GRAPH REPRESENTATION AND DISTRIBUTED CONTROL OF LUMPED AND DISTRIBUTED PARAMETER SYSTEM NETWORKS

A DISSERTATION

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

By

MANJIRI ARUN MOHARIR

in partial fulfillment for the requirements of the degree of

DOCTOR OF PHILOSOPHY

in Chemical Engineering

Advised by Prof. Prodromos Daoutidis

May 2019

Acknowledgments

I begin by expressing my utmost gratitude to my late father, Prof. Arun Moharir, Ph.D. Chemical Engineering, for the immeasurable efforts and sacrifices he made for me.

My advisor, Prof. Prodromos Daoutidis, has been far more than an academic advisor for these past five years. I am sincerely grateful to him for all of his time, efforts, advice, guidance, support and kindness, and feel honored to have been his student.

My family - my parents, Arun and Sarita Moharir, my sister, Sharayu Moharir and my brother-in-law, Ateet Parikh, and Sravya Paluri - have been a source of unconditional, boundless support and understanding. I am grateful to them for all that they have done for me.

I also thank my lab-members Dr. Seongmin Heo, Dr. Adam Kelloway, Dr. Abdulla Malek, Dr. Nahla Al Amoodi, Dr. Michael Zachar, Dr. Nitish Mittal, Dr. Udit Gupta, Dr. Andrew Allman, Matthew Palys, Wentao Tang, Shaaz Khatib, Ilias Mitrai, and Hanchu Wang for creating a supportive and lively working space, and my friends Bharat Menon, Inacio Fernandes, Aditya Bikram Bhandari, Meera Shete, Kristeen Joseph, Sajna Hameed, Sagar Udyavara, Sushant Garud, Ankita Naik, Eeshani Godbole, Anurag Kumar, Saurabh Modi, Vaidyanathan Sethuraman, Arjun Bindu Jyothikumar, and Shubhasree Kal for their support and understanding.

To My Father

Abstract

Chemical plants are complex, integrated networks of individual process systems. The process system dynamics along with the interconnections among them make the task of controlling chemical plants challenging. Distributed control is a promising approach towards achieving plant-wide control of tightly integrated networks. The identification of sparsely interacting sub-networks in a given chemical network is key towards achieving superior performance from the distributed control structure. To this end, community detection algorithms have been adopted to determine the optimal decompositions for chemical networks by maximization of modularity. These algorithms are based on equation graph representations of the network. For lumped parameter system (LPS) networks, such representations are standard. Since chemical networks usually comprise lumped as well as distributed parameter systems (DPSs), this thesis aims at incorporating within the framework described above, the variables and topology of DPSs, to develop a unified framework to obtain optimal network decompositions (control structures) for distributed control. To this end, an equation graph representation for a generic DPS and a parameter which captures the strength of structural interactions among its variables analogous to relative degree in LPSs are proposed. A relationship is established between the length of the input-output path in the equation graph and the structural interaction parameter, which enables the incorporation of DPSs variables in the graph based community detection algorithms.

Also, since in chemical networks, often the measurement of the entire state is not

available and estimation of the unmeasured variables is a computationally expensive task, this thesis also addresses the problem of combined distributed state estimation and distributed control, using community detection for determining network decompositions for estimation as well as control.

Contents

List of Figures viii

List of Tables xi

1 Introduction 1

2 Structural Interaction Based Equation Graphs of Convection - Reaction Systems 7

2.1 Introduction 7

2.2 Generic Convection-Reaction Systems 10

2.3 Structural Interaction Parameters for Hyperbolic PDEs 12

2.4 Equation Graphs of a Network of Hyperbolic PDE Systems 16

2.5 Illustrative Example 19

2.6 Conclusions 21

3 Distributed Control of an Amine Gas Sweetening Plant 22

3.1 Introduction 22

3.2 The Amine Gas Sweetening Plant 26

3.2.1 Plant Description 26

3.2.2 Plant Model 26

3.3 Model Predictive Control Architectures 33

3.3.1 Centralized MPC 34
3.3.2 Distributed MPC 35
3.4 Network Decomposition 37
3.4.1 Input/Output Clustering 39
3.4.2 Modularity Based Decomposition of Equation Graph 43
3.4.3 The Optimal Decomposition 44
3.5 Simulation Results and Discussion 48

4 Structural Interaction Based Equation Graphs of Diffusion - Convection - Reaction Systems 56

4.1 Introduction 56
4.2 Standard Form of Diffusion-Convection-Reaction Systems 57
4.3 Reduced Order Representation 58
4.4 Graph-based Network Decomposition 60
4.4.1 Classification of Variables 60
4.4.2 Graph Representation 63
4.4.3 Structural Interaction Parameter 64
4.5 Conclusions 67

5 Distributed Control of a Reaction-Separation Network 69

5.1 Introduction 69
5.2 Plant Model 72
5.2.1 Tubular Reactor Model 72
5.2.2 Flash Tank Model 74
5.2.3 Heat Exchanger Model 75
5.2.4 Model Simulation 75
5.3 Network Decomposition 76

5.4 Simulation Results and Discussion 78

6 Distributed Estimation and Nonlinear MPC using Community Detection 85

6.1 Introduction 85

6.2 Model predictive control and moving horizon estimation 87

6.2.1 Distributed estimation and control models 90

6.2.2 DMPC formulation for output regulation 92

6.2.3 DMHE formulation for state estimation 94

6.2.4 Decompositions for control and estimation 95

6.3 An integrated process case study: Benzene alkylation with ethylene . . . 96

6.3.1 Process description 97

6.3.2 Model description 98

6.3.3 Distributed estimation and control architecture 103

6.4 Simulation Results and Discussion 108

7 Summary and Conclusions 115

8 Future Work 121

8.1 Structurally Controllable Decompositions 121

8.1.1 LPS Network Decomposition 121

8.1.2 DPS Network Decomposition 122

8.2 Structurally Observable Decompositions 122

8.3 DPS Graphs based on Comprehensive Interaction 123

9 Bibliography 125

Appendices 141

List of Figures

2.1 Types of input - output combinations: (*a*) Velocity input - Boundary output, (*b*) Distributed input - Distributed output, (*c*) Boundary input - Boundary output 11

2.2 CSTR and PFR in series 19

2.3 Equation graph of CSTR-PFR in series 20

3.1 Amine gas sweetening plant 27

3.2 Discretization of the distributed variables 31

3.3 Centralized MPC 34

3.4 Iterative distributed MPC 36

3.5 Controllable decompositions 45

3.6 Optimal decomposition 47

3.7 Input Profiles 49

3.8 Output Profiles 49

3.9 Performance indices of the four control architectures for the three control problems considered 52

3.10 Average computation time per sampling period of the four different control architectures for the control problems, with the horizontal bar showing the sampling period 53

4.1 Equation graph of the tubular reactor 65

5.1 Prototypical reaction-separation plant 70
5.2 Communities in the reaction-separation plant 77
5.3 Dimensionless Performance Indices (DPI) of the CMPC and optimal decomposition - based DMPC for the set-point tracking problems 81
5.4 Computation time (seconds/sample) of the CMPC and optimal decomposition-based DMPC for the set-point tracking problems 82
5.5 Input profiles from DMPC implementation based on the optimal decomposition 82
5.6 Output profiles from DMPC implementation based on the optimal decomposition 83

6.1 An schematic of the combined estimation and control problem (MHE+MPC) 90
6.2 Iterative combined DMHE and DMPC structure for different system decompositions for the control problem (L subsystems) and the estimation problem (N subsystems) 95
6.3 Graph representations of a typical process system for (a) control and (b) estimation problems 96
6.4 Benzene alkylation with ethylene 97
6.5 The set of manipulated inputs for the benzene alkylation with ethylene . . 104
6.6 The set of measured outputs for the benzene alkylation with ethylene . . . 104
6.7 The graph representation for control problem 106
6.8 The graph representation for estimation problem 106
6.9 System decomposition for control 106
6.10 System decomposition for state estimation 107

6.11 Dynamic behavior of selected output variables during open-loop, $t \in [0\ 600)$, and closed-loop, $t \in [600\ 2400]$, process operation 110
6.12 Selected manipulated inputs during open-loop, $t \in [0\ 600)$, and closed-loop, $t \in [600\ 2400]$, process operation 111
6.13 Performance index for different control and estimation architectures . . . 112
6.14 Computation times for different control and estimation architectures . . . 113

List of Tables

3.1 Process parameter values 31

3.2 Boundary conditions of the model equations 32

3.3 List of inputs 38

3.4 List of outputs 38

3.5 Communities in the optimal decomposition 46

3.6 Communities in the sub-optimal decomposition 47

5.1 Manipulated variables 71

5.2 Controlled variables 71

5.3 Communities in the optimal decomposition 78

5.4 Model Parameters 79

5.5 Initial conditions of the input variables 80

5.6 Initial conditions of state variables in the flash tanks 80

5.7 Set-points of the controlled variables 80

5.8 Steady-state values of the manipulated variables 81

6.1 Process variables. 101

6.2 Process parameters 103

6.3 Reference (steady state) values of the state variables 105

6.4 Reference (steady state) values of the manipulated inputs 107

6.5 Reference (steady state) values of the state variables 108

Chapter 1

Introduction

Chemical plants are large networks of interconnected process systems. The plant-wide control problem of a generic chemical network is challenging owing to the large number of process systems with varied dynamics which makes a centralized control architecture impractical, and the interconnections between the systems which render a decentralized control architecture ineffective. Distributed control, which entails the implementation of local controllers to control sub-networks of a plant with some information sharing (communication), is a promising approach towards achieving plant-wide control, since it overcomes the limitations of the aforementioned approaches [1].

The control performance of distributed control is affected by the network decomposition, i.e., the choice of sub-networks for the local controllers [2]. Determining the optimal decomposition for a generic chemical network is an open and challenging problem [3]. Recently, our group has proposed two main methods of obtaining the optimal decomposition for the distributed control of a chemical plant, the first of which is hierarchical clustering of the input and output variables of the network [4–6], and the second is the application of community detection algorithms [7–9]. These approaches are based on the equation graph of the chemical network (possibly weighted to account for strength of in-

teractions), and determine the optimal decomposition as quantified by the maximization of modularity. An equation graph is a set of nodes and edges that captures the structure of a given network and modularity is a measure of the statistical significance of the interactions (i.e. the number of edges in the graph) among the variables (i.e. the nodes in the graph) within a community as compared to the variables across different communities.

Relative degree is a measure that captures the strength of structural interactions among the variables of lumped parameter systems (LPSs) and has been used as the basis for determining optimal decompositions [4–7, 10–14]. In the equation graphs for networks of LPSs, the length of the shortest path (i.e., the smallest number of edges from an input node to an output node), l, is related algebraically to the relative degree of the output variable with respect to the input variable, r, as [10]:

$$r = l - 1 \tag{1.1}$$

A limitation of the relative degree based network decomposition is that the equation graph representations and the concept of relative degree are restricted to variables of LPSs only. However, most chemical plants are networks of LPSs interconnected with distributed parameter systems (DPSs), the variables of which are functions of both space and time. This thesis aims at the extension of the equation graph representation and relative degree to the variables of DPSs, so as to facilitate the use the community detection algorithms for distributed control of generic chemical plants comprising LPSs and DPSs.

Another limitation of the aforementioned work on plant-wide distributed control is that the control problems considered are based on the assumption that the full state measurement is available for feed-back control. However, often in chemical networks, not all states can be measured in real time. This requires the use of a state estimator. The computational cost associated with state estimation can compromise the control perfor-

mance in the closed-loop process operation [15]. Hence, the estimation of the states with minimum computational cost is a challenging and relevant problem. Given the analogous nature of the challenges associated with state estimation and plant-wide control of large chemical plants, distributed estimation is a promising approach towards mitigating the computational cost in state estimation [16, 17]. Hence, this thesis aims at the integration of distributed control and distributed estimation for plant-wide control using equation graphs that capture the connectivity among the relevant variables for the controller as well as the state estimator.

In this thesis, chapters 2-5 are directed towards the distributed control of LPS-DPS networks, and chapter 6 addresses the problem of combined distributed estimation and control. The details of the specific problems addressed in each of the following chapters are discussed below.

Chapter 2: Structural Interaction Based Equation Graphs of Convection-Reaction Systems

We consider DPSs that are convection-reaction systems modeled by linear hyperbolic partial differential equations (PDEs). We define a parameter called the structural interaction parameter (SIP) for the different types of input and output variables [18–20], and show that the SIP captures the strength of the structural interactions among the DPS variables analogous to relative degree among the variables of LPSs. We propose an equation graph representation for convection-reaction systems and establish a relationship between the graph and SIP, which is also analogous to that between LPS graphs and relative degree. We illustrate the process of calculation of the SIPs of the network's variables from its equation graph using a small chemical network consisting of a continuous stirred tank reactor and a plug flow reactor. This chapter presents the theoretical framework for the case study in the next chapter.

Chapter 3: Distributed Control of an Amine Gas Sweetening Plant

We consider an amine gas sweetening plant consisting of a two stage absorption and desorption process. In each stage, sour natural gas is contacted with an aqueous solution of amine, which absorbs the acidic components (carbon dioxide and hydrogen sulfide) from the natural gas. The spent solvent is sent to the desorbers (strippers) where steam removes the acidic gases from the solvent stream. The regenerated solvent is recycled to the absorbers. Two heat exchangers in the network exchange heat across the process streams to reduce the utility consumption in the plant. Plants like this are integral to the liquified natural gas train. The process units in the plant are modeled as linear hyperbolic PDEs. From the equation graph representation and SIP definition for convection-reaction systems, we obtain the optimal decomposition for distributed control using two algorithms - agglomerative clustering [4] and modularity maximization [21]. We simulate the plant-wide control of the plant using the optimal decomposition as the basis for distributed model predictive control (DMPC). Model predictive control (MPC) is a control strategy that entails the calculation of the optimum input profiles over a finite horizon by minimizing an objective function (in our case the control effort and the deviation of the output variables from the set-point) under the constraints applicable on the plant (e.g., material and energy balances). Three cases are simulated - the nominal start-up, start-up with measurement noise, and start-up with disturbance in the feed natural gas composition. The performance of DMPC is compared with that of fully centralized and fully decentralized control architectures.

Chapter 4: Structural Interaction Based Equation Graphs of Diffusion-Convection-Reaction Systems

Diffusion-convection-reaction systems are the more generic kind of DPSs since diffusion is present in nearly all DPSs to some extent. We consider diffusion-convection-reaction

systems modeled by quasi-linear parabolic PDEs. For the different types of input and output variables that have been identified in the literature for such systems, we define an SIP that is analogous to relative degree for LPSs and SIP for convection-reaction systems. We propose a graph representation of these systems as well, in which the length of the input-output path is related to the SIP as in the case of convection-reaction systems. We consider the model of a tubular reactor, and determine the SIPs among the different variables using the definition of the SIP as well as using the equation graph to illustrate the relationship between the two. This chapter presents the unified equation graph based framework to obtain the optimal decompositions for the distributed control of generic chemical networks comprising LPSs and DPSs.

Chapter 5: Distributed Control of a Reaction-Separation Network

We consider the case study of a reaction-separation plant that is encountered in the production of sulphuric acid [22]. It consists of four tubular reactors in series, with interstage heat-exchangers. An exothermic reversible reaction occurs in the reactors and the heat-exchangers drive the equilibrium toward the product side by taking away heat from the reaction mixture. Downstream of the fourth tubular reactor are two flash tanks that separate the unreacted reactant from the product. Material recycle and heat integration make the plant-wide control of this network challenging. Using the equation graph representation from the previous chapter, we simulate the plant-wide control for the set-point tracking problem and compare the performance of the optimal decomposition based DMPC with that of fully centralized MPC.

Chapter 6: Distributed Estimation and Nonlinear MPC using Community Detection

In this chapter we consider the problem of controlling a network when the entire state measurement is not available. For estimating the unmeasured states, we use moving hori-

zon estimation (MHE) [23–25]. Similar to the underlying optimization problem in MPC, MHE involves the repeated solution of an optimization problem to calculate the state variable vector, and, thus, can be computationally expensive. Analogous to distributed control, the distributed approach for MHE is an alternative problem formulation which has the potential to reduce the computational cost. In this chapter, we develop the framework for combined distributed MHE (DMHE) [16, 17] along with DMPC for LPS networks. The optimum network decomposition for DMPC is based on the LPS equation graph that relates the input-output path to the corresponding relative degree [4, 5, 10]. For MHE, we propose an equation graph that relates the measured variables to the state variables. The modularity maximization algorithm discussed in Section 3.4.2 is applied to both of these graphs to obtain the optimum decomposition for the control and estimation tasks. We demonstrate this framework on a case study of a benzene alkylation plant consisting of four continuous stirred tank reactors and a flash tank. We compare the closed-loop performance of different architectures for control and estimation to that of the optimum decomposition based DMPC and DMHE.

Chapter 7: Summary and Conclusions & Chapter 8: Future Work

These chapters contain a brief summary of the findings of this work, discuss the conclusions and propose directions of further research that stem from this work.

Chapter 2

Structural Interaction Based Equation Graphs of Convection - Reaction Systems *

2.1 Introduction

Chemical plants are complex, integrated networks of process systems that can be broadly classified as LPSs, which are described by ordinary differential equations (ODEs) (e.g., well-mixed reactors, staged separators, etc.) and DPSs (e.g., heat exchangers, plug-flow reactors, etc.), which are described by PDEs (hyperbolic or parabolic PDEs). The individual process systems' dynamics and the emerging dynamic behavior due to the interconnections among the systems limit the effectiveness of fully decentralized control of the network [26]. Also, fully centralized control has practical limitations in terms of design and tuning of the controller [27]. Distributed control, a middle-ground between fully de-

*Reprinted with permission from Manjiri Moharir, Lixia Kang, Ali Almansoori and Prodromos Daoutidis, *Computers and Chemical Engineering* 106 (2017), https://doi.org/10.1016/j.compchemeng.2017.07.005.

centralized and fully centralized control, is commonly used for the control of networked systems [3,28–37]. It involves the decomposition of the network into smaller subsystems, which are controlled individually with some transfer of information among the controllers of each sub-system.

The decomposition of a network into sub-networks for distributed control is an open and challenging problem [3]. Some methods have been proposed in the literature [38,39], but there is no generically applicable framework for computing decompositions which are optimal in a well-defined sense [3]. Recently, methods from network theory have been used for systematic decomposition based on the strength of the variable interactions. The contributions in [4–6, 10–14] highlight relative degree as a measure of structural connectivity among a system's variables that can be used to address this problem. The relative degree of a controlled variable with respect to a manipulated variable for continuous-time ODE systems is the smallest order of the time derivative of the output that directly depends on the input. It therefore captures how direct the effect of an input is on an output, and how sluggish the corresponding response is. In [10], it was shown that relative degrees could be calculated from an equation graph capturing the connectivity among the variables in an ODE system, as the shortest path length between the input and the output nodes. Building on this idea, hierarchical clustering procedures [4,5] have been developed to determine input/output clusters for process networks modeled by ODE systems at various levels of decentralization based on structural closeness (in a relative degree sense) within and among clusters. A notion of modularity [21] was used in [6] to evaluate the resulting hierarchy of clusters and identify the optimal decomposition. This approach results in the identification of input-output clusters that are allocated to separate controllers, but does not explicitly allocate state variables to the resulting clusters; yet state variables represent information sharing in a distributed control framework. An algorithm proposed in [7] considers the network equation graph, which is bisected in a hierarchical manner into smaller

blocks so as to minimize input-state and state-state interactions among the resulting clusters. The change in the modularity value [40] with each bisection is maximized to obtain the optimal decomposition. The optimal decomposition thus obtained has been shown to allow for computationally more efficient solutions of the distributed model predictive control (DMPC) design problem without compromising on the closed loop performance as compared to centralized model predictive control (CMPC) [2]. The above-mentioned algorithms, taking advantage of well-established graph theory algorithms, are generically applicable to and well suited for large process networks. They are, however, limited to networks modeled by ODE systems. Chemical plants often comprise interconnected LPSs and DPSs. Currently there exists no method to represent networks of a combination of ODE and PDE systems on a common equation graph that captures the interconnections among all the network variables. A very commonly encountered class of DPSs is that of convection-reaction processes (e.g., heat exchangers, plug flow reactors, adsorption and absorption columns, etc.), which are modeled by first-order hyperbolic PDEs. This chapter discusses an extension of the aforementioned methods to include variables of such first-order hyperbolic PDE systems.

Specifically,

- we propose a structural interaction parameter (SIP) which quantifies the interactions among the variables of first-order hyperbolic PDE systems in a way analogous to relative degree in ODE systems;
- we propose an equation graph representation for hyperbolic PDE systems and establish the graph theoretic interpretation of the structural interaction parameter in terms of the length of input-output paths; and
- we generalize the methods in [4, 6] and [7] to allow for decomposition of process networks modeled by ODE and hyperbolic PDE systems based on the structural

interaction parameters.

2.2 Generic Convection-Reaction Systems

We consider linear, first-order hyperbolic PDE systems of the form:

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{A}\frac{\partial \mathbf{x}}{\partial z} + \mathbf{B}\mathbf{x} + \mathbf{G}\mathbf{u}(z,t) \tag{2.1}$$

where, $\mathbf{x}(z,t) = [x_1(z,t)\cdots x_{n_x}(z,t)]^T \in \mathbb{R}^{n_x}$ is the vector of distributed state variables, n_x is the number of state variables, $z \in [0,L] \subset \mathbb{R}$ is the spatial coordinate (L being the length of the spatial domain), $t \in [0,\infty)$ is the time, and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{G}$ are constant matrices of conforming dimensions. The matrix $\mathbf{A}$ is diagonal and contains the velocities, v_i contributing to the convective transport in the system. Non-linear first-order hyperbolic PDEs can be linearized around a steady state to the generic form described above.

The boundary conditions are:

$$\mathcal{K}_1\mathbf{x}(0,t) + \mathcal{K}_2\mathbf{x}(L,t) = \mathcal{R}(t) \tag{2.2}$$

where $\mathcal{K}_1$ and $\mathcal{K}_2$ are constants and $\mathcal{R}(t)$ is a smooth function of time. The initial conditions are:

$$\mathbf{x}(z,0) = \mathbf{x}_0(z) \tag{2.3}$$

The manipulated inputs to the system could be of the following types:

- Velocity inputs: the elements of the diagonal matrix $\mathbf{A}$.
- Distributed inputs: the elements of the vector $\mathbf{u}(z,t)$.
- Boundary inputs: the elements of the vector $\mathcal{R}(t)$.

The controlled outputs of the system could be of the following types:

- Distributed outputs: functions of state variables that are dependent on space and time:

$$\mathbf{y}_d(z,t) = \mathbf{H}_d\mathbf{x} \tag{2.4}$$

  where $\mathbf{H}_d$ is a constant matrix of conforming dimensions.

- Boundary outputs: functions of the values of the state variables at a certain spatial location (typically a boundary), that are functions of time alone, e.g.,

$$\mathbf{y}_b(t) = \mathbf{H}_b\mathbf{x}|_{z=L} \tag{2.5}$$

  where $\mathbf{H}_b$ is a constant matrix of conforming dimensions.

  Boundary inputs and outputs can be treated as a special case of distributed ones, as will be discussed in the subsequent sections.

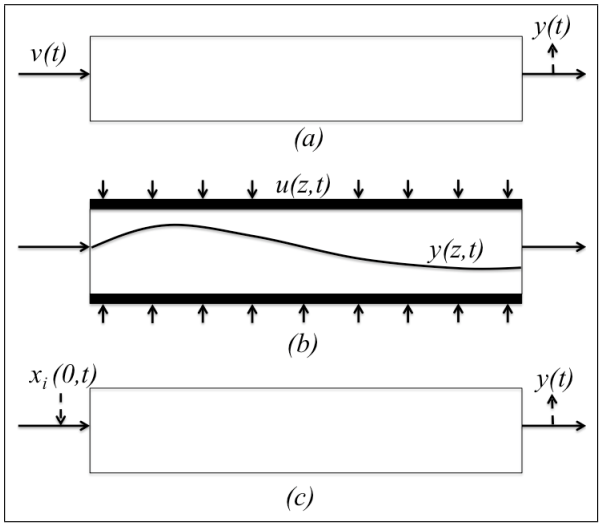


Figure 2.1: Types of input - output combinations: (*a*) Velocity input - Boundary output, (*b*) Distributed input - Distributed output, (*c*) Boundary input - Boundary output

Figure 2.1 shows combinations of different types of inputs and outputs that are encountered in distributed chemical processes modeled by first-order hyperbolic PDEs.

The goal is to develop a method to represent on a graph the structural interactions among the input, state and output variables of multiple DPSs of the above form possibly connected to LPSs. The first step to this end is the selection of parameters that quantify the structural proximity among the variables. In what follows, we define the *structural interaction parameter* (SIP) for specific, typical combinations of inputs and outputs, allowing for a natural generalization of the concept of relative degree as a measure of structural interactions and a unified graph representation of such interactions for hyperbolic PDE systems.

2.3 Structural Interaction Parameters for Hyperbolic PDEs

1. *Velocity Input - Boundary Output Case*: In this case (figure 2.1-(a)), if the fluid is assumed to be incompressible, then despite the distributed nature of the system, the input and output variables depend solely on time (as in the case of an ODE system).

   In [19], a concept analogous to relative degree was postulated for such systems under the assumption that there is a single stream of an incompressible fluid flowing in the system. For such systems, the SIP of an output with respect to the manipulated flow velocity can be defined as this relative degree analogue, i.e. the smallest order time derivative of the output which explicitly depends on the manipulated flow velocity. For example, if the first time-derivative of the output y_j ($j \in \{1, n_y\}$), given by:

$$\frac{dy_j}{dt} = \sum_{i=1}^{n_x} \left(h^b_{ji} v_i \frac{\partial x_i}{\partial z} \right) \Bigg|_{z=L} + (\mathbf{h}^{b,T}_j \mathbf{B}\mathbf{x}) \Big|_{z=L} \tag{2.6}$$

   where h^b_{ji} is the $(j,i)^{th}$ element of $\mathbf{H}_b$ and $\mathbf{h}^{b,T}_j$ is the j^{th} row of $\mathbf{H}_b$, explicitly de-

pends on the velocity v_i, that is:

$$h^b_{ji}\frac{\partial x_i}{\partial z}\bigg|_{z=L} \neq 0 \tag{2.7}$$

then, the SIP of y_j with respect to v_i is equal to 1. If equation (2.7) is not satisfied, then the SIP is greater than 1, and can be determined by further differentiation of the output variable.

This concept can be directly extended to systems with multiple flow velocities (e.g., counter-current heat-exchangers, absorber columns, etc.).

2. *Distributed Input - Distributed Output Case*: In this case, the input and the output are functions of space and time as shown in (figure 2.1-(b)).

   Consider a single-input single-output (SISO) first-order hyperbolic PDE system as described by equation (2.1). In the case of distributed actuation, it is customary to assume that the distributed input variable, $u(z,t)$, is of the form of a finite number (n_z) of actuators distributed over the length of the DPS, and each actuator acts on a spatial interval (e.g., $[z_{k-1}, z_k] \quad \forall k \in \{1, n_z\}$) [18]. Mathematically, this implies that:

$$u(z,t) = p_k(z)\hat{u}_k(t) \tag{2.8}$$

   where $p_k(z)$ is a smooth function defined over the k^{th} spatial interval ($z \in [z_{k-1}, z_k]$). Also, the variable $y_d(z,t) = \mathbf{h}_d^T \mathbf{x}(z,t)$ is usually controlled over a finite number of spatial intervals ($[z_{k-1}, z_k] \; \forall \; k \in \{1, n_z\}$) by considering n_z spatially averaged output variables given by:

$$\hat{y}_k(t) = \mathcal{C}_k y(z,t) \tag{2.9}$$

where $\mathcal{C}_k$ is a bounded operator defined over the interval $[z_{k-1}, z_k]$ of the form:

$$\mathcal{C}_k y(z,t) = \int_{z_{k-1}}^{z_k} c_k(z) y(z,t) dz \tag{2.10}$$

with $c_k(z)$ a smooth function. Hence, the variables whose interactions are relevant are:

$$\hat{\mathbf{u}}(t) = [\hat{u}_1(t) ... \hat{u}_{n_z}(t)]^T \tag{2.11}$$

$$\hat{\mathbf{y}}(t) = [\hat{y}_1(t) ... \hat{y}_{n_z}(t)]^T \tag{2.12}$$

These are finite dimensional vectors, which depend solely on time. For these finite dimensional input and output vectors, the concept of characteristic index $\sigma^{(k)}$, of $\hat{y}_k$ with respect to $\hat{u}_k$ was defined in [18] as the smallest order of time derivative of $\hat{y}_k$ in which $\hat{u}_k$ appears explicitly. This is equivalent to determining the smallest integer, $\sigma^{(k)}$, for which the following equation is satisfied:

$$\mathbf{h}_d^T \left(\mathbf{A} \frac{\partial}{\partial z} + \mathbf{B} \right)^{\sigma^{(k)}-1} \mathbf{g} \neq 0 \tag{2.13}$$

Typically, the characteristic indices for all pairs $\{\hat{u}_k / \hat{y}_k\}$ are equal, and the corresponding value ($\sigma^{(1)} = \sigma^{(2)} ... \sigma^{(n_z)} = \sigma$) can be viewed as the characteristic index of $\hat{\mathbf{y}}(t)$ with respect to $\hat{\mathbf{u}}(t)$ [18]. This concept was defined for a SISO system in [18], but, it can be directly extended to a multiple-input multiple-output (MIMO) system. This concept of characteristic index will be used as the SIP in this case of distributed actuation.

3. *Boundary Input - Boundary Output Case*: This can be considered as a specific instance of the distributed input - distributed output case, wherein the input has the

form:

$$u(z,t) = \delta(z)x_m(z,t) \quad m \in \{1, n_x\} \tag{2.14}$$

and the output has the form:

$$y(z,t) = \delta(z-L)\mathbf{h}_b^T\mathbf{x} \tag{2.15}$$

where $\delta(z)$ is the Dirac delta function.

Note that the input and output variables above are at different spatial locations (non-collocated). If the SIP definition from the distributed input - distributed output case is directly applied to this case, the SIP of the output with respect to the input would be ∞ [20]. To overcome this problem, the contribution in [41] defined a modified output variable as:

$$\hat{y}(t) = \int_0^L c(z)\mathbf{h}_b^T\mathbf{x}dz \tag{2.16}$$

where $c(z)$ is a smooth shaping function defined on $[0, L]$.

The first time-derivative of the new output is given by:

$$\frac{d\hat{y}(t)}{dt} = \int_0^L c(z)\mathbf{h}_b^T\frac{\partial\mathbf{x}}{\partial t}dz \tag{2.17}$$

Substituting for $\partial\mathbf{x}/\partial t$ from the system dynamics with the input given by equation (2.14), we get:

$$\frac{d\hat{y}(t)}{dt} = \int_0^L c(z)\mathbf{h}_b^T\left(\mathbf{A}\frac{\partial\mathbf{x}}{\partial z} + \mathbf{B}\mathbf{x} + \mathbf{G}\delta(z)x_m(z,t)\right)dz \tag{2.18}$$

The characteristic index of $\hat{y}(t)$ with respect to input $u(z,t)$ will be finite if and only

if $c(z)$ is chosen such that:

$$\int_0^L c(z)\delta(z)dz \neq 0 \tag{2.19}$$

Hence, any smooth function $c(z)$ that approximates $\delta(z-L)$ but also satisfies $c(0) \neq 0$ can be chosen for the new output to "collocate" the input and output variables [20]. We adopt this definition of the characteristic index as the SIP for the case of boundary input - boundary output. This definition can be directly extended to a MIMO boundary input - boundary output case too.

2.4 Equation Graphs of a Network of Hyperbolic PDE Systems

We now propose a graph representation of a network comprising hyperbolic PDE systems. This representation can be seamlessly integrated with a standard equation graph representation of ODE systems, thus allowing the graph theoretic analysis of networks comprising both LPSs and DPSs. For this purpose, the following graph representation of a network is proposed:

- There is a node for every input variable, which could be a manipulated velocity $v_i(t)$, or a finite dimensional distributed input $\hat{\mathbf{u}}_i(t)$ (equation (2.11)) or a boundary manipulation $x_i(z_b,t)$ acting at spatial location $z_b = 0$ or L.
- There is a node for every output variable, which could be the finite dimensional distributed output $\hat{\mathbf{y}}_j(t)$ (equation (2.12)), or spatially averaged output $\hat{y}(t)$ defined to capture the boundary outputs (equation (2.16)).
- There is a node for every spatially averaged state variable defined by:

$$\hat{x}_m(t) = \int_0^L c_m(z)x_m(z,t)dz \tag{2.20}$$

where $c_m(z)$ is a smooth shaping function defined on $[0, L]$. The spatial averaging for the state variables is performed for consistency since every input and output variable on the equation graph is a function of time alone. The weight functions ($c_m(z)$ in equation (2.20)) are consistent with those used for the output variables.

- There is an edge from $v_i(t)$ to $\hat{x}_m(t)$ if $v_i(t)$ contributes to the convective transport of state variable $x_m(t)$.
- There is an edge from $\hat{\mathbf{u}}_i(t)$ to $\hat{x}_m(t)$ if $\hat{\mathbf{u}}_i(t)$ appears in the first-order time derivative of $\hat{x}_m(t)$.
- There is an edge from $x_i(z_b, t)$ to $\hat{x}_m(t)$ if $x_i(z_b, t)$ is the boundary condition of $x_m(z, t)$.
- There is an edge from $\hat{x}_m(t)$ to $\hat{x}_n(t)$ if $\hat{x}_m(t)$ appears in the first-order time derivative of $\hat{x}_n(t)$.
- There is an edge from $\hat{x}_m(t)$ to any output variable node if the output variable is a linear function of the state variable $\hat{x}_m(t)$.

In this equation graph,

- A *path* is defined as the sequence of edges such that an edge terminates at the node from which the succeeding edge of the sequence begins [10]. An input-output path begins at an input node and terminates at an output node.
- The number of edges contained in a path is called the *length of the path*.

In an equation graph as that defined above, the following theorems hold, which generalize a similar result for ODE systems [10].

Theorem 1: *In the equation graph of a linear first-order hyperbolic PDE system with velocity manipulation $v_i(t)$ and a boundary output $\hat{y}_j(t)$, let the length of the shortest path from $v_i(t)$ to $\hat{y}_j(t)$ be l_{ij} and the corresponding SIP be σ_{ij}. Then*

$$\sigma_{ij} = l_{ij} - 1 \tag{2.21}$$

The proof of Theorem 1 is given in the Appendix A.

