 *Similar to Lemma 11, consider the individual distributed NMPC controllers in the optimization problem (3.17) at sampling time t_k, provided that the stage cost l^D is a positive definite function and ϵ_0 is sufficiently small, if there exists a trajectory-based $\alpha \in [0,1]$ such that:*

$$V_N^D(x(t_k)) - V_N^D(x(t_{k+1})) \geq \alpha l^D\Big(x_{opt}(t_k), u_{opt_{RHC}}(N, x(t_k))\Big) \tag{3.42}$$

then V_N^D is a finite-time Lyapunov function for the CDNMPC system (3.17) and α is a rough estimate of closed-loop sub-optimality degree w.r.t. the corresponding infinite horizon plant-wide problem (3.17). Note that all NMPC controllers in (3.17) are assumed to have the same number of prediction horizons as is dictated by the minimum required horizon length to satisfy (3.42), namely $\alpha \geq \alpha_{\min} \geq 0$.

Based on Remark 26, an a posteriori algorithm is presented to calculate the required

horizon length of N that ensures closed-loop stability of the CDNMPC problem (3.17). This procedure is explained in Algorithm 10:

Algorithm 10: A posteriori adaptive horizon algorithm for the CDNMPC scheme

Initialization: prediction horizon N; number of sub-systems m; initial state $x_{opt}(n)$; Q_{ii}; R_{ii};

Coordinator - Local NMPCs: Calculate $V_N^D(x(n))$ using Algorithm 9 with $\big(N, m, x_{opt}(n), Q_{ii}, R_{ii}\big)$;

Local NMPCs: Apply $u_{opt_{RHC}}(N, x(n))$ to the internal model;

Local NMPCs: Calculate $x_{opt}(n+1)$;

Coordinator - Local NMPCs: Calculate $V_N^D(x(n+1))$ using Algorithm 9 with $\big(N, m, x_{opt}(n+1), Q_{ii}, R_{ii}\big)$;

if $V_N^D(x(n)) \leq V_N^D(x(n+1))$ **then**

  Local NMPCs: Prompt: Solution may be unstable;

else

  Coordinator: Calculate $\alpha = \frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x_{opt}(n), u_{opt_{RHC}}(N, x(n))\right)}$;

  if $\alpha \geq \alpha_{min}$ **then**

    Coordinator - Local NMPCs: Perform Algorithm 11;

  else

    Coordinator - Local NMPCs: Perform Algorithm 12;

Output: Y_{opt} based on the accepted value of N;

Similar to Section 3.1.3, if the calculated $\alpha_N \geq \alpha_{\min}$ the prediction horizon length is decreased so that the minimum amount of prediction horizon, which satisfies this condition, is found. This is usually denoted as the horizon shortening strategy, described in Algorithm 11:

Algorithm 11: Horizon Shortening Algorithm for the CDNMPC problem (3.17)

Initialization: $\alpha_{\min}$; prediction horizon N; number of sub-systems m; initial state $x_{opt}(n)$; Q_{ii}; R_{ii};

while $N \geq 2$ **do**

Local NMPCs: Save the current trajectory of the open-loop system;

Local NMPCs: $N = N - 1$;

Coordinator - Local NMPCs: Calculate $\alpha = \frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x_{opt}(n), u_{RHC}(N, x(n))\right)}$ with $\left(N, m, x_{opt}(n), Q_{ii}, R_{ii}\right)$;

Local NMPCs: Save the shortened horizon trajectory of the open-loop system;

if $\alpha < \alpha_{\min}$ **then**

Local NMPCs: $N = N + 1$;

Local NMPCs: Restore the current stored trajectory of the open-loop system;

STOP;

else

Local NMPCs: Save the shortened horizon trajectory as the current trajectory;

Output: Current trajectory of the system, and N.

Conversely, if $\alpha_N < \alpha_{\min}$ is increased the prediction horizon to find the minimum N that satisfies $\alpha_N \geq \alpha_{\min}$, described in algorithm 12:

Algorithm 12: Horizon Prolongation Algorithm for the CDNMPC problem (3.17)

Initialization: $\alpha_{\min}$; prediction horizon N; number of sub-systems m; initial state $x_{opt}(n)$; Q_{ii}; R_{ii};

while $\alpha < \alpha_{\min}$ **do**

Local NMPCs: $N = N + 1$;

Coordinator - Local NMPCs: Calculate $\alpha = \frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x_{opt}(n), u_{RHC}(N, x(n))\right)}$ with $\left(N, m, x_{opt}(n), Q_{ii}, R_{ii}\right)$;

Local NMPCs: Save the prolonged size trajectory as the current trajectory of the open-loop system;

Output: Current trajectory of the system, and N.

3.2.3 The overall CDNMPC Algorithm

In this section, an overall summary of the CDNMPC algorithm, applied to the problem (3.15), is presented. Based on this approach, the hierarchical dual decomposition of the hypothetical centralized controller defined in (3.7) is related with the modification applied to the corresponding existing network of local decentralized NMPC controllers. The modification applied to the existing local NMPC controllers is equivalent to penalizing the local versions of the overall interaction error vector using the price vector. The local NMPC controllers formulate their own nonlinear programming problems and exchange the local sensitivity information, as required, with the coordinator. A wrapper procedure, Algorithm 10, is responsible to ensures stability of the closed-loop system by finding the minimum required length of prediction horizon. Local NMPCs, then apply the receding horizon control actions to the plant with the optimal variables. The overall algorithm of the proposed CDNMPC scheme is given in Algorithm 13:

Algorithm 13: The overall algorithm to solve the CDNMPC problem (3.15)

Initialization: number of sub-systems m, initial state $x(t_k)$, positive definite matrix Q, positive definite matrix R, $\alpha_{\min} \in (0,1)$, initial prediction horizon N_0, initial penalty μ_0, initial trust-region radius Δ_0, the minimum trust region radius $\Delta_{\min}$, maximum iteration number $k_{\max}$, $\epsilon_0 \in (0,1)$, $\beta \in (0,1)$, $\beta_{\min} \in (0,1)$, $\gamma_{success} \geq 1$, and $\gamma_{failure} \in (0,1)$;

Coordinator-Local NMPCs: Perform Algorithm 8 with $(x(t_k), \eta_0, \mu_0, \Delta_0, \Delta_{\min}, k_{\max}, \epsilon_0, \beta, \beta_{\min}, \gamma_{success}, \gamma_{failure})$ to calculate Y_{opt};

Coordinator-Local NMPCs: Perform Algorithm 10 with $(\alpha_{\min}, N0, m, x(t_k), Q, R)$ to calculate N and update Y_{opt};

Local NMPCs: Calculate $x(t_{k+1})$ by applying the RHC action to the plant;

Local NMPCs: Update $N_0 = N$;

Output: $x(t_{k+1})$, and N;

3.3 Two-CSTR Process Case Study

In this section, the centralized NMPC scheme (3.7) and the CDNMPC scheme (3.17) are implemented on a two continuous stirred tank reactor (CSTR) benchmark problem, which was taken from [96]. The system is comprised of two non-isothermal reactive CSTRs with interconnections and a recycle stream as depicted in Figure 3.2. The following set of exothermic reactions take place in these reactors with substances A and B: (i) A $\xrightarrow{k_1} B$, (ii) A $\xrightarrow{k_2} UP$, and (iii) A $\xrightarrow{k_3} DP$; where UP and DP stand for the undesired product and the desired product, respectively. As in Figure 3.2, CSTR I has two feed streams (one has fresh stream of substance 'A' with molar concentration C_{A0}, flow-rate F_0 and temperature T_0, and the other streams is from the output of CSTR II containing the recycle stream of unreacted substance A at flow-rate F_r, molar concentration C_{A2} and

temperature T_2) and CSTR II is fed from the output of CSTR I as well as another fresh stream of substance A at flow-rate F_3, molar concentration CA_{03}, and temperature T_{03}. The two CSTRs are equipped with jackets to remove/provide heat, due to non-isothermal nature of reactions. In [96], the following set of ODEs are developed for transient material and energy balances:

$$\frac{dT_1}{dt} = \frac{F_0}{\mathcal{V}_1}\big(T_0 - T_1\big) + \frac{F_r}{\mathcal{V}_1}\big(T_2 - T_1\big) + \sum_{i=1}^{3} \mathcal{G}_i(T_1)C_{A1} + \frac{Q_1}{\rho c_p \mathcal{V}_1} \tag{3.43a}$$

$$\frac{dC_{A1}}{dt} = \frac{F_0}{\mathcal{V}_1}\big(C_{A0} - C_{A1}\big) + \frac{F_r}{\mathcal{V}_1}\big(C_{A2} - C_{A1}\big) + \sum_{i=1}^{3} \mathcal{R}_i(T_1)C_{A1} \tag{3.43b}$$

$$\frac{dT_2}{dt} = \frac{F_1}{\mathcal{V}_2}\big(T_1 - T_2\big) + \frac{F_3}{\mathcal{V}_2}\big(T_{03} - T_2\big) + \sum_{i=1}^{3} \mathcal{G}_i(T_2)C_{A2} + \frac{Q_2}{\rho c_p \mathcal{V}_2} \tag{3.43c}$$

$$\frac{dC_{A2}}{dt} = \frac{F_1}{\mathcal{V}_2}\big(C_{A1} - C_{A2}\big) + \frac{F_3}{\mathcal{V}_2}\big(C_{A03} - C_{A2}\big) + \sum_{i=1}^{3} \mathcal{R}_i(T_2)C_{A2} \tag{3.43d}$$

where: $\mathcal{R}_i(T_j) = k_{i0}e^{-\frac{E}{RT_i}}$ and $\mathcal{G}_i(T_j) = -\frac{\Delta H_i}{\rho c_p}\mathcal{R}_j(T_j)$, T_j denote the temperature of the reactor, C_{Aj} denotes the concentration of 'A', Q_j denotes the rate of heat input to the reactor, and $\mathcal{V}_j$ denotes the reactor volume, for $j = 1, 2$. In addition: H_i denotes the enthalpies, k_i denotes exponential coefficients and E_i denotes the activation energies of the three reactions, for $i = 1, 2, 3$; c_p is the heat capacity and ρ is the density of fluid in the reactor. The values of model parameters are listed in Table 3.1 [96].

The open-loop system has three steady-states. Two of them are locally asymptotically stable but one is unstable. The unstable steady state occurs at $(T_1^s, C_{A1}^s, T_2^s, C_{A2}^s)$= $(457.9[K], 1.77[kmol/m^3], 415.5[K], 1.75[kmol/m^3])$, for the plant with $Q_1 = Q_2 = 0$, $C_{A0} = C_{A0}^s$, $C_{A03} = C_{A03}^s$ and a recycle ratio of $r = 0.5$. The control objective is to stabilize the system around the open-loop unstable steady-state operating point. This avoids high temperatures, while simultaneously achieving reasonable conversion. The manipulated variables of the system are the heat input rates Q_1 and Q_2, and the inlet

Parameter	Value
F_0	4.998 $[m^3/h]$
F_1	39.996 $[m^3/h]$
F_3	30.0 $[m^3/h]$
F_r	34.998 $[m^3/h]$
$\mathcal{V}_1$	1.0 $[m^3]$
$\mathcal{V}_2$	3.0 $[m^3]$
R	8.314 $[kJ/kmolK]$
T_0	300 $[K]$
T_{03}	300 $[K]$
C^s_{A0}	4.0 $[kmol/m^3]$
C^s_{A03}	2.0 $[kmol/m^3]$
ΔH_1	-5.0×10^4 $[kJ/kmol]$
ΔH_2	-5.2×10^4 $[kJ/kmol]$
ΔH_3	-5.4×10^4 $[kJ/kmol]$
k_{10}	3.0×10^6 $[h^{-1}]$
k_{20}	3.0×10^5 $[h^{-1}]$
k_{30}	3.0×10^5 $[h^{-1}]$
$E1$	5.0×10^4 $[kJ/kmol]$
$E2$	7.53×10^4 $[kJ/kmol]$
$E3$	7.53×10^4 $[kJ/kmol]$
ρ	1000.0 $[kg/m^3]$
c_p	0.231 $[kJ/kgK]$
T^s_1	457.9 $[K]$
C^s_{A1}	1.77 $[kmol/m^3]$
T^s_2	415.5 $[K]$
C^s_{A2}	1.75 $[kmol/m^3]$

Table 3.1: The Two-CSTR process parameters

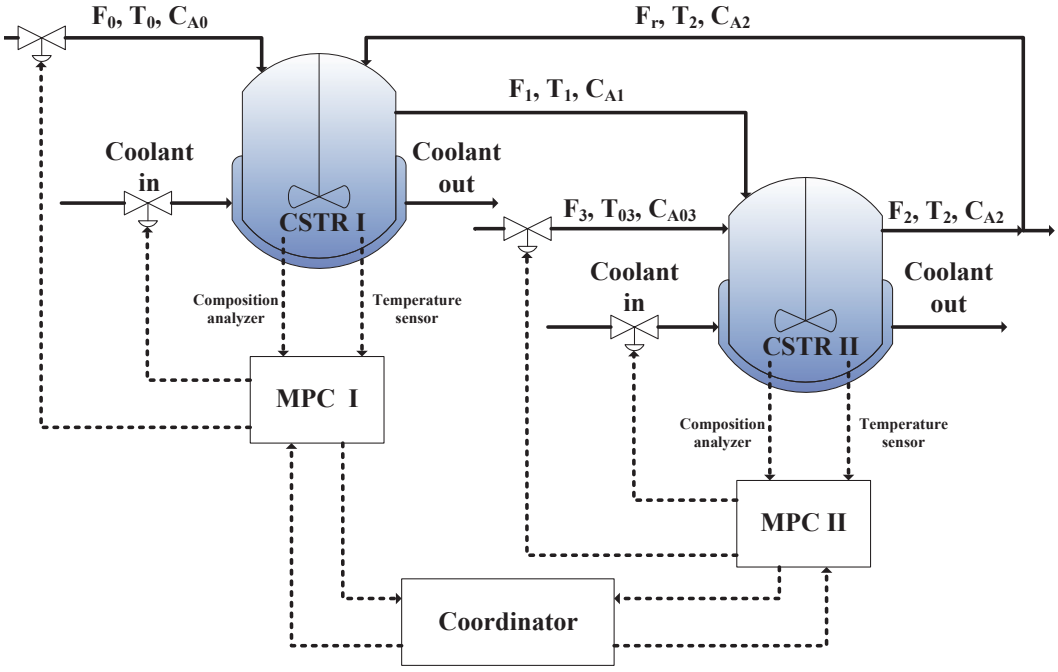


Figure 3.2: Schematic of the Two CSTR Case Study

concentrations C_{A0} and C_{A03}.

The ODE system (3.43a) can be decomposed into two subsystems by defining interaction variables v_1 and v_2 for sub-system I (containing CSTR I) and subsystem II (containing CSTR II) as:

$$\text{Sub-system I:} \begin{cases} \frac{dT_1}{dt} &= \frac{F_0}{\mathcal{V}_1}\big(T_0 - T_1\big) + \frac{F_r}{\mathcal{V}_1}\big(v_{1,1} - T_1\big) + \sum_{i=1}^{3} \mathcal{G}_i(T_1) C_{A1} + \frac{Q_1}{\rho c_p \mathcal{V}_1} \\ \frac{dC_{A1}}{dt} &= \frac{F_0}{\mathcal{V}_1}\big(C_{A0} - C_{A1}\big) + \frac{F_r}{\mathcal{V}_1}\big(v_{1,2} - C_{A1}\big) + \sum_{i=1}^{3} \mathcal{R}_i(T_1) C_{A1} \end{cases} \tag{3.44a}$$

$$\text{Sub-system II:} \begin{cases} \frac{dT_2}{dt} &= \frac{F_1}{\mathcal{V}_2}\big(v_{2,1} - T_2\big) + \frac{F_3}{\mathcal{V}_2}\big(T_{03} - T_2\big) + \sum_{i=1}^{3} \mathcal{G}_i(T_2) C_{A2} + \frac{Q_2}{\rho c_p \mathcal{V}_2} \\ \frac{dC_{A2}}{dt} &= \frac{F_1}{\mathcal{V}_2}\big(v_{2,2} - C_{A2}\big) + \frac{F_3}{\mathcal{V}_2}\big(C_{A03} - C_{A2}\big) + \sum_{i=1}^{3} \mathcal{R}_i(T_2) C_{A2} \end{cases} \tag{3.44b}$$

where v_1 and v_2 replace informations of sub-systems II and sub-system I, respectively.

To obtain the discretized model, the fifth order Radau IIA method is used, which allows the internal coefficients of the IRK method be expressed in a Butcher's tableau [44]:

$$\begin{array}{c|ccc} c^1 & a_1^1 & \cdots & a_s^1 \\ \vdots & \vdots & & \vdots \\ c^s & a_1^s & \cdots & a_s^s \\ \hline & b^1 & \cdots & b^s \end{array} \tag{3.45a}$$

or:

$$\begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array} \tag{3.45b}$$

Note that, in this paper, the real-time solutions are obtained using ACADO toolkit [48].

The interaction error over a single prediction horizon, $e(t_{k+l}|t_k)$, can be formulated as:

$$e(t_{k+l}|t_k) = \begin{bmatrix} v_{1,1}(t_{k+l}|t_k) - T_2(t_{k+l}|t_k) \\ v_{1,2}(t_{k+l}|t_k) - C_{A2}(t_{k+l}|t_k) \\ v_{2,1}(t_{k+l}|t_k) - T_1(t_{k+l}|t_k) \\ v_{2,2}(t_{k+l}|t_k) - C_{A1}(t_{k+l}|t_k) \end{bmatrix} \tag{3.46a}$$

$$= e_1(t_{k+l}|t_k) + e_2(t_{k+l}|t_k) \tag{3.46b}$$

$$= \begin{bmatrix} v_{1,1}(t_{k+l}|t_k) \\ v_{1,2}(t_{k+l}|t_k) \\ -T_1(t_{k+l}|t_k) \\ -C_{A1}(t_{k+l}|t_k) \end{bmatrix} + \begin{bmatrix} -T_2(t_{k+l}|t_k) \\ -C_{A2}(t_{k+l}|t_k) \\ v_{2,1}(t_{k+l}|t_k) \\ v_{2,2}(t_{k+l}|t_k) \end{bmatrix} \tag{3.46c}$$

$$\begin{aligned} = & \left[\begin{array}{cccc|cc} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{array}\right] \times \begin{bmatrix} T_1(t_{k+l}|t_k) \\ C_{A1}(t_{k+l}|t_k) \\ Q_1(t_{k+l}|t_k) \\ C_{A0}(t_{k+l}|t_k) \\ v_{1,1}(t_{k+l}|t_k) \\ v_{1,2}(t_{k+l}|t_k) \end{bmatrix} + \\ & \left[\begin{array}{cccc|cc} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \times \begin{bmatrix} \bar{T}_2(t_{k+l}|t_k) \\ \bar{C}_{A2}(t_{k+l}|t_k) \\ \bar{Q}_2(t_{k+l}|t_k) \\ \bar{C}_{A03}(t_{k+l}|t_k) \\ V_{2,1}(t_{k+l}|t_k) \\ V_{2,2}(t_{k+l}|t_k) \end{bmatrix} \end{aligned} \tag{3.46d}$$

According to (3.13), the overall interaction error (over the whole prediction horizon) can be expressed, using (3.46), as:

$$E(t_k) = \sum_{i=1}^{2} E_i(t_k) \tag{3.47a}$$

$$= \Theta_1 \times \begin{bmatrix} \bar{T}_1(t_k) \\ \bar{C}_{A1}(t_k) \\ \bar{Q}_1(t_k) \\ \bar{C}_{A0}(t_k) \\ V_{1,1}(t_k) \\ V_{1,2}(t_k) \end{bmatrix} + \Theta_2 \times \begin{bmatrix} \bar{T}_2(t_k) \\ \bar{C}_{A2}(t_k) \\ \bar{Q}_2(t_k) \\ \bar{C}_{A03}(t_k) \\ V_{2,1}(t_k) \\ V_{2,2}(t_k) \end{bmatrix} \tag{3.47b}$$

where, for example, $\bar{T}_1 = [\hat{T}(t_{k+1}|t_k)^T, \cdots, \hat{T}(t_{k+N}|t_k)^T]^T$ denotes the stacked vector of predicted temperature profile $\hat{T}$ in CSTR I, over the whole prediction horizon at sampling time t_k, i.e. T. Based on variable bounds and operating conditions of the system, the open-loop model is scaled between 0.1 and 1 to simplify the procedure. The controller parameters and the variable bounds of the these two sub-systems are listed in Table 3.2. Additionally, the CDNMPC algorithm is executed with $\eta_0 = 10^{-3}; \beta = 0.5$; penalty parameter $\mu = 1$; $\Delta_0 = 10^{-2}$ the minimum trust region radius $\Delta_{\min} = 10^{-4}$; maximum iteration number $k_{\max} = 20$; termination tolerance $\epsilon = 10^{-12}$; $\beta_{\min} = 0.1$; $\gamma_{success} = 1.1$; and $\gamma_{failure} = 0.5$.

	NMPC I	NMPC II
Initial Conditions:	$x_1(0) = \big[462.9[K], 0.27[kmol/m^3]\big]$	$x_2(0) = \big[410.5[K], 3.45[kmol/m^3]\big]$
Weighting Matrices:	$Q_1 = 100I,\ R_1 = I$	$Q_2 = 100I,\ R_2 = I$
Upper bounds:	$lb_{u1} = \big[5[kJ/hr], 8[kmol/m^3]\big]$	$lb_{u2} = \big[5[kJ/hr], 4[kmol/m^3]\big]$
Lower Bounds:	$ub_{u1} = \big[-5[kJ/hr], 0[kmol/m^3]\big]$	$ub_{u2} = \big[-5[kJ/hr], 0[kmol/m^3]\big]$
Sampling Period	$0.003[s]$	$0.003[s]$

Table 3.2: The Two-CSTR process variable bounds and controller parameters

The continuous-time open-loop system is discretized between two consecutive sampling times via the Radau IIA IRK method. The MATLAB interface to ACADO toolkit [48] is used to perform the discretization with 100 number of integration steps. The optimization problems for the centralized NMPC (3.7) and CDNMPC (3.17) are

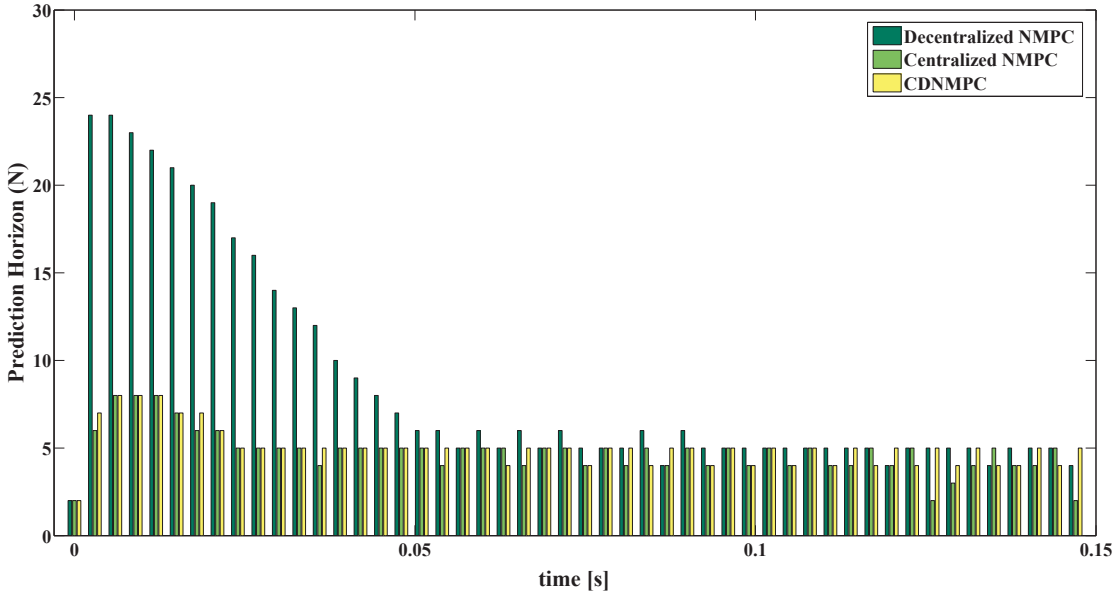
implemented in MATLAB using YALMIP optimization interface [62]. IPOPT [6] is deployed as the main optimization solver for our quadratic programming and nonlinear programming sub-problems. The Intel(R) Math Kernel Library (MKL) is used for online differentiation purposes. Simulations are performed on an intel core-i7 processor with 8 GB of memory under Microsoft Windows 7 operating system.

The adaptive horizon scheme for ensuring the stability of closed-loop systems defined in this paper is tested for three values of $\alpha_{\min} = 0.2$, 0.4, and 0.7. The comparison of the required number of predictions horizons (N) between the decentralized NMPC, the centralized NMPC and the CDNMPC trajectories is depicted in Figure 3.3. This comparison shows that, after a certain amount of simulation time, the three schemes require roughly the same number of prediction horizons to ensure the stability; however the decentralized NMPC schemes demands the highest number of prediction horizons, specially during the first half of the simulation time. As expected from the CDNMPC algorithm, this networked control system is acting similarly to an equivalent monolithic centralized NMPC in terms of their stability criteria.

