

## **SYSTEMATIC APPROACH FOR CONTROL STRUCTURE DESIGN**

# SYSTEMATIC APPROACH FOR CONTROL STRUCTURE DESIGN

By

Yongsong Cai, B.Eng, M.Eng.

A Thesis

Submitted to the School of Graduate Studies

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

McMaster University

© Copyright by Yongsong Cai, March 2009

DOCTOR OF PHILOSOPHY (2009)     McMaster University  
(Chemical Engineering)     Hamilton, Ontario, Canada

TITLE:   Systematic approach for control structure design

AUTHOR:   Yongsong Cai, B.Eng., M.Eng. (Zhejiang University, P. R. China)

SUPERVISOR: Professor T.E. Marlin

NUMBER OF PAGES: ix, 187

# Abstract

Control structure design is an essential step in control system synthesis and has big impact on achievable closed-loop performance. This thesis develops a systematic approach of selecting optimal control structures based on closed-loop dynamic performance and other criteria, such as integrity.

The main contribution of this thesis is a rigorous mathematical formulation for control structure design problem that includes full closed-loop transient analysis with additional integrity requirement. The multi-objective framework is extendable so that different control performance objectives can be easily added. Unique process requirements and engineer inputs can be taken into account as additional constraints. The proposed formulation is a Mixed Integer Nonlinear Programming (MINLP) with complementarity constraints. The research scope is limited to linear process models and linear controller algorithms.

The tailored solving strategy that makes this challenging problem computationally tractable is introduced in this thesis. The modified Branch and Bound algorithm takes advantage of the special problem structure by using control knowledge to generate valid lower bound efficiently. Prior knowledge can be cooperated as heuristic tuning parameters to guide the solving process so that a reasonably good solution can be found early in the solving process. The complexity study shows the solving strategy can attack



design problem size up to  $8 \times 8$ . Considering the percentage of good structures needing evaluation will decrease with problem size even larger problems will be tractable.

The common control structures in process industries, such as square and non-square Single-Input-Single-Output (SISO) loop pairing using PID controller and block-centralized structure using Model Predictive Controller (MPC), are addressed in this thesis. The usefulness of this research has been demonstrated by several case studies, include Tennessee Eastman problem. The proposed methodology finds a physically sound pairing with good performance for Tennessee Eastman problem in less than one hour, while several off- the-shelf NLP, MINLP and global solvers cannot find a solution in five days.

# Acknowledgements

I would like to express my gratitude to my supervisor, Dr. T. E. Marlin, for his supervision throughout this thesis. Without his support and guidance I would have never finished this long journey.

I would also like to thank my supervisory committee members, Dr. C. Swartz, Dr. T. Terlaky and Dr. A. Deza, for their valuable insight and suggestion on my topic.

Thanks go out to all my friends in the process control group for their support and friendship throughout this work. In particular, I would like to thank San, Honglu, Manish, Adam, Danielle, Rhoda, Kevin, and Mark-John.

Finally, I would like to thank my family for their unconditional support, understanding and encouragement.

# Table of Contents

|           |                                       |    |
|-----------|---------------------------------------|----|
| Chapter 1 | Introduction .....                    | 1  |
| 1.1       | Problem Definition.....               | 1  |
| 1.2       | Thesis Objective and Scope .....      | 4  |
| 1.3       | Thesis Outline .....                  | 5  |
| 1.4       | Terms and Definitions .....           | 6  |
| Chapter 2 | Technology Survey .....               | 10 |
| 2.1       | Control Structure Design.....         | 10 |
| 2.1.1     | Heuristic Methods .....               | 10 |
| 2.1.2     | Shortcut Metrics .....                | 13 |
| 2.1.3     | Optimization Approaches.....          | 15 |
| 2.2       | Optimal Tuning Problem.....           | 17 |
| 2.3       | Solving Technology of MINLP.....      | 19 |
| 2.4       | State of Art/Challenges .....         | 20 |
| Chapter 3 | Model and Problem Definition.....     | 22 |
| 3.1       | Defining the Inputs.....              | 23 |
| 3.2       | Multiple-objective Optimization ..... | 24 |

|           |  |    |
|-----------|--|----|
| 3.3       | Mathematical Formulation .....   | 26 |
| 3.3.1     | Linear Process Model.....  | 27 |
| 3.3.2     | PI Controller .....  | 27 |
| 3.3.3     | Saturation .....   | 28 |
| 3.3.4     | Integer Structural Constraints.....  | 31 |
| 3.3.5     | Model Mismatch .....   | 33 |
| 3.3.6     | Formulation with Additional Criteria.....                                  | 36 |
| 3.4       | Case Study: Fluidized Catalytic Cracker (2×2).....                         | 36 |
| 3.5       | Summary .....  | 46 |
| Chapter 4 | Systematic Selection of Loop Pairings.....                                 | 47 |
| 4.1       | Combine Transient with Shortcut Metrics .....                              | 49 |
| 4.2       | MINLP Solvers and MILP Reformulation.....                                  | 50 |
| 4.2.1     | MINLP Solvers .....  | 50 |
| 4.2.2     | MILP Reformulation.....  | 51 |
| 4.3       | Tailored Solving Strategy for Transient Evaluation .....                   | 56 |
| 4.3.1     | Controllability Check .....  | 57 |
| 4.3.2     | Branching Strategy on Pairings.....  | 59 |
| 4.3.3     | Estimating Lower Bounds at Intermediate Nodes Using Control Knowledge..... | 65 |
| 4.3.4     | Grid Search on Loop Tuning.....  | 68 |
| 4.4       | Additional Criteria as Constraints.....                                    | 70 |
| 4.5       | Solving the Full-integer Candidates.....                                   | 72 |
| 4.6       | Non-square Systems.....  | 73 |
| 4.7       | Software Implementation .....  | 75 |
| 4.8       | Case Studies .....   | 76 |
| 4.8.1     | Fluidized Catalytic Cracker (FCC) .....                                    | 77 |
| 4.8.2     | Fired Heater.....  | 77 |
| 4.8.3     | Non-square System .....  | 82 |
| 4.9       | Summary .....  | 86 |
| Chapter 5 | Tennessee Eastman Problem.....   | 88 |

|           |  |     |
|-----------|--|-----|
| 5.1       | Process Description .....                          | 88  |
| 5.2       | Published Designs .....                            | 93  |
| 5.3       | Define Control Objectives.....                     | 95  |
| 5.4       | Selection of Controlled Variables .....            | 96  |
| 5.5       | Selection of Manipulated Variables .....           | 97  |
| 5.6       | Selection of Safety Loops .....                    | 99  |
| 5.7       | Selection of Remaining Loop Pairings.....          | 101 |
| 5.7.1     | Controllability Check .....                        | 102 |
| 5.7.2     | Heuristic Search Parameters.....                   | 103 |
| 5.7.3     | Integrate with Integrity Requirement .....         | 105 |
| 5.7.4     | Paring for Best Dynamic Performance.....           | 108 |
| 5.8       | Problem Complexity Analysis .....                  | 112 |
| 5.9       | Summary .....                                      | 117 |
| Chapter 6 | Selection of Block Centralized Structures .....    | 119 |
| 6.1       | Block Centralized Design Problem Definition.....   | 120 |
| 6.1.1     | Unconstrained MPC Controller.....                  | 120 |
| 6.1.2     | Block Centralized MPC Controller.....              | 123 |
| 6.1.3     | Block Centralized Structural Constraints.....      | 125 |
| 6.1.4     | Block Centralized Design Problem Formulation ..... | 127 |
| 6.2       | Tailored Solving Strategy.....                     | 129 |
| 6.2.1     | Branch Strategy on Blocks.....                     | 130 |
| 6.2.2     | Generate Lower Bounds Using Control Knowledge..... | 132 |
| 6.2.3     | Integrate Transients with Shortcut Metrics .....   | 134 |
| 6.3       | Case Study.....                                    | 134 |
| 6.3.1     | Solving Process .....                              | 136 |
| 6.3.2     | Results Analysis .....                             | 138 |
| 6.3.3     | Modified Process.....                              | 143 |
| 6.4       | Summary .....                                      | 147 |
| Chapter 7 | Conclusion.....                                    | 149 |

|            |  |     |
|------------|--|-----|
| 7.1        | Summary .....  | 149 |
| 7.2        | Contributions.....   | 151 |
| 7.3        | Future Works.....  | 152 |
| 7.3.1      | Global Optimal Tuning for Multiple-Loop and Multiple-MPC Controller  | 152 |
| 7.3.2      | Additional Decentralized Controller Algorithms.....                  | 153 |
| 7.3.3      | Control Structure with Hierarchy .....                               | 154 |
| 7.3.4      | Software Structure.....  | 154 |
|            | Nomenclature .....   | 156 |
|            | Reference.....   | 160 |
| Appendix A | Integrity .....  | 171 |
| Appendix B | Relative Gain Array (RGA) .....                                      | 174 |
| Appendix C | Block Relative Gain (BRG) .....                                      | 177 |
| Appendix D | Interior Point Method with Complementarity Constraints.....          | 180 |
| D.1        | Actuator Saturation as Complementarity Constraints.....              | 180 |
| D.2        | Interior Point Method for General NLP .....                          | 181 |
| D.3        | Interior Point Method for NLP with Complementarity Constraints ..... | 182 |
| Appendix E | Capital Cost Estimation for FCCU Air Blower .....                    | 184 |
| Appendix F | Simulation Configurations .....                                      | 186 |

# Chapter 1

## Introduction

### 1.1 Problem Definition

A modern chemical plant may have thousands of measurements and hundreds of actuators. However, the control system does not include all available measurements and actuators as controlled variables and manipulated variables. The decision making process for selecting the set of controlled variables, manipulated variables and the interconnection between these two is called *Control Structure Design*.

Control structure design is an important step before choosing control algorithms and tunings. Inappropriate control structures may impose limitations on achievable closed-loop performance that cannot be corrected by advanced control algorithms or tunings (Lee and Morari, 1990).

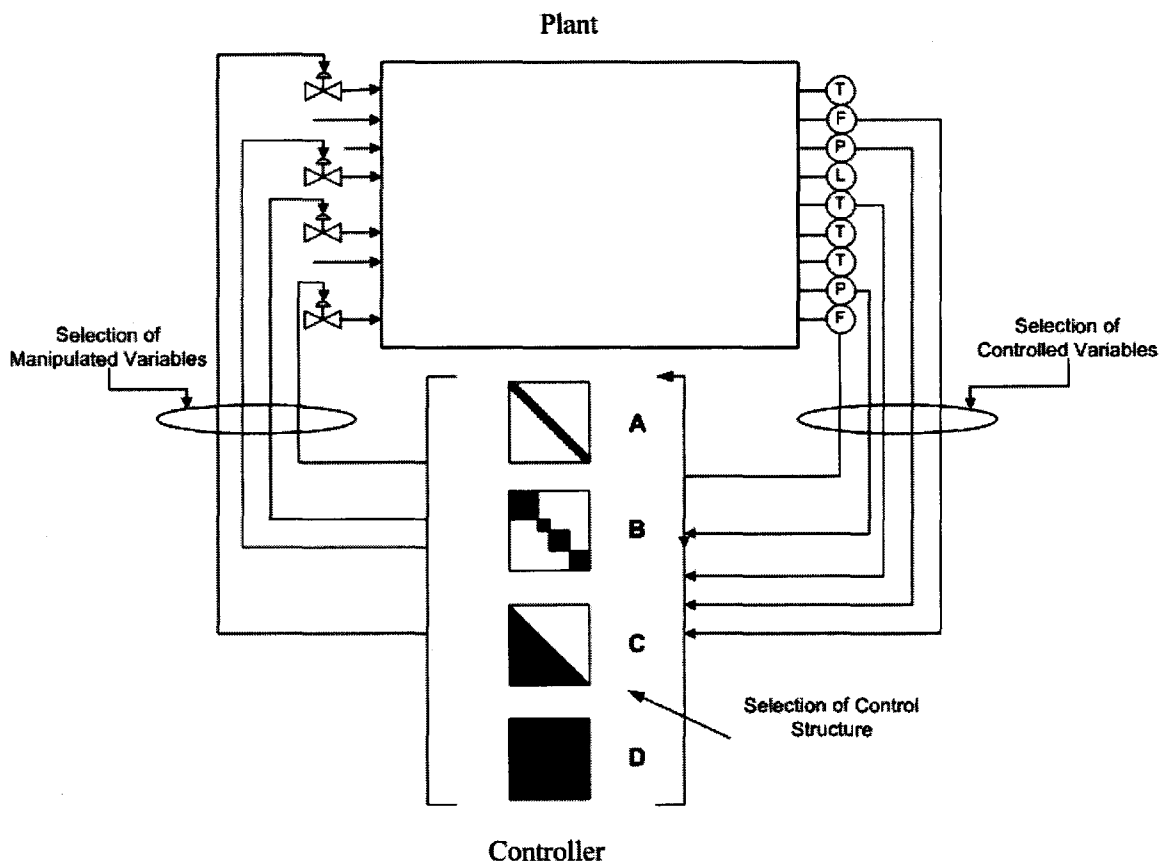


Figure 1.1 Control structure design

The control structure design includes the selection of measurements as controlled variables, actuators as manipulated variables and the interconnection between controlled variables and manipulated variables (Figure 1.1). In this research we focus on the control structure, that is, the interconnection between controlled variables and manipulated variables.

The control structure can be represented as an interconnection matrix, in which each row is a manipulated variable, each column is a controller variable, and matrix element, containing a control law, defines whether the controller uses a CV when determining the MV. Figure 1.1 shows several possible structures represented by the shapes of the non-zero elements (marked in dark color) in the interconnection matrices, such as (A) diagonal, (B) block centralized, (C) lower triangular and (D) fully centralized. The diagonal structure is also referred to as multiloop structure that consists



of Single Input Single Output (SISO) controllers. The centralized structure would be implemented by one multivariable controller that connects all controlled variables and manipulated variables. A block centralized structure is a combination of several multivariable controllers and/or SISO controllers. The block centralized structure is a very flexible structure, which can vary from a diagonal structure (with all controllers are SISO) to a fully centralized structure (with only one multivariable controller).

Because of computing limits SISO was the only practical technology from the beginning of process control to 1960's (Bennett, 1993). Today many chemical plants are still controlled by SISO controllers safely and profitably. Advances in computing and control technology enable us to build large centralized control structures. The question we should ask is: what is the extent of centralization justified by improved control performance? Nett (1989) noted that we should "*minimized control system complexity subject to the achievement of accuracy specification in the face of uncertainty*" despite increasing plant complexity. In practice a simple control structure can bring us:

- Easy maintenance. Everyone from the operator to the manager can understand how it works.
- High failure tolerance. One unrecognized failure would not be propagated through the controller calculations to manipulated variables throughout the plant.
- Robustness. The simple control system can be well-conditioned even when the process itself is ill-conditioned.

Luyben (1998) stated, "*Our governing philosophy is it is always best to utilize the simplest control system that will achieve the desired objectives*".

Foss (1973) identified that the control structure selection was the most important issue in process control theory. He suggested that optimization could be a general method for solving control structure design problem: "*The task is to make decisions about control structure, and it is here where optimization can be most intelligently used. Such optimization problems are, however, a new breed. The standard formulation of optimal control will seldom suffice.*" He also foresaw that the selection of the objective function and multi-objective formulation are the main challenges to the use of optimization

methods: *“The objective function by which control system structure is judged and selected must be carefully formulated. The choice is crucial to the applicability of the theory and the tractability of the analysis. The scalar indices, used almost exclusively in the past, will no longer suffice for most multivariable processes; vector valued indices will be needed.”*

## 1.2 Thesis Objective and Scope

The research goal is to find a *systematic* method to select a control structure based on *dynamic closed-loop performance*. This research aims to solve real industrial design problems, which can be challenging in many aspects, such as multiple design criteria definition, complex dynamic interaction and moderate to large problem size.

A systematic methodology is needed for general processes. This research focuses on highly integrated and/or new processes with several units, which may not be covered by existing heuristic design rules for typical unit operations. An optimization problem will be proposed as a general framework to integrate different design aspects into one rigorous mathematical formulation.

The design objective by which the control structure is selected has to be carefully formulated. The final design is affected by many very different criteria, such as dynamic performance, integrity, robustness and complexity. Some of these criteria may conflict with each other. Therefore, a formulation that represents the complexity of the design problem is needed.

A complete dynamic transit, which is overlooked by most of heuristic rules and shortcut metrics, is essential for evaluating complex highly-integrated systems. In addition, many realistic factors, such as plant model mismatch, major disturbances, actuator capacity, measurement noise and so on, should be considered to get accurate results.

Due to combinatorial nature of the control structure design, the computing time for solving this problem suffers from the *curse of dimensionality*. In order to solve relatively large problems, such as systems with eight controlled variables and eight

manipulated variables, we have to develop specialized solving technology that take advantage of the problem structure and utilize the control theory to guide the control structure search.

The processes we studied in the research are square systems (the number of controlled variables is equal to the number of manipulated variables), and then the methodology is extended to handle non-square systems in Chapter 4.

We first attack the multiloop (diagonal) structure design problem, which is also referred as the loop pairing problem. Then the methodology is extended to handle more general block centralized structure, which includes one or several multivariable controllers and closely resembles the control system schemes used in many chemical plants. The designs resulting from this research can be applied to wide variety of control systems from single-loop instrumentation to large scale Distributed Control Systems (DCS) on which multivariable controllers are easily implemented.

### **1.3 Thesis Outline**

Chapter 2 reviews the state of the art of the control structure design technology. Three categories of the design methods are reviewed, which are heuristic methods, shortcut metrics, and optimization methods. The challenges of solving optimal tuning problem and mixed-integer, non-linear programming (MINLP) are also discussed.

Chapter 3 presents the new mathematical formulation that addresses the multi-objective and realistic scenario for dynamic transient evaluation. The shortcut metrics appear as additional constraints in the formulation. A fluidized catalytic cracker example shows each part of the formulation is essential to evaluate the dynamic transient accurately.

Chapter 4 provides the systematic solution procedure for selection of the loop pairing problem formulated in Chapter 3. We tailor the branch and bound algorithm specifically for the control structure design problem, which takes advantage of the problem structure to guide the branching procedure and use control theory to generate valid lower bound for bounding procedure.

Chapter 5 examines the proposed control structure design methodology with the medium size industrial test problem, Tennessee Eastman problem. Combining the process knowledge and the proposed methodology a loop-pairing structure with good performance was found in less than one hour. A scaling study based on the Tennessee Eastman pairing problem shows our methodology can handle 8x8 system with proper heuristic search parameters. We expect that even larger problems will be tractable because the percentage of good pairings needing evaluation will decrease with problem size.

Chapter 6 extends the loop pairing methodology to handle the more general block centralized structure. The unconstrained block centralized MPC is derived as the controller equation in the optimization formulation. Then, a set of integer constraints is developed to enforce the block centralized structure on the controller model. A special solving method is also developed for this complex MINLP.

Chapter 7 summarizes the results obtained in the thesis, draws some conclusions and highlights some areas for future work.

## 1.4 Terms and Definitions

We assume the readers have basic chemical process control and mathematical programming background. The followings are some comments on the terms introduced above or will be used in the following chapters.

*Plant* and *process* are essentially synonymous terms. They all refer to any system to be controlled, which can be the physical process itself or include part of the low-level control systems. However, the term *plant* sometime means the whole factory, which consists of many *process* units.

*Control* is to maintain desired conditions in a process by adjusting selected variables in the process (Marlin, 2000). This thesis is limited to *feedback control*, which makes use of an output of a process to influence an input to the process. The process output used by feedback control is called the *controlled variable*, and the process input influenced by feedback control is called the *manipulated variable*.

Controlled variables can be directly measured, such as pressure and temperature, or can be estimated as a function of some other easy-to-measure variables. The selection of controlled variables is based on the control objectives. The most common manipulated variables in chemical plant are valve openings and motor speeds. They have physical limitations on the ranges and speeds that they can be adjusted, which are translated to hard bounds in optimization. The dynamic transient of manipulated variables is also an important control objective because it may directly relate to economy performance and equipment capacity, performance and life.

The simplest controller has only one controlled variable and one manipulated variable, called Single-Input-Single-Output (SISO) controller. *PID* is the most common SISO feedback control algorithm. *PID* means Proportional, Integral and Derivative. The output of *PID* controller is the summation of the proportion, integration and derivative of the error signal or the controlled variables.

Multiple-Input-Multiple-Output (MIMO) controller has multiple controlled variables and manipulated variables. Model Predictive Control (*MPC*) is a common MIMO control algorithm. It uses a process dynamic model explicitly and solves the optimal open-loop control using mathematical programming at each controller execution. It is particularly useful for multivariable system with inequality constraints on controlled and manipulated variables.

Chemical plants are controlled by many controllers, some are SISO controllers and some are MIMO controllers. There are there possible control structures:

- *Decentralized structure*, also called *multi-loop structure*, consists of many SISO controllers.
- *Block-centralized structure* is also called *block diagonal structure*. The whole control system consists of MIMO controllers of various sizes and SISO controllers.
- *Centralized control* refers to one MIMO controller controls the whole plant.

There are many different control system design objectives. We list some considered in this research.

- *Controllability* determines whether certain dynamic behavior can be achieved with specific selection of controlled and manipulated variables independent of controller design (Rosenbrock, 1974).
- A system has *integrity* when the (reduced) system is stable after turning off some of the control loops or some actuators become saturated. There are many different definitions of *integrity* (Campo and Morari, 1994). The one most relevant to this research is Integral Controllability with Integrity (ICI) for Decentralized control, which says that the feedback system can be stabilized with a feedback controller with integral mode and any combination of controllers can be reduced to zero. There is no easy necessary and sufficient conditions to ensure ICI. The commonly used criterion, Relative Gain Array (RGA), is a necessary condition for ICI.
- *Robustness* describes how sensitive (or insensitive) a control system is with respect to the design uncertainties, such as model mismatch, actuator inaccuracy and measurement drift. A control system should perform well over possible operating conditions defined at the design stage.
- *Performance* describes how well a process behaviors respect to certain criterion. There are many different performance criteria, such as controlled variable close to set point for regulatory control, controlled variable tracking set point fast and smooth for servo control, minimal controlled variable deviation with respect to disturbance, minimal energy consumption for manipulated variable and so on. The performance criteria are usually summarized as scalar numbers for easy comparison. For instance, the dynamic performance criterion we use in this research is Integral of Squared Error (ISE) of controlled variables, which is a scalar number that measures how far the controller variables are from their set points over the whole dynamic transient with respect to disturbance.
- *Complexity* of control systems is synonymous the complexity of the controller algorithms in this research. The complexity of the controller algorithms is the

amount of resource, basically time, required for the execution of algorithms. The common metric is the computational time that the algorithm takes to find a solution. It can also be measured by the number of the control loops. MIMO controller is more complex than SISO controller, and bigger MIMO controller is more complex than smaller MIMO controller.

## **Chapter 2**

# **Technology Survey**

The control structure design is an essential step in control system synthesis. After Buckley (1964) first emphasized the control structure design in his textbook more than 40 years ago, various approaches have been proposed to attack different aspects of this problem. However, we still lack systematic methods for the synthesis of control structures for general processes.

In this chapter a review of the approaches and related technologies for control structure design are presented. Although there are a large number of publications on this area only the most important and relevant are presented here.

## **2.1 Control Structure Design**

First, the following three categories of the state of arts of control structure design methods, heuristic methods, shortcut metrics, and optimization approaches, are discussed.

### **2.1.1 Heuristic Methods**

The control structure design is part of the chemical process design. People begin with trial-and-error methods and eventually summarize their experience in some rules of



thumb, which are heuristics. These heuristics are very helpful at design stage because they might guide you to a reasonably good solution in very short time if the assumptions on which they are based are met. We will also use some of these guidelines in our method to improve the efficiency of the systematic methods.

But heuristics are not rigorous. You cannot prove the results are in any sense optimal. Most of these rules of thumb are developed for certain type of processes or unit operations. You cannot guarantee it will work if applied outside of the experience.

### **2.1.1.1 Process specific rules**

Certain process units are employed frequently in process plants and heuristics have been developed for the units.