Theorem 2: *In the equation graph of a linear first-order hyperbolic PDE system with a distributed input $\hat{\boldsymbol{u}}_i(t)$ and a distributed output $\hat{\boldsymbol{y}}_j(t)$, let the length of the shortest path from input $\hat{\boldsymbol{u}}_i(t)$ to output $\hat{\boldsymbol{y}}_j(t)$ be l_{ij} and the corresponding SIP be σ_{ij}. Then*

$$\sigma_{ij} = l_{ij} - 1 \tag{2.22}$$

The proof of Theorem 2 is given in Appendix B.

Corollary 1: *In the equation graph of a linear first-order hyperbolic PDE system with a boundary input $u_i(t)$ and a boundary output $\hat{y}_j(t)$, let the length of the shortest path from input $u_i(t)$ to output $\hat{y}_j(t)$ be l_{ij} and the corresponding SIP be σ_{ij}. Then*

$$\sigma_{ij} = l_{ij} - 1 \tag{2.23}$$

Corollary 1 follows from Theorem 2, because a boundary input - boundary output pair is a special case of a distributed input - distributed output pair; its proof is therefore omitted.

2.5 Illustrative Example

The proposed graph representation is now illustrated by an example comprising of both LPSs and DPSs. Specifically, consider the network shown in figure 2.2, of an ideal continuous stirred tank reactor (CSTR, an ODE system) connected in series with a plug flow reactor (PFR, a first-order hyperbolic PDE system).

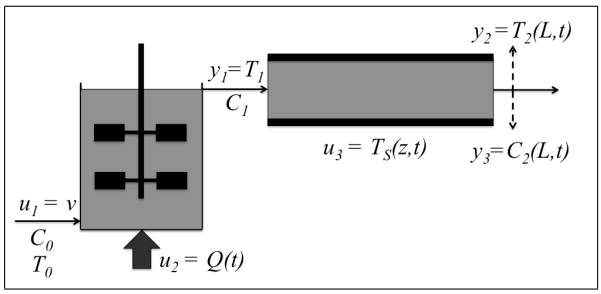


Figure 2.2: CSTR and PFR in series

The fluid in these reactors is incompressible. A reactant at concentration $C_0(t)$ is fed to the CSTR, where it reacts at concentration $C_1(t)$. The temperature in the CSTR is $T_1(t)$. In the PFR, the concentration profile is $C_2(z,t)$ and the temperature profile is $T_2(z,t)$. The rate constant of the reaction follows the Arrhenius equation, $k(T) = k_0 e^{-E_A/RT}$, and the enthalpy of the reaction is ΔH. The manipulated variables in the network are:

- $v(t)$: the flow velocity of the fluid in the system.
- $Q(t)$: the heat input given to the CSTR.
- $T_S(z,t)$: the temperature profile in the jacket of the PFR.

The controlled outputs in the network are:

- $T_1(t)$: the temperature in the CSTR.
- $T_2(L,t)$: the temperature at the exit of the PFR.
- $C_2(L,t)$: the concentration of the reactant at the exit of the PFR.

The CSTR has volume V_1 and residence time τ_1. The PFR has space-time τ_2. The fluid has density ρ and specific heat capacity C_p.

The material and energy balance equations, with the boundary inputs incorporated using the Dirac delta function are given by:

$$\frac{dC_1}{dt} = \frac{C_0 - C_1}{\tau_1} - k(T_1)C_1 \tag{2.24}$$

$$\frac{dT_1}{dt} = \frac{T_0 - T_1}{\tau_1} + \frac{Q}{\rho C_p V_1} + \frac{-\Delta H}{\rho C_p} k(T_1)C_1 \tag{2.25}$$

$$\frac{\partial C_2}{\partial t} = -v\frac{\partial C_2}{\partial z} - k(T_2)C_2 + v\delta(z)C_1 \tag{2.26}$$

$$\frac{\partial T_2}{\partial t} = -v\frac{\partial T_2}{\partial z} + \frac{hA(T_s - T_2)}{\rho C_p} - \frac{\Delta H}{\rho C_p} k(T_2)C_2 + v\delta(z)T_1 \tag{2.27}$$

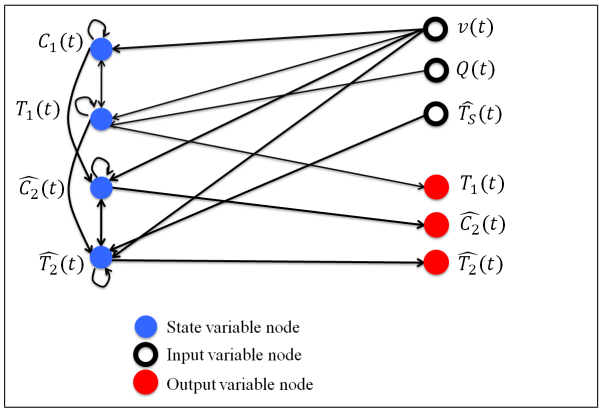


Figure 2.3: Equation graph of CSTR-PFR in series

The equation graph for this network, after reformulating the variables as discussed previously, is given in figure 2.3. It can be seen from the equation graph that the length of the path from $v(t)$ to $T_1(t)$ is 2. Equivalently, $v(t)$ appears in the first time derivative of $T_1(t)$, so the relative degree of $T_1(t)$ with respect to $v(t)$ is 1. Similar is the case for all

other input-output interactions. The SIPs can, thus, be calculated from the equation graph using the theorems and the corollary stated above as:

$$\begin{array}{c} \\ v(t) \\ Q(t) \\ \hat{T}_S(t) \end{array} \begin{array}{c} \begin{array}{ccc} T_1 & \hat{T}_2(t) & \hat{C}_2(t) \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 2 \\ \infty & 2 & 1 \end{bmatrix} \end{array}$$

2.6 Conclusions

We consider the case of convection-reaction systems modeled using linear, first-order hyperbolic PDEs. For the different types of input and output variables in these systems, we propose a relative degree analogue called the SIP, and establish a relationship between the SIP and the equation graph representation of the hyperbolic PDE systems. This relationship is also analogous to that between relative degree and LPS equation graphs. We use this relationship to determine the structural distance among different input and output variables in a network of a CSTR and a PFR. This chapter forms the basis for the case study of plant-wide control of an amine gas sweetening plant discussed in the next chapter.

Chapter 3

Distributed Control of an Amine Gas Sweetening Plant *

3.1 Introduction

Natural gas is a widely used petroleum based fuel that provided nearly a quarter of the global energy demand in 2010 [42]. Natural gas from a large fraction of reserves contains sour gases like hydrogen sulphide (H_2S) and carbon dioxide (CO_2) [42]. CO_2 has no heating value. Also, these sour gases cause corrosion in the pipelines. To minimize corrosion and meet the fuel specifications, the sour gases need to be removed from natural gas before it can be used as a fuel [43, 44]. Several technologies have been implemented for the removal of sour gases, such as physical absorption, chemical absorption, adsorption, and membrane based separation [42, 45]. The conventional approach for the simultaneous removal of these sour gases is the use of amines [42, 44]. Absorption of H_2S and CO_2 by contacting natural gas in packed columns with monoethanolamine (MEA) is a widely

*Reprinted with permission from Manjiri Moharir, Davood B. Pourkargar, Ali Almansoori and Prodromos Daoutidis, *Industrial & Engineering Chemistry Research*, 57(39) (2018), https://pubs.acs.org/doi/abs/10.1021/acs.iecr.8b01291. Copyright ©2018, American Chemical Society

used commercial technology [42, 46]. Due to economic and environmental concerns, the spent solvent is regenerated in stripper columns using steam, wherein the sour gases are removed from the solvent. This regenerated solvent is recycled into the absorber column. To ensure that the specifications of the sour gas content in the sweetened gas are met with minimum waste of amine, effective control of the gas sweetening plant is crucial. Using amines for sour gas removal is of interest not just for natural gas processing, but also for post-combustion CO_2 capture [45].

A lot of research has been dedicated towards amine-based absorption and solvent regeneration for sour gas removal. Areas of focus have been the kinetics and solubilities of sour gases in amine solvents [47–51], degradation of and corrosion from solvents [52–54], selection of solvents [46, 54–57] etc. The models of sour gas absorption in amine solvents are derived following an equilibrium-based approach (in which the liquid and gas streams are assumed to attain chemical equilibrium at a theoretical stage) or a rate-based approach (the rates of mass and heat transfer are incorporated into the model). Rate-based approaches are more rigorous in general [44, 45, 58]. Several studies have focused on building a realistic rate-based model for the absorption of sour gases in amine solvents with the aim of estimating the enhancement factors in the absorption kinetics [59–62], coupling of interfacial transport processes with material and energy balances in the column [63–65], and gauging dominating film resistances and thermal effects [58]. The modeling of the absorber and the regenerator dynamics has also been a focus of research [66,67]. For post-combustion CO_2 absorption, performance maintenance during varying loads, the effects of start-up and load variation, and the effects of energy consumption and reboiler duty on meeting the specifications have been studied via rigorous dynamic models [67–70]. For the gas sweetening process, the absorber and regenerator were modeled in TSWEET [71], HYSYS [43,72,73] and ASPEN [74] to study the effects of mixed amine solutions on sour gas absorption from natural gas [71, 72], the effect of the incoming solvent temperature

on sour gas absorption [73], the effects of temperature, solvent circulation rate and amine concentration on the amine absorption rate [74], and to determine the nominal process parameters for optimal economics [43].

On the control side, SISO generalized predictive control [75] and MPC [76] have been implemented on a CO_2 absorption column, neglecting the effect of the interactions among different process units in the plant. Control structures have been designed for single-stage post-combustion amine absorption and solvent regeneration for minimizing the cost and energy demand for varying operating ranges [77] and for plant-wide control based on heuristics [78]. Post combustion carbon capture is a different operation as compared to the amine gas sweetening process in terms of operating conditions, possible disturbances and product specifications. To the best of our knowledge, little work seems to have been done towards the plant-wide control of the gas sweetening process [76]. Motivated by this, we aimed to design and evaluate plant-wide model-based controllers for a two-stage amine gas sweetening plant.

MPC is a widely implemented control strategy for chemical processes. A potential limitation of MPC is the solvability of the underlying optimization problem in real time for large scale processes. The development of optimization methods to overcome this limitation is an active subject of research [79–81]. One of the ways of overcoming this limitation is to implement MPC through separate local controllers, each of which handles the control problem of a smaller part of the large scale process, with varying degrees of interaction and communication. This is the distributed MPC (DMPC) approach, which represents a middle ground between a fully centralized control of a large scale process and a fully decentralized control, which is often rendered ineffective by a high degree of material and energy interactions among the process units [3, 28, 35]. The decomposition of the large control problem into smaller sub-problems has a significant effect on the control performance of DMPC [2]. To this end, as mentioned in the previous chapter,

network theory-based methods to obtain decompositions of process networks comprising LPSs modeled by ODEs with varying degrees of decentralization have been recently developed [4, 5] and their extension to process networks that include convection-reaction systems modeled by first-order hyperbolic differential equations has been proposed [82]. Also, a method to obtain the optimal decomposition based on the notion of modularity [7] has been proposed. The impact of the optimal decomposition has been investigated for distributed control designs considering different classes of dynamic optimization solvers, different levels of cooperation and communication between local controllers, for general set-point tracking problem over a wide operating range [83]. A systematic study [2] that compares the performances of DMPC based on various decomposition algorithms that have been proposed in the literature, concludes that implementing DMPC on the optimal distributed architecture obtained through the modularity maximization algorithm [7] has a performance comparable to that of fully centralized MPC but with significantly reduced computational cost.

This motivates us to test the application of DMPCs to the amine gas sweetening plant. More specifically, our aims are:

1. to implement DMPC on a rate-based model of a two-stage amine gas sweetening network, which includes material and heat integration among six convection-reaction systems, using the optimal decomposition obtained from modularity maximization [7];

2. to test the efficacy of the above distributed control structure, the impact of different decompositions, and the performance and computational requirements of different architectures including fully centralized and decentralized ones.

As such, this work is the first DMPC implementation of network decomposition algorithms for distributed parameter process networks, as well as a model-based plant-wide

control study of a realistic amine gas sweetening plant.

3.2 The Amine Gas Sweetening Plant

3.2.1 Plant Description

The gas sweetening plant we considered is a two-stage absorption and regeneration network (Figure 3.1) [84]. There are two absorbers with inert packing connected in series. Natural gas containing sour gases like hydrogen sulphide and carbon dioxide is contacted with solvent streams containing aqueous MEA solution in the absorber columns. The solvent streams from each of the absorber columns are sent to the respective regenerator columns, where the streams are contacted with fresh steam. The flowrates of steam to each of the regenerator columns are separately manipulated. The steam absorbs the sour gases from the aqueous MEA, thus regenerating the solvent. The lean solvent is then recycled to the absorber columns. The heat exchangers are used for pre-heating the feed gas going into the first-stage absorber and cooling the solvent stream going into the second-stage absorber. There are other process units in the plant, such as pumps and valves, whose dynamics are much faster as compared to the aforementioned processes. Also, buffer tanks are often included in the plant to reduce the interactions among the columns. In our study, we have excluded these processes. Instead we focus at the core process network in the integrated gas sweetening plant.

3.2.2 Plant Model

The absorption and regeneration columns were modeled assuming negligible axial and radial dispersion. The liquid and gas phase flowrates, heat capacities and densities were assumed to be constant since the concentrations of the species transferred across phases

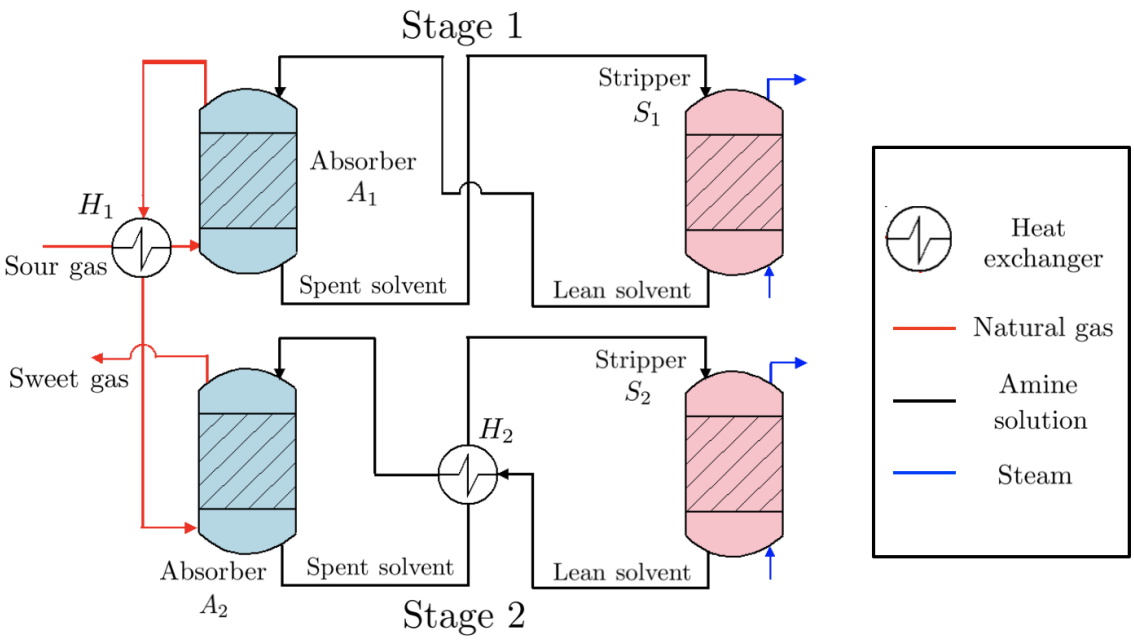


Figure 3.1: Amine gas sweetening plant

are typically low. Mass transfer rate-limited absorption was modeled using the two-film theory [45]. In the liquid phase, CO_2 and H_2S react with the amine. The reactions, which are assumed to be in equilibrium, are:

$$CO_2 + 2MEA \rightleftharpoons MEAH^+ + MEACOO^- \tag{3.1}$$

$$H_2S + MEA \rightleftharpoons MEAH^+ + HS^- \tag{3.2}$$

The total carbon and sulfide content in the liquid phase is the sum of the concentrations of the ionic and non-ionic forms of the species:

$$C_C^l = C_{CO_2}^l + C_{MEACOO^-}^l \tag{3.3}$$

$$C_S^l = C_{H_2S}^l + C_{HS^-}^l \tag{3.4}$$

We assume that no reaction occurs in the gas phase due to negligible concentration of amine. With the aforementioned assumptions, the material balance in the gas and liquid is given by:

$$\frac{\partial C_i^g}{\partial t} = -v^g \frac{\partial C_i^g}{\partial z} - N_i \tag{3.5}$$

$$\frac{\partial C_i^l}{\partial t} = v^l \frac{\partial C_i^l}{\partial z} + N_i \tag{3.6}$$

In the above equations, C_i^g and C_i^l represent the concentration of species i (carbon or sulphide) in the gas and liquid phase respectively (mol/m^3), t is time (s), v^g is the gas velocity (m/s), v^l is the liquid velocity (m/s), z is the axial coordinate (m). N_i is the mass transfer rate in mol/m^3-s, which is given by

$$N_i = K_{ov} a (P - P_{eq}(T)) \tag{3.7}$$

where a is the area per unit volume available for the given packing, and K_{ov} is the overall mass transfer coefficient given by

$$\frac{1}{K_{ov}} = \frac{\delta^l}{E \mathcal{D}^l} + \frac{R T^g \delta^g}{\mathcal{D}^g} \tag{3.8}$$

where R is the universal gas constant, T^g is the gas side temperature, E is the enhancement factor (which reduces the liquid side resistance [58]), $\mathcal{D}^l$ and $\mathcal{D}^g$ are the diffusion coefficients of a species in the liquid and gas phase respectively [85], and δ^l and δ^g are the widths of the stagnant films on the liquid and gas side [86] respectively. The driving force for mass transfer depends on the partial pressure of the species in the gas phase (P in equation (3.7)) and the equilibrium pressure ($P_{eq}(T)$) dictated by the vapor-liquid equilibrium. The partial pressure in the gas phase is calculated from the gas phase concentration assuming ideal gas behavior. The equilibrium pressure is calculated assuming that Henry's

Law holds. Henry's constant is calculated using the following equation:

$$H(T) = H^0 exp\left(\frac{d(ln(H(T)))}{d(1/T)}\left(1/T - 1/T^0\right)\right) \tag{3.9}$$

where H^0 is the Henry's constant value at the reference temperature T^0. The value of $d(ln(H(T)))/\ d(1/T)$ is taken as a species-specific constant [87]. The energy balance equations for the absorber and the regenerator columns are given by:

$$\frac{\partial T^l}{\partial t} = v^l \frac{\partial T^l}{\partial z} + \frac{\sum_j N_j}{\sum_j (C_j^l \tilde{C}_{p,j})} \Delta \tilde{H}_v - \frac{ah(T^l - T^g)}{\sum_j (C_j^l \tilde{C}_{p,j}^l)} \tag{3.10}$$

$$\frac{\partial T^g}{\partial t} = -v^g \frac{\partial T^g}{\partial z} + \frac{ah(T^l - T^g)}{\sum_j (C_j^g \tilde{C}_{p,j}^g)} \tag{3.11}$$

where $\tilde{C_p}$ (J/mol-K) is the molar specific heat capacity, h (J/m^2-s-K) is the heat transfer coefficient, t (s) is the time, T (K) is the temperature of the stream, and $\Delta \tilde{H}_v$ (J-m^3/mol) is the enthalpy released by the mass transfer. The superscript l denotes the liquid stream, and g denotes the gaseous stream. The heat transfer coefficient is calculated using the following equation:

$$1/h = 1/h^l + 1/h^g \tag{3.12}$$

where h^l and h^g are the heat transfer coefficients for this system on the liquid and gas side respectively [85].

The heat exchangers are modeled assuming negligible axial and radial dispersion. There is no phase change in either of the heat exchangers. The change of heat capacities with respect to the temperature is neglected. The model equations of the heat exchangers are given by:

$$\frac{\partial T_h}{\partial t} = -v_h \frac{\partial T_h}{\partial z} - \frac{UA}{m_h C_{ph}} (T_h - T_c) \tag{3.13}$$

$$\frac{\partial T_c}{\partial t} = v_c \frac{\partial T_c}{\partial z} + \frac{UA}{m_c C_{pc}}(T_h - T_c) \tag{3.14}$$

C_p (J/kg-K) is the mass specific heat capacity, m (kg/s) is the mass flow rate, v (m/s) is the flow velocity, UA (J/K) is the overall heat transfer coefficient multiplied by the area available in the heat exchangers, subscript h represents the hot stream in the heat exchanger, and subscript c denotes the cold stream in the heat exchanger.

The system is simulated from the start-up conditions, so the initial values of all the state variables were assigned a number as close to zero as was acceptable for the solver without making the system singular. The process parameter values are listed in Table 3.1 and the boundary conditions for each of the columns which are dependent on the state variables of the interconnected columns are listed in Table 3.2. In the tables, C represents concentration of the component being absorbed, T is the stream temperature, subscripts A_j, S_j and HE_j represent that the variable is associated with the absorber, stripper or heat exchanger of index j, t is the time, L is the axial length of the column, and the superscripts l and g represent the liquid or gaseous phase of the stream.

Since all the columns and heat exchangers operate in counter-current mode, we implemented forward and backward finite-differences to discretize the PDEs of the model (Figure 3.2). This discretization scheme was chosen over more advanced ones for simplicity. For the streams where the boundary conditions were applied at $z = 0$, backward finite differences were implemented:

$$\left.\frac{\partial x}{\partial z}\right|_k = \frac{x_k - x_{k-1}}{\Delta z} \tag{3.15}$$

For the streams where the boundary conditions were applied at $z = L$, forward finite differences were implemented:

$$\left.\frac{\partial x}{\partial z}\right|_k = \frac{x_{k+1} - x_k}{\Delta z} \tag{3.16}$$

Table 3.1: Process parameter values

Parameters	Values
a	198 m^{-1}
$\tilde{C_p}^l$	4.19 kJ/kg-K
$\tilde{C_p}^g$ - Natural gas	2.8 kJ/kg-K
$\tilde{C_p}^g$ - Steam	1.9 kJ/kg-K
h	0.5 W/m^2-K
H^0 - CO_2	0.035 mol/kg-bar
H^0 - H_2S	0.1 mol/kg-bar
$d(ln(H(T)))/\ d(1/T)$ - CO_2	2400 K
$d(ln(H(T)))/\ d(1/T)$ - H_2S	2300 K
$\Delta\tilde{H}_v$	-84 kJ/mol
K_{ov}	9.4×10^{-6} mol/m^2-s-Pa
L_A	10 m
L_S	10 m
L_{HE}	3 m
ρ^l	1000 kg/m^3
ρ^g	1 kg/m^3
T_0	298 K
UA	10 W/m^2-s

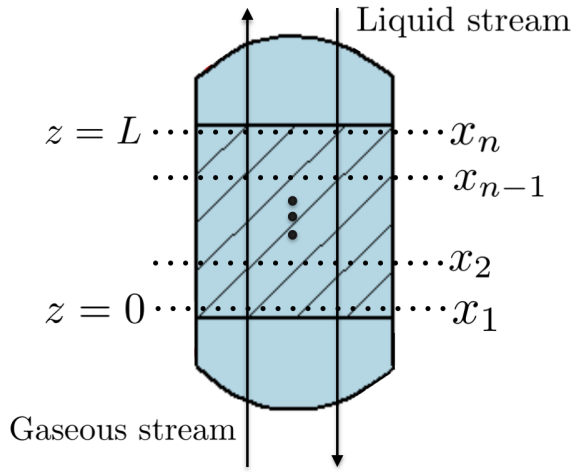


Figure 3.2: Discretization of the distributed variables

Table 3.2: Boundary conditions of the model equations

Variables	**Boundary condition**
$C^g_{A_1,CO_2}(t,0)$	2.23 mol/l
$C^g_{A_1,H_2S}(t,0)$	3.23 mol/l
$C^l_{A_1,CO_2}(t,L_{A_1})$	$C^l_{A_2,CO_2}(t,0)$
$C^l_{A_1,H2S}(t,L_{A_1})$	$C^l_{A_2,H2S}(t,0)$
$T^g_{A_1}(t,0)$	$T^c_{HE_1}(t,0)$
$T^l_{A_1}(t,L_{A_1})$	$T^l_{A_2}(t,0)$
$C^g_{S_1,CO_2}(t,0)$	0 mol/l
$C^g_{S_1,H_2S}(t,0)$	0 mol/l
$C^l_{S_1,CO_2}(t,L_{S_1})$	$C^l_{A_1,CO_2}(t,0)$
$C^l_{S_1,H_2S}(t,L_{S_1})$	$C^l_{A_1,H_2S}(t,0)$
$T^g_{S_1}(t,0)$	373 K
$T^l_{S_1}(t,L_{S_1})$	$T^l_{A_1}(t,0)$
$T^h_{HE_1}(t,0)$	$T^g_{A_1}(0,L_{A_1})$
$T^c_{HE_1}(t,L_{HE_1})$	273 K
$C^g_{A_2,CO_2}(t,0)$	$C^g_{A_1,CO_2}(t,L_{A_1})$
$C^g_{A_2,H_2S}(t,0)$	$C^g_{A_1,H_2S}(t,L_{A_1})$
$C^l_{A_2,CO_2}(t,L_{A_2})$	$C^l_{S_2,CO_2}(t,0)$
$C^l_{A_2,H2S}(t,L_{A_2})$	$C^l_{S_2,H2S}(t,0)$
$T^g_{A_2}(t,0)$	$T^h_{HE_1}(t,L_{HE_1})$
$T^l_{A_2}(t,L_{A_2})$	$T^h_{HE_2}(t,L_{HE_2})$
$C^g_{S_2,CO_2}(t,0)$	0 mol/l
$C^g_{S_2,H_2S}(t,0)$	0 mol/l
$C^l_{S_2,CO_2}(t,L_{S_2})$	$C^l_{A_2,CO_2}(t,0)$
$C^l_{S_2,H_2S}(t,L_{S_2})$	$C^l_{A_2,H_2S}(t,0)$
$T^g_{S_2}(t,0)$	373 K
$T^l_{S_2}(t,L_{S_2})$	$T^c_{HE_2}(t,0)$
$T^h_{HE_2}(t,0)$	$T^l_{S_2}(t,0)$
$T^c_{HE_2}(t,L_{HE_2})$	$T^l_{A_2}(t,0)$

In the above equations, k represents the index of the discrete variable, and Δz represents the spatial discretization step. By doing this the system of PDEs is transformed into a system of ODEs of the form:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t) \tag{3.17}$$

$$\mathbf{y}(t) = h(\mathbf{x}(t)) \tag{3.18}$$

where $\mathbf{x}$ is the vector of all the discretized distributed (i.e., functions of both space and time) variables, $\mathbf{y}$ is the vector of the outputs of the system, $f(\mathbf{x}(t))$, $g(\mathbf{x}(t))$ and $h(\mathbf{x}(t))$ are smooth functions. $\mathbf{u}(t)$ is the vector of manipulated inputs to the network. We used 30 discretization points for the absorber and stripper columns, and 10 points for the heat exchangers, since with finer discretization we found the steady state profiles to change negligibly. The model has a total of 741 discretized state variables (from the original 28 distributed state variables). For the control implementation, we used a reduced model which has 233 discrete state variables. It was found that using the reduced model produces a solution within 10% from the model's exact solution in open-loop simulation and reduces the computation time for the controller, without significantly compromising the closed-loop performance. Thus, the state vector $\mathbf{x}$ has dimensions of 741×1 for the model and 233×1 for the controller. We considered 6 inputs and 5 outputs, which are discussed in the sections that follow. So, $\mathbf{u}$ has the dimensions of 6×1, and $\mathbf{y}$ has the dimensions of 5×1.

3.3 Model Predictive Control Architectures

Two MPC formulations were considered for the plant-wide control. These are discussed below.

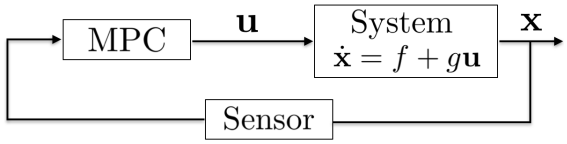


Figure 3.3: Centralized MPC

3.3.1 Centralized MPC

A single controller is used (Figure 3.3) for which the optimized input variable profiles are calculated by solving the following optimization problem:

$$\begin{aligned}
\underset{u}{\text{minimize}} \quad & \int_{t_k}^{t_{k+N}} ((\mathbf{y}-\mathbf{y}_{SS})^T \mathbf{P}(\mathbf{y}-\mathbf{y}_{SS}) + (\mathbf{u}-\mathbf{u}_{SS})^T \mathbf{W}(\mathbf{u}-\mathbf{u}_{SS}))dt \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t), \quad \mathbf{y}(t) = h(\mathbf{x}) \\
& \mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}(t) \\
& \mathcal{F}(\mathbf{x},\mathbf{u},t) \leq 0 \\
& \mathcal{G}(\mathbf{x},\mathbf{u},t) = 0
\end{aligned} \tag{3.19}$$

where t_k is the sampling time for the k^{th} sample, and N is the number of samples in the prediction horizon. The subscript SS denotes the steady-state values of the state and manipulated input variables. The matrices $\mathbf{P}$ and $\mathbf{W}$ are positive definite matrices of conforming dimensions used to allocate weights to the state regulation errors and manipulated input variables. The weights are chosen as the inverse of the magnitudes of the input and output variables at steady state to ensure that the contribution of all the manipulated and controlled variable errors in the objective function have comparable orders of magnitudes. The first constraint comes from the plant model (equation (3.17)). The second constraint implements bounds on the manipulated input variables. In a general MPC formulation, the third and fourth constraints are equalities and inequalities associated with the system. In our work, we incorporated these constraints in the model equations, e.g., the temperature dependence of the mass transfer coefficient as shown in equation (3.8) was implicitly

satisfied in the MPC formulation through the model equations. This problem computes the temporal profiles of the piece-wise constant decision variables, which are the manipulated inputs over the prediction horizon $[t_k, t_{k+N}]$ and implements the first of the calculated values (which is a constant over the period $[t_k, t_{k+1})$). The same procedure is repeated for the next prediction horizon $[t_{k+1}, t_{k+1+N}]$.

3.3.2 Distributed MPC

In DMPC multiple local controllers are used. A prerequisite to DMPC is the decomposition of the plant into M interacting subsystems given by:

$$\dot{\mathbf{x}}_m(t) = f_m(\mathbf{x}(t)) + g_m(\mathbf{x}(t))\mathbf{u}_m(t) \quad \forall m \in \{1, M\} \tag{3.20}$$

$$\mathbf{y}_m(t) = h_m(\mathbf{x}) \tag{3.21}$$

where the subscript m is the index of the subsystem. The subsystems comprise the components of the original plant, so that

$$\mathbf{x}^T = [\mathbf{x}_1^T \ \mathbf{x}_2^T \ \dots \ \mathbf{x}_M^T] \tag{3.22}$$

$$\mathbf{u}^T = [\mathbf{u}_1^T \ \mathbf{u}_2^T \ \dots \ \mathbf{u}_M^T] \tag{3.23}$$

and

$$\mathbf{y}^T = [\mathbf{y}_1^T \ \mathbf{y}_2^T \ \dots \ \mathbf{y}_M^T] \tag{3.24}$$

f_m and g_m are the components of f and g respectively which capture the dynamics of $\mathbf{x}_m$. We use a modularity-based decomposition algorithm [40] to obtain the optimal decomposition of the amine gas sweetening plant, which is discussed subsequently in this section.

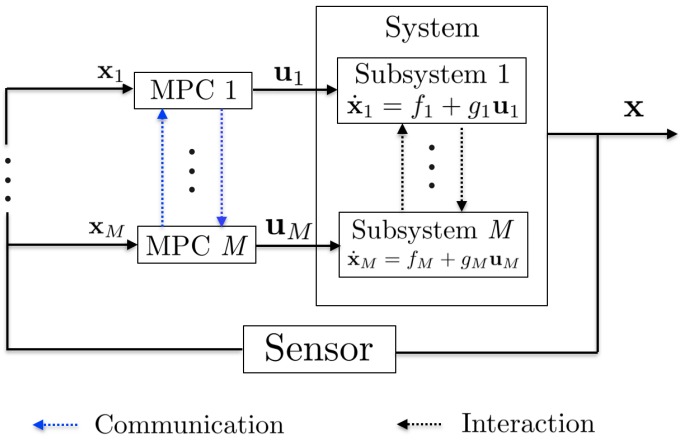


Figure 3.4: Iterative distributed MPC

We have implemented an iterative, non-cooperative DMPC scheme (Figure 3.4) [2], in which the objective function of the local controller of each subsystem includes only the variables associated with that subsystem, and the different controllers 'communicate' the latest values of their individual manipulated inputs to each other to iteratively approach the optimal profiles for all the input variables [2,88]. The optimization problem for a local controller which calculates the optimal inputs for the m^{th} subsystem is given by:

$$\begin{aligned}
\underset{u_{(m)}}{\text{minimize}} \quad & \int_{t_k}^{t_{k+N}} ((\mathbf{y}-\mathbf{y}_{SS})^T\mathbf{P}_{(m)}(\mathbf{y}-\mathbf{y}_{SS}) + (\mathbf{u}-\mathbf{u}_{SS})^T\mathbf{W}_{(m)}(\mathbf{u}-\mathbf{u}_{SS}))dt \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t), \quad \mathbf{y}_m(t) = h_m(\mathbf{x}) \\
& \mathbf{u}_{(m)}^{min} \leq \mathbf{u}_{(m)}(t) \leq \mathbf{u}_{(m)}^{max}(t) \\
& \mathcal{F}_{(m)}(\mathbf{x}_{(m)}, \mathbf{u}_{(m)}, t) \leq 0 \\
& \mathcal{G}_{(m)}(\mathbf{x}_{(m)}, \mathbf{u}_{(m)}, t) = 0
\end{aligned} \tag{3.25}$$

where $\mathbf{P}_{(m)}$ and $\mathbf{W}_{(m)}$ contain the weights associated with the inputs and outputs of the m^{th} subsystem, and $\mathcal{F}_{(m)}$ and $\mathcal{G}_{(m)}$ are the corresponding inequality and equality constraints. Note that the entire model is included as a constraint in all the local controllers' optimization problems, since such a formulation has a better performance as compared to assigning

individual subsystem models as constraints to their corresponding controllers. For DMPC, a crucial step is the decomposition of the network into sub-networks such that the interaction among the sub-networks is minimal. The following section describes the method used to obtain such a decomposition for the amine gas sweetening plant.