Based on the three sets of values for the parameter $\alpha_{\min}$, observe that a higher number of prediction horizons is required as this parameter is increased. As discussed before, there has to be a compromise between the expected performance of controllers and the computational cost of using this adaptive scheme. Based on the simulation results, $\alpha_{\min} = 0.4$ is a good choice between the three suggested values because the settling time of the closed-loop systems is satisfactory and, at the same time, the profile of required 'N' is smoother than $\alpha_{\min} = 0.2$ while not being as computationally expensive as $\alpha_{\min} = 0.7$.

The state and manipulated input trajectories of the three control scheme, for $\alpha_{\min} = 0.4$, are shown in Figures 3.4 and 3.5, respectively. In addition, the scaled overall objective functions of the three distributed control schemes J_{DC} defined in (3.16), J_C defined in (3.7), and J_D defined in (3.15) are compared in Figure 3.6. The mean square error

(a) $\alpha_{\min} = 0.2$



(b) $\alpha_{\min} = 0.4$

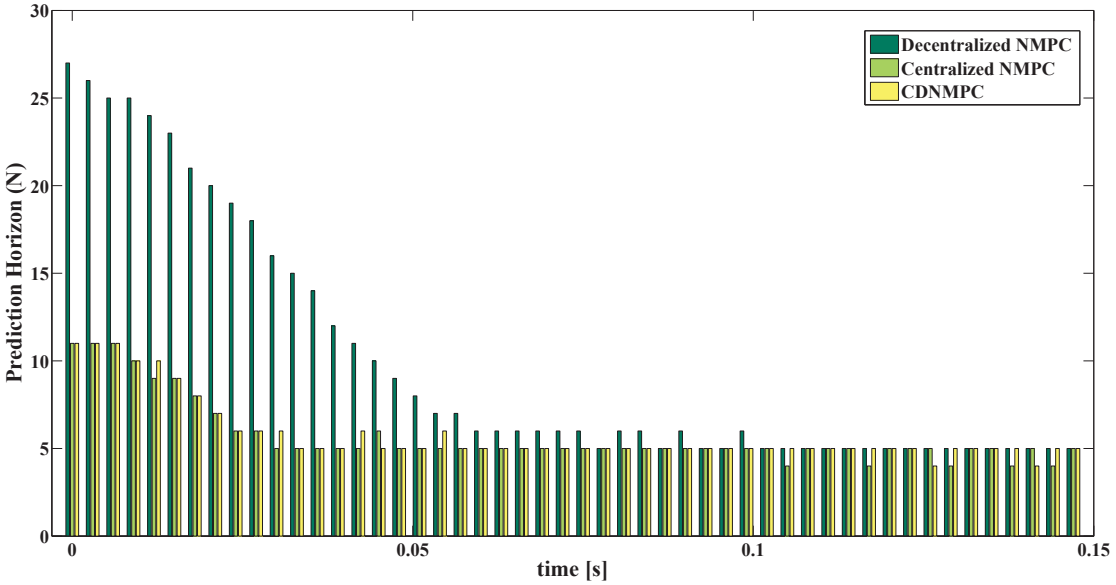


Figure 3.3: Comparison of number predictions horizons needed (N) between the Decentralized NMPC, the Centralized NMPC and the CDNMPC trajectories

(c) $\alpha_{\min} = 0.7$

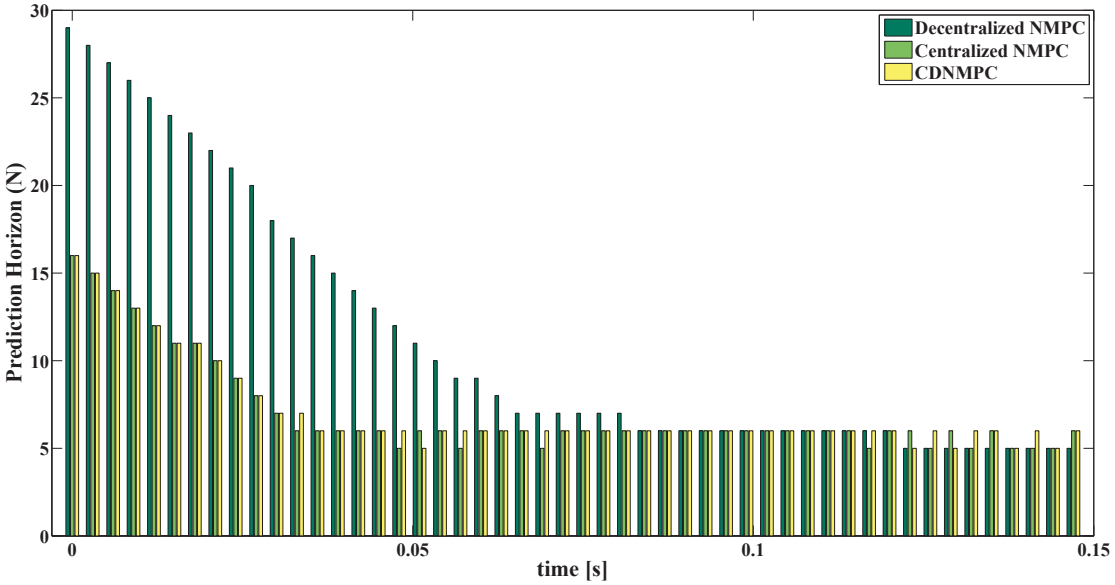


Figure 3.3: [continued] Comparison of number predictions horizons needed (N) between the Decentralized NMPC, the Centralized NMPC and the CDNMPC trajectories

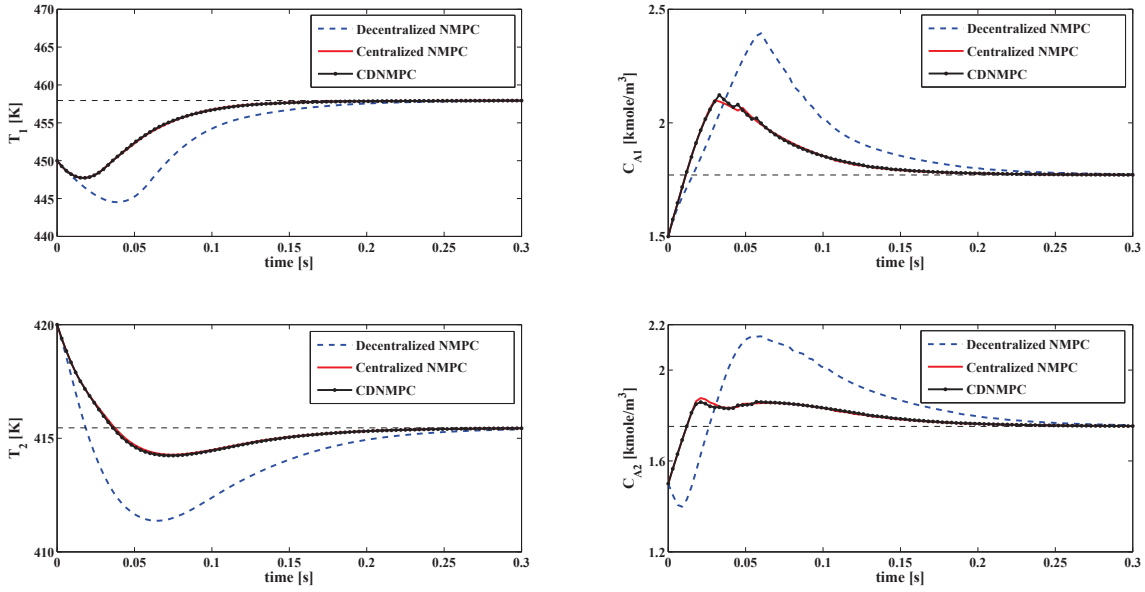


Figure 3.4: Comparison of state variables trajectories between the Decentralized NMPC, the Centralized NMPC and the CDNMPC schemes

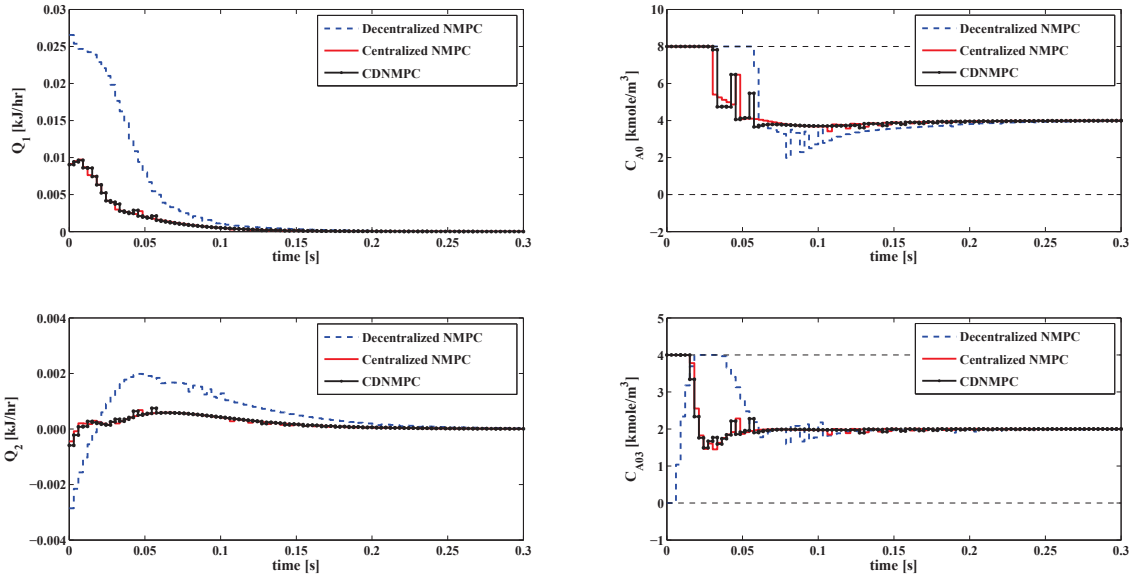


Figure 3.5: Comparison of manipulated input variables trajectories between the Decentralized NMPC, the Centralized NMPC and the CDNMPC schemes

between the objective function of the CDNMPC and the objective function of the centralized NMPC is at least four orders of magnitude smaller than that of the decentralized NMPC. These show that the CDNMPC is capable of achieving performance equivalent to a centralized NMPC. In the proposed approach, it improves the decentralized NMPC scheme by reducing the number of required prediction horizons too ensure stability and the overall performance of the system by converging to a trajectory equivalent to the centralized NMPC trajectory for the plant.

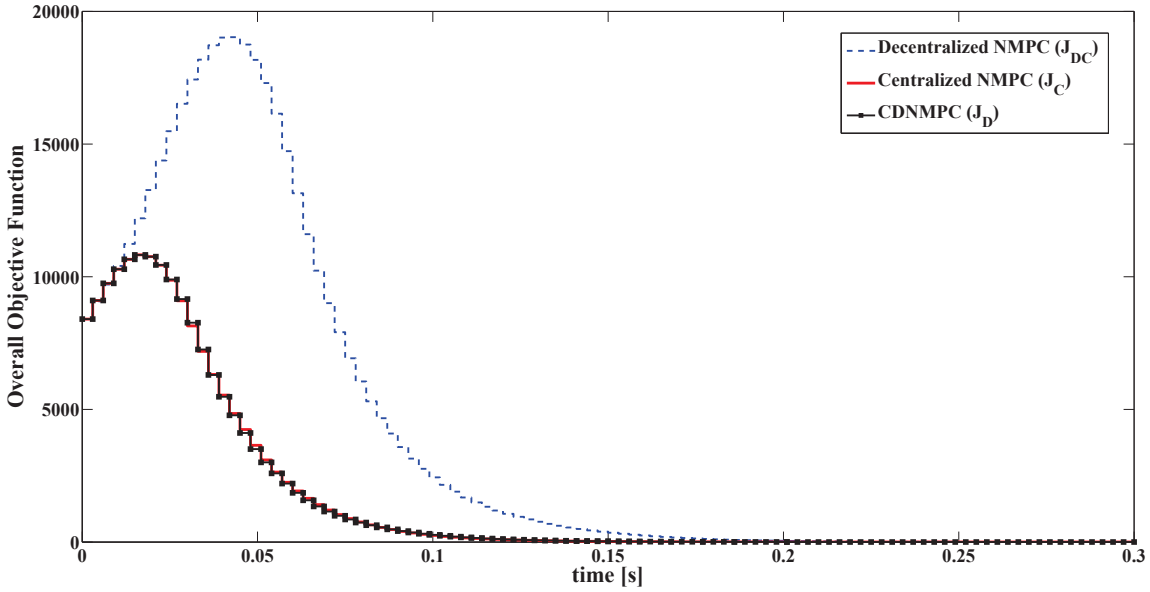


Figure 3.6: Comparison of the overall objective functions between the Decentralized NMPC, the Centralized NMPC and the CDNMPC schemes

3.4 Conclusion

In this chapter, a price-driven coordination distributed NMPC scheme is proposed for continuous-time nonlinear DAE systems. In such a scheme, the coordination level is designed as an upper level optimization module and the local NMPC controllers construct the lower level structure. The bi-level CDNMPC problem is relaxed into a series of quadratic programming problems subject to a predefined trust-region radius. Sufficient conditions for global convergence of the algorithm to a unique local solution are derived. In addition, an adaptive prediction horizon selection algorithm is implemented at the coordination level to ensure the finite-time closed-loop stability of entire plant w.r.t. the infinite horizon optimality, when the overall interaction error is sufficiently small. Nevertheless, the minimum required horizon length is a compromise between the associated computational cost and achieving the performance of infinite horizon NMPC controllers over the whole plant.

Chapter 4

A Bi-level Optimization Approach To Price-driven Coordination of Distributed Model Predictive Control Systems

In this chapter, two algorithms are proposed to coordinate distributed model predictive control (DMPC) systems. The main idea behind the coordinated distributed model predictive (CDMPC) is to improve the performance of an existing decentralized MPC system with a minor modification applied to the network. The CDMPC problem can be interpreted as a bilevel optimization problem, which consists of modified local controllers and a coordination level that ensures optimal centralized behavior of the plant. Modification of the decentralized MPC controllers would be equivalent to relaxing local versions of the overall interaction constraint using a price vector to penalize violations. The proposed algorithms solve the CDMPC problem using the price-driven approach. The first algorithm is an analytic approach that derives the solution to plant-wide problem in absence of inequality constraints in local controllers. On the other hand, the second algorithm is an iterative approach that determines a local optimal solution to the general CDMPC problem via finding feasible directions for local variables and the price vector, in each iteration. Unlike nested price-driven CDMPC approaches, the proposed CDMPC algorithms

are globally convergent and are capable of stabilizing open-loop unstable dynamics. The effectiveness of proposed algorithms is illustrated using two chemical process examples.

4.1 Introduction

Traditionally, centralized and decentralized control are the two primary frameworks for the control of large-scale systems. While the decentralized control is easy to implement, it may lead to degradation of plant-wide performance or even loss of closed-loop stability since in decentralized control the interactions between subsystems are in general neglected. On the other hand, centralized control is expected to give the best performance; however, it may become too complicated to implement as the size of the control problem grows. These considerations motivate the significant efforts in the control of large-scale systems. One important class of DMPC systems is called cooperative DMPC; some of the important recent work in this category includes [84, 94], and Lyapunov-based sequential DMPC (e.g., [61, 60]). Another important type of DMPC, pertaining to the present work, is coordinated DMPC (CDMPC), which strives to reach the optimal plan-wide performance by adding a coordination level to a currently installed decentralized network.

In a CDMPC scheme, distributed MPCs communicate with a coordinator to achieve improved performance. This problem can be interpreted as a bi-level programming (BLP) problem, which consists of modified local controllers and a coordinator to ensure optimal centralized plant behavior is achieved. Different algorithms have been developed for CDMPC including price-driven approach (e.g., [20, 21, 76, 57]), primal decomposition approach (e.g., [35, 25]), prediction-driven approach (e.g., [66, 76]), chanced-constrained coordination method [76], and pseudo-model coordination approach [76]. The main differences between these coordination methods are the ways that the interactions between

the subsystems are addressed [76].

Usually, CDMPC schemes are solved using nested approaches, in which MPC controllers solve their local problems for a fixed value of price and communications between the local controllers and the coordinator are repeated until an optimal price is determined, which coincides with the solution to the centralized problem. However, nested approaches may exhibit convergence problems [76]. In this work, the price-driven CDMPC problem is solved with a BLP approach to overcome such convergence problems. The two levels of this BLP problem consist of the local MPC controllers as the lower-level problem and the coordinator as the upper-level problem. In this scope, the lower-level problems are convex for fixed values of the price vector. Several solution algorithms for BLP problems with lower-level convex formulation have been proposed in [9, 29, 27]. These solution methods can be categorized as: enumeration (branch-and-bound), evolutionary-based, penalty-based, and descent feasible direction methods.

In the branch-and-bound approach, the lower-level problem is replaced by the equivalent Karush Kuhn-Tucker (KKT) system. This is the idea used in the approaches of Edmunds and Bard [32], Al-Khayyal et al. [4] and more recently Thoai et al. [97]. Similarly, in evolutionary methods, the bi-level problem is transformed into a single level non-convex problem and then, several evolutionary heuristic methods such as genetic algorithm [9] are applied to find a global optimum. The penalty method category also attempts to compute stationary point and local optimal solutions to nonlinear BLP problems [1, 2, 92, 24].

Descent approaches, on the other hand, attempt to find local optima of the BLP problems by defining an implicit relation between the lower-level variable(s) as a function of upper-level variable(s). In this method, it is assumed that the there is a unique upper-level variable for the optimal solution of the lower-level problem. The major issue is to find a decreasing feasible direction, based on the gradient of the upper-level objective

function. Several approximations of the gradient information are proposed by Kolstad and Lasdon [54] and Savard and Gauvin [87]. In this category, Falk and Liu [34] presented a bundle method where the decrease of the upper-level objective is adapted according to sub-gradient information of the lower-level problem.

In this chapter, two different scenarios of CDMPC problem are considered. In the first scenario, where all of the constraints are active at all times, an analytic method is presented. Accordingly, a closed-form solution is derived for the price vector and local variables which exactly leads to the corresponding optimal centralized solution. In the second scenario, an iterative approach is suggested based on finding a descent feasible direction to solve the general CDMPC problem with inequality constraints in the local controllers. For both of the scenarios, it is assumed that an implicit relation exists between the variables of local MPCs and variables belonging to the coordinator, namely the price vector. In addition, global convergence and closed-loop stability of the proposed schemes are proved. Performance of the proposed CDMPC algorithms is illustrated via the applications to a forced-circulation evaporation process and a two-CSTR process.

4.2 Preliminaries

In this work, the entire plant is considered to be composed of m interconnected subsystems with the following state-space representation:

$$\dot{x}(t) = \Psi_c x(t) + \Gamma_c u(t) \tag{4.1}$$

where $x = [x_1^T \cdots x_i^T \cdots x_m^T]^T \in R^{n_x}$ denotes the vector of state variables, and $u = [u_1^T \cdots u_i^T \cdots u_m^T]^T \in R^{n_u}$ is the vector of manipulated input variables. Then, Ψ_c and Γ_c

are state and input coefficient matrices, defined as:

$$\Psi_c = \begin{bmatrix} \Psi_{c_{11}} & \cdots & \Psi_{c_{1j}} & \cdots & \Psi_{c_{1m}} \\ \vdots & \ddots & & & \vdots \\ \Psi_{c_{i1}} & \cdots & \Psi_{c_{ii}} & \cdots & \Psi_{c_{im}} \\ \vdots & & & \ddots & \vdots \\ \Psi_{c_{m1}} & \cdots & \Psi_{c_{mi}} & \cdots & \Psi_{c_{mm}} \end{bmatrix}, \quad \Gamma_c = \begin{bmatrix} \Gamma_{c_{11}} & \cdots & \Gamma_{c_{1j}} & \cdots & \Gamma_{c_{1m}} \\ \vdots & \ddots & & & \vdots \\ \Gamma_{c_{i1}} & \cdots & \Gamma_{c_{ii}} & \cdots & \Gamma_{c_{im}} \\ \vdots & & & \ddots & \vdots \\ \Gamma_{c_{m1}} & \cdots & \Gamma_{c_{mi}} & \cdots & \Gamma_{c_{mm}} \end{bmatrix} \tag{4.2}$$

In this model, the subsystems are interconnected through states and manipulated input variables, (i.e., the manipulated variable of one sub-system might affect other sub-systems as an internal variable within the models). In addition, it is assumed that x_i and u_i belong to a local set of convex constraints $\mathcal{C}_i$ defined as follows:

$$\mathcal{C}_i \triangleq \{(x_i, u_i) \in \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{n_{u_i}} | g_i(x_i, u_i) \leq 0\} \tag{4.3}$$

for $i = 1, \ldots, m$ and g_i is the vector of convex functions that contains mixed constraints, and bounds over local states and manipulated input variables of sub-systems. For the purpose of control, the continuous-time system (4.1) is exactly discretized, with sampling time T, to form the following discrete-time model:

$$x(k+1) = \Psi x(k) + \Gamma u(k) \tag{4.4a}$$

where:

$$\Psi = \begin{bmatrix} \Psi_{11} & \cdots & \Psi_{1j} & \cdots & \Psi_{1m} \\ \vdots & \ddots & & & \vdots \\ \Psi_{i1} & \cdots & \Psi_{ii} & \cdots & \Psi_{im} \\ \vdots & & & \ddots & \vdots \\ \Psi_{m1} & \cdots & \Psi_{mi} & \cdots & \Psi_{mm} \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \Gamma_{11} & \cdots & \Gamma_{1j} & \cdots & \Gamma_{1m} \\ \vdots & \ddots & & & \vdots \\ \Gamma_{i1} & \cdots & \Gamma_{ii} & \cdots & \Gamma_{im} \\ \vdots & & & \ddots & \vdots \\ \Gamma_{m1} & \cdots & \Gamma_{mi} & \cdots & \Gamma_{mm} \end{bmatrix} \tag{4.4b}$$

In the rest of this section, two common optimal control strategies to design a monolithic controller based on the discretized model (4.4a) are compared.

4.2.1 Optimal Control via Linear Quadratic Regulator

In practice, linear quadratic regulator (LQR) is used to obtain the optimal control trajectory of linear state-space systems when only active inequality constraint are present, e.g. the optimal control problem is only subject to the state-space constraints. Consider the discrete-time system (4.4a), the LQR cost function can be defined as:

$$J_{LQR} = \frac{1}{2} \sum_{k=0}^{N_{LQR}-1} \Big(x(k)^T Q_{LQR} x(k) + u(k)^T R_{LQR} u(k) \Big) + \frac{1}{2} x(N_{LQR})^T \bar{P}_{LQR} x(N_{LQR}) \tag{4.5}$$

where: N_{LQR} denotes the final time; Q_{LQR} is a positive-semi-definite weighting matrix for the states; $\bar{P}_{LQR}$ is a positive-semi-definite weighting matrix for the final states; and R_{LQR} is a positive-definite weighting matrix for the manipulated input variables.

Note that, this optimal control problem is convex [1], and the closed-form solution can be derived analytically. With the state-space equation (4.4a) and the cost function (4.5), the Hamiltonian can be defined as:

$$H_{LQR}\big(x(k),u(k),\lambda(k{+}1)\big)=\frac{1}{2}\Big(x(k)^T Q_{LQR}x(k){+}u(k)^T R_{LQR}u(k)\Big){+}\lambda(k{+}1)^T\Big(\Psi x(k){+}\Gamma u(k)\Big) \tag{4.7}$$

where: λ is the vector of Lagrange multipliers. The stationary solution of the LQR problem gives:

$$\frac{\partial H_{LQR}\big(x(k),u(k),\lambda(k+1)\big)}{\partial u(k)}=R_{LQR}u(k)+\Gamma^T\lambda(k+1)=0 \tag{4.8}$$

Based on (4.8), the control action is given by:

$$u(k)=-R_{LQR}^{-1}\Gamma^T\lambda(k+1) \tag{4.9}$$

Therefore, the closed-loop system can be derived as:

$$x(k+1)=\Psi x(k)-\Gamma R_{LQR}^{-1}\Gamma^T\lambda(k+1) \tag{4.10}$$

In addition, the co-state relation [45] gives the current value of $\lambda(k)$ as a function of

[1] A convex optimization problem is of the form [12]:

$$\min_x\ f_0(x) \tag{4.6a}$$
$$\text{s.t. } f_i(x)\le 0 \tag{4.6b}$$
$$ax_i=b_i \tag{4.6c}$$

where: $f_{i=\{0,\cdots,m\}}$ are convex functions, i.e. for $\forall x_1,x_2\in\mathbf{dom}(f_i),\forall\theta\in(0,1):\theta f_i(x_1)+(1-\theta)f_i(x_2)\ge f_i(\theta x_1+(1-\theta)x_2)$.