Shinskey (1984) used process insight like material balances and thermodynamic concepts to develop a shortcut design metric, Relative Gain Array (RGA), to select optimal distillation control structure for productivity and energy conservation purposes. Shinskey (1988) listed several commonly used multiple loop control structures, such as ratio control, cascade control, override control, and valve position control, with their typical applications, such as distillation tower and evaporator.

Dukelow (1986) published a book of boiler control design based on his experience. He gave detailed instructions for basic boiler on-line controls, such as feed water control, combustion control and so on. These specific design rules are very useful for boiler control design, but difficult to apply to other processes.

Liptak (1999) published the basic control systems for a number of unit operations, such as boiler, compressor, cooling tower, dryer and so on. He also showed the control strategies that achieve near-optimal operations using basic instrumentation and control equipment, such as single-loop controllers, high/low selector, multiplier/divider and so on.

These rules cannot address changes in unit designs or complex process structures with multiple units.

### 2.1.1.2 General Design Rules

Buckley (1964) suggested the idea of separating the control system design into two parts: low-frequency material balance control and high-frequency product quality control. The time constant of inventory control loops might be much larger than the time constant of product quality control loops. The frequency difference permits independent tuning of two parts. This design rule is very useful for defining the control hierarchy.

Douglas (1988) pointed out that the cost of raw materials and the products are usually much greater than the cost of capital and energy. Therefore, we should minimize the loss of raw materials and products by tightly controlling the stream compositions within the process. He claims that we should not regulate the gas recycle flow; instead, maximize it to promote product conversion.

Downs (1992) insightfully pointed out the importance of the overall component balances for the whole process. Since chemical process often acts as pure integrators any imbalance in components results the process gradually filling up or drawing dry.

Luyben (1998) summarized the heuristic design rules and established the following nine steps of plant-wide control structure design procedure, which are

- Establish control objectives
- Determine control degrees of freedom
- Establish energy management system
- Set Production rate
- Control Product quality and handle safety, environment, and operational constraints
- Fix a flow in every recycle loop and control inventories
- Check component balances
- Control individual unit operations
- Optimize economics and improve dynamic controllability

These steps provide a good high-level guideline for large-scale control structure design problem without needs of detailed models. However, these design rules are not specific enough to select individual control loops.

### 2.1.2 Shortcut Metrics

Shortcut metrics use limited process model information, mainly steady state gain, and only require simple calculations. They can provide information, such as loop interaction, system integrity and stability, which are very important for control structure decisions. However, these easily calculated performance measures do not provide detailed transient information, and they can be misleading in some instances.

RGA (Relative Gain Array) is among the most recognized shortcut metrics. It is defined as the ratio between the open-loop gain and the gain of the same loop when all other feedback control loops are perfectly controlled (Bristol, 1966). A brief introduction of RGA can be found in Appendix B. Many results have been published on RGA, such as McAvoy (1983) and Shinskey (1988). The common RGA design rules are

- Select loop pairings with positive RGA values.
- Select loop pairings with RGA values “close to one”.

These guidelines have a rigorous theoretical basis as explained in the following.

- Pairing only positive RGA is necessary condition for Integral Controllability with Integrity (Grosdidier and Morari, 1985). Pairing a loop with negative RGA value will cause at least one of three undesired situations: the loop is unstable, the whole closed-loop system is unstable or the closed-loop system without the loop is unstable.
- The norm of RGA bounds the minimal condition number of process gain matrix from the bottom, which establishes solid link with robustness (Skogestad and Morari, 1987). Pairings with large magnitude RGA value make the control system very sensitive to model mismatch.

However, there is no direct relation between RGA value and closed-loop dynamic performance. The rule requiring pairing loops with RGA values close to one may result poor closed-loop dynamic performance for a triangular system, because RGA values are always equal to one for triangular system (Grosdidier and Morari, 1985). The possible undetected “one-way” interaction may seriously degrade the dynamic performance. Also,

Skogestad *et al.* (1990) pointed out that steady state RGA could be inadequate for selecting the distillation controller structure due to lack of high-frequency information. They suggested frequency-dependent RGA as a remedy, however, there is no systematic interpretation with regard to the relation to the dynamic performance.

Manousiouthakis, *et al.* (1986) extended RGA to Block Relative Gain (BRG), which can handle multivariable block structures. A brief introduction of BRGA can be found in Appendix C. BRG has similar properties as RGA regarding integrity and robustness. Similarly, BRG does not predict the closed-loop dynamic performance directly; and it does not consider disturbance rejection.

Stanley *et al.* (1985) proposed Relative Disturbance Gain (RDG) that directly addresses disturbance rejection issues, and it can be used to analyze how the loop interactions help to reject the disturbance. RDG was proved to be the ratio of the integral error of controlled variable under perfect control to the value under single-loop control. Integral error gives RDG the limited ability to predict the dynamic performance. Smaller RDG is, better dynamic performance is expected. However, integral error has its disadvantage that positive and negative error cancel out in the integration, which means an oscillatory transient may have a very small integral error. Obviously, a very oscillatory transient is not the dynamic performance we want. Therefore, RDG cannot distinguish between a well-controlled system and a very oscillatory system. Chang and Yu (1992) extend RDG to Generalized Relative Disturbance Gain (GRDG) to handle any control structure including block-centralized structure. GRDG has the same properties and disadvantage as RDG.

Cao and Rossiter (1997) proposed Single-Input Effectiveness (SIE), which they claim is useful to select manipulated variables for non-square systems. It measures how effective the selected manipulated variable affects the control variables and has close relation with non-square RGA. SIE can be efficient screening tool, but it does not provide information about dynamic transients and stability.

Skogestad and Morari (1987) introduced Disturbance Condition Number that measures the alignment of the directionality of the plant and the disturbance. If the

disturbance aligns to the plant directionality associated with the maximum singular value, which gives the direction in which the plant has the maximum power to reject it. Disturbance Condition Number gives us a tool to select a subsystem that has the right directionality to reject certain disturbances. But it does not provide information about the controller configuration (such as loop pairing).

Trierweiler (1997) proposed the Robust Performance Number (RPN) that is a measure of “*how potentially difficult it is for a given system to achieve the desired performance robustly*”. RPN is derived from the calculation of structured singular value with inverse-based controller. The smaller RPN is, the more likely the inverse-based controller will achieve the desired performance that is specified in the frequency domain. RPN provides information about how to select the subsystem variables (i.e., select controlled and manipulated variables), but does not help the selection of the controller structure (such as loop pairing).

### 2.1.3 Optimization Approaches

Many approaches for designing “optimal” controllers have been developed. However, many of these approaches assume the fully centralized controller structure, and use the mathematical programming to find the best parameters of the controller algorithm. For very simple system, the optimal controller may have a clear structural pattern (Harris and MacGregor, 1987). But generally the optimal controllers are fully centralized.

Robinson *et al.* (2001) proposed a method to extract information from centralized optimal controller, which decomposes the controller gain matrix into diagonal part and off-diagonal part; the small off-diagonal elements are disregarded by comparing with an arbitrary threshold. McAvoy *et al.* (2001) proposed to calculate the RGA of the optimal controller gain matrix, then use it to guide the structure selection. However, the relation between RGA of the optimal controller and the closed-loop dynamic performance is not clear.

McAvoy (1999) propose a stepwise approach that uses heuristics, shortcut metrics and optimization in different steps. First, he uses heuristic rules to separate the controlled variables into smaller prioritized groups in order to reduce the problem size. The second step is to select the manipulated variables for each controlled variable group by MILP, which minimizes the steady-state manipulated variable movements. The loop pairings of each controlled variable group are determined using the RGA in step three. This method can handle large process easily by using different design tools together. But different design tools are used in different design steps, which make it difficult to trade off among different criteria. And the optimization step does not consider dynamic transients.

Wang and McAvoy (2001) selected manipulated variables based on dynamic transients with an open-loop formulation (without controller equations). The formulation is an MILP, but the transient is very different from closed-loop system behaviors and cannot be used to select control configuration.

Including the dynamic transients explicitly complicates the optimization problem significantly. Bansal *et al.* (2003) published the progress in solving Mixed Integer Dynamic Optimization (MIDO), which use detailed nonlinear dynamic models to design the process as well as the control system including control structure. It is non-convex MINLP and very difficult to solve. The control system in case study used in that paper involves only 2 PI controllers with fixed pairings.

Kookos (2002) proposed an approach to simplify the optimal control structure design problem with dynamic transients. He used linear dynamic model and replace the continuous tuning parameters by predetermined discrete values. The idea is to keep everything linear; then the resulting MILP can be solved reliably. The biggest case study he showed is 2x3. The simplification uses many extra integer variables, which make the formulation intractable for large systems. We will show later in Chapter 4 that this formulation cannot solve the reduced Tennessee Eastman problem (5x5 system).

In order to handle large systems, Kookos (2001) proposed another approach based solely on shortcut metrics. The approach trades off several shortcut metrics in a MILP. Since the shortcut metrics only use steady state gains the problem is relatively easy to

solve and capable to handle large systems. But no dynamic transient information is taken into account.

The state of the art of optimization approaches is either solving large system with only shortcut metrics objectives or solving very small systems, such as  $2 \times 2$  system, with dynamic transient objectives. We are seeking a formulation that can be solved for larger systems combining shortcut metrics and the complete closed-loop dynamic transient.

## **2.2 Optimal Tuning Problem**

The closed-loop dynamic transients depend on both control structure and parameters of the controller algorithm. Therefore, determining the proper parameters for controller algorithm (controller tuning problem) is a required step in the selection of control structure if the dynamic transient is evaluated explicitly. And the controller must be tuned to satisfy the needs of the specific process.

Surprisingly, the tuning problem is still an active research area. If you search EI database using key words “PID controller tuning”, you can find about 1500 papers from year 2000 to 2008. Most of research addresses single loop PID controller tuning.

Single loop PID controller tuning research started as heuristic correlations, such as Ziegler-Nichols method (1942) and Cohen-Coon method (1953). Then more rigorous stability or robustness based method, such as Astrom-Hagglund method (1984) and Internal Model Control (IMC) method (Morari, 1984), emerged. Most of these methods assume the process models are first-order-plus-time-delay and do not consider the dynamic transient explicitly.

The multi-loop PID controller tuning is a real challenge problem because of the complex interactions among loops. Luyben (1986) developed Biggest Log Modulus (BLT) tuning method for multi-loop PI controllers. Each loop is tuned individually using Ziegler-Nichols method, then decrease all the proportional gain and increase all the integral time by the same factor  $F$ . The factor  $F$  is increased until the closed-loop stability is guaranteed. This method tends to be conservative, and the dynamic performance is not selected to satisfy control objectives, which can have different importance on different

variables. The multi-loop PID controller tuning problem can also be formulated as an optimization with integral error objective (e.g., Zhuang and Atherton, 1994). However, it is only suitable for small system (2x2 system in Zhuang's paper), and it cannot guarantee the global solution due to non-convexity of the formulations.

MPCs have a large number of tuning parameters, such as the controller execution period, the prediction horizon, the control horizon, the weighting  $Q$  for relative importance of the controlled variables, and the move suppression factor  $R$ . There are limited general guidelines about how to choose these parameters.

Execution period should be a small fraction of the closed-loop dynamic, e.g. less than one-tenth of the process time constant plus deadtime, if possible (Marlin, 2000). The prediction horizon is normally chosen to be long enough for the close-loop system to approach its steady state. The close-loop stability is guaranteed (for no plant-model mismatch) if the prediction horizon is infinite (Rawlings and Muske, 1993). The control horizon is normally chosen to be one-fourth to one-third of the prediction horizon. The control horizon should be shorter than or equal to the prediction horizon, and at least equal to the number of unstable poles in the plant (Rawlings and Muske, 1993). However, Morari and Lee (1999) pointed out that the system behavior is relatively insensitive to change in both prediction horizon and control horizon over a wide range of values. Therefore, weighting matrix  $Q$  and  $R$  are the typical tuning parameters adjusted to affect performance.

$Q$  is used to address the relative importance of the controlled variables, which can have very different engineering units and acceptable variations.  $Q$  also acts as the normalization factor to bring all controlled variables to the same base for comparison. The move suppression factor  $R$  is the weighting for manipulated variable movements. Increasing  $R$  penalizes the excessive incremental manipulated variable move, which slows down the feedback control and degrades the controlled variable performance. On the other hand, increasing  $R$  improves the robustness of the closed-loop system to model mismatch. But for finite prediction horizon the effect of the move suppression factor  $R$



may be “non-monotonic”, for example, increasing  $R$  may lead to instability, which is counter-intuitive. In practice, the weighting matrices  $Q$  and  $R$  are diagonal in most cases.

Choosing the weighting matrix  $Q$  and  $R$  quantitatively according dynamic performance is a very challenging problem. Shridhar and Cooper (1998) derived an analytical expression for computing move suppression factor  $R$ . They use  $R$  to reduce the condition number of the system matrix ( $A^T Q A + R$ ), but this does not directly relate to the dynamic performance criteria. Trierweiler and Farina (2003) used Robust Performance Number (RPN) as a scaling factor for  $R$ . A process that is difficult to control has big RPN value, which subsequently increases the magnitude of  $R$ . Again, this tuning strategy does not relate to dynamic performance directly.

## 2.3 Solving Technology of MINLP

Mixed integer nonlinear programming (MINLP) problems are encountered in a various applications from engineering to operation research. The control structure design problem is formulated as an MINLP in this research.

Dealing with MINLP problems presents two major challenges, the combinatorial nature of the integer variables and non-convexity of the continuous variable sub-problem. As the number of the integer variables increase, the possible combinations grow exponentially in a general mixed integer problem. The determination of a global solution of the nonconvex problem can take a very long computing time.

Many algorithms are developed for convex MINLP, such as Generalized Benders Decomposition (GBD) (Floudas *et al.*, 1989), Outer Approximation (OA) (Duran and Grossmann, 1986) and Branch and Bound (BB) (Quesada and Grossmann, 1992). All of these methods rely on relatively easy-to-compute relaxations to generate lower bounds. The lower bounds of GBD are based on duality theory. OA uses the accumulation of the linearized objective function and constraints to generate lower bounds. The lower bounds of BB are based on the continuous relaxation of integer variables.

Global algorithms are needed for nonconvex MINLP. Branch and reduce algorithm (Ryoo and Sahinidis, 1995) and  $\alpha$ BB algorithm (Adjiman *et al.*, 2000) are two common deterministic global algorithms for nonconvex MINLP. They all build on branch and bound framework with valid convex underestimation of the nonconvex relaxations.

The control structure design problem in this research is a special nonconvex MINLP. The general MINLP algorithm cannot solve it reliably, and often ends up in the worst-case scenario enumerating all possible combinations. The key to solve the problem efficiently is to take advantage of the special problem structure.

## 2.4 State of Art/Challenges

Heuristic methods are very convenient for designing known processes. They can be very efficient guide to reasonably good designs for known process. But these design rules are process specific and the ability to extrapolate to new conditions is unproved. If you want to design a control system for a new process or a new combination of the unit operators, heuristic rules may not be valid anymore.

Shortcut metrics are very powerful tools and have been widely used in control system synthesis. They only need very limited process information, mainly steady state process gain, and give very important insight about process directionality, robustness, integrity and performance. But the advantage is also disadvantage. Shortcut metrics are simple scalar indicators that ignore detailed dynamic transients, which crucial for complex dynamic process. Sometimes the design solely based on shortcut metrics can be misleading.

Optimization methods can consider the dynamic transient explicitly. They provide a very general framework to include many difficult control design specifications such as dynamic performance, robustness, integrity and constraints into one rigorous mathematical formulation. But the difficulty is how to solve them efficiently. The most optimal controller design methods assume a centralized structure. Any structural constraint imposed on controller would introduce non-convexity into the formulation, which increasing the solving difficulty significantly. In order to automate the structure

selection integer variables are needed to address the discrete structure decisions. Due to the combinatory nature of integer variables the computing time can be so long that the problem become intractable.

Controller tuning problem is an inherent part of the optimization method if the dynamic transient is evaluated explicitly. Very thorough research has been done in single loop tuning problem. However, there are very few publications on the tuning problem of multivariable system. Due to non-convexity, the optimal tuning problem of multivariable system is difficult to find global solution.

Our research would resolve these challenging problems by

- Integrating the shortcut metrics and the transient evaluation into one general multiple-objective formulation, from which we can obtain rigorous solutions.
- Evaluating the complete dynamic transient based on realistic scenario setting, which gives details of complex behavior cannot be characterized by a simple scalar.
- Tailoring the branch & bound algorithm by using process knowledge to guide the branching procedure and taking advantage of control theory to evaluate valid lower bounds at each branch and bound tree node.

## **Chapter 3**

### **Model and Problem Definition**

To design controls, a formulation is needed that systemically takes into account all important objectives, which requires a form of multi-objective optimization. In addition, the formulation should yield an optimization problem that is computationally tractable for a moderate to large number of variables. The model formulation used throughout this research is presented in this chapter. It is capable of representing many different aspects of control structure design, such as detailed dynamic transients, variable constraints, saturation behavior and structural specification, in one formulation. After the model formulation is presented, it is applied to the control design for a small (2x2), but realistic, loop pairing control design problem. The application demonstrates the importance of proper model and problem formulation, while being small enough to be solved by an exhaustive enumeration method. More efficient solution methods are developed in subsequent chapters using the formulation developed here.

We have already noted the essential role of control structure in control design in the process industries (Foss, 1973). Previous studies and industrial practices have determined that the structure has the dominant effect on performance when standard control algorithms (PID for single loop and MPC for multivariable) are used.

In this work, the goal is to determine the design for “normal” operation around a desired steady state for a continuous process subject to non-stationary disturbances. This desired steady state is typically determined through optimization studies performed off-line or in real-time. The common control objective is to keep the process as close to the desired steady state as possible. Although the process is inherently nonlinear, under these circumstances, a linear model, linearized at the desired steady state, is normally sufficient to represent the process behavior around the desired steady state; therefore, linearized models are used in this research. As a result, the designs might not be appropriate for atypical behavior, such as start-up or shutdown of the process.

Linear controllers are the industrial standard for regulatory control. Only few nonlinear controllers are used when large changes are made to operations, for example, startup or reactor grade changes. Therefore, this search considers only linear controllers.

### **3.1 Defining the Inputs**

The control design criteria are affected by many closed-loop characteristics, which if ignored (as in some previously published methods), can lead to poorly performing control systems. In this section, we will briefly introduce the important inputs that must be included in the control design definition and model.

There are three major exogenous inputs to the closed-loop system: setpoints of the controlled variables, measurement noise, and disturbances (Figure 3.1). Setpoints are the desired trajectory of the controlled variables, which can be a constant variable like product quality or can be time-varying as determined to optimize profit by a higher-level function that adjusts setpoints in the controller. Normally, we assume we have perfect knowledge and control of the current and future value of setpoints. However, disturbances and noise in the processes are not known precisely and must be represented by models developed from process experience and knowledge of sensors and the process. Note that we may measure some disturbances (and apply feedforward), but we cannot manipulate them. The disturbances in a multivariable system usually are correlated and have a strong directionality. As we will see, the direction of the disturbance and

magnitude of sensor noise can have a strong effect on control design. However, some researchers have ignored disturbances and noise to simplify the problem formulation.

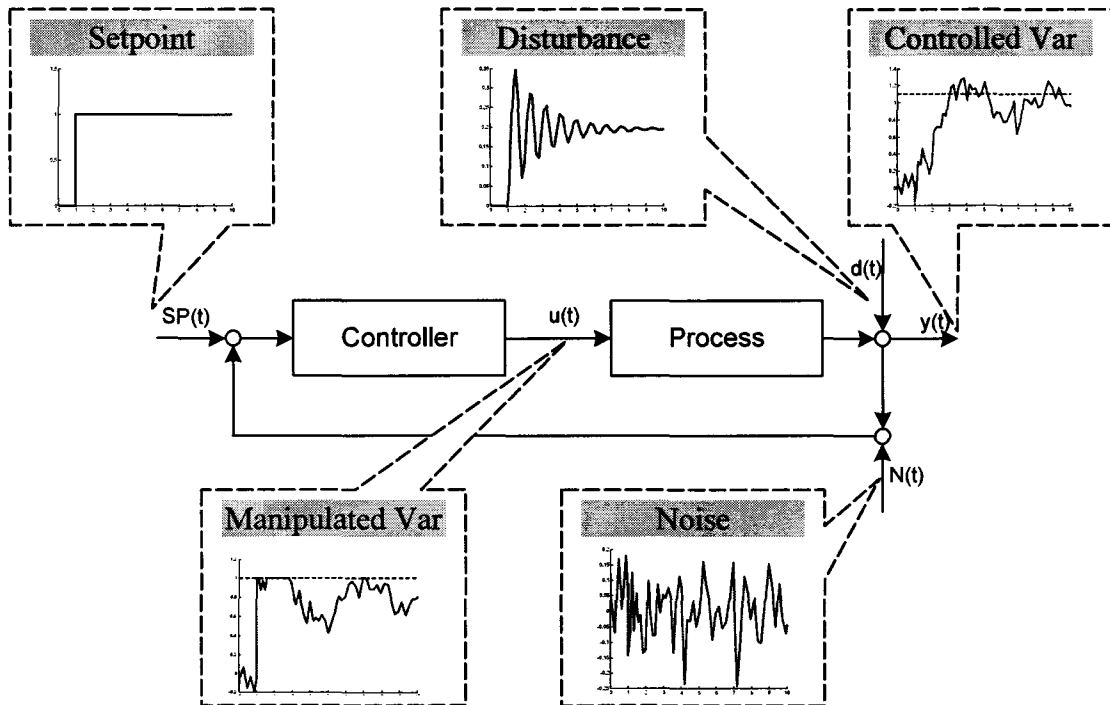


Figure 3.1 Realistic scenario for closed-loop systems

The formulation should also capture the physical limitations of the inputs, such as how fast and far the valve can open/close, the maximum capacity of the pump and compressor and so on. These specifications can be addressed by constraints in optimization model. A case study later in this chapter will show that different constraints can strongly affect the control structure decisions.

## 3.2 Multiple-objective Optimization

Control design involves many criteria. For example, Marlin (2000) defines seven categories of objectives; (1) safety, (2) environmental protection, (3) equipment protection, (4) smooth operation, (5) product quality, (6) profit, and (7) monitoring and diagnosis. For a typical process, several objectives exist within each category. Luyben (1998) offers a similar but somewhat more limited set of categories. The design

procedure should recognize the relative importance of competing objectives and definitely should not ignore important objectives for the sake of simplifying the computations.

Optimization provides a generic framework to deal with multiple criteria decision-making. Individually, each objective can be optimized, but the optimal solutions may conflict, so optimizing one objective sacrifices other objectives. We can seek a set of solutions referred to as *non-dominated* or *Pareto optimum* that finds a proper balance among all objectives in which one objective can be improved only at the expense of other objectives.

One way to formulate the multiple-objective optimization is to use a utility function  $U$  that relates all objectives based on some common basis as in equation (3.1).

$$\begin{aligned} \min_{\mathbf{x}} & U[J_1(\mathbf{x}), J_2(\mathbf{x}), \dots, J_{n_j}(\mathbf{x})] \\ \text{s.t. } & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0 \end{aligned} \quad (3.1)$$

where  $\mathbf{x}$  is  $r \times 1$  vector,  $\mathbf{h}(\mathbf{x})$  is  $n_h \times 1$  equality constraints and  $\mathbf{g}(\mathbf{x})$  is  $n_g \times 1$  inequality constraints. Multiple objective functions  $J_1(\mathbf{x}), J_2(\mathbf{x}), \dots, J_{n_j}(\mathbf{x})$  is converted into a scalar as  $U[J_1(\mathbf{x}), J_2(\mathbf{x}), \dots, J_{n_j}(\mathbf{x})]$ . The optimum of the utility function  $U$  is a Pareto optimum of all objectives. Because of the optimality condition any small perturbation from the optimum will cause the utility function value to increase, in other words, some of the objective values will increase, which subsequently increases the utility function value.