3.4 Network Decomposition

The absorber columns are interconnected through the natural gas stream. The first-stage absorber and regenerator column are interconnected with a common solvent stream, as are the second-stage absorber and regenerator columns. So all the columns interact strongly with each other. The natural gas exiting the second-stage absorber column is required to meet the sour gas concentration specifications required for it to be used as a fuel. We select as controlled outputs, the sour gas content in the natural gas streams leaving the first and second absorber columns. We also select the H_2S composition in the solvent stream leaving the first-stage absorber column to ensure that the solvent to natural gas ratio is maintained at a value that effectively removes most of the corrosive sour gas in the first-stage itself. Due to the interconnections among the absorber and regenerator columns, the manipulated inputs available are the flow velocities of the solvent and of steam in the two stages, and the temperature of the feed natural gas. Tables 3.3 and 3.4 list the manipulated inputs and controlled outputs of the plant.

The decomposition of the plant that would result in effective control using DMPC is not obvious due to the interactions among the various process units. The method discussed in the previous chapter [82] for obtaining the optimal decomposition of a network of convection-reaction systems based on the strength of the structural interactions among the variables of the network, captured by an equation graph is applied to obtain the optimal decomposition of the amine gas sweetening plant. Two algorithms are proposed for

Table 3.3: List of inputs

Symbol	Description
u_1	Inlet natural gas temperature
u_2	First-stage solvent flow velocity
u_3	Second-stage solvent flow velocity
u_4	Second-stage steam flow velocity
u_5	First-stage steam flow velocity
u_6	Steam temperature

Table 3.4: List of outputs

Symbol	Description
y_1	CO_2 concentration in the gaseous stream exiting absorber A-01
y_2	H_2S concentration in the gaseous stream exiting absorber A-01
y_3	H_2S concentration in the liquid stream exiting absorber A-01
y_4	CO_2 concentration in the gaseous stream exiting absorber A-02
y_5	H_2S concentration in the gaseous stream exiting absorber A-02

the decomposition, adopted from the corresponding algorithms for ODE system networks. The first algorithm considers the input-output interaction strength as captured by the SIP values, and determines hierarchical decompositions of only the input and the output variables [4]. This algorithm follows a bottom-up approach, that is, starting from a fully decentralized control configuration, the variables are clustered hierarchically to a fully centralized configuration. The optimal decomposition is selected based on the modularity values of the decompositions at various hierarchies [6]. The second algorithm considers the interactions among the input, state and output variables, and determines the optimal decomposition of these variables based on modularity maximization [7]. This algorithm follows a top-down approach, that is, starting from a fully centralized configuration, the clusters of input, output and state variables are bisected until further decomposition does not yield an increase in modularity are found to be less that the previous decomposition.

3.4.1 Input/Output Clustering

Step 1: Structural Interaction Matrix Calculation

We consider a network of interconnected systems modeled by ODEs and first-order hyperbolic PDEs. From the network model, the equation graph of the network is obtained. The shortest path lengths from each input variable to each output variable are calculated from the equation graph using Dijkstra's algorithm, which determines the shortest path between any two vertices of a graph. We form the *structural interaction matrix* (SIM), **M**, as a matrix whose elements, σ_{ij}, are the SIPs (if it is a PDE system) or the relative degrees (if it is an ODE system) of the output y_j with respect to the input u_i.

Step 2: Decentralized Decomposition

An optimal fully decentralized decomposition is obtained by maximizing the value of [4]:

$$J_{DC} = \sum_{i=1}^{n_u} \sum_{j=1}^{n_u} (\sigma_{ij} - \sigma_{ii}) \tag{3.26}$$

i.e. pairing inputs and outputs so that the differences between the off-diagonal and diagonal SIPs are maximized. n_u is the number of input and output variables in the network (assumed equal). We define the structural decoupling matrix (SDM) whose ij^{th} element is given by [6]:

$$s_{ij} = \max \left\{ \sum_{k=1}^{n_u} \sigma_{ik} + \sum_{k=1}^{n_u} \sigma_{kj} - (n_u + n_u)\sigma_{ij}, 0 \right\} \tag{3.27}$$

s_{ij} is a measure of how decoupled an input-output pair is from all other inputs and outputs in the network. Using s_{ij}, the optimization problem can be reformulated as an integer program by re-defining the objective function to be maximized as:

$$J_{DC} = \sum_{i=1}^{n_u} \sum_{j=1}^{n_y} s_{ij} p_{ij} \tag{3.28}$$

where p_{ij} is a binary variable which has the value 1 if u_i and y_j are selected as an input-output pair and the value 0 otherwise. The optimal solution obtained from this integer programing problem yields a $n_u \times n_u$ matrix P_{opt} containing all the p_{ij}'s. The SIM of the optimal decentralized-control decomposition is given by:

$$M_{opt} = P_{opt}^T M \tag{3.29}$$

The diagonal elements of M_{opt} are the SIPs corresponding to each input-output pair. It is possible that there are multiple such optimal decompositions.

In the case where the inputs are more than the outputs ($n_u > n_y$), the above procedure results in a square $n_y \times n_y$ matrix M_{opt}, in which $n_u - n_y$ inputs are not associated with any output and are not considered further.

Step 3: Agglomerative Clustering Procedure

Starting from the optimal decentralized decomposition, clusters of input-output pairs are formed in a hierarchical manner until a single cluster is obtained. This agglomerative clustering procedure is as follows [4]:

- Two input-output pairs, $\{u_i, y_i\}$, $\{u_j, y_j\}$ are structurally close if the $\{u_i, y_j\}$ and $\{u_j, y_i\}$ interactions are comparable to the $\{u_i, y_i\}$, $\{u_j, y_j\}$ interactions (i.e., the off-diagonal SIPs, σ_{ij} and σ_{ji} are comparable to the diagonal SIPs, σ_{ii} and σ_{jj}). The following measures of distance among input-output pairs are considered:

  1. $d_{ij} = \max\{\sigma_{ij}, \sigma_{ji}\} - \sigma_{ii} + \max\{\sigma_{ij}, \sigma_{ji}\} - \sigma_{jj}$
  2. $\delta_{ij} = \min\{\sigma_{ij}, \sigma_{ji}\} - \sigma_{ii} + \min\{\sigma_{ij}, \sigma_{ji}\} - \sigma_{jj}$
  3. $\Delta_{ij} = \max\{\sigma_{ij}, \sigma_{ji}\}$

  d_{ij} captures the largest difference between off-diagonal and diagonal SIPs for the

two input-output pairs. δ_{ij} is the smallest difference between off-diagonal and diagonal SIPs. Δ_{ij} is simply the largest off-diagonal SIP.

- A distance among input-output *clusters* is then defined. The following are the corresponding measures of the distance among clusters:

  1. $d(A,B) = max\{d(a,b) : a \in A, b \in B\}$
  2. $\delta(A,B) = min\{\delta(a,b) : a \in A, b \in B\}$
  3. $\Delta(A,B) = max\{\Delta(a,b) : a \in A, b \in B\}$

  where, A and B are input-output clusters, a and b are individual inputs or outputs in the clusters.

- For clustering the following procedure is followed:

  1. At each level of clustering, compute the matrices $\mathbf{D}_d$, $\mathbf{D}_\delta$ and $\mathbf{D}_\Delta$ for all input-output clusters (or pairs, in the case of the first level of clustering); these matrices have as elements the distance measures d, δ and Δ, respectively, between two clusters.
  2. Determine the pair of input-output clusters for which the distance d is minimum, and merge them.
  3. If there are more than one such pair of clusters, compare the values of δ and Δ of the two input-output clusters, and merge the ones with the minimum distance.
  4. If there are more than one pair of clusters with identical distance measures d, δ and Δ, create separate configurations for further clustering.
  5. Continue 1-4 until a single cluster is obtained.

The algorithms used for the calculation of the SIM and the agglomerative clustering are polynomial and linearithmic time algorithms respectively, and hence scale well with the size of the network [6].

Step 4: Optimal Process Network Decomposition

A notion of modularity [21] is used to compare the process network decompositions at all levels of the hierarchical clustering. A decomposition with a high modularity value is such that the variables within a sub-network have strong interactions among themselves, and weak interactions with the variables outside the sub-network. The decomposition with the highest value of modularity is optimal. The modularity matrix, $\mathcal{B}$ is expressed as [6]:

$$\mathcal{B} = \left\{ b_{ij} | b_{ij} = \frac{\widetilde{m}_{ij}}{\sum_{ij} \widetilde{m}_{ij}} - \frac{\sum_k \widetilde{m}_{ik} \sum_k \widetilde{m}_{kj}}{(\sum_{ij} \widetilde{m}_{ij})^2} \right\} \tag{3.30}$$

where $\widetilde{m}_{ij}$ is the inverse of m_{ij}, the ij^{th} entry in $\mathbf{M}_{opt}$. δ_{ijk} is a binary variable defined as

$$\delta_{ijk} = \begin{cases} 1, & \text{if } u_i \text{ and } y_j \text{ are included in the } k^{th} \text{ cluster} \\ 0, & \text{otherwise} \end{cases} \tag{3.31}$$

The overall modularity of the network with n_k clusters is:

$$Q = \sum_{k=1}^{n_k} \sum_{i,j} \left(\frac{\widetilde{m}_{ij}}{\sum_{ij} \widetilde{m}_{ij}} - \frac{\sum_k \widetilde{m}_{ik} \sum_k \widetilde{m}_{kj}}{(\sum_{ij} \widetilde{m}_{ij})^2} \right) \delta_{ijk} \tag{3.32}$$

The modularities of all decompositions are calculated and the one with the maximum modularity is selected as the optimal.

3.4.2 Modularity Based Decomposition of Equation Graph

The network is modeled as interconnected LPSs and DPSs. From the model, the equation graph of the network is obtained. The equation graph is used to determine the adjacency matrix $\mathcal{A}$ (a matrix such that the element $\mathcal{A}_{ij}$ of the matrix is 1 if there is an edge from the node j to the node i, and 0 otherwise). The value of the modularity is [7, 40]:

$$Q = \frac{1}{4m}\mathbf{s}^T(\mathcal{B} + \mathcal{B}^T)\mathbf{s} \tag{3.33}$$

where $\mathcal{B}$, the modularity matrix, is a matrix whose elements are given by:

$$\mathcal{B}_{ij} = \mathcal{A}_{ij} - \frac{k_i^{in} k_j^{out}}{m} \tag{3.34}$$

where m is the total number of edges in the equation graph of the network, $\mathcal{A}_{ij}$ is the ij^{th} element of $\mathcal{A}$, and k_i^{in} and k_i^{out} are the in-degree and out-degree of the node i respectively. The vector $\mathbf{s}$ is the bisection vector, whose elements are given by:

$$s_i = \begin{cases} 1 & \text{if node } i \text{ belongs to community 1} \\ -1 & \text{if node } i \text{ belongs to community 2} \end{cases} \tag{3.35}$$

In order to determine $\mathbf{s}$ (representing the decomposition) that maximizes the modularity, the following approach is followed [40]:

- Determine the largest positive eigenvalue λ_{max} of $(\mathcal{B} + \mathcal{B}^T)$ and the corresponding eigenvector $\bar{\mathbf{v}}_{max}$.
- Assign the node i to community 1 if the i^{th} element of $\bar{\mathbf{v}}_{max}$ is positive and to community 2 of it is negative.
- Transfer each node to the other community one at a time and check if the modularity

increases by doing so. If it does, update the configuration.

Further bisections of the smaller communities are obtained by maximizing the change in modularity when a community is bisected, which is given by:

$$\Delta Q = \frac{1}{4m}\mathbf{s}^T(\mathcal{B}_{(c)} + \mathcal{B}_{(c)}^T)\mathbf{s} \tag{3.36}$$

where $\mathcal{B}_{(c)}$ is the modularity matrix of the bisected community. The decomposition in which no further community bisection increases the modularity is the optimal one. This algorithm has an average running time of $O(n^2 \log n)$ where n is the total number of nodes in the equation graph [7].

3.4.3 The Optimal Decomposition

All the decompositions obtained from the modularity based algorithm discussed in Section 3.4.2 in which each community has at least as many manipulated input variables as controlled output variables are shown in Figure 3.5.

The optimal decomposition is the one with the maximum modularity value. We obtain the same optimum decomposition from both the aforementioned algorithms. In this decomposition, the variables of the first-stage absorption-regeneration form one community and those of the second stage form the other community. This decomposition is shown in Table 3.5 and in Figure 3.6. In Figure 3.6, the inputs and the outputs and their interconnections with each other and with the state variables are shown in the equation graph. The state variables (shown as the unlabeled nodes on the equation graph) are listed around the corresponding process units in the process flow diagram. The higher modularity value of this decomposition indicates that the interactions among the process units linked with a common solvent stream (the absorber and regenerator columns in each stage) are stronger than the interactions among the units associated with a common natural gas stream like

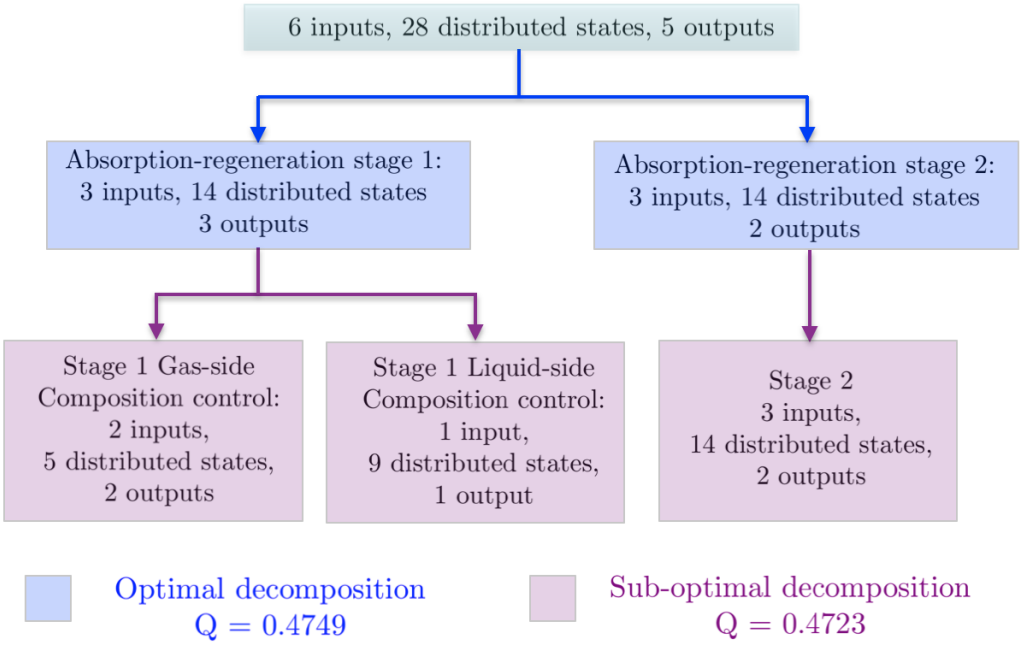


Figure 3.5: Controllable decompositions

the first- and second-stage absorber columns. This is because the sour gas concentrations in all the liquid and gaseous streams in the plant are predominantly influenced by the absorption into and from the solvent stream. Also, the heat released in the solvent stream due to the mass transfer has an impact on the temperatures of the streams in each of the columns.

A sub-optimal decomposition obtained is one in which the community containing the variables associated with the first-stage absorption-regeneration is further bisected to separate the liquid-side composition control problem from the gas-side composition control problem. Since the material (and energy) balances of both the streams in an absorption-regeneration stage are intertwined, the separation of these control problems results in a decomposition which is slightly less than optimal as reflected in the modularity values. The variables of the communities in the sub-optimal decomposition are listed in Table 3.6. Further bisection of the aforementioned communities results in communities which have fewer inputs than outputs, and are thus, not considered.

Table 3.5: Communities in the optimal decomposition

Variables	**Community 1**	**Community 2**
Input	u_1	u_3
	u_2	u_4
	u_5	u_6
State	$C^g_{A_1,CO_2}$	$C^g_{A_2,CO_2}$
	$C^g_{A_1,H_2S}$	$C^g_{A_2,H_2S}$
	$C^l_{A_1,CO_2}$	$C^l_{A_2,CO_2}$
	$C^l_{A_1,H_2S}$	$C^l_{A_2,H_2S}$
	$T^g_{A_1}$	$T^g_{A_2}$
	$T^l_{A_1}$	$T^l_{A_2}$
	$C^g_{S_1,CO_2}$	$C^g_{S_2,CO_2}$
	$C^g_{S_1,H_2S}$	$C^g_{S_2,H_2S}$
	$C^l_{S_1,CO_2}$	$C^l_{S_2,CO_2}$
	$C^l_{S_1,H_2S}$	$C^l_{S_2,H_2S}$
	$T^g_{S_1}$	$T^g_{S_2}$
	$T^l_{S_1}$	$T^l_{S_2}$
	$T^h_{HE_1}$	$T^h_{HE_2}$
	$T^c_{HE_1}$	$T^c_{HE_2}$
Output	y_1	y_4
	y_2	y_5
	y_3	

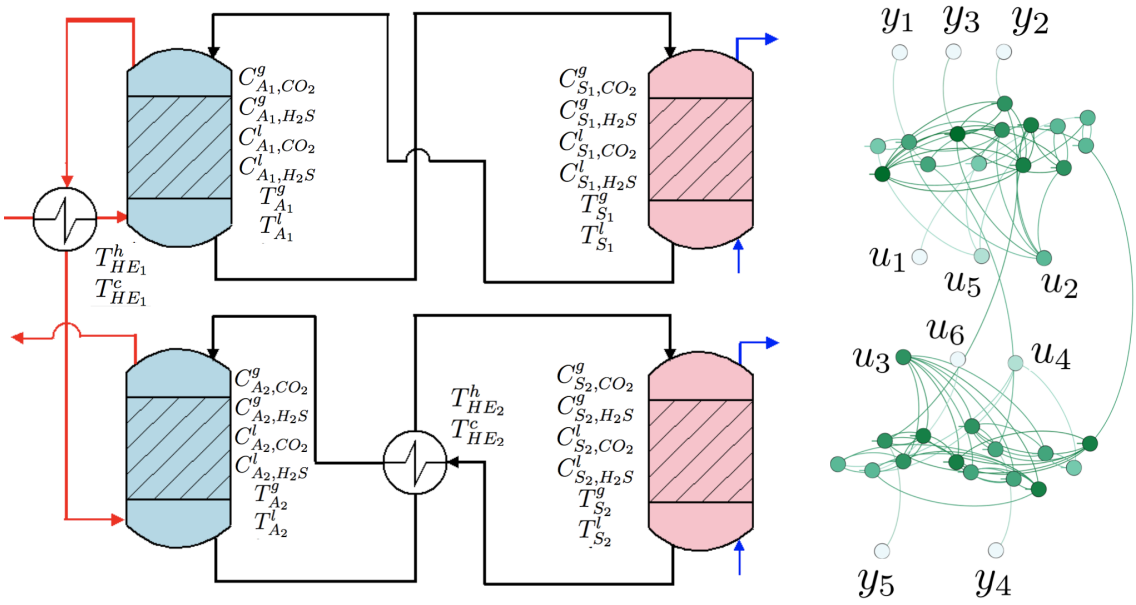


Figure 3.6: Optimal decomposition

Table 3.6: Communities in the sub-optimal decomposition

Variables	Community 1	Community 2	Community 3
Input	u_2 u_5	u_1	u_3 u_4 u_6
State	$C^l_{A_1,H_2S}$ $T^g_{A_1}$ $C^g_{S_1,H_2S}$ $C^l_{S_1,CO_2}$ $T^l_{S_1}$	$C^l_{A_1,CO_2}$ $C^g_{A_1,H_2S}$ $C^l_{A_1,CO_2}$ $T^l_{A_1}$ $C^l_{S_1,CO_2}$ $C^l_{S_1,H_2S}$ $T^g_{S_1}$ $T^h_{HE_1}$ $T^c_{HE_1}$	$C^g_{A_2,CO_2}$ $C^g_{A_2,H_2S}$ $C^l_{A_2,CO_2}$ $C^l_{A_2,H_2S}$ $T^g_{A_2}$ $T^l_{A_2}$ $C^g_{S_2,CO_2}$ $C^g_{S_2,H_2S}$ $C^l_{S_2,CO_2}$ $C^l_{S_2,H_2S}$ $T^g_{S_2}$ $T^l_{S_2}$ $T^h_{HE_2}$ $T^c_{HE_2}$
Output	y_1 y_2	y_3	y_4 y_5

The decompositions are meaningful, and highlight the differences in the degrees of interconnections among process units based on the nature of the shared streams, which are not immediately apparent from the plant model. We use the optimal decomposition for DMPC, and compare the performance with other control architectures, as is discussed subsequently.

3.5 Simulation Results and Discussion

Initially, DMPC was implemented on the optimal decomposition of the amine gas sweetening plant, with a sampling period of 320 seconds, and a prediction horizon of 1600 seconds. The profiles with respect to time of the input and output variables in deviation form are shown in Figure 3.7 and Figure 3.8. The simulations were performed in *MATLAB*® using a 3.4 *GHz Intel*® *Core*^{TM} $i7-6700$ processor. We employed interior point optimization (IPOPT) [89] to solve the underlying nonlinear constrained dynamic optimization problem. To ensure that the optimizing procedure is not terminated before converging to the optimal solution, we consider sufficiently high values of the maximum allowed number of iterations and function evaluations for the IPOPT solver. The indices which are used to evaluate the closed-loop performance of the control architecture are:

$$\text{ISE} = \int_0^{t_f} e^T(t) P e(t) dt \tag{3.37}$$

$$\text{ISC} = \int_0^{t_f} \bar{u}^T(t) W \bar{u}(t) dt \tag{3.38}$$

ISE is the integral square error, that is the deviation of the output variable values from their steady states:

$$e_p = y_p - y_p^{SS} \tag{3.39}$$

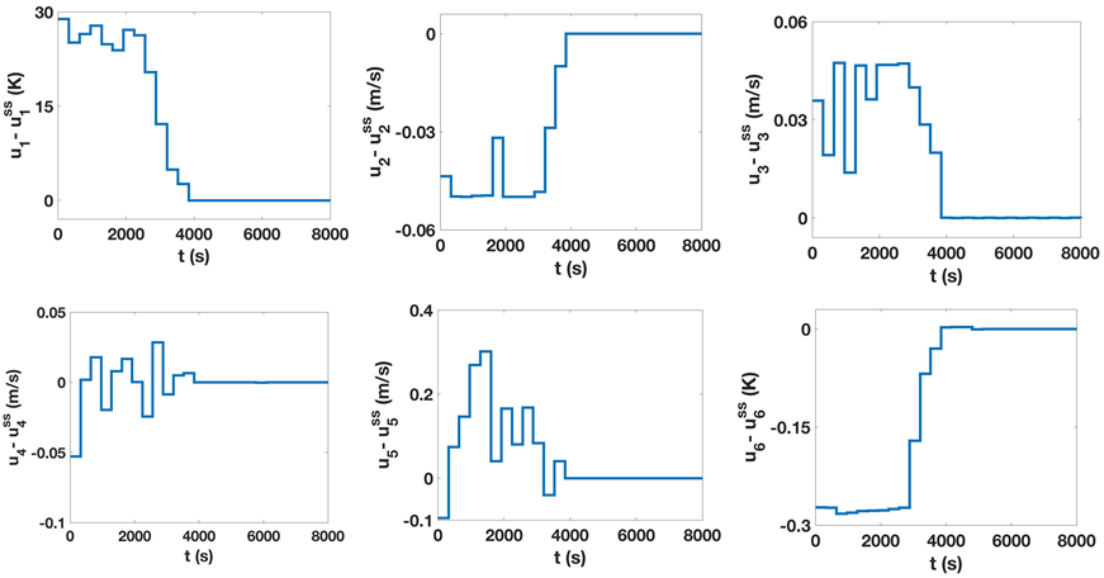


Figure 3.7: Input Profiles

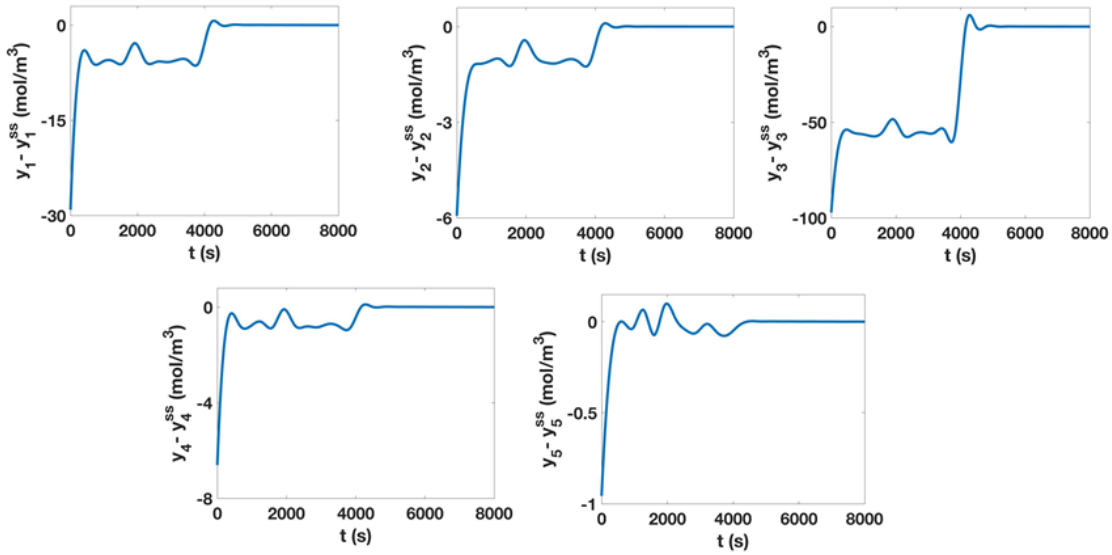


Figure 3.8: Output Profiles

where e_p is the p^{th} element in the vector e, y_p is the p^{th} output variable and y_p^{SS} is its steady state value. *ISC* is the integral square of the control actions from their steady state values:

$$\bar{u}_q = u_q - u_q^{SS} \tag{3.40}$$

where $\bar{u}_q$ is the q^{th} element in the vector $\bar{u}$, u_q is the q^{th} input variable and u_q^{SS} is its steady state value. The sum $ISE + ISC$ is called the dimensionless performance index (DPI), which is included in the objective function minimized in the MPC problem. It represents the regulation and control effort costs for the time interval $[0, t_f]$, and thus, is a meaningful measure of the overall closed-loop performance of the architectures studied for the network control [2, 90]. To gauge the performance of the DMPC, we select the following architectures for comparison:

- Fully centralized MPC: The controller optimizes an objective function that includes all the network variables and constraints.
- Distributed MPC on the sub-optimal decomposition: There are three separate controllers, one each for the following control problems:
    1. Composition control of the gas-side of the first-stage of absorption and regeneration.
    2. Composition control of the liquid-side of the first-stage of absorption and regeneration.
    3. Composition control of the second-stage of absorption and regeneration.
- Decentralized MPC on the optimal decomposition: One controller each is used to control the two stages of absorption and regeneration. There is no communication (information sharing in the form of sharing of the latest input variables for iterative convergence towards the all the optimal input profiles) between the two controllers.

We assess the performance of the aforementioned control architectures for the following control problems:

- *Case I:* Start-up without disturbances or measurement errors
- *Case II:* Start-up with errors in output measurements of the following form:

$$y_m = (1 + w_m)y \tag{3.41}$$

  where y_m is the measured value of output y, and w_m is white noise with zero mean and signal to noise ratio of 20 db.

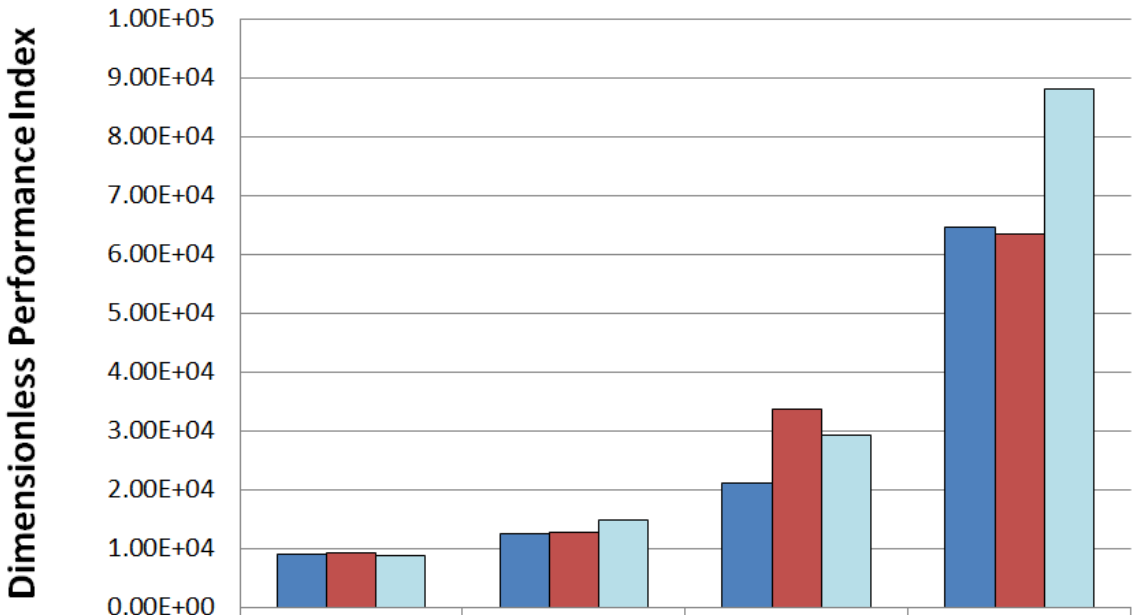
- *Case III:* Start-up with disturbances caused by fluctuations in the feed natural gas composition of the following form:

$$C = (1 + w_c)C_0 \tag{3.42}$$

  where C is the concentration of sour gases in the feed, C_0 the nominal concentration of the sour gases in the feed, and w_c is white noise with zero mean and signal to noise ratio of 40 db.

The performance evaluation indices of the four control architectures for *case I* along with the other cases are shown in Figure 3.9. It can be seen that DMPC implemented on the optimal decomposition has a performance index close to that of Centralized MPC (CMPC), and is better than decentralized MPC as well as DMPC implemented on the sub-optimal decomposition.

The average computation time per sampling period, which is the computation time requirement averaged over all the sampling periods for solving the optimization problem for the four control architectures for all the cases is shown in Figure 3.10. CMPC typically

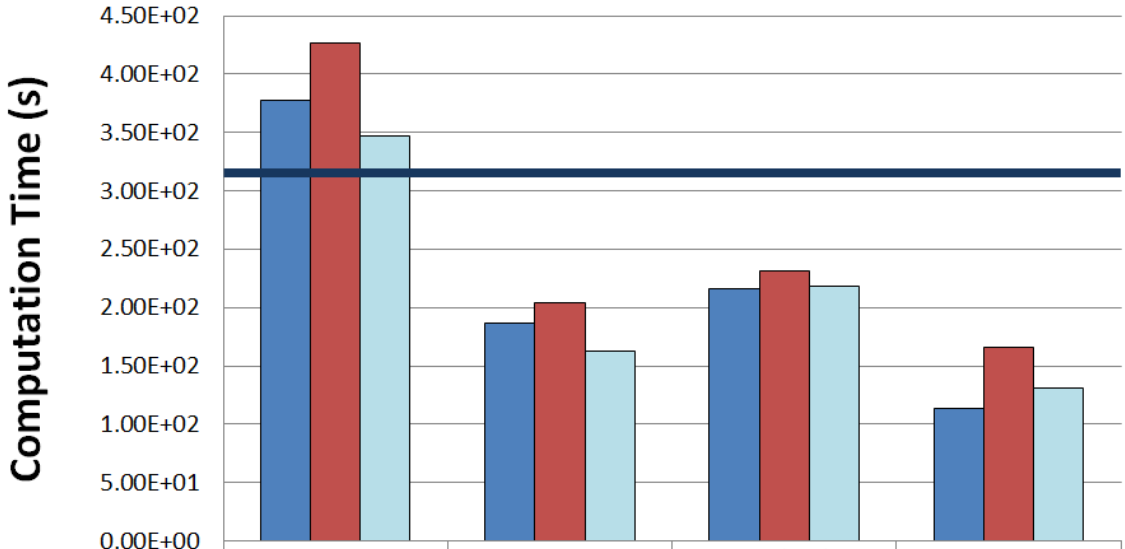


	Centralized	Distributed (Optimal decomposition)	Distributed (Suboptimal decomposition)	Decentralized
Nominal case	8.91E+03	1.25E+04	2.10E+04	6.45E+04
Measurement noise case	9.36E+03	1.26E+04	3.37E+04	6.33E+04
Inlet disturbance case	8.83E+03	1.49E+04	2.93E+04	8.81E+04

Figure 3.9: Performance indices of the four control architectures for the three control problems considered

requires a large computation time since it solves a large-scale dynamic optimization problem. In this case, the computation time required for CMPC is larger than the sampling period itself, which means that CMPC cannot be implemented in real time. By comparing the computational burden of each sampling time for DMPC using optimal and sub-optimal decompositions, we observe that in spite of the smaller size of subsystems, DMPC based on the sub-optimal decomposition requires more iterations at each sampling period which results in a larger computation time. Optimal decomposition-based DMPC has a small enough computational time, with a performance close to that of CMPC, and thus, is the most favorable control architecture.

In Figure 3.9, it is observed that the performance of DMPC based on the optimal decomposition is consistently comparable to that of CMPC, whereas that of the decentralized



	Centralized	Distributed (Optimal decomposition)	Distributed (Suboptimal decomposition)	Decentralized
Nominal case	377.6	186.9	215.8	112.9
Measurement noise case	426.9	203.6	231.5	165.6
Inlet disturbance case	346.8	162.2	218.6	131.1

Figure 3.10: Average computation time per sampling period of the four different control architectures for the control problems, with the horizontal bar showing the sampling period

MPC is consistently the poorest. With the addition of a disturbance in the measurement or the feed composition, the performance of the optimal decomposition-based DMPC is affected only slightly, whereas the performance of the sub-optimal decomposition-based DMPC and decentralized MPC are strongly impacted.

Although CMPC has the best performance index for all three cases, its computational cost renders it impractical. It can be seen from Figure 3.10 that in all the three cases, the computation time required for CMPC is larger than the sampling period itself, as indicated by the black horizontal bar at 320 seconds. Optimal decomposition-based DMPC has a lower computational cost than CMPC as well as the sub-optimal decomposition-based DMPC for all three cases. The decentralized control architecture has the lowest computational cost, but is unfavorable due to its performance index.

We infer the following from these results:

- For the amine gas sweetening plant, a process network with a high degree of interaction (heat and material integration) among separate process units, the implementation of centralized control (CMPC) is impractical due to its high computational cost, and the implementation of decentralized control is ineffective at achieving plant-wide control and rejecting disturbances with minimal control effort as is apparent from its exceedingly high performance index.