$\lambda(k+1)$:

$$\lambda(k) = \frac{\partial H_{LQR}\big(x(k), u(k), \lambda(k+1)\big)}{\partial x(k)} = Q_{LQR}x(k) + \Psi^T\lambda(k+1) \tag{4.11}$$

In this formulation, it is assumed that there exists a vector P_{LQR} such that $\lambda(k) = P_{LQR}(k)x(k)$. This can be used to rewrite equations (4.10) and (4.11) as:

$$x(k+1) = \Psi x(k) - \Gamma R_{LQR}^{-1}\Gamma^T P_{LQR}(k+1)x(k+1) \tag{4.12a}$$

$$P_{LQR}(k)x(k) = Q_{LQR}x(k) + \Psi^T P_{LQR}(k+1)x(k+1) \tag{4.12b}$$

Solving (4.12a) for $x(k+1)$, and replacing $x(k+1)$ in (4.12b), the following is obtained for P_{LQR}:

$$P_{LQR}(k)x(k) = Q_{LQR}x(k) + \Psi^T P_{LQR}(k+1)\Big(I + \Gamma R_{LQR}^{-1}\Gamma^T P_{LQR}(k+1)\Big)^{-1}\Psi x(k) \tag{4.13}$$

If relation (4.13) holds for all values of $x(k)$, the discrete-time Ricatti equation relates two consecutive values of P_{LQR} as:

$$P_{LQR}(k) = Q_{LQR} + \Psi^T P_{LQR}(k+1)\Big(I + \Gamma R_{LQR}^{-1}\Gamma^T P_{LQR}(k+1)\Big)^{-1}\Psi \tag{4.14}$$

or in a more compact form as:

$$\begin{aligned} P_{LQR}(k) = {} & Q_{LQR} + \Psi^T P_{LQR}(k+1)\Psi - \\ & \Psi^T P_{LQR}(k+1)\Gamma[R_{LQR} + \Gamma^T P_{LQR}(k+1)\Gamma]^{-1}\Gamma^T P_{LQR}(k+1)\Psi \end{aligned} \tag{4.15}$$

Then, the quadratic Ricatti equation (4.15) is solved for P_{LQR}, backwards in time from

sampling time N_{LQR} [45]. In addition, the boundary conditions of the LQR problem at final time N_{LQR} are:

$$\lambda(N_{LQR}) = \bar{P}_{LQR}x(N_{LQR}) \tag{4.16a}$$

$$P_{LQR}(N_{LQR}) = \bar{P}_{LQR} \tag{4.16b}$$

Control action can be written as:

$$\begin{aligned} u(k) = & -R_{LQR}^{-1}\Gamma^T P_{LQR}(k+1)x(k+1) \\ = & -R_{LQR}^{-1}\Gamma^T P_{LQR}(k+1)\Big(\Psi x(k) + \Gamma u(k)\Big) \end{aligned} \tag{4.17}$$

Solving for u(k), the control action can be written in a state feedback format:

$$\begin{aligned} u(k) = & -\Big(I + R_{LQR}^{-1}\Gamma^T P_{LQR}(k+1)\Gamma\Big)^{-1} R_{LQR}^{-1}\Gamma^T P_{LQR}(k+1)\Psi x(k) \\ = & -\Big(R_{LQR} + \Gamma^T P_{LQR}(k+1)\Gamma\Big)^{-1}\Gamma^T P_{LQR}(k+1)\Psi x(k) \\ = & -K_{LQR}(k)x(k) \end{aligned} \tag{4.18}$$

where $K_{LQR}(k)$ is the LQR gain at time k.

In order to show the discrete-time LQR is stable, consider the cadidate Lyapunov function $V_{LQR}(x) = x(k)^T P(k)x(k) > 0$. Then, the difference form is given by:

$$\Delta V_{LQR}(x) = x(k+1)^T P(k+1)x(k+1) - x(k)^T P(k)x(k) \tag{4.19a}$$

Substituting the relation obtained for $K_{LQR}(k)$ from (4.18) into the Ricatti equation

(4.15), the following holds true [28]:

$$P(k) = \Psi^T P(k+1)\Psi - \Psi^T P(k+1)\Gamma K_{LQR}(k) + Q_{LQR} \tag{4.19b}$$

$$P(k) = [\Psi - \Gamma K_{LQR}]^T P(k+1)[\Psi - \Gamma K_{LQR}] + K_{LQR}^T R_{LQR} K_{LQR} + Q_{LQR} \tag{4.19c}$$

Furthermore, substituting $x(k+1) = \Psi x(k) - \Gamma K_{LQR}(k)x(k)$ and (4.19b) into (4.19a) gives [28]:

$$\Delta V_{LQR}(x) = -x(k)^T[K_{LQR}(k)^T R_{LQR} K_{LQR}(k) + Q_{LQR}]x(k) \tag{4.19d}$$

Since R_{LQR} is positive definite and Q_{LQR} is positive semi-definite $\Delta V_{LQR}(x) \leq 0$; thus, $V_{LQR}(x)$ is the Lyapunov function of the LQR problem and the closed-loop system is stable.

4.2.2 Optimal Control via Model Predictive Control

Unlike the LQR methodology, MPC can be used to obtain the optimal control trajectory of linear state-space systems subject to active and inactive inequality constraint. The finite-time centralized MPC formulation, at time instant t_k, can be defined as:

$$\min_{X,U} J_C = \frac{1}{2}\Big((X(k) - X_{set})^T Q (X(k) - X_{set}) + U(k)^T R U(k)\Big) \tag{4.20a}$$

$$\text{s.t. } \hat{x}(k+l+1|k) = \Psi\hat{x}(k+l|k) + \Gamma\hat{u}(k+l|k), \tag{4.20b}$$

$$(\hat{x}_i, \hat{u}_i) \in \mathcal{C}_i \tag{4.20c}$$

for $l = 0, \cdots, N-1$ with N being the prediction horizon [2], and $\hat{x}$ and $\hat{u}$ are states and manipulated input variables inside the controller, respectively. Additionally: $X(k) = [\hat{x}(k+1|k)^T, \cdots, \hat{x}(k+N|k)^T]^T$ is the vector of the predicted state trajectory; $U(k) = [\hat{u}(k|k)^T, \cdots, \hat{u}(k+N-1|k)^T]^T$ is the vector of the calculated manipulated variable moves; Q is a positive-definite block-diagonal weighting matrix for the states (i.e., $Q = diag\{Q_{ii}\}$); and R is a positive-definite block-diagonal weighting matrix for the manipulated variables of the overall system (i.e., $R = diag\{R_{ii}\}$). The states of the m subsystems, x_i $(i = 1, \ldots, m)$, are assumed to be sampled synchronously at time instants $t_k = k\Delta t$ (with $k = 0, 1, \ldots$). Note that in the remainder of this chapter, k is used to denote t_k in the discrete-time model. Then, in the current sampled-data system, control actions are calculated using the discrete-time model (4.4a) and these actions are applied to the continuous plant (4.1).

Throughout this chapter, MPC is used to provide a unified approach to obtain the optimal trajectory benchmark for the CDMPC scheme. This method uses conventional optimization approaches, such as interior-point method, to solve large-scale optimal control problems; so that, it does not require to solve a Ricatti equation backward in time. In addition, this approach is capable of solving optimal control problems subject to active/inactive inequality constraints efficiently. In this work, closed-loop stability of the MPC scheme is ensured via an adaptive horizon scheme, described in Section 4.5. This can also be extended to the CDMPC problem without reformulation of the plant-wide problem, i.e. it does not require addition of terminal costs or terminal constraints to the original MPC problem (4.20).

[2] In this chapter, for the sake of simplicity, it is assumed the control horizon and the prediction horizon of the system are the same. Although it might lead to a higher computation load, longer control horizons can improve the performance and compensate for package dropouts in networked control systems [80]

4.2.3 Distributed MPC Formulation

Local MPCs are formulated based on local discretized model of the plant and the calculated control actions are applied to the continuous subsystems. Specifically, for subsystem i, $i = 1, \ldots, m$, the prediction model used in the formulation of the subsystem MPC at time instant k takes the following form:

$$\hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k) + \sum_{j\neq i}(1-\beta)\Psi_{ij}\hat{x}_j(k|k) + \hat{v}_i(k+l|k) \tag{4.21a}$$

$$\hat{x}_i(k|k) = x_i(k) \tag{4.21b}$$

with:

$$\beta = \begin{cases} 0 & l = 0 \\ 1 & l = 1, \cdots, N-1 \end{cases} \tag{4.21c}$$

for $l = 0, \cdots, N-1$. In addition, Ψ_{ii} and Γ_{ii} are matrices corresponding to sub-system i, and $\hat{v}_i$ is defined as the interacting or linking variable that contains unknown interaction information between different subsystems. Note that $\hat{v}_i$ characterizes the interaction of subsystem i with other subsystems. To proceed, define e_i as follows:

$$e_i(k+l|k) \triangleq \hat{v}_i(k+l|k) - \sum_{j\neq i}^{m} \big(\beta\Psi_{ij}\hat{x}_j(k+l|k) + \Gamma_{ij}\hat{u}_j(k+l|k)\big) \tag{4.22}$$

The vector $e_i(k+l|k)$ denotes the difference between the interaction of subsystem i with the other subsystems as captured by the plant-wide model and the interaction characterized by $\hat{v}_i$. A specific objective of the coordinator is to find a price for subsystem i such that the interaction term $\hat{v}_i$ determined by the price renders $e_i(k+l|k) = 0$. This

will ensure that the CDMPC approaches the performance of the corresponding centralized MPC. The overall interaction error over the prediction horizon can be described as follows:

$$E(k|k) \triangleq \begin{bmatrix} E_1(k|k) \\ \vdots \\ E_m(k|k) \end{bmatrix} \tag{4.23a}$$

where:

$$E_i(k|k) = \begin{bmatrix} e_i(k|k) \\ e_i(k+1|k) \\ \vdots \\ e_i(k+N-1|k) \end{bmatrix} \tag{4.23b}$$

Based on (4.22), $E(k|k)$ can be written as a separable additive form:

$$E(k|k) = \sum_{i=1}^{m} \Theta_i(k) \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix} \tag{4.24}$$

where: $X_i(k) = [\hat{x}_i(k+1|k)^T, \cdots, \hat{x}_i(k+N|k)^T]^T$ is the vector of the predicted state trajectory; $U_i(k) = [\hat{u}_i(k|k)^T, \cdots, \hat{u}_i(k+N-1|k)^T]^T$ is the vector of the calculated manipulated variable moves; and $V_i(k) = [\hat{v}_i(k|k)^T, \cdots, \hat{v}_i(k+N-1|k)^T]^T$ is the vector of predicted linking variables for subsystem i. In (4.24), $\Theta_i(k)$ is the coefficient matrix for the linking constraints that is defined as:

$$\Theta_i(k) = \begin{bmatrix} \theta_{1,i}^T, & \cdots, & \theta_{i,i}^T, & \cdots, & \theta_{m,i}^T \end{bmatrix}^T \tag{4.25a}$$

where:

$$\theta_{j,i} = \begin{cases} \left[0_{Nn_{x_i} \times Nn_{x_i}}, 0_{Nn_{x_i} \times Nn_{u_i}}, -I\right] & \text{for } j = i \\ \left[\theta_{\Psi_{j,i}}, \theta_{\Gamma_{j,i}}, 0_{Nn_{x_j} \times Nn_{x_i}}\right] & \text{for } j \neq i \end{cases} \tag{4.25b}$$

with 0 denoting zero matrices of appropriate dimensions, I being an identity matrix of size $Nn_{x_i} \times Nn_{x_i}$, as well as $\theta_{\Psi_{j,i}}$ and $\theta_{\Gamma_{j,i}}$ being $Nn_{x_j} \times Nn_{x_i}$ and $Nn_{x_j} \times Nn_{u_i}$ matrices, respectively:

$$\theta_{\Psi_{j,i}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \Psi_{ji} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & \Psi_{ji} & 0 \end{bmatrix}, \tag{4.25c}$$

$$\theta_{\Gamma_{j,i}} = \begin{bmatrix} \Gamma_{ji} & 0 & \cdots & 0 \\ 0 & \Gamma_{ji} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Gamma_{ji} \end{bmatrix} \tag{4.25d}$$

Matrix Θ_i can be written in terms of an augmented matrix of coefficients for X_i, U_i, and V_i as follows:

$$\Theta_i = \left[\begin{array}{c|c|c} \theta_{\Psi_{1,i}} & \theta_{\Gamma_{1,i}} & 0 \\ \vdots & \vdots & \vdots \\ 0_{Nn_{x_i} \times Nn_{x_i}} & 0_{Nn_{x_i} \times Nn_{u_i}} & -I_{Nn_{x_i} \times Nn_{x_i}} \\ \vdots & \vdots & \vdots \\ \theta_{\Psi_{m,i}} & G_{\Gamma_{m,i}} & 0 \end{array}\right] \tag{4.26a}$$

or:

$$\Theta_i = [\Theta_{X_i}, \Theta_{U_i}, \Theta_{V_i}] \tag{4.26b}$$

Therefore, the centralized MPC problem (4.20) can be reformulated as the following overall plant-wide MPC formulation:

$$\min_{X,U,V} J_P = \sum_{i=1}^{m} J_{P_i} \tag{4.27a}$$

$$\text{s.t. } \hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k) + \sum_{j\neq i}(1-\beta)\Psi_{ij}\hat{x}_j(k|k) + \hat{v}_i(k+l|k) \tag{4.27b}$$

$$g_i(\hat{x}_i(k+l|k), \hat{u}_i(k+l|k)) \leq 0 \tag{4.27c}$$

$$\sum_{i=1}^{m} \Theta_i \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix} = 0 \tag{4.27d}$$

where:

$$J_{P_i} = \frac{1}{2}\Big((X_i(k) - X_{i,set})^T Q_{ii}(X_i(k) - X_{i,set}) + U_i(k)^T R_{ii} U_i(k)\Big) \tag{4.27e}$$

for $l = 0, \ldots, N-1$. Optimization problem (4.27) provides the basics for the formulation of the subsystem MPCs and the coordinator. Note that the cost function and constraints are separable with respect to the subsystems.

Remark 27 *The decentralized scheme in general ignores the interactions between subsystems, and the overall decentralized MPC problem is defined as:*

$$\min_{X,U,V} J_{DC} = \sum_{i=1}^{m} J_{DC_i} \tag{4.28a}$$

$$s.t.\ \hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k) \tag{4.28b}$$

$$g_i(\hat{x}_i(k+l|k), \hat{u}_i(k+l|k)) \leq 0 \tag{4.28c}$$

where:

$$J_{DC_i} = \frac{1}{2}\Big((X_i(k) - X_{i,set})^T Q_{ii}(X_i(k) - X_{i,set}) + U_i(k)^T R_{ii} U_i(k)\Big) \tag{4.28d}$$

for $l = 0, \ldots, N-1$*, with* J_{DC} *being the overall objective function of the decentralized MPC problem.*

4.3 Coordination Algorithm I: Equality Constraints

In this section, an analytic CDMPC algorithm is presented for linear systems. A schematic of the proposed CDMPC is shown in Fig. 4.1. In the proposed formulation, a local MPC is formulated for each subsystem and a coordinator coordinates the actions of the subsystem MPCs. The coordinator calculates the optimal price vector analytically for the subsystem MPCs to coordinate their actions for improved performance. Each subsystem MPC calculates control inputs minimizing a local cost function based on subsystem state measurements and the price received from the coordinator.

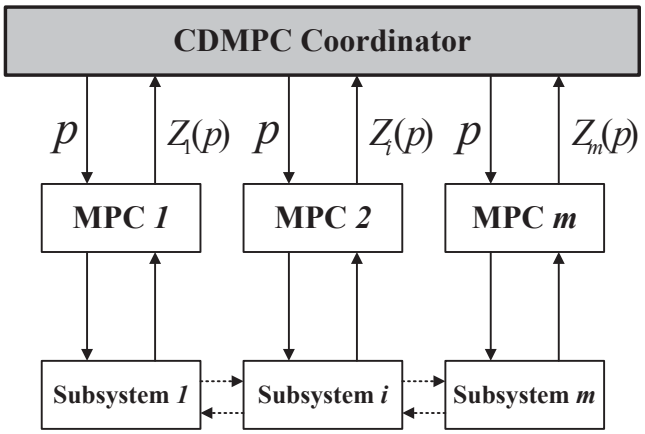


Figure 4.1: Architecture and information flow of the proposed CDMPC.

The idea is to relax constraint (4.27d), which characterizes the interactions, via a price vector so that it is also separable in terms of subsystems. Specifically, the equivalent

overall problem is formulated as follows:

$$\min_{X,U,V} J_D = \sum_{i=1}^{m} J_{D_i} \tag{4.29a}$$

$$\text{s.t. } \hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k) + \sum_{j\neq i}(1-\beta)\Psi_{ij}\hat{x}_j(k|k) + \hat{v}_i(k+l|k) \tag{4.29b}$$

where:

$$J_{D_i} = \frac{1}{2}\Big((X_i(k) - X_{i,set})^T Q_{ii}(X_i(k) - X_{i,set}) + U_i(k)^T R_{ii} U_i(k)\Big) + p^T \Theta_i \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix} \tag{4.29c}$$

for $l = 0, \ldots, N-1$. In the separable optimization problem (4.29), p is a price vector calculated by the coordinator to provide the plant-wide solution for the distributed system. In (4.29), the price vector p can be considered to be the Lagrange multiplier associated with interaction equality constraints. In order to find a plant-wide solution to the distributed system, the bilevel optimization problem (4.30) is defined as:

$$\min_{p} \quad -J_D(p, \bar{Z}^*) \tag{4.30a}$$

$$\bar{Z}^* = \arg\min\{J_D = \sum_{i=1}^{m} \bar{J}_{D_i}\} \tag{4.30b}$$

$$\text{s.t. } A_i \bar{Z}_i(k) = b_i \tag{4.30c}$$

where:

$$\bar{Z}_i = \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix}, \tag{4.30d}$$

$$A_i = [A_{X_i}, A_{U_i}, A_{V_i}], \tag{4.30e}$$

$$A_{X_i} = \begin{bmatrix} I_{n_{x_i}} & 0 & \cdots & 0 \\ -\Psi_{ii} & I_{n_{x_i}} & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & -\Psi_{ii} & I_{n_{x_i}} \end{bmatrix}, \tag{4.30f}$$

$$A_{U_i} = \begin{bmatrix} -\Gamma_{ii} & 0 & \cdots & 0 \\ 0 & -\Gamma_{ii} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\Gamma_{ii} \end{bmatrix}, \tag{4.30g}$$

$$A_{V_i} = I_{Nn_{x_i} \times Nn_{x_i}} \tag{4.30h}$$

$$b_i = \begin{bmatrix} \sum_{j \neq i} (1-\beta)\Psi_{ij}\hat{x}_j(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.30i}$$

In the bi-level optimization problem (4.30), the lower-level problem represents the distributed network of MPC controllers and the upper-level unconstrained optimization problem is handled by the coordinator. The Lagrange function of the lower-level problem

can be written as follows:

$$\begin{aligned}\mathcal{L} =&\frac{1}{2}\begin{bmatrix}X\\U\end{bmatrix}^T\begin{bmatrix}Q & 0\\0 & R\end{bmatrix}\begin{bmatrix}X\\U\end{bmatrix} - [X_{sp}^T Q, 0]\begin{bmatrix}X\\U\end{bmatrix} + p^T[\Theta_X, \Theta_U]\begin{bmatrix}X\\U\end{bmatrix}\\ &+ p^T\Theta_V V + \nu^T[A_X, A_U]\begin{bmatrix}X\\U\end{bmatrix} + \nu^T A_V V - \nu^T b\end{aligned} \tag{4.31}$$

The KKT conditions of the lower problem can be written as:

$$\begin{cases} QX - QX_{sp} + \Theta_X^T p + A_X^T \nu &= 0\\ RU + \Theta_U^T p + A_U^T \nu &= 0\\ -p + \nu &= 0\\ A_X X + A_U U + V - b &= 0 \end{cases} \tag{4.32}$$

which includes the following relations for ν and V:

$$\nu = p \tag{4.33a}$$

$$V = b - (A_X X + A_U U) \tag{4.33b}$$

Then, the Lagrange function of the lower-level problem can be stated as:

$$\mathcal{L}(\bar{Z}) = \mathcal{L}(Z) = \mathcal{L}_Z + p^T \mathcal{L}_p \tag{4.34a}$$

where:

$$\mathcal{L}_Z = \frac{1}{2} Z^T \Upsilon Z - [X_{sp}^T Q, 0] Z \tag{4.34b}$$

$$\mathcal{L}_p = [\Theta_X + A_X, \Theta_U + A_U] Z - b \tag{4.34c}$$

$$\Upsilon = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \tag{4.34d}$$

$$Z = \begin{bmatrix} X \\ U \end{bmatrix} \tag{4.34e}$$

The KKT conditions of the lower problem can be written as:

$$\nabla_Z \mathcal{L} = \frac{d\mathcal{L}_Z}{dZ} + p^T \frac{d\mathcal{L}_p}{dZ} = 0 \tag{4.35a}$$

$$[A_X, A_U]Z + V - b = 0 \tag{4.35b}$$

Then, the bi-level problem (4.30) can be written as a single layer optimization problem as:

$$\min_{p,Z} \; - J_D(p, Z) \tag{4.36a}$$

$$\nabla_Z \mathcal{L} = \frac{d\mathcal{L}_Z}{dZ} + p^T \frac{d\mathcal{L}_p}{dZ} = 0 \tag{4.36b}$$

$$[A_X, A_U]Z + V - b = 0 \tag{4.36c}$$

Solving for Z and V, the closed-loop solution of the lower problem w.r.t. the current value of the price vector, provided by the coordinator is:

$$\begin{aligned} Z^*(k;p) =& \Upsilon^{-1}\Big(\begin{bmatrix} QX_{sp} \\ 0 \end{bmatrix} - [\Theta_X + A_X, \Theta_U + A_U]^T p \Big) \\ =& \begin{bmatrix} X_{sp} \\ 0 \end{bmatrix} - \begin{bmatrix} -Q^{-1}\big(\Theta_X + A_X\big)^T \\ -R^{-1}\big(\Theta_U + A_U\big)^T \end{bmatrix} p \end{aligned} \tag{4.37a}$$

$$V^*(k;p) = b - [A_X, A_U]Z^*(k;p) \tag{4.37b}$$

The following theorem, expresses the nature of the closed form solution (4.37a) as a predictor-corrector term.

Theorem 28 *The closed-loop solution $Z^*(p)$ can be written in predictor-corrector form around the non-coordinated trajectory, namely "$p = 0$":*

$$Z^*(k;p) = Z_{predictor} + Z_{corrector}p = Z(k;0) + \frac{dZ}{dp^T}|_{p=0}p \tag{4.38}$$

Proof. The non-coordinated trajectory is derived directly from KKT conditions (4.35) by taking "$p = 0$":

$$\nabla_Z \mathcal{L}(0) = \frac{\mathcal{L}_Z}{dZ} = 0 \tag{4.39a}$$

Then it can be shown that:

$$\Upsilon Z(k;0) - [X_{sp}^T Q, 0]^T = 0 \tag{4.39b}$$

which implies:

$$Z(0) = \Upsilon^{-1}\Big(\begin{bmatrix} QX_{sp} \\ 0 \end{bmatrix} \Big) \tag{4.39c}$$

$$= \begin{bmatrix} X_{sp} \\ 0 \end{bmatrix} \tag{4.39d}$$

In order to find $\frac{dZ}{dp^T}$, $\nabla_{Z,p}\mathcal{L}$ is calculated:

$$\begin{aligned}\nabla_{Z,p}\mathcal{L} =& \frac{d}{dp}\Big(\frac{d\mathcal{L}_Z}{dZ} + \frac{d\mathcal{L}_p^T}{dZ}p\Big) \\ =& \frac{dZ^T}{dp}\Big(\frac{d^2\mathcal{L}_Z}{dZ^T dZ} + p^T\frac{d^2\mathcal{L}_p}{dZ^T dZ}\Big) + \frac{d\mathcal{L}_p}{dZ^T} \\ =& 0 \end{aligned} \tag{4.40a}$$

Thus, $\frac{dZ}{dp^T}$ can be written as:

$$\Big(\frac{dZ}{dp^T}\Big)^T = -\frac{d\mathcal{L}_p}{dZ^T}\Big(\frac{d^2\mathcal{L}_Z}{dZ^T dZ} + p^T\frac{d^2\mathcal{L}_p}{dZ^T dZ}\Big)^{-1} \tag{4.40b}$$

and:

$$\frac{dZ}{dp^T}|_{p=0} = -\begin{bmatrix} -Q^{-1}\big(\Theta_X + A_X\big)^T \\ -R^{-1}\big(\Theta_U + A_U\big)^T \end{bmatrix} \tag{4.40c}$$

■

Parametrizing Z^* as a function of the price vector p and V^* as a function of Z^*, using (4.37a) and (4.37b), the single layer optimization (4.36) can be converted into an unconstrained optimization problem with respect to the price vector as:

$$\min_p \ -J_D(p, Z^*(k;p)) \tag{4.41}$$

Similar to (4.38), the Lagrangian of the overall problem can be stated as a quadratic function of the price vector around the non-coordinated trajectory.