Defining a proper utility function is a challenge. One of most often used utility functions is a weighted linear summation of all objectives.

$$U[J_1(\mathbf{x}), J_2(\mathbf{x}), \dots, J_{n_j}(\mathbf{x})] = \sum_{q=1}^{n_j} w_q J_q(\mathbf{x}) \quad (3.2)$$

where  $w_q$  is the weighting for each objective. In spite of its mathematical simplicity, this method has drawbacks. The weighting should reflect the relative importance among criteria, but in most cases we do not know this sensitivity information ( $w_q$ ) before we solve the problem.

For multi-objective optimization in this research, we choose the *e-constraint method* (Steuer, 1986). It optimizes one objective with the other objectives bounded as constraints.

$$\begin{aligned}
 & \min_{\mathbf{x}} J_1(\mathbf{x}) \\
 & \text{s.t. } J_q(\mathbf{x}) \leq J_q^U \quad \forall q \neq 1 \\
 & \quad \mathbf{h}(\mathbf{x}) = 0 \\
 & \quad \mathbf{g}(\mathbf{x}) \leq 0
 \end{aligned} \tag{3.3}$$

where  $J_1(\mathbf{x})$  is the objective that is minimized and  $J_q(\mathbf{x})$  are the objectives bounded by  $J_q^U$ . This method can be solved by standard optimization software, and the result is easy to interpret. Steuer (1986) points out that by solving the problem several times with different  $J_q^U$ , we can get a very useful byproduct, sensitivity to changes in each bound (3.4), which is very useful information for trade-off decisions.

$$\left. \frac{\Delta J_1(\mathbf{x})}{\Delta J_q^U} \right|_{J_q^U} \tag{3.4}$$

Our major focus is dynamic performance; therefore, a measure of performance of the controlled variable deviation from set point is very natural choice for the objective function,  $J_1$ . Previous studies have indicated that the specific structure of this measure, e.g., error squared, absolute value, etc., has little impact on the optimization result.

Many other criteria could be included as the ancillary objectives,  $J_q$ . For example, one widely used metric is relative gain array (RGA), which is a necessary condition for good integrity.

### 3.3 Mathematical Formulation

In this section we develop a mathematical formulation that integrates all the previously discussed control design factors.



### 3.3.1 Linear Process Model

Our method is a model-based approach. The process models should be complex enough to capture the essential dynamic behaviors of the processes while being simple enough to yield tractable optimization problems. As previously discussed, linear dynamic models will be used in this work. These models can be determined from the linearization of a fundamental model or empirically, based on experiments in the operating process (Qin and Badgwell, 2003). We express the dynamic, linear model as a time-invariant discrete state space model of the process (Brogan, 1991).

$$\begin{aligned}\mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{W}\mathbf{d}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{V}\mathbf{d}(t) + \mathbf{N}(t)\end{aligned}\tag{3.5}$$

where  $\mathbf{x}$  is  $r \times 1$  state vector,  $\mathbf{y}$  is  $n \times 1$  controlled variable vector,  $\mathbf{u}$  is  $m \times 1$  manipulated variable vector,  $\mathbf{N}$  is  $n \times 1$  measurement noise vector,  $\mathbf{d}$  is  $d \times 1$  disturbances vector,  $\mathbf{A}$  is  $r \times r$  state coefficient matrix,  $\mathbf{B}$  is  $r \times m$  input coefficient matrix,  $\mathbf{C}$  is  $n \times r$  output coefficient matrix,  $\mathbf{W}$  is  $r \times d$  state disturbance coefficient matrix, and  $\mathbf{V}$  is  $n \times d$  output disturbance coefficient matrix.

### 3.3.2 PI Controller

PID controller is the most commonly used single-input, single-output (SISO) controller used in process control. Despite of its mathematical simplicity the PID controller is widely accepted by industry for its rapid and reliable calculation, intuitive tuning, and extensions to cascade, signal select and other structures (Astrom *et. al.*, 2001). Therefore, we choose the PI controller (a simplified version of PID controller without derivative mode) in our formulation.

Equations (3.6) is the full-position form of discrete PI controller, where  $sp_i(t)$  is the setpoint of  $i$ th controlled variables  $y_i(t)$ ,  $e_i(t)$  is the control error of  $i$ th controlled variables  $y_i(t)$ ,  $u_j(t)$  is the  $j$ th controller output and  $\Delta T$  is the execution time.

$$\begin{aligned}
 e_i(t) &= sp_i(t) - y_i(t) \\
 u_j(t) &= K_{Cji} \left[ e_i(t) + \frac{\Delta T}{T_{Iji}} \sum_t e_i(t) \right]
 \end{aligned}
 \tag{3.6}$$

The tuning parameters are controller gain  $K_{Cji}$  and integral time  $T_{Iji}$ . The cumulative summation term in PI controller equation provides integral effect that eliminates steady-state offset for disturbances that reach a constant value as time approaches infinity.

In 3.3.3 we enhance the SISO PI controller equation to handle saturation through anti-reset windup. The multi-loop PI control equation, which includes numerous single-loop controllers, will be introduced in Section 3.3.4.

### 3.3.3 Saturation

All manipulated variables have physical limitations. When large disturbances or setpoint changes occur, the controller outputs may exceed their desired bounds while the actual manipulated variables are always limited to within their maximum or minimum, as shown in Figure 3.2, which is referred to as *saturation*. Saturation can cause difficulty for the PI controller algorithm in (3.6). When the valve cannot be adjusted during the saturation period, the integral mode continues accumulating the nonzero error term  $e_i(t)$  for long periods of time; the result is a large magnitude of controller output that is beyond the valve's physical limitation. The situation is known as *reset windup*. Reset windup deteriorates the control performance because the difference between the large controller output and the actual manipulated variable take long time to unwind, and the valve cannot be adjusted until the controller output is equal to the actual manipulated variable. So the PI controller algorithm should have *anti-reset windup* protection.

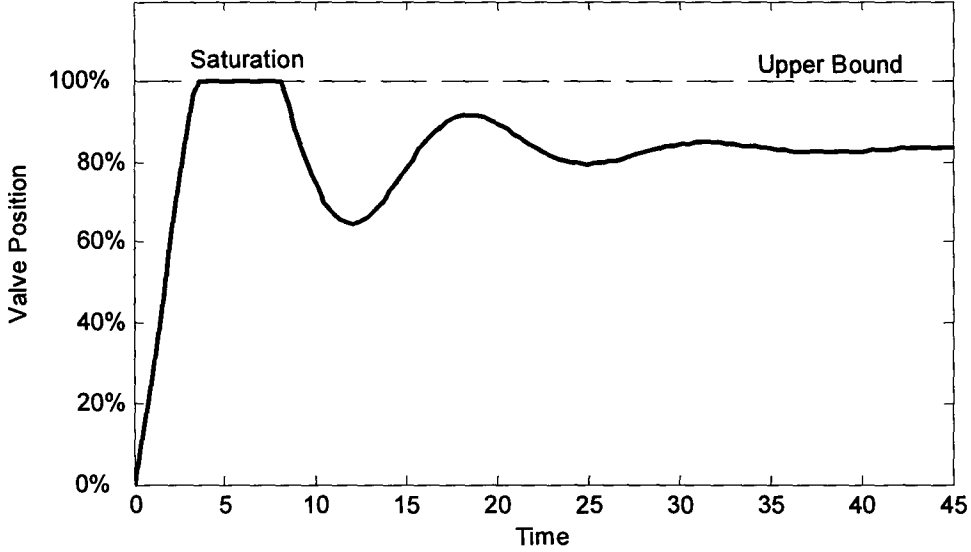


Figure 3.2 Valve saturation behavior

In our equation-based optimization framework we choose the velocity form of discrete PI controller in equation (3.7), which can be derived from (3.6).  $u_j^L$  and  $u_j^U$  are lower and upper bounds of the  $j$ th manipulated variables.

$$\begin{aligned}
 e_i(t) &= sp_i(t) - y_i(t) \\
 \Delta u_j(t) &= K_{Cji} \left[ e_i(t) - e_i(t-1) + \frac{\Delta T}{T_{Iji}} e_i(t) \right] \\
 &= K_{Cji} [e_i(t) - e_i(t-1)] + K_{Iji} e_i(t) \\
 u_j(t) &= u_j(t-1) + \Delta u_j(t) \\
 \text{if } u_j(t) &> u_j^U \text{ then } u_j(t) = u_j^U \\
 \text{if } u_j(t) &< u_j^L \text{ then } u_j(t) = u_j^L
 \end{aligned} \tag{3.7}$$

$$K_{Iji} = \frac{K_{Cji} \Delta T}{T_{Iji}} \tag{3.8}$$

The controller output is the increment of manipulated variable  $\Delta u_j(t)$ . If saturation happens, we can clamp  $u_j(t)$ , so that  $\Delta u_j(t)$  does not accumulate the integral error (Marlin, 2000). Therefore, equation (3.7) provides one approach to anti-reset windup.

The logical condition in equation (3.7) can be expressed mathematically as:

$$u_j(t) = \begin{cases} u_j^L & u_j(t-1) + \Delta u_j(t) \leq u_j^L \\ u_j(t-1) + \Delta u_j(t) & u_j^L \leq u_j(t-1) + \Delta u_j(t) \leq u_j^U \\ u_j^U & u_j(t-1) + \Delta u_j(t) \geq u_j^U \end{cases} \quad (3.9)$$

Equation (3.9) is part of the algebraic equation system that describes the dynamic behavior of the control system. However, it is difficult to be solved simultaneously using standard numerical methods, e.g. Newton method.

Equation (3.9) can be translated into an equivalent bilinear system with complementarity constraints (Marlin and Young, 1998; Baker and Swartz, 2004).

$$u_j(t) = u_j(t-1) + \Delta u_j(t) - s_j^U(t) + s_j^L(t) \quad (3.10a)$$

$$[u_j(t) - u_j^L] \cdot s_j^L(t) = 0 \quad (3.10b)$$

$$[u_j^U - u_j(t)] \cdot s_j^U(t) = 0 \quad (3.10c)$$

$$u_j^L \leq u_j(t) \leq u_j^U \quad (3.10d)$$

$$\begin{aligned} s_j^U(t) &\geq 0 \\ s_j^L(t) &\geq 0 \end{aligned} \quad (3.10e)$$

Equations (3.10a) to (3.10e) are equivalent to equation (3.9). Two slack variables,  $s_j^U(t)$  and  $s_j^L(t)$ , express the amount that the controller output violates the bound on the manipulated variable. For instance, if the calculated action  $u_j(t-1) + \Delta u_j(t)$  is less than its lower bound  $u_j^L$ , the actual manipulated variable output  $u_j(t)$  is clamped to its lower bound by allowing slack variable  $s_j^L(t)$  to be nonzero. This corresponds to the first case in (3.9).

The complementarity constraints, (3.10b) and (3.10c), imply the logical switches among three different cases in (3.9), which guarantee that slack variables become nonzero only when the manipulated variables reach their limits. For instance, if manipulated variable  $u_j(t)$  is less than its upper bound  $u_j^U$ , that means inequality  $u_j(t) \leq u_j^U$  is not active, and by the complementarity constraints (3.10c), the slack variable  $s_j^U(t)$  must be zero. We note that this formulation prevents reset-windup because it stops integration when a manipulated variable saturates.

### 3.3.4 Integer Structural Constraints

The previous PI controller and saturation discussions are based on SISO system, which can be easily extended to multivariable controller system. Equation (3.11) and (3.12) are the multivariable formulas for PI controllers, which exchange the scalar variables with corresponding vector or matrix variables given in Table 3.1. The symbol  $n$  is the number of controller variables, and  $m$  is the number of manipulated variables. The variables in multivariable controller are vectors except  $\mathbf{K}_C$  and  $\mathbf{K}_I$  are matrices, which are the tuning parameters for all possible loop pairings.

$$\begin{aligned}\mathbf{e}(t) &= \mathbf{sp}(t) - \mathbf{y}(t) \\ \Delta \mathbf{u}(t) &= \mathbf{K}_C [\mathbf{e}(t) - \mathbf{e}(t-1)] + \mathbf{K}_I \mathbf{e}(t) \\ \mathbf{u}(t) &= \mathbf{u}(t-1) + \Delta \mathbf{u}(t)\end{aligned}\tag{3.11}$$

$$K_{I_{ji}} = \frac{K_{C_{ji}} \Delta T}{T_{I_{ji}}}\tag{3.12}$$

Equation (3.10) can be extended to multivariable system as equation (3.13) in the same fashion.

$$\begin{aligned}\mathbf{u}(t) &= \mathbf{u}(t-1) + \Delta \mathbf{u}(t) - \mathbf{s}^U(t) + \mathbf{s}^L(t) \\ [u_j(t) - u_j^L] \cdot s_j^L(t) &= 0 \\ [u_j^U - u_j(t)] \cdot s_j^U(t) &= 0 \\ \mathbf{u}^L &\leq \mathbf{u}(t) \leq \mathbf{u}^U \\ \mathbf{s}^U(t) &\geq 0 \\ \mathbf{s}^L(t) &\geq 0\end{aligned}\tag{3.13}$$

Table 3.1 Variables in multivariable controller

|                                     | SISO controller | Multivariable controller |              |
|-------------------------------------|-----------------|--------------------------|--------------|
|                                     | Symbol          | Symbol                   | Dimension    |
| Controlled variable                 | $y_i(t)$        | $\mathbf{y}(t)$          | $n \times 1$ |
| Control error variable              | $e_i(t)$        | $\mathbf{e}(t)$          | $n \times 1$ |
| Set point                           | $sp_i(t)$       | $\mathbf{sp}(t)$         | $n \times 1$ |
| Manipulated variable                | $u_j(t)$        | $\mathbf{u}(t)$          | $m \times 1$ |
| Upper bound of manipulated variable | $u_j^U$         | $\mathbf{u}^U$           | $m \times 1$ |
| Lower bound of manipulated variable | $u_j^L$         | $\mathbf{u}^L$           | $m \times 1$ |
| MV slack variable                   | $s_j^U$         | $\mathbf{s}^U$           | $m \times 1$ |
| MV slack variable                   | $s_j^L$         | $\mathbf{s}^L$           | $m \times 1$ |
| Controller output                   | $\Delta u_j(t)$ | $\Delta \mathbf{u}(t)$   | $m \times 1$ |
| Controller proportional gain        | $K_{Cji}$       | $\mathbf{K}_C$           | $m \times n$ |
| Controller integral gain            | $K_{Iji}$       | $\mathbf{K}_I$           | $m \times n$ |

The values of the elements of the controller tuning parameter matrices  $\mathbf{K}_C$  and  $\mathbf{K}_I$  directly represent the controller structure, that is, the connection between controlled variables and manipulated variables. Where no feedback exists, the values of the tuning constants between a measurement and controller output are zero. Where feedback is allowed, the tuning constant values are allowed within the upper and lower bounds (that are non-zero). This method for defining the controller structure can be further modeled by a  $m \times n$  binary matrix  $\delta$  that has the same size as  $\mathbf{K}_C$  and  $\mathbf{K}_I$ . The value of each element in  $\delta$  is defined in as following

$$\delta_{ji} = \begin{cases} 1 & u_j \text{ can be influenced by feedback from } y_i \\ 0 & \text{Otherwise} \end{cases} \quad (3.14)$$

where subscription  $j$  is for manipulated variables and subscription  $i$  is for controlled variables. If we want to model a multivariable process with individual, single-loop controllers, additional constraints are required, as shown in the following.

$$\begin{aligned} -M_{ji}\delta_{ji} &\leq K_{Cji} \leq M_{ji}\delta_{ji} \\ -M_{ji}\delta_{ji} &\leq K_{Iji} \leq M_{ji}\delta_{ji} \end{aligned} \quad (3.15)$$

$$\sum_i \delta_{ji} = 1, \quad \sum_j \delta_{ji} = 1 \quad (3.16)$$

where  $M_{ji}$  is a large positive number that bounds  $K_{Cji}$  and  $K_{Iji}$ .

Equation (3.15) builds the logical relation between binary matrix  $\delta$  and controller tuning parameter matrices  $\mathbf{K}_C$  and  $\mathbf{K}_I$ . It states that each controller tuning parameter,  $K_{Cji}$  and  $K_{Iji}$  can be nonzero only when the corresponding  $\delta_{ji}$  is one, which means the pairing  $u_j - y_i$  is chosen. Equation (3.16) limits the possible structure to having only one controller (with non-zero tuning parameters) between a measurement and manipulated variable. We often refer to this as a “diagonal structure” that corresponds to a loop-pairing problem.

### 3.3.5 Model Mismatch

The model mismatch is inevitable due to the complexity of chemical processes and the simplification of the linearization. It is assumed that the best control structure design selected with the linear model would also yield the best closed-loop performance on the real nonlinear chemical processes. Therefore, the robustness of the design to model mismatch must be considered in the problem formulation.

In order to keep the formulation simple yet realistic, we use multiple process models to address model mismatch.

$$\begin{aligned} \mathbf{x}_k(t+1) &= \mathbf{A}_k \mathbf{x}_k(t) + \mathbf{B}_k \mathbf{u}_k(t) + \mathbf{W}_k \mathbf{d}(t) \\ \mathbf{y}_k(t) &= \mathbf{C}_k \mathbf{x}_k(t) + \mathbf{V}_k \mathbf{d}(t) + \mathbf{N}(t) \end{aligned} \quad k \in [1, \dots, n_k] \quad (3.17)$$

Equation (3.17) represents  $n_k$  different process models; subscription  $k$  denotes different (mismatch) process models. The same disturbance  $\mathbf{d}(t)$  and measurement noise  $\mathbf{N}(t)$  are used across different models to make the result comparable. Although the identical dimension of  $\mathbf{x}_k$ ,  $\mathbf{A}_k$ ,  $\mathbf{B}_k$ ,  $\mathbf{C}_k$ ,  $\mathbf{W}_k$  and  $\mathbf{V}_k$  between different models is generally not required, we assume it is true in this research.

The selection of mismatched model is not rigorous in this research. In case studies we choose several cases that would reduce the stability margin for the closed-loop system, e.g. increase all steady state gains or dead time in the measurements, to get realistic controller tunings.

$$\begin{aligned}
& \min_{\delta, \mathbf{K}_C, \mathbf{K}_I, \mathbf{s}_k^U(t), \mathbf{s}_k^L(t)} \sum_k \sum_{t=1}^{tf} [\mathbf{e}_k(t)^T \mathbf{Q} \mathbf{e}_k(t)] \\
& s.t. \\
& \mathbf{x}_k(t+1) = \mathbf{A}_k \mathbf{x}_k(t) + \mathbf{B}_k \mathbf{u}_k(t) + \mathbf{W}_k \mathbf{d}(t) \\
& \mathbf{y}_k(t) = \mathbf{C}_k \mathbf{x}_k(t) + \mathbf{V}_k \mathbf{d}(t) + \mathbf{N}(t) \\
& \mathbf{e}_k(t) = \mathbf{s} \mathbf{p}(t) - \mathbf{y}_k(t) \\
& \Delta \mathbf{u}_k(t) = \mathbf{K}_C [\mathbf{e}_k(t) - \mathbf{e}_k(t-1)] + \mathbf{K}_I \mathbf{e}_k(t) \\
& \mathbf{u}_k(t) = \mathbf{u}_k(t-1) + \Delta \mathbf{u}_k(t) - \mathbf{s}_k^U(t) + \mathbf{s}_k^L(t) \\
& \mathbf{u}^L(t) \leq \mathbf{u}_k(t) \leq \mathbf{u}^U(t) \\
& \mathbf{y}^L(t) \leq \mathbf{y}_k(t) \leq \mathbf{y}^U(t) \\
& [\mathbf{u}_j(t) - \mathbf{u}_j^L] \cdot \mathbf{s}_j^L(t) = 0 \\
& [\mathbf{u}_j(t) - \mathbf{u}_j^U] \cdot \mathbf{s}_j^U(t) = 0 \\
& \mathbf{s}_k^U(t) \geq 0 \\
& \mathbf{s}_k^L(t) \geq 0 \\
& -M_{ji} \delta_{ji} \leq K_{C,ji} \leq M_{ji} \delta_{ji} \\
& -M_{ji} \delta_{ji} \leq K_{I,ji} \leq M_{ji} \delta_{ji} \\
& \sum_i \delta_{ji} = 1, \quad \sum_j \delta_{ji} = 1 \\
& k \in [1, \dots, n_k] \quad t \in [1, \dots, tf] \quad j \in [1, \dots, m] \quad i \in [1, \dots, n]
\end{aligned} \tag{3.18}$$

The optimization formulation (3.18) combines multiple linear state space models (3.17), PI controller equations (3.11), manipulated variable saturation equations (3.13), and structural constraints (3.15) and (3.16). The sum of squared error of controlled variables is the objective function since the dynamic performance of the controlled variables is the major concern here. The robust controller structure and tuning is chosen to optimize the performance when considering multiple process models simultaneously,



in several scenarios.  $\mathbf{x}_k$  is  $r \times 1$  state vector,  $\mathbf{y}_k$  is  $n \times 1$  controlled variable vector,  $\mathbf{u}_k$  is  $m \times 1$  manipulated variable vector,  $\mathbf{N}$  is  $n \times 1$  measurement noise vector,  $\mathbf{d}$  is  $d \times 1$  disturbances vector,  $\mathbf{A}_k$  is  $r \times r$  state coefficient matrix,  $\mathbf{B}_k$  is  $r \times m$  input coefficient matrix,  $\mathbf{C}_k$  is  $n \times r$  output coefficient matrix,  $\mathbf{W}_k$  is  $r \times d$  state disturbance coefficient matrix,  $\mathbf{V}_k$  is  $n \times d$  output disturbance coefficient matrix,  $\mathbf{e}_k$  is  $n \times 1$  control error vector,  $\mathbf{sp}$  is  $n \times 1$  set point vector,  $\mathbf{K}_C$  is  $m \times n$  proportional gain matrix,  $\mathbf{K}_I$  is  $m \times n$  integral gain matrix,  $\mathbf{u}^L$  and  $\mathbf{u}^U$  are  $m \times 1$  upper and lower bound vectors of manipulated variables,  $\mathbf{y}^L$  and  $\mathbf{y}^U$  are  $n \times 1$  upper and lower bound vectors of controlled variables,  $\mathbf{s}^L$  and  $\mathbf{s}^U$  are  $m \times 1$  slack variable vectors,  $\delta$  ( $\delta_{ji}$ ) is  $m \times n$  binary matrix that represents control structure,  $\mathbf{M}$  ( $M_{ji}$ ) is  $m \times n$  matrix with large position numbers that bound  $K_{Cji}$  and  $K_{Iji}$ ,  $tf$  is the time horizon that the optimization problem is defined upon, and  $\mathbf{Q}$  is  $n \times n$  weighting matrix that defines the relative importance among the controlled variables. Values in  $\mathbf{Q}$  are limited to yield a positive definite diagonal matrix, with diagonal elements that represent the relative importance of the deviation of each controlled variable from its set point. Subscription  $k$  denotes the index of multiple process models,  $i$  denotes the index of controlled variables,  $j$  denotes the index of manipulated variables.

Note that the controller design parameters, i.e.,  $\mathbf{K}_C$ ,  $\mathbf{K}_I$  and  $\delta_{ji}$ , do not have subscription  $k$ . This indicates the same controller design is applied to all mismatch cases. In this way, robustness in the dynamic performance is included in the optimization problem (3.18).

Stability is an important control property. In order to check closed-loop stability, we need to evaluate the eigenvalues of the closed-loop system, which demands intensive computation. Therefore, we did not include the stability criteria explicitly in the problem formulation. In the systematic solving process (please refer to Chapter 4) we can add stability as an additional criterion to post-check the full-integer candidate solutions. If the solution is unstable, we do not update the upper bound and discard the solution. Because there will be few full-integer candidates to be evaluated this stability check approach is computationally tractable.

### 3.3.6 Formulation with Additional Criteria

Extra constraints can be added into the optimization formulation (3.18) to address additional design requirements. The following equation forces the loop pairing we choose to have a positive RGA value, which is a necessary condition for integrity.

$$\sum_j \lambda_{ij} \delta_{ji} \geq 0 \quad \forall i \quad (3.19)$$

where  $\lambda_{ij}$  is the elements of the RGA matrix  $\Lambda$  for all possible pairings. Each element is a member of the following matrix.

$$\Lambda = \mathbf{G}(0) \otimes [\mathbf{G}(0)^{-1}]^T \quad (3.20)$$

where  $\mathbf{G}(0)$  is  $n \times m$  steady state gain matrix of the process model,  $\otimes$  is the Hadamard (element-by-element) product (Horn and Johnson, 1994).