- Distributed control, i.e., the implementation of separate controllers for controlling different sub-networks within the network, which communicate with one another to optimize their respective manipulated inputs, is a superior architecture.

- The network decomposition, i.e., the choice of allocation of input and output variables to the controllers used for distributed control has a significant impact on the performance of the DMPC.

- The modularity maximization algorithm is effective in determining the optimal net-

work decompositions for DMPC, as is demonstrated by the fact that DMPC on the sub-optimal decomposition showed worse performance with a higher computational cost as compared to that based on the optimal decomposition.

These results, thus, demonstrate the efficacy of distributed control architectures based on the optimal decomposition obtained by modularity maximization for controlling the amine gas sweetening plant. Motivated by these results, we aimed to incorporate the more generic class of DPSs, the diffusion-convection-reaction systems into the graph based community detection framework. This extension is discussed in the next chapter.

Chapter 4

Structural Interaction Based Equation Graphs of Diffusion - Convection - Reaction Systems *

4.1 Introduction

Since some degree of diffusion is present in all real systems, a generic DPS is, strictly speaking, a diffusion - convection - reaction system, the dynamics of which are generally described by quasi or non-linear parabolic PDEs (PPDEs) [91]. In this chapter we consider diffusion-convection-reaction systems modeled by quasi-linear PPDEs, for which we

- propose an equation graph representation;
- propose the extension of the structural interaction parameter (SIP) definition to quantify the strength of interaction among PPDE system variables;

*Reprinted with permission from Manjiri Moharir, Davood B. Pourkargar, Ali Almansoori and Prodromos Daoutidis, *Chemical Engineering Science*, 204 (2019), https://doi.org/10.1016/j.ces.2018.11.062. Copyright ©2019 Elsevier Ltd.

- establish a relationship between the equation graph and the SIP, which is analogous to that for hyperbolic PDE systems.

This allows us to extend the network theory-based decomposition algorithms to a generic chemical plant which comprises interconnected LPSs and DPSs and determine the optimal network decompositions for plant-wide control.

4.2 Standard Form of Diffusion-Convection-Reaction Systems

The standard form of quasi-linear PPDE systems is [92]:

$$\frac{\partial \bar{\mathbf{x}}}{\partial t} = \mathbf{A}\frac{\partial \bar{\mathbf{x}}}{\partial z} + \mathbf{B}\frac{\partial^2 \bar{\mathbf{x}}}{\partial z^2} + \mathbf{w}u(z,t) + \mathbf{f}(\bar{\mathbf{x}}) \tag{4.1}$$

$$\mathbf{y} = \mathbf{h}(\bar{\mathbf{x}}) \tag{4.2}$$

with the boundary conditions

$$\mathbf{C}_1\bar{\mathbf{x}}(\alpha,t) + \mathbf{D}_1\frac{\partial \bar{\mathbf{x}}}{\partial z}(\alpha,t) = Q_1 \tag{4.3}$$

$$\mathbf{C}_2\bar{\mathbf{x}}(\beta,t) + \mathbf{D}_2\frac{\partial \bar{\mathbf{x}}}{\partial z}(\beta,t) = Q_2 \tag{4.4}$$

and initial condition

$$\bar{\mathbf{x}}(z,0) = \bar{\mathbf{x}}_0(z) \tag{4.5}$$

where $\bar{\mathbf{x}}$ is the $n_s \times 1$ vector of the n_s distributed state variables, $\mathbf{y}$ is the $n_y \times 1$ vector of the n_y output variables, and $u(z,t)$ is the $n_d \times 1$ vector of the n_d distributed input variables, $\mathbf{A}$ and $\mathbf{B}$ are diagonal matrices of conforming dimensions, $\mathbf{w}$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{D}_1$, and $\mathbf{D}_2$ are matrices of conforming dimensions, $\mathbf{f}$, $\mathbf{h}$ and $\bar{\mathbf{x}}_0(z)$ are vectors of functions, α and β are

the boundaries of the spatial domain of the DPS, $z \in \{\alpha, \beta\}$ is the axial coordinate, $t \in \mathbb{R}$ is the time, and Q_1 and Q_2 are vectors. An alternate representation of the equations is one in which Q_1 and Q_2 from equations (4.3) and (4.4) are incorporated into the differential equations as shown below [41]:

$$\frac{\partial \bar{\mathbf{x}}}{\partial t} = \mathbf{A}\frac{\partial \bar{\mathbf{x}}}{\partial z} + \mathbf{B}\frac{\partial^2 \bar{\mathbf{x}}}{\partial z^2} + \mathbf{w}u(z,t) + \mathbf{f}(\bar{\mathbf{x}}) + \delta(z-\alpha)Q_1 + \delta(z-\beta)Q_2 \tag{4.6}$$

where δ is the Dirac delta function. This is done in order to homogenize the boundary conditions into the following form:

$$\mathbf{C}_1\bar{\mathbf{x}}(\alpha,t) + \mathbf{D}_1\frac{\partial \bar{\mathbf{x}}}{\partial z}(\alpha,t) = 0 \tag{4.7}$$

$$\mathbf{C}_2\bar{\mathbf{x}}(\beta,t) + \mathbf{D}_2\frac{\partial \bar{\mathbf{x}}}{\partial z}(\beta,t) = 0 \tag{4.8}$$

The homogenization aids in the equation graph representation of this system which will be discussed subsequently.

4.3 Reduced Order Representation

The dynamics of the system described by the non-linear PPDEs of the form in equation (4.6) can be approximated using Galerkin's method by considering the eigenproblem corresponding to the operator in the k^{th} PPDE given by

$$\mathbf{A}_k\frac{\partial \phi_i^{(k)}(z)}{\partial z} + \mathbf{B}_k\frac{\partial^2 \phi_i^{(k)}(z)}{\partial z^2} = \lambda_i^{(k)}\phi_i^{(k)}(z)$$

$$k \in \{1, 2, \ldots, n_s\} \tag{4.9}$$

coupled with the homogenized boundary conditions of the k^{th} PPDE. In equation (4.9), $\mathbf{A}_k$ and $\mathbf{B}_k$ are the diagonal values in the k^{th} rows of $\mathbf{A}$ and $\mathbf{B}$ of equation (4.6). This eigenproblem has infinitely many solutions $\{\lambda_i^{(k)}, \phi_i^{(k)}(z)\}\ \forall i \in \{1, \infty\}$ where $\lambda_i^{(k)}$ are the eigenvalues corresponding to the eigenfunctions $\phi_i^{(k)}(z)$. A finite number of the eigenfunctions can be used to approximate the dynamics of the PPDE system; these are determined on the basis of the real parts of their corresponding eigenvalues [92]. If the eigenvalues are arranged in the decreasing order of their real parts as

$$Re(\lambda_1^{(k)}) \geq Re(\lambda_2^{(k)}) \geq \cdots \geq Re(\lambda_{N_s}^{(k)}) >> Re(\lambda_{N_s+1}^{(k)}) \geq \ldots \tag{4.10}$$

where $Re(\lambda_{N_s}^{(k)}) < 0$, then, the dynamics of the PPDE can be approximated using N_s of its eigenfunctions corresponding to the eigenvalues $\{\lambda_1^{(k)}, \lambda_2^{(k)}, \ldots, \lambda_{N_s}^{(k)}\}$, as

$$\frac{d\mathbf{a}_s^k}{dt} = \mathcal{A}_s^k \mathbf{a}_s^k + \mathcal{B}_s^k \mathbf{u} + f_s^k(\mathbf{a}_s) \tag{4.11}$$

where

$$\mathbf{a}_s^k = \mathcal{P}_s x_k \tag{4.12}$$

$$\mathcal{A}_s^k = \mathcal{P}_s \left(\mathbf{A}_k \frac{\partial}{\partial z} + \mathbf{B}_k \frac{\partial^2}{\partial z^2} \right) \tag{4.13}$$

$$\mathcal{B}_s^k = \mathcal{P}_s w^k \tag{4.14}$$

$$f_s^k = \mathcal{P}_s \left(f^k + \sum_i \delta(z - \zeta_i) Q_{\zeta_i} \right) \tag{4.15}$$

w^k and f^k are the k^{th} rows of w and f in equation (4.6), $\zeta_i \in \{\alpha, \beta\}$ represents the boundary coordinates, Q_{ζ_i} represents the corresponding boundary condition, $\mathcal{A}_s^k$ is a diagonal ma-

trix containing the N_s eigenvalues $(\lambda_k^{(1)}, \lambda_k^{(2)}, \ldots, \lambda_k^{(N_s)})$, and $\mathcal{P}_s$ is the orthogonal projector to the sub-space spanned by the N_s corresponding eigenfunctions. The distributed state variable $x_k(z,t) \in \bar{\mathbf{x}}$ is then approximated using the time-dependent variables $a_{s_i}^k(t)$ as:

$$x_k \approx \sum_{i=1}^{N_s} a_{s_i}^k(t)\phi_i^k(z) \tag{4.16}$$

4.4 Graph-based Network Decomposition

4.4.1 Classification of Variables

The inputs to a PPDE system can be classified as:

- Distributed inputs ($u(z,t)$): The input variable is a function of space and time.
- Boundary input ($\bar{\mathbf{x}}(\zeta,t)$): The boundary conditions of the variables of the system.
- Velocity inputs ($\mathbf{A}$): The convective flow velocities of the system.

The outputs of the system can be classified as:

- Distributed outputs: The outputs are profiles in space and time, i.e.

$$y_j = x_i(z,t)$$

$$i \in \{1, n_s\},\ j \in \{1, n_y\} \tag{4.17}$$

- Boundary outputs: The outputs are the values of state variables at the boundary of

the system, i.e.,

$$y_j = x_i(\zeta, t)$$

$$\zeta \in \{\alpha, \beta\},\ i \in \{1, n_s\},\ j \in \{1, n_y\} \tag{4.18}$$

- Spatially varying time-dependent outputs: The outputs are the values of one or more state variables at a spatial coordinate that varies with time, i.e.

$$y_j = x_i(z_{y_j}(t), t)$$

$$i \in \{1, n_s\},\ j \in \{1, n_y\} \tag{4.19}$$

  where z_{y_j} is a time-dependent spatial coordinate at which the output of interest is located. An example of this kind of output is the hot-spot temperature in a reactor, which is usually a function of other process parameters such as the inlet composition (boundary input) or flow velocity (velocity input).

Distributed inputs can be captured by the values of the finite number of actuators located through the spatial domain in a vector, $\mathbf{u}(t) = [u_1(t), u_2(t), \ldots, u_{n_a}(t)]^T$ (n_a is the number of actuators) with a finite number of elements which are functions of time alone [18]. Then in the model equations $\mathbf{u}(t)$ can be incorporated as:

$$\mathbf{w}u(z,t) = \mathbf{w}b(z)\mathbf{u}(t) \tag{4.20}$$

where $b(z) = [b_1(z), b_2(z), \ldots, b_{n_a}(z)]$ captures the distribution of each element of $\mathbf{u}(t)$ on its respective spatial interval [18]. A distributed output $y_j(z,t)$ can also be represented by a finite dimensional vector of N_{y_j} elements, the p^{th} element of which is a function of time

alone [18] and given by

$$y_{j_p}(t) = \int_{z_p}^{z_p+dz} c_p(z) y_j(z,t) dz \tag{4.21}$$

where c_p is a smooth shaping function on the interval $[z_p, z_p + dz]$ $\forall p \in \{1, N_{y_j}\}$. We can then define a generic modified output to be of the form:

$$\hat{y}_j(t) = h_j(\hat{\mathbf{x}}) \tag{4.22}$$

$$j \in \{1, n_y\} \tag{4.23}$$

$\hat{\mathbf{x}}$ is the vector of modified state variables, whose i^{th} element is given by

$$\hat{x}_i = \int_{\alpha}^{\beta} \gamma_i(z) x_i(z,t) dz \tag{4.24}$$

$$i \in \{1, n_s\} \tag{4.25}$$

γ_i is a piece-wise continuous function on the spatial domain $[\alpha, \beta]$ such that:

- For distributed outputs, γ_i has the same profile as all the c_i in equation (4.21).
- For boundary outputs, γ_i has a value close to 1 at the boundary where the output lies (α or β), a small non-zero value at the other boundary, and zero everywhere else.
- For the third kind of outputs (equation (4.19)), the function γ_i takes a value close to 1 in the range of z_{y_j} and a small non-zero value at the boundary where the inlet conditions are incorporated using the Dirac delta function in equation (4.6).

The small non-zero value of γ_i in the case of boundary and spatially varying outputs is to ensure that the input variables appear explicitly in a finite-order derivative of the modified

state variable on which the output is dependent as per equation (4.22) if the input variables have an effect on the PDE corresponding to the original state variable. In a sense, this modification 'collocates' the input and output variables and aids in capturing the dependency of the output variable's derivatives on the input variable [82].

4.4.2 Graph Representation

The goal is to capture on an equation graph, the relationships of the input, output and state variables of a PPDE system in a form that is analogous to the existing graph representations so that the network theory-based algorithms can be extended to chemical networks that include DPSs described by PPDEs.

The proposed graph representation of the PPDEs in equation (4.6) for outputs defined in equation (4.22) and velocity, boundary or distributed manipulation (equation (4.20)) is as follows:

1. There is a node representing every input variable, distributed state variable and output variable.

2. There is an edge from the input variable to a state variable if the input variable appears in the PDE that captures the state variable dynamics.

3. There is an edge from a state variable x_i to x_j if x_i appears in the PDE corresponding to x_j.

4. There is an edge from a state variable to an output variable if the output is related to the corresponding modified state variable as per equation (4.22).

This equation graph captures the strength of the interaction among the variables of a PPDE system as is discussed subsequently.

4.4.3 Structural Interaction Parameter

We define a parameter called the SIP as a relative degree analogue for the variables associated with PPDEs. The relative degree of an output with respect to an input for LPSs is defined as the smallest order derivative of the output which shows an explicit dependence on the input. On similar lines, the SIP of an output with respect to an input for a PPDE system is the smallest order derivative of the output which is explicitly dependent on the input variable (which is possible since the input and output variables are functions of time alone, as described earlier). For this definition of the SIP and the aforementioned equation graph representation, we establish the following theorem:

Theorem 3: In the equation graph of a PPDE system, if l is the length of the shortest path from an input node to an output node and s is the SIP between the two variables, then,

$$\mathrm{s} = l - 1 \tag{4.26}$$

The proof of this theorem is discussed in Appendix C. We demonstrate the application of Theorem 3 through the following example.

Example 1: Consider a tubular reactor with two reactants participating in an irreversible reaction:

$$R_1 + R_2 \rightarrow \text{product} \tag{4.27}$$

The concentrations of R_1 and R_2 at any axial coordinate $z \in [0,1]$ (where 1 is the normalized length of the reactor) at time t are $x_1(z,t)$ and $x_2(z,t)$ respectively. The reaction is first-order in both the reactants. The rate constant, which follows the Arrhenius Law, is $k = k_0 e^{-E_a/RT}$, where k_0 is the pre-exponential factor, E_a is the activation energy, R is the universal gas constant and T is stream temperature. The model equations of the tubular

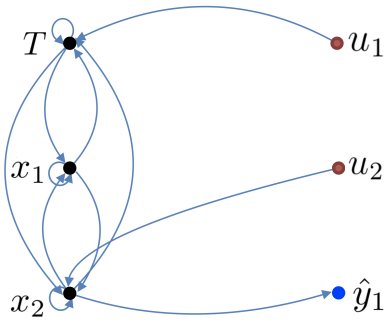


Figure 4.1: Equation graph of the tubular reactor

reactor are

$$\frac{\partial x_1}{\partial t} = -v\frac{\partial x_1}{\partial z} + \mathcal{D}_1\frac{\partial^2 x_1}{\partial z^2} - kx_1x_2 + \delta v x_1(0,t) \tag{4.28}$$

$$\frac{\partial x_2}{\partial t} = -v\frac{\partial x_2}{\partial z} + \mathcal{D}_2\frac{\partial^2 x_2}{\partial z^2} - kx_1x_2 + \delta v x_2(0,t) \tag{4.29}$$

$$\frac{\partial T}{\partial t} = -v\frac{\partial T}{\partial z} + \frac{k}{\rho C_p}\frac{\partial^2 T}{\partial z^2} + k_1x_1x_2\frac{-\Delta H}{\rho C_p} + \delta v T(0,t) \tag{4.30}$$

where v is the convective velocity, $\mathcal{D}_1$ and $\mathcal{D}_2$ are the mass diffusivities of R_1 and R_2 respectively, k is the thermal conductivity of the stream, ρ is its mass density and C_p is its specific heat capacity, and $-\Delta H$ is the reaction enthalpy. The boundary conditions, $x_1(0,t)$, $x_2(0,t)$ and $T(0,t)$ are incorporated in the equations using the Dirac delta function δ. The two inputs, u_1 and u_2 to this reactor are the inlet temperature and inlet concentration of R_2, while the controlled variable, y_1, of the reactor is the exit concentration of R_2. Thus,

$$u_1 = T(0,t)$$

$$u_2 = x_2(0,t)$$

$$y_1 = x_2(1,t)$$

The equation graph of this system of PPDEs is given in figure 4.1. From the equation graph we can determine the length of the shortest path from u_1 to $\hat{y}_1$ to be 3 and that from

u_2 to $\hat{y}_1$ to be 2.

The modified state variable $\hat{x}_2$ is defined as

$$\hat{x}_2(t) = \int_0^1 \gamma(z) x_2(z,t) dz \tag{4.31}$$

where γ has a value close to 1 at $z = 1$, a small non-zero value at $z = 0$ and a value of 0 for all $z \in (0,1)$. The modified output, then, is $\hat{y}_1 = \hat{x}_2$. The first derivative with respect to time of $\hat{y}_1$ is

$$\frac{d\hat{y}_1}{dt} = \int_0^1 \gamma(z) \frac{\partial x_2(z,t)}{\partial t} dz \tag{4.32}$$

Substituting for $\frac{\partial x_2(z,t)}{\partial t}$ from equation (4.29), we get

$$\frac{d\hat{y}_1}{dt} = \int_0^1 \gamma(z) \left(-v \frac{\partial x_2}{\partial z} + \mathcal{D}_2 \frac{\partial^2 x_2}{\partial z^2} - k x_1 x_2 + \delta v u_2 \right) dz \tag{4.33}$$

From equation (4.33), we can observe that the first derivative of $\hat{y}_1$ has an explicit dependence on u_2 since

$$\int_0^1 \gamma(z) \delta v u_2 dz \neq 0 \tag{4.34}$$

Hence, the SIP of y_1 with respect to u_2 is 1.

For the second derivative with respect to time of $\hat{y}_1$, we use the product rule of differentiation on the term $k x_1 x_2$ in equation (4.33). Since k is a function of temperature, the second derivative of $\hat{y}_1$ depends on the term

$$\int_0^1 \gamma(z) \frac{\partial T}{\partial t} dz \tag{4.35}$$

Substituting for $\partial T / \partial t$ in equation (4.35) from equation (4.30), and noting that

$$\int_0^1 \gamma(z) \delta v u_1 dz \neq 0 \tag{4.36}$$

we can conclude that $d^2\hat{y}_1/dt^2$ explicitly depends on u_1. So the SIP of y_1 with respect to u_1 is 2.

Hence, the relationship between the SIP, s, and the length of the shortest path from the input to the output node, l, as established in Theorem 3 does indeed hold.

Theorem 3 relates the equation graph representation of PPDE system variables to the SIP, which captures the interaction between an input and an output variable along with the intermediate state variables. This relationship enables us to use the edges connecting the nodes in the equation graph of the PPDE system to identify strongly and weakly interacting subsystems by implementing network theory based algorithms in a form analogous to that for ODEs and hyperbolic PDEs [10, 82]. As a result of this analogy, we can represent on a common equation graph all the variables corresponding to a generic chemical plant that includes LPSs and DPSs, and determine the network decomposition for plant-wide distributed control.

4.5 Conclusions

This chapter presents a generically applicable equation graph-based algorithm to obtain optimal decompositions for a chemical plant comprising diffusion-convection-reaction systems described by parabolic PDEs, which are a broad class of commonly observed DPSs in chemical plants. We defined a Structural Interaction Parameter (SIP), which is a measure of the strength of structural interactions among all the types of input and output variables in such a system, analogous to relative degree for ODE systems. We proposed equation graph representations of the variables of a parabolic PDE system and related the lengths of input-output paths to the SIP. These equation graphs, and their relationship to the SIP closely resemble those for ODE systems, and thus facilitate the representation of all the variables of a chemical network and their interactions on a common equation graph.

These equation graphs allow the use of network theory-based algorithms for the identification of strongly interacting sub-networks within a chemical network for distributed control.

In the following chapter, we implement the proposed approach on a chemical plant consisting of four tubular reactors in series with interstage cooling achieved by three heat exchangers and followed by two flash columns. We determine the optimal network decomposition based on the network's equation graph and use that as the basis for DMPC. We test the performance of DMPC on the optimal decomposition for the set-point tracking problem.

Chapter 5

Distributed Control of a Reaction-Separation Network *

5.1 Introduction

Following from the graph representation and its relationship with the SIP as discussed in the previous chapter [93], to test the efficacy of the graph based network decomposition algorithm for process networks that include diffusion-convection-reaction systems, we consider as a case study a plant designed for the production and separation of a component B from component A. A is converted to B via a reversible, elementary reaction:

$$A \underset{k_1}{\overset{k_2}{\rightleftharpoons}} B \tag{5.1}$$

where k_1 and k_2 are the rate constants of the forward and backward reaction respectively, and follow the Arrhenius law. The reaction is carried out in four tubular reactors (R-1, R-2, R-3, R-4). The forward reaction ($A \rightarrow B$) is exothermic, and so an increase in temperature

*Reprinted with permission from Manjiri Moharir, Davood B. Pourkargar, Ali Almansoori and Prodromos Daoutidis, *Chemical Engineering Science*, 204 (2019), https://doi.org/10.1016/j.ces.2018.11.062. Copyright ©2019 Elsevier Ltd.

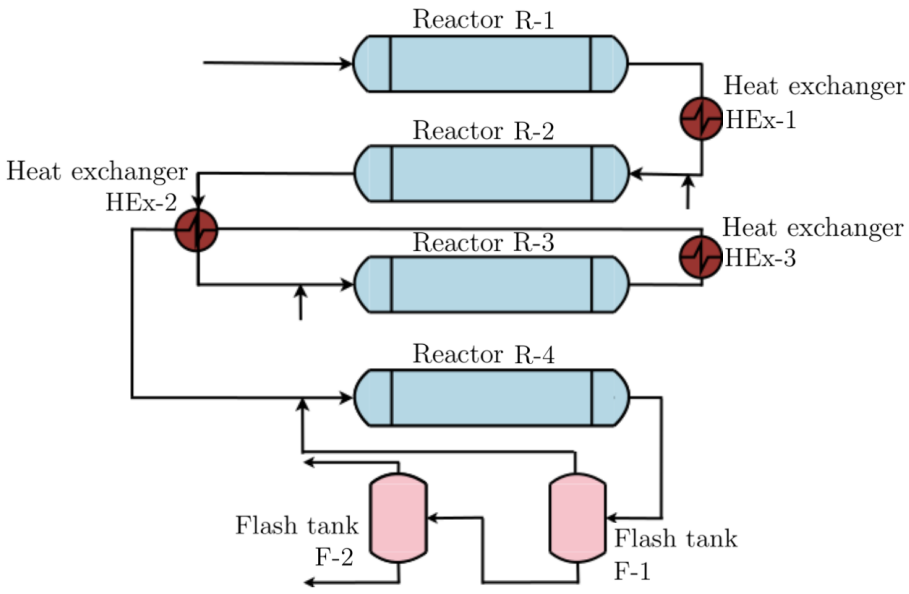


Figure 5.1: Prototypical reaction-separation plant

drives the equilibrium towards the reactant. To drive the equilibrium towards the product, three interstage coolers (HEx-1, HEx-2, HEx-3) and two additional feed streams to R-2 and R-3 are used [22]. The separation of A and B and the purification of B is achieved with the help of two flash tanks (F-1, F-2). The distillate of the F-1, which is rich in A is recycled to R-4. Cooling water is used to cool the process streams in HEx-1 and HEx-3. The cooled stream of HEx-3 is used to cool the hotter stream exiting R-2 in HEx-2. The plant is shown in figure 5.1. These types of plants are encountered in the production and separation of sulphuric acid [22]. The plant-wide control of such plants is challenging due to the high sensitivity of the conversion to the process parameters like temperature and reactant concentration, as well as the tight integration among the different process systems in the plant. For this network, we implement DMPC and compare its performance for the set-point tracking problem against that of CMPC. The manipulated and controlled variables of the plant are listed in tables 5.1 and 5.2.

Table 5.1: Manipulated variables

Symbol	Variable
u_1	Feed inlet temperature
u_2	Cooling fluid temperature to HEx-1
u_3	Cooling fluid temperature to HEx-3
u_4	Heat input to F-1
u_5	Heat input to F-2
u_6	Recycle ratio to R-4
u_7	Fresh feed flow-rate to R-2
u_8	Fresh feed flow-rate to R-3

Table 5.2: Controlled variables

Symbol	Variable
y_1	Inlet temperature of R-2
y_2	Inlet temperature of R-3
y_3	Inlet temperature of R-4
y_4	Concentration of A exiting R-1
y_5	Concentration of A exiting R-2
y_6	Concentration of A exiting R-3
y_7	Concentration of A exiting R-4
y_8	Concentration of B in liquid stream exiting F-2

5.2 Plant Model

5.2.1 Tubular Reactor Model

The PPDEs describing the dynamics of each of the tubular reactors are [94]:

$$\frac{\partial T^{(i)}}{\partial \bar{t}} = -v^{(i)}\frac{\partial T^{(i)}}{\partial \bar{z}} + \frac{\mathcal{K}}{\rho C_p}\frac{\partial^2 T^{(i)}}{\partial \bar{z}^2} + \frac{1}{\rho C_p}\{\Delta H_A(k_1 C_A^{(i)} - k_2 C_B^{(i)})\} + v^{(i)}\delta(0)T_f^{(i)} \tag{5.2}$$

$$\frac{\partial C_A^{(i)}}{\partial \bar{t}} = -v^{(i)}\frac{\partial C_A^{(i)}}{\partial \bar{z}} + \mathcal{D}_A\frac{\partial^2 C_A^{(i)}}{\partial \bar{z}^2} - k_1 C_A^{(i)} + k_2 C_B^{(i)} + v^{(i)}\delta(0)C_{Af}^{(i)} \tag{5.3}$$

where T is the temperature, C is the concentration, v is the convective flow velocity, $\mathcal{D}$ is the mass diffusivity, $\mathcal{K}$ is the heat conductivity, ρ is the mass density, C_p is the specific heat capacity, ΔH_A is the heat of the forward reaction. The subscripts A and B denote the components. The reaction rate constants follow an exponential temperature dependence as per the Arrhenius law given by

$$k_1 = k_1^0 \exp\left(-\frac{E_{a1}}{RT}\right) \tag{5.4}$$

$$k_2 = k_2^0 \exp\left(-\frac{E_{a2}}{RT}\right) \tag{5.5}$$

where k_1^0 and k_2^0 are the pre-exponential factors, E_{a1} and E_{a2} are the activation energies for the forward and backward reaction respectively. Constant molar density through the reactor implies

$$C_B^{(i)} = C_A^{0(i)} - C_A^{(i)} \tag{5.6}$$

where $C_A^{0(i)}$ is the concentration of the reactant in the fresh feed, which contains no product. We use the following dimensionless variables:

$$t = \frac{\bar{t}v^{(1)}}{L} \quad z = \frac{\bar{z}}{L} \quad Pe_1^{(i)} = \frac{\rho C_p v^{(i)} L}{\mathcal{K}} \quad Pe_2^{(i)} = \frac{v^{(i)}L}{\mathcal{D}_A}$$

$$x_1 = \frac{T - T_0}{T_0} \quad x_2 = \frac{C_A - C_{A0}}{C_{A0}} \quad x_{1f} = \frac{T_f - T_0}{T_0} \quad x_{2f} = \frac{C_{Af} - C_{A0}}{C_{A0}}$$

$$\gamma_1 = \frac{E_{a1}}{RT_0} \quad \gamma_2 = \frac{E_{a2}}{RT_0} \quad B_{C_1}^{(i)} = \frac{k_1^0 \exp(\frac{-E_{a1}}{RT_0})L}{v^{(i)}} \quad B_{C_2}^{(i)} = \frac{k_2^0 \exp(\frac{-E_{a2}}{RT_0})L}{v^{(i)}}$$

$$B_T = \frac{-\Delta H C_{A0}}{\rho C_p T_0}$$

to non-dimensionalize the PPDEs into the following form [94]:

$$\frac{\partial x_1^{(i)}}{\partial t} = -\frac{\partial x_1^{(i)}}{\partial z} + \frac{1}{Pe_1^{(i)}} \frac{\partial^2 x_1^{(i)}}{\partial z^2} + B_T \left[B_{c_1}^{(i)} \exp\left(\frac{\gamma_1 x_1^{(i)}}{1 + x_1^{(i)}} \right) (1 + x_2^{(i)}) + B_{c_2}^{(i)} \exp\left(\frac{\gamma_2 x_1^{(i)}}{1 + x_1^{(i)}} \right) (x_2^{(i)}) \right]$$

$$+ \delta(0) x_{1f}^{(i)} \tag{5.7}$$

$$\frac{\partial x_2^{(i)}}{\partial t} = -\frac{\partial x_2^{(i)}}{\partial z} + \frac{1}{Pe_2^{(i)}} \frac{\partial^2 x_2^{(i)}}{\partial z^2} - B_{c_1}^{(i)} \exp\left(\frac{\gamma_1 x_1^{(i)}}{1 + x_1^{(i)}} \right) (1 + x_2^{(i)}) - B_{c_2}^{(i)} \exp\left(\frac{\gamma_2 x_1^{(i)}}{1 + x_1^{(i)}} \right) (x_2^{(i)})$$

$$+ \delta(0) x_{2f}^{(i)} \tag{5.8}$$

In this form the boundary conditions are

$$\frac{\partial x_1^{(i)}}{\partial z}(0) = Pe_1^{(i)} x_1^{(i)} \tag{5.9}$$

$$\frac{\partial x_2^{(i)}}{\partial z}(0) = Pe_2^{(i)} x_2^{(i)} \tag{5.10}$$

$$\frac{\partial x_1^{(i)}}{\partial z}(1) = 0 \tag{5.11}$$

$$\frac{\partial x_2^{(i)}}{\partial z}(1) = 0 \tag{5.12}$$

Using Galerkin's method, the distributed concentration variables are approximated as:

$$x_1^{(i)} \approx \sum_{l=1}^{N_s} t_l(t) \Gamma_l(z) \tag{5.13}$$

$$x_2^{(i)} \approx \sum_{l=1}^{N_s} a_l(t)\Phi_l(z) \tag{5.14}$$

where N_s is the number of dominant modes (assumed to be identical for both the state variables in a reactor), t_l and a_l are the time-dependent coefficients, and Γ_l and Φ_l are the eigenfunctions corresponding to spatial differential operators of the state variables.

Then, the differential equations describing the dynamics of a_k and t_k for $k \in [1, N_s]$ are given by:

$$\begin{aligned}
\frac{dt_k}{dt} = &-\sum_{l=1}^{N_s}\left(t_l\int_0^1 \frac{d\Gamma_l}{dz}\Gamma_k^* dz\right) + \frac{1}{Pe_1^{(i)}}\sum_{l=1}^{N_s}\left(t_l\int_0^1 \frac{\partial^2\Gamma_l}{\partial z^2}\Gamma_k^* dz\right) \\
&+B_T B_{c_1}^{(i)}\int_0^1 \exp\left(\frac{\gamma_1\sum_{l=1}^{N_s} t_l(t)\Gamma_l(z)}{1+\sum_{l=1}^{N_s} t_l(t)\Gamma_l(z)}\right)\left(1+\sum_{l=1}^{N_s} a_l(t)\Phi_l(z)\right)\Gamma_k^* dz \\
&+B_T B_{c_2}^{(i)}\int_0^1 \exp\left(\frac{\gamma_2\sum_{l=1}^{N_s} t_l(t)\Gamma_l(z)}{1+\sum_{l=1}^{N_s} t_l(t)\Gamma_l(z)}\right)\left(\sum_{l=1}^{N_s} a_l(t)\Phi_l(z)\right)\Gamma_k^* dz + x_{1f}\Gamma_k^*(0)
\end{aligned} \tag{5.15}$$

$$\begin{aligned}
\frac{da_k}{dt} = &-\sum_{l=1}^{N_s}\left(a_l\int_0^1 \frac{d\Phi_l}{dz}\Phi_k^* dz\right) + \frac{1}{Pe_2^{(i)}}\sum_{l=1}^{N_s}\left(a_l\int_0^1 \frac{\partial^2\Phi_l}{\partial z^2}\Phi_k^* dz\right) \\
&-B_{c_1}^{(i)}\int_0^1 \exp\left(\frac{\gamma_1\sum_{l=1}^{N_s} t_l(t)\Gamma_l(z)}{1+\sum_{l=1}^{N_s} t_l(t)\Gamma_l(z)}\right)\left(1+\sum_{l=1}^{N_s} a_l(t)\Phi_l(z)\right)\Phi_k^* dz \\
&-B_{c_2}^{(i)}\int_0^1 \exp\left(\frac{\gamma_2\sum_{l=1}^{N_s} t_l(t)\Gamma_l(z)}{1+\sum_{l=1}^{N_s} t_l(t)\Gamma_l(z)}\right)\left(\sum_{l=1}^{N_s} a_l(t)\Phi_l(z)\right)\Phi_k^* dz + x_{2f}\Phi_k^*(0)
\end{aligned} \tag{5.16}$$

5.2.2 Flash Tank Model

In the flash columns, we assume that vapor-liquid equilibrium is established instantaneously and that the feed is in the liquid phase. Hence, the model equations are [95]:

$$\frac{dV^{(j)}}{d\bar{t}} = F_f^{(j)} - F_l^{(j)} - F_v^{(j)} \tag{5.17}$$

$$\frac{d(V^{(j)}C_{A_l}^{(j)})}{d\bar{t}} = F_f^{(j)}C_{A_f}^{(j)} - F_l^{(j)}C_{A_l}^{(j)} - F_v^{(j)}C_{A_v}^{(j)} \tag{5.18}$$

$$\frac{d(V^{(j)}C_{B_l}^{(j)})}{d\bar{t}} = F_f^{(j)}C_{B_f}^{(j)} - F_l^{(j)}C_{B_l}^{(j)} - F_v^{(j)}C_{B_v}^{(j)} \tag{5.19}$$

$$\frac{d(V^{(j)}T_l^{(j)})}{d\bar{t}} = F_f^{(j)}T_f^{(j)} - F_l^{(j)}T_l^{(j)} - F_v^{(j)}T_v^{(j)}\frac{\rho_v C_{p_v}}{\rho_l C_{p_l}} \tag{5.20}$$

where V is the hold-up, F is the volumetric flow-rate, C is the concentration, T is the temperature, ρ is the mass density, C_p is the specific heat capacity, subscripts f, l and v denote the feed, the exiting liquid stream and the exiting vapor stream respectively, superscripts A and B denote the components, and the superscript $(j) \in \{1,2\}$ denotes the index of the flash tank. We use dimensionless forms of the variables for time, concentration and temperature as defined for the tubular reactors for consistency.