Theorem 29 *The Lagrange function of the single layer problem (4.41) can be written*

as a quadratic function of "p" around the non-coordinated trajectory, namely "p = 0":

$$\bar{\mathcal{L}}(p, Z^*(k;p)) = \bar{\mathcal{L}}(p) = \frac{1}{2} p^T \nabla^2_{p,p} \bar{\mathcal{L}}(p)|_{p=0} p + \nabla_p \bar{\mathcal{L}}(p)|_{p=0} p + \bar{\mathcal{L}}(0) \tag{4.42}$$

Proof. According to (4.41), the Lagrangian of the unconstrained problem is derived as:

$$\bar{\mathcal{L}}(p, Z^*(k;p)) = -J_D(p, Z^*(k;p)) \tag{4.43a}$$

Based on the convex nature of the lower-level problem, the equality constraint (4.36c) is always satisfied, i.e. $[A_X, A_U]Z^* + V^* - b = 0$. Therefore, penalizing this constraint using ν at $Z^*(p)$ and adding it to the right-hand-side (RHS) of (4.43a) would result in:

$$\bar{\mathcal{L}}(p, Z^*(k;p)) = - J_D(p, Z^*(k;p)) - \nu^T \big([A_X, A_U]Z^* + V^* - b\big) \tag{4.43b}$$

or at $Z^*(p)$:

$$\bar{\mathcal{L}}(p, Z^*(k;p)) = -\mathcal{L}(p, Z^*(k;p)) \tag{4.43c}$$

Substituting (4.37a) into $\bar{\mathcal{L}}(p, Z^*(k;p))$ yields:

$$\bar{\mathcal{L}}(p) = -\frac{1}{2}\Big(\Upsilon^{-1}\big(-[\theta_X,\theta_U]^T p + \begin{bmatrix} QX_{sp} \\ 0 \end{bmatrix}\big)\Big)^T \Upsilon\Big(\Upsilon^{-1}\big(-[\theta_X,\theta_U]^T p + \begin{bmatrix} QX_{sp} \\ 0 \end{bmatrix}\big)\Big)$$
$$+ [X_{sp}Q,0]\Upsilon^{-1}\big(-[\theta_X,\theta_U]^T p + \begin{bmatrix} QX_{sp} \\ 0 \end{bmatrix}\big) + p^T b$$
$$- p^T[\theta_X,\theta_U]\Upsilon^{-1}\big(-[\theta_X,\theta_U]^T p + \begin{bmatrix} QX_{sp} \\ 0 \end{bmatrix}\big) \tag{4.44a}$$
$$= \frac{1}{2}p^T[\theta_X,\theta_U]\Upsilon^{-1}[\theta_X,\theta_U]^T p - \big([X_{sp}^T Q,0]\Upsilon^{-1}[\theta_X,\theta_U]^T - b^T\big)p$$
$$+ \frac{1}{2}[X_{sp}^T Q,0]\Upsilon^{-1}\begin{bmatrix} QX_{sp} \\ 0 \end{bmatrix} \tag{4.44b}$$

where:

$$\theta_X = \Theta_X + A_X \tag{4.44c}$$
$$\theta_U = \Theta_U + A_U \tag{4.44d}$$

but:

$$\nabla^2_{p,p}\bar{\mathcal{L}}(p)|_{p=0} = \frac{d\bar{\mathcal{L}}_p}{dZ^T}\frac{dZ^T}{dp}|_{p=0} = [\theta_X,\theta_U]\Upsilon^{-1}[\theta_X,\theta_U] \tag{4.44e}$$
$$\nabla_p\bar{\mathcal{L}}(p)|_{p=0} = \bar{\mathcal{L}}_p(0) = -[X_{sp}^T Q,0]\Upsilon^{-1}[\theta_X,\theta_U]^T + b^T \tag{4.44f}$$
$$\bar{\mathcal{L}}(0) = \bar{\mathcal{L}}_Z(0) = \frac{1}{2}[X_{sp}^T Q,0]\Upsilon^{-1}\begin{bmatrix} QX_{sp} \\ 0 \end{bmatrix} \tag{4.44g}$$

■

As a result of (4.41) and (4.43c), the corresponding KKT condition would be:

$$\frac{d\bar{\mathcal{L}}(Z^*,p)}{dp} = -\frac{d\mathcal{L}(Z^*,p)}{dp} \tag{4.45a}$$

$$= -\Big(\frac{d\mathcal{L}_Z(Z^*)}{dZ^{*T}} + p^T\frac{d\mathcal{L}_p(Z^*(k;p))}{dZ^{*T}}\Big)\frac{dZ^*}{dp} - \mathcal{L}_p^T(Z^*) \tag{4.45b}$$

but:

$$\frac{d\mathcal{L}_Z(Z^*)}{dZ^{*T}} + p^T\frac{d\mathcal{L}_p(Z^*(p))}{dZ^{*T}} = 0 \tag{4.45c}$$

which according to (4.35), this is equivalent to:

$$\frac{d\bar{\mathcal{L}}(Z^*,p)}{dp} = -\mathcal{L}_p^T(Z^*(k;p)) = 0 \tag{4.45d}$$

To solve the CDMPC problem using conventional methods [66], (4.45d) together with (4.36c) are calculated numerically. Given sensitivity information of the local controllers (4.40b), the coordinator uses an iterative gradient-based procedure such as Newton's method to update the price vector. Thus in order to converge to the optimal plant-wide solution, the coordinator and local controllers communicate, in coordination cycles, until the interaction constraint (4.27d) is satisfied [20, 21, 66, 76, 57].

In the proposed approach, the KKT condition (4.45d) is solved analytically to obtain the optimal price vector, based on the parametrized values of Z^* and V^*, explained in (4.37a)-(4.37b). Using (4.34c) and (4.37a), the price vector can be derived as the solution to (4.45d):

$$[\theta_X,\theta_U]^T p\begin{bmatrix} -Q^{-1}\big(\theta_X\big)^T p + X_{sp} \\ -R^{-1}\big(\theta_U\big)^T p \end{bmatrix} - b = 0 \tag{4.46a}$$

From (4.46a), it can be found that:

$$p = \Big([\theta_X, \theta_U] \Upsilon^{-1} [\theta_X, \theta_U]^T \Big)^{-1} \Big(\big(\theta_X\big) X_{sp} - b \Big) \tag{4.46b}$$

The next two lemmas are required to prove (4.46b) is not singular.

Lemma 30 *[47] A real symmetric matrix M is positive definite if and only if a real non-singular matrix Y exists such that:*

$$M = YY^T \tag{4.47}$$

Lemma 31 *[47] Summation preserves the positive definiteness property of matrices.*

Next, a theorem is provided to show why (4.46b) always exists.

Theorem 32 *The analytic solution to the price vector (4.46b) is not singular.*

Proof. Expand the quadratic coefficient in equation (4.46b) as:

$$[\theta_X, \theta_U] \Upsilon^{-1} [\theta_X, \theta_U]^T = \theta_X Q^{-1} \theta_X^T + \theta_U R^{-1} \theta_U^T \tag{4.48a}$$

Since the weighing matrices Q and R are positive definite, their inverse matrices Q^{-1} and R^{-1} are also positive definite. According to Lemma 30, Q^{-1} and R^{-1} can be written in terms of non-singular matrices Y_Q and Y_R as follows:

$$Q^{-1} = Y_Q Y_Q^T \tag{4.48b}$$

$$R^{-1} = Y_R Y_R^T \tag{4.48c}$$

Hence, the RHS of (4.48a) can be written as:

$$\theta_X Q^{-1}\theta_X^T + \theta_U R^{-1}\theta_U^T = \theta_X Y_Q Y_Q^T \theta_X^T + \theta_U Y_R Y_R^T \theta_U^T \tag{4.48d}$$

$$= (\theta_X Y_Q)(\theta_X Y_Q)^T + (\theta_U Y_R)(\theta_U Y_R)^T \tag{4.48e}$$

Since θ_X and θ_U are also non-singular matrices [57], $(\theta_X Y_Q)(\theta_X Y_Q)^T$ and $(\theta_U Y_R)(\theta_U Y_R)^T$ are positive definite matrices. Therefore, according to Lemma 31, left-hand-side (LHS) of (4.48a) is positive definite, and (4.46b) is not singular. ■

Finally, the closed-loop solution to the CDMPC problem can be derived as a result of (4.37a), (4.37b) and (4.46b):

$$Z^*(k;p) = \begin{bmatrix} X_{sp} \\ 0 \end{bmatrix} + \begin{bmatrix} -Q^{-1}\big(\Theta_X + A_X\big)^T \\ -R^{-1}\big(\Theta_U + A_U\big)^T \end{bmatrix} \Big([\theta_X, \theta_U]\Upsilon^{-1}[\theta_X, \theta_U]^T\Big)^{-1}\Big(\big(\theta_X\big)X_{sp} - b\Big) \tag{4.49a}$$

and

$$V^*(k;p) = b - [A_X, A_U]\left(\begin{bmatrix} X_{sp} \\ 0 \end{bmatrix} + \begin{bmatrix} -Q^{-1}\big(\Theta_X + A_X\big)^T \\ -R^{-1}\big(\Theta_U + A_U\big)^T \end{bmatrix} \times \Big([\theta_X, \theta_U]\Upsilon^{-1}[\theta_X, \theta_U]^T\Big)^{-1}\Big(\big(\theta_X\big)X_{sp} - b\Big)\right) \tag{4.49b}$$

The analytic CDMPC method is summarized in Algorithm 14:

Algorithm 14: The analytic CDMPC algorithm

Initialization: Number of sub-systems m, initial state $x(k)$, positive definite matrix Q, positive definite matrix R, initial prediction horizon N;

Local MPCs: Given $x(k)$ and N, formulate local KKT conditions from (4.30);

Local MPCs: Calculate θ_{X_i} and θ_{U_i};

Local MPCs: Send θ_{X_i}, θ_{U_i}, Υ_i, X_{sp_i}, b_i to Coordinator;

Coordinator: Calculate the optimal price vector from (4.46b);

Coordinator: Send the optimal price vector and N to the local controllers;

Local MPCs: Calculate $Z_i^*(k)$ and $V_i^*(k)$ using (4.37);

Local MPCs: Apply the RHC action to the plant;

Output: Z^*, V^*, and p;

4.4 Coordination Algorithm II: Inequality Constraints

In this section, a feasible descent direction approach is presented to coordinate distributed MPCs subject to equality and convex set of inequality. In the proposed formulation, subsystem MPCs and the coordinator communicate and exchange information iteratively every sampling time to find the optimal price and state trajectories. Each subsystem MPC calculates control inputs minimizing a local cost function based on subsystem state measurements and the price received from the coordinator.

Feasible direction methods improve the solution at each iteration, such that if the initial stage is feasible all subsequent iterations do not leave the feasibility region. This can lead to an acceptable solution in early steps of the optimization [56]. Such methods often require first-order derivate information of the optimization problem, which defines the main content of communication between the coordinator and local controllers. The theory behind feasible direction methods was originally developed by Zoutendijk in 1960, then several variations were proposed (i.e., [99, 10, 55, 56]) to improve the algorithm and

assure convergence.

Descent like feasible direction methods were applied to bi-level problems and some important examples are [87] and [70]. In this work, a modified version of [87] is implemented for the general case of CDMPC problem. This hybrid solution algorithm is initiated with a relaxed version of KKT conditions of the lower-level problem and subsequently solves a single level optimization. Once the initial active set of constraints is identified, a feasible direction for the price vector is iteratively calculated through coordination cycles between local MPCs and the coordinator. Once the optimal price is found, it is sent to local MPCs and a plant-wide optimal receding horizon action is applied to sub-systems.

4.4.1 Algorithm Description

In this section, the network of MPC controllers subject to local inequality constraints g_i, defined in (4.3), is analyzed. The overall formulation of problem takes the following form:

$$\min_{X,U,V} J_D = \sum_{i=1}^{m} J_{D_i} \tag{4.50a}$$

$$\text{s.t. } \hat{x}_i(k+l+1|k) = \Psi_{ii}\hat{x}_i(k+l|k) + \Gamma_{ii}\hat{u}_i(k+l|k) + \sum_{j\neq i}(1-\beta)\Psi_{ij}\hat{x}_j(k|k) + \hat{v}_i(k+l|k) \tag{4.50b}$$

$$g_i(\hat{x}_i(k+l|k), \hat{u}_i(k+l|k)) \leq 0 \tag{4.50c}$$

where:

$$J_{D_i} = \frac{1}{2}\Big((X_i(k) - X_{i,set})^T Q_{ii}(X_i(k) - X_{i,set}) + U_i(k)^T R_{ii} U_i(k)\Big) + p^T \Theta_i \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix} \tag{4.50d}$$

for $l = 0, \ldots, N-1$. In order to find a plant-wide solution to the distributed system (4.50), the following bi-level optimization problem is defined:

$$\min_p \; F(p, \bar{Z}^*) \tag{4.51a}$$

$$\bar{Z}^* = \arg\min f(p, \bar{Z}) \tag{4.51b}$$

$$\text{s.t. } H_{i,j}(p, \bar{Z}_i) = 0, \quad j \in \mathcal{J}_{H_i} \tag{4.51c}$$

$$G_{i,j}(p, \bar{Z}_i) \leq 0, \quad j \in \mathcal{J}_{G_i} \tag{4.51d}$$

where:

$$\bar{Z}_i = \begin{bmatrix} X_i(k) \\ U_i(k) \\ V_i(k) \end{bmatrix}, \tag{4.51e}$$

$$F(p, \bar{Z}^*) = -J_D(p, \bar{Z}^*), \tag{4.51f}$$

$$f(p, \bar{Z}_i) = J_D(p, \bar{Z}) = \sum_{i=1}^{m} J_{D_i}(p, \bar{Z}_i), \tag{4.51g}$$

$$H_{i,j}(p, \bar{Z}_i) = A_i \bar{Z}_i - b_i, \tag{4.51h}$$

$$A_i = [A_{X_i}, A_{U_i}, A_{V_i}], \tag{4.51i}$$

$$A_{X_i} = \begin{bmatrix} I_{n_{x_i}} & 0 & \cdots & 0 \\ -\Psi_{ii} & I_{n_{x_i}} & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & -\Psi_{ii} & I_{n_{x_i}} \end{bmatrix}, \tag{4.51j}$$

$$A_{U_i} = \begin{bmatrix} -\Gamma_{ii} & 0 & \cdots & 0 \\ 0 & -\Gamma_{ii} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\Gamma_{ii} \end{bmatrix}, \tag{4.51k}$$

$$A_{V_i} = I_{Nn_{x_i} \times Nn_{x_i}}, \tag{4.51l}$$

$$b_i = \begin{bmatrix} \sum_{j \neq i}(1-\beta)\Psi_{ij}\hat{x}_j(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{4.51m}$$

$$G_{i,j}(p, \bar{Z}_i) = \begin{bmatrix} g_i(\hat{x}_i(k|k), \hat{u}_i(k|k)) \\ \vdots \\ g_i(\hat{x}_i(k+N-1|k), \hat{u}_i(k+N-1|k)) \end{bmatrix} \tag{4.51n}$$

and J_{H_i}, J_{G_i} are finite set of indexes corresponding to equality and inequality constraints in subsystem 'i'. In the BLP problem (4.51), the constrained lower-level optimization problem represents the distributed network of MPC controllers and the upper-level unconstrained optimization problem is handled by the coordinator.

Denote $\mathcal{S}_i(p) = \{(\bar{Z}_i) \in \mathbb{R}^{2\times n_{x_i}+n_{u_i}} | H_{i,j}(p, \bar{Z}_i) = 0, G_{i,j}(p, \bar{Z}_i) \leq 0, j \in \mathcal{J}_{H_i} \cup \mathcal{J}_{G_i}\}$ set of feasible solutions, and $\mathcal{J}_{G_{i,act}}(p) = \{j \in \mathcal{J}_{G_i} | G_{i,j}(p, \bar{Z}_i) = 0\}$ set of indexes of active constraints of subsystem 'i', corresponding to the current value of the price vector.

Assumption 33 *The BLP problem (4.51) is well-posed, i.e. for any fixed value of the*

price vector 'p', the lower-level problem is convex [3] *and there exists a unique optimal solution '$\bar{Z}^*(p)$'.*

Assumption 34 *To guarantee that there is at least one solution to the BLP problem (4.51), it is assumed that the feasible solution set of the lower-level problem based on the price vector ($\mathcal{S}_i(p)$) is nonempty and uniformly compact.*

Assumption 35 *For every local optimum of the lower-level problem in (4.51), there exists an unique optimal price vector associated with the distributed MPC network.*

For any fixed value of 'p', the Lagrange function of the lower-level problem can be stated as:

$$\mathcal{L}(p, \bar{Z}, \nu, \lambda) = \sum_{i=1}^{m} \mathcal{L}_i(p, \bar{Z}_i, \nu_i, \lambda_i) \tag{4.52a}$$

where:

$$\mathcal{L}_i(p, \bar{Z}_i, \nu_i, \lambda_i) = f_i(p, \bar{Z}_i) + \sum_{j \in \mathcal{J}_{H_i}} \nu_{i,j} H_{i,j}(p, \bar{Z}_i) + \sum_{j \in \mathcal{J}_{G_{i,act}}} \lambda_{i,j} G_{i,j}(p, \bar{Z}_i) \tag{4.52b}$$

and ν, λ are the Lagrange multiplier associated with equality and inequality constraints, respectively. Additionally, consider the following set of assumptions on the lower-level problem for a fixed value of p to make sure that regularity of constraints holds at $\bar{Z}_i^* \in (\mathcal{S}_i(p))$:

Assumption 36 *The vectors $\nabla_{\bar{Z}_i} H_{i,j}(p, \bar{Z}_i^*)$ and $\nabla_{\bar{Z}_i} G_{i,j}(p, \bar{Z}_i^*)$, for $j \in \mathcal{J}_{H_i} \cup \mathcal{J}_{G_{i,act}}$, are linearly independent.*

[3] For a fixed value of the price vector, the lower-level problem is convex i.e. $f(p, \bar{Z}_i)$ and $G_{i,j}(p, \bar{Z}_i)$ are convex functions and $H_{i,j}(p, \bar{Z}_i)$ is affine [12].

Assumption 37 *The strict complementarity slackness (SCS) property holds at* $\bar{Z}_i^* \in (\mathcal{S}_i(p))$ *w.r.t.* (ν_i, λ_i)[4].

Assumption 38 *Second order sufficient condition (SOSC) [11] holds at* $\bar{Z}_i^* \in (\mathcal{S}_i(p))$ [5].

Lemma 39 *Assumptions 36 and 37 together with Assumption 38 ensure that* $\bar{Z}_i$ *is a Lipschitz function of the price vector [9].*

The KKT conditions of the lower-level problem, for a fixed value of 'p', can be written as:

$$\nabla_{\bar{Z}_i} f_i(p, \bar{Z}_i) + \left(\nabla_{\bar{Z}_i} H_i(p, \bar{Z}_i)\right)^T \nu_i + \left(\nabla_{\bar{Z}_i} G_{i,act}(p, \bar{Z}_i)\right)^T \lambda_{i,act} = 0, \tag{4.54a}$$

$$H_{i,j\in\mathcal{J}_{H_i}}(p, \bar{Z}_i) = 0, \tag{4.54b}$$

$$G_{i,j\in\mathcal{J}_{G_i}}(p, \bar{Z}_i) \leq 0, \tag{4.54c}$$

$$\lambda_{i,j\in\mathcal{J}_{G_i}} G_{i,j\in J_{G_i}}(p, \bar{Z}_i) = 0, \tag{4.54d}$$

$$\lambda_{i,j\in\mathcal{J}_{G_i}} \geq 0 \tag{4.54e}$$

where:

$$G_{i,act}(p, \bar{Z}_i) = \left[G_{i,j}(p, \bar{Z}_i)\right]_{j\in\mathcal{J}_{G_{i,act}}}, \quad \lambda_{i,act} = \left[\lambda_{i,j}\right]_{j\in\mathcal{J}_{G_{i,act}}} \tag{4.54f}$$

Based on Assumptions 33, 34, 35, 36, and 37, the corresponding Lagrange multipliers ν_i and λ_i are bounded and can be uniquely determined.

[4]SCS requires that for all $\lambda_i > 0$: $G_i(p, \bar{Z}_i^*)$=0.

[5]SOSC holds if for all non-zero feasible directions d that $w_i^T \nabla_Z^2 L_i(p, \bar{Z}_i) w_i \geq 0$, it results:

$$\begin{cases} w_i^T \nabla_{\bar{Z}_i} G_{i,j}(p, \bar{Z}_i^*) = 0, & \text{for } j \in J_{G_{i,act}} \\ w_i^T \nabla_{\bar{Z}_i} H_{i,j}(p, \bar{Z}_i^*) = 0 \end{cases} \tag{4.53}$$

.

In order to determine the initial set of active constraints $\mathcal{J}_{G_{i,act}}(p)$ for the BLP problem (4.51), consider the following optimization:

$$\min_{p,\bar{Z}_i,\nu_i,\lambda_i} \; F(p,\bar{Z}) \tag{4.55a}$$

$$\text{s.t. } \nabla_{\bar{Z}_i} f_i(p,\bar{Z}_i) + \left(\nabla_{\bar{Z}_i} H_i(p,\bar{Z}_i)\right)^T \nu_i + \left(\nabla_{\bar{Z}_i} G_{i,act}(p,\bar{Z}_i)\right)^T \lambda_{i,act} = \; 0, \tag{4.55b}$$

$$H_{i,j\in\mathcal{J}_{H_i}}(p,\bar{Z}_i) = \; 0, \tag{4.55c}$$

$$G_{i,j\in\mathcal{J}_{G_i}}(p,\bar{Z}_i) \leq \; 0, \tag{4.55d}$$

$$\lambda_{i,j\in\mathcal{J}_{G_i}} G_{i,j\in\mathcal{J}_{G_i}}(p,\bar{Z}_i) = \; 0, \tag{4.55e}$$

$$\lambda_{i,j\in\mathcal{J}_{G_i}} \geq \; 0 \tag{4.55f}$$

where the lower-level problem is replaced by the corresponding KKT conditions, defined in (4.54). The resulting problem can also be regarded as a mathematical program with equilibrium constraints (MPEC). Note that, this reformulation of (4.51) is neither differentiable nor regular [18]. Instead, a perturbed problem, for a given vector η, is considered, in which the corresponding complementarity constraints were refined along a central path:

$$\min_{p,\bar{Z}_i,\nu_i,\lambda_i} \; F(p,\bar{Z}) \tag{4.56a}$$

$$\text{s.t. } \nabla_{\bar{Z}_i} f_i(p,\bar{Z}_i) + \left(\nabla_{\bar{Z}_i} H_i(p,\bar{Z}_i)\right)^T \nu_i + \left(\nabla_{\bar{Z}_i} G_{i,act}(p,\bar{Z}_i)\right)^T \lambda_{i,act} = \; 0, \tag{4.56b}$$

$$H_{i,j\in\mathcal{J}_{H_i}}(p,\bar{Z}_i) = \; 0, \tag{4.56c}$$

$$G_{i,j\in\mathcal{J}_{G_i}}(p,\bar{Z}_i) \leq \; 0, \tag{4.56d}$$

$$\lambda_{i,j\in\mathcal{J}_{G_i}} G_{i,j\in\mathcal{J}_{G_i}}(p,\bar{Z}_i) = \; \eta^2, \tag{4.56e}$$

$$\lambda_{i,j\in\mathcal{J}_{G_i}} \geq \; 0 \tag{4.56f}$$

Thus, the central path starts from $\eta > 0$ and approaches to zero to find an approxi-

mate solution to (4.51). In this work, the nonlinear program (4.56) with complementary constraints is smoothed using a CHKS smoothing function [33] [6] Π defined as:

$$\Pi(a, b, \eta) = a + b - \sqrt{(a-b)^2 + 4\eta^2} \tag{4.58}$$

Remark 40 *For a given value of* $\eta \geq 0$, $\Pi(-G_{i,j\in J_{G_i}}(p, \bar{Z}_i), \lambda_{i,j\in\mathcal{J}_{G_i}}, \eta_i) = 0$, *is equivalent to the following condition:*

$$\lambda_{i,j\in\mathcal{J}_{G_i}} G_{i,j\in\mathcal{J}_{G_i}}(p, \bar{Z}_i) = \ \eta_i^2 \tag{4.59}$$

provided $-G_{i,j\in\mathcal{J}_{G_i}}(p, \bar{Z}_i) \geq 0$, *and* $\lambda_{i,j\in\mathcal{J}_{G_i}} \geq 0$ *hold.*

Then, reformulate (4.56) via the smoothing function (4.59) to form the following optimization problem:

$$\min_{p,\bar{Z}_i,\nu_i,\lambda_i} F(p, \bar{Z}) \tag{4.60a}$$

$$\text{s.t. } \nabla_{\bar{Z}_i} f_i(p, \bar{Z}_i) + \left(\nabla_{\bar{Z}_i} H_i(p, \bar{Z}_i)\right)^T \nu_i + \left(\nabla_{\bar{Z}_i} G_{i,act}(p, \bar{Z}_i)\right)^T \lambda_{i,act} = \ 0, \tag{4.60b}$$

$$H_{i,j\in\mathcal{J}_{H_i}}(p, \bar{Z}_i) = \ 0, \tag{4.60c}$$

$$G_{i,j\in\mathcal{J}_{G_i}}(p, \bar{Z}_i) \leq \ 0, \tag{4.60d}$$

$$\Pi(-G_{i,j\in\mathcal{J}_{G_i}}(p, \bar{Z}_i), \lambda_{i,j\in\mathcal{J}_{G_i}}, \eta) = 0, \tag{4.60e}$$

$$\lambda_{i,j\in\mathcal{J}_{G_i}} \geq \ 0 \tag{4.60f}$$

This problem can be solved using efficient NLP methods such as interior-point method [6].