As we will see in the case studies, enforcing integrity has a major effect on the control performance. Therefore, we will solve the problem with and without (3.19) to determine whether the good integrity property leads to excessive degradation in dynamic control performance.

The following case study is solved for several different formulations modified from equation (3.18). It demonstrates that all components in equation (3.18) are essential for proper performance evaluation.

## 3.4 Case Study: Fluidized Catalytic Cracker (2×2)

We apply the design model and objectives to the loop-pairing problem for a fluidized catalytic cracker (FCC) unit shown in Figure 3.3. Again, this example was chosen for its small number of controlled and manipulated variables and for the challenges in the design. The process converts heavy oil into lighter and more valuable products, such as gasoline and fuel oil. The reaction section has two major units, one is a plug flow (transportation) reactor called a riser, which has a residence time of only a few seconds. The heavy oil and hot catalyst enter the bottom of the riser. The reaction finishes in seconds and the vapor products leave from the top of the riser while the

catalyst separates due to reduced velocity and cyclones. The catalyst is collected in the riser vessel, where it is steam stripped to remove volatile hydrocarbons, and flows into the regenerator. The regenerator is a fluidized bed vessel, in which the catalyst is regenerated by burning the coke produced in the reactor using air entering the vessel from the bottom. The inventory of catalyst in a typical regenerator is on the order of 60 tons (Gary and Handwerk, 1984). Two temperatures must be controlled to achieve smooth operation. The temperature of the riser ( $T_{ris}$ ) is directly related to product quality and yield and to temperature limits of key equipment; therefore, it should be tightly controlled. The temperature of the regenerator ( $T_{rgn}$ ) is not as crucial as the riser temperature ( $T_{ris}$ ), but it should be controlled in a range about its set point. Two manipulated variables are flow of catalyst ( $F_{cat}$ ) and flow of air ( $F_{air}$ ), indicated by the two valves in Figure 3.3.

The FCC unit has different operating conditions. In this case study we consider the partial combustion mode that has insufficient air in the regenerator to convert all the coke to  $\text{CO}_2$ ; therefore, the flue gas is a mixture of nitrogen, carbon monoxide and carbon dioxide and water vapor.

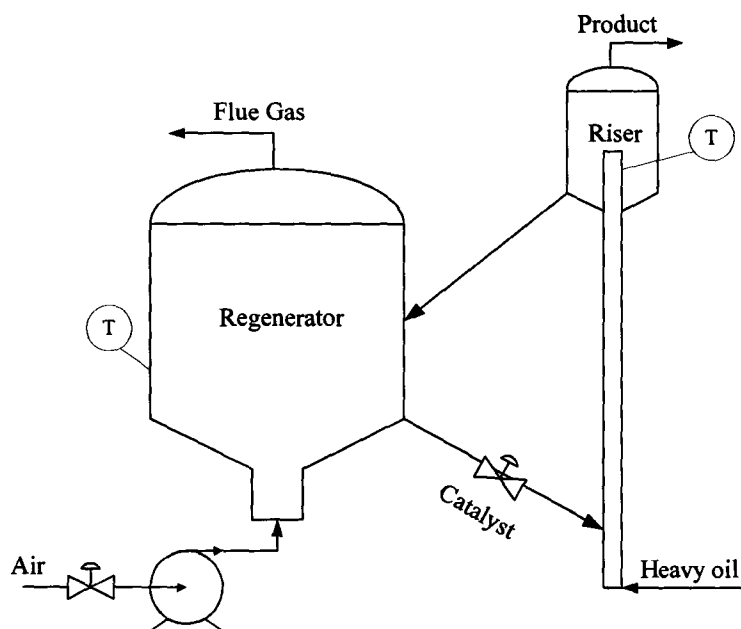


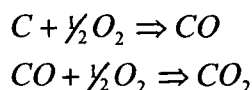
Figure 3.3 Fluidized catalytic cracker

The linearized model (Table 3.2) used in this study is based on Arbel *et al.* (1996). In Arbel's original work, the transfer function for loop  $T_{ris}-F_{cat}$  has the same number of zeros and poles; therefore, the model is not strictly proper. Strictly proper transfer function means the numerator order is smaller than the denominator order at least by one. Real physical processes are strictly proper. So, we have added a fast pole (denoted by bold font) that is 10 times bigger than the fastest pole into the transfer function for loop  $T_{ris}-F_{cat}$  to make the system strictly proper without changing the dynamic behavior significantly.

Table 3.2 Open-loop transfer function of FCC model (time in seconds)

| $F_{air}$ (lb air/lb feed) |  |
|----------------------------|--|
| $T_{rgn}$ (°F)             | $\frac{1.5(.0003 + s)(.005 + s)(.0084 + s)(.010 + s)}{(.00029 + s)(.00085 + s)(.0047 + s)(.0093 + s)(.01 + s)}$                                    |
| $T_{ris}$ (°F)             | $\frac{0.86(.0003 + s)(.0055 + s)(.0084 + s)(.0099 + s)}{(.00029 + s)(.00085 + s)(.0047 + s)(.0093 + s)(.01 + s)}$                                 |
| $F_{cat}$ (lb cat/lb feed) |  |
| $T_{rgn}$ (°F)             | $\frac{-0.13(.0003 + s)(.0095 + s)(.0047 \pm .0013i + s)}{(.00029 + s)(.00085 + s)(.0047 + s)(.0093 + s)(.01 + s)}$                                |
| $T_{ris}$ (°F)             | $\frac{22(-0.00086 + s)(.00031 + s)(.0053 + s)(.0084 + s)(.0096 + s)}{(.00029 + s)(.00085 + s)(.0047 + s)(.0093 + s)(.01 + s)(\mathbf{0.1s + 1})}$ |

The open-loop step response (Figure 3.4) shows the system has slow dynamics that are close to first-order except for the loop  $T_{ris}-F_{cat}$ , which has a fast inverse response. This unique dynamic response is associated with the partial-combustion operating mode of regenerator, in which the coke on the surface of the catalyst reacts with the oxygen in the air to produce carbon monoxide and carbon dioxide.



The combustion of coke to  $\text{CO}_2$  generates three times as much heat and uses twice as much the air compared to its combustion to  $\text{CO}$  (Arbel *et al.*, 1995). Note that with constant airflow and increased coke concentration, the heat generated per mass of coke decreases due to the much faster rate of the reaction to  $\text{CO}$ . As the result, increasing the flow of catalyst  $F_{cat}$  will raise the riser temperature initially by adding more hot catalyst to the transport (essentially plug flow) reactor and generate more coke; however, increasing coke concentration reduces combustion heat in the regenerator; therefore, the regenerator bed temperature  $T_{rgn}$  will decrease, but slowly due to the large mass of catalyst and eventually will decrease the riser temperature. The inverse response of the loop  $T_{ris}-F_{cat}$  is the combination of these two opposite dynamics.

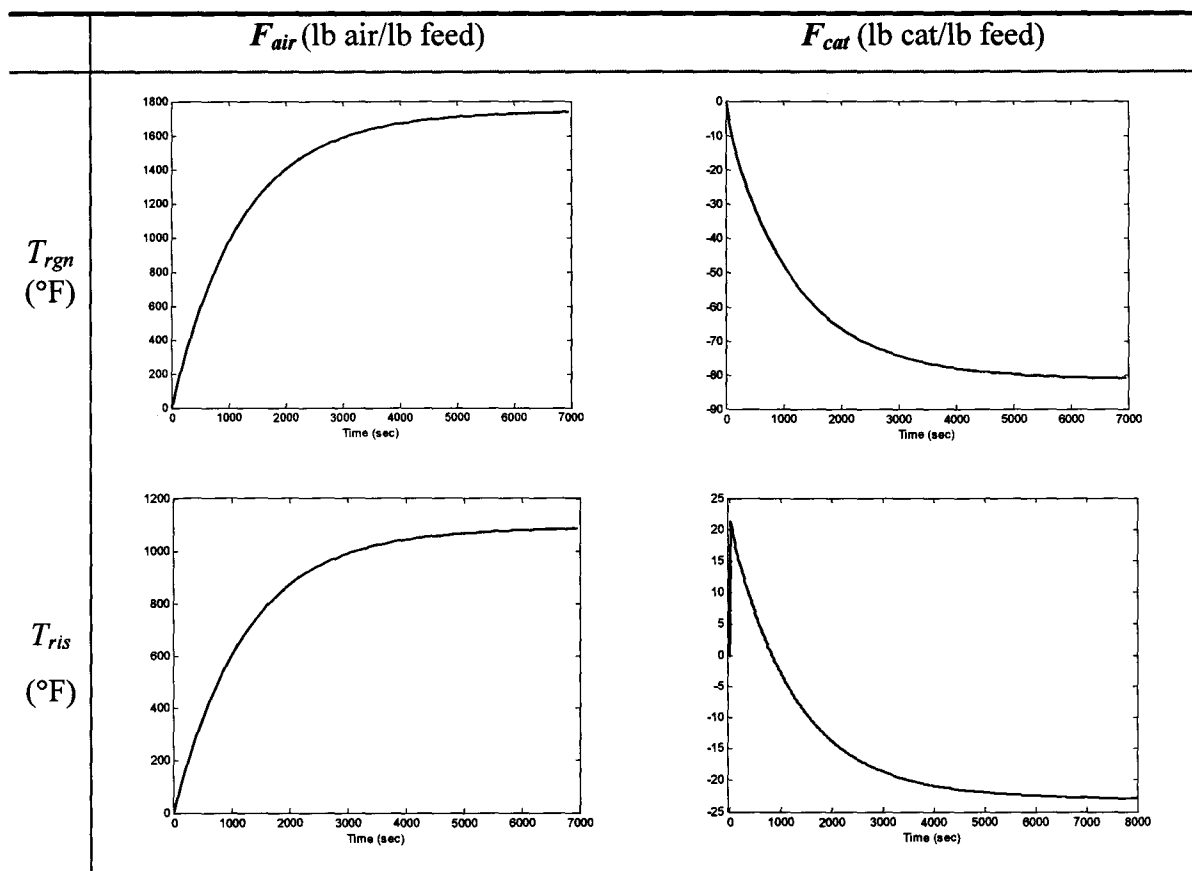


Figure 3.4 Open-loop step response of FCC model to step introduced at time = 0.

The RGA for this 2x2 system is shown in Table 3.3. If we want a control system with good integrity, we have only one choice, pairing on  $T_{reg}-F_{cat}$ ,  $T_{ris}-F_{air}$ . The other pairing has poor integrity, so that opening one control loop could result in the remaining system being unstable (without automatic retuning). Many heuristic design approaches (Shinskey, 1988) would dictate that the relative gain must be positive.

Table 3.3 RGA of FCC system

|           | $F_{air}$ | $F_{cat}$ |
|-----------|-----------|-----------|
| $T_{reg}$ | -0.8      | 1.8       |
| $T_{ris}$ | 1.8       | -0.8      |

However, we will proceed with a systematic evaluation of the dynamic performance. The evaluation will begin with the simplest (unrealistic) scenario and add additional factors in the scenario until a realistic definition is attained. Using this procedure, we will be able to follow the effects of the scenario on the control design. Table 3.4 lists five test cases, where the symbol “×” in the table indicates that the factor has been included in the dynamic simulation of the process.

Table 3.4 Cases for different scenarios

| Case                              | A                    | B                   | C          | D          | E          |
|-----------------------------------|----------------------|---------------------|------------|------------|------------|
| Pairing on RGA                    | +                    | +                   | +          | +          | -          |
| Mismatch                          |                      | ×                   | ×          | ×          | ×          |
| Noise                             |                      |                     | ×          | ×          | ×          |
| Capacity                          |                      |                     |            | ×          | ×          |
| $\text{Max}(K_p * K_c) \approx$   | $10^6$               | 100                 | 1          | 1          | 1          |
| $K_c(T_{reg}) (1/^\circ\text{F})$ | $-2 \times 10^4$     | $-2 \times 10^{-3}$ | -0.02      | -0.23      | 0.097      |
| $K_c(T_{ris}) (1/^\circ\text{F})$ | $2.5 \times 10^{-4}$ | 0.26                | 0.1        | 0.024      | 0.009      |
| $T_I(T_{reg}) (\text{min})$       | $5 \times 10^4$      | 0.01                | 13         | 20         | 0.05       |
| $T_I(T_{ris}) (\text{min})$       | $1 \times 10^{-2}$   | 100                 | 0.05       | 0.2        | 0.01       |
| ISE( $T_{ris}$ )                  | 123.2                | 142.0               | 338.7      | 390.7      | 125.1      |
| Dynamic transient                 | Figure 3.5           | Figure 3.6          | Figure 3.7 | Figure 3.8 | Figure 3.9 |

Case A uses only nominal model without any constraints; Case B is Case A with two additional mismatch models added; Case C is Case B with measurement noise added; Case D is Case C with capacity constraints added. Case A, B, C and D are controlled by positive RGA pairings; while Case E is Case D controlled by negative RGA pairing.  $ISE(T_{ris})$  is integral squared error of  $T_{ris}$  of the nominal model. The maximum  $K_p \cdot K_c$  is from the two controller pairings considered; the value is dimensionless.

Since this study is 2x2 system there are only 2 possible pairings. We will enumerate all possible pairings (by case studies with the integer variables fixed a priori to define each case) and evaluate the dynamic performance by solving equation (3.18) that optimizes the tuning for the scenario and loop pairing. The input signal is  $T_{ris}$  setpoint change +10 °F. ISE values in Table 3.4 and plots in Figure 3.5 - 3.9 are results for the optimized closed-loop transients with the nominal plant model.

We begin with **Case A**, the simplest scenario, without mismatch model, noise and valve saturation constraints. The performance looks very good (Figure 3.5). However, we notice an inconsistency with typical control practice. If the measurements and the manipulated variables are in proper ranges, a rough tuning guideline is that the product of the process gain  $K_p$  and the controller gain  $K_c$  should be around one. This is a practical guideline that takes into account of the smoothness of manipulated variable dynamic transient. This comes from a 1% change in the manipulated variables yielding about a 1% change in the controlled variable. If a larger controller gain is used aggressive manipulated variable movements can be expected, which may cause equipment damage. In Case A the product of one controller gain and process gain has a value of about  $10^6$ ! If we don't consider model mismatch, this tuning gives good performance, but a very small model mismatch will make the closed-loop system unstable.

Figure 3.6 shows the result for **Case B**, optimizing the base case and mismatch models simultaneously; recall that the same tuning is used for all mismatch scenarios. There are three models considered in this case study, the nominal model (Table 3.2), the nominal model with 20% gain increase in all loops and the nominal model with one second time delay added in all loops. Three closed-loop transients are calculated for three

different models with the same controller tuning, and the objective function is the summation of ISE of three closed-loop transients. In order to keep the closed-loop responses of all three models stable, the controller gain is significantly reduced but still larger than normally acceptable.  $T_{rgn}$  only moves about  $0.5^{\circ}\text{F}$  during the transient, which will be overwhelmed by sensor noise and process variability in a real plant. The big controller gain would amplify the noise and make the performance much worse than its prediction without noise.

Figure 3.7 shows the result of **Case C** with noise added in the formulation. The tuning now agrees with our guideline, and the performance is good. But in order to achieve this good performance,  $F_{air}$  has to make a rapid, large initial increase, which would require an air blower with about 600% extra capacity beyond its steady-state value. In practice most of chemical plants are operated very close to their maximum capacity to achieve maximum throughput. Arbel *et al.* (1996) pointed out that the air blower is normally operated very close to its maximum capacity, and a typical design provides no more than 2% for dynamic control. A 600% extra capacity requires a much bigger air blower than the normal operation needed, which would involve an increase in plant capital investment of more than three million dollars. The details of the capital cost estimation for the extra capacity of the air blower are in Appendix E. In addition, the large airflow rate during the transient would require higher capacity cyclones and piping and would likely result in costly catalyst losses due to entrainment of catalyst.

Figure 3.8 shows the performance of **Case D** with a realistic plant capacity, which is the formulation we propose. The tuning and transient becomes reasonable, and the dynamic performance is realistic. In Figure 3.8, the trajectory of  $F_{air}$  stays at its upper bound for several steps, which is saturation handled properly with the formulation in problem (3.18). We note that the performance of the riser temperature in Figure 3.8 is likely not adequate for this critical variable.

Therefore, we choose to relax the requirement for a positive RGA for the loop pairing. We report the results for the most demanding scenario (the same conditions as Case D) in Figure 3.9, which shows the excellent performance for the riser temperature.



The ISE with the negative RGA pairing is one-third of the value achieved with the positive RGA pairing.

The final decision on the control structure is made by the engineer based on the optimization results. For FCC process, the riser temperature is crucial, since it is directly related to yields, profit and equipment protection. We want to keep it as close to its set point as possible. If the regenerator temperature controller were placed in manual, the control system would be unstable because lack of integrity. However, the regenerator temperature would drift away slowly because of its large catalyst holdup, and the operators would have time to respond by placing the riser temperature controller in manual. Based on this analysis we would choose dynamic performance and give up integrity; we choose negative RGA pairing, which is  $T_{rgn}-F_{air}$ ,  $T_{ris}-F_{cat}$ . This is the industry standard control design for the FCC process considered in this example (Arbel, *et al.*, 1996). Note that it violates a commonly cited control heuristic (requiring good integrity), but it provides good dynamic performance and is recommended by engineers with experience controlling the FCC reactors.

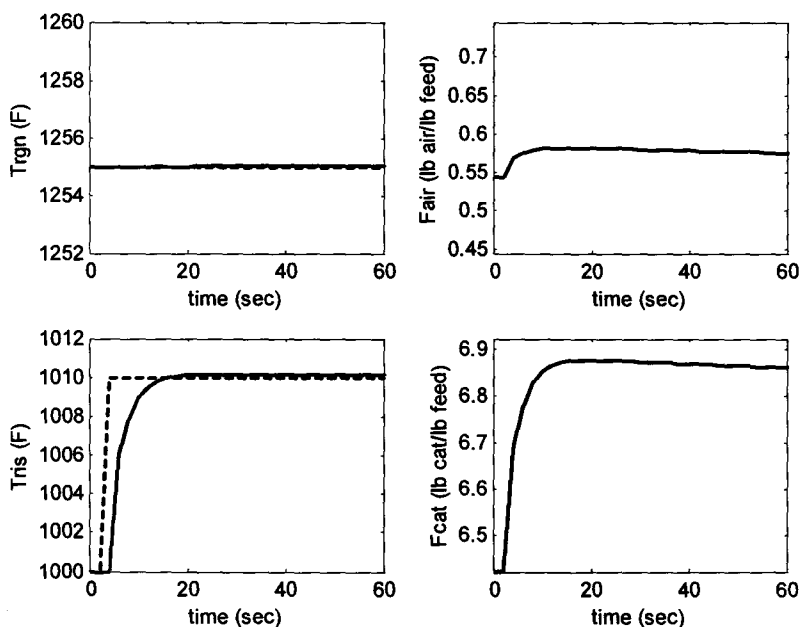


Figure 3.5 Case A

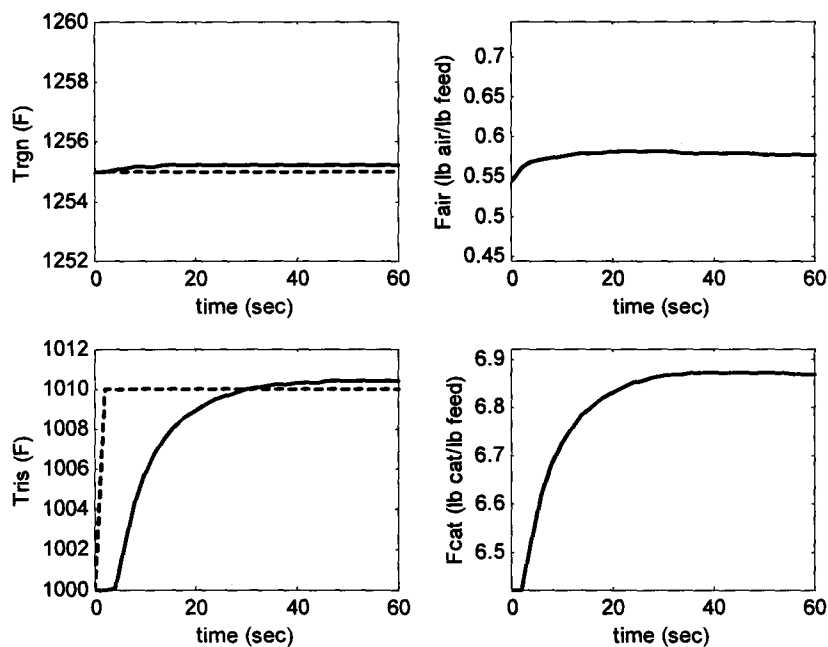


Figure 3.6 Case B

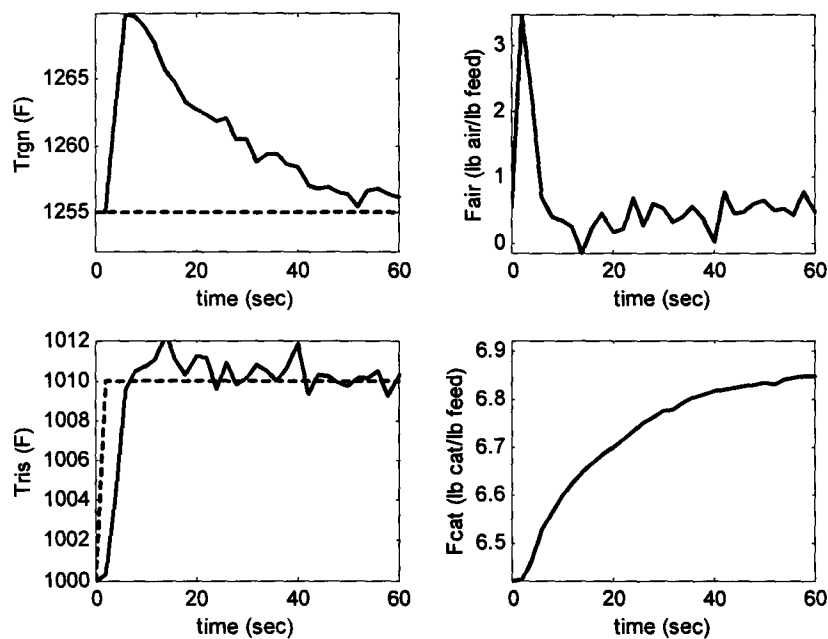


Figure 3.7 Case C

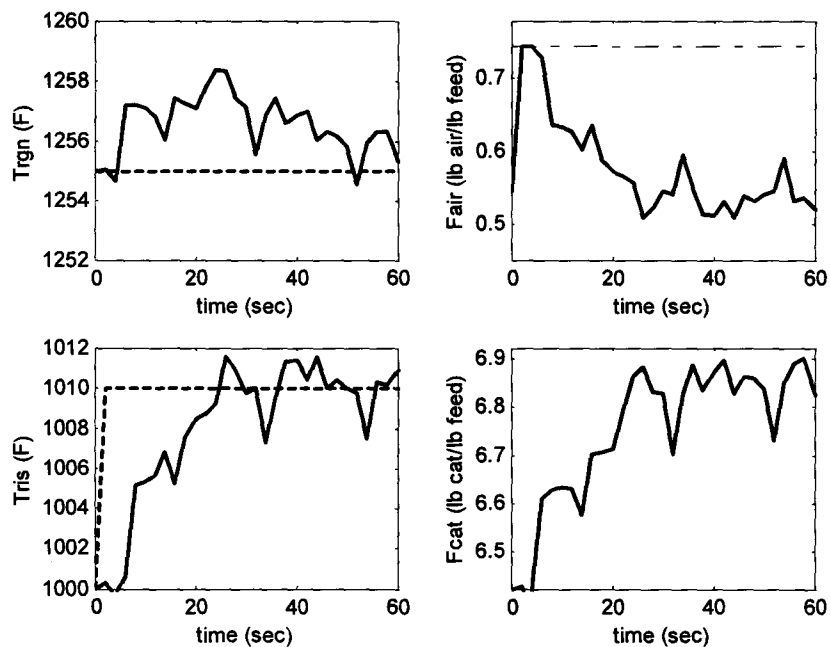


Figure 3.8 Case D

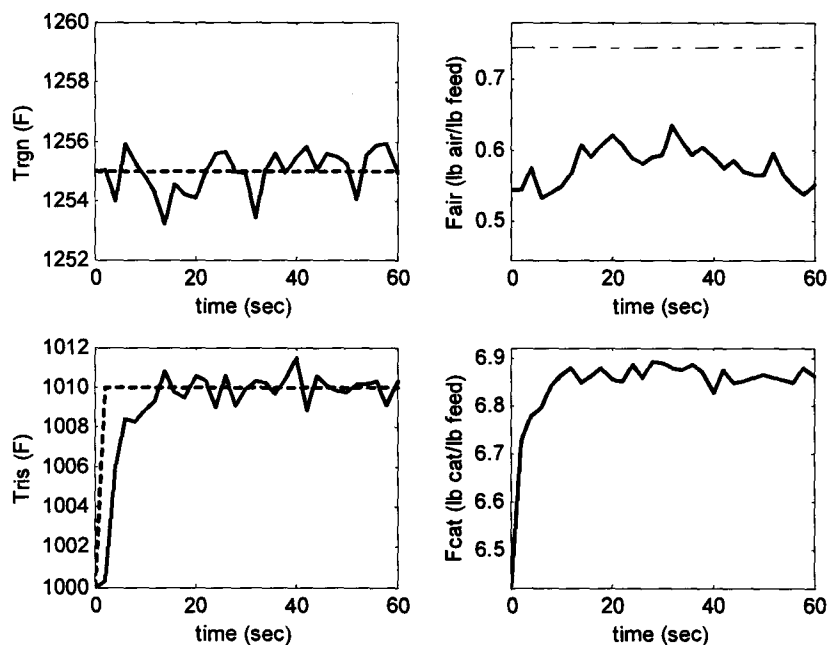


Figure 3.9 Case E

### 3.5 Summary

In this chapter, we developed a formulation that includes important control objectives, such as robustness (by multiple process models), noise, saturation (by complementarity constraints), equipment capacity (by variable bounds) and optional design criteria (e.g. RGA for integrity), into one multiple objective optimization problem. The goal is to select the best dynamic control system structure based on a realistic scenario. Further controller tuning might be required during implementation.