5.2.3 Heat Exchanger Model

We use a lumped model for the heat exchangers, since the space-time in the heat exchanger is an order of magnitude less than that in the tubular reactors. The model equations are given by:

$$\frac{dT}{d\bar{t}} = \frac{T_f - T}{\tau} - \frac{hA}{\rho C_p}(T - T_c) \tag{5.21}$$

where T denotes the temperature in the heat exchanger, T_f is the feed temperature, h is the heat transfer coefficient, A is the area available for heat transfer, T_c is the temperature of the cooling fluid and τ is the residence time in the heat-exchanger. We non-dimensionalize the time and temperature variables as was done for the tubular reactors and flash tanks for consistency.

5.2.4 Model Simulation

We use sixteen modes corresponding to the slowest eigenvalues of the spatial differential operator of the PPDE of each distributed variable to capture the dominant dynamics of

the tubular reactors. Thus, every tubular reactor is described by thirty-two ODEs. We require four ODEs to capture the heat exchanger energy balances, two of which correspond to the process streams in HEx-1 and HEx-3, and the third and the fourth capture the heat exchange in HEx-2 between the exit stream of R-2 and the cooled process stream from HEx-3. For each of the flash columns, there are four equations. Our discretized model, thus, consists of 140 ODEs. This model was used both for simulation and control purposes.

5.3 Network Decomposition

We use the modularity maximization algorithm discussed in Section 3.4.2 for obtaining the network decomposition [40]. The algorithm obtains the optimal decomposition by successive bisection of a larger community until further bisection doesn't lead to an increase in the modularity measure. The decomposition obtained from two successive bisections that has a modularity of 0.4553 is given in table 5.3. Further bisection of Community 1 and Community 2 leads to an increase in modularity, but the communities thus obtained are not controllable. Hence, for control purposes, the decomposition in table 5.3 is the optimal decomposition. Figure 5.2 shows the interconnections between the variables in the optimal decomposition.

Community 1 allocates the inlet feed temperature of the streams, the utility fluid temperature in HEx-1 and the fresh feed flow to R-2 as the inputs for controlling the exit concentration of R-1 and R-2 and the exit temperature of R-2. Community 2 allocates utility fluid temperature to HEx-3 and the fresh feed flow-rate to R-3 as the manipulated variables to control the entry stream temperature to and exit stream composition of R-3. Community 3 allocates the heat inputs to the two flash columns and the recycle ratio to control the entry stream temperature for R-4, and the exit stream compositions of R-4 and

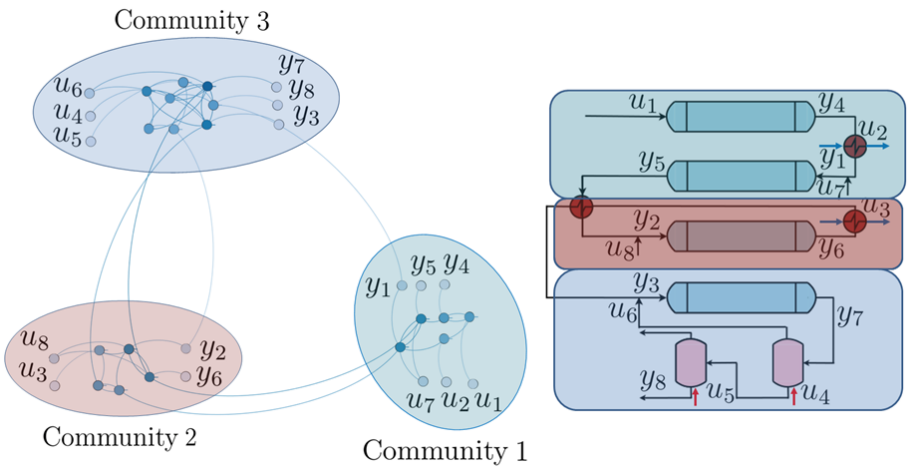


Figure 5.2: Communities in the reaction-separation plant

F-2. The resulting decomposition is not obvious based simply on intuition.

Table 5.3: Communities in the optimal decomposition

Variable Type	**Community 1**	**Community 2**	**Community 3**
Input	u_1	u_3	u_4
	u_2	u_8	u_5
	u_7		u_6
State	$C_A^{\text{R-1}}$	$C_A^{\text{R-3}}$	$C_A^{\text{R-4}}$
	$T^{\text{R-1}}$	$T^{\text{R-3}}$	$T^{\text{R-4}}$
	$C_A^{\text{R-2}}$	$T_h^{\text{HEx-2}}$	$C_A^{\text{F-1}}$
	$T^{\text{R-2}}$	$T_c^{\text{HEx-2}}$	$C_B^{\text{F-1}}$
	$T_h^{\text{HEx-1}}$	$T_h^{\text{HEx-3}}$	$T^{\text{F-1}}$
			$C_A^{\text{F-2}}$
			$C_B^{\text{F-2}}$
			$T^{\text{F-2}}$
Output	y_1	y_2	y_3
	y_4	y_6	y_7
	y_5		y_8

5.4 Simulation Results and Discussion

MPC, as discussed in Chapter 3 was implemented based on the optimal decomposition, and on the fully-centralized control structure. We employed interior point optimization (IPOPT) [89] to solve the underlying nonlinear constrained dynamic optimization problem. The plant model was simulated from the start-up conditions, and control was imple-

mented for the set-point tracking problem. The model parameters are given in table 5.4, the initial conditions of the input variables are given in table 5.5, and the initial conditions of the state variables of the flash tanks, which are the only variables starting from a non-zero value, are given in table 5.6. In the dimensionless form, a concentration value of -1 implies that the species is not present in the tank. Four set-point changes were imposed which are shown in table 5.7 and the corresponding steady-state values of the manipulated variables are given in table 5.8.

Table 5.4: Model Parameters

System	Parameter	Value
Reactors	B_{C_1}	0.1
	B_{C_2}	0.05
	B_T	3
	γ_1	10
	γ_2	30
	Pe_1	5
	Pe_2	5
	Slowest mode	-500
	Tubular reactor length	1
Heat-exchangers	$hA/\rho C_p$	10
	τ	0.1
Flash-tanks	ρ_l	1000 kg/m^3
	ρ_v	1 kg/m^3
	C_{p_l}	4186 J/kg-K
	C_{p_v}	2000 J/kg-K
	H_A	100
	H_B	0.01

Table 5.5: Initial conditions of the input variables

Input variable	**Initial condition**
u_1	0.5
u_2	0.2
u_3	0.4
u_4	3
u_5	0.8
u_6	0.5
u_7	0.2
u_8	0.1

Table 5.6: Initial conditions of state variables in the flash tanks

Variable	**Initial condition**
Hold-up in flash tanks	0.5
Concentration of A in flash tanks (dimensionless)	-1
Concentration of B in flash tanks (dimensionless)	-1
Temperature in flash tanks (dimensionless)	0

Table 5.7: Set-points of the controlled variables

Output	**Set-point 1**	**Set-point 2**	**Set-point 3**	**Set-point 4**
y_1	0.2030	0.3070	0.1553	0.2064
y_2	0.1630	0.2755	0.1567	0.2293
y_3	0.1163	0.2339	0.0557	0.2037
y_4	-0.0011	0.0002	-0.0003	-0.0005
y_5	-0.0095	-0.0023	-0.0091	-0.004
y_6	-0.3161	-0.3264	-0.3108	-0.3190
y_7	-0.4582	-0.5403	-0.4504	-0.5002
y_8	3.6772	4.1832	3.7584	4.2851

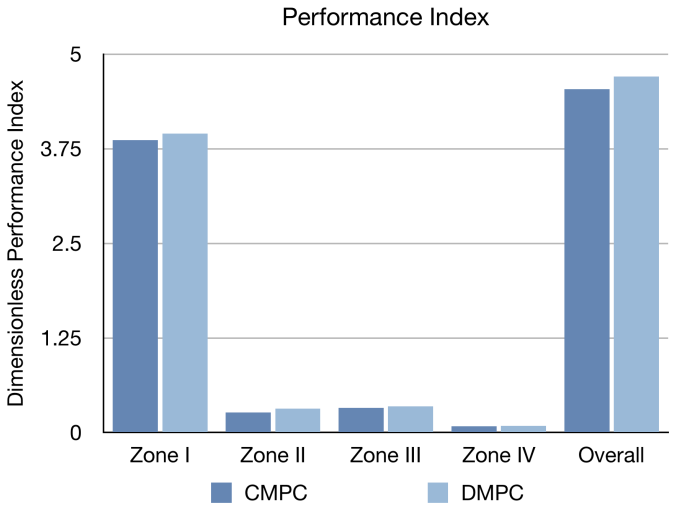


Figure 5.3: Dimensionless Performance Indices (DPI) of the CMPC and optimal decomposition - based DMPC for the set-point tracking problems

Table 5.8: Steady-state values of the manipulated variables

Input	**Steady-state for Set-point 1**	**Steady-state for Set-point 2**	**Steady-state for Set-point 3**	**Steady-state for Set-point 4**
u_1	0.1	0.4	0.3	0.2
u_2	-0.1	0.2	-0.3	0.1
u_3	-0.1	0.2	-0.3	0.1
u_4	1	0.7	1.2	0.5
u_5	1	0.7	1.2	0.5
u_6	0.1	0.5	0.2	0.3
u_7	0.5	0.8	0.6	1
u_8	0.5	0.8	0.6	1

The control performance was evaluated using the dimensionless performance index (DPI), as used in the previous case study, consists of the Integral Square Error (ISE) and

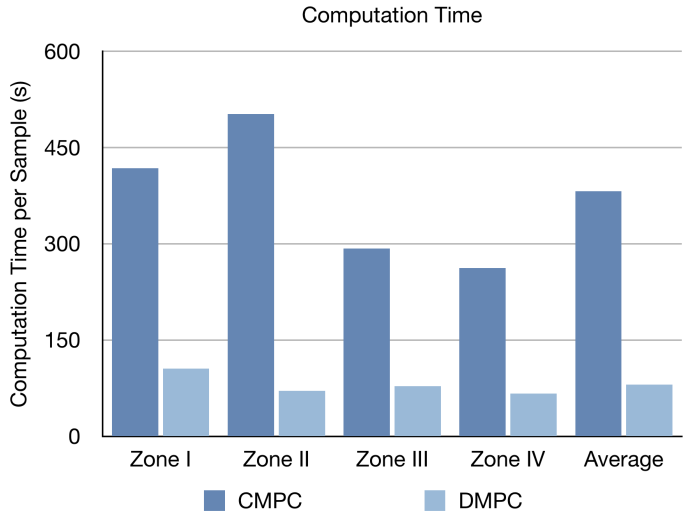


Figure 5.4: Computation time (seconds/sample) of the CMPC and optimal decomposition-based DMPC for the set-point tracking problems

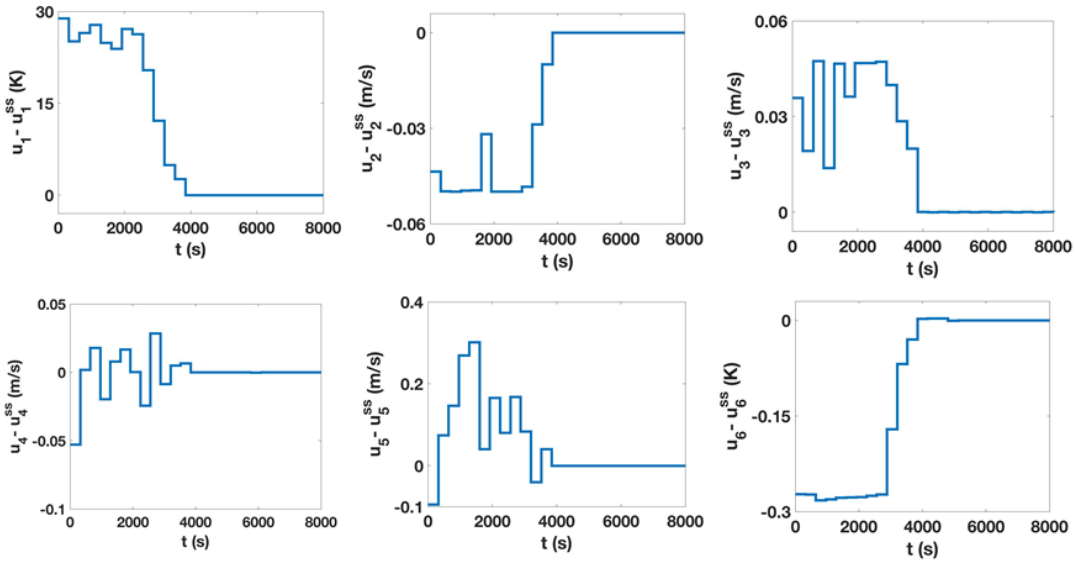


Figure 5.5: Input profiles from DMPC implementation based on the optimal decomposition

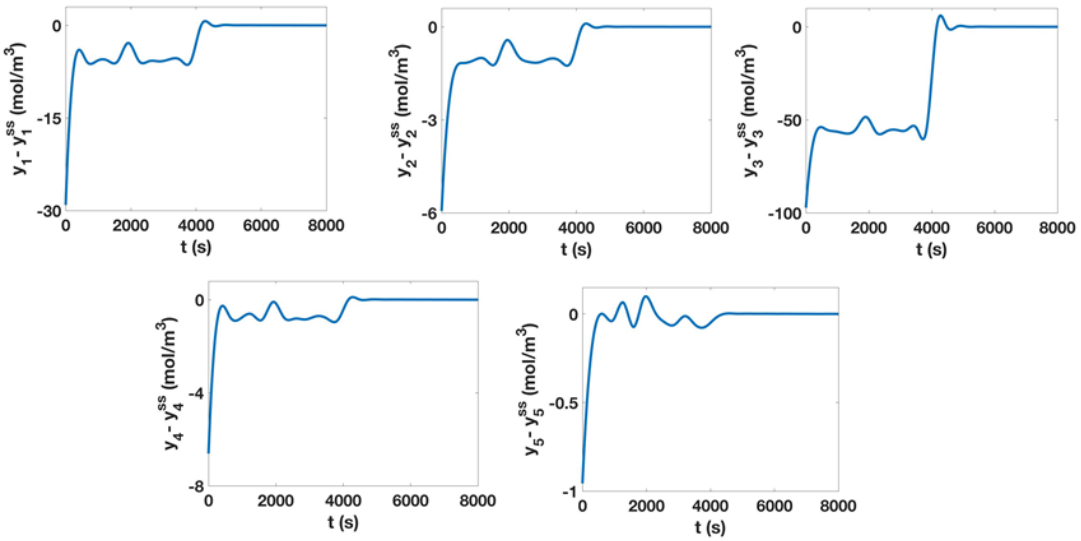


Figure 5.6: Output profiles from DMPC implementation based on the optimal decomposition

the Integral Square Control (ISC) defined by [90]:

$$\text{DPI} = \text{ISE} + \text{ISC} = \int_0^{t_f} e^T(t) P e(t) dt + \int_0^{t_f} \bar{u}^T(t) W \bar{u}(t) dt \tag{5.22}$$

ISE and ISC are the two integral terms in the objective function of the MPC optimization formulation and represent the deviation of the controlled and manipulated variables from the set-point and steady-state values respectively.

The performance indices and the computation time for each sample on the implementation of DMPC based on the optimal decomposition and of CMPC are shown in figures 5.3 and 5.4 respectively. The input and output profiles for DMPC are also given in figures 5.5 and 5.6 respectively. It can be observed that the performance of DMPC is comparable to that of CMPC, with a nearly 80% lower computation time per sample. The sampling time in the dimensionless form is 0.5. For a typical residence time of 300-400 seconds, the sampling time becomes 150-200 seconds. For such a case, CMPC is rendered impractical for real-time implementation, and DMPC based on the optimal decomposition

presents itself as a superior control structure based on the control performance and low computational time.

Chapter 6

Distributed Estimation and Nonlinear MPC using Community Detection *

6.1 Introduction

Model predictive control (MPC) is a prevalent control strategy in the chemical industry. MPC design casts the control problem in the form of a repeated constrained dynamic optimization problem which computes a sequence of future manipulated inputs at each sampling time [96, 97]. Consequently, the applicability of MPC hinges on the real-time solvability of the underlying optimization problem which is a challenging task for large-scale process systems. This has motivated continued efforts to increase the computational efficiency of MPC [98–102]. An alternative to centralized model predictive control (CMPC), with the potential to accelerate computations, is distributed model predictive control (DMPC) whereby the control system is decomposed into smaller ones with some level of cooperation and communication among these local controllers [3, 103–107].

Many studies have focused on the distributed control problem assuming full state infor-

*Accepted in *Industrial & Engineering Chemistry Research* (2019)

mation at each sampling time. However, such an assumption cannot be generally invoked in practical cases, where the use of a state estimation method is required along with the control algorithm. Among the several available methods for state estimation, MHE has attracted a lot of attention since it can be formulated as a similar constrained dynamic optimization problem [23–25]. However, solving such an optimization-based estimation problem at each sampling time for a large-scale process system is also computationally challenging. Any delay in such computations will directly affect the closed-loop performance [15] since the distributed controllers require the initial values of all state variables at each sampling time. Like DMPC, an alternative approach to overcome the computational challenges of centralized estimation is a system decomposition and a distributed MHE (DMHE) structure consisting of local estimators with some level of cooperation and communication [16, 17]. Extending the community-based decomposition to the combined control and estimation problem is the subject of this chapter.

Specifically, we hereby propose an algorithmic framework to address the feedback control problem for nonlinear process systems through a combined iterative DMPC and DMHE architecture built around community-based optimal decompositions for the control and estimation problems. These decompositions are obtained through distinct graphs corresponding to the estimation and control problem respectively, and thus are not necessarily the same. The proposed distributed output-feedback control strategy is implemented on the process of benzene alkylation with ethylene. The closed-loop performance and the average computation time are evaluated using detailed simulations for the optimal and suboptimal decompositions, as well as for centralized estimation and control implementations.

The rest of the chapter is organized as follows. In Section “Model predictive control and moving horizon estimation”, we present the mathematical description of the studied class of nonlinear systems and review MPC and MHE synthesis for output regulation. In

Section "Distributed estimation and control", we describe the community-based system decomposition for estimation and control, and the distributed estimation/control architecture. Finally, the impact of system decomposition on the closed-loop performance and computational effort of distributed estimation and control is examined for the process of benzene alkylation with ethylene, emphasizing on the importance of system decomposition in distributed output feedback control.

6.2 Model predictive control and moving horizon estimation

We consider nonlinear process systems defined by the following state-space model

$$\begin{aligned} \dot{x}(t) &= f\big(x(t), u(t)\big) + \omega(t) \\ y(t) &= g\big(x(t), u(t)\big) \\ \hat{y}(t) &= h\big(x(t), u(t)\big) + \nu(t) \end{aligned} \tag{6.1}$$

where $x(t) = [x_1(t)\ \ x_2(t)\ \ \cdots\ \ x_n(t)]^T \in \mathbb{R}^n$ is the vector of state variables, $u(t) = [u_1(t)\ \ u_2(t)\ \ \cdots\ \ u_m(t)]^T \in \mathbb{R}^m$ the vector of manipulated inputs, $\omega(t) = [\omega_1(t)\ \ \omega_2(t)\ \ \cdots\ \ \omega_n(t)]^T \in \mathbb{R}^n$ the vector of additive disturbances to the process, $y(t) = [y_1(t)\ \ y_2(t)\ \ \cdots\ \ y_r(t)]^T \in \mathbb{R}^r$ the vector of controlled outputs, $\hat{y}(t) = [\hat{y}_1(t)\ \ \hat{y}_2(t)\ \ \cdots\ \ \hat{y}_s(t)]^T \in \mathbb{R}^s$ the vector of measured outputs, $\nu(t) = [\nu_1(t)\ \ \nu_2(t)\ \ \cdots\ \ \nu_s(t)]^T \in \mathbb{R}^s$ the vector of measurement noises, t is the time, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^r$, and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^s$ denote smooth functions. Without loss of generality, we assume that the origin is the equilibrium point for the unforced model dynamics. The availability of measured outputs of the system is also assumed at predetermined sampling times.

The CMPC design for the system of (6.1), as done in previous chapters, can be de-

scribed by the following constrained nonlinear dynamic optimization problem

$$\begin{aligned}
\min_{u} \quad & \int_{t_k}^{t_k+N_cT} \left(\left\| y - y^{ref} \right\|_P^2 + \left\| u - u^{ref} \right\|_W^2 \right) dt \\
\text{s.t.} \quad & \dot{x} = f(x,u) + \omega \\
& y = g(x,u) \\
& u^{min} \leq u \leq u^{max} \\
& \xi(x,u,t) \leq 0 \\
& \phi(x,u,t) = 0
\end{aligned} \tag{6.2}$$

where

$$\begin{aligned}
\left\| y - y^{ref} \right\|_P^2 &= (y - y^{ref})^T P (y - y^{ref}) \\
\left\| u - u^{ref} \right\|_W^2 &= (u - u^{ref})^T W (u - u^{ref})
\end{aligned} \tag{6.3}$$

and t_k indicates the k^{th} sampling time, N_c denotes the number of sampling times in the control horizon, and T is the sampling time period. The desired controlled outputs and the corresponding steady state manipulated inputs are denoted by $y^{ref}(t) = [y_1^{ref}(t) \;\; y_2^{ref}(t) \;\; \cdots \;\; y_r^{ref}(t)]^T \in \mathbb{R}^r$ and $u^{ref}(t) = [u_1^{ref}(t) \;\; u_2^{ref}(t) \;\; \cdots \;\; u_m^{ref}(t)]^T \in \mathbb{R}^m$, respectively. These can be adjusted during closed-loop process operation due to safety concerns and production demands. We also consider the positive definite weight matrices of $P \in \mathbb{R}^{r \times r}$ and $W \in \mathbb{R}^{m \times m}$ in the optimization objective function to penalize the output regulation errors and manipulated variables. The lower and upper bounds of the vector of manipulated inputs are denoted by $u^{min} \in \mathbb{R}^m$ and $u^{max} \in \mathbb{R}^m$, respectively. The system inequality and equality constraints are also defined by considering the general nonlinear vector functions of ξ and ϕ. The temporal profiles of all manipulated inputs are computed for the prediction horizon of $[t_k \;\; t_k + N_cT]$ by solving the dynamic optimization problem of (6.2) subject to the process model, lower/upper bounds, and equality/inequality constraints. The values of manipulated inputs at time t_k are then applied to the process during $t \in [t_k \;\; t_k + T)$, the

prediction horizon is shifted forward, and then the optimization problem is resolved for the new prediction horizon $[t_k + T \;\; t_k + (N_c + 1)T]$.

Through reconciling the past information of the system of (6.1), we can formulate the centralized MHE (CMHE) problem by the following constrained nonlinear dynamic optimization problem

$$\begin{aligned} \min_{x_k} \quad & \int_{t_k - N_m T}^{t_k} \left(\|\nu\|_R^2 + \|\omega\|_Q^2 \right) \, dt \\ \text{s.t.} \quad & \dot{x} = f(x,u) + \omega \\ & \hat{y} = g(x,u) + \nu \\ & x_k^{min} \leq x_k \leq x_k^{max} \\ & \xi(x,u,t) \leq 0 \\ & \phi(x,u,t) = 0 \end{aligned} \tag{6.4}$$

where

$$\begin{aligned} \|\nu\|_R^2 &= \nu^T R \nu \\ \|\omega\|_Q^2 &= \omega^T Q \omega \end{aligned} \tag{6.5}$$

and x_k is the vector of the estimated state variables at the sampling time t_k, N_m denotes the number of sampling times in the estimation horizon, and $R \in \mathbb{R}^{s \times s}$ and $Q \in \mathbb{R}^{n \times n}$ are positive definite weight matrices which penalize the magnitude of additive disturbances and measurement noises. The lower and upper bounds of the estimated state variables are denoted by $x_k^{min} \in \mathbb{R}^n$ and $x_k^{max} \in \mathbb{R}^n$, respectively. By solving the dynamic optimization problem of (6.4) subject to the process model, lower/upper bounds, and equality/inequality constraints, we can compute the estimated values of the state variables at the sampling time t_k which are used as the initial values of the predictive model in the CMPC structure. Such a two-tier dynamic optimization problem (MHE+MPC) must be repeatedly solved at each sampling time. The schematic of the combined estimation and control problem is

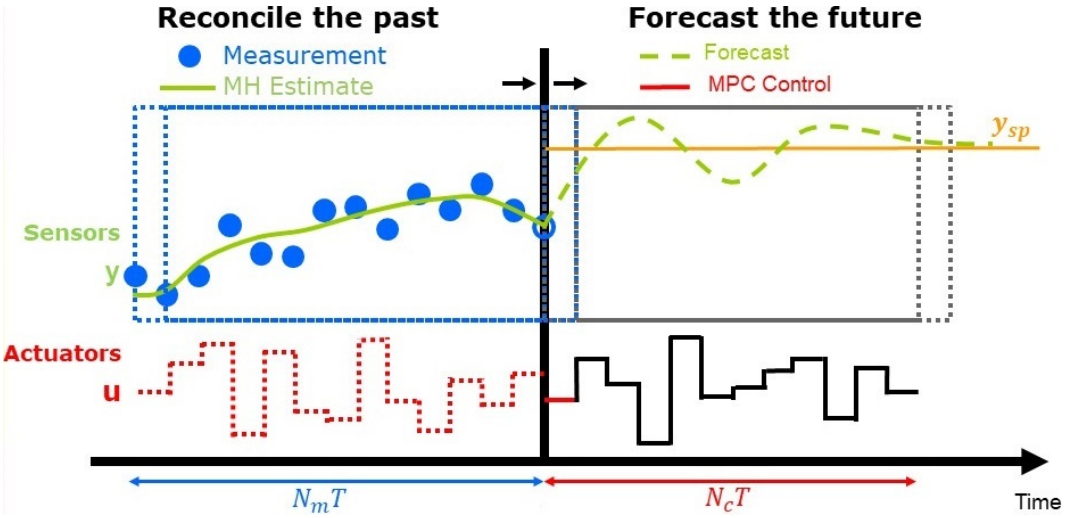


Figure 6.1: An schematic of the combined estimation and control problem (MHE+MPC)

presented in Figure 6.1.

6.2.1 Distributed estimation and control models

For distributed control, as before, we can decompose the system into L interconnected subsystems described by a set of state space submodels

$$\begin{aligned} \dot{x}_{(i)}(t) &= f_{(i)}\big(x(t), u_{(i)}(t)\big) + \omega_{(i)}(t), \qquad i = 1, \ldots, L \\ y_{(i)}(t) &= g_{(i)}\big(x(t), u_{(i)}(t)\big) \end{aligned} \tag{6.6}$$

where

$$\begin{aligned} x &= [x_{(1)}^T \;\; x_{(2)}^T \;\; \cdots \;\; x_{(L)}^T]^T \\ u &= [u_{(1)}^T \;\; u_{(2)}^T \;\; \cdots \;\; u_{(L)}^T]^T \\ \omega &= [\omega_{(1)}^T \;\; \omega_{(2)}^T \;\; \cdots \;\; \omega_{(L)}^T]^T \\ y &= [y_{(1)}^T \;\; y_{(2)}^T \;\; \cdots \;\; y_{(L)}^T]^T \end{aligned} \tag{6.7}$$

and

$$
\begin{aligned}
f &= [f_{(1)}^T \;\; f_{(2)}^T \;\; \cdots \;\; f_{(L)}^T]^T \\
g &= [g_{(1)}^T \;\; g_{(2)}^T \;\; \cdots \;\; g_{(L)}^T]^T
\end{aligned}
\tag{6.8}
$$

with $f_{(i)} : \mathbb{R}^n \times \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$ and $g_{(i)} : \mathbb{R}^n \times \mathbb{R}^{m_i} \to \mathbb{R}^{r_i}$, being the corresponding components of f and g for the i^{th} subsystem. The state variables, manipulated inputs, additive disturbances, and output variables of the subsystems can be represented by

$$
\begin{aligned}
x_{(i)} &= [x_{(i),1} \;\; x_{(i),2} \;\cdots\; x_{(i),n_i}]^T \\
u_{(i)} &= [u_{(i),1} \;\; u_{(i),2} \;\cdots\; u_{(i),m_i}]^T \\
\omega_{(i)} &= [\omega_{(i),1} \;\; \omega_{(i),2} \;\cdots\; \omega_{(i),n_i}]^T \\
y_{(i)} &= [y_{(i),1} \;\; y_{(i),2} \;\cdots\; y_{(i),r_i}]^T
\end{aligned}
\tag{6.9}
$$

where $n = \sum_{i=1}^{L} n_i$, $m = \sum_{i=1}^{L} m_i$, and $r = \sum_{i=1}^{L} r_i$.

Similarly, the system of (6.1) can be decomposed into N subsystems for distributed estimation of the system state variables in the following form

$$
\begin{aligned}
\dot{x}_{(j)}(t) &= f_{(j)}\big(x(t), u(t)\big) + \omega_{(j)}(t), \qquad j = 1, \ldots, N \\
\hat{y}_{(j)}(t) &= h_{(j)}\big(x(t), u(t)\big) + \nu_{(j)}(t)
\end{aligned}
\tag{6.10}
$$

where

$$
\begin{aligned}
x &= [x_{(1)}^T \;\; x_{(2)}^T \;\; \cdots \;\; x_{(N)}^T]^T \\
\omega &= [\omega_{(1)}^T \;\; \omega_{(2)}^T \;\; \cdots \;\; \omega_{(N)}^T]^T \\
\nu &= [\nu_{(1)}^T \;\; \nu_{(2)}^T \;\; \cdots \;\; \nu_{(N)}^T]^T \\
\hat{y} &= [\hat{y}_{(1)}^T \;\; \hat{y}_{(2)}^T \;\; \cdots \;\; \hat{y}_{(N)}^T]^T
\end{aligned}
\tag{6.11}
$$

and

$$
\begin{aligned}
f &= [f_{(1)}^T \;\; f_{(2)}^T \;\; \cdots \;\; f_{(N)}^T]^T \\
h &= [h_{(1)}^T \;\; h_{(2)}^T \;\; \cdots \;\; h_{(N)}^T]^T
\end{aligned}
\tag{6.12}
$$

with $f_{(j)} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n_j}$ and $h_{(j)} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{s_i}$. The state variables, additive disturbances, measured outputs, and measurement noises can be represented by

$$
\begin{aligned}
x_{(j)} &= [x_{(j),1} \;\; x_{(j),2} \;\cdots\; x_{(j),n_j}]^T \\
\omega_{(j)} &= [\omega_{(j),1} \;\; \omega_{(j),2} \;\cdots\; \omega_{(j),n_j}]^T \\
\nu_{(j)} &= [\nu_{(j),1} \;\; \nu_{(j),2} \;\cdots\; \nu_{(j),s_j}]^T \\
\hat{y}_{(j)} &= [\hat{y}_{(j),1} \;\; \hat{y}_{(j),2} \;\cdots\; \hat{y}_{(j),s_j}]^T
\end{aligned}
\tag{6.13}
$$

where $n = \sum_{j=1}^{N} n_j$ and $s = \sum_{j=1}^{N} s_j$.

6.2.2 DMPC formulation for output regulation

The optimization problem for the i^{th} local controller corresponding to the i^{th} subsystem is formulated as

$$
\begin{aligned}
\min_{u_{(i)}} \quad & \int_{t_k}^{t_k+N_cT} \left(\left\| y_{(i)} - y_{(i)}^{ref} \right\|_{P_{(i)}}^2 + \left\| u_{(i)} - u_{(i)}^{ref} \right\|_{W_{(i)}}^2 \right) dt \\
\text{s.t.} \quad & \dot{x} = f(x, u_{(i)}) + \omega \\
& y_{(i)} = g_{(i)}(x, u_{(i)}) \\
& u_{(i)}^{min} \le u \le u_{(i)}^{max} \\
& \xi(x, u_{(i)}, t) \le 0 \\
& \phi(x, u_{(i)}, t) = 0
\end{aligned}
\tag{6.14}
$$

where the positive definite matrices $P_{(i)}$ and $W_{(i)}$ include only the components corresponding to the outputs and manipulated inputs of the i^{th} subsystem.

Note that we did not decompose the system state-space model (dynamic constraint in the dynamic optimization problem) and the algebraic equality/inequality constraints, to guarantee that each local controller accounts for the entire system dynamics and the related constraints. The proposed architecture distributes only the manipulated inputs and the controlled outputs. The local controllers communicate over the network by sharing the latest values of their computed manipulated inputs. At each iteration at the sampling time t_k, each local controller computes the future trajectory of its manipulated inputs by solving the corresponding optimization problem of (6.14) for the prediction horizon of $[t_k \ t_k + N_c T]$, based on the estimated state variables and the latest values of the manipulated inputs received from other the local controllers. Then the local controllers interchange the manipulated input trajectories computed at the latest iteration and re-evaluate the input trajectories by solving (6.14) for $i = 1, \ldots, L$ in parallel. For the first iteration at each sampling time, the required trajectories are initialized by the values obtained from the last iteration of the previous sampling time. This iterative procedure terminates when the Euclidean norm of the difference between the vectors of manipulated inputs of two consecutive iterations becomes smaller than a desired threshold [3, 83]. We determine the required computation time at each sampling time by the longest computation time of the local controllers. The readers may refer to [**?**, 83] for a detailed discussion on different communication and cooperation protocols possible between the local controllers and their impact on the overall performance of the distributed control design.