[6] A perturbed version of Fischer-Burmeister function [37] can also be used for this application, which is defined as :

$$\Pi(a, b, \eta) = a + b - \sqrt{a^2 + b^2 + 2\eta} \tag{4.57}$$

The initial set of active constraints $J_{G_{i,act}}(p)$ for the original BLP problem (4.51), at sampling time 'k', is determined via solving NLP problem (4.60) for a decreasing sequence of values for η until it reaches certain tolerance.

Remark 41 *Consider the scenario where the inequality constraints $G_i(p, \bar{Z}_i) \leq 0$ are affine functions of local optimization variables $\bar{Z}_i$, defined as follows:*

$$G_i(p, \bar{Z}_i) = A_i^{ineq} \bar{Z}_i - b_i^{ineq} \leq 0 \tag{4.61a}$$

where:

$$A_i^{ineq} = [A_{X_i}^{ineq}, A_{U_i}^{ineq}, A_{V_i}^{ineq}] \tag{4.61b}$$

Then, the NLP problem (4.60) can be simplified into:

$$\min_{p, \bar{Z}_i, \nu_i, \lambda_i} \; - J_D(p, \bar{Z}) \tag{4.62a}$$

$$s.t. \begin{bmatrix} Q_{ii} & & \\ & R_{ii} & \\ & & 0 \end{bmatrix} \bar{Z}_i - \begin{bmatrix} X_{i,set}^T Q_{ii} \\ 0 \\ 0 \end{bmatrix} + \Theta_i^T p + A_i^T \nu_i + A_{i,act}^{ineq^T} \lambda_{i,act} = \; 0, \tag{4.62b}$$

$$A_i \bar{Z}_i - b_i = \; 0, \tag{4.62c}$$

$$A_i^{ineq} \bar{Z}_i - b_i^{ineq} \leq \; 0, \tag{4.62d}$$

$$\Pi(-A_{i,j \in \mathcal{J}_{G_i}}^{ineq}(p, \bar{Z}_i), \lambda_{i,j \in \mathcal{J}_{G_i}}, \eta) = 0, \tag{4.62e}$$

$$\lambda_{i,j \in \mathcal{J}_{G_i}} \geq \; 0 \tag{4.62f}$$

This stage of the CDMPC algorithm, which identifies an initial guess for $\mathcal{J}_{G_{i,act}}$ in the lower-level problem of the BLP (4.51), is listed in Algorithm 15:

Algorithm 15: Initial guess for $\mathcal{J}_{G_{i,act}}$ of the BLP problem (4.51)

Input: $0 < \eta_{\min} < \eta^0 \leq 1$, $l = 0$, $\beta \in (0,1)$, $Z_i^0(k)$, $p^0 = 0$;
while $(\eta^l \geq \eta_{\min})$ **do**
  Coordinator - Local MPCs: Solve the NLP problem (4.60), i.e. using the interior-point method [6];
  $\eta^{l+1} = \beta\eta^l$;
  $l = l + 1$;
  Coordinator - Local MPCs: update $\bar{Z}_i^l(k)$ and p^l;
for $j \in \mathcal{J}_{G_i}$ **do**
  if $G_{i,j}(p^l, \bar{Z}_i) = 0$ **then**
    Coordinator: $\mathcal{J}^0_{G_{i,act}} = \{j \cup J^0_{G_{i,act}}\}$;
Output: $\mathcal{J}^0_{G_{i,act}}$, $\bar{Z}_i^l(k)$, and p^l;

Once the initial stage is finished, an approximate local solution to the original BLP (4.51) has been determined. In the next stage, this solution is improved using a parametric optimization approach, until the change in the overall objective function becomes insignificant. In this approach, the lower-level problem can be thought of as a parametric optimization problem, where the main optimization variable is '$\bar{Z}_i$' and the price vector 'p' is a parameter. Then, for a perturbation in 'p' along direction 'd', the impact on the lower-level problem can be measured. Consider the following linear programming (LP):

$$\min_{w} \ \nabla_p f(p, \bar{Z})d + \nabla_{\bar{Z}} f(p, \bar{Z})w \tag{4.63a}$$

$$\text{s.t. } \nabla_p H_i(p, \bar{Z})d + \nabla_{\bar{Z}_i} H_i(p, \bar{Z}_i)w_i = 0 \tag{4.63b}$$

$$\nabla_p G_{i,act}(p, \bar{Z})d + \nabla_{\bar{Z}_i} G_{i,act}(p, \bar{Z}_i)w_i \leq 0 \tag{4.63c}$$

and $w = [w_1^T, \ldots, w_i^T, \ldots, w_m^T]^T$ denotes the feasible direction of $\bar{Z}$. The Lagrangian

associated with problem (4.63) can be stated as:

$$\mathcal{L}_{LP}(w, \nu_{LP}, \lambda_{LP}) = \sum_{i=1}^{m} \mathcal{L}_{LP,i}(w_i, \nu_{LP,i}, \lambda_{LP,i}) \tag{4.64a}$$

where:

$$\begin{aligned}\mathcal{L}_{LP,i}(w_i, \nu_{LP,i}, \lambda_{LP,i}) = &\big(\nabla_p f_i(p, \bar{Z})d + \nabla_{\bar{Z}} f_i(p, \bar{Z})w_i\big) + \nu_{LP,i}^T\big(\nabla_p H_i(p, \bar{Z})d + \\ &\nabla_{\bar{Z}_i} H_i(p, \bar{Z}_i)w_i\big) + \lambda_{LP,i}^T\big(\nabla_p G_{i,act}(p, \bar{Z})d + \nabla_{\bar{Z}_i} G_{i,act}(p, \bar{Z}_i)w_i\big)\end{aligned} \tag{4.64b}$$

and $\nu_{LP,i}$ and $\lambda_{LP,i}$ are Lagrange multipliers associated with equality and inequality constraints of subsystem 'i' in problem (4.63). Then, define the Lagrange dual of problem (4.63) as:

$$\mathcal{F}_{LP}(\nu_{LP}, \lambda_{LP}) = \inf_{w} \mathcal{L}_{LP}(w, \nu_{LP}, \lambda_{LP}) \tag{4.65a}$$

where:

$$\inf_{w} \mathcal{L}_{LP}(w, \nu_{LP}, \lambda_{LP}) = \sum_{i=1}^{m} \inf_{w_i} \mathcal{L}_{LP,i}(w_i, \nu_{LP,i}, \lambda_{LP,i}) \tag{4.65b}$$

$$\begin{aligned}&= \sum_{i=1}^{m} \big(\nabla_p f_i(p, \bar{Z}) + \nu_{LP,i}^T \nabla_p H_i(p, \bar{Z}) + \lambda_{LP,i}^T \nabla_p G_{i,act}(p, \bar{Z})\big)d + \\ &\quad \sum_{i=1}^{m} \inf_{w_i} \big(\nabla_{\bar{Z}} f_i(p, \bar{Z}) + \nu_{LP,i}^T \nabla_{\bar{Z}_i} H_i(p, \bar{Z}_i) + \lambda_{LP,i}^T \nabla_{\bar{Z}_i} G_{i,act}(p, \bar{Z}_i)\big)w_i\end{aligned} \tag{4.65c}$$

The dual function $\mathcal{F}_{LP}(\nu_{LP,i}, \lambda_{LP,i})$ can easily be determined analytically, since $\mathcal{L}_{LP,i}$ is a linear function of w_i and a linear function is bounded below only when its slope is zero. Thus, $\mathcal{F}_{LP}(\nu_{LP,i}, \lambda_{LP,i})$ is finite only when $\big(\nabla_{\bar{Z}} f_i(p, \bar{Z}) + \nu_{LP,i}^T \nabla_{\bar{Z}_i} H_i(p, \bar{Z}_i) +$

$\lambda_{LP,i}^T \nabla_{\bar{Z}_i} G_i(p, \bar{Z}_i)\big) = 0$ holds for all $i = 1, \ldots, m$. Considering (4.52), this can be summarized as:

$$\mathcal{F}_{LP}(\nu_{LP}, \lambda_{LP}) = \begin{cases} \sum_{i=1}^{m} \nabla_p \mathcal{L}_i(p, \bar{Z}_i, \nu_{LP,i}, \lambda_{LP,i}) d, & \sum_{i=1}^{m} \nabla_{\bar{Z}_i} \mathcal{L}_i(p, \bar{Z}_i, \nu_{LP,i}, \lambda_{LP,i}) = 0 \\ -\infty, & \text{otherwise} \end{cases} \tag{4.66a}$$

Thus, the dual problem of the LP defined in (4.63) can be formulated as:

$$\max_{\nu_{LP,i}, \lambda_{LP,i}} \quad \mathcal{F}_{LP}(\nu_{LP}, \lambda_{LP}) = \sum_{i=1}^{m} \nabla_p \mathcal{L}_i(p, \bar{Z}_i, \nu_{LP,i}, \lambda_{LP,i}) d \tag{4.67a}$$

$$\text{s.t.} \sum_{i=1}^{m} \nabla_{\bar{Z}_i} \mathcal{L}_i(p, \bar{Z}_i, \nu_{LP,i}, \lambda_{LP,i}) = 0 \tag{4.67b}$$

$$\lambda_{LP,i} \succeq 0 \tag{4.67c}$$

Remark 42 *The optimal value of the dual problem defined in (4.67), $\mathcal{F}_{LP}(\nu_{LP,i}^*, \lambda_{LP,i}^*)$, is the optimum value of the LP problem defined in (4.63), since the equality and inequality constraints of problem (4.63) are all affine. This can also be interpreted as a refinement of Slater's constraint qualification [12]. As a result, the following relation holds:*

$$\nabla_p f(p, \bar{Z}) d + \nabla_{\bar{Z}} f(p, \bar{Z}) w^* = \sum_{i=1}^{m} \nabla_p \mathcal{L}_i(p, \bar{Z}_i, \nu_{LP,i}^*, \lambda_{LP,i}^*) d \tag{4.68}$$

Considering Remark 42, and Assumptions 36, 37, and 38, the lower-level problem (4.51) can be written in the following quadratic programming (QP) form, w.r.t. a perturbation in 'p' along direction 'd':

$$\min_{w} \quad \begin{bmatrix} d^T, & w^T \end{bmatrix} \nabla^2_{p,\bar{Z}} \mathcal{L}(p, \bar{Z}, \nu_{LP}, \lambda_{LP}) \begin{bmatrix} d \\ w \end{bmatrix} \tag{4.69a}$$

$$\text{s.t. } \nabla_p H_i(p, \bar{Z}) d + \nabla_{\bar{Z}_i} H_i(p, \bar{Z}_i) w_i = 0 \tag{4.69b}$$

$$\nabla_p G_{i,act}(p, \bar{Z}) d + \nabla_{\bar{Z}_i} G_{i,act}(p, \bar{Z}_i) w_i \leq 0 \tag{4.69c}$$

$$\nabla_p f(p, \bar{Z}) d + \nabla_{\bar{Z}} f(p, \bar{Z}) w = \sum_{i=1}^{m} \nabla_p \mathcal{L}_i(p, \bar{Z}_i, \nu^*_{LP,i}, \lambda^*_{LP,i}) d \tag{4.69d}$$

where:

$$\nabla^2_{p,\bar{Z}} \mathcal{L}(p, \bar{Z}, \nu_{LP}, \lambda_{LP}) = \begin{bmatrix} 0 & \Theta \\ \Theta^T & \nabla^2_{\bar{Z}} \mathcal{L}(p, \bar{Z}, \nu_{LP}, \lambda_{LP}) \end{bmatrix} \tag{4.69e}$$

The set of optimal solutions of this QP is equal to the set of optimal solutions of the LP defined in (4.63), according to [39]. This would lead to the following lemma.

Lemma 43 *Let Assumptions 36, 37, and 38 hold for any direction 'd', then the QP defined in (4.69) has a unique optimal solution [39, 87].*

Following the work of [87], the CDMPC problem, defined in (4.51), can be formulated as a separable optimization problem to find the steepest descent direction:

$$\min_{d} \ \nabla_p F(p, \bar{Z}(k;p))d + \nabla_{\bar{Z}} F(p, \bar{Z}(k;p))w \tag{4.70a}$$

$$\text{s.t. } ||d||_\infty \leq 1 \tag{4.70b}$$

$$\min_{w} \ \begin{bmatrix} d^T, & w^T \end{bmatrix} \nabla^2_{p,\bar{Z}(k;p)} \mathcal{L}(p, \bar{Z}(k;p), \nu_{LP}, \lambda_{LP}) \begin{bmatrix} d \\ w \end{bmatrix} \tag{4.70c}$$

$$\text{s.t. } \nabla_p H_i(p, \bar{Z}(k;p))d + \nabla_{\bar{Z}_i} H_i(p, \bar{Z}_i(k;p))w_i = 0 \tag{4.70d}$$

$$\nabla_p G_{i,act}(p, \bar{Z}(k;p))d + \nabla_{\bar{Z}_i} G_{i,act}(p, \bar{Z}_i(k;p))w_i \leq 0 \tag{4.70e}$$

$$\nabla_p f(p, \bar{Z}(k;p))d + \nabla_{\bar{Z}} f(p, \bar{Z}(k;p))w = \sum_{i=1}^{m} \nabla_p \mathcal{L}_i(p, \bar{Z}_i(k;p), \nu^*_{LP,i}, \lambda^*_{LP,i})d \tag{4.70f}$$

where the coordinator finds the optimal direction of the price vector 'd' and the local controllers optimize over their corresponding feasible direction 'w_i' to minimize the plant-wide objective function $F(p, \bar{Z})$. The main idea behind the quadratic BLP (QBLP) problem (4.70) is to find a feasible set of directions 'd' and 'w_i' that results in decreasing the upper-level objective function of the original BLP problem (4.51), i.e. $\nabla_p F(p, \bar{Z}(k;p))d + \nabla_{\bar{Z}} F(p, \bar{Z}(k;p))w < 0$.

Remark 44 *For the scenario described in Remark 41, the QBLP problem (4.70) would be simplified to:*

$$\min_{d} \ \nabla_p(-\Theta\bar{Z})d + \left(- \begin{bmatrix} Q & & \\ & R & \\ & & 0 \end{bmatrix} \bar{Z} + \begin{bmatrix} X_{sp}^T Q \\ 0 \\ 0 \end{bmatrix} - \Theta^T p \right)^T w \tag{4.71a}$$

$$s.t. \ ||d||_\infty \leq 1 \tag{4.71b}$$

$$\min_{w} \ \begin{bmatrix} d^T, & w^T \end{bmatrix} \nabla^2_{p,\bar{Z}(p)} \begin{bmatrix} 0 & \Theta \\ \Theta^T & \begin{bmatrix} Q & \\ & R \end{bmatrix} \\ & \begin{bmatrix} & \\ & 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} d \\ w \end{bmatrix} \tag{4.71c}$$

$$s.t. \ A_i w_i = 0 \tag{4.71d}$$

$$A_{i,act} w_i \leq 0 \tag{4.71e}$$

$$\left(\begin{bmatrix} Q & & \\ & R & \\ & & 0 \end{bmatrix} \bar{Z} - \begin{bmatrix} X_{sp}^T Q \\ 0 \\ 0 \end{bmatrix} + \Theta^T p \right)^T w = 0 \tag{4.71f}$$

According to Lemma 43, the lower-level problem of the QBLP (4.70) can be replaced by its KKT conditions to form the single level optimization:

$$\min_{d,w,\nu_{QP},\lambda_{QP}} \quad \nabla_p F(p,\bar{Z}(k;p))d + \nabla_{\bar{Z}} F(p,\bar{Z})w \tag{4.72a}$$

$$\text{s.t. } ||d||_\infty \leq 1 \tag{4.72b}$$

$$2(\nabla^2_{\bar{Z}}\mathcal{L}(p,\bar{Z},\nu_{LP},\lambda_{LP})w + \Theta^T d) + \begin{bmatrix} \nabla_{\bar{Z}} H(p,\bar{Z}(k;p)) \\ \nabla_{\bar{Z}} f(p,\bar{Z}(k;p)) \end{bmatrix}^T \nu_{QP} + G_{act}(p,\bar{Z}(k;p))^T\lambda_{QP} = 0 \tag{4.72c}$$

$$\nabla_p H_i(p,\bar{Z}(k;p))d + \nabla_{\bar{Z}_i} H_i(p,\bar{Z}_i(k;p))w_i = 0 \tag{4.72d}$$

$$\nabla_p f(p,\bar{Z}(k;p))d + \nabla_{\bar{Z}} f(p,\bar{Z}(k;p))w - \sum_{i=1}^{m} \nabla_p \mathcal{L}_i(p,\bar{Z}_i(k;p),\nu^*_{LP,i},\lambda^*_{LP,i})d = 0 \tag{4.72e}$$

$$\nabla_p G_{i,act}(p,\bar{Z}(k;p))d + \nabla_{\bar{Z}_i} G_{i,act}(p,\bar{Z}_i(k;p))w_i \leq 0 \tag{4.72f}$$

$$\lambda^T_{QP,i}\big(\nabla_p G_{i,act}(p,\bar{Z}(k;p))d + \nabla_{\bar{Z}_i} G_{i,act}(p,\bar{Z}_i(k;p))w_i\big) = 0 \tag{4.72g}$$

$$\lambda_{QP,i} \geq 0 \tag{4.72h}$$

where $\nu_{QP} = [\nu^T_{QP,1},\ldots,\nu^T_{QP,m}]^T$ and $\lambda_{QP} = [\lambda^T_{QP,1},\ldots,\lambda^T_{QP,m}]^T$ are Lagrange multipliers associated with equality and inequality constraints of the QP problem (4.69), respectively.

Using a similar approach as in Remark (40), a smoothed version of (4.72) is formulated as:

$$\min_{d,w,\nu_{QP},\lambda_{QP}} \quad \nabla_p F(p,\bar{Z}(k;p))d + \nabla_{\bar{Z}} F(p,\bar{Z}(k;p))w \tag{4.73a}$$

$$\text{s.t. } ||d||_\infty \leq 1 \tag{4.73b}$$

$$2(\nabla^2_{\bar{Z}}\mathcal{L}(p,\bar{Z},\nu_{LP},\lambda_{LP})w + \Theta^T d) + \begin{bmatrix} \nabla_{\bar{Z}} H(p,\bar{Z}(k;p)) \\ \nabla_{\bar{Z}} f(p,\bar{Z}(k;p)) \end{bmatrix}^T \nu_{QP} + G_{act}(p,\bar{Z}(k;p))^T\lambda_{QP} = 0 \tag{4.73c}$$

$$\nabla_p H_i(p,\bar{Z}(k;p))d + \nabla_{\bar{Z}_i} H_i(p,\bar{Z}_i(k;p))w_i = 0 \tag{4.73d}$$

$$\nabla_p f(p,\bar{Z}(k;p))d + \nabla_{\bar{Z}} f(p,\bar{Z}(k;p))w - \sum_{i=1}^{m} \nabla_p \mathcal{L}_i(p,\bar{Z}_i(k;p),\nu^*_{LP,i},\lambda^*_{LP,i})d = 0 \tag{4.73e}$$

$$\nabla_p G_{i,act}(p,\bar{Z}(k;p))d + \nabla_{\bar{Z}_i} G_{i,act}(p,\bar{Z}_i(k;p))w_i \leq 0 \tag{4.73f}$$

$$\Pi\big(-\nabla_p G_{i,act}(p,\bar{Z}(k;p))d - \nabla_{\bar{Z}_i} G_{i,act}(p,\bar{Z}_i(k;p))w_i, \lambda_{QP,i}, \eta\big) = 0 \tag{4.73g}$$

$$\lambda_{QP,i} \geq 0 \tag{4.73h}$$

and solve this problem with a similar approach used in Algorithm 15, i.e. using an interior-point method. Once the optimal solution to the QBLP (4.70) is found, the price vector 'p' and local optimization variables '$\bar{Z}_i$' need to be updated along optimal directions 'd^s' and 'w_i^s' for the next iteration. Consider 's' the current iteration number, the following feasibility problem finds the minimum positive step size 'α' such that $F(p+\alpha d, \bar{Z}(k;p+\alpha d)) < F(p,\bar{Z}(k;p))$ and the solution '$\bar{Z}_i^{s+1}$' remains feasible:

$$\min_{\alpha} \ \alpha \tag{4.74a}$$

$$\text{s.t. } \alpha > 0 \tag{4.74b}$$

$$F(p^s+\alpha d^s, \bar{Z}(k;p^s+\alpha d^s)) < F(p^s,\bar{Z}(k;p^s)) \tag{4.74c}$$

$$\bar{Z}_i(k;p^s+\alpha d^s) \in \mathcal{S}_i(p^s+\alpha d^*) \tag{4.74d}$$

In other words, the main intention to perform the line search (4.74) is to find a proper step size along direction d, such that: the solution to the lower-level problem stays feasible.

Finally, the price vector and local optimization variables are updated according to the optimal value of step size α^s:

$$p^{s+1} = p^s + \alpha^s d^s \tag{4.75}$$

$$\bar{Z}_i^{s+1} = \bar{Z}(k; p^s + \alpha^s d^s) \tag{4.76}$$

The algorithm to find a feasible descent direction to the BLP (4.51) is listed in Algorithm 16:

Algorithm 16: Feasible descent direction algorithm to solve the BLP (4.51)

Input: $0 < \eta_{\min} < \eta^0 \leq 1$, $l = 0$, $s = 0$, $\beta \in (0,1)$, $\bar{Z}_i^l(k)$, p^l, $\mathcal{J}^0_{G_{i,act}}$;

while $(\nabla_p F(p, \bar{Z}(p))d + \nabla_{\bar{Z}} F(p, \bar{Z})w < 0)$ **do**

  while $(\eta^l \geq \eta_{\min})$ **do**

    Coordinator - Local MPCs: Solve the NLP problem (4.73), i.e. using the interior-point method [6] ;

    $\eta^{l+1} = \beta\eta^l$;

    $l = l + 1$;

    Coordinator - Local MPCs: update d^l and w^l;

  $s = s + 1$;

  Set $d^s = d^l$ and $w^s = w^l$;

  Coordinator - Local MPCs: Solve problem (4.74);

  Coordinator: Update $p^{s+1} = p^s + \alpha^s d^s$;

  Local MPCs: Update $\bar{Z}_i^{s+1} = \bar{Z}(k; p^s + \alpha^s d^s)$;

  for $j \in \mathcal{J}_{G_i}$ **do**

    if $G_{i,j}(p^s, \bar{Z}_i^s) = 0$ **then**

      Coordinator: $\mathcal{J}^s_{G_{i,act}} = \{j \cup \mathcal{J}^s_{G_{i,act}}\}$;

  Set $l = 0$;

Output: p^*, $\bar{Z}_i^*(k)$;

4.4.2 Convergence Analysis

In this section, convergence property of the proposed CDMPC algorithm to a local optimal solution of BLP problem (4.51) is studied. First, the initialization stage is studied, in which the initial set of active constraints $\mathcal{J}_{G_{i,act}}(p)$ are determined based on a series of NLP problems (4.56), for a decreasing sequence of η_i.

Theorem 45 *For any value of η, when the solution to (4.60) belongs to the set $\bar{\mathcal{S}}(p,\eta)$, then for every $\hat{\eta}$ there exists a compact set $\bar{\mathcal{C}}(\hat{\eta})$ such that $\bar{\mathcal{S}}(p,\eta) \subseteq \bar{\mathcal{C}}(\hat{\eta})$ holds for all $\eta \in (0,\hat{\eta}]$.*

Proof. According to Assumption 35, for every solution that belongs to $\mathcal{S}_\eta$ and $\eta > 0$, there exist a unique bounded optimal price vector. Also, based on the properties of the function Π in (4.58), if a feasible solution is found w.r.t. the lower-level constraints, it would belong to a compact set according to Assumption 34. Additionally, since regularity (Assumption 36) holds in the lower-level problem, the Lagrange multipliers ν_i and λ_i would be bounded and belong to a compact set. This follows the conclusion that $\bar{\mathcal{S}}(p,\eta) \subseteq \bar{\mathcal{C}}(\hat{\eta})$ holds for every $\eta \in (0,\hat{\eta}]$. ■

Remark 46 *Based on Theorem 45, the solution set $\bar{\mathcal{S}}(p,\eta(k))$ is a non-empty and compact set for all values of $0 \leq \eta \leq 1$ [33]. This produces a continuous central path [33, 52] towards the solution of problem (4.55), as a function of the decreasing sequence η starting from a value between $(0,1)$ towards 0.*

Once the initial set of active constraints is identified, the QBLP problem (4.70) along with step size calculation (4.74) are solved iteratively to find a unique local solution to the original BLP (4.51). In the following theorem, it is proved that the CDMPC algorithm is globally convergent to a unique local optimal solution of the plant wide problem (4.51).