The formulation was applied to a small problem to demonstrate the efficacy for the formulation: incomplete scenario definition led to incorrect designs. All possible control structures were enumerated in order to find the optimum.

The proposed formulation belongs to a category of mathematical programming called Mixed Integer Nonlinear Programming (MINLP). The number of all possible control structures grows exponentially with the number of integer variables. The full enumeration of possible cases for larger problems would be computationally intractable. The nonlinear controller equations are classified as non-convex, which implies the potential of multiple local optimums. Complementarity constraints introduced by saturation formulation can cause convergence problem for general NLP solvers. Since we solve the dynamic transient explicitly in time domain the number of continuous variables increases very fast when the time horizon  $t_f$  increases. Finding an optimal control structure for medium to large-scale systems becomes a challenging problem due to these unique properties of the formulation.

The next chapter addresses the computational issues by presenting a systematic method for the loop-pairing control structure design problem that involves integer and continuous decisions in a much more efficient manner than enumeration.

## Chapter 4

# Systematic Selection of Loop Pairings

In the previous chapter we developed the mathematical formulation (3.18) that includes all important control objectives. The mathematical formulation is repeated here as problem (4.1) for reference.

$$\begin{aligned}
 & \min_{\delta, \mathbf{K}_C, \mathbf{K}_I, \mathbf{s}_k^U(t), \mathbf{s}_k^L(t)} \sum_k \sum_{t=1}^{tf} [\mathbf{e}_k(t)^T \mathbf{Q} \mathbf{e}_k(t)] \\
 & s.t. \\
 & \mathbf{x}_k(t+1) = \mathbf{A}_k \mathbf{x}_k(t) + \mathbf{B}_k \mathbf{u}_k(t) + \mathbf{W}_k \mathbf{d}(t) \\
 & \mathbf{y}_k(t) = \mathbf{C}_k \mathbf{x}_k(t) + \mathbf{V}_k \mathbf{d}(t) + \mathbf{N}(t) \\
 & \mathbf{e}_k(t) = \mathbf{s} \mathbf{p}(t) - \mathbf{y}_k(t) \\
 & \Delta \mathbf{u}_k(t) = \mathbf{K}_C [\mathbf{e}_k(t) - \mathbf{e}_k(t-1)] + \mathbf{K}_I \mathbf{e}_k(t) \\
 & \mathbf{u}_k(t) = \mathbf{u}_k(t-1) + \Delta \mathbf{u}_k(t) - \mathbf{s}_k^U(t) + \mathbf{s}_k^L(t) \\
 & \mathbf{u}^L(t) \leq \mathbf{u}_k(t) \leq \mathbf{u}^U(t) \\
 & \mathbf{y}^L(t) \leq \mathbf{y}_k(t) \leq \mathbf{y}^U(t)
 \end{aligned} \tag{4.1}$$

$$\begin{aligned}
[u_j(t) - u_j^L] \cdot s_j^L(t) &= 0 \\
[u_j(t) - u_j^U] \cdot s_j^U(t) &= 0 \\
s_k^U(t) &\geq 0 \\
s_k^L(t) &\geq 0 \\
-M_{ji}\delta_{ji} &\leq K_{Cji} \leq M_{ji}\delta_{ji} \\
-M_{ji}\delta_{ji} &\leq K_{Iji} \leq M_{ji}\delta_{ji} \\
\sum_i \delta_{ji} &= 1, \quad \sum_j \delta_{ji} = 1 \\
k \in [1, \dots, n_k] \quad t \in [1, \dots, tf] \quad j \in [1, \dots, m] \quad i \in [1, \dots, n]
\end{aligned}$$

The optimum result of problem (4.1) gives us the candidates for the best control structures. But solving this formulation is a challenging task. Although we solved a small 2×2 system by enumerating all possible pairings in Chapter 3, enumeration is not a general solution for control structure design problem. According Equation (4.2) the number of all possible pairings increases exponentially with the problem size, as shown in Table 4.1.

$$N(n) = n! \quad (4.2)$$

Table 4.1 The number of possible pairings

| System size (n)                            | 2      | 4      | 6      | 8                 | 10                | 12                |
|--|--------|--------|--------|-------------------|-------------------|-------------------|
| # of possible pairings (N)                 | 2      | 24     | 720    | $4.0 \times 10^4$ | $3.6 \times 10^6$ | $4.8 \times 10^8$ |
| Computing Time<br>assuming 1 second/node   | 2 sec  | 24 sec | 12 min | 11 hr             | 42 day            | 15 yr             |
| Computing Time<br>assuming 10 minutes/node | 20 min | 4 hr   | 5 day  | 277 day           | 68 yr             |                   |

For a computing time of 10 minutes per node, which is low for the solution of the non-linear, non-convex tuning problem per node, the computation becomes excessive for more than 4 variables. Even with a very optimistic computing time of 1 second/node, the enumeration method is unsuitable for more than 8 variables. Therefore, a systematic solving strategy is needed for large systems. We focus on square systems (the number of

controlled variables is equal to the number of manipulated variables) first, and then the methodology is extended to handle non-square systems.

In this chapter we limit attention to the loop-pairing problem that involves multiple single-loop feedback controllers. In Chapter 6 we will extend this methodology to a more challenging, block decentralized structure.

## 4.1 Combine Transient with Shortcut Metrics

In this chapter we first solve the optimization problem for the transient behavior using two generic MINLP solvers and one MILP reformulation approach. We demonstrate that these approaches cannot solve the optimization problem with reasonable computing time. Then, we present a new framework for solving the control structure design problem. The framework uses control concepts to guide branching procedure and a novel relaxation formulation to efficiently compute a useful lower bound. As described in Chapter 3, the design problem involves other criteria, which can be stated as logical conditions that can be added to each branch and bound node to represent additional design criteria. These logical constraints involve rapid computations and often eliminate many nodes; therefore, they are separated from and evaluated before the larger dynamic optimization. The new control design method can be easily extended to non-square systems. Two case studies demonstrate the efficiency of the method.

The solving strategy can be separated into two parts: transient evaluation and logical screening conditions including shortcut metrics and other heuristic rules. These two parts can be evaluated independently. Logical conditions that represent additional design objectives are formulated as constraints, which potentially save computing time by reducing the feasible region. Transient behavior, the major design criteria, is expensive to compute. Therefore, the idea is to perform the logical conditions screening (relatively low computational demanding) first to reduce the number of the candidate pairings, and then evaluate the dynamic transient for the remaining candidates, the number of which is much smaller than all possible pairings.

## 4.2 MINLP Solvers and MILP Reformulation

The control structure selection based entirely on dynamic performance (4.1) is a non-convex mixed integer nonlinear programming (MINLP). The difficulty of this problem is two-fold. First, the number of the possible structures grows exponentially with the problem size. Second, the non-convex controller tuning problem must be solved at each node in order to evaluate the dynamic transient. Based on experience we know there are many local optimum for the non-convex controller tuning problem (e.g., Vlachos *et al.*, 1999), and when using conventional optimization technology (based on local information), the final results depend on the initial values used for the tuning constants. Also, we can allow negative RGA pairings; recall that the relative gain is the ratio of gains for the open-loop and closed-loop system, and a negative value indicates that the “sign of the corrective action for an error” for the single-loop controller in a multiloop system is not the same as for the correction without interacting controllers. Therefore, some of the controller gains may have different signs from the open-loop process gains in order to stabilize the overall system. In addition, we do not know which controller gains should be switched before we solve the optimal tuning problem. Therefore, the results can differ greatly from commonly-used, single-loop tuning guidelines, so that good initial values for the controller tuning are not generally available.

We have applied state-of-the-art solvers designed for generic MINLP problems. Also, we have reformulated the problem as an MILP to avoid the non-convexity. Results from these studies, which indicate that the methods are not appropriate for this problem, are presented in this section.

### 4.2.1 MINLP Solvers

Two generic MINLP solvers, SBB and BARON, have been tested within the GAMS environment. SBB is a general local MINLP solver, which combines a standard branch and bound algorithm with the continuous sub-problems solved with a NLP solver supported by GAMS. The advantage of SBB is that you can choose the NLP solver, from



candidates such as CONOPT, MINOS and SNOPT, whichever is suitable for your specific NLP problem. However, the standard branch and bound algorithm is not optimized specifically for control structure selection problem.

BARON is a global solver that can handle MINLP directly. It uses the Branch-And-Reduce Optimization Navigator (BARON) method developed by Sahinidis (1996). Neumaier *et al.* (2005) compared many available global optimization solvers and concluded that BARON is the “*fastest and most robust*”. Unfortunately, BARON can only use MINOS or SNOPT as NLP solver, which in our experience do not perform well for the non-convex controller tuning problem.

#### 4.2.2 MILP Reformulation

The bilinear term in PI controller is the source of non-convexity. Kookos and Perkins (2002) proposed a linear disjunctive reformulation of PID controller that avoids the non-linearity in the bilinear terms. The approach is demonstrated on a P-only controller for simplicity in this section. The following equations represent the controller model for P-only controllers with structural constraints similar to problem (4.1).

$$\begin{aligned}
 u_j(t) &= \sum_i K_{ji} e_i(t) \\
 K_{ji}^L \delta_{ji} &\leq K_{ji} \leq K_{ji}^U \delta_{ji} \\
 \sum_j \delta_{ji} &= 1 \\
 \sum_i \delta_{ji} &= 1 \\
 j &\in [1, \dots, m] \quad i \in [1, \dots, n]
 \end{aligned} \tag{4.3}$$

where  $u_j$  is  $m \times 1$  manipulated variable vector,  $K_{ji}$  is  $m \times n$  proportional gain matrix,  $e_i$  is  $n \times 1$  control error vector,  $K_{ji}^L$  and  $K_{ji}^U$  are  $m \times n$  lower and upper bound matrix of  $K_{ji}$ ,  $\delta_{ji}$  is  $m \times n$  binary matrix represents control structure.

Equation (4.3) is a simple P-only controller with structural constraints, i.e., loop pairings. The controller equation has bilinear terms since  $K_{ji}$  and  $e_i(t)$  are variables. In order to remove the bilinear term in the controller equation, Kookos and Perkins (2002)

replace continuous controller gain matrix  $K_{ji}$  by a  $m \times n \times v$  constant controller gain matrix  $\kappa_{jil}$ , which consist of  $v$  pre-determined values for each controller gain in the  $l$  dimension in equation (4.4).

$$\begin{aligned}
 \Delta u_j(t) &= \sum_i \sum_l \kappa_{jil} \delta_{jil} e_i(t) = \sum_i \sum_l \kappa_{jil} E_{jil}(t) \\
 E_{jil}(t) &= \delta_{jil} e_i(t) \\
 \sum_j \delta_{jil} &= 1 \\
 \sum_i \delta_{jil} &= 1 \\
 \sum_l \delta_{jil} &= 1 \\
 j &\in [1, \dots, m] \quad i \in [1, \dots, n] \quad l \in [1, \dots, n_l]
 \end{aligned} \tag{4.4}$$

where the binary variable matrix  $\delta$  is expanded to include one additional dimension  $l$  to represent the choice of the predetermined values for each controller gain. The controller equation becomes linear, but there are still bilinear terms involve binary variables  $\delta_{jil}$  and continuous variables  $e_i(t)$  when  $E_{jil}(t)$  is calculated. These terms set all feedback errors to zero except the one error that corresponds to the selected pairing ( $ij$ ) of measurement and adjusted variables and the selected disjunctive controller tuning ( $l$ ).

Each bilinear term in the calculation of  $E_{jil}(t)$  can be replaced by four inequality constraints as shown in the following (Kookos and Perkins, 2002), which results in zero values for the error not selected by the disjunctive model through the integer variable  $\delta_{ijl}$ .

$$\begin{aligned}
 \Delta u_j(t) &= \sum_i \sum_l \kappa_{jil} E_{jil}(t) \\
 e_i^L \delta_{jil} &\leq E_{jil}(t) \leq e_i^U \delta_{jil} \\
 e_i(t) - e_i^U (1 - \delta_{jil}) &\leq E_{jil}(t) \leq e_i(t) - e_i^L (1 - \delta_{jil}) \\
 \sum_j \delta_{jil} &= 1 \\
 \sum_i \delta_{jil} &= 1 \\
 \sum_l \delta_{jil} &= 1 \\
 j &\in [1, \dots, m] \quad i \in [1, \dots, n] \quad l \in [1, \dots, n_l]
 \end{aligned} \tag{4.5}$$

This concept is extended to the PI controller and applied to the tuning problem. The conversion to a convex sub-problem is successful, but the formulation in equation (4.4) and its extension to PI control have two disadvantages with regard to computing time. First, the number of the binary variables in the linear formulation is significantly higher than the nonlinear formulation and depends on the number of the predetermined controller gain values (disjunctions) and the number of tuning constants ( $K_C$  and  $K_I$ ). Naturally, a large number of binary variables can prolong the computing time. Second, the equations in (4.5) are equivalent to the equations in (4.4) only when each binary variable is equal to zero or one. In the branch and bound sub-problems with continuous relaxation of binary variables,  $E_{jil}(t)$  can be any value between the maximum  $e_i^U$  and the minimum  $e_i^L$ , which means that the optimizer can adjust the value of the errors to give the best values of the adjusted variables (independent of the controller structure or tuning)! This relaxed sub-problem enables the optimizer to use information in the future to determine adjustments, which places almost no constraints on control action  $u_f(t)$ . Clearly, there is no clear structure for the controller when the binary variables are relaxed because the controller equations are not enforced for the unpaired loops. As a result, the lower bound calculation for the sub-problems is very inefficient, i.e., it underestimates the performance objective function and gives a very low lower bound value. Therefore, we expect that the branch and bound solution method will not be efficient.

We conducted a comparison study applying state-of-art MINLP and MILP solvers on control structure design problem. The test problem is the Tennessee Eastman problem, which will be described in detail in chapter 5. For now, we can describe the problem as a 5x5 control design (loop-pairing) problem. The modeling environment is GAMS 21.4. The test computer was a Pentium III with 512MB memory. Table 4.2 lists the problem size and solving time, along with the solvers used in the evaluation. All solvers use default settings. Table 4.3 lists the results after five days of calculation. Since none of the solvers reach a proved optimum, the upper and lower bound of the optimum is listed.

Table 4.2 Comparison of different solvers

| Solver                                | # of continuous variables | # of integer variables | Solving Time |
|---------------------------------------|---------------------------|------------------------|--------------|
| SBB (with CONOPT3)                    | 6011                      | 25                     | > 5 days     |
| BARON 7.2                             | 6011                      | 25                     | > 5 days     |
| CPLEX 9.0<br>(with reformulated MILP) | 14601                     | 200                    | > 5 days     |

Table 4.3 Comparison of the unproved solutions after 5 days

| Solver                                | Upper bound of the optimum | Lower bound of the optimum |
|---------------------------------------|----------------------------|----------------------------|
| SBB (with CONOPT3)                    | 92.79                      | 37.90                      |
| BARON 7.2                             | 92.79                      | 45.19                      |
| CPLEX 9.0<br>(with reformulated MILP) | 92.79                      | 23.51                      |

We choose CONOPT3 as the NLP solver in SBB, because it handles complementarity constraints better than MINOS. However, SBB did not take advantage of the special structure of the control problem. The general branch and bound algorithm branches each binary variable to zero or one equally, but branching to zero does not generate a clear block structure. The free binary variables are relaxed continuously between zero and one, which does not have clear physical meaning and generate difficult optimal controller tuning problems. Each of these difficult relaxation problems took more than 10 minutes to solve. SBB spent most of time computing these very loose relaxation problems, which did not improve the lower bound in the last four days of computation.

BARON can only utilize MINOS and SNOPT to solve NLP problem generated from binary variable relaxation. However, MINOS and SNOPT do not handle this particular non-convex NLP very well and about 5% of the relaxation problems did not converge to a local optimum. As the result, the lower bound was improved very slowly. The gap between the upper bound and lower bound did not reduce after days of computation even when the initial point was very close to the optimum.

CPLEX can solve each branch and bound node very fast because each sub-problem is an LP. However, without control system knowledge built in, CPLEX is unable to search the branch and bound tree efficiently. The solving process showed the branch and bound tree size (i.e., number of active nodes) grew monotonically until maximum memory usage limit reached, which indicated the search strategy is very inefficient and CPLEX basically enumerated all possible combinations. A large number of binary variables added in order to reformulate the problem as MILP made the number of possible combinations so large that a solution was not reached in five days.

The general MINLP solvers, SBB and BARON cannot find a solution after five-days of computation time; neither can the reformulated MILP be solved by CPLEX. These results show that the formulations using the general MINLP solvers and the MILP reformulation cannot solve the control structure design problem efficiently for large systems. As we will see, the 5x5 loop-pairing problem considered in this sub-section can be solved in 1 to 3 hours of computing using methods developed in this research.

### 4.3 Tailored Solving Strategy for Transient Evaluation

A tailored solving strategy is developed to take advantage of the control problem structure and to combine several existing solvers. The strategy using the general branch and bound framework has the three levels shown in Figure 4.1.

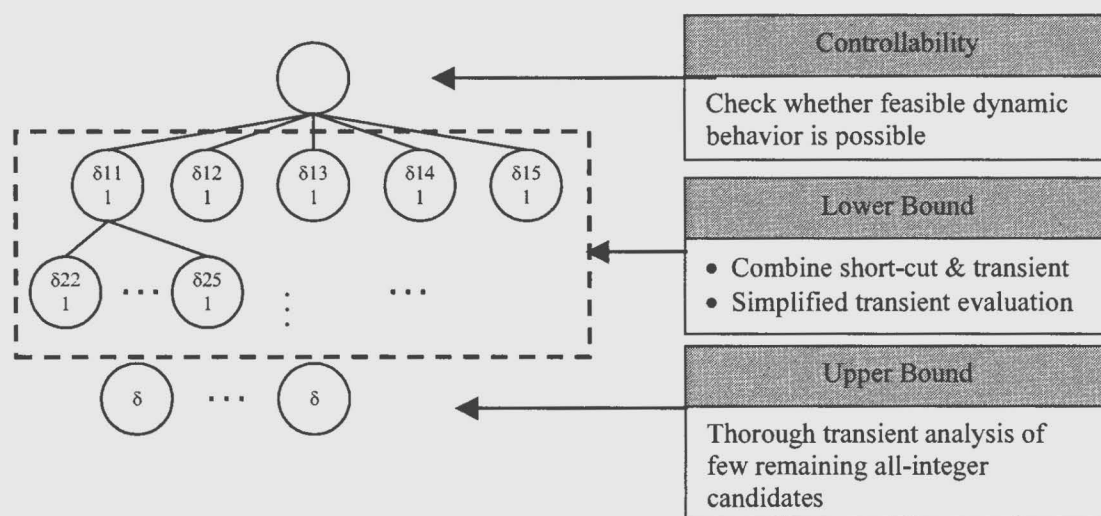


Figure 4.1 Brand & bound solving strategy

The first level contains the first node in the branch and bound tree that is usually the “all continuous” relaxation of the mixed integer problem. Here, we use the top node to determine the “best possible” feedback performance, which we can compare to the required performance. If a solution does not satisfy the hard constraints, we know that the problem cannot be solved by any control loop pairing and that the problem must be reformulated. We could relax some constraints, or we could modify the process to give more favorable dynamic responses. We will describe this level as checking the controllability of the formulation. In general, the term controllability is used for the ability to achieve a specified level of control performance. Many definitions of “controllability” exist (Rosenbrock, 1972), and we will consider achieving all performance requirements for the problem, i.e., any feasible solution, to indicate that the system is controllable. Only the dynamic performance is considered at this level.

The second level evaluates the optimum performance of partially relaxed problems. Both the dynamic performance and additional constraints (used to specify additional goals) are considered at this level. As will be explained, the formulation is tailored to the control design problem. The results are used to update the lower bound on the control performance.

At the third level, both the dynamic performance and additional objectives are considered. The third level calculates the detailed transients for all remaining fully integer candidates by optimizing the tuning. Each solution at the third level provides a candidate for the upper bound on the control performance, and the lowest upper bound is the best solution. We will discuss the three levels in detail in the following sub-sections.

### 4.3.1 Controllability Check

In the first level controllability is checked to determine whether the dynamic performance objectives can be achieved for this process. If not, the dynamic performance objectives or the process itself needs to be modified. In mathematical terms, we check the *feasibility* of the formulation.

$$\begin{aligned}
 & \min_{\mathbf{u}_k(t)} \sum_k \sum_{t=1}^{tf} [\mathbf{e}_k(t)^T \mathbf{Q} \mathbf{e}_k(t)] \\
 & s.t. \\
 & \mathbf{x}_k(t+1) = \mathbf{A}_k \mathbf{x}_k(t) + \mathbf{B}_k \mathbf{u}_k(t) + \mathbf{W}_k \mathbf{d}(t) \\
 & \mathbf{y}_k(t) = \mathbf{C}_k \mathbf{x}_k(t) + \mathbf{V}_k \mathbf{d}(t) + \mathbf{N}(t) \\
 & \mathbf{e}_k(t) = \mathbf{sp}(t) - \mathbf{y}_k(t) \\
 & \mathbf{y}^L(t) \leq \mathbf{y}_k(t) \leq \mathbf{y}^U(t) \\
 & \mathbf{u}^L(t) \leq \mathbf{u}_k(t) \leq \mathbf{u}^U(t) \\
 & k \in [1, \dots, n_k] \quad t \in [1, \dots, tf]
 \end{aligned} \tag{4.6a}$$

We check the controllability of problem (4.6a), which is a relaxation of the original problem (4.1) by removing the controller equations and integer constraints for the controller structure. The solution set of problem (4.6a) contains the solution set of the original problem (4.1) because it removes some of the constraints and enlarges the feasible region. Therefore, it provides a lower bound on every all-integer solution.