6.2.3 DMHE formulation for state estimation

For distributed estimation, we use the following formulation

$$
\begin{aligned}
\min_{x_{k,(j)}} \quad & \int_{t_k - N_m T}^{t_k} \left(\left\| \nu_{(j)} \right\|^2_{R_{(j)}} + \left\| \omega_{(j)} \right\|^2_{Q_{(j)}} \right) dt \\
\text{s.t.} \quad & \dot{x} = f(x,u) + \omega \\
& \hat{y}_{(j)} = g_{(j)}(x,u) + \nu_{(j)} \\
& x^{min}_{k,(j)} \leq x_{k,(j)} \leq x^{max}_{k,(j)} \\
& \xi(x,u,t) \leq 0 \\
& \phi(x,u,t) = 0
\end{aligned}
\tag{6.15}
$$

where the positive definite matrices $R_{(j)}$ and $Q_{(j)}$ correspond to the measured outputs and estimated state variables of the j^{th} subsystem. Similar to the iterative distributed control design, the j^{th} local estimator computes its state variables at the sampling time and exchanges these values over the network. Then the local estimators re-compute the state variables by resolving the dynamic optimization problem of (6.15). This iterative procedure terminates when the Euclidean norm of the difference between the vectors of estimated states of two consecutive iterations becomes smaller than a desired threshold. Similar to the proposed DMPC, the system state-space model and the algebraic equality/inequality constraints are not decomposed to guarantee that each local estimator satisfies the system dynamics and the related constraints. We only consider the distribution of the unmeasurable state variables and the measured outputs over the network of variables. Figure 6.2 illustrates the block diagrams of the closed-loop process for the proposed combined distributed estimation and control. In this proposed distributed estimation/control architecture, each local estimator/controller solves a small dynamic optimization problem (with a particular objective function) in parallel to other estimators/controllers while they share their information over the network. Each local estimator/controller deals only with

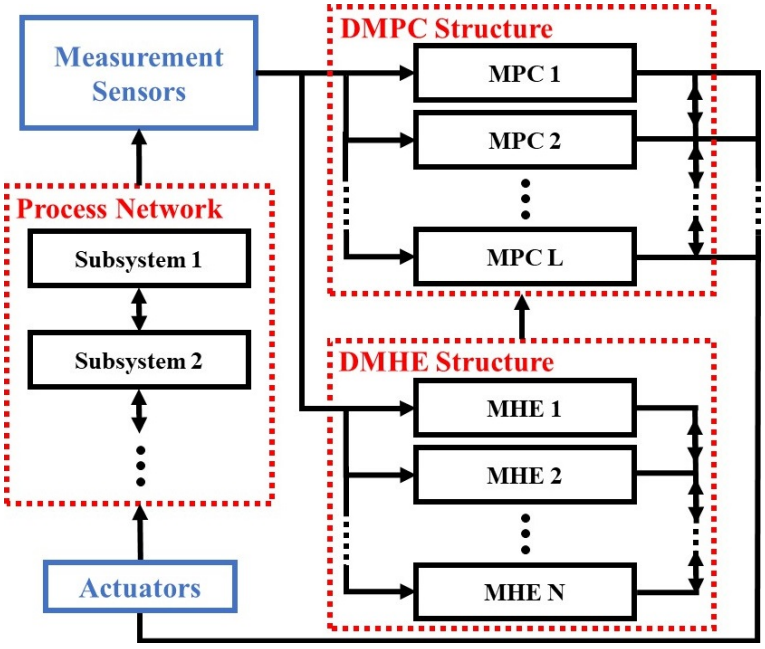


Figure 6.2: Iterative combined DMHE and DMPC structure for different system decompositions for the control problem (L subsystems) and the estimation problem (N subsystems)

the outputs which belong to its corresponding subsystem. The local estimators/controllers communicate over the network by sharing their computed state values/temporal profiles of manipulated inputs.

6.2.4 Decompositions for control and estimation

To implement the proposed combined distributed estimation and control, we require a decomposition of the system into smaller subsystems. The decomposition approach considered in this chapter is based on the community detection in the equation graphs (graph representations of the process systems) that capture the structural interactions among the different variables in the system. For the control problem, the graph captures the relationships of the manipulated inputs, state variables of the system, and the controlled outputs, as is standard for ODE systems [10]. The graph captures the algebraic relationships of the measured outputs, state variables of the system, and the unmeasurable state variables

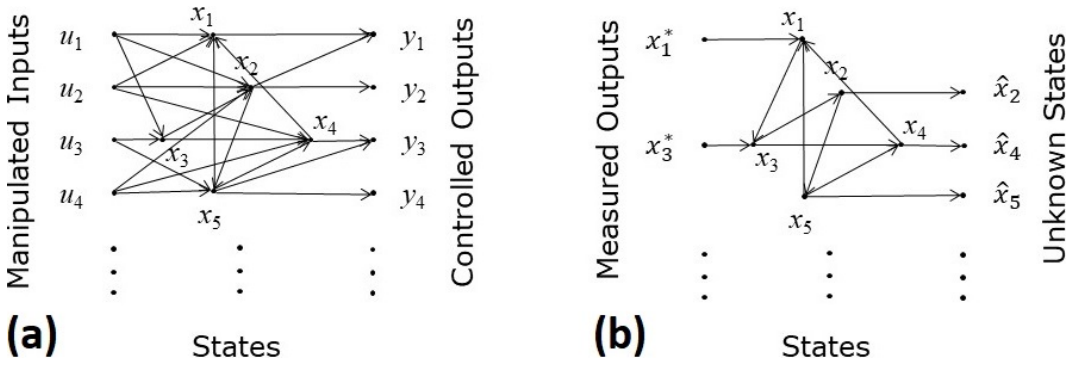


Figure 6.3: Graph representations of a typical process system for (a) control and (b) estimation problems

for the estimation problem. We used the modularity maximization algorithm discussed in Section 3.4.2 for the optimal decomposition determination.

The two different types of graphs considered are shown in Figure 6.3. Community detection methods applied to these graphs will in general lead to different optimal decompositions. Consequently, the distributed control and estimation architectures designed based on these decompositions will be different in general.

6.3 An integrated process case study: Benzene alkylation with ethylene

In this section, we employ the proposed distributed estimation and control approach to address the output feedback regulation problem for an integrated process of benzene alkylation with ethylene to produce ethylbenzene. This is a widely used process in the petrochemical industry where its main product (ethylbenzene) is utilized as an intermediate to produce styrene (by dehydrating ethylbenzene), the precursor component for important polymers and copolymers [105, 108]. Ethylbenzene is also used as a solvent in many industrial applications [109].

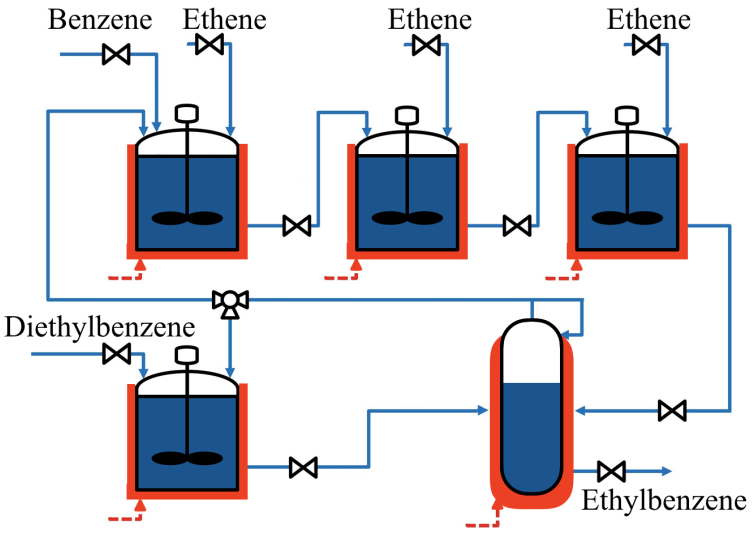


Figure 6.4: Benzene alkylation with ethylene

6.3.1 Process description

The benzene alkylation process considered consists of four continuous stirred-tank reactors (CSTRs) and a flash tank separator (Figure 6.4). The process is assumed to operate at high pressure which results in a liquid phase reaction with constant molar volumes. Three CSTRs in series are used for the following two catalytic reactions

$$\begin{aligned} &\textit{Reaction } 1: \quad \textit{Benzene } (A) + \textit{Ethylene } (B) \rightarrow \textit{Ethylbenzene } (C) \\ &\textit{Reaction } 2: \quad \textit{Ethylbenzene } (C) + \textit{Ethylene } (B) \rightarrow 1,3\,\textit{Diethylbenzene } (D) \end{aligned} \tag{6.16}$$

where ethylbenzene is the desired product and diethylbenzene is the byproduct [110]. Pure benzene and pure ethylene are fed to the first CSTR. The feed streams to the second and third CSTR only contain pure ethylene. The effluent of the third CSTR is then fed to the separator. We assume that the reaction in the separator is negligible. The majority of the benzene is separated and recycled to the reaction units (first and fourth CSTRs) and the bottom product of the separator is collected as the ultimate product of the plant. We also feed a pure 1,3-diethylbenzene to the fourth CSTR from another separator unit

whose dynamics are not considered in this case study for simplicity [105]. A catalyzed transalkylation reaction takes place in the fourth reactor

$$Reaction\ 3: \ \ Benzene\ (A) + 1,3\ Diethylbenzene\ (D) \rightarrow Ethylbenzene\ (C) \tag{6.17}$$

The effluent of the fourth CSTR is then fed to the separator. All of the reactors and the flash tank separator have external jackets used to provide/remove heat from the units.

6.3.2 Model description

By applying material and energy balances and under standard modeling assumptions [105, 111, 112], the following ODEs can be obtained that describe the dynamics of hold-up, temperature, and concentration of the different components in the four CSTRs and the

flash tank separator

$$
\begin{aligned}
\frac{dV_1}{dt} &= F_1 + F_2 + F_{r2} - F_3 \\
\frac{dT_1}{dt} &= \frac{Q_1 + F_1 C_{A0} H_A(T_{A0}) + F_2 C_{B0} H_B(T_{B0}) + F_{r2} C_{ir} H_i(T_4) - F_3 C_{i1} H_i(T_1)}{\sum_i^{A,B,C,D} C_{i1} C_{pi} V_1} \\
&\quad + \frac{-\Delta H_{r1} r_1(T_1, C_{A1}, C_{B1}) - \Delta H_{r2} r_2(T_1, C_{B1}, C_{C1}, C_{D1})}{\sum_i^{A,B,C,D} C_{i1} C_{pi}} \\
\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}) \\
\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_{B1}, C_{C1}, C_{D1}) \\
\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}, C_{D1}) \\
\frac{dC_{D1}}{dt} &= \frac{F_{r2} C_{Dr} - F_3 C_{D1}}{V_1} + r_2(T_1, C_{B1}, C_{C1}, C_{D1}) \\
\frac{dV_2}{dt} &= F_3 + F_4 - F_5 \\
\frac{dT_2}{dt} &= \frac{Q_2 + F_4 C_{B0} H_B(T_{B0}) + F_3 C_{i1} H_i(T_1) - F_5 C_{i2} H_i(T_2)}{\sum_i^{A,B,C,D} C_{i2} C_{pi} V_2} \\
&\quad + \frac{-\Delta H_{r1} r_1(T_2, C_{A2}, C_{B2}) - \Delta H_{r2} r_2(T_2, C_{A2}, C_{B2}, C_{D2})}{\sum_i^{A,B,C,D} C_{i2} C_{pi}} \\
\frac{dC_{A2}}{dt} &= \frac{F_3 C_{A1} - F_5 C_{A2}}{V_2} - r_1(T_2, C_{A2}, C_{B2}) \\
\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}, C_{D2}) \\
\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}, C_{D2}) \\
\frac{dC_{D2}}{dt} &= \frac{F_3 C_{D1} - F_5 C_{D2}}{V_2} + r_2(T_2, C_{B2}, C_{C2}, C_{D2}) \frac{dV_3}{dt} = F_5 + F_6 - F_7 \\
\frac{dT_3}{dt} &= \frac{Q_3 + F_6 C_{B0} H_B(T_{B0}) + F_5 C_{i2} H_i(T_2) - F_7 C_{i3} H_i(T_3)}{\sum_i^{A,B,C,D} C_{i3} C_{pi} V_3} \\
&\quad + \frac{-\Delta H_{r1} r_1(T_3, C_{A3}, C_{B3}) - \Delta H_{r2} r_2(T_3, C_{A3}, C_{B3}, C_{D3})}{\sum_i^{A,B,C,D} C_{i3} C_{pi}} \\
\frac{dC_{A3}}{dt} &= \frac{F_5 C_{A2} - F_7 C_{A3}}{V_3} - r_1(T_3, C_{A3}, C_{B3})
\end{aligned}
\tag{6.18}
$$

$$\frac{dC_{B3}}{dt} = \frac{F_5 C_{B2} + F_6 C_{B0} - F_7 C_{B3}}{V_3} - r_1(T_3, C_{A3}, C_{B3}) - r_2(T_3, C_{B3}, C_{C3}, C_{D3})$$

$$\frac{dC_{C3}}{dt} = \frac{F_5 C_{C2} - F_7 C_{C3}}{V_3} + r_1(T_3, C_{A3}, C_{B3}) - r_2(T_3, C_{B3}, C_{C3}, C_{D3})$$

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

$$\frac{dV_4}{dt} = F_7 + F_9 - F_8 - F_{r1} - F_{r2}$$

$$\frac{dT_4}{dt} = \frac{Q_4 + \sum_i^{A,B,C,D} \left(F_7 C_{i3} H_i(T_3) + F_9 C_{i5} H_i(T_5) - M_i H_i(T_4) - F_8 C_{i4} H_i(T_4) - M_i H_{vap,i}\right)}{\sum_i^{A,B,C,D} C_{i4} C_{pi} V_4}$$

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

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

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

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

$$\frac{dV_5}{dt} = F_{10} + F_{r1} - F_9$$

$$\frac{dT_5}{dt} = \frac{Q_5 + F_{10} C_{D0} H_D(T_{D0}) + \sum_i^{A,B,C,D} \left(F_{r1} C_{ir} H_i(T_4) - F_9 C_{i5} H_i(T_5)\right)}{\sum_i^{A,B,C,D} C_{i5} C_{pi} V_5}$$
$$+ \frac{-\Delta H_{r2} r_2(T_5, C_{B5}, C_{C5}, C_{D5}) - \Delta H_{r3} r_3(T_5, C_{A5}, C_{D5})}{\sum_i^{A,B,C,D} C_{i5} C_{pi}}$$

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

$$\frac{dC_{B5}}{dt} = \frac{F_{r1} C_{Br} - F_9 C_{B5}}{V_5} - r_2(T_5, C_{B5}, C_{C5}, C_{D5})$$

$$\frac{dC_{C5}}{dt} = \frac{F_{r1} C_{Cr} - F_9 C_{C5}}{V_5} - r_2(T_5, C_{B5}, C_{C5}, C_{D5}) + 2r_3(T5, C_{A5}, C_{D5})$$

$$\frac{dC_{D5}}{dt} = \frac{F_{r1} C_{Dr} + F_{10} C_{D0} - F_9 C_{D5}}{V_3} + r_2(T_5, C_{B5}, C_{C5}, C_{D5}) - r_3(T_5, C_{B5}, C_{C5})$$

where the definition of the process variables is presented in Table 6.1.

Table 6.1: Process variables.

Variables	Definition
F_3, F_5, F_7, F_8, F_9	Flow rates of the streams leaving the units
$F_1, F_2, F_4, F_6, F_{10}$	Flow rates of the streams leaving the units
F_r, F_{r1}, F_{r2}	Flow rates of the recycle streams
Q_1, Q_2, Q_3, Q_4, Q_5	External heat flows to each units
$C_{A1}, C_{B1}, C_{C1}, C_{D1}$	Concentrations of A, B, C, D in the first CSTR
$C_{A2}, C_{B2}, C_{C2}, C_{D2}$	Concentrations of A, B, C, D in the second CSTR
$C_{A3}, C_{B3}, C_{C3}, C_{D3}$	Concentrations of A, B, C, D in the third CSTR
$C_{A4}, C_{B4}, C_{C4}, C_{D4}$	Concentrations of A, B, C, D in the flash tank separator
$C_{A5}, C_{B5}, C_{C5}, C_{D5}$	Concentrations of A, B, C, D in the fourth CSTR
$C_{Ar}, C_{Br}, C_{Cr}, C_{Dr}$	Concentrations of A, B, C, D in the recycle stream
V_1, V_2, V_3, V_4, V_5	Hold-ups in each unit
T_1, T_2, T_3, T_4, T_5	Temperatures in each unit
T_{ref}	Reference temperature
$H_{vap,A}, H_{vap,B}, H_{vap,C}, H_{vap,D}$	Enthalpies of vaporization of A, B, C, D in the first CSTR
$H_{vap,A}, H_{vap,B}, H_{vap,C}, H_{vap,D}$	Enthalpies of vaporization of A, B, C, D in the first CSTR
$\Delta H_{r1}, \Delta H_{r2}, \Delta H_{r3}$	Heat of reactions 1, 2, 3
T_{A0}, T_{B0}, T_{D0}	Temperatures of the pure feed streams

The reaction rate expressions are

$$
\begin{aligned}
r_1(T, C_A, C_B) &= 0.0840 \exp\left(\frac{-9502}{RT}\right) C_A^{0.32} C_B^{1.5} \\
r_2(T, C_B, C_C, C_D) &= \frac{0.0850 \exp\left(\frac{-20643}{RT}\right) C_B^{2.5} C_C^{0.5}}{1 + k_{EB2} C_D} \\
r_3(T, C_A, C_D) &= \frac{66.1 \exp\left(\frac{-61280}{RT}\right) C_A^{1.0218} C_D}{1 + k_{EB3} C_A}
\end{aligned}
\tag{6.19}
$$

where

$$
\begin{aligned}
k_{EB2} &= 0.152 \exp\left(\frac{-3933}{RT}\right) \\
k_{EB3} &= 0.490 \exp\left(\frac{-50870}{RT}\right)
\end{aligned}
\tag{6.20}
$$

We consider a linear dependency of the enthalpies of the reactants on the temperature

$$H_i(T) = H_{ref,i} + C_{pi}(T - T_{ref}), \qquad i = A,B,C,D \tag{6.21}$$

where the heat capacities of the species, C_{pi} for $i = A,B,C,D$, are assumed to be constant, and $H_{ref,i}$ for $i = A,B,C,D$ denotes the enthalpies of the species at the reference temperature, T_{ref}. We also consider a linear temperature dependency of the relative volatilities of the species

$$\begin{aligned} \alpha_A &= 0.0449T_4 + 10 \\ \alpha_B &= 0.0260T_4 + 10 \\ \alpha_C &= 0.0065T_4 + 0.5 \\ \alpha_D &= 0.0058T_4 + 0.25 \end{aligned} \tag{6.22}$$

The concentration of the species in the recycle stream can be calculated by

$$C_{ir} = \frac{M_i}{\sum_j^{A,B,C,D} \frac{M_i}{C_{j0}}}, \qquad i = A,B,C,D \tag{6.23}$$

where C_{j0} for $j = A,B,C,D$ denotes the molar densities of the pure species and M_i presents the molar flow rates of the species in the overhead stream of the flash tank separator

$$M_i = k\frac{\alpha_i(F_7C_{i3} + F_9C_{i5})\sum_j^{A,B,C,D}(F_7C_{j3} + F_9C_{j5})}{\sum_j^{A,B,C,D}\alpha_j(F_7C_{j3} + F_9C_{j5})}, \qquad i = A,B,C,D \tag{6.24}$$

and k denotes the fraction of overhead flow recycled to the CSTRs.

The parameters of the model are presented in Table 6.2, and the reference values (steady states) of the system states and manipulated inputs are shown in Tables 6.2 and 6.4, respectively.

The hold-up, the temperature, and the concentration of the desired product (ethylben-

Table 6.2: Process parameters

Parameters	Values	Units	Parameters	Values	Units
$H_{vap,A}$	3.073×10^4	J/mol	$H_{vap,B}$	1.35×10^4	J/mol
$H_{vap,C}$	4.226×10^4	J/mol	$H_{vap,D}$	4.55×10^4	J/mol
$H_{ref,A}$	7.44×10^4	J/mol	$H_{ref,B}$	5.91×10^4	J/mol
$H_{ref,C}$	2.02×10^4	J/mol	$H_{ref,D}$	-2.89×10^4	J/mol
$C_{p,A}$	184.6	$J/molK$	$C_{p,B}$	59.1	$J/molK$
$C_{p,C}$	247	$J/molK$	$C_{p,D}$	301.3	$J/molK$
C_{A0}	1.126×10^4	mol/m^3	C_{B0}	2.028×10^4	mol/m^3
C_{D0}	6485	mol/m^3	T_{ref}	450	K
T_{A0}	473×10^4	K	T_{B0}	473×10^4	K
T_{D0}	473	K	k	0.8	$-$
ΔH_{r1}	-1.53×10^5	J/mol	ΔH_{r2}	-1.118×10^5	J/mol
ΔH_{r3}	4.141×10^5	J/mol	R	8.314	$J/molK$

zene) in the four CSTRs and the separator are considered as the controlled outputs. As illustrated in Figure 6.5, the stream flow rates and heat flows to the units are the manipulated inputs. Also, it is assumed that we can only measure the hold-up and temperature inside the reactors and the separator (Figure 6.6) which necessitates estimating the concentrations of the species required by the controller to compute the manipulated inputs. Thus we have 30 state variables, 17 manipulated inputs, 15 controlled outputs, and only 10 measured outputs.

6.3.3 Distributed estimation and control architecture

Figures 6.7 and 6.8 show the graph representations for the control and estimation problems, respectively. In both graph representations, the core structures which involve the interactions between the state variables are identical. In the graph representation of the control problem, we consider specifically the interaction between manipulated inputs (stream flow rates and heat flows to the reactors and the separator), the state variables of the system (hold-up, temperature, and concentrations of the species in the reactors and the separator),

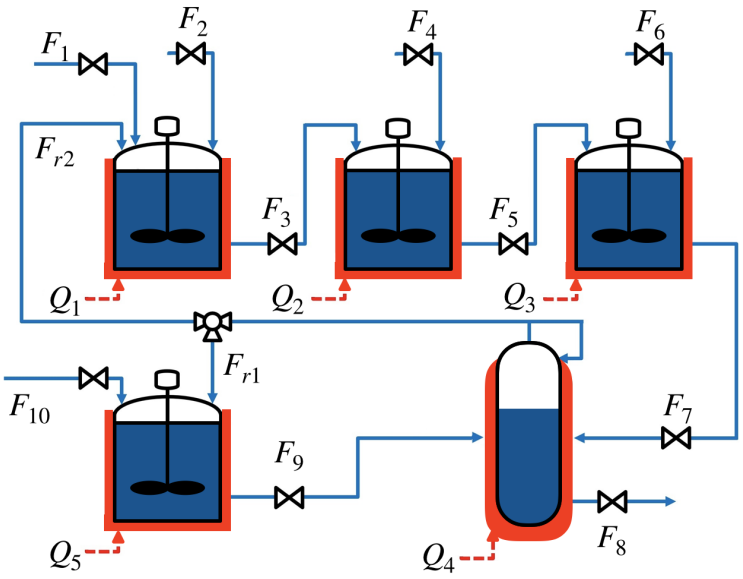


Figure 6.5: The set of manipulated inputs for the benzene alkylation with ethylene

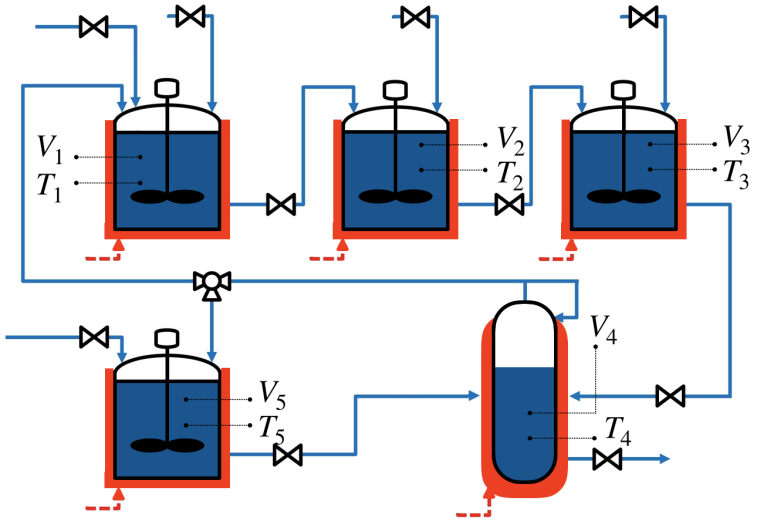


Figure 6.6: The set of measured outputs for the benzene alkylation with ethylene

Table 6.3: Reference (steady state) values of the state variables

Parameters	Values	Units	Parameters	Values	Units
V_1	1	m^3	V_2	1	m^3
V_3	1	m^3	V_4	3	m^3
V_5	1	m^3	T_1	474.36	K
T_2	473.65	K	T_3	469.76	K
T_4	465.42	K	T_5	467.58	K
C_{A1}	9.101×10^3	mol/m^3	C_{B1}	22.38	mol/m^3
C_{C1}	1.116×10^3	mol/m^3	C_{D1}	2.204×10^2	mol/m^3
C_{A2}	7.548×10^3	mol/m^3	C_{B2}	23.73	mol/m^3
C_{C2}	1.906×10^3	mol/m^3	C_{D2}	3.731×10^2	mol/m^3
C_{A3}	6.163×10^3	mol/m^3	C_{B3}	25.16	mol/m^3
C_{C3}	2.614×10^3	mol/m^3	C_{D3}	5.056×10^2	mol/m^3
C_{A4}	1.756×10^3	mol/m^3	C_{B4}	14.03	mol/m^3
C_{C4}	5.402×10^3	mol/m^3	C_{D4}	7.396×10^2	mol/m^3
C_{A5}	5.805×10^3	mol/m^3	C_{B5}	4.630	mol/m^3
C_{C5}	3.716×10^3	mol/m^3	C_{D5}	2.091×10^2	mol/m^3

and the controlled outputs (hold-up, temperature, and concentrations of ethylbenzene in the reactors and the separator). In the graph representation of the estimation problem, we consider the interactions between measured outputs (hold-up and temperature in the reactors and the separator), the state variables of the system, and the unmeasurable state variables (concentrations of the species in the reactors and the separator). These interactions define the adjacency matrices that are used in the community detection algorithm.

If the subsystems for estimation and control are more densely connected internally than the rest of the network, the resulting distributed architecture may improve the state estimation quality and closed-loop performance. Also, it may minimize the required iterations in the distributed estimation and control which in turn reduces the computation time. Figures 6.9 and 6.10 show the optimal system decompositions using community detection (maximizing modularity) for the control and the state estimation problems and the corresponding distributions of the resulting subsystems on the process flowsheet. As

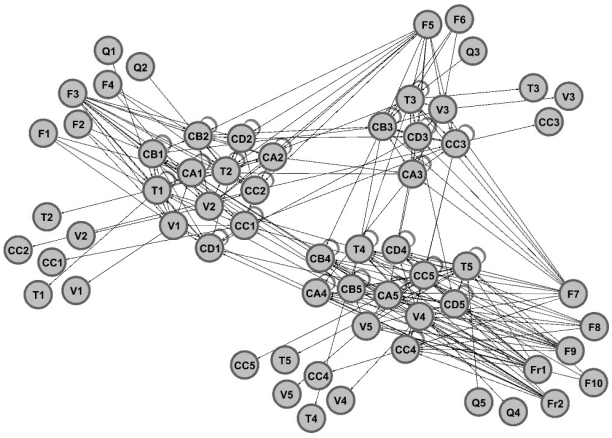


Figure 6.7: The graph representation for control problem

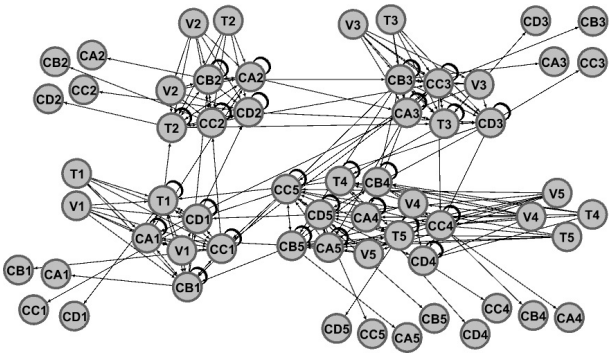


Figure 6.8: The graph representation for estimation problem

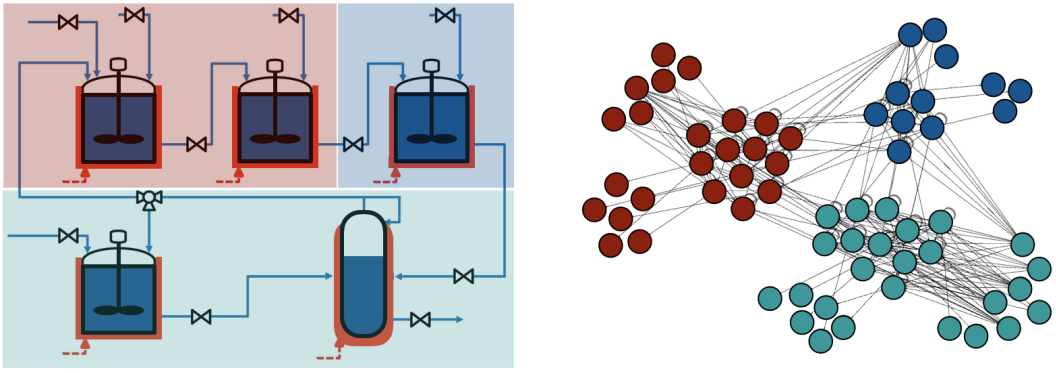


Figure 6.9: System decomposition for control

Table 6.4: Reference (steady state) values of the manipulated inputs

Parameters	Values	Units	Parameters	Values	Units
F_1	7.1×10^{-3}	m^3/s	F_2	8.697×10^{-4}	m^3/s
F_3	1.4×10^{-2}	m^3/s	F_4	8.697×10^{-4}	m^3/s
F_5	1.48×10^{-2}	m^3/s	F_4	8.697×10^{-4}	m^3/s
F_7	1.57×10^{-2}	m^3/s	F_8	1.2×10^{-2}	m^3/s
F_9	8.3×10^{-3}	m^3/s	F_{r1}	6×10^{-3}	m^3/s
F_{r2}	6×10^{-3}	m^3/s	Q_1	-4.4×10^6	J/s
Q_2	-4.6×10^6	J/s	Q_3	-4.7×10^6	J/s
Q_4	9.2×10^6	J/s	Q_5	5.6×10^6	J/s

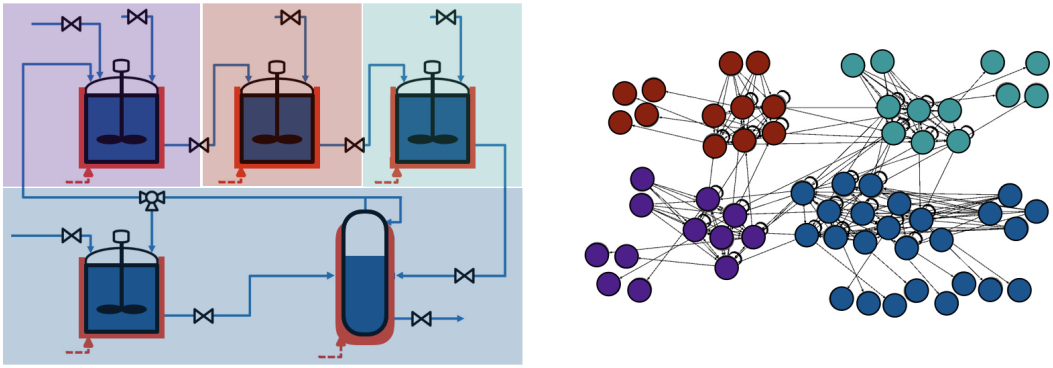


Figure 6.10: System decomposition for state estimation

mentioned in the previous sections, we obtain different decompositions since the structure of the two graphs is different. In the system decomposition for control, we obtain three subsystems consisting of (1) the first and second CSTRs, (2) the third CSTR, and (3) the fourth CSTR and the flash tank separator. For the state estimation problem, four subsystems are obtained that involve (1) the first CSTR, (2) the second CSTR, (3) the third CSTR, and (4) the fourth CSTR and the flash tank separator. In both of the decompositions the fourth CSTR and the separator are placed in the same community since there are tightly connected through the material recycle streams. The difference between the decomposed architectures for control and estimation relates to the first and second CSTRs. In the control decomposition the first and second CSTR are in the same subsystem. In the

state estimation decomposition the first and second CSTR form different subsystems. The reason for this is that the flow rate of the pure benzene entering the first CSTR, which is a manipulated input, has a strong effect on both the first and the second reactor. On the other hand, for the estimation problem we are just estimating concentrations from temperature measurements, and there is no advantage in considering these reactors together.

6.4 Simulation Results and Discussion

The number of sampling periods in the moving horizon for solving the optimization-based control and estimation problems is set to $N_c = N_m = 10$, where the sampling time is $T = 60s$. The initial values of the system states are given in Table 6.5.

Table 6.5: Reference (steady state) values of the state variables

Parameters	Values	Units	Parameters	Values	Units
V_1	1.5	m^3	V_2	0.5	m^3
V_3	2	m^3	V_4	1.5	m^3
V_5	1.6	m^3	T_1	423.15	K
T_2	423.15	K	T_3	423.15	K
T_4	423.15	K	T_5	423.15	K
C_{A1}	1.2×10^4	mol/m^3	C_{B1}	100	mol/m^3
C_{C1}	100	mol/m^3	C_{D1}	100	mol/m^3
C_{A2}	1.2×10^4	mol/m^3	C_{B2}	100	mol/m^3
C_{C2}	100	mol/m^3	C_{D2}	100	mol/m^3
C_{A3}	1.2×10^4	mol/m^3	C_{B3}	100	mol/m^3
C_{C3}	100	mol/m^3	C_{D3}	100	mol/m^3
C_{A4}	1.2×10^4	mol/m^3	C_{B4}	100	mol/m^3
C_{C4}	100	mol/m^3	C_{D4}	100	mol/m^3
C_{A5}	1.2×10^4	mol/m^3	C_{B5}	100	mol/m^3
C_{C5}	100	mol/m^3	C_{D5}	100	mol/m^3

We consider white noise with signal-to-noise ratio of 20 dB and 40 dB per sample for the output measurements and process disturbances, respectively. For both the control and estimation problems we consider a $\pm\%50$ bound on the manipulated inputs and estimated

state variables with respect to their reference values. Since the hold up dynamics are non-self regulating, *i.e.* they may tend towards a different steady state or be unstable if the corresponding flow rates are chosen freely, we add three equality constraints to the optimization problems to ensure that the time derivative of the hold up (right-hand side of the hold up dynamic equations) is always proportional to the negative values of the hold up variables. The diagonal elements of the required weighting matrices are tuned by the inverse values of their corresponding reference or steady state values. Such a tuning strategy balances the importance of output regulation errors and manipulated inputs and the impact of estimation errors in the objective functions of the control and estimation problems, respectively.