Theorem 47 *Let assumptions 33, 34, 35, 36, 37, and 38 hold; then:*

(i) *For the sequence $\eta \to 0$, the initialization stage, i.e. Algorithm (15), finds a unique guess of the set of active constraints for the BLP problem (4.51).*

(ii) *Denote $(p^*, \bar{Z}^*)$ to be an optimal solution to the BLP problem (4.51). Then, the following holds:*

$$\nabla_p F(p^*, \bar{Z}(k,p^*))d + \nabla_{\bar{Z}} F(p^*, \bar{Z}(k,p^*))w(p^*,d) \geq 0 \tag{4.77}$$

where 'd' and 'w' are the optimal solutions of the QBLP problem (4.70).

(iii) *The proposed CDMPC algorithm is globally convergent to a unique optimal solution of (4.51).*

Proof. **(i)** Let the stated assumptions hold, in addition consider Theorem 45 and the central path defined by Remark 46. Then for any value of the decreasing sequence $\eta \to 0$, there exists a bounded solution that forms a compact set and converges to a local optimal solution of problem (4.60) [33]. The interior-point method is a globally convergent Algorithm [6] and based on Assumption 35 a unique local solution exists for the price vector of the single level optimization problem. Thus a unique set of active constraints for problem (4.51) can be determined during the initialization stage of the CDMPC algorithm.

(ii) Let $(p^*, \bar{Z}^*)$ be an optimal solutions to the BLP problem (4.51), then the following holds:

$$F(p^*, \bar{Z}(k, p^*)) \leq F(p^* + \alpha d, \bar{Z}(k, p^* + \alpha d)), \qquad 0 < \alpha \leq \hat{\alpha} \tag{4.78}$$

Furthermore, based on the given assumptions, there always exists an $\hat{\alpha} > 0$ such that the solution set of the lower-level problem for $0 < \alpha \leq \hat{\alpha}$ is not empty [39]. Rearranging (4.78), and dividing by α the following relation can be derived for the upper-level objective function:

$$\lim_{\alpha \to 0} \frac{(F(p^* + \alpha d, \bar{Z}(k; p^* + \alpha d)) - F(p^*, \bar{Z}(k; p^*)))}{\alpha} \geq 0 \tag{4.79}$$

which is equivalent to the directional derivative of the upper-level objective function:

$$\nabla_p F(p^*, \bar{Z}(k; p^*)) \frac{dp}{d\alpha} + \nabla_{\bar{Z}} F(p^*, \bar{Z}(k; p^*)) \frac{d\bar{Z}}{d\alpha} \geq 0 \tag{4.80}$$

but the QBLP problem (4.70) has a unique solution [87] that refines (4.80) into:

$$\nabla_p F(p^*, \bar{Z}(k; p^*))d + \nabla_{\bar{Z}} F(p^*, \bar{Z}(k; p^*))w(p^*, d) \geq 0 \tag{4.81}$$

(iii) Based on part (ii) of this proof, the stopping criteria for the proposed CDMPC algorithm satisfies (4.81) when no further descent direction can be found. Additionally, since there is a unique optimal price for the well-posed BLP problem (4.51), according to assumptions 35 and 33, a unique local optimal solution is obtained in the initialization stage (according to part(i) of this proof) to solve the NLP problem (4.60). Also, a unique optimal solution can be found in the iterative stage (according to part(ii) of this proof) including the QLBP problem (4.70) and step size calculation (4.74). Thus, the whole CDMPC algorithm is globally convergent to unique local optimal solution of the BLP problem (4.51). ■

4.5 Stability Analysis

In this work, the idea of [80, 41, 42] was followed to implement stability criteria for the centralized MPC and the proposed CDMPC schemes. The main idea is to provide the criteria without defining any terminal constraints or any terminal cost functions for the MPC optimization problem. This criteria is applicable to MPC schemes subject to admissible sets of bound and/or mixed states and input constraints [80].

Denote the optimum values of state and input variables of the centralized MPC optimization problem (4.27) as the open-loop optimal solution, and the first move of the system as the receding horizon control (RHC) action. Then, $u_{RHC}(N, x(n))$ is the RHC action calculated by solving (4.27) with prediction horizon N and the initial state $x(n)$.

Define a dynamic programming value function [80] based on the RHC trajectory of

the centralized MPC as:

$$V_N^C\big(x(n)\big) = \sum_{n=0}^{N-1} l^C\Big(x(n), u_{RHC}(N-n, x(n))\Big) \tag{4.82a}$$

where l^C is a predefined stage cost based on the centralized trajectory, defined as:

$$l^C\Big(x(n), u_{RHC}(N-n, x(n))\Big) = \sum_{i=1}^{m} \Big(\big(x_i(n) - x_{i,sp}(k)\big)^T Q_{ii}(n)\big(x_i(n) - x_{i,sp}(k)\big) + \big(u_{i,RHC}(N-n, x_i(n))\big)^T R_{ii}(n)\big(u_{i,RHC}(N-n, x_i(n))\big)\Big) \tag{4.82b}$$

with Q_{ii} and R_{ii} defined over the prediction horizon as:

$$Q_{ii} = \begin{bmatrix} Q_{ii}(1) & & \\ & \ddots & \\ & & Q_{ii}(N) \end{bmatrix}, \quad R_{ii} = \begin{bmatrix} R_{ii}(1) & & \\ & \ddots & \\ & & R_{ii}(N) \end{bmatrix} \tag{4.82c}$$

Based on (4.82), in order to calculate V_N^C at the current sampling time k, namely $V_N^C\big(x(k)\big)$, a multi-step calculation is performed and move forward in prediction horizon. The procedure to calculate (4.82) is listed in Algorithm 17:

Algorithm 17: The algorithm to calculate the finite-time value function (4.82)

Initialization: prediction horizon N, number of sub-systems m, initial state $x(n)$, Q_{ii}, R_{ii}, $V_N^C = 0$;

for $n := 0$ ***to*** $N-1$ **do**

Perform the plant-wide optimization problem (4.27) with prediction horizon $N-n$ and initial value $x_i(0) = x_i(n)$;

Update the RHC control action move $u_{RHC}(N-n, X(n))$;

Calculate the centralized MPC stage cost $l^C\Big(x(n), u_{RHC}(N-n, x_i(n))\Big)$;

$V_N^C = V_N^C + l^C\Big(x(n), u_{RHC}(N-n, x(n))\Big)$;

Apply $u_{RHC}(N-n, x(n))$ to the model and update $x_i(n+1)$;

Output: $V_N^C(x(n)) \leftarrow V_N^C$;

Remark 48 *The difference between the open-loop control and receding horizon control (RHC) trajectories for a dynamic programming problem at sampling time t_k is depicted in Figure 4.2. In this example, at the sampling time t_k the control system performs $N = 5$ open-loop optimization problems. Once the open-loop trajectory for $N = 5$ is found, the first control action is applied to the model (and not to the plant itself). The next open-loop optimization is performed with $N = 4$ from the initial move previously computed. This process continues until five RHC points are obtained and the value function V_N^C is calculated.*

Lemma 49 *Consider the optimization problem (4.27) at sampling time t_k, provided that the stage cost l^C is a positive definite function, if there exists a trajectory-based function V_N^C such that*

$$V_N^C(x(k)) - V_N^C(x(k+1)) \geq l^C\Big(x(k), u_{RHC}(N, x(k))\Big) \tag{4.83}$$

then V_N^C is a Lyapunov function for the plant-wide MPC system (4.27), such that the per-

formance of the finite-time receding horizon closed-loop system tends to the performance of infinite-time receding horizon control [80].

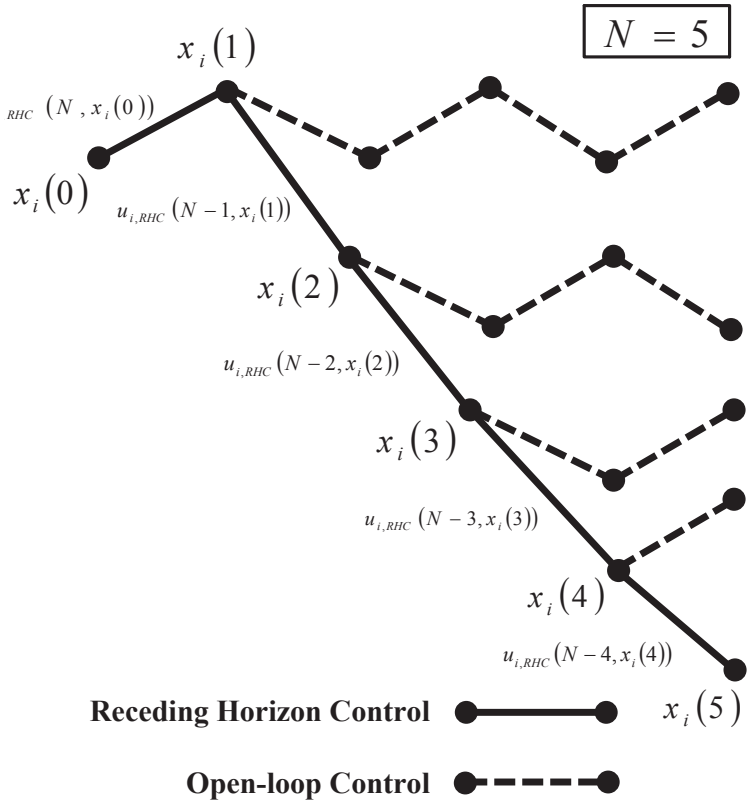


Figure 4.2: Receding horizon control (RHC) vs open-loop control trajectories for subsystem i with a typical prediction horizon length of $N = 5$.

Based on Lemma 49, an a posteriori algorithm can be presented to adjust the minimum required prediction horizon length N adaptively, at the current sampling time t_k. This procedure is listed in Algorithm 18:

Algorithm 18: A posteriori adaptive horizon algorithm [80] for the MPC (4.27)

Initialization: prediction horizon N, number of sub-systems m, initial state $x(n)$, Q_{ii}, R_{ii};

Calculate $V_N^C(x(n))$ using Algorithm 17 with $\big(N, m, x(n), Q_{ii}, R_{ii}\big)$;

Calculate $l^C\big(x(n), u_{RHC}(N, x(n))\big)$ using (4.82);

Apply $u_{RHC}(N, x(n))$ to the model and update $x(n+1)$;

Calculate $V_N^C(x(n+1))$ using Algorithm 17 with $\big(N, m, x(n+1), Q_{ii}, R_{ii}\big)$;

if $V_N^C(x(n)) \leq V_N^C(x(n+1))$ **then**

  Prompt: Solution may be unstable;

else if $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)} \geq 1$ **then**

  Perform Horizon Shortening (Algorithm 19);

else

  Perform Horizon Prolongation (Algorithm 20);

end

Output: Apply the RHC action to the plant based on the accepted value of N;

According to Algorithm 18, as long as the condition $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)} \geq 1$ is satisfied, the length of prediction horizon is decreased so that the minimum amount of prediction horizon required is found. This procedure is called the horizon shortening strategy, which is explained in Algorithm 19:

Algorithm 19: Horizon Shortening Algorithm for the MPC problem (4.27)

Initialization: prediction horizon N, number of sub-systems m, initial state $x(k)$, Q_{ii}, R_{ii};

while $N \geq 2$ **do**

Save the current trajectory of the open-loop system;

$N = N - 1$;

Calculate $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)}$ with $\left(N, m, x(k), Q_{ii}, R_{ii}\right)$;

Save the shortened horizon trajectory of the open-loop system;

if $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)} < 1$ **then**

$N = N + 1$;

Restore the current stored trajectory of the open-loop system;

STOP;

else

Save the shortened horizon trajectory as the current trajectory;

Output: Current trajectory of the system, and N.

On the other hand, if $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)} < 1$ the length of prediction horizon is increased in order to find the minimum N that satisfies $\frac{V_N^C(x(n)) - V_N^C(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)} \geq 1$. This procedure is listed in Algorithm 20:

Algorithm 20: Horizon Prolongation Algorithm for the MPC problem (4.27)

Initialization: prediction horizon N, number of sub-systems m, initial state $x(k)$, Q_{ii}, R_{ii};

while $\frac{V_N^C(x(n))-V_N^C(x(n+1))}{l^C\left(x(n),u_{RHC}(N,x(n))\right)} < 1$ **do**

$N = N + 1$;

Calculate $\frac{V_N^C(x(n))-V_N^C(x(n+1))}{l^C\left(x(n),u_{RHC}(N,x(n))\right)}$ with $\left(N, m, x(k), Q_{ii}, R_{ii}\right)$;

Save the prolonged horizon trajectory as the current trajectory of the open-loop system;

Output: Current trajectory of the system, and N.

4.5.1 Stability of the CDMPC scheme

Similar to stability analysis of the centralized trajectory, a stability criterion can be stated without terminal costs or constraints for the proposed CDMPC schemes based on adaptive horizon techniques [80, 41, 42].

Denote $Z_{opt}(k) = [X_{opt}(k)^T, U_{opt}(k)^T, V_{opt}(k)^T]^T$ as the open-loop control solution of the CDMPC problem. Then, define the receding horizon control (RHC) move as: $u_{opt_{i,RHC}}$-$(N, x_{opt_i}(n))$, which is calculated by solving either problem (4.30) or (4.51) with prediction horizon length of N and the initial value $x_i(n)$.

Similar to (4.82), a dynamic programming value function is defined for the CDMPC problem:

$$V_N^D\big(x(n)\big) = \sum_{n=0}^{N-1} l^D\Big(x_{opt_i}(n), u_{RHC}(N-n, x(n))\Big) \tag{4.84a}$$

where the stage cost l^D is defined as:

$$l^D\Big(x_{opt}(n), u_{RHC}(N-n, x(n)))\Big) = \sum_{i=1}^{m} \Big(x_{opt_i}(n) - x_{i,sp}(k)\Big)^T Q_{ii}(n)\Big(x_{opt_i}(n) - x_{i,sp}(k)\Big)$$

$$+\Big(u_{i,RHC}(N-n,x_i(n))\Big)^T R_{ii}(n)\Big(u_{i,RHC}(N-n,x_i(n))\Big) \tag{4.84b}$$

Therefore, in order to calculate V_N^D at the current sampling time k, namely $V_N^D\big(x(k)\big)$, a multi-step calculation has to be performed forward in time. The procedure to calculate (4.84) is listed in Algorithm 21:

Algorithm 21: The algorithm to calculate the finite-time value function (4.84)

Initialization: $n = 0$, prediction horizon N, number of sub-systems m, initial state $x(n) = x(k)$, Q_{ii}, R_{ii}, $V_N^D = 0$;

for $n := 0$ ***to*** $N-1$ **do**

  if *inequality constraints present in local MPC problems* **then**

    Coordinator - Local MPCs: Perform the CDMPC optimization problem (4.51) with $N-n$ and initial value $x_i(0) = x_i(n)$;

  else

    Coordinator - Local MPCs: Perform the CDMPC optimization problem (4.30) with $N-n$ and initial value $x_i(0) = x_i(n)$;

  Coordinator - Local MPCs: Update the RHC control action move $u_{opt_{RHC}}(N-n, x_{opt}(n))$;

  Coordinator: Calculate $l^D\Big(x_{opt}(n), uopt_{RHC}(N-n, x_{opt}(n))\Big)$, and $V_N^D = V_N^D + l^D\Big(x_{opt}(n), u_{opt_{RHC}}(N-n, x_{opt}(n))\Big)$;

  Local MPCs: Apply $u_{opt_{RHC}}(N-n, x(n))$ to the internal model;

  Local MPCs: Calculate $x_{opt}(n+1)$;

Output: $V_N^D(x(n)) \leftarrow V_N^D$;

Remark 50 *Similar to Lemma 49, consider the individual distributed MPC controllers in the optimization problems (4.30) or (4.51) at sampling time t_k, provided that the stage*

cost l^D is a positive definite function, if the following is satisfied:

$$V_N^D(x(k)) - V_N^D(x(k+1)) \geq l^D\Big(x(k), u_{opt_{RHC}}(N, x(k))\Big) \tag{4.85}$$

then V_N^D is a finite-time Lyapunov function for the proposed CDMPC schemes that tends to performance of an equivalent infinite horizon problem. Note that, all local MPC controllers are assumed to have the same length of prediction horizons, which are dictated by the minimum required horizon length to satisfy (4.85).

Based on Remark 50, an a posteriori algorithm can be presented to satisfy (4.85), at the sampling time t_k with minimum required prediction horizon length, i.e. N. This procedure is explained in Algorithm 22:

Algorithm 22: A posteriori adaptive horizon algorithm for the CDMPC schemes

Initialization: $n = 0$, prediction horizon N, number of sub-systems m, initial state $x(n) = x(k)$, Q_{ii}, R_{ii};

Coordinator - Local MPCs: Calculate $V_N^D(x(n))$ using Algorithm 21 with $\big(N, m, x_{opt}(n), Q_{ii}, R_{ii}\big)$;

Local MPCs: Apply $u_{opt_{RHC}}(N, x(n))$ to the internal model;

Local MPCs: Calculate $x_{opt}(n+1)$;

Coordinator - Local MPCs: Calculate $V_N^D(x(n+1))$ using Algorithm 21 with $\big(N, m, x_{opt}(n+1), Q_{ii}, R_{ii}\big)$;

if $V_N^D(x(n)) \leq V_N^D(x(n+1))$ **then**

  Local MPCs: Prompt: Solution may be unstable;

else

  if $\frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x_{opt}(n), u_{opt_{RHC}}(N, x(n))\right)} \geq 1$ **then**

    Coordinator - Local MPCs: Perform Algorithm 23;

  else

    Coordinator - Local MPCs: Perform Algorithm 24;

Output: X_{opt} based on the accepted value of N;

According to Algorithm 22, as long as the condition $\frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x(n), u_{RHC}(N, x(n))\right)} \geq 1$ is satisfied, the prediction horizon length is decreased so that the minimum required prediction horizon is found. This procedure is called the horizon shortening strategy, which is explained in Algorithm 23:

Algorithm 23: Horizon Shortening Algorithm for the CDMPC schemes

Initialization: $n = 0$, prediction horizon N, number of sub-systems m, initial state $x(n) = x(k)$, Q_{ii}, R_{ii};

while $N \geq 2$ **do**

Local MPCs: Save the current trajectory of the open-loop system;

Local MPCs: $N = N - 1$;

Coordinator - Local MPCs: Calculate $\frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x_{opt}(n), u_{RHC}(N, x(n))\right)}$ with $\left(N, m, x_{opt}(n), Q_{ii}, R_{ii}\right)$;

Local MPCs: Save the shortened horizon trajectory of the open-loop system;

if $\frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x_{opt}(n), u_{RHC}(N, x(n))\right)} < 1$ **then**

Local MPCs: $N = N + 1$;

Local MPCs: Restore the current stored trajectory of the open-loop system;

STOP;

else

Local MPCs: Save the shortened horizon trajectory as the current trajectory;

Output: Current trajectory of the system, and N.

On the other hand, if $\frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x(n), u_{RHC}(N, x(n))\right)} < 1$ the prediction horizon length is increased to find the minimum N that satisfies $\frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^C\left(x(n), u_{RHC}(N, x(n))\right)} \geq 1$. This procedure is listed in Algorithm 24:

Algorithm 24: Horizon Prolongation Algorithm for the CDMPC schemes

Initialization: $n = 0$, prediction horizon N, number of sub-systems m, initial state $x(n) = x(k)$, Q_{ii}, R_{ii};

while $\frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x_{opt}(n), u_{RHC}(N, x(n))\right)} < 1$ **do**

Local MPCs: $N = N + 1$;

Coordinator - Local MPCs: Calculate $\frac{V_N^D(x(n)) - V_N^D(x(n+1))}{l^D\left(x_{opt}(n), u_{RHC}(N, x(n))\right)}$ with $\left(N, m, x_{opt}(n), Q_{ii}, R_{ii}\right)$;

Local MPCs: Save the prolonged size trajectory as the current trajectory of the open-loop system;

Output: Current trajectory of the system, and N.

4.5.2 The overall CDMPC Algorithm

In this section, a summary of the CDMPC algorithm applied to the plant-wide problem is provided. The hierarchical dual decomposition of the hypothetical centralized controller defined in (4.27) is related with the modification applied to the corresponding existing network of local decentralized MPC controllers. The modification can be interpreted as penalizing local violations of the overall interaction equality constraints via the price vector. The local MPC controllers formulate their own optimization problems and send the required local information conditions to the coordination level. The coordinator receives decides whether to apply the analytic CDMPC or feasible descent direction CDMPC based on existence of inequality constraints in subproblem formulations. Then, the coordinator, based on the chosen CDMPC method, finds the minimum required prediction horizon to ensure stability of the closed-loop system. Once the plant-wide optimum solution, i.e. $Z(k)$, is found, the receding horizon control action is applied to the plant. The overall algorithm is presented in Algorithm 25:

Algorithm 25: The overall CDMPC algorithm

Initialization: Number of sub-systems m, initial state $x(k)$, positive definite matrix Q, positive definite matrix R, initial prediction horizon N_0, $0 < \eta_{\min} < \eta^0 \leq 1$, and $\beta \in (0, 1)$;

if *inequality constraints present in local MPC problems* **then**

  Local MPCs: Given $x(k)$ and N_0, formulate local KKT conditions of problem (4.51);

else

  Local MPCs: Given $x(k)$ and N_0, formulate local KKT conditions of problem (4.30);

Coordinator - Local MPCs: Perform Algorithm 22 with $\big(x(k)$, Q, R, N_0, $\eta_{\min}, \eta^0$, $\beta\big)$;

Coordinator: Send the optimal price vector and N to the local controllers;

Local MPCs: Calculate the optimum local vector of variables $Z_i^*(t_k)$ using $p^*(t_k)$ and N;

Local MPCs: Apply the RHC action to the plant;

Local MPCs: Update $N_0 = N$;

4.6 Simulation Case Studies

In this section, two benchmarks were studied to illustrate the CDMPC algorithms discussed in this chapter. The first case study is a forced-circulation evaporator process [50], which demonstrates an open-loop stable system without any inequality constraints. The second case study is a process composed of two interconnected CSTRs [96], which represents an open-loop unstable system subject to bounds on the control actions. The optimization problems were formulated inside MATLAB using the YALMIP optimization interface [62], in which IPOPT [6] is employed as the main optimization solver.

Simulations are performed on an Intel Core-i7 processor with 8 GB of memory under Microsoft Windows 7 operating system.

4.6.1 Evaporator Process

The forced-circulation evaporator system [50] is used as a benchmark to show the efficiency of the CDMPC strategy described in section 4.3, where no inequality constraints are present in local MPCs. This system is illustrated in Fig. 4.3. The dynamics of the

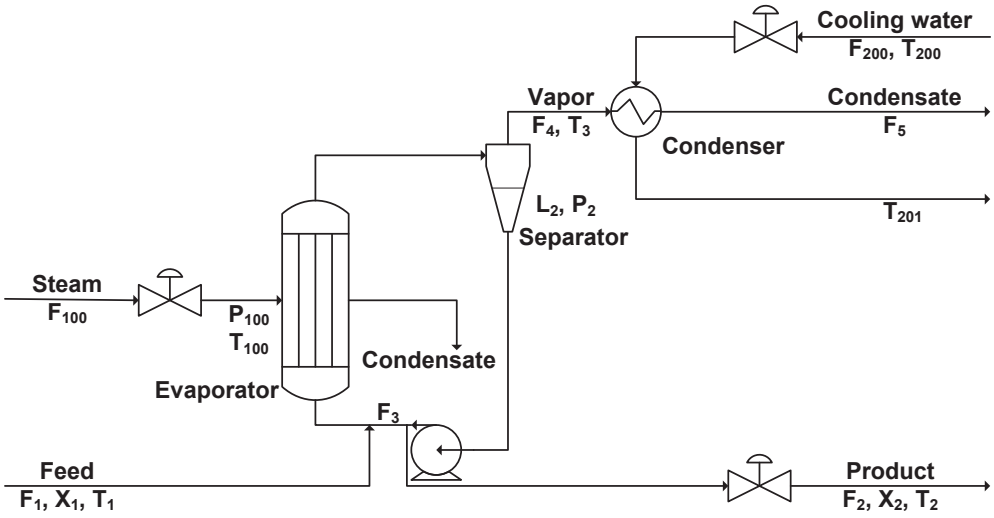


Figure 4.3: The Forced-Circulation Evaporator Process

forced-circulation evaporator consists of three measured states (L_2: separator level $[m]$, X_2: product composition $[\%]$, and P_2: operating pressure $[kPa]$), three input variables (F_2: product flow rate $[kg/min]$, P_{100}: steam pressure $[kPa]$, and F_{200}: cooling water flow rate $[kg/min]$). In Fig. 4.3, T_2 is product temperature $[^{\circ}C]$, T_{100} is steam temperature $[^{\circ}C]$, Q_{100} is heater duty $[kW]$, F_{100} is steam flow rate $[kg/min]$, T_3 is vapor temperature, F_4 is vapour temperature $[^{\circ}C]$, Q_{200} is condenser duty $[kW]$, T_{201} is cooling water outlet temperature $[^{\circ}C]$, and F_5 is condensate flow rate $[kg/min]$. The continuous-time plant

model of this process can be written as the following state-space formulation:

$$\dot{x} = \begin{bmatrix} 1 & 0.10445 & 0.37935 \\ 0 & -0.1 & 0 \\ 0 & -0.01034 & -0.054738 \end{bmatrix} x + \begin{bmatrix} -0.1 & 0.37266 & 0 \\ -0.1 & 0 & 0 \\ 0 & 0.036914 & -0.0075272 \end{bmatrix} u \tag{4.86}$$

Accordingly, two sub-systems are defined in the overall process model:

$$x^T = ([x_{1,1}, x_{1,2}]|x_{2,1})^T = ([L_2, X_2]|P_2)^T \tag{4.87}$$

$$u^T = ([u_{1,1}, u_{1,2}]|u_{2,1})^T = ([F_2, P_{100}]|F_{200})^T \tag{4.88}$$

and the corresponding discretized-time model, for a sampling time of $T_s = 1\ [min]$, can be derived as:

$$\hat{x}(k+1) = \left[\begin{array}{cc|c} 1 & 0.0975 & 0 \\ 0 & 0.9048 & 0 \\ \hline 0 & -0.0096 & -0.9467 \end{array}\right] \hat{x}(k) + \left[\begin{array}{cc|c} -0.1050 & 0.3795 & -0.0014 \\ -0.0952 & 0 & 0 \\ \hline 0.0005 & 0.0359 & -0.0073 \end{array}\right] \hat{u}(k) \tag{4.89}$$

According to the analytic derivations in section 4.3, the CDMPC algorithm is capable of perfectly tracking the centralized trajectory. Then, in the following results, the centralized MPC and CDMPC represent a single trajectory that is compared to the corresponding decentralized MPC. The comparison between the required length of the predictions horizons (N) between the decentralized MPC, the centralized MPC and the CDMPC trajectories is depicted in Figure 4.4. This shows after a certain amount of simulation time the centralized MPC and CDMPC schemes roughly require $N = 18$ while the minimum prediction horizon required by decentralized is $N = 15$.