Problem (4.6a) is a convex QP problem that can be solved very efficiently. Having a feasible solution for the first level does not guarantee a feasible integer solution, but if problem (4.5) does not have a feasible solution, no integer solution will.

Properly selected manipulated variable and controlled variable bounds are critical for the controllability check. In general, these variable bounds should reflect physical limitation of equipment and the control objectives. For instance, the controlled variable bounds can be the maximum allowable variation for safety and product quality. The bounds could be formulated as a funnel shape bounds that converge to the setpoint, so that the controlled variable is returned to its set point near the end of the transient.

Problem (4.6a) defines an open-loop controller that may not maintain causality as required for a feedback controller. Figure 4.2 shows the difference between the closed-loop feedback controller and a non-causal open-loop controller. In Figure 4.2a, if a disturbance  $d$  enters the system at time  $t_0$ , the effect may be measured in a controlled variable  $y$  at a later time  $t^*$  due to dead time in the disturbance process and discrete sampling. Since the manipulated variable of the feedback controller is a function of the current and past error signals (deviation of controlled variables from set points), the manipulated variable  $u$  can only change due to feedback after time  $t^*$ . In Figure 4.2b, the open-loop controller responds to the disturbance even before the disturbance occurs, because the optimizer (that involves a simultaneous solution to the defining process equations) has information about the entire dynamic response. Without the closed-loop controller equations, solving problem (4.6a) (without modification) generates open loop control behavior. This result would be an unrealistic lower bound on performance because this emulates a feedforward-feedback control system that anticipates a future disturbance and has perfect models. However, this open-loop behavior underestimates the dynamic performance objective function of all feedback controllers and can be a very loose lower bound of control performance for feedback control.



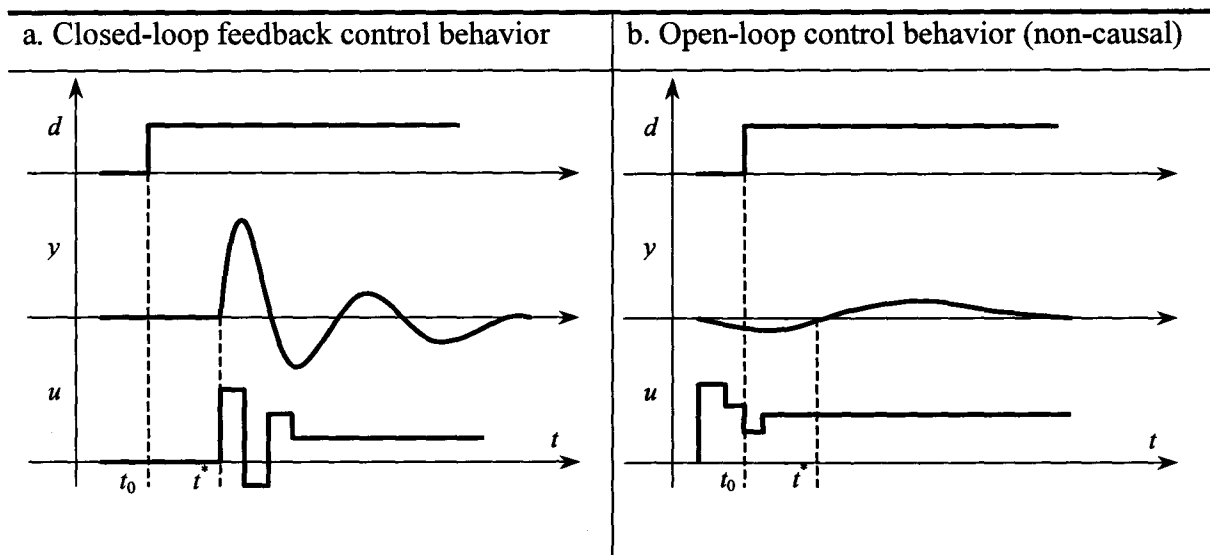


Figure 4.2 Comparison of open-loop and closed-loop feedback control behavior

In order to make the open-loop optimization observe causality, we use extra constraints (4.6b) to force changes in the manipulated variables to be zero until the effect of disturbance is apparent in controlled variables.

$$\mathbf{u}_k(t) = 0 \quad \forall t < t^* \quad (4.6b)$$

These constraints are added until  $t^*$ , where  $t^*$  is the smallest dead time among all the possible loops. With modeling in deviation variables, we force the manipulated variable to its initial value (of zero because the models are in deviation variables) until feedback action is allowed.

The controllability check (4.6) gives a bound on feedback performance, which we use as an initial check on the controllability of the process. If the problem passes the controllability check, the solution proceeds to its next stage, a modified branch and bound procedure. If not, the process design or control objectives must be changed.

### 4.3.2 Branching Strategy on Pairings

The branch and bound method belongs to the class of implicit enumeration methods. It has two parts: One part is called *branching* that is a way of covering the mixed integer feasible region by smaller feasible subregions. These subregions form a

tree structure, called *branch and bound tree*. Its *nodes* are the constructed subregions. The other part of the method is called *bounding* that is a way to calculate the upper and lower bound for the objective function within a feasible subregion. If the lower bound of any subregion in the branch and bound tree is greater than the upper bound of feasible solution to the original problem, then this subregion can be safely deleted, which is called *pruning*. The algorithm stops when all nodes in the branch and bound tree are pruned or the upper and lower bounds are equal (perhaps, within a tolerance limit).

The conventional branching method for the binary integer problem is to set each binary variable to one and zero. This general branching method is independent of problem structure and in the worst case, enumerates all possible combinations of binary variables. Figure 4.3 shows the conventional binary branch and bound tree, where the numbers are determined by branching and integer variables, the gray parts are the undetermined integers that are relaxed to continuous variables between 0 and 1 and need to be determined in the further branches.

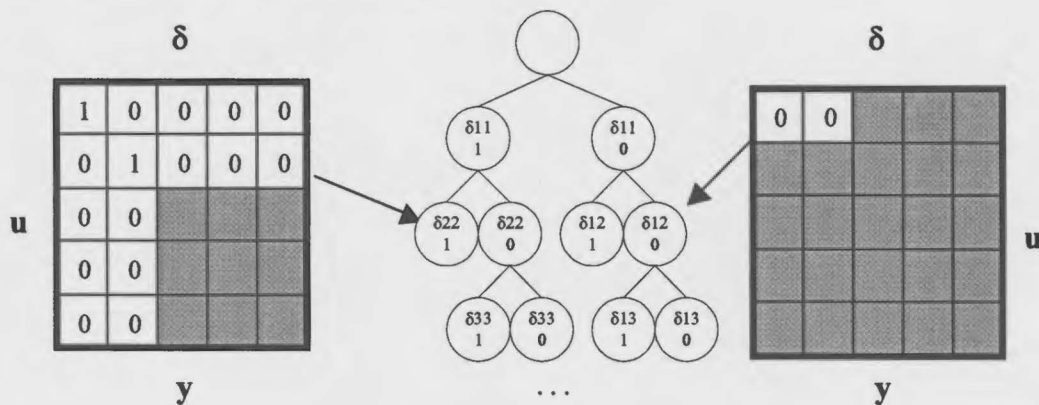


Figure 4.3 Conventional branching method

$$\delta_{ji} = \begin{cases} 1 & \text{pairing } u_j(t) \text{ with } y_i(t) \text{ is chosen} \\ 0 & \text{pairing } u_j(t) \text{ with } y_i(t) \text{ is NOT chosen} \end{cases} \quad (4.7)$$

$$\sum_i \delta_{ji} = 1, \quad \sum_j \delta_{ji} = 1 \quad (4.8)$$

The control structure selection problem uses the binary variables  $\delta$  in (4.7) as an interconnection matrix to address structural constraints in problem (4.1). Equations (4.8)

allow only one connection to be selected for each row and column, which limits the use of one measured control variable and one manipulated variable per controller (i.e., the loop pairing problem). Setting  $\delta_{ji}$  to one explicitly selects a pairing and also requires all other binary variables in the  $j^{\text{th}}$  row and  $i^{\text{th}}$  column to zeros by equations (4.8). In contrast, setting  $\delta_{ji}$  to zero does not choose a pairing explicitly and leaves a large number of pairings possible (right part in Figure 4.3). The conventional branching method does not take advantage of equations (4.8). It branches binary variable to zero and one equally. Also, the relaxed problem generated by branching the binary variable on zero doesn't have the block structure with clear physical meaning. Therefore, we conclude that a specialized branch and bound method, which enforces constraints (4.8), can reduce the computations for this problem.

In order to improve the solving efficiency for the control structure design problem, the branch and bound method is modified to take advantage of the constraints (4.8). Instead of calculating the lower bound at each node of the conventional branching, we only calculate the lower bound at a node generated by setting a binary variable to one. This modified logic can be achieved by two methods. One would add additional constraints; however, this would require the solution of many additional continuous relaxations, most of which would be infeasible. Therefore, we chose to implement extra logical conditions added to the branching at each node that tests whether the current branching variable is set to one or zero (Figure 4.4). The logical condition represents the control knowledge that guides the branch and bound procedure.

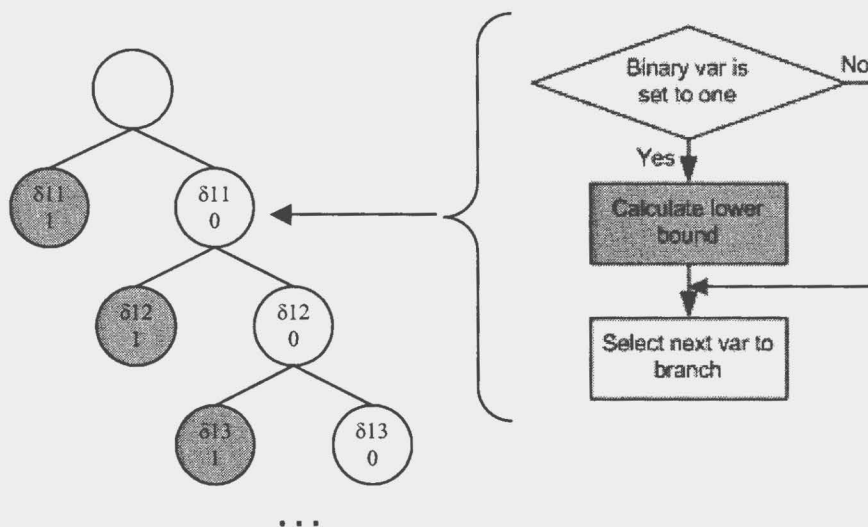


Figure 4.4 Calculating the lower bound only at the node branching on one

Unfortunately, we do not have an off-the-shelf mixed-integer solver that can incorporate this customized logic. Therefore, we coded our own branch and bound algorithm to gain full control of the search process. The code is not a general solver but specialized to handle the control structure design problem.

Instead of branching binary variables to zero, the algorithm developed and used for this research branches on loop pairings (Figure 4.5). At each node, one variable within the row of unbranched binary variables is set to one, which represents one feasible solution of  $\sum_j \delta_{ji} = 1$ . Equation  $\sum_i \delta_{ji} = 1$  is checked at each node to ensure that a paired manipulated variable is not paired again; the infeasible nodes are discarded. In this way, binary variables are only set to one and the branch and bound tree covers all possible pairings.

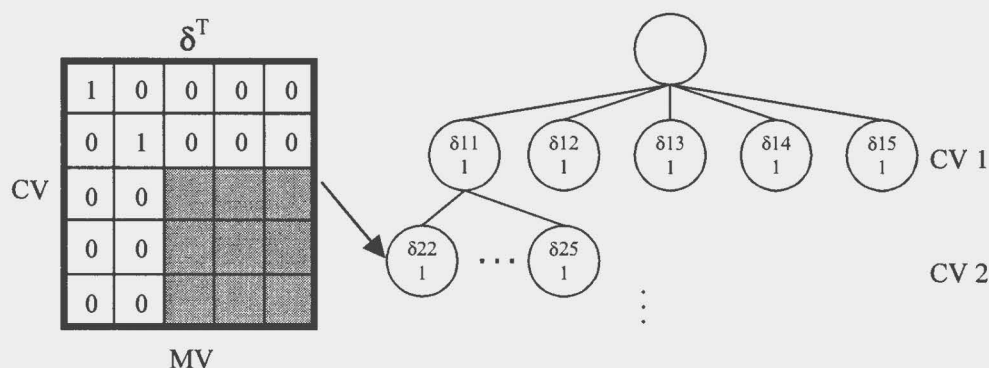


Figure 4.5 Branch on loop pairings

This branch and bound tree structure has clear physical meaning. Each level (row) of the branch and bound tree represents one controlled variable and each node (column) in the level represent a manipulated variable pairing of the controlled variable. At each node one binary variable  $\delta_{ji}$  is set to one, which chooses pairing  $i$ th controlled variable with  $j$ th manipulated variable. At each node there is a clear block structure consisting of several paired loops and one remaining block that contains all unpaired controlled and manipulated variables (shown in grey in Figure 4.5). This formulation enables us to use control concepts to calculate the lower bound efficiently, as explained in the next section.

Engineering insight and process knowledge play very important roles in the control structure design procedure. In order to utilize this information, the algorithm has three adjustable heuristic parameters to guide the search process:

- The branching sequence of the controlled variables: We influence the sequence of branching by selecting the row for each controlled variable. The controlled variable at upper levels will be branched earlier. The controlled variable that should be selected for the upper levels of the tree have the largest differences in gain and input-output dynamics (dead times, time constants and so on) among the manipulated variable candidates. Branching these controlled variables early can potentially prune the branch and bound tree at top levels, thus reducing the total number of nodes evaluated.
- The branching sequence of the manipulated variables: The sequence for evaluating manipulated variable candidates depends on the location of the

manipulated variable in the column sequence. The pairings having fast dynamics and strong effects (large steady-state process gains) should be evaluated early in sequence to find a good overall pairing quickly.

Additional shortcut metric rules can also help choose the sequence, such as first evaluating the pairing with relative gain array element close to one and small relative disturbance gain (RDG). These were not implemented in the case studies in this research.

- The search strategy, which defines the rule for selecting the next node to evaluate.
  - Depth-first search: the most recently created node is branched first.
  - Width-first search: the node at the same level is branched first.
  - Best-value search: the node with the best objective function value branch first.

A tight upper bound can potentially prune many branches at early stage. Therefore, we have chosen the depth-first search in all case studies to reach a full integer solution quickly and update the upper bound more frequently.

We do not introduce an adjustable tolerance for pruning a branch, i.e. we prune a branch if its lower bound is greater than the best upper bound within the machine precision ( $10^{-16}$ ).

Figure 4.6 shows the branch and bound procedure developed in this research.

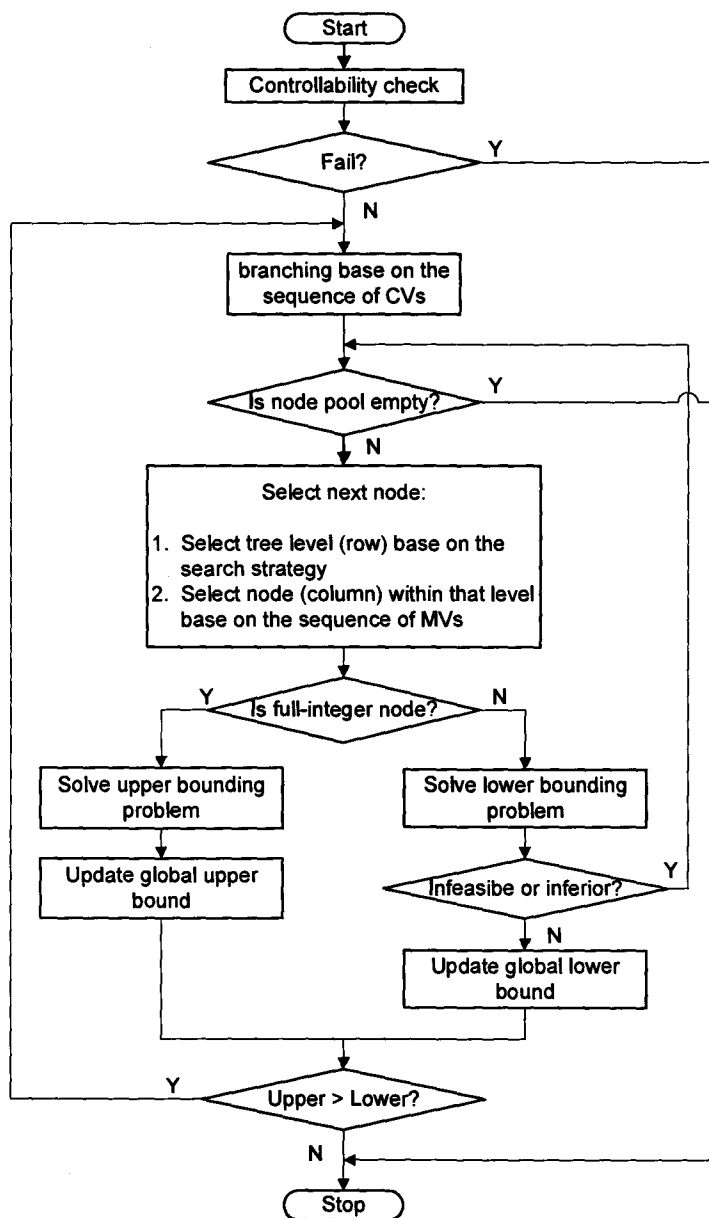


Figure 4.6 Flowchart of branching procedure

### 4.3.3 Estimating Lower Bounds at Intermediate Nodes Using Control Knowledge

The bounding procedure at each intermediate branch and bound node requires the calculation of the lower bound of the objective function, Integral Square Error (ISE) of

the dynamic transients. Here, we use the term “intermediate” to designate all nodes below the top node and above the all-integer solutions. The common way to calculate the lower bound of the objective function is to relax the undetermined binary variables to continuous variables. In our design problem, this common relaxation approach represents a situation in which all relaxed measured control variables and manipulated variables would be connected with single-loop controllers. As shown in Figure 4.7, this relaxation creates many PI controllers. The solution to this formulation would provide a valid lower bound on the dynamic performance objective function. However, all controllers must be tuned, since the performance bound requires good controller tuning. Therefore, the relaxation yields a high-dimensional, non-convex non-linear programming problem that would require excessive computation at each node. Therefore, we developed a modified relaxation approach.

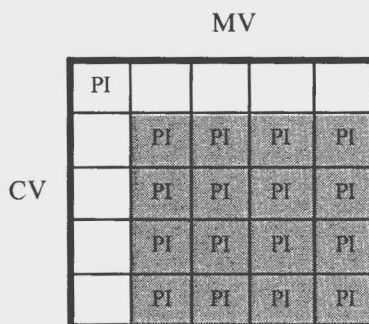


Figure 4.7 Optimal tuning problem generated from continuous relaxation of binary variables (one paired loop and four undetermined pairings).

From control theory we know that an optimal multivariable controller with perfect models and with an exact disturbance prediction provides the best possible control performance. Also, an open-loop control calculation provides the same performance as the optimal controller (with no model mismatch). Therefore, we replace all PI controllers within the unpaired block (grey area in Figure 4.7) by an open-loop trajectory optimization, as shown in Figure 4.8. We delete all controller equations in the unpaired block, which gives the optimizer the freedom to find the best possible performance by adjusting the unpaired manipulated variables without limitation by a control law. From



the mathematical point of view this procedure gives a lower bound on the performance because we relax the problem by removing control-law constraints, i.e., increasing the feasible region. (Note that the “open-loop” controller is modified to include the causality as described in Figure 4.2 by adding constraints (4.6b).

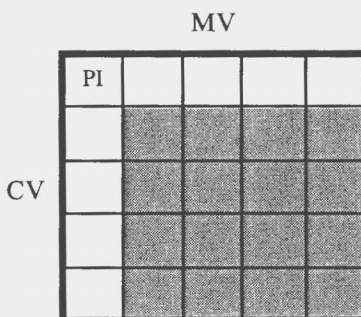


Figure 4.8 Optimal tuning problem generated from open-loop controller relaxation

The open-loop controller relaxation is very different from the common continuous relaxation. Instead of relaxing the variable bounds the open-loop controller relaxation uses a different equation system, which is supported by control knowledge and also is mathematically sound.

Problem (4.9) is solved at each intermediate node.

$$\min_{K_{C,ji}, K_{I,ji}} \sum_k \sum_{t=1}^{tf} [\mathbf{e}_k(t)^T \mathbf{Q} \mathbf{e}_k(t)]$$

s.t.

$$\begin{aligned} \mathbf{x}_k(t+1) &= \mathbf{A}_k \mathbf{x}_k(t) + \mathbf{B}_k \mathbf{u}_k(t) + \mathbf{W}_k \mathbf{d}(t) \\ \mathbf{y}_k(t) &= \mathbf{C}_k \mathbf{x}_k(t) + \mathbf{V}_k \mathbf{d}(t) + \mathbf{N}(t) \\ \mathbf{e}_k(t) &= \mathbf{sp}(t) - \mathbf{y}_k(t) \\ \mathbf{y}^L(t) &\leq \mathbf{y}_k(t) \leq \mathbf{y}^U(t) \end{aligned} \quad (4.9)$$

$$\left[ \begin{array}{c} \neg \delta_{ji} \\ u_j^L(t) \leq u_{k,j}(t) \leq u_j^U(t) \\ u_{k,j}(t) = 0 \quad \forall t \leq t^* \end{array} \right] \vee \left[ \begin{array}{c} \delta_{ji} \\ \Delta u_{k,j}(t) = K_{C,ji} [e_{k,i}(t) - e_{k,i}(t-1)] + K_{I,ji} e_{k,i}(t) \\ u_{k,j}(t) = u_{k,j}(t-1) + \Delta u_{k,j}(t) \\ u_j^L(t) \leq u_{k,j}(t) \leq u_j^U(t) \end{array} \right]$$

The controller equations in problem (4.9) are selected by the binary variables  $\delta_{ji}$ , which are determined by the branching procedure. The unpaired binary variables are all set to zero. So problem (4.9) includes the controller equations for only the paired loops where  $\delta_{ji}$  is set to one by the branching procedure, and includes the open-loop control problem (modified for causality) where  $\delta_{ji}$  is set to zero by the branching procedure. When the controller tuning parameters  $K_{C,ji}$  and  $K_{I,ji}$  for the paired loops are constant, problem (4.9) is a convex QP that can be solved efficiently. The next section describes how the loop-tuning problem is solved.

#### 4.3.4 Grid Search on Loop Tuning

In section 4.3.3, we reduced the number of the PI controllers to be tuned in the lower bounding problem (4.9) using control knowledge. In this section the solution method is further simplified by using the grid search instead of gradient-based numerical optimization to find reasonably good tunings. The reasons are:

- Because of interactions, the magnitude of the controller tuning values can be very different from conventional single-loop tuning rules.

- A wide range of the tuning parameters must be explored. Since we can allow negative RGA pairings, the controller gains can vary between different signs (positive or negative). Local numerical optimization is prone to converge to a local optimal “near” the initial estimate.
- Good (not optimal) tuning is required to achieve our goal of selecting the best controller structure. While reasonable controller tuning is essential, tuning values in the right range is good enough to distinguish bad structures from good structures. As will be explained, the controller tuning is reevaluated for every all-integer solution.

The grid is defined in the space of the tuning parameters,  $K_{Cji}$  and  $K_{Iji}$  and at each grid point problem (4.9) is solve to give an objective function value. This grid depends on the process dynamics and the controller loop pairing.

$$\begin{aligned}
 K_{C,ji} &= factor \cdot \frac{\min(\lambda_{ij}, 1)}{G_{ij}(0)} = factor \cdot \frac{\min(G_{ij}(0) \cdot [G^{-1}(0)]_{ji}, 1)}{G_{ij}(0)} \\
 &= factor \cdot \min(1/G_{ij}(0), [G^{-1}(0)]_{ji})
 \end{aligned} \tag{4.10}$$

Equation (4.10) defines the grid values for controller gains  $K_{Cji}$ , which are determined to be around a base value from the following heuristic tuning rule: the process gain times controller gain should be around a valve of 1.0. The controller gains are modified by relative gain  $\lambda_{ij}$  to take into account the loop interactions on the process gain, while not increasing the controller gain over its single-loop value. Thus, this heuristic gives an initial estimate for the controller gain to be  $K_{Cji} = \min(\lambda_{ij}, 1)/G_{ij}(0)$ . We use a term, *factor*, which is chosen from values between  $-10$  to  $10$ , to define the range over which the grid search is performed around  $K_{Cji}$ . Note that the factor extends the grid range to the opposite sign from single-loop tuning, which is required to stabilize pairings with negative relative gains. The final formulation is given in (4.10), which is valid even when relative gain is zero.