We employ the interior point optimizer (IPOPT) to solve the nonlinear constrained dynamic optimization at each sampling time [89] by considering sufficiently high values of a maximum allowed number of iterations and function evaluations for the solver, to ensure that the optimizing procedure is not terminated before converging to the optimal point. Simulations are performed in $MATLAB^{®}$ using a $3.4\ GHz\ \ Intel^{®}\ Core^{TM}\ i7-6700$ processor. Figure 6.11 presents selected temporal profiles of the output regulation errors for the combined DMPC and DMHE using the optimal decomposition for the control and estimation problems. During the time period of $[0\ \ 600]$, we have an open-loop process where the decision variables are constant values chosen randomly. This short open-loop period (highlighted area) is used to collect the data history to initialize the estimation problem. The resulting ensemble of data is used by the MHE to estimate the system state variables which are then applied by the MPC as the initial values of the of the state variables at each sampling time.

After $t = 600s$, we have closed-loop operation with combined MHE and MPC. The output regulation errors converge to zero with smooth transients and without any chattering which confirms the regulation of the system outputs at their desired reference values.

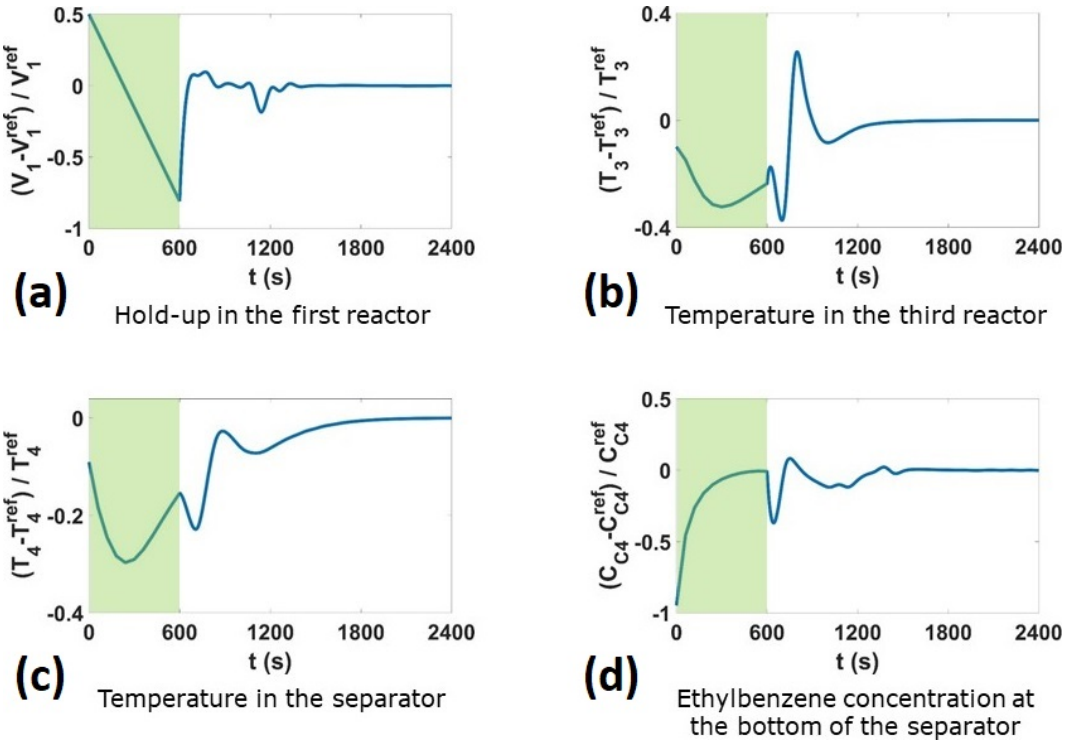


Figure 6.11: Dynamic behavior of selected output variables during open-loop, $t \in [0\ \ 600)$, and closed-loop, $t \in [600\ \ 2400]$, process operation

Figure 6.12 shows selected temporal profiles of the manipulated inputs in deviation form where the control actions remain constant between sampling times. Note that in the open-loop period, the manipulated inputs are generated randomly. After activating the output feedback control loop, we observe that the control actions gradually converge to their steady state values.

We use, as in the previous case studies, the dimensionless performance index (DPI) [83] to compare the closed-loop performance of the proposed architecture with other potential configurations

$$\text{DPI} = \text{ISE} + \text{ISC} = \int_0^{t_f} \left(\left\| y - y^{ref} \right\|_P^2 + \left\| u - u^{ref} \right\|_W^2 \right) \, dt \tag{6.25}$$

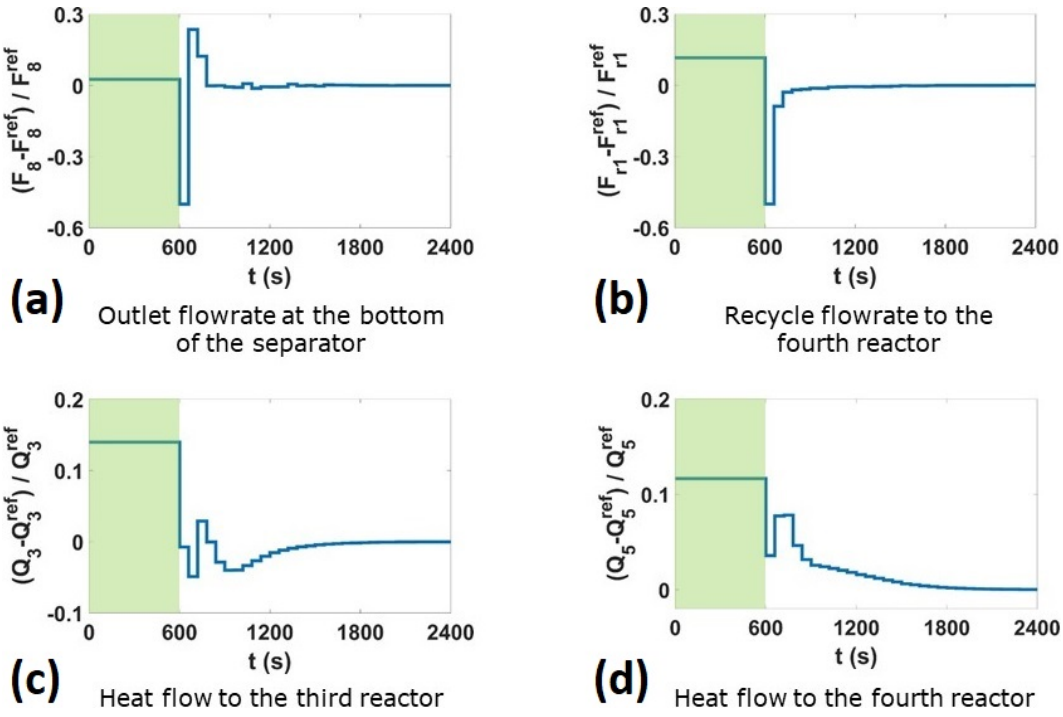


Figure 6.12: Selected manipulated inputs during open-loop, $t \in [0\ \ 600)$, and closed-loop, $t \in [600\ \ 2400]$, process operation

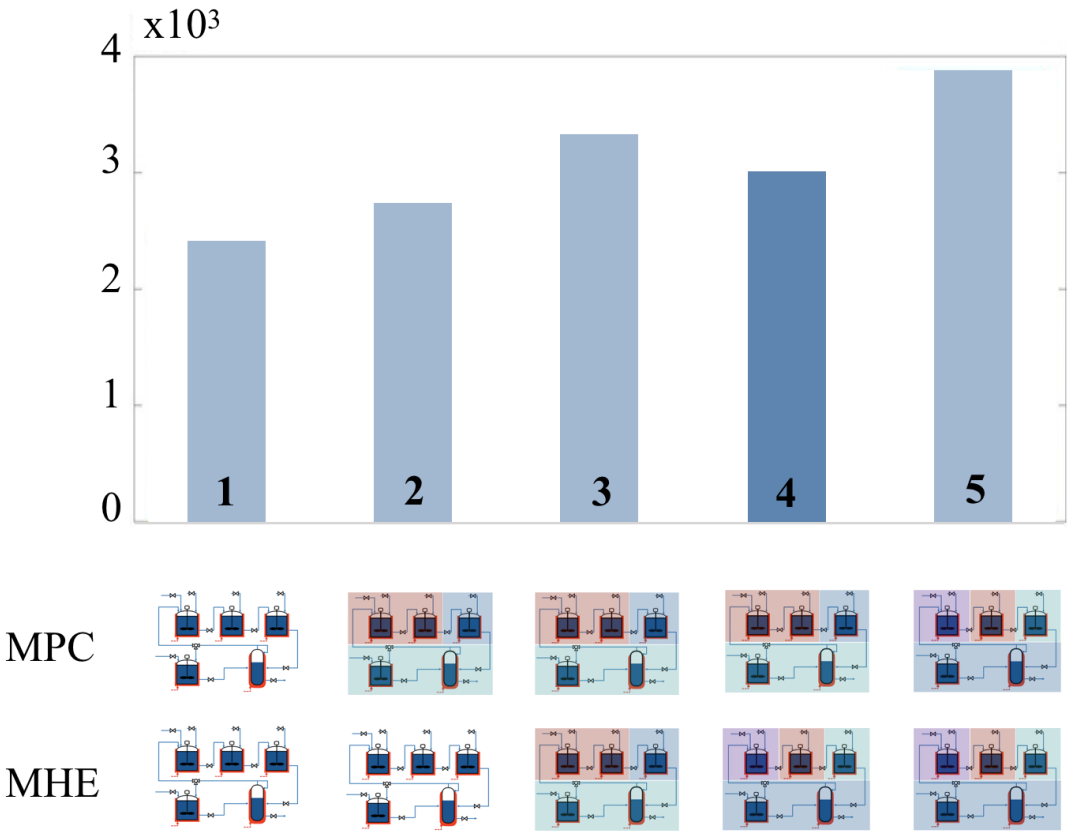


Figure 6.13: Performance index for different control and estimation architectures

where ISE and ISC indicate the integral squared errors and the squared control actions, respectively. Figure 6.13 illustrates the DPI for the proposed distributed output feedback control architecture (fourth column) compared to other potential architectures. The first column corresponds to the centralized control and estimation architecture where an individual controller/estimator is designed to compute the manipulated inputs/state variables by solving a dynamic optimization problem at each sampling time.

The second column corresponds to the DMPC with an optimal architecture combined with CMHE. The third and the fifth columns correspond to the DMPC with the optimal architecture and the DMHE with a suboptimal architecture, and the DMPC with a suboptimal architecture and the DMHE with the optimal architecture, respectively. Figure 6.14

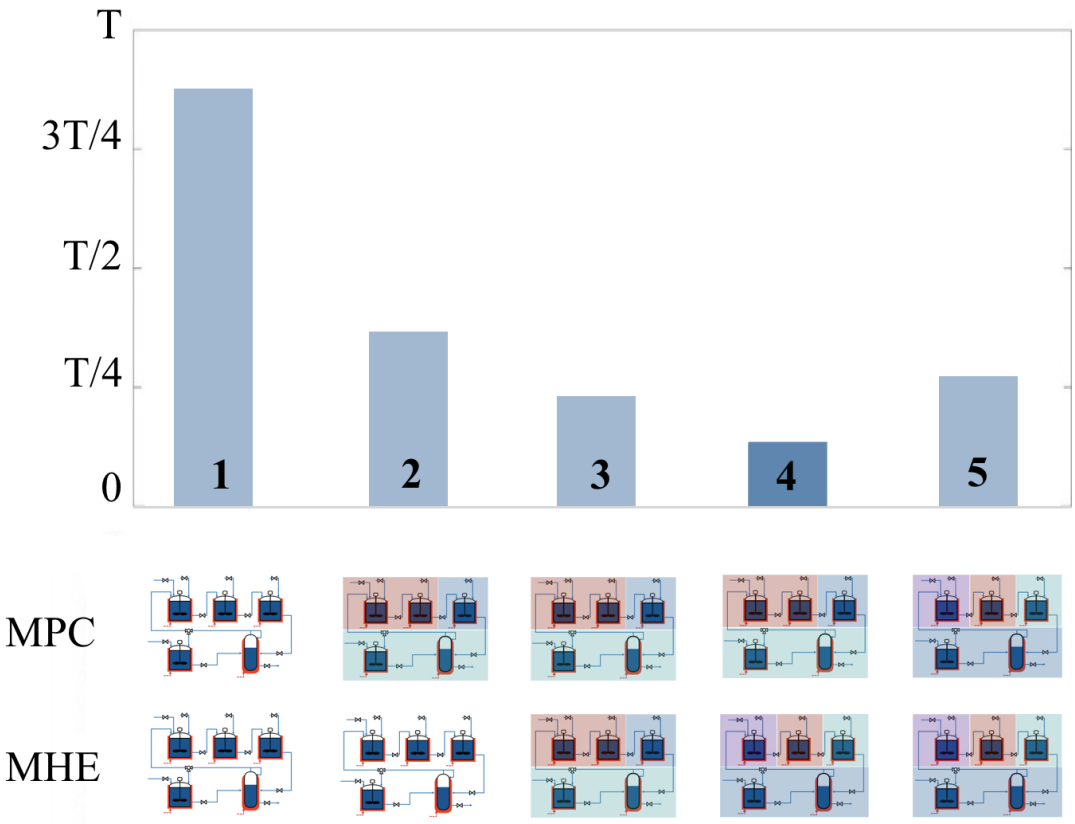


Figure 6.14: Computation times for different control and estimation architectures

demonstrates the average computation times per sampling period, T, for the same cases as in 6.13. For a combined CMPC and CMHE, it is shown that achieving the solution of the dynamic optimization problems takes almost %87 time of the sampling period. In such a case, the estimated state variables and the manipulated inputs are implemented as soon as they become available [15]. However, the time delay between the sampling time and when the corresponding states and inputs are implemented may result in instability and poor closed-loop performance. This impact has not been considered in the performance index presented in Figure 6.13. Note that the required computational effort to solve a centralized estimation/control is raised by an increase in the size and the complexity of the process system. In specific cases, it may also result in a computation time greater than the

sampling period which make the control implementation impractical [83].

By comparing the DPI and the computation time for these cases, we note a performance degradation due to distribution along with a significant reduction in the computation time. By considering the second and the third columns (both use the same optimal DMPC architecture, where one employs CMHE and the other DMHE with a suboptimal decomposition), we observe a slight reduction in the computation time and a closed-loop performance degradation. The third, the fourth, and the fifth columns use distributed estimation and control design. However, for the third and the fifth columns we use one suboptimal structure. We observe that the fourth case (red column) which employs a combined distributed estimation and control based on optimal architectures obtained by applying community detection has the best closed-loop performance (low DPI and required computation time). In this distributed output feedback control architecture, we utilized system decompositions that minimizes the interactions between the subsystems for both estimation and control problems. Such a proposed architecture enables closed-loop performance close to that of the CMPC, while reduces the computational effort significantly.

Chapter 7

Summary and Conclusions

This thesis dealt with developing a unified framework for distributed control and state estimation of generic chemical networks. The first part of this thesis presented a framework for determining the optimal network decomposition of networks comprising DPSs interconnected with LPSs. The second part of the thesis addressed the problem of obtaining optimal decompositions for distributed control as well as distributed state estimation for chemical plants consisting of LPSs.

The identification of sparsely interacting sub-networks within LPS networks using community detection algorithms that maximize modularity has been shown to be a promising approach towards obtaining the optimal decomposition for distributed control of LPS networks. The community detection algorithms are based on the equation graphs of the network, which are standard for LPSs. But most chemical plants consist of LPSs interconnected with DPSs. In Chapters 2-5 of this thesis, we extended the equation graph representation to DPSs. For this, we considered the two broad classifications of DPSs, namely, convection-reaction systems and diffusion-convection-reaction systems. The former class is modeled by first-order hyperbolic PDEs, and the latter by parabolic PDEs. We considered the three different kinds of inputs identified in the literature for these systems:

1. Boundary inputs: These input variables are the boundary conditions of the PDEs that capture the system dynamics.

2. Velocity inputs: These input variables are the flow velocities of the streams in the PDE system.

3. Distributed inputs: These input variables are functions of space and time distributed along the spatial coordinate of the PDE system.

For convection-reaction systems, we considered only linear systems, and two types of output variables

1. Boundary outputs: These are located at the boundary of the PDEs that capture the system dynamics.

2. Distributed outputs: The variables that are distributed along the spatial coordinate of the PDE system.

For diffusion-convection-reaction systems, we considered quasi-linear parabolic PDE systems, and, in addition to the aforementioned classifications of the output variables, we considered a third kind of output - the spatially varying but time-dependent output variable, e.g., the hot-spot in a tubular reactor. Velocities as output variables are not considered for either of the PDE systems, as they are not natural in this setting.

For both of these kinds of systems, we proposed a relative degree analogue that we call the Structural Interaction Parameter or SIP, which captures the strength of interactions among the variables of the DPS. In order to do this, we redefined the distributed input and output variables so as to capture their information in finite-dimensional vectors, the elements of which are functions of time alone [18]. We also redefined the boundary input and output variables to collocate the variables in case they are located at different spatial coordinates [19]. This collocation ensures that the output's dependence on the input is

captured in finite order time-derivatives of the output. For both classes of DPSs, we proposed equation graph representations. For the convection-reaction systems, we linked the length of the input to output path in the equation graph to the SIP using the properties of finite-dimensional coefficient matrices of the set of linear, first-order hyperbolic PDEs that capture the system dynamics. For the diffusion-convection-reaction system, we linked the SIP to the equation graph using the Galerkin projection of the PDEs. The SIP is related to the length of the shortest input-output path in the equation graph with the same algebraic relationship that relates relative degree and the input-output paths in LPS graphs. This allows us to use the equation graph based network decomposition algorithms for equation graphs of interconnected LPS and DPS networks to obtain the optimal decomposition for distributed plant-wide control of generic chemical networks.

The optimality of the optimal decomposition obtained using the community detection algorithms on LPS-DPS networks was verified using two case studies.

The first case study is a two stage amine gas sweetening plant (discussed in Chapter 3), which contains two absorbers in series to absorb the sour components (carbon dioxide and hydrogen sulphide), and two downstream regenerator columns to regenerate the spent solvent, along with two process stream heat-exchangers. This network was chosen due to its significance in the liquified natural gas train and the fact that plant-wide control of such a network has not yet been addressed in the literature. We observed that the optimal network decomposition gives insight into the dominant factor that determines the degree of integration among the process units in the plant, which is the reaction between the sour components of the natural gas, and is not obvious from the plant process flow diagram or the plant model. We simulated three plant-wide control problems

1. Nominal plant start-up

2. Start-up with measurement noise in all the output measurements

3. Start-up with disturbance in feed composition

We compared the performance of DMPC based on the optimal network decomposition with centralized MPC, DMPC based on a sub-optimal decomposition and decentralized control based on the optimal decomposition.

The second chemical network considered is a reaction-separation plant (discussed in Chapter 5), that consists of four tubular reactors with interstage cooling, upstream of two flash columns. An exothermic, reversible reaction occurs in the tubular reactors and the product is separated from the unreacted reactant in the flash tanks. This network is also tightly integrated by a material recycle stream and heat integration between the interstage heat-exchangers. This type of network is encountered in the production and separation of sulphuric acid [22]. We used the community detection algorithm to determine the optimal decomposition for this network. The optimal decomposition identifies the dominating factors that affect the interactions among the process units as the integration between the heat-exchangers and the material recycle from the flash tanks to the fourth reactor. We implemented DMPC on the set-point tracking control problem, and compared its performance with CMPC.

For both the case studies, we used as the measures for gauging the control performance the performance index (a measure of the deviation of the outputs from the steady-state values and the control effort required to bring the outputs to the steady-state value) and the computation time required to compute the optimum values of the input variables. We observed that DMPC consistently gives for all the control problems, a performance index that is close to that of centralized control, and far superior to that of decentralized control, with a computation time that is significantly lower than that for centralized control. We observed, also, that the performance of the DMPC on the optimal decomposition as obtained from the equation graph based community detection is superior to that on a sub-optimal decomposition obtained from the same algorithm.

The aforementioned case studies of DMPC for plant-wide control are based on the assumption that the entire state vector is available. However, in practice, the entire state is not often measurable. Hence, in chapter 6, we considered the problem of plant-wide control of an LPS network where the state needs to be estimated from limited measurements. In particular, we focused on MHE, which is a promising formulation for state estimation. MHE is based on the repeated solution of a constrained optimization problem where the decision variables are the state variables. MHE for large networks can be computationally expensive, and can compromise the closed-loop control performance. Analogous to the DMPC approach, which has shown to have lower computational cost than CMPC, we formulated a combined distributed MPC and state estimation problem. For this we used two equation graphs, one for the control architecture and one for estimation architecture. The graph for the control architecture was the same as the graph associated with relative degree for LPS variables, and captured the connectivity of the input variables to the output variables through the state variables. The graph for the estimator captured in a similar manner the connectivity of the measured variables and the state variable vector. A community detection algorithm that uses the spectral decomposition of the modularity matrix corresponding to the network [40] was used to obtain the optimum network decompositions for estimation and control. As a case study, a combination of DMHE and DMPC based on these decompositions was then used to address the output regulation problem for a benzene alkylation plant that consisted of four CSTRs and one flash column. Benzene and ethylene are fed to the first CSTR in a series of three CSTRs. Additional benzene is fed to the two downstream CSTRs. The reaction in these CSTRs produces ethylbenzene. The fourth CSTR produces ethylbenzene from diethylbenze, which is a byproduct of the first CSTRs in the series of three CSTRs. The flash tank separates ethylbenzene from ethylene. It was observed that the optimum control configuration was different from the optimum architecture for state estimation. This would be the expectation in a typical

case of combined DMHE and DMPC since the equation graphs on which the decompositions are based are different. It was also observed that the decompositions, which were not intuitive or obvious from the process flow diagram or model, provided insight into the dominant interactions in the network. Several combinations of the control and estimation architectures, including fully centralized and sub-optimal architectures, were also simulated for the combined estimation and control problem to compare the closed-loop performance with the DMHE-DMPC based on the respective optimal architectures. The parameters for comparison were the same as for the previous case studies, i.e., the performance index and the computation time to converge upon the decision variables (which were the state variables for the estimation problem and the input variables for the control problem). It was observed in the simulations that the proposed distributed output feedback control design based on the optimal decompositions minimizes the interactions between the distributed local estimators and controllers, and enables a closed-loop performance close to that of the centralized architecture while reducing the computational effort significantly.

These studies highlight the importance of choosing the optimal network decomposition for distributed control as well as distributed estimation and validate the relationship established in this thesis between the equation graphs of the network variables and the strength of the structural interactions as captured by the SIP for DPSs.

Chapter 8

Future Work

8.1 Structurally Controllable Decompositions

A limitation of the existing community detection algorithms for network decomposition is that the communities or the subsystems in the decomposition are not necessarily controllable, and a post-processing step is required to check for the controllability of the decomposition. If the decomposition with the maximum modularity is found to be uncontrollable, then the controllability check is performed on the sub-optimal decompositions until a controllable decomposition is obtained. In the future, structural controllability criteria could be incorporated into the community detection algorithm to eliminate the need for this post-processing step.

8.1.1 LPS Network Decomposition

For LPSs, in the literature there exist equation graph based criteria related to the rank of the system's controllability gramian [113–115], namely the input connectivity condition and the no-dilation condition, which could be incorporated in the community detection algorithm for ensuring that the decompositions are controllable. The input connectivity

condition requires the identification of the strongly connected sets in the equation graph, which could be done by analysing the elements of the adjacency matrix [115]. The no-dilation requires the calculation of the maximum matching in the bipartite graph that captures the connectivity between the input and state variables. The bipartite graph could be determined from the equation graph, and the algorithms for determining the size of the maximum matching [116, 117] could be used to check if the no-dilation condition is satisfied by a certain decomposition.

8.1.2 DPS Network Decomposition

The assessment of controllability and controller design for PDE systems has been extensively studied (see for example, [92, 94, 118–123] and the references therein). Even so, controllability of PDEs is an open problem [118]. To the best of our knowledge, there don't exist results for assessing structural controllability in PDE systems.

The work in [124] discusses controllability for scalar PDE systems as well as non-scalar PDE systems, with boundary and distributed actuations. For some classes of PDE systems, with certain inherent model attributes (e.g., autonomous systems with diagonal diffusion matrix, non-autonomous systems with identity diffusion matrix etc.), the work in [124] reports controllability criteria that are related to the corresponding Kalman matrix or operator. These criteria could be the starting point in developing a graph based framework that forms the basis for assessing structural controllability in PDE systems.

8.2 Structurally Observable Decompositions

Similar to the incorporation of structural controllability criteria in the network decomposition algorithms, structural observability criteria could be incorporated in the algorithm to determine decompositions that are controllable as well as observable. To this

end, using the input connectivity and no-dilation conditions along with the controllability-observability duality, the equivalent observability criteria could be determined for the networks of LPSs. The input connectivity criterion on the dual of a given network would translate to an output connectivity criterion in the original network, and would be based on the corresponding adjacency matrix. The no-dilation condition on the dual network would be related to the cardinality of the maximum matching in the original network's bipartite graph that relates the output variables to the state variables. These conditions could be incorporated in the decomposition algorithm in a similar way as the incorporation of the controllability criteria.

8.3 DPS Graphs based on Comprehensive Interaction

The work in this thesis [82, 93, 125–127] focused on extending the equation graph representation to DPSs. To this end, an analogue of relative degree was saught that captures the structural distance of the controlled outputs from the manipulated inputs for DPSs. Although the equation graph representations and their relationship with the SIP have shown to be a promising basis for community detection in generic PDE systems, it remains to be seen whether the quantification of the strength of interaction among the variables of the DPS based on factors other than (or, in addition to) the structural interaction would result in different decompositions for distributed control, and if those architectures can form the basis of computationally superior controller. The identification of relevant time-constants related to the system (e.g., the residence time of the fluid in the column) or the relative characteristic time or length scales (e.g. Damköhler number, Schmidt number) and their incorporation in the equation graphs (e.g., as weights on the edges) or in the decomposition algorithm might enable the use of community detection algorithms to identify communities of variables that are strongly interacting, but whose strength of interaction is

not accurately captured by the SIP.

Chapter 9

Bibliography

[1] P. D. Christofides, J. Liu, and D. M. De La Pena. Networked and distributed predictive control. SpringerVerlag: London,, 2011.

[2] D. B. Pourkargar, A. Almansoori, and P. Daoutidis. Impact of decomposition on distributed model predictive control: A process network case study. *Industrial & Engineering Chemistry Research*, 56(34):9606–9616, 2017.

[3] P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.

[4] S. Heo, W. A. Marvin, and P. Daoutidis. Automated synthesis of control configurations for process networks based on structural coupling. *Chemical Engineering Science*, 136:76–87, 2015.

[5] S. Heo and P. Daoutidis. Control-relevant decomposition of process networks via optimization-based hierarchical clustering. *AIChE Journal*, 62(9):3177–3188, 2016.

[6] L. Kang, W. Tang, Y. Liu, and P. Daoutidis. Control configuration synthesis using agglomerative hierarchical clustering: A graph-theoretic approach. *Journal of Process Control*, 46:43–54, 2016.

[7] S. S. Jogwar and P. Daoutidis. Community-based synthesis of distributed control architectures for integrated process networks. *Chemical Engineering Science*, 172:434–443, 2017.

[8] W. Tang and P. Daoutidis. Network decomposition for distributed control through community detection in input–output bipartite graphs. *Journal of Process Control*, 64:7–14, 2018.

[9] W. Tang, D. B. Pourkargar, and P. Daoutidis. Relative time-averaged gain array (rtaga) for distributed control-oriented network decomposition. *AIChE Journal*, 64(5):1682–1690, 2018.

[10] P. Daoutidis and C. Kravaris. Structural evaluation of control configurations for multivariable nonlinear processes. *Chemical Engineering Science*, 47(5):1091–1107, 1992.

[11] X. Yin, K. Arulmaran, J. Liu, and J. Zeng. Subsystem decomposition and configuration for distributed state estimation. *AIChE Journal*, 62(6):1995–2003, 2016.

[12] K. M. Hangos and Z. Tuza. Optimal control structure selection for process systems. *Computers & Chemical Engineering*, 25(11):1521–1536, 2001.

[13] T. Schné and K. M. Hangos. Decentralised controller structure design and retrofit of process systems based on graph theory. *International Journal of Systems Science*, 42(6):1023–1033, 2011.

[14] M. Ellis and P. D. Christofides. Selection of control configurations for economic model predictive control systems. *AIChE Journal*, 60(9):3230–3242, 2014.

[15] M. Guay, V. Adetola, and D. DeHaan. *Robust and adaptive model predictive control of nonlinear systems*. IET, London, 2015.

[16] X. Yin, B. Decardi-Nelson, and J. Liu. Subsystem decomposition and distributed moving horizon estimation of wastewater treatment plants. *Chemical Engineering Research and Design*, 134:405–419, 2018.

[17] X. Yin and J. Liu. Subsystem decomposition of process networks for simultaneous distributed state estimation and control. *AIChE Journal*, 2018.

[18] P. D. Christofides and P. Daoutidis. Feedback control of hyperbolic pde systems. *AIChE Journal*, 42(11):3063–3086, 1996.

[19] P. K. Gundepudi and J. C. Friedly. Velocity control of hyperbolic partial differential equation systems with single characteristic variable. *Chemical Engineering Science*, 53(24):4055–4072, 1998.

[20] A. Maidi, M. Diaf, and J. Corriou. Boundary control of a parallel-flow heat exchanger by input–output linearization. *Journal of Process Control*, 20(10):1161–1174, 2010.

[21] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[22] A. A. Kiss, C. S. Bildea, and J. Grievink. Dynamic modeling and process optimization of an industrial sulfuric acid plant. *Chemical Engineering Journal*, 158(2):241–249, 2010.

[23] J. Rawlings and B. Bakshi. Particle filtering and moving horizon estimation. *Computers & Chemical Engineering*, 30(10-12):1529–1541, 2006.

[24] D. Copp and J. Hespanha. Simultaneous nonlinear model predictive control and state estimation. *Automatica*, 77:143–154, 2017.

[25] C.V. Rao, J.B. Rawlings, and D.Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2):246–258, 2003.

[26] H. Cui and E. W. Jacobsen. Performance limitations in decentralized control. *Journal of Process Control*, 12(4):485–494, 2002.

[27] A. Kumar and P. Daoutidis. Nonlinear dynamics and control of process systems with recycle. *Journal of Process Control*, 12(4):475–484, 2002.

[28] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems*, 22(1):44–52, 2002.

[29] R. Olfati-Saber and R. M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. *IFAC Proceedings Volumes*, 35(1):495–500, 2002.

[30] F. Bullo, J. Cortés, and S. Martinez. *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[31] G. Weiss. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 1999.

[32] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, 22(3):52–67, 2002.

[33] J. Liu, D. M. De La Peña, and P. D. Christofides. Distributed model predictive control of nonlinear process systems. *AIChE Journal*, 55(5):1171–1184, 2009.

[34] M. J. Tippett and J. Bao. Dissipativity based distributed control synthesis. *Journal of Process Control*, 23(5):755–766, 2013.

[35] R. Scattolini. Architectures for distributed and hierarchical model predictive control–a review. *Journal of Process Control*, 19(5):723–731, 2009.

[36] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460–469, 2010.

[37] G. Antonelli. Interconnected dynamic systems: An overview on distributed control. *IEEE Control Systems*, 33(1):76–88, 2013.

[38] N. Motee and B. Sayyar-Rodsari. Optimal partitioning in distributed model predictive control. In *Proceedings of the American Control Conference, 2003*, volume 6, pages 5300–5305. IEEE, 2003.

[39] W. Al-Gherwi, H. Budman, and A. Elkamel. Selection of control structure for distributed model predictive control in the presence of model errors. *Journal of Process Control*, 20(3):270–284, 2010.

[40] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Physical review letters*, 100(11):118703, 2008.

[41] A. Maidi and J. P. Corriou. Boundary control of nonlinear distributed parameter systems by input-output linearization. *IFAC Proceedings Volumes*, 44(1):10910–10915, 2011.

[42] T. E. Rufford, S. Smart, G. C. Y. Watson, B. F. Graham, J. Boxall, J. C. Diniz Da Costa, and E. F. May. The removal of CO2 and N2 from natural gas: a review of conventional and emerging process technologies. *Journal of Petroleum Science and Engineering*, 94:123–154, 2012.

[43] S. S. Dashti, A. Shariati, and M. R. K. Nikou. Sensitivity analysis for selection of an optimum amine gas sweetening process with minimum cost requirement. *Asia-Pacific Journal of Chemical Engineering*, 10(5):709–715, 2015.

[44] B. Mandal and S. S. Bandyopadhyay. Simultaneous absorption of CO2 and H2S into aqueous blends of n-methyldiethanolamine and diethanolamine. *Environmental Science & Technology*, 40(19):6076–6084, 2006.

[45] M. Wang, A. Lawal, P. Stephenson, J. Sidders, and C. Ramshaw. Post-combustion CO2 capture with chemical absorption: a state-of-the-art review. *Chemical Engineering Research and Design*, 89(9):1609–1624, 2011.

[46] A. B. Rao, E. S. Rubin, and M. B. Berkenpas. An integrated modeling framework for carbon management technologies. *Department of Engineering and Public Policy, Pittsburgh, PA*, 15213:3890, 2004.

[47] S. Bishnoi and G. Rochelle. Thermodynamics of piperazine/methyldiethanolamine/water/carbon dioxide. *Industrial & Engineering Chemistry Research*, 41(3):604–612, 2002.

[48] F. Seibert, E. Chen, M. Perry, S. Briggs, R. Montgomery, and G. Rochelle. Ut/srp CO2 capture pilot plantoperating experience and procedures. *Energy Procedia*, 4:1616–1623, 2011.

[49] A. Aboudheir, P. Tontiwachwuthikul, A. Chakma, and R. Idem. Kinetics of the reactive absorption of carbon dioxide in high CO2-loaded, concentrated aqueous

monoethanolamine solutions. *Chemical Engineering Science*, 58(23):5195–5210, 2003.

[50] P. D. Vaidya and E. Y. Kenig. CO2-alkanolamine reaction kinetics: A review of recent studies. *Chemical Engineering & Technology*, 30(11):1467–1474, 2007.

[51] J. D. Lawson and A. W. Garst. Gas sweetening data: equilibrium solubility of hydrogen sulfide and carbon dioxide in aqueous monoethanolamine and aqueous diethanolamine solutions. *Journal of Chemical and Engineering Data*, 21(1):20–30, 1976.

[52] S. Chi and G. Rochelle. Oxidative degradation of monoethanolamine. *Industrial & Engineering Chemistry Research*, 41(17):4178–4186, 2002.

[53] J. Davis and G. Rochelle. Thermal degradation of monoethanolamine at stripper conditions. *Energy Procedia*, 1(1):327–333, 2009.

[54] R. M. Davidson. *Post-combustion carbon capture from coal fired plants: solvent scrubbing*. IEA Clean Coal Centre London, 2007.