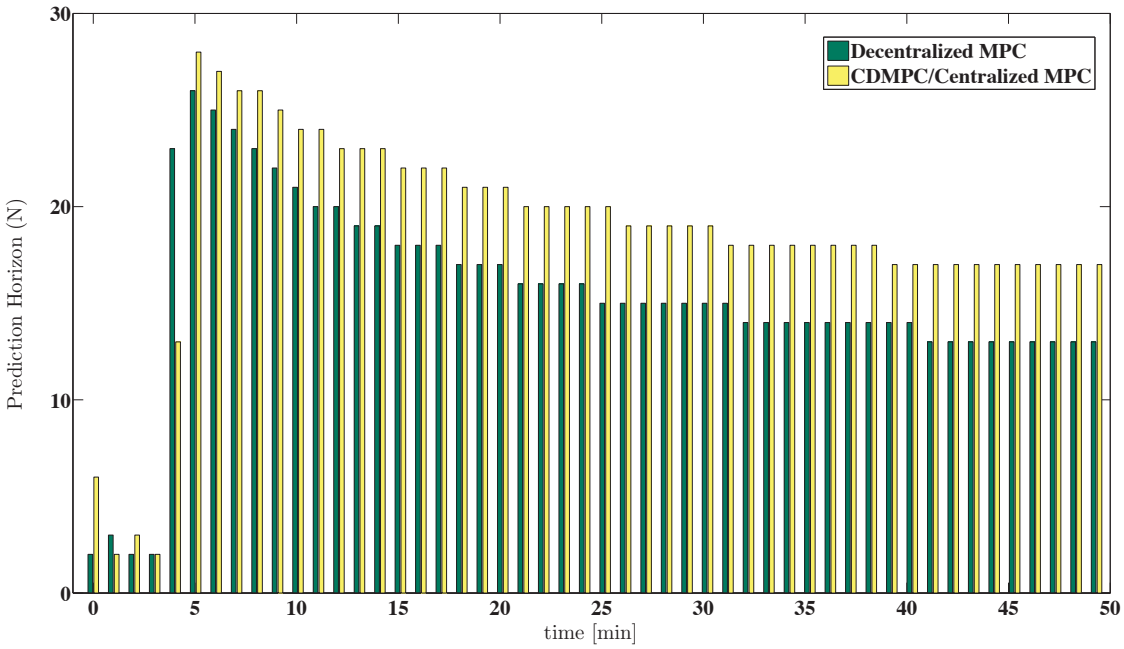


Figure 4.4: Comparison of number predictions horizons needed (N) between the Decentralized MPC, the Centralized MPC and the CDMPC trajectories vs simulation time

In addition, state trajectories and manipulated input variables of the evaporation process are depicted in Figure 4.5. As shown, the CDMPC and the centralized MPC schemes settle to the set-points with a higher load of performance compared to the decentralized trajectory.

Correspondingly, the overall objective functions of the three distributed control schemes J_{DC}, J_C, and J_D are compared in Figure 4.6. Within this window, the overall objective function obtaind by the centralized MPC/CDMPC is lower than the one obtained from decentralized scheme. These show that the CDMPC improves the decentralized MPC network with minor modifications applied to the local controllers.

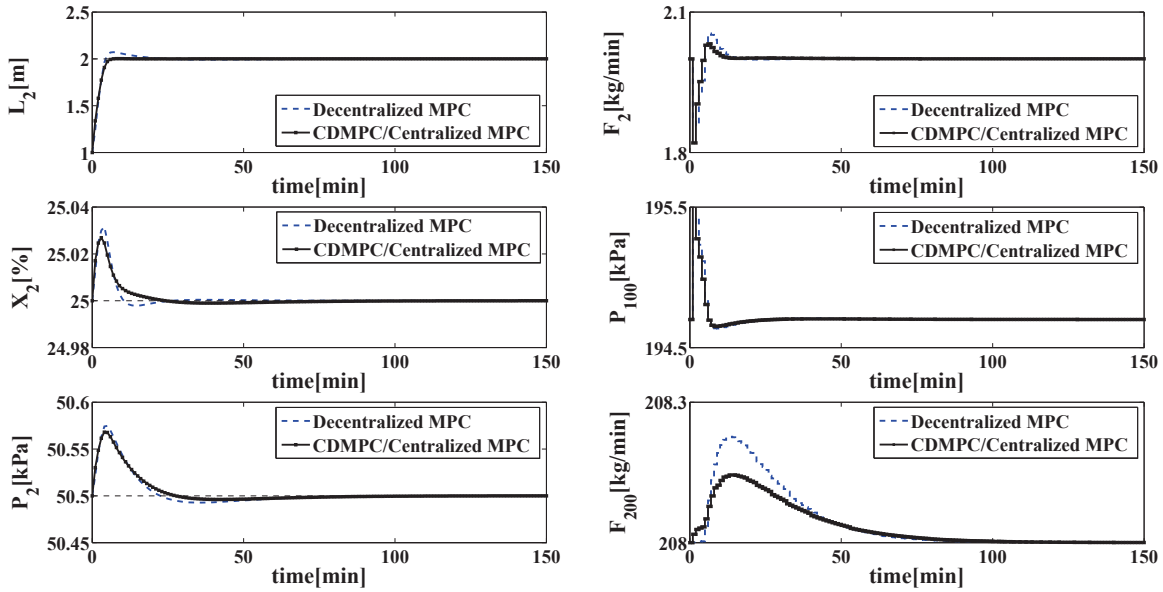


Figure 4.5: Comparison of state variables trajectories and manipulated variables trajectories between the Decentralized MPC, the Centralized MPC and the CDMPC trajectories vs simulation time

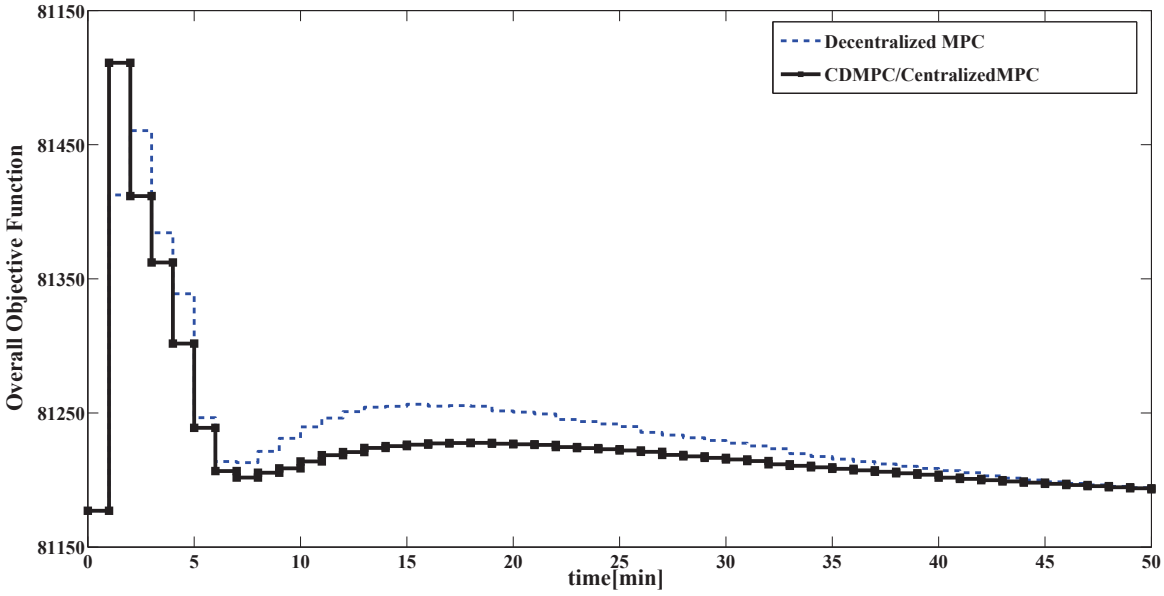


Figure 4.6: Comparison of overall objective functions between the Decentralized MPC, the Centralized MPC and the CDMPC trajectories vs simulation time

4.6.2 Two-CSTR Process

In this section, the feasible direction scheme is implemented on a two reactive continuous stirred tank reactor (CSTR) benchmark, taken from [96]. The system is comprised of two non-isothermal reactive CSTRs with interconnections and a recycle stream as depicted in Figure 4.7. The following set of exothermic reactions take place in these reactors with substances A and B: (i) A $\xrightarrow{k_1} B$, (ii) A $\xrightarrow{k_2} UP$, and (iii) A $\xrightarrow{k_3} DP$; where UP and DP stand for the undesired product and the desired product, respectively. As in Figure 4.7, CSTR I has two feed streams (one has fresh stream of substance 'A' with molar concentration C_{A0}, flow-rate F_0 and temperature T_0, and the other streams is from the output of CSRT II containing the recycle stream of unreacted substance A at flow-rate F_r, molar concentration C_{A2} and temperature T_2) and CSTR II is fed from the output of CSTR I as well as another fresh stream of substance A at flow-rate F_3, molar concentration CA_{03}, and temperature T_{03}. The two CSTRs are equipped with jackets to remove/provide heat, due to non-isothermal nature of reactions.

The main focus of this simulation is on stabilize the system around the open-loop unstable steady-state operating point, to avoid high temperatures, while simultaneously achieving reasonable conversion. The manipulated variables of the system are the heat input rates Q_1 and Q_2, and the inlet concentrations C_{A0} and C_{A03}.

The continuous-time plant model of this process can be derived based on linearizion around its unstable steady-state operating condition, i.e.

$$(T_1^s, C_{A1}^s, T_2^s, C_{A2}^s) = (457.9[K], 1.77[kmol/m^3], 415.5[K], 1.75[kmol/m^3]) \tag{4.90a}$$

$$(Q_1^s, C_{A0}^s, Q_2^s, C_{A03}^s) = (0[kJ/hr], 4[kmol/m^3], 0[kJ/hr], 2[kmol/m^3]) \tag{4.90b}$$

as the following:

$$\dot{x} = \begin{bmatrix} 25.2 & 1284.3 & 35 & 0 \\ -0.3 & -45.9 & 0 & 35 \\ 13.3 & 0 & -2.8 & 336.2 \\ 0 & 13.3 & -0.1 & -24.9 \end{bmatrix} x + \begin{bmatrix} 0.0043 & 0 & 0 & 0 \\ 0 & 4.998 & 0 & 0 \\ 0 & 0 & 0.0014 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} u \tag{4.91}$$

The ODE system (4.91) can now be decomposed into two subsystems: sub-system I (containing CSTR I) and subsystem II (containing CSTR II), this is depicted in Figure 4.7 . The two sub-systems are defined in the overall process model as:

$$x^T = ([x_{1,1}, x_{1,2}]||[x_{2,1}, x_{2,2}])^T = ([T_1, C_{A1}]||[T_2, C_{A2}])^T \tag{4.92}$$

$$u^T = ([u_{1,1}, u_{1,2}]||[u_{2,1}, u_{2,2}])^T = ([Q_1, C_{A0}]||[Q_2, C_{A03}])^T \tag{4.93}$$

and the corresponding discretized-time model, for a sampling time of $T_s = 0.005$ $[s]$, can be derived as:

$$\hat{x}(k+1) = \left[\begin{array}{cc|cc} 1.1357 & 6.1467 & 0.1852 & 0.6696 \\ 0.0014 & 0.7954 & 0.0002 & 0.1466 \\ \hline 0.0704 & 0.2556 & 0.9917 & 1.5870 \\ 0.0001 & 0.0557 & 0.0005 & 0.8875 \end{array}\right] \hat{x}(k) + \left[\begin{array}{cc|cc} 0.0000 & 0.0778 & 0.0000 & 0.0113 \\ 0.0000 & 0.0223 & 0.0000 & 0.0039 \\ \hline 0.0000 & 0.0022 & 0.0000 & 0.0404 \\ 0.0000 & 0.0007 & 0.0000 & 0.0471 \end{array}\right] \hat{u}(k) \tag{4.94}$$

The CDMPC trajectory is calculated according to the approach discussed in Section

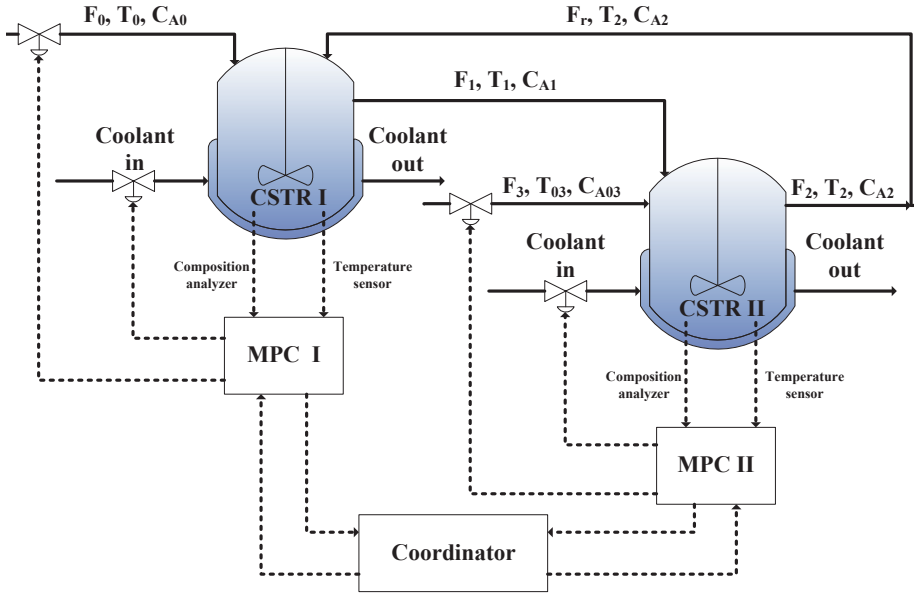


Figure 4.7: Schematic of the Two CSTR Case Study

	MPC I	MPC II
Initial Conditions:	$x_1(0) = [462.9[K], 0.27[kmol/m^3]]$	$x_2(0) = [410.5[K], 3.45[kmol/m^3]]$
Weighting Matrices:	$Q_1 = 5I,\ R_1 = 15I$	$Q_2 = 5I,\ R_2 = 15I$
Upper bounds:	$lb_{u1} = [5[kJ/hr], 8[kmol/m^3]]$	$lb_{u2} = [5[kJ/hr], 4[kmol/m^3]]$
Lower Bounds:	$ub_{u1} = [-5[kJ/hr], 0[kmol/m^3]]$	$ub_{u2} = [-5[kJ/hr], 0[kmol/m^3]]$
Sampling Period	$0.005[s]$	$0.005[s]$

Table 4.1: The Two-CSTR variable bounds and controller parameters

4.3, based on the adaptive horizon scheme explained in Section 4.5.1. The summary of controller parameters and the variable bounds of the these two sub-systems are listed in Table 4.1.

The comparison between the required length of predictions horizons (N) between the decentralized MPC, the centralized MPC and the CDMPC trajectories is depicted in Figure 4.8. This comparison shows that, after a certain amount of simulation, time the centralized MPC and CDMPC schemes roughly require $N = 6$, while the minimum prediction horizon required by Decentralized scheme oscillates between $N = 2$ and $N = 12$. As expected from the CDMPC algorithm, this networked control system is acting very

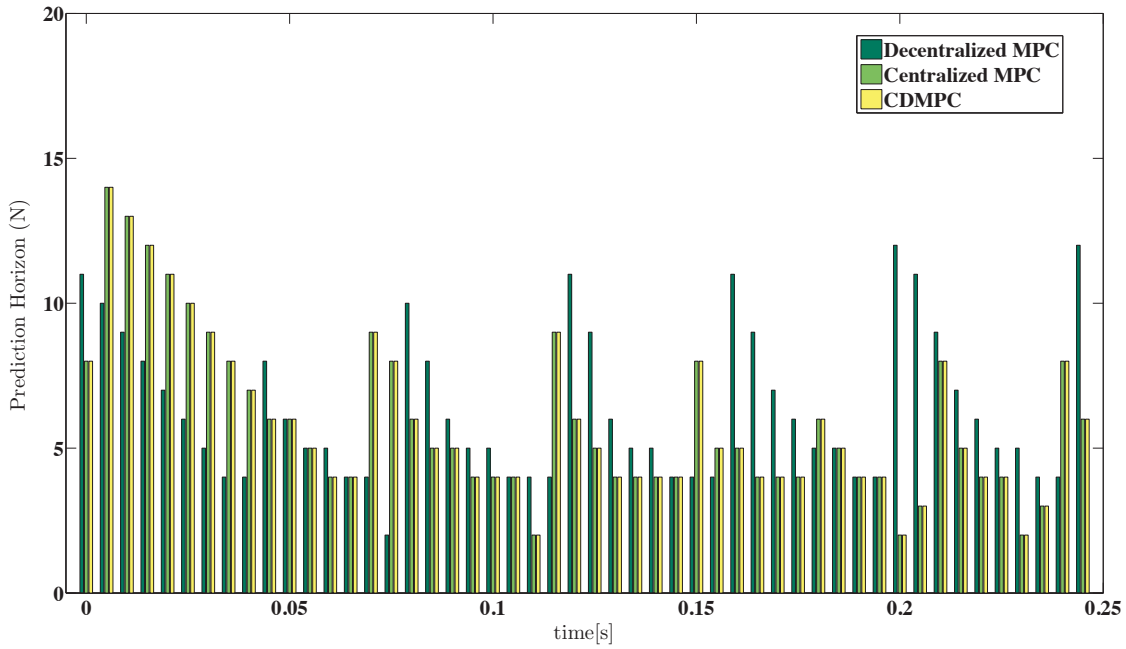


Figure 4.8: Comparison of number predictions horizons needed (N) between the Decentralized MPC, the Centralized MPC and the CDMPC trajectories vs simulation time

similar to an equivalent monolithic centralized MPC in terms of their stability criteria.

The state trajectories and manipulated input variables of the Two CSTR system are depicted in Figures 4.9 and 4.10, respectively. As shown, the CDMPC trajectory follows the centralized MPC trend perfectly while the decentralized MPC shows a lower performance in terms of the time required to settle on the unstable operating condition. Therefore, issues due to convergence of iterative approach reported by [76] for the open-loop unstable case study is resolved using Algorithm 25.

In addition, the overall objective functions of the three distributed control schemes J_{DC}, J_C, and J_D are compared in Figure 4.11. The mean square error between the objective function of the CDMPC and the objective function of the centralized MPC is at least twelve orders of magnitude smaller than that of the decentralized NMPC. These results show that, although the decentralized MPC scheme requires longer prediction

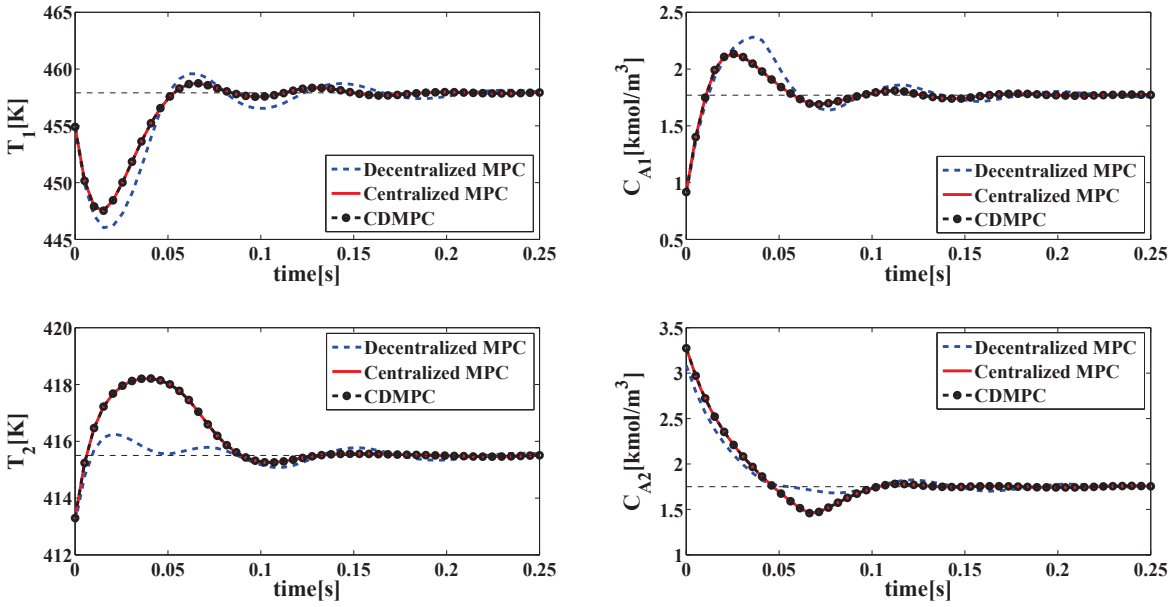


Figure 4.9: Comparison of state variables trajectories and manipulated variables trajectories between the Decentralized MPC, the Centralized MPC and the CDMPC trajectories vs simulation time

horizon in majority of sampling times, the CDMPC scheme achieves the centralized MPC performance with the lowest possible overall objective function. In other words, it improves the decentralized MPC overall performance by applying minor modifications to the currently installed decentralized MPC network.