In order to limit the dimensionality of the search grid, we perform the search for tuning one loop at a time. We only search the tuning of the loop being introduced at the current node, and we retain the tuning for paired loops from previous levels, as shown

schematically in Figure 4.9. This simplification is reasonable because the controller gains for the candidates already paired (shown as “same tuning” in Figure 4.9) were tuned with all other controlled variables regulated by the optimal multivariable closed-loop controller. This approach ensures that process interactions were considered in all loop tunings. We emphasize that we are not employing one-at-a-time loop tuning with the other loops being *open*, which will certainly provide poor tuning.

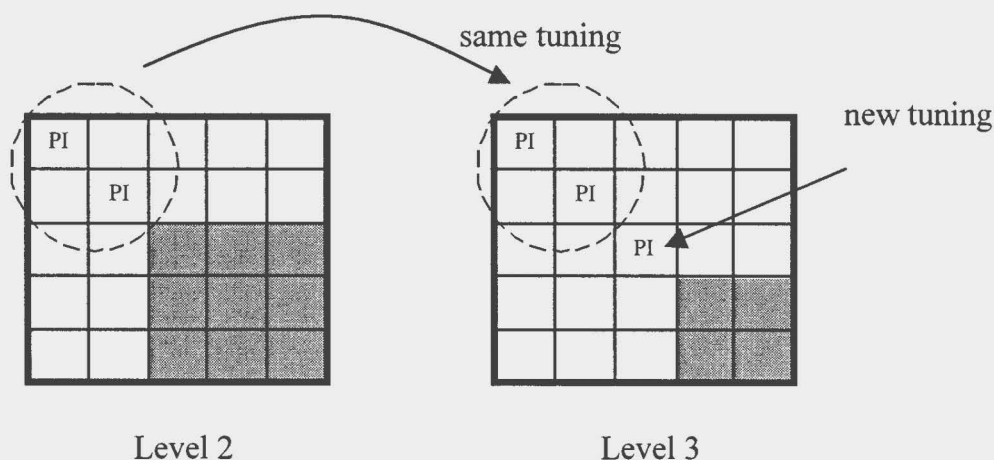


Figure 4.9 Tune one loop at a time

Thus, the method for loop tuning provides a practical approach that is computable for large control systems, i.e., those with a large number of controlled and manipulated variables. It does not ensure a global optimum for the design problem, which could only be achieved by a different optimization technology such as global optimization. (Recall that we have evaluated the performance of these methods in Section 4.2 and found them to be inadequate.)

## 4.4 Additional Criteria as Constraints

In Chapter 3 the multi-objective formulation included all additional criteria as constraints. In the solving procedure these additional criteria should be evaluated along with the transient performance evaluation at each branch and bound node, as shown in “shortcut metrics” Figure 4.10. The additional criteria are tested at each branch and

bound node as a necessary condition *before* evaluating the QP for the lower bound of the dynamic transient.

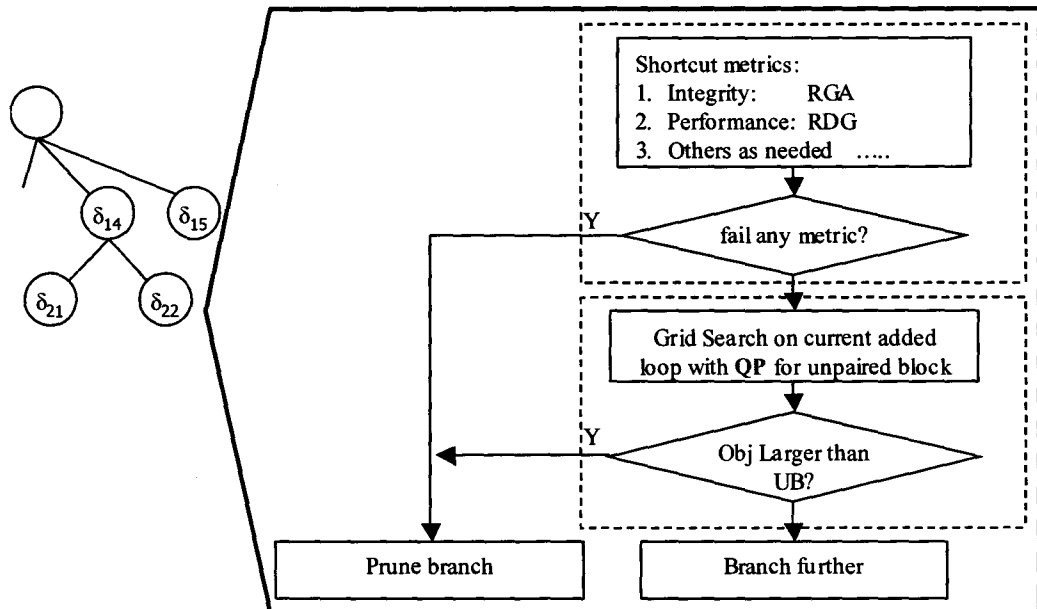


Figure 4.10 Integrate transients with shortcut metrics

Table 4.4 lists some of the additional criteria that could be integrated in the branch and bound procedure. The testing conditions is very flexible, *e.g.*, the engineer could require the control design to be paired on nonzero relative gains elements to provide good integrity. In contrast, the engineer could allow pairing on no more than one negative relative gain.

Table 4.4 Some criteria that could be integrated into the current design method\*

| Objective          | Criteria  | Condition   |
|--------------------|---|---|
| Integrity          | Relative gain (RGA)                                     | <ul style="list-style-type: none"> <li>• All RGA positive</li> <li>• No RGA equal to zero</li> <li>• No more than one RGA negative</li> </ul> |
| Performance        | Relative disturbance gain (RDG)                         | RDG less than the ISE of current best pairing   |
| Stability          | Niederlinski Index (NI)                                 | NI positive   |
| Robust performance | Robust performance number (RPN) (for non-square system) | RPN less than a given threshold   |

\* Only the relative gain is applied in the case studies in this thesis.

These additional criteria can be calculated relatively quickly when compared with optimizing the dynamic transient. When a metric indicates an unacceptable property, we can prune the branch based on these criteria; this would be the case for the relative gain. Some criteria in Table 4.3 do not represent new objectives, but rather, they are heuristics that could prevent evaluating the dynamic behavior for a node, which can potentially save considerable computing time.

## 4.5 Solving the Full-integer Candidates

At the bottom of the branch and bound tree all integer variables have been assigned. The original MINLP problem (4.1) becomes a continuous NLP problem (4.11) that defines the upper bound on dynamic performance, which includes all important control objectives, such as multiple plant models for robustness, measurement noise and actuator saturation.

$$\begin{aligned}
 & \min_{K_c, K_I, s_k^U(t), s_k^L(t)} \sum_k \sum_{t=1}^{tf} [\mathbf{e}_k(t)^T \mathbf{Q} \mathbf{e}_k(t)] \\
 & \text{s.t.} \\
 & \mathbf{x}_k(t+1) = \mathbf{A}_k \mathbf{x}_k(t) + \mathbf{B}_k \mathbf{u}_k(t) + \mathbf{W}_k \mathbf{d}(t) \\
 & \mathbf{y}_k(t) = \mathbf{C}_k \mathbf{x}_k(t) + \mathbf{V}_k \mathbf{d}(t) + \mathbf{N}(t) \\
 & \mathbf{y}^L(t) \leq \mathbf{y}_k(t) \leq \mathbf{y}^U(t) \\
 & \mathbf{e}_k(t) = \mathbf{s}\mathbf{p}(t) - \mathbf{y}_k(t) \\
 & \Delta \mathbf{u}_{k,j}(t) = K_{C,ji} [e_{k,i}(t) - e_{k,i}(t-1)] + K_{I,ji} e_{k,i}(t) \quad i, j \in \text{paired loops} \\
 & \mathbf{u}_k(t) = \mathbf{u}_k(t-1) + \Delta \mathbf{u}_k(t) - \mathbf{s}_k^U(t) + \mathbf{s}_k^L(t) \\
 & \mathbf{u}^L(t) \leq \mathbf{u}_k(t) \leq \mathbf{u}^U(t) \\
 & [u_j(t) - u_j^L] \cdot s_j^L(t) = 0 \\
 & [u_j(t) - u_j^U] \cdot s_j^U(t) = 0 \\
 & s_k^U(t) \geq 0 \\
 & s_k^L(t) \geq 0
 \end{aligned} \tag{4.11}$$

The tunings from the best solution in upper level of the branch and bound tree for the paired loops can be used as initial values for the solution of the optimal tuning.

Before solving Problem (4.11), the initial tuning value for the last paired loop is also determined by a grid search. At each grid, the NLP problem (4.11) is solved using the grid point as the initial values for tuning, and the best solution is selected as the optimum.

The optimal solution of problem (4.11) is used to update the upper bound of the control structure design problem. If the new all-integer dynamic performance has a lower ISE than the current upper bound (from other all-integer solutions), the new ISE value becomes the upper bound. Because of the branch and bound algorithm described in Section 4.3, problem (4.11) should be solved for a small number of pairing candidates.

Problem (4.11) includes actuator saturation formulated as complementarity constraints. A mathematic program with complementarity constraints (MPCC) may cause difficulties with classical NLP technologies (Fletcher *et al.*, 2002). Therefore, a special designed solving technology is needed to handle complementarity constraints explicitly. In this research we choose IPOPT-C to solve problem (4.11). IPOPT-C (Raghuathan and Biegler, 2003) is an algorithm for solving MPCC based on the interior point code IPOPT (Wachter, 2002), where the Interior Point Method is modified to treat complementarity constraints using the strategy described previously. IPOPT-C outperforms common NLP solvers, such as MINOS and CONOPT2, when solving MPCC (Baker, 2006). Our experience with IPOPT-C confirms it can solve problem (4.11) fast and reliably. However, IPOPT-C only uses local information for search and problem (4.11) is non-convex and nonlinear; therefore, global optimality is not guaranteed. A more complete discussion of MPCC and Interior Point Method is given in Appendix D.

## 4.6 Non-square Systems

To this point, the solution method has assumed that the number of controlled and manipulated variables are equal. However, the number of controlled variables does not always match the number of manipulated variables, and we term these systems “non-square”. The design goal is to find a square control sub-system within the overall non-square process, which gives the best overall control performance. Naturally, the square

control system must have a dimension less than or equal to the minimum of the number of controlled and manipulated variables. Although, the previous method is based on square systems, it can be extended to non-square systems.

For fat systems (the number of controlled variables is less than the number of manipulated variables), we can use exactly the same branch and bound procedure, which is shown in Figure 4.11.

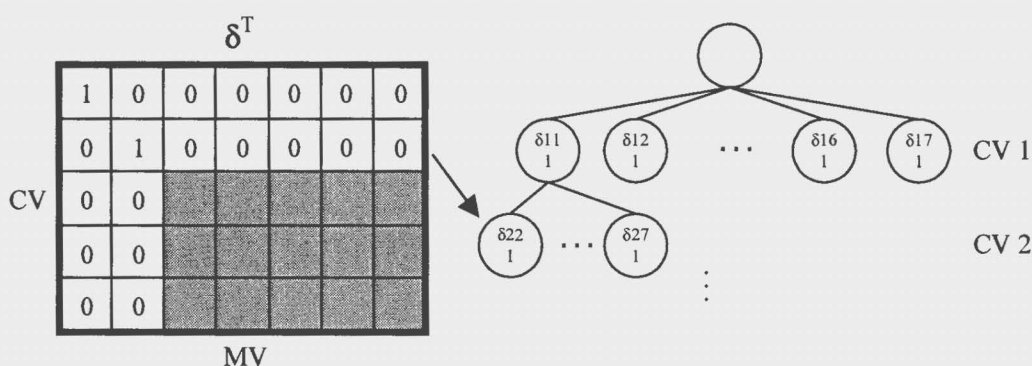


Figure 4.11 Fat system (#CV < #MV)

$$\sum_i \delta_{ji} \leq 1, \quad \sum_j \delta_{ji} = 1 \quad (4.12)$$

The only modification is the integer structural constraints (4.12). Since one or more of the manipulated variables are not used in the final control structure, the summation of each column can be zero.

When all the loops are paired, the unused manipulated variables are not adjusted, which is achieved by fixing their controller gains to zero, which is reasonable because we use linear models with deviation variables in the problem formulation. In the plant implementation, the unused manipulated variables will be kept at constant values determined from economical optimization, which is not part of this dynamic control design problem.

For slim systems (the number of controlled variables is greater than the number of manipulated variables), the branch and bound procedure needs to be modified. The meaning of the binary matrix  $\delta$  is transposed, with the rows representing manipulated



variables. The levels of branch and bound tree are defined by manipulated variables, as shown in Figure 4.12. Since there are more controlled variables than manipulated variables, we pair controlled variables to each manipulated variable instead of the opposite.

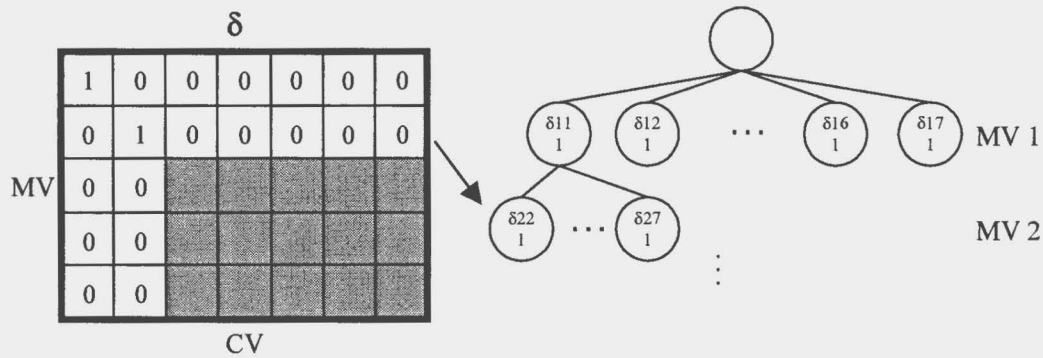


Figure 4.12 Slim system ( $\#CV > \#MV$ )

$$\sum_i \delta_{ji} = 1, \quad \sum_j \delta_{ji} \leq 1 \quad (4.13)$$

The integer structural constraints (4.13) are modified accordingly. Since some of the controlled variables are not regulated in the final control structure, the summation of each row can be zero.

Due to smaller number of manipulated variables, we cannot regulate all controlled variables to their set points. The controlled variables chosen for loop pairing are returned to their set points by the end of the simulation horizon (assuming no saturation). The unpaired controlled variables deviate from their set points at the end of the simulation horizon, i.e., at steady state. The dynamic objective function remains unchanged, which includes all the CV's.

## 4.7 Software Implementation

The implementation of this algorithm is a combination of several different modeling environments and solvers, which is shown schematically in Figure 4.13. MATLAB provides a developer-friendly environment for data analysis and visualization.

Its general programming language provides interfaces to other programs. The branch and bound strategy is programmed in MATLAB, which calls optimization solvers through MATLAB/GAMS interface (Ferris, 1998) and MATLAB/AMPL interface (developed as part of this research). These interfaces use files stored on hard disk to exchange information with MATLAB and GAMS (AMPL). The QP lower bounding problems generated at each intermediate branch and bound node are solved by CPLEX in the GAMS environment. The NLP problems with all integer variables assigned at the bottom of the branch and bound tree are solved by IPOPT-C in the AMPL environment, because IPOPT-C did not have GAMS interface at the time of this research.

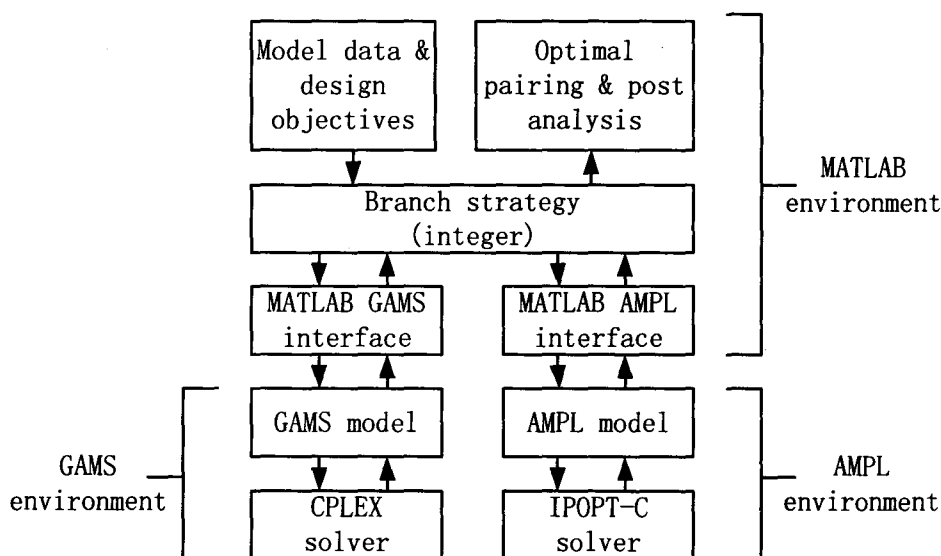


Figure 4.13 Software structure

## 4.8 Case Studies

In this section, we demonstrate the method on the FCC example presented in Chapter 3, and a medium size, 4x4 fired heater. Finally, a 2x3 system is used to demonstrate how the method handles non-square system.

### 4.8.1 Fluidized Catalytic Cracker (FCC)

This example has been introduced in Chapter 3 as a case study; please refer to Section 3.4 for the detailed description. We applied the solving method presented in this chapter to the FCC example. It took one minute to reach a solution if positive RGA is required, two minutes if no integrity constraint is added. The same result of ISE and transient as Chapter 3 case study is obtained.

If pairing on a positive RGA is required, the loop pairings we get are the riser temperature  $T_{ris}$  is controlled by the flow of air  $F_{air}$  and the regenerator temperature  $T_{rgn}$  is controlled by the flow of catalyst  $F_{cat}$ ; however, the dynamic performance is poor. The best dynamic performance is achieved when the control design allows negative RGA pairings, which are controlling the riser temperature  $T_{ris}$  by the flow of catalyst  $F_{cat}$ , and controlling the regenerator temperature  $T_{rgn}$  by the flow of air  $F_{air}$ .

This case study shows the solving method presented in this chapter returns a result that matches the grid search and physical intuition presented in Chapter 3.

### 4.8.2 Fired Heater

The fired heater is a furnace that has one firebox with four burners and four heating coils, as shown in Figure 4.14. Oil flows through the coils and is heated by radiation and convection. The coil (oil) outlet temperatures ( $T_1$ ,  $T_2$ ,  $T_3$ ,  $T_4$ ) are controlled by adjusting the fuel valves ( $V_1$ ,  $V_2$ ,  $V_3$ ,  $V_4$ ) for each of the burners; excess oxygen is assumed to be supplied by natural draft, so that sufficient air is always available. This problem and the transfer function matrix (4.14) was originally proposed in Rosenbrock (1974) without engineering unit specifications for all variables. Therefore, we treat all variables as dimensionless deviation variables in this case study. Manousiouthakis *et al.* (1986) used this process to demonstrate the synthesis of decentralized control structure using Block Relative Gain. The dynamics of each input-output is first-order, and the system has strong interactions since the manipulation of one burner affects all four coil temperatures. The disturbance considered in this case study is

the fuel composition change, with the transfer function in equation (4.15). Although we do not have detailed fuel heating value model, we can assume the fuel composition change has the same qualitative effect as fuel flow change. Since all burners use one fuel header they respond to the fuel composition change simultaneously in the same manner.

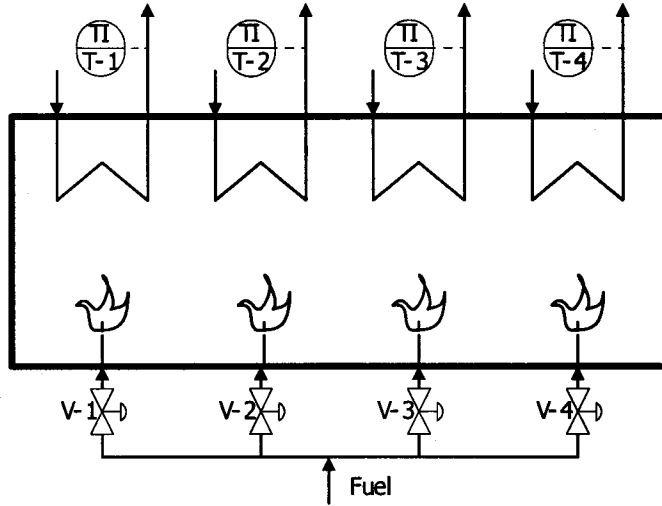


Figure 4.14 Fired Heater

$$G(s) = \begin{bmatrix} \frac{1}{1+4s} & \frac{0.7}{1+5s} & \frac{0.3}{1+5s} & \frac{0.2}{1+5s} \\ \frac{0.6}{1+5s} & \frac{1}{1+4s} & \frac{0.4}{1+5s} & \frac{0.35}{1+5s} \\ \frac{0.35}{1+5s} & \frac{0.4}{1+4s} & \frac{1}{1+5s} & \frac{0.6}{1+5s} \\ \frac{0.2}{1+5s} & \frac{0.3}{1+5s} & \frac{0.7}{1+4s} & \frac{1}{1+5s} \end{bmatrix} \quad (4.14)$$

$$\text{Fuel composition disturbance: } G_d(s) = \begin{bmatrix} \frac{1}{1+4s} \\ \frac{1}{1+4s} \\ \frac{1}{1+4s} \\ \frac{1}{1+4s} \end{bmatrix} \quad (4.15)$$

The design goal for this case study is to find the loop pairing that gives the best dynamic performance with respect to the fuel composition change, subject to constraints on the design integrity. Choosing pairings with positive relative gain (RGA) is the most widely used pairing criterion for integrity and will be evaluated in this study; therefore, we initially require pairings on positive relative gains. The relative gain in equation (4.16) shows that there are only two possible positive RGA pairings (showed in bold font) for this case study: ([T1-V1], [T2-V2], [T3-V3], [T4-V4]) and ([T4-V1], [T2-V2], [T3-V3], [T1-V4]).

$$RGA = \begin{bmatrix} \mathbf{1.748} & -0.686 & -0.096 & \mathbf{0.034} \\ -0.727 & \mathbf{1.874} & -0.092 & -0.055 \\ -0.055 & -0.092 & \mathbf{1.874} & -0.727 \\ \mathbf{0.034} & -0.096 & -0.686 & \mathbf{1.748} \end{bmatrix} \quad (4.16)$$

First, we formulate the control structure design problem with the requirement of positive RGA pairings. Figure 4.15 shows the search path for the branch and bound tree. When a negative RGA pairing is found at a node, the method prunes the branch without evaluating the dynamic performance. There are only two positive RGA pairings at level 1, which results in method pruning the branch and bound tree substantially. The strategy is depth-first, so we quickly reach the node with a star at the bottom of the branch and bound tree, which generates a tight upper bound by solving problem (4.11). The last positive RGA pairing at level 1 (V4) is pruned because its dynamic performance, which is a lower bound to any ultimate all-integer solution, is worse than the upper bound in the branch and bound method. Since there is no positive pairing left in the branch and bound tree, we have proved that the node with a star is the best pairing, which is the diagonal structure ([T1-V1], [T2-V2], [T3-V3], [T4-V4]). We find the best pairing in one minute of computing time because many branches are pruned by positive RGA requirement before evaluating the transients.