[55] A. Aroonwilas and A. Veawab. Integration of CO2 capture unit using single and blended amines into supercritical coal-fired power plants: implications for emission and energy management. *International Journal of Greenhouse Gas Control*, 1(2):143–150, 2007.

[56] S. Reddy, J. Scherffius, S. Freguia, and C. Roberts. Fluor's econamine fg plus sm technology. In *Proceedings of the second annual conference on carbon sequestration*, pages 5–8, 2003.

[57] V. Abkhiz and I. Heydari. Comparison of amine solutions performance for gas sweetening. *Asia-Pacific Journal of Chemical Engineering*, 9(5):656–662, 2014.

[58] M. A. Pacheco and G. Rochelle. Rate-based modeling of reactive absorption of CO2 and H2S into aqueous methyldiethanolamine. *Industrial & Engineering Chemistry Research*, 37(10):4107–4117, 1998.

[59] W. J. DeCoursey. Enhancement factors for gas absorption with reversible reaction. *Chemical Engineering Science*, 37(10):1483–1489, 1982.

[60] W. J. DeCoursey and R. W. Thring. Effects of unequal diffusivities on enhancement factors for reversible and irreversible reaction. *Chemical engineering science*, 44(8):1715–1721, 1989.

[61] D. A. Glasscock and G. Rochelle. Approximate simulation of CO2 and H2S absorption into aqueous alkanolamines. *AIChE Journal*, 39(8):1389–1397, 1993.

[62] R. A. Tomcej, F. D. Otto, H. A. Rangwala, and B. R. Morrell. Tray design for selective absorption. In *Laurance Reid Gas Conditioning Conference, Norman, OK*, 1987.

[63] M. S. Sivasubramanian. Heat and mass transfer rate approach for the simulation and design of acid gas treating units. Technical report, Clarkson University, Potsdam, NY (USA), 1985.

[64] T. R. Carey, J. E. Hermes, and G. Rochelle. A model of acid gas absorption/stripping using methyldiethanolamine with added acid. *Gas separation & purification*, 5(2):95–109, 1991.

[65] R. Taylor and R. H. Weiland. The rate approach to the computer-aided design and simulation of multicomponent separation processes. *Proceedings of the International Congress CEF*, 87:479.

[66] S. Freguia and G. Rochelle. Modeling of CO2 capture by aqueous monoethanolamine. *AIChE Journal*, 49(7):1676–1686, 2003.

[67] S. Ziaii, G. Rochelle, and T. F. Edgar. Dynamic modeling to minimize energy use for CO2 capture in power plants by aqueous monoethanolamine. *Industrial & Engineering Chemistry Research*, 48(13):6105–6111, 2009.

[68] A. Lawal, M. Wang, P. Stephenson, and H. Yeung. Dynamic modelling of CO2 absorption for post combustion capture in coal-fired power plants. *Fuel*, 88(12):2455–2462, 2009.

[69] H. M. Kvamsdal, J. P. Jakobsen, and K. A. Hoff. Dynamic modeling and simulation of a CO2 absorber column for post-combustion co 2 capture. *Chemical Engineering and Processing: Process Intensification*, 48(1):135–144, 2009.

[70] A. Lawal, M. Wang, P. Stephenson, and H. Yeung. Dynamic modeling and simulation of CO2 chemical absorption process for coal-fired power plants. *Computer Aided Chemical Engineering*, 27:1725–1730, 2009.

[71] J. C. Polasek, G. A. Inglesias-Silva, and J. A. Bullin. Using mixed amine solutions for gas sweetening. In *Proceedings of the annual convention-gas processors association*, pages 58–58. Gas Processors Association, 1992.

[72] B. Sohbi, M. Meakaff, M. Emtir, and M. Elgarni. The using of mixing amines in an industrial gas sweetening plant. *World Academy of Science, Engineering and Technology*, 31(1):301–305, 2007.

[73] R. Abdulrahman and I. Sebastine. Effect of lean amine temperature on amine gas sweetening: case study and simulation. *Chemistry and Technology of Fuels and Oils*, 49(4):293–297, 2013.

[74] Z. Aliabad and S. Mirzaei. Removal of CO2 and H2S using aqueous alkanolamine solutions. *World Academy of Science, Engineering and Technology, International Journal of Chemical, Molecular, Nuclear, Materials and Metallurgical Engineering*, 3(1):50–59, 2009.

[75] K. Najim and V. Ruiz. Long-range predictive control of an absorption packed column. *Applied Mathematical Modelling*, 19(1):39–45, 1995.

[76] A. Bedelbayev, T. Greer, and B. Lie. Model based control of absorption tower for CO2 capturing. *Telemark University College, Porsgrunn, Norvège*, page 11, 2008.

[77] M. Schach, R. Schneider, H. Schramm, and J. Repke. Control structure design for CO2-absorption processes with large operating ranges. *Energy Technology*, 1(4):233–244, 2013.

[78] Y. Lin, T. Pan, D. S. Wong, S. Jang, Y. Chi, and C. Yeh. Plantwide control of CO2 capture by absorption and stripping using monoethanolamine solution. *Industrial & Engineering Chemistry Research*, 50(3):1338–1345, 2010.

[79] L. T. Biegler. New nonlinear programming paradigms for the future of process optimization. *AIChE Journal*, 63(4):1178–1193, 2017.

[80] Y. Fang and A. Armaou. Carleman approximation based quasi-analytic model predictive control for nonlinear systems. *AIChE Journal*, 62(11):3915–3929, 2016.

[81] V. M. Zavala and L. T. Biegler. The advanced-step nmpc controller: Optimality, stability and robustness. *Automatica*, 45(1):86–93, 2009.

[82] M. Moharir, L. Kang, A. Almansoori, and P. Daoutidis. Graph representation and decomposition of ode/hyperbolic pde systems. *Computers & Chemical Engineering*, 106:532–543, 2017.

[83] D. B. Pourkargar, A. Almansoori, and P. Daoutidis. Comprehensive study of decomposition effects on distributed output tracking of an integrated process over a wide operating range. *Chemical Engineering Research and Design*, 2018.

[84] A. B. Tamadher. Train-3 acid gas removal unit performance. In *SOGAT Proceedings of 8th International Conference, Abu Dhabi, UAE*, volume 21, 2012.

[85] W. L. McCabe, J. C. Smith, and P. Harriott. *Unit operations of chemical engineering*, volume 5. McGraw-Hill New York, 1993.

[86] E. L. Cussler. *Diffusion: mass transfer in fluid systems*. Cambridge university press, 2009.

[87] R. Sander. Henry's law constants in nist chemistry webbook, nist standard reference database number 69, eds. *PJ Linstrom and WG Mallard, National Institute of Standards and Technology, Gaithersburg MD*, 20899, 2000.

[88] W. Tang, A. Allman, D. B. Pourkargar, and P. Daoutidis. Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection. *Computers & Chemical Engineering*, (111):43–54, 2018.

[89] L.T. Biegler. *Nonlinear Programming: Concepts, Algorithms and Applications to Chemical Processes*. SIAM, Philadelphia, 2010.

[90] D. B. Pourkargar, A. Almansoori, and P. Daoutidis. Distributed model predictive control of process networks: Impact of control architecture. *IFAC-PapersOnLine*, 50(1):12452–12457, 2017.

[91] J. C. Friedly. *Dynamic behavior of processes*. Prentice-Hall, 1972.

[92] P. D. Christofides. *Nonlinear and robust control of PDE systems: Methods and applications to transport-reaction processes*. Springer Science & Business Media, 2012.

[93] M. Moharir, D. B. Pourkargar, A. Almansoori, and P. Daoutidis. Graph representation and distributed control of diffusion-convection-reaction system networks. *Chemical Engineering Science*, 2019.

[94] C. Antoniades and P. D. Christofides. Studies on nonlinear dynamics and control of a tubular reactor with recycle. *Nonlinear Analysis-Theory Methods and Applications*, 47(9):5933–5944, 2001.

[95] W. L. Luyben. *Process modeling, simulation and control for chemical engineers*. McGraw-Hill Higher Education, 1989.

[96] J.B. Rawlings, D.Q. Mayne, and M. Diehl. *Model predictive control: Theory, computation, and design*. Nob Hill Publishing, Madison, 2017.

[97] M. Morari and J.H. Lee. Model predictive control: Past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, 1999.

[98] G. Pannocchia, J.B. Rawlings, and S.J. Wright. Fast, large-scale model predictive control by partial enumeration. *Automatica*, 43(5):852–860, 2007.

[99] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2010.

[100] R. Lopez-Negrete, F.J. DAmato, L.T. Biegler, and A. Kumar. Fast nonlinear model predictive control: Formulation and industrial process applications. *Computers & Chemical Engineering*, 51:55–64, 2013.

[101] J. Jaschke, X. Yang, and L.T. Biegler. Fast economic model predictive control based on NLP-sensitivities. *Journal of Process Control*, 24(8):1260–1272, 2014.

[102] P. Daoutidis, A. Allman, S. Khatib, M. Moharir, M. Palys, D. B. Pourkargar, and W. Tang. Distributed decision making for intensified process systems. *Current Opinion in Chemical Engineering*, 2019.

[103] R. Scattolini. Architectures for distributed and hierarchical model predictive control: A review. *Journal of Process Control*, 19:723–731, 2009.

[104] E. Camponogara, D. Jia, B.H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22:44–52, 2002.

[105] P.D. Christofides, J. Liu, and D. Munoz de la Pena. *Networked and Distributed Predictive Control*. Springer-Verlag, London, 2011.

[106] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460–469, 2010.

[107] M.J. Tippett and J. Bao. Distributed model predictive control based on dissipativity. *AIChE J.*, 59(3):787–804, 2013.

[108] E. Khlebnikova, A. Bekker, E. Ivashkina, I. Dolganova, and E. Yurev. Thermodynamic analysis of benzene alkylation with ethylene. *Procedia Chemistry*, 15:42–48, 2015.

[109] N. Hansen, T. Bruggemann, A.T. Bell, and F.J. Keil. Theoretical investigation of benzene alkylation with ethene over H-ZSM-5. *Journal of Physical Chemistry C*, 112(39):15402–15411, 2008.

[110] H. You, W. Long, and Y. Pan. The mechanism and kinetics for the alkylation of benzene with ethylene. *Petroleum Science & Technology*, 24(9):1079–1088, 2006.

[111] W.J. Lee and Froment G.F. Ethylbenzene dehydrogenation into styrene: Kinetic modeling and reactor simulation. *Industrial & Engineering Chemistry Research*, 47(23):9183–9194, 2008.

[112] H. Ganji, J.S. Ahari, A. Farshi, and M. Kakavand. Modelling and simulation of benzene alkylation process reactors for production of ethylbenzene. *Petroleum & Coal*, 46(1):55–63, 2004.

[113] C. T. Lin. Structural controllability. *IEEE Transactions on Automatic Control*, 19(3):201–208, 1974.

[114] C. Commault and J. M. Dion. Controllability through input addition for graph-based systems. *IFAC Proceedings Volumes*, 46(2):522–527, 2013.

[115] C. Commault and J. M. Dion. Input addition and leader selection for the controllability of graph-based systems. *Automatica*, 49(11):3322–3328, 2013.

[116] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[117] J. E. Hopcroft and R. M. Karp. An nˆ5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.

[118] M. Krstic and A. Smyshlyaev. *Boundary control of PDEs: A course on backstepping designs*, volume 16. Siam, 2008.

[119] M. Krstic. *Delay compensation for nonlinear, adaptive, and PDE systems*. Springer, 2009.

[120] R. F. Curtain. Model reduction for control design for distributed parameter systems. In *Research directions in distributed parameter systems*, pages 95–121. SIAM, 2003.

[121] R. F. Curtain and H. Zwart. *An introduction to infinite-dimensional linear systems theory*, volume 21. Springer Science & Business Media, 2012.

[122] B. Bamieh, F. Paganini, and M. A. Dahleh. Distributed control of spatially invariant systems. *IEEE Transactions on Automatic Control*, 47(7):1091–1107, 2002.

[123] P. D. Christofides. Robust control of parabolic pde systems. *Chemical Engineering Science*, 53(16):2949–2965, 1998.

[124] F. Ammar-Khodja, A. Benabdallah, M. González Burgos, and M. de la Luz de Teresa de Oteyza. Recent results on the controllability of linear coupled parabolic problems: a survey. *Mathematical Control and Related Fields, 1 (3), 267-306.*, 2011.

[125] M. Moharir, Pourkargar D.B., Almansoori A., and Daoutidis P. Distributed model predictive control of an amine gas sweetening plant. *Industrial & Engineering Chemistry Research*, 57:13103–13115, 2018.

[126] M. Moharir, D. B. Pourkargar, A. Almansoori, and P. Daoutidis. Decomposition and distributed control of integrated lumped and distributed parameter process networks. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2908–2913. IEEE, 2018.

[127] L. Kang, M. Moharir, A. Almansoori, and P. Daoutidis. Modularity-based control structure selection for process networks: An extension to distributed parameter systems. In *2017 American Control Conference (ACC)*, pages 1145–1150. IEEE, 2017.

[128] S. Roman. The formula of faa di bruno. *The American Mathematical Monthly*, 87(10):805–809, 1980.

Appendices

Appendix A: Theorem 1 Proof

For notational simplicity, we drop the subscripts of the SIP (σ_{ij}) and the length of the path (l_{ij}).

The length of the shortest path in the digraph from an input v_β to an output y_α is l, if the path traverses exactly $l-1$ state variables between v_β to y_α and hence, there exist a set of distinct indices $\{\alpha, k_2, \ldots, k_{l-1}, \beta\}$ such that:

$$h_{\alpha k_2} b_{k_2 k_3} \ldots b_{k_{l-1}\beta} \left.\frac{\partial x_\beta}{\partial z}\right|_{z=L} \neq 0 \tag{1}$$

and for any $p < l-1$, for all possible sets of indices $\{\alpha, k_1, \ldots, k_p, \beta\}$,

$$h_{\alpha k_1} b_{k_1 k_2} \ldots b_{k_p \beta} \left.\frac{\partial x_\beta}{\partial z}\right|_{z=L} = 0 \tag{2}$$

Given that the output is

$$\mathbf{y} = \mathbf{H}\mathbf{x}|_{z=L} \tag{3}$$

and the system dynamics are

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{A}\frac{\partial \mathbf{x}}{\partial z} + \mathbf{B}\mathbf{x} \tag{4}$$

the α^{th} element of $\mathbf{y}$, y_α is given by

$$y_\alpha = \sum_{k_1=1}^{n_x} h_{\alpha k_1} x_{k_1}|_{z=L} \tag{5}$$

The first time derivative of y_α is given by

$$\frac{dy_\alpha}{dt} = \sum_{k_1=1}^{n_x} h_{\alpha k_1} \left.\frac{\partial x_{k_1}}{\partial t}\right|_{z=L} \tag{6}$$

Substituting for $\partial x_{k_1}/\partial t$ in equation (6) from equation (4), we get

$$\frac{dy_\alpha}{dt} = \sum_{k_1=1}^{n_x} h_{\alpha k_1} \left\{ v_{k_1} \frac{\partial x_{k_1}}{\partial z} + \sum_{k_2=1}^{n_x} b_{k_1 k_2} x_{k_2} \right\} \Bigg|_{z=L} \tag{7}$$

Assuming,

$$\left. \frac{\partial x_{k_1}}{\partial z} \right|_{z=L} \neq 0 \quad \forall k_1 \in \{1, n_x\} \tag{8}$$

the SIP of y_α with respect to x_β is 1 if

$$h_{\alpha\beta} \neq 0 \tag{9}$$

If $h_{\alpha\beta} = 0$, then we calculate the second time derivative of y_α:

$$\frac{d^2 y_\alpha}{dt^2} = \sum_{k_1=1}^{n_x} h_{\alpha k_1} \left[\sum_{k_2=1}^{n_x} b_{k_1 k_2} v_{k_2} \frac{\partial x_{k_2}}{\partial z} \right] \Bigg|_{z=L} + \text{ other terms} \tag{10}$$

The SIP of y_α with respect to v_β is 2 if

$$h_{\alpha k_1} b_{k_1 \beta} \left. \frac{\partial x_\beta}{\partial z} \right|_{z=L} \neq 0 \tag{11}$$

Generalizing from the above derivations, the SIP of y_α with respect to v_β is σ if there is a set of distinct indices $\{k_2, ..., k_\sigma\}$ such that,

$$h_{\alpha k_2} b_{k_2 k_3} \ldots b_{k_\sigma \beta} \left. \frac{\partial x_\beta}{\partial z} \right|_{z=L} \neq 0 \tag{12}$$

But then, from equation (1) it can be concluded that the length of the shortest path is $\sigma + 1$, which establishes the result.

Appendix B: Theorem 2 Proof

Consider a SISO first-order hyperbolic PDE system with distributed actuation for which the input is $u(z,t)$ and the output is $y(z,t)$. Hence, $\mathbf{G}$ and $\mathbf{H}$ are a column and a row vector respectively, henceforth represented as $\mathbf{g}$ and $\mathbf{h}^T$.

Defining an operator $f_i(x)$ as:

$$f_i(x) = a_{ii}\frac{\partial x_i}{\partial z} + \sum_{j=1}^{n} b_{ij}x_j \tag{13}$$

where a_{ii} are the diagonal elements of $\mathbf{A}$, and b_{ij} are the $(i,j)^{th}$ elements of $\mathbf{B}$, yields the following representation of the system dynamics:

$$\frac{\partial x_i}{\partial t} = f_i(x) + g_i u(z,t) \qquad \forall i \in \{1,n\}$$

$$y(z,t) = \mathbf{h}^T\mathbf{x} \tag{14}$$

It must be noted that, as was the case for velocity actuation, if the length of the shortest path in the digraph from $\hat{\mathbf{u}}(t)$ to $\hat{\mathbf{y}}(t)$ is l, then the path traverses exactly $l-1$ state variables between $\hat{\mathbf{u}}(t)$ to $\hat{\mathbf{y}}(t)$ and hence, there exists a set of distinct indices $\{\mu_1,\mu_2,\mu_3,\ldots,\mu_{l-1}\}$ such that:

$$h_{\mu_{l-1}}\frac{\partial f_{\mu_{l-1}}}{\partial x_{\mu_{l-2}}}\frac{\partial f_{\mu_{l-2}}}{\partial x_{\mu_{l-3}}}\cdots\frac{\partial f_{\mu_2}}{\partial x_{\mu_1}}g_{\mu_1} \neq 0 \tag{15}$$

The SIP between $\hat{\mathbf{y}}(t)$ and $\hat{\mathbf{u}}(t)$ is σ if [18],

$$\mathbf{h}^T\left(\mathbf{A}\frac{\partial}{\partial z} + \mathbf{B}\right)^{\sigma-1}\mathbf{g} \neq 0 \tag{16}$$

and

$$\mathbf{h}^T\left(\mathbf{A}\frac{\partial}{\partial z} + \mathbf{B}\right)^{\sigma-k}\mathbf{g} = 0 \qquad \forall \;\; k \in \{2,\sigma\} \tag{17}$$

From the definitions of **A**, **B** and equation (13), a matrix **J** can be defined as:

$$\mathbf{J} = \left(\mathbf{A}\frac{\partial}{\partial z} + \mathbf{B}\right) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{18}$$

Substituting **J** into equations (16) and (17), implies that if the SIP is σ then:

$$\mathbf{h}^T(\mathbf{J})^{\sigma-1}\mathbf{g} \neq 0 \tag{19}$$

$$\mathbf{h}^T(\mathbf{J})^{\sigma-k}\mathbf{g} = 0 \qquad \forall \quad k \in \{2, \sigma\} \tag{20}$$

Initially, it is assumed that only one element in **h** and **g** is non-zero. That is, there are indices α and β such that $\mathbf{h} = [0 \quad 0 \quad \ldots \quad h_\alpha \quad \ldots \quad 0 \quad 0 \quad]^T$, $\mathbf{g} = [0 \quad 0 \quad \ldots \quad g_\beta \quad \ldots \quad 0 \quad 0 \quad]^T$. This implies that the manipulated input affects the dynamics of only one state variable (namely, $x_\beta(z,t)$), and the controlled output is a function of only one state variable ($x_\alpha(z,t)$). It is subsequently demonstrated that the results are true for arbitrary **h** and **g** too.

If for the input-output pair $\{u(z,t), y(z,t)\}$, the SIP $\sigma = p+1$, for an arbitrary p, then from the definition of SIP, and equations (19) and (20), the following equations are satisfied:

$$\mathbf{h}^T(\mathbf{J})^{p}\mathbf{g} \neq 0 \tag{21}$$

and

$$\mathbf{h}^T(\mathbf{J})^{p-k}\mathbf{g} = 0 \qquad \forall \qquad k \in \{1, p\} \tag{22}$$

Lemma: *The p^{th} power of an $n_x \times n_x$ matrix of the form of **J** is such that its $(i,j)^{th}$ term is*

given by:

$$\sum_{k_p=1}^{n_x}\sum_{k_{p-1}=1}^{n_x}\sum_{k_{p-2}=1}^{n_x}\cdots\sum_{k_3=1}^{n_x}\sum_{k_2=1}^{n_x}\frac{\partial f_i}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_{k_3}}\cdots\frac{\partial f_{k_{p-1}}}{\partial x_{k_p}}\frac{\partial f_{k_p}}{\partial x_j} \tag{23}$$

The lemma is proved using mathematical induction. For p=2,

$$J^2=\begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n}\end{bmatrix}\begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n}\end{bmatrix}$$

$$=\begin{bmatrix}\sum_{k_2=1}^{n}\frac{\partial f_1}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_1} & \sum_{k_2=1}^{n}\frac{\partial f_1}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_2} & \cdots & \sum_{k_2=1}^{n}\frac{\partial f_1}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_n}\\ \sum_{k_2=1}^{n}\frac{\partial f_2}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_1} & \sum_{k_2=1}^{n}\frac{\partial f_2}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_2} & \cdots & \sum_{k_2=1}^{n}\frac{\partial f_2}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \sum_{k_2=1}^{n}\frac{\partial f_n}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_1} & \sum_{k_2=1}^{n}\frac{\partial f_n}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_2} & \cdots & \sum_{k_2=1}^{n}\frac{\partial f_n}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_n}\end{bmatrix} \tag{24}$$

Hence, the $(i,j)^{th}$ term of $\boldsymbol{J}^2$ is

$$\sum_{k_2=1}^{n}\frac{\partial f_i}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_j}$$

So the lemma is, indeed, true for p=2.

Assuming that the lemma is true for an arbitrary p, implies that the $(i,j)^{th}$ term of $\boldsymbol{J}^p$ is

$$\sum_{k_p=1}^{n}\sum_{k_{p-1}=1}^{n}\sum_{k_{p-2}=1}^{n}\cdots\sum_{k_3=1}^{n}\sum_{k_2=1}^{n}\frac{\partial f_i}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_{k_3}}\cdots\frac{\partial f_{k_{p-1}}}{\partial x_{k_p}}\frac{\partial f_{k_p}}{\partial x_j} \tag{25}$$

Then, $\boldsymbol{J}^{p+1}$ =

$$\mathbf{J}^p\times\mathbf{J}=\mathbf{J}^p\times\begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n}\end{bmatrix}$$

From the rules of matrix multiplication, it can be concluded that the $(i,j)^{th}$ term of $\boldsymbol{J}^{p+1}$ is

$$\sum_{k_{p+1}=1}^{n}\sum_{k_p=1}^{n}\sum_{k_{p-1}=1}^{n}\cdots\sum_{k_3=1}^{n}\sum_{k_2=1}^{n}\frac{\partial f_i}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_{k_3}}\cdots\frac{\partial f_{k_p}}{\partial x_{k_{p+1}}}\frac{\partial f_{k_{p+1}}}{\partial x_j} \tag{26}$$

So, if the lemma is true for an arbitrary p, it is true for $p+1$. Hence, it must be true for all $p > 2$. Therefore, the $(i,j)^{th}$ term of $\boldsymbol{J}^p$ is given by:

$$\sum_{k_p=1}^{n}\sum_{k_{p-1}=1}^{n}\sum_{k_{p-2}=1}^{n}\cdots\sum_{k_3=1}^{n}\sum_{k_2=1}^{n}\frac{\partial f_i}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_{k_3}}\cdots\frac{\partial f_{k_{p-1}}}{\partial x_{k_p}}\frac{\partial f_{k_p}}{\partial x_j} \tag{27}$$

For $\sigma = p+1$, equations (21)-(23) imply the following:

$$h_\alpha\sum_{k_p=1}^{n_x}\sum_{k_{p-1}=1}^{n_x}\cdots\sum_{k_3=1}^{n_x}\sum_{k_2=1}^{n_x}\frac{\partial f_\alpha}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_{k_3}}\cdots\frac{\partial f_{k_{p-1}}}{\partial x_{k_p}}\frac{\partial f_{k_p}}{\partial x_\beta}g_\beta \neq 0 \tag{28}$$

and

$$h_\alpha\sum_{k_{p-l}=1}^{n_x}\sum_{k_{p-l-1}=1}^{n_x}\cdots\sum_{k_3=1}^{n_x}\sum_{k_2=1}^{n_x}\frac{\partial f_\alpha}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_{k_3}}\cdots\frac{\partial f_{k_{p-l-1}}}{\partial x_{k_{p-l}}}\frac{\partial f_{k_{p-l}}}{\partial x_\beta}g_\beta = 0 \qquad \forall \qquad l \in \{1,p\} \tag{29}$$

Equation (29) implies that, in equation (28) the indices $(k_2,k_3,\ldots,k_p,\alpha,\beta)$ are all distinct.

Since the left-hand-side of equation (28) is a summation of several terms, and is not equal to zero, hence, there is at least one term on the left-hand-side which is non-zero. This implies that there exists at least one set of distinct indices $(k_2,k_3,\ldots,k_p)$ such that:

$$h_\alpha\frac{\partial f_\alpha}{\partial x_{k_2}}\frac{\partial f_{k_2}}{\partial x_{k_3}}\cdots\frac{\partial f_{k_{p-1}}}{\partial x_{k_p}}\frac{\partial f_{k_p}}{\partial x_\beta}g_\beta \neq 0 \tag{30}$$

It is observed that equation (30) is identical to equation (15). But equation (15) implies that the shortest path from the input to the output variable has length l, and there are $l-1$ nodes corresponding to state variables in the path from the input node to the output node. Since the left hand side of equation (30) involves nodes corresponding to $p+1$ state variables, namely, $x_\alpha, x_\beta, x_{k_2}, x_{k_3}, \ldots, x_{k_p}$, it is concluded that the shortest path from $\hat{\mathbf{u}}$ to

$\hat{\mathbf{y}}$ is of length $p+2$. Hence, $\sigma = l-1$, for $\sigma = p+1$. But since p is arbitrary, it can be concluded that, for any value of σ,

$$\sigma = l-1 \tag{31}$$

Note that, it is possible that for some systems there are several sets of indices $(k_2, k_3, \ldots, k_p)$ which satisfy equation (30), but do not satisfy equation (28), since the sum of all those terms is zero. This corresponds to a singularity which does not invalidate the generic validity of the result.

Remark: In the case of general forms of **h** and **g**, the right hand side of equation (19) will be as below:

$$\mathbf{h}^T(\mathbf{J})^{\sigma-1}\mathbf{g} = \sum_{\alpha=1}^{n_x}\sum_{\beta=1}^{n_x} h_\alpha[(\alpha,\beta)^{th}\ \ term\ \ of\ \ \mathbf{J}^{\sigma-1}]g_\beta \tag{32}$$

In case of a MIMO system given by:

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{A}\frac{\partial \mathbf{x}}{\partial z} + \mathbf{B}\mathbf{x} + \mathbf{G}\mathbf{u}(z,t) \tag{33}$$

the right hand side of equation (19) will be as below:

$$\mathbf{h}_i(\mathbf{J})^{\sigma_{ij}-1}\mathbf{g}_j = \sum_{\alpha=1}^{n_x}\sum_{\beta=1}^{n_x} h_{i\alpha}[(\alpha,\beta)^{th}\ \ term\ \ of\ \ \mathbf{J}^{\sigma-1}]g_{\beta j} \tag{34}$$

where h_i is the i^{th} row of the matrix **H** and g_j is the j^{th} column of matrix **G**. The rest of the proof remains identical.

Appendix C: Theorem 3 Proof

Theorem 3: In the equation graph of a PPDE system, if l is the length of the shortest path from an input node to an output node and s is the SIP between the two variables, then,

$$\mathrm{s} = l - 1 \tag{35}$$

Proof 1: The time-derivative of the output, $\hat{y}_j$, is given by:

$$\frac{d\hat{y}_j}{dt} = \frac{dh_j(\hat{\mathbf{x}})}{dt} \tag{36}$$

h_j is a linear function, and thus,

$$\frac{d\hat{y}_j}{dt} = h_j\left(\frac{d(\hat{\mathbf{x}})}{dt}\right) \tag{37}$$

We assume, for the sake of illustration, without loss of generality, that

$$\hat{y}_j = \hat{x}_{m_1} = \int_{\alpha}^{\beta} \gamma_{m_1}(z) x_{m_1} dz \tag{38}$$

for an index $m_1 \in \{1, n_s\}$. Hence,

$$\frac{d\hat{y}_j}{dt} = \int_{\alpha}^{\beta} \gamma_{m_1}(z) \frac{\partial x_{m_1}}{\partial t} dz \tag{39}$$

Using Galerkin's method, we can write the following approximation:

$$x_{m_1} \approx \sum_{i=1}^{N_s} a_i^{(m_1)}(t) \phi_i^{(m_1)}(z) \tag{40}$$

where, N_s is the number of dominant slow modes, $a_i^{(m_1)}$ are the time-dependent coefficients, and $\phi_i^{(m_1)}$ are the eigenfunctions of the spatial differential operator in the PPDE

corresponding to x_{m_1}. Hence, $\frac{d\hat{y}_j}{dt}$ can be written as

$$\frac{d\hat{y}_j}{dt} = \int_{\alpha}^{\beta} \gamma_{m_1}(z)\left(\sum_{i=1}^{N_s} \frac{da_i^{(m_1)}}{dt}\phi_i^{(m_1)}(z)\right)dz \tag{41}$$

$$= \int_{\alpha}^{\beta} \gamma_{m_1}(z)\left(\sum_{i=1}^{N_s} (\lambda_i^{(m_1)} a_i^{(m_1)} + \mathcal{B}_i^{(m_1)}\mathbf{u} + f_i^{(m_1)}(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(n)}))\phi_i^{(m_1)}(z)\right)dz \tag{42}$$

where N_s is the number of dominant modes, $\lambda_i^{(m_1)}$ is the i^{th} eigenvalue of the spatial differential operator corresponding to x_{m_1}, $\mathcal{B}_i^{(m_1)}$ is the i^{th} element of the operator $\mathcal{B}$, and $\{\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(n)}\}$ are the vectors of the time-dependent coefficients corresponding to the n_s distributed state variables. The second derivative of the output is then given by

$$\frac{d^2\hat{y}_j}{dt^2} = \int_{\alpha}^{\beta} \gamma_{m_1}(z)\left(\sum_{i=1}^{N_s}\left(\lambda_i^{(m_1)}\frac{da_i^{(m_1)}}{dt} + \mathcal{B}_i^{(m_1)}\frac{d\mathbf{u}}{dt} + \frac{df_i^{(m_1)}}{dt}(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(n)})\right)\phi_i^{(m_1)}(z)\right)dz \tag{43}$$

Now, $\frac{df_i^{(m_1)}}{dt}$ can be evaluated using the chain rule as,

$$\frac{df_i^{(m_1)}}{dt} = \sum_{j=1}^{n}\sum_{k=1}^{N_s} \frac{\partial f_i^{(m_1)}}{\partial a_k^{(j)}}\frac{da_k^{(j)}}{dt} \tag{44}$$

Using Faa di Bruno's formula [128], the highest order derivatives of the time-dependent coefficients in the p^{th} order derivatives of $f_i^{(m_1)}$ are

$$\sum_{j=1}^{n}\sum_{k=1}^{N_s} \frac{\partial f_i^{(m_1)}}{\partial a_k^{(j)}}\frac{d^p a_k^{(j)}}{dt^p} \tag{45}$$

If the SIP of the output with respect to an input variable is s, i.e., if s is the smallest order of the output derivative that is explicitly dependent on the input, then the input must appear in at least one of the s^{th} derivatives of the time-dependent coefficient of x_{m1} given

by

$$\frac{d^s a_i^{(m_1)}}{dt^s} = \left(\frac{d^{(s-1)} f_i^{(m_1)}}{dt^{(s-1)}} + \text{lower order terms} \right) \tag{46}$$

for some $i \in \{1, N_s\}$. Equations (44) and (45) imply

$$\frac{d^s a_i^{(m_1)}}{dt^s} = \left(\sum_{j=1}^{n} \sum_{k=1}^{N_s} \frac{\partial f_i^{(m_1)}}{\partial a_k^{(j)}} \frac{d^{(s-1)} a_k^{(j)}}{dt^{(s-1)}} + \text{lower order terms} \right) \tag{47}$$

which means that the input appears in the $(s-1)^{th}$ derivative of at least one of the time-dependent coefficients corresponding to another state variable $x_{m_2}(z,t)$, which in turn, appears in the PPDE corresponding to $x_{m_1}(z,t)$. In that case, in the equation graph, there is an edge from $x_{m_2}(z,t)$ (for some $m_2 \in \{1, n_s\}$) to $x_{m_1}(z,t)$. Since equation (47) is true for any non-negative integer s, we can conclude that there is a series of s distinct state variables, $\{x_{m_1}, \ldots, x_{m_s}\}$ whose $(s-p)^{th}$ derivatives are explicitly dependent on the input variable for all $p \in \{0, s-1\}$, resulting in a sequence of $s-1$ edges from the variable whose first derivative depends on the input to the variable whose s^{th} derivative depends on the input. There is an edge from the input to the state variable whose first derivative explicitly depends on the input (x_{m_s}) and from the state variable x_{m_1} to the output variable. Hence, from the input variable to the output variable, the least number of possible edges is $s+1$.

Conversely, if the shortest path length from an input variable to an output variable on the equation graph is l, it means that there is a series of $l-2$ distinct state variables $\{x_{m_{l-2}}, \ldots, x_{m_1}\}$, such that x_{m_p} appears in the PPDE corresponding to $x_{m_{p+1}}$ $\forall p \in \{1, l-2\}$. The input appears in the PPDE corresponding to x_{m_1} and the output is algebraically related to $x_{m_{l-2}}$. Then, equation (47) implies that the SIP of the output with respect to the input must be $l-1$.

Remark 1: Since the aforementioned equations use only the differential or continuous

or discrete summation operators, which are linear, equations (38)-(47) hold for any generic linear operator that relates a state variable to the output variable as in equation (4.22).