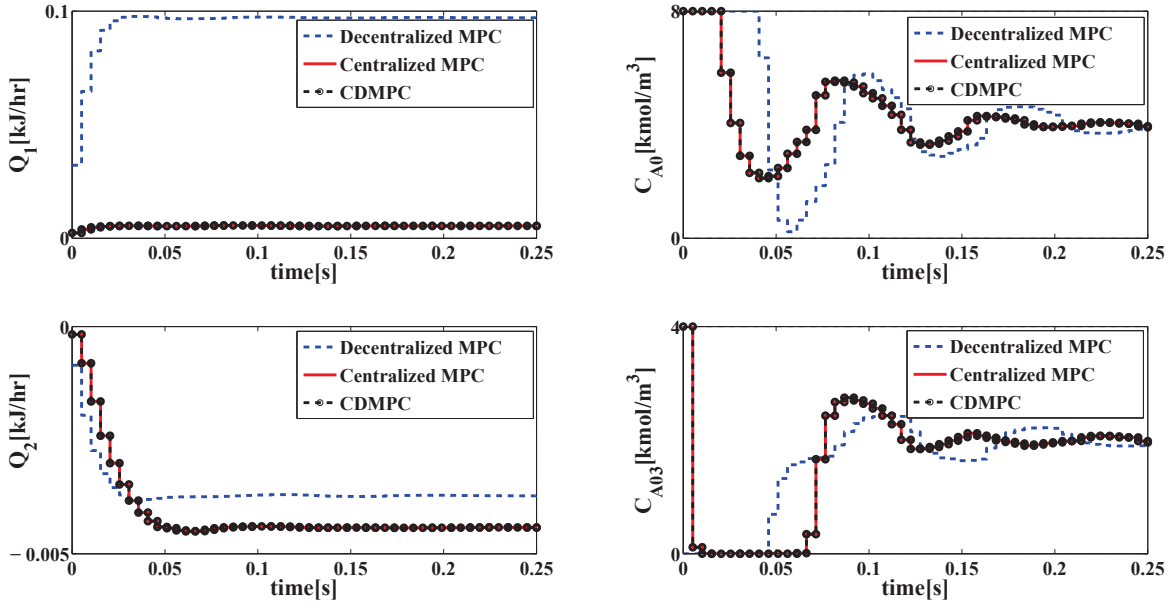


Figure 4.10: Comparison of manipulated variables trajectories between the Decentralized MPC, the Centralized MPC and the CDMPC trajectories vs simulation time

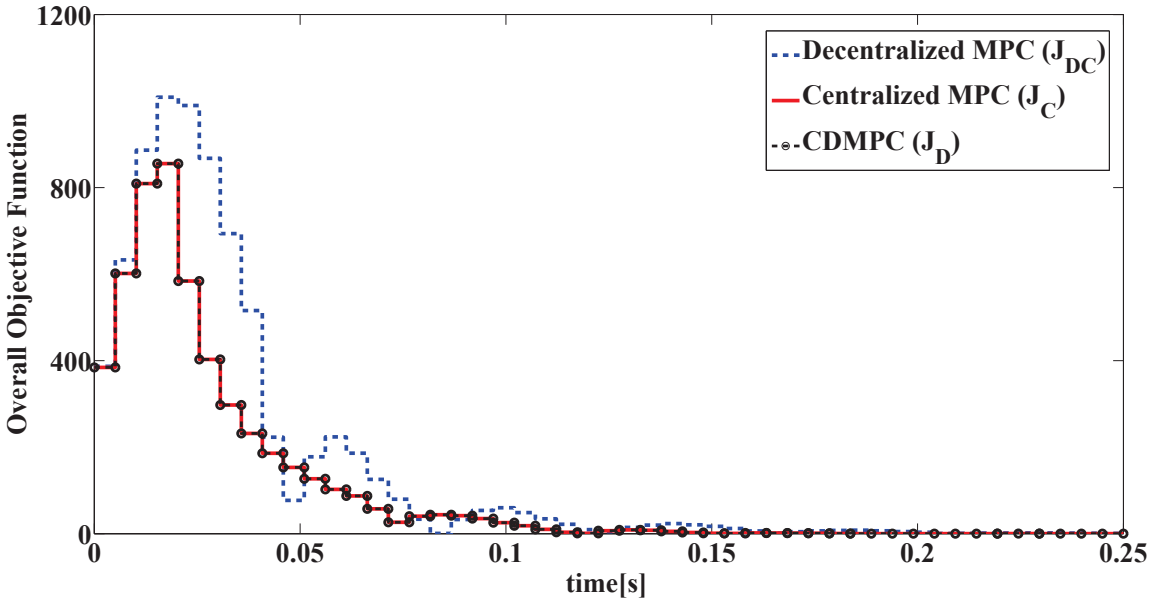


Figure 4.11: Comparison of overall objective functions between the Decentralized MPC, the Centralized MPC and the CDMPC trajectories vs simulation time

4.7 Conclusions

Two novel solution algorithms to coordinate distributed MPC controllers, using price-driven strategy, were proposed. The analytic CDMPC scheme presented in this chapter is capable of coordinating MPC networks subject to equality constraints and reduces the coordination cycles of conventional methods to one. The feasible descent direction approach CDMPC proposed in this chapter is guaranteed to be globally convergent to a unique local optimal plan-wide solution for the general case when inequality constraints are also present in the network. This scheme together with the adaptive horizon scheme is capable of stabilizing open-loop unstable systems under certain conditions, thereby resolves the convergence problems reported by conventional Newton-based approaches [76]. Results are promising and the coordinated system can perfectly track the centralized optimum trajectory, thus the performance of the decentralized system is improved with minor modifications applied to the network.

Chapter 5

Conclusions and Future work

This thesis is focused on the development and implementation of price-driven CDMPC algorithms using bilevel optimization methods. The primary goal was to design a systematic approach to coordinate decentralized networks of linear and nonlinear MPC systems for improving the performance of the existing decentralized installation. In essence, the coordinator is synthesized to establish a plant-wide decision making process to compensate for interaction model violations inside local prediction models.

To date, various CDMPC algorithms [20, 19, 21, 67, 66, 76, 57] have been devoted to linearly constrained dynamical systems. Such methods employ iterative Newton methods and can exhibit poor convergence properties. In addition, the existing body of CDNMPC research [89, 76, 68] has not properly addressed global convergence and closed-loop stability guarantees for the coordinated network. This work focuses on overcoming convergence issues with CDMPC structures and ensuring stability of the coordinated network. Also, two on-line strategies were proposed to coordinate nonlinear decentralized MPC systems. This thesis provided an approach for comparison of the performance of price-driven coordinated schemes with respect to decentralized control systems.

In chapter 2, a price-driven coordinated distributed MPC scheme was proposed for nonlinear systems. This interior-point approach extended ideas discussed in [20, 19, 21, 67, 66, 76, 57] and eliminates the need to identify the correct set of active constraints

at the end of each coordination cycle. Assuming that a centralized MPC based on the successive linearization can be designed to stabilize the entire system, sufficient conditions were derived for the CDMPC network to ensures the closed-loop stability. Another highlight of this chapter was the investigation of sufficient conditions to ensure stability of the system when the algorithm is stopped prior to convergence to the desired optimal solution. This scheme was limited to nonlinear systems that are open-loop stable, and is further limited by the successive linearization method.

In chapter 3, an on-line price-driven coordination distributed NMPC scheme was proposed for continuous-time nonlinear DAE systems, which extended the ideas introduced in [89, 68, 76]. A novel structure was proposed via bi-level nonlinear optimization. In this structure the coordinator constructs the upper level problem and the distributed NMPC controllers belong to the lower level problem. Under certain preconditions, the CDNMPC problem was relaxed into a series of quadratic programming problems subject to a predefined trust-region radius. Sufficient conditions for global convergence of the algorithm and stability guarantees were derived. The scheme can be applied both to open-loop stable and unstable dynamics. The algorithm ensures that the overall interaction error remains sufficiently small and minimum required length of prediction horizon is selected

In chapter 4, two novel solution algorithms to coordinate distributed MPC controllers were proposed, based on the price-driven concept. Similar to Chapter 3, bi-level optimization techniques were deployed to solve the price-driven CDMPC problem. The first scheme was an analytic approach to find closed-form solutions of plant-wide problems, where all the constraints inside the structure of local controllers remain active at all times. The proposed analytic CDMPC was able to reduce the computational load on the coordination systems and derive the optimal plant-wide solution with the minimum number of coordination cycles. The second algorithm was an iterative approach to finding a

local optimal solution to the general CDMPC problem via method of feasible directions. This algorithm offered a globally convergent method for the general case when inequality constraints were present in the network. These schemes, together with an adaptive horizon algorithms to ensure stability of the network, attempt to render the centralized optimal solution. The scheme can be applied both to open-loop stable and unstable dynamics. Similar to Chapter 3, the dynamic optimization approach enabled the controlled network to stabilize unstable open-loop dynamical systems without any modifications to the native CDMPC formulation.

5.1 Recommendations for future research

This thesis provided a fundamental insight into the design and implementation of hierarchical control architectures to coordinate network of distributed MPC and NMPC systems. This section explores open research possibilities in the area of coordinated distributed model predictive control.

In this study, the interactions between subsystems were defined in the form of equality constraints. A more detailed study can be carried out to include inequality-based interrelations. Such inequality constraints may be interpreted as shared resources and/or internal dependencies between neighboring subsystems. The procedure can be performed by assigning an additional price vector for local violations of equality inequality binding constraints. The resulting bi-level optimization problem would impose restrictions on the price vector assigned to inequality interactions into the upper-level optimization problem. In other words, the coordination level must fulfill characteristics such as obtaining non-negative values for the price vector assigned to the inequalities.

One approach is to develop an on-line hybrid coordination scheme based on the methods proposed in this thesis and the work of [68] to satisfy both types of aforementioned

interaction models. Nonetheless, depending on the type of plant-wide inequality dependencies, global convergence and closed-loop stability of proposed methods should be considered. One may also consider developing a globally convergent bi-level optimization method to satisfy both the upper-level and lower-level constraints in order to seek a unique plant-wide optimal solution.

The control architectures introduced in this thesis were designed based on a particular sampling rate. A broader scope may consider processes with different time scales and controllers with multi-rate structures. In this regard, one may choose to extend the idea introduced [66] to design a multi-rate coordination scheme for nonlinear systems. In addition, throughout this work, the local information from subsystems was assumed to be available at any rate as required by the coordinator. A more realistic approach would be to design proper soft-sensing systems inside the coordinated structure to estimate the required information. Nevertheless, incorporation of single-rate/multi-rate estimation systems into the coordinated network would introduce new challenges such as maintaining the closed-loop stability of the overall system. Accordingly, proper observability analysis should be considered to ensure reliability and stability of the closed-loop system. Multi-rate Kalman filtering procedures can be embedded into coordination systems to estimate the unknown information based on available measurements.

Throughout this work, a number of CDMPC and CDNMPC algorithms were proposed. A thorough study can be carried out to investigate computational properties of these methods. For the CDMPC scheme introduced in Chapter 4, random abstractive networks of distributed MPCs can be generated that differ in size and interaction complexity. Thus, the ideas proposed by [19] and [57] can be extended to distributed MPC systems with internal inequality constraints. Finally, it is beneficial to implement such coordination algorithms for pilot sized industrial processing units. Proper communication routines and protocols need to be designed under the distributed control system, prior

to investigating advantages and disadvantages of CDMPC schemes in industrial applications. Four tank systems and the Two CSTR problem can be regarded as candidate case studies to embark on such investigations. It is worth mentioning that, simulation environments such as DeltaV Simulate® or Aspen PLUS Dynamics® are good starting points to perform such analyses before any hardware related testing.

References

[1] E. Aiyoshi and K. Shimizu. Hierarchical decentralized systems and its new solution by a barrier method. *IEEE Trans. Syst. Man. Cybern.*, SMC-11(6):444–449, 1981.

[2] E. Aiyoshi and K. Shimizu. A solution method for the static constrained Stackelberg problem via penalty method. *IEEE Trans. Automat. Contr.*, AC-29(12):1111–1114, 1984.

[3] W. Al-Gherwi, H. Budman, and A. Elkamel. A robust distributed model predictive control algorithm. *J. Process Control*, 21:1127–1137, 2011.

[4] F. A. Al-Khayyal, R. Horst, and P. M. Pardalos. Global optimization of concave functions subject to quadratic constraints: An application in nonlinear bilevel programming. *Ann. Oper. Res.*, 34:125–147, 1992.

[5] M. A. Amouzegar. A global optimization method for nonlinear bilevel programming problems. *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, 29(6):771–7, January 1999.

[6] W. Andreas and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1):25–57, 2006.

[7] A. Antoniou and W. Lu. *Practical Optimization: Algorithms and Engineering Applications*. Springer Science + BusinessMedia, LLC., 2007.

[8] L. Bakule. Decentralized control: an overview. *Annu. Rev. Control*, 32:87–98, 2008.

[9] J. F. Bard. *Practical Bilevel Optimization : Algorithms and Applications*. Kluwer Academic Publishers, 1998.

[10] A. D. Belegundu, L. Berke, and S. N. Patnaik. An optimization algorithm based on the method of feasible directions. *Struct. Optim.*, 9(2):83–88, April 1995.

[11] J. Bosco and E. Etoa. A filter method to solve nonlinear bilevel programming problems. In *Inf. Comput. Appl.*, volume 6377, pages 395–406. Springer, 2010.

[12] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[13] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz. On the convergence of successive linear-quadratic programming algorithms. *SIAM J. Optim.*, 16(2):471–489, 2005.

[14] R. H. Byrd, J. Nocedal, and R. A Waltz. Steering exact penalty methods for nonlinear programming. Technical report, Optimization Technology Center Northwestern University, Evanston, 2007.

[15] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Syst. Mag.*, 22:44–52, 2002.

[16] L. M. Case. *An l1 penalty function approach to the nonlinear bilevel programming problem*. PhD Dissertation, University of Waterloo, 1997.

[17] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[18] Y. Chen and M. Florian. The nonlinear bilevel programming problem:formulations,regularity and optimality conditions. *Optimization*, 32(3):193–209, January 1995.

[19] R. Cheng. *Decomposition and coordination of large-scale operations optimization*. PhD Dissertation, University of Alberta, 2007.

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

[21] R. Cheng, J. F. Forbes, and W. S. Yip. Dantzig Wolfe decomposition and plant-wide MPC coordination. *Comput. Chem. Eng.*, 32(7):1507–1522, July 2008.

[22] P. D. Christofides, J. Liu, and D. M. de la Peña. *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications*. Advances in Industrial Control Series. Springer-Verlag, London, England, 2011.

[23] P. D. Christofides, R. Scattolini, D. M. de la Peña, and J. Liu. Distributed model predictive control: a tutorial review and future research directions. *Comput. Chem. Eng.*, 51:21–41, 2013.

[24] P. A. Clark and A. W. Westerberg. Bilevel programming for steady-state chemical process designI. Fundamentals and algorithms. *Comput. Chem. Eng.*, 14(1):87–97, 1990.

[25] G. Cohen. Optimization by decomposition and coordination: a unified approach. *IEEE Trans. Automat. Contr.*, 23(2):222–232, April 1978.

[26] B. Colson, P. Marcotte, and G. Savard. A trust-region method for nonlinear bilevel programming: algorithm and computational. *Comput. Optim. Appl.*, 30:211–227, 2005.

[27] B. Colson, P. Marcotte, and G. Savard. Bilevel programming: a survey. *4OR A Q. J. Oper. Res.*, 3(2):87–107, June 2005.

[28] J. L. Crassidis and J. L. Junkins. *Optimal Estimation of Dynamic Systems*. CRC Press, first edition, 2004.

[29] S. Dempe. Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization*, 52(3):333–359, June 2003.

[30] Q. T. Dinh, C. Savorgnan, and M. Diehl. Real-time sequential convex programming for nonlinear model predictive control and application to a hydro-power plant. In *50th IEEE Conf. Decis. Control Eur. Control Conf.*, pages 5905–5910, 2011.

[31] W. B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. Automat. Contr.*, 52:1249–1263, 2007.

[32] T. A. Edmunds and J. F. Bard. Algorithms for nonlinear bilevel mathematical programs. *IEEE Trans. Automat. Contr.*, 21(1), 1991.

[33] F. Facchinei, H. Jiang, and L. Qi. A smoothing method for mathematical programs with equilibrium constraints. *Math. Program.*, 85:107–134, 1999.

[34] J. E. Falk and J. Liu. On bilevel programming, part I : general. *Math. Program.*, 70:47–72, 1995.

[35] N. J. Feb, G. B. Dantzig, and P. Wolfe. Decomposition principle for linear programs: decomposition principle for linear programs. In *Rand Symp. Math. Program.*, volume 8, pages 101–111, 1959.

[36] R. Findeisen, F. N. Bailey, M. A. Brdys, K. Malinowski, P. Tatjewski, and A. Wozniak. *Control and coordination in hierarchical systems*. John Wiley & Sons, 1980.

[37] A. Fischer and H. Jiang. Merit functions for complementarity and related problems: a survey. *Comput. Optim. Appl.*, 17:159–182, 2001.

[38] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, New York, second edition, 1987.

[39] J. Gauvin and R. Janin. Directional behaviour of optimal solutions in nonlinear mathematical programming. *Math. Oper. Res.*, 13(4):629–649, 1988.

[40] P. E. Gili and H. W. Ma. Interior methods for nonlinear optimization. *SIAM Rev.*, 2(4):525–597, 2002.

[41] L. Grüne and J. Pannek. Practical NMPC suboptimality estimates along trajectories. *Syst. Control Lett.*, 58(3):161–168, March 2009.

[42] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control: Theory and Algorithm*. Springer-Verlag London Limited, 2011.

[43] Z. H. Gümü and C. A. Floudas. A global optimization method for nonlinear bilevel programming problems. *J. Glob. Optim.*, 20:1–31, January 2001.

[44] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag London Limited, London, second edition, 2010.

[45] E. Hendricks, O. Jannerup, and P. H. Sorensen. *Linear Systems Control Deterministic and Stochastic Methods*. Springer Berlin Heidelberg, first edition, 2008.

[46] J. Herskovits, Mario T. Filho, and A. Leontiev. An interior point technique for solving bilevel programming problems. *Optim. Eng.*, 14(3):381–394, May 2012.

[47] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

[48] B. Houska, H. J. Ferreau, and M. Diehl. ACADO toolkit: an open-source framework for automatic control and dynamic optimization. *Optim. Control Appl. Methods*, 32(3):298–312, 2011.

[49] T. G. Hovgaard, S. Boyd, L. F. S. Larsen, and J. B. Jø rgensen. Nonconvex model predictive control for commercial refrigeration. *Int. J. Control*, 86(8):1349–1366, August 2013.

[50] D. Ibrahim, A. El, and A. H. F. Naser. Control of a forced-circulation evaporator system as a case study for control engineering education. In *IEEE 46th Midwest Symp. Circuits Syst.*, pages 1122–1129, 2003.

[51] Y. Ishizuka and E. Aiyoshi. Double penalty method for bilevel optimization problems. *Ann. Oper. Res.*, 34:73–88, 1992.

[52] Y. Jiang, X. Li, C. Huang, and X. Wu. An augmented Lagrangian multiplier method based on a CHKS smoothing function for solving nonlinear bilevel programming problems. *Knowledge-Based Syst.*, 55:9–14, January 2014.

[53] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, Inc., second edition, 1996.

[54] C. D. Kolstad and L. S. Lasdon. Derivative evaluation and computational experience with large bilevel mathematical programs. *J. Optim. Theory Appl.*, 65(3):485–499, 1990.

[55] J. Korycki and M. Kostreva. Norm-relaxed method of feasible directions: application in structural optimization. *Struct. Optim.*, 11(3-4):187–194, June 1996.

[56] M. M. Kostreva and X. Chen. A superlinearly convergent method of feasible directions. *Appl. Math. Comput.*, 116(3):231–244, December 2000.

[57] S. Li. *Computational properties of unconstrained linear distributed model predictive control*. MSc Thesis, University of Alberta, 2013.

[58] S. Li, Y. Zhang, and Q. Zhu. Nash-optimization enhanced distributed model predictive control applied to the Shell benchmark problem. *Inf. Sci.*, 170:329–349, 2005.

[59] G. Lin, M. Xu, and J. J. Ye. On solving simple bilevel programs with a nonconvex lower level program. *Math. Program.*, 144(1-2):277–305, January 2013.

[60] J. Liu, X. Chen, D. M. de la Peña, and P. D. Christofides. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE J.*, 56:2137–2149, 2010.

[61] J. Liu, D. M. de la Peña, and P. D. Christofides. Distributed model predictive control of nonlinear process systems. *AIChE J.*, 55:1171–1184, 2009.

[62] J. Löfberg. YALMIP: a toolbox for modeling and optimization in MATLAB. In *IEEE Int. Symp. Comput. Aided Control Syst. Des.*, pages 284–289, Taipei, Taiwan, 2004.

[63] J. Z. Lu. Challenging control problems and emerging technologies in enterprise optimization. *Control Eng. Pract.*, 11(8):847–858, August 2003.

[64] J. M. Maestre, D. M. de la Peña, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optim. Control Appl. Methods*, 32:153–176, 2011.

[65] N. Maratos. *New exact penalty functions for nonlinear constrained optimization problems*. PhD Dissertation, University of London, 1978.

[66] N. I. Marcos. *Coordinated-distributed optimal control of large-scale linear dynamic systems*. PhD Dissertation, University of Alberta, 2011.

[67] N. I. Marcos, J. F. Forbes, and M. Guay. Coordination of distributed model predictive controllers for constrained dynamic processes. In *Proc. IFAC ADCHEM*, Istanbul, Turkey, 2009.

[68] R. Martí, D. Sarabia, D. Navia, and C. de Prada. A method to coordinate decentralized NMPC controllers in oxygen distribution networks. *Comput. Chem. Eng.*, 59:122–137, December 2013.

[69] Mehmet Mercangöz and Francis J. Doyle. Distributed model predictive control of an experimental four-tank system. *J. Process Control*, 17(3):297–308, March 2007.

[70] A. G. Mersha and S. Dempe. Feasible direction method for bilevel programming problem. *Optimization*, 61(5):597–616, May 2012.

[71] M. D. Mesarovic, D. Macko, and Y. Takahara. *Theory of Hierarchical, Multilevel, Systems*. Academic Press, New York and London, 1970.

[72] M. D. Mesarovic, D. Macko, and Y. Takahara. Two coordination principles and their application in large scale systems control. *Automatica*, 6:261–270, 1970.

[73] R. R. Meyer. Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *J. Comput. Syst. Sci.*, 12:108–121, 1976.

[74] A. Mitsos, B. Chachuat, and P. I. Barton. Towards global bilevel dynamic optimization. *J. Glob. Optim.*, 45(1):63–93, January 2009.

[75] A. Mitsos, P. Lemonidis, and P. I. Barton. Global solution of bilevel programs with a nonconvex inner program. *J. Glob. Optim.*, 42(4):475–513, December 2007.

[76] P. G. Mohseni. *Coordination techniques for distributed model predictive control*. PhD Dissertation, University of Alberta, 2013.

[77] M. Morari and J. H. Lee. Model predictive control: past, present and future. *Comput. Chem. Eng.*, 23(4-5):667–682, May 1999.

[78] D. Nesic and A. R. Teel. A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. *IEEE Trans. Automat. Contr.*, 49(7):1103–1122, 2004.

[79] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Science + BusinessMedia, LLC., second edition, 2006.

[80] J. Pannek. *Receding horizon control: a suboptimality-based approach*. PhD Dissertation, University of Bayreuth, 2009.

[81] J. D. Pearson. Dynamic decomposition techniques. In *Optim. Methods Large-Scale Syst. with Appl.* McGraw-Hill, 1971.

[82] R. Quirynen, M. Vukov, and M. Diehl. Auto generation of implicit integrators for embedded NMPC with microsecond sampling. In *4th IFAC Nonlinear Model Predict. Control Conf.*, pages 175–180, 2012.

[83] T. D. Quoc, C. Savorgnan, and M. Diehl. Real-Time Sequential Convex Programming for Optimal Control Applications. In *Model. Simul. Optim. Complex Process.*, pages 91–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[84] J. B. Rawlings and B.T. Stewart. Coordinating multiple optimization-based controllers: new opportunities and challenges. *J. Process Control*, 18:839–845, 2008.

[85] J. L. Salle and S. Lefschietz. *Stability by Liapunov's Direct Method With Applications*. Academic Press, 1967.

[86] N. R. Sandell, P. Varajya, M. Athans, and M. C. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Trans. Automat. Contr.*, 23:108–128, 1978.

[87] G. Savard and J. Gauvin. The steepest descent direction for the nonlinear bilevel programming problem. *Oper. Res. Lett.*, 15:265–272, 1994.

[88] R. Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *J. Process Control*, 19:723–731, 2009.

[89] H. Scheu, J. Busch, and W. Marquardt. Nonlinear distributed dynamic optimization based on first order sensitivities. In *2010 Am. Control Conf.*, pages 1574–1579, 2010.

[90] H. Scheu and W. Marquardt. Sensitivity-based coordination in distributed model predictive control. *J. Process Control*, 21(5):715–728, June 2011.

[91] R. Sharma, D. Nesic, and C. Manzie. Idle speed control using linear time varying model predictive control and discrete time approximations. In *IEEE Int. Conf. Control Appl.*, pages 1140–1145, Yokohama, Japan, 2010.

[92] K. Shimizu and E. Aiyoshi. A new computational method for Stackelberg and min-max problems by use of a penalty method. *IEEE Trans. Automat. Contr.*, AC-26(2):460–466, 1981.

[93] B. T. Stewart. *Plantwide cooperative distributed model predictive control*. PhD Dissertation, University of Wisconsin-Madison, 2010.

[94] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Coorperative distributed model predictive control. *Syst. Control Lett.*, 59:460–469, 2010.

[95] B. T. Stewart, S. J. Wright, and J. B. Rawlings. Cooperative Distributed Model Predictive Control for Nonlinear Systems. *J. Process Control*, 21:698–704, 2011.

[96] Y. Sun and N. H. El-Farra. Quasi-decentralized model-based networked control of process systems. *Comput. Chem. Eng.*, 32:2016–2029, 2008.

[97] N. V. Thoai, Y. Yamamoto, and A. Yoshise. Global optimization method for solving mathematical programs with linear complementarity constraints. *J. Optim. Theory Appl.*, 124(2):467–490, 2005.

[98] M. J. Tippett and J. Bao. Distributed model predictive control based on dissipativity. *AIChE J.*, 59(3):787–804, 2013.

[99] D. M. Topkis and A. F. Veinott. On the convergence of some feasible direction algorithms for nonlinear programming. *SIAM J. Control*, 5(2):268–279, 1967.

[100] A. N. Venkat. *Distributed model predictive control : theory and applications*. PhD Dissertation, University of Wisconsin-Madison, 2006.

[101] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed model predictive control. In *Proc. 44th IEEE Conf. Decis. Control Eur. Control Conf. ECC 2005*, pages 6680–6685, Seville, Spain, 2005.

[102] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Implementable distributed model predictive control with guaranteed performance properties. *2006 Am. Control Conf.*, pages 613–618, 2006.