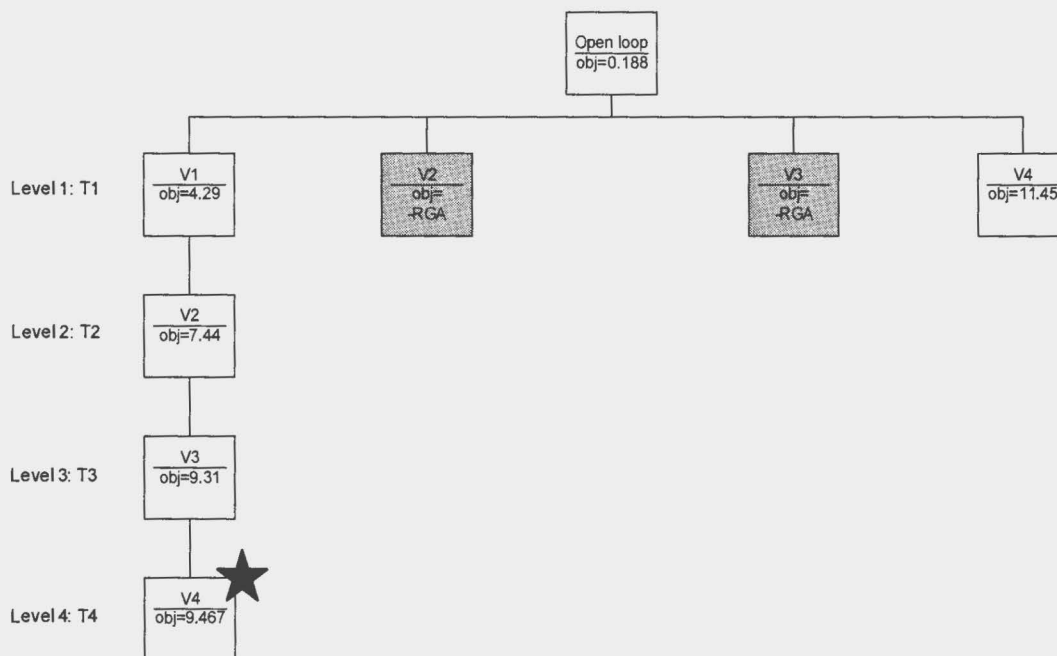


Figure 4.15 Branch and bound tree for fired heater with positive RGA

As a second case, suppose that the engineer wanted to know if a significantly better dynamic response were possible. Therefore, we design control for the same fired heater, without the constraint on the sign of the relative gain elements for the pairings. When the engineer proceeds in this manner, he/she has already evaluated the best design for the previous problem. The previous calculated best positive RGA pairing can be used as a good initial point for the performance upper bound, which is set to 9.467 rather than the default value of  $1 \times 10^{10}$ . We solve the formulation again. The algorithm goes through all nodes shown in Figure 4.16, both white nodes that were explored in the previous case and gray nodes that are newly explored in this case, and it proves the best positive RGA pairing is also the pairing gives the best dynamic performance. The solving time is three minutes. The tight upper bound generated from the good initial point helps prune branches rapidly. The best pairing is also physically sound because the temperature is controlled by the burner that is the closest to that coil.

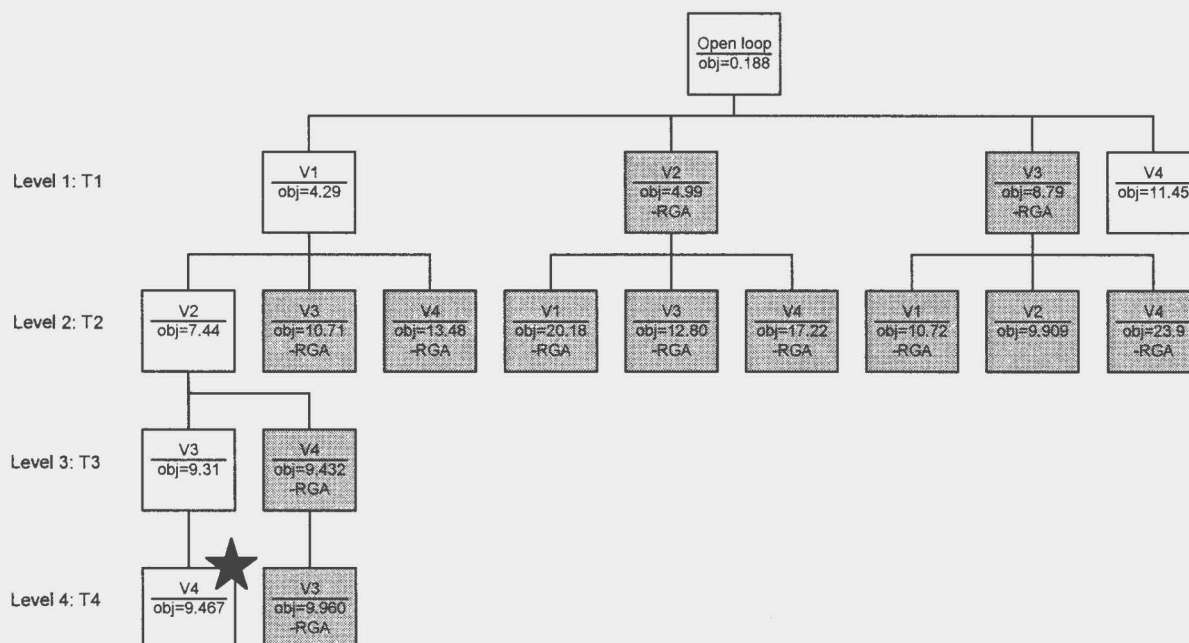


Figure 4.16 Branch and bound tree for fired heater for best dynamic performance

Figure 4.17 shows the dynamic response of the best pairing ([T1-V1], [T2-V2], [T3-V3], [T4-V4]) in response to the fuel composition disturbance.

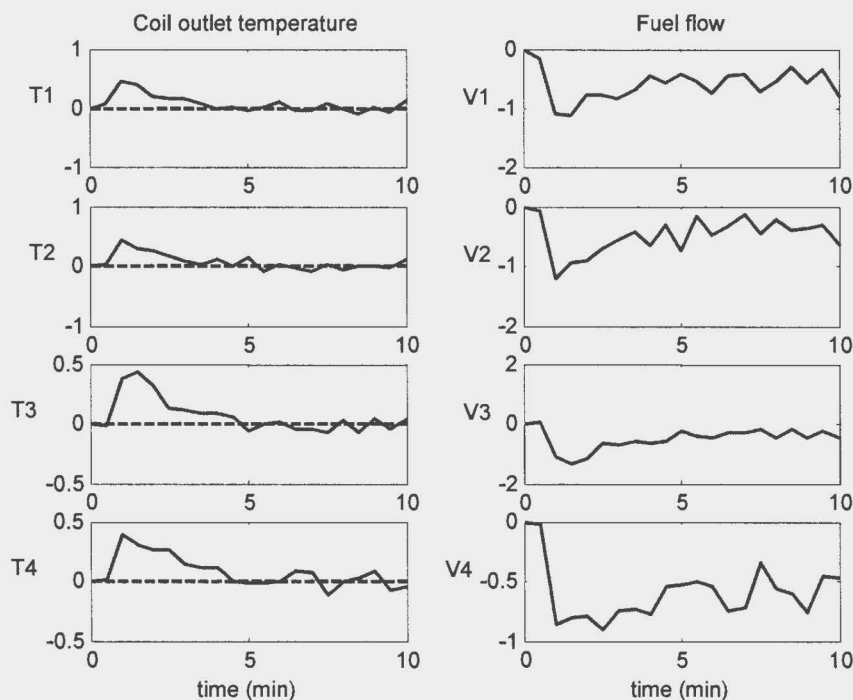


Figure 4.17 Fuel composition disturbance response for the best pairing

This case study demonstrates that:

- The tailored solving strategy can find the best control structure in relatively short computing time.
- The additional criteria can accelerate the solving process,

Again, we note that this design procedure does not *guarantee* the globally optimal controller loop pairing.

### 4.8.3 Non-square System

The method presented in this chapter can handle non-square systems. The test case is a fat system (4.17) with two controlled variables ( $y_1, y_2$ ) and three manipulated variables ( $u_1, u_2, u_3$ ). This case study does not represent any physical system and was defined as part of this research. Our goal is to design a 2x2 control system that gives the best dynamic performance with respect to the disturbance  $d$ .

The subsystem ( $y_1, y_2; u_1, u_2$ ) has one-way interaction that cannot be detected by RGA, which means a design method that used only RGA as criteria could not predict the dynamic performance accurately. Therefore, the full closed-loop dynamic transient must be taken into account as in our proposed design method. (Please refer to Appendix B for the detailed discussion of RGA properties.) The transfer function between  $y_1$  and  $u_3$  has a fast inverse response at the beginning then slowly settles to a value in the opposite direction.

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1-s}{(5s+1)^2} & 0 & \frac{0.15-s}{(100s+1)(0.1s+1)} \\ \frac{10(1+s)}{(5s+1)^2} & \frac{1-s}{(5s+1)^2} & \frac{2(1-s)}{(5s+1)^2} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \\ u_3(s) \end{bmatrix} + \begin{bmatrix} \frac{1}{0.2s+1} \\ \frac{1}{0.2s+1} \end{bmatrix} d(s) \quad (4.17)$$



The steady state gain matrix:

$$G(0) = \begin{bmatrix} 1 & 0 & 0.15 \\ 10 & 1 & 2 \end{bmatrix}$$

$$G_d(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$
(4.18)

Kookos and Perkins (2001) proposed a method for designing square control systems for non-square processes using shortcut metrics. They proposed using the following objective function for an optimization-based control design.

$$\rho \|RGA - I\|_{sum} + (1 - \rho) \|\hat{G}^{-1}G_d\|_{\infty}$$
(4.19)

where  $\|\Lambda\|_{sum} = \sum_{i,j} |\lambda_{ij}|$  and  $\|\Lambda\|_{\infty} = \max_i \sum_j |\lambda_{ij}|$ .

The objective (4.19) is evaluated at steady state.  $\|RGA - I\|_{sum}$  measures how close the relative gain is to the identity matrix.  $\hat{G}^{-1}G_d$  measures the alignment of a specific square feedback control system  $\hat{G}$  along the direction of the disturbance  $G_d$ ; a small value indicates a design that is less sensitive to the disturbance in the steady state, but the method *does not consider dynamics*. The parameter  $\rho$  is the weighting that trades off the two components of the metric. The objective (4.19) favors the pairings with relative gain close to identity matrix and low (steady-state) sensitivity to the disturbance, but it does not have direct connection to the *dynamic* performance.

Table 4.5 Shortcut metrics for non-square system based on objective (4.19) with  $\rho = 0.50$

| Pairings           | RGA  | $\hat{G}^{-1}G_d$                          | Value of (4.19) | Comments                    |
|--------------------|--|--|-----------------|-----------------------------|
| [y1-u1]<br>[y2-u2] | $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$   | $\begin{bmatrix} 1 \\ -9 \end{bmatrix}$    | 4.5             |                             |
| [y1-u3]<br>[y2-u1] | $\begin{bmatrix} -3 & 4 \\ 4 & -3 \end{bmatrix}$ | $\begin{bmatrix} -16 \\ 1.7 \end{bmatrix}$ | 16              | The worst by this objective |
| [y1-u3]<br>[y2-u2] | $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$   | $\begin{bmatrix} -6.7 \\ 1 \end{bmatrix}$  | 3.35            | The best by this objective  |

Table 4.5 lists the shortcut metrics for different control structures and predictions from the steady-state analysis. Based on the objective (4.19) (Kookos and Perkins approach) the loop pairing of  $[y1-u3]$  and  $[y2-u2]$  is the best. The loop pairing of  $[y1-u3]$  and  $[y2-u1]$  is the worst.

However, applying the non-square system extension (Section 4.6) of our control structure selection method gives a different answer because we focus on the dynamic behavior. For this example, the control selection problem is formulated with only the dynamic performance (ISE) as the objective; no constraints on the relative gain were included. Table 4.6 shows the dynamic transient for the control structures listed in Table 4.5. Based on the integral square error (ISE) of the controlled variables, the ranking of the control structures is opposite of the one based on the steady state shortcut metrics. The loop pairing of  $[y1-u3]$  and  $[y2-u1]$  has the best dynamic performance, which is judged as the worst structure by the shortcut metrics method. On the other hand, the best structure based on the steady state shortcut metrics has the worst dynamic performance.

This case study demonstrates that:

- The methodology from this research can handle non-square systems and find the structure that gives the best dynamic performance.
- The structure selection based on only steady-state metrics may have very poor dynamic performance.

Table 4.6 Disturbance dynamic transients of non-square system

| Pairing  | RGA  | Obj<br>(ISE)   | Dynamic transient |
|--|--|----------------|-------------------|
| $\begin{bmatrix} y1-u1 \\ y2-u2 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$   | 44.5           |                   |
| $\begin{bmatrix} y1-u3 \\ y2-u1 \end{bmatrix}$ | $\begin{bmatrix} -3 & 4 \\ 4 & -3 \end{bmatrix}$ | 2.04<br>(best) |                   |
| $\begin{bmatrix} y1-u3 \\ y2-u2 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$   | 325<br>(worst) |                   |

## 4.9 Summary

The original MINLP formulation for control structure design presents a challenge for common MINLP solvers. We have tested several NLP, MINLP and global solvers and found that none of them can solve a medium size control structure design problem in a reasonable time. Therefore, we developed a specialized solving method for the loop pairing design problem in this chapter.

The solving method is a specialized branch and bound algorithm that integrates with control knowledge and uses process insight to guide the searching process. The main parts are:

- The controllability check tests the feasibility of the control structure design problem with the feedback causality constraints to tighten lower bound.
- The branching strategy takes advantage of the inherent structure of the loop pairing problem. The transients are only evaluated at the nodes that branch on one.
- The efficient lower bound at the intermediate branch and bound nodes is derived from the open-loop controller concept for unpaired variables.
- Additional shortcut design metrics can be added to further reduce the number of the possible loop pairings and speed up solving process.
- The tuning grid search is implemented at intermediate branch and bound nodes to explore large tuning ranges and speed up the solving process.
- IPOPT-C is used to explicitly handle complementarity constraints introduced by actuator saturation formulation.
- The design method does not ensure global optimality.

As part of this research we coded this solver in MATLAB and called CPLEX and IPOPT-C to solve QP and NLP sub-problems, respectively. Two case studies, i.e. FCC and fired heater, demonstrate the efficiency of this method as well as the usefulness of the

additional shortcut metrics in the formulation. This method was extended to handle non-square system and was applied to the final example, the non-square loop pairing case.

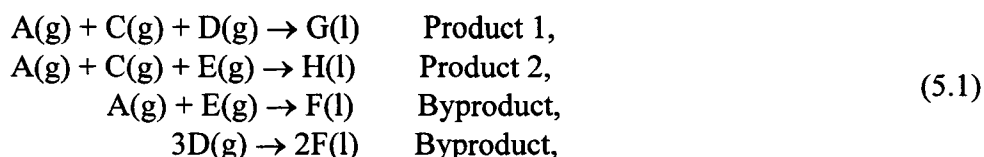
## Chapter 5

### Tennessee Eastman Problem

The Tennessee Eastman problem is a challenging plant-wide design problem first published by Downs and Vogel (1993). The problem is based on an actual industrial process and can be used to examine a number of process control topics, such as plant-wide control strategy design, multivariable control design, optimization, model identification and so on. In this chapter, the Tennessee Eastman problem is used to demonstrate the ability of the proposed control structure design formulation and solving method to handle a realistic plant-wide design problem.

#### 5.1 Process Description

The Tennessee Eastman problem involves the process in Figure 5.1 that produces two liquid products, G and H, and one byproduct F from four gaseous reactants, A, C, D and E. All reactions are irreversible and exothermic. The reactions are (g is gas and l is liquid phase)



Downs and Vogel (1993) described the chemical reaction mechanism as “The gaseous reactants are fed to the reactor where they react to form liquid products. The gas phase reactions are catalyzed by a nonvolatile catalyst dissolved in the liquid phase. ... The products leave the reactor as vapors along with the unreacted feeds.”

The only stream leaving the reactor vessel is vapor that contains both unreacted components and products. This design makes the reactor pressure, level and temperature highly correlated.

The process consists of five major units, a chemical reactor, a condenser, a vapor-liquid separator, a recycle compressor and a stripper column. The plant has four feed streams, one product stream, and one purge stream required to prevent the accumulation of one component (B) that does not appear in the product stream.

Products along with the unreacted components leave the reactor as vapor that passes through a water-cooled condenser followed by a vapor-liquid separator. The vapor stream leaves the separator and recycles back through a compressor. The liquid stream leaves the separator and is purified by a product stripper, which recovers some unreacted feed components. The product stream leaves from the bottom of the stripper column. The process system has 41 measurements and 12 manipulated variables that are defined Table 5.1 and Table 5.2.

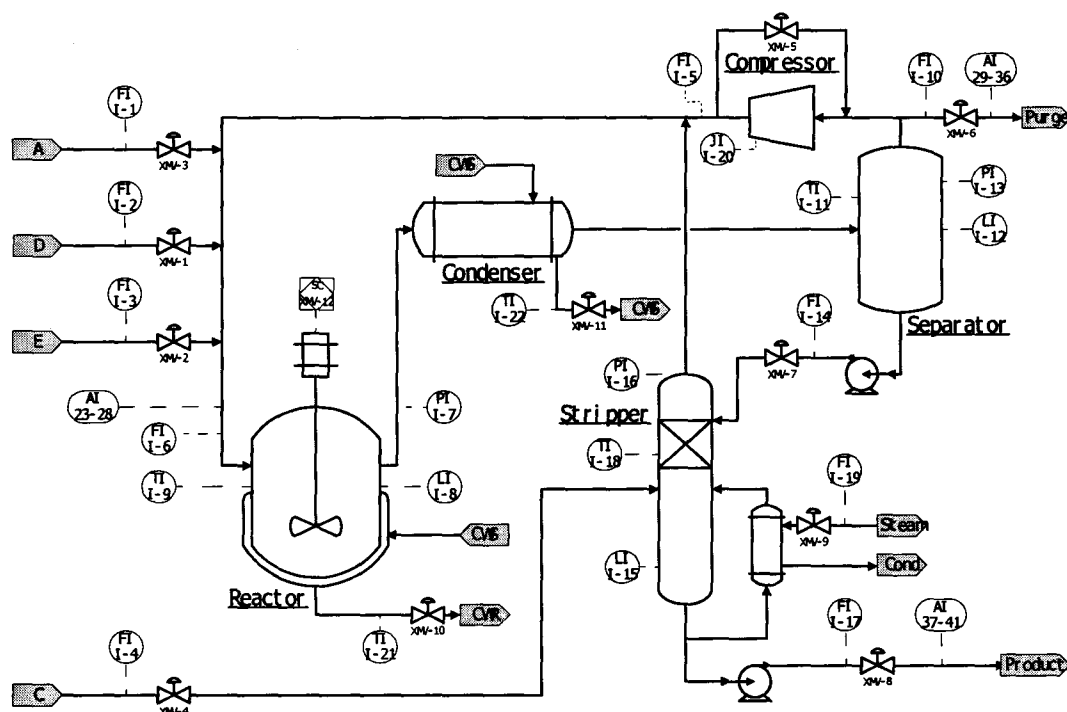


Figure 5.1 Flowsheet of Tennessee Eastman problem

Table 5.1 List of manipulated variables

| Variable name                          | Variable | Units                      |
|--|----------|----------------------------|
| D feed flow valve                      | MV(1)    | $\text{kg h}^{-1}$         |
| E feed flow valve                      | MV(2)    | $\text{kg h}^{-1}$         |
| A feed flow valve                      | MV(3)    | $\text{kscmh}$             |
| C feed flow valve                      | MV(4)    | $\text{kscmh}$             |
| Compressor recycle valve               | MV(5)    | %                          |
| Purge value                            | MV(6)    | %                          |
| Separator bottom liquid flow           | MV(7)    | $\text{m}^3 \text{h}^{-1}$ |
| Stripper bottom liquid flow (products) | MV(8)    | $\text{m}^3 \text{h}^{-1}$ |
| Stripper stream valve                  | MV(9)    | %                          |
| Reactor cooling water flow             | MV(10)   | $\text{m}^3 \text{h}^{-1}$ |
| Condenser cooling water flow           | MV(11)   | $\text{m}^3 \text{h}^{-1}$ |
| Agitator speed                         | MV(12)   | rpm                        |



Table 5.2 List of process measurements

| Variable name                              | Variable      | Units                          |
|--|---------------|--------------------------------|
| A feed flow                                | PV(1)         | kscmh                          |
| D feed flow                                | PV(2)         | kg <sup>h</sup> <sup>-1</sup>  |
| E feed flow                                | PV(3)         | kg <sup>h</sup> <sup>-1</sup>  |
| C feed flow                                | PV(4)         | kscmh                          |
| Recycle flow                               | PV(5)         | kscmh                          |
| Reactor feed flow                          | PV(6)         | kscmh                          |
| Reactor pressure                           | PV(7)         | kPa gauge                      |
| Reactor level                              | PV(8)         | %                              |
| Reactor temperature                        | PV(9)         | °C                             |
| Purge Rate                                 | PV(10)        | kscmh                          |
| Product separator temperature              | PV(11)        | °C                             |
| Product separator level                    | PV(12)        | %                              |
| Product separator pressure                 | PV(13)        | kPa gauge                      |
| Product separator underflow                | PV(14)        | m <sup>3</sup> h <sup>-1</sup> |
| Stripper level                             | PV(15)        | %                              |
| Stripper pressure                          | PV(16)        | kPa gauge                      |
| Stripper underflow                         | PV(17)        | m <sup>3</sup> h <sup>-1</sup> |
| Stripper temperature                       | PV(18)        | °C                             |
| Stripper steam flow                        | PV(19)        | kg <sup>h</sup> <sup>-1</sup>  |
| Compressor work                            | PV(20)        | kW                             |
| Reactor cooling water outlet temperature   | PV(21)        | °C                             |
| Condenser cooling water outlet temperature | PV(22)        | °C                             |
| Reactor feed analysis component A~F        | PV(23)~PV(28) | mol%                           |
| Purge gas analysis component A~H           | PV(29)~PV(36) | mol%                           |
| Product analysis component D~H             | PV(37)~PV(41) | mol%                           |

Six different production modes for the Tennessee Eastman problem are given in Table 5.3. We choose Mode 1, the base case, in our case study.

Table 5.3 List of production modes

| Mode          | G/H mass ratio | Production rate              |
|---------------|----------------|------------------------------|
| 1 (base case) | 50/50          | 7038 kg/h G and 7038 kg/h H  |
| 2             | 10/90          | 1408 kg/h G and 12669 kg/h H |
| 3             | 90/10          | 10000 kg/h G and 1111 kg/h H |
| 4             | 50/50          | Maximum production rate      |
| 5             | 10/90          | Maximum production rate      |
| 6             | 90/10          | Maximum production rate      |

Downs and Vogel (1993) listed 20 different process disturbances that can be used to study the process behaviors. We choose DV(1), A/C composition step change in C feed, as the main disturbance in our case study.

Table 5.4 List of process disturbances

| Disturbance variable | Process variable   | Type       |
|----------------------|--|------------|
| DV(1)                | In C feed, A/C composition ratio, B composition constant | Step       |
| DV(2)                | In C feed, B composition, A/C composition ratio constant | Step       |
| DV(3)                | D feed temperature                                       | Step       |
| DV(4)                | Reactor cooling water inlet temperature                  | Step       |
| DV(5)                | Condenser cooling water inlet temperature                | Step       |
| DV(6)                | A feed loss  | Step       |
| DV(7)                | C feed header pressure loss, reduced availability        | Step       |
| DV(8)                | In C feed, B and C composition                           | Random     |
| DV(9)                | D feed temperature                                       | Random     |
| DV(10)               | C feed temperature                                       | Random     |
| DV(11)               | Reactor cooling water inlet temperature                  | Random     |
| DV(12)               | Condenser cooling water inlet temperature                | Random     |
| DV(13)               | Reactor kinetics   | Slow drift |
| DV(14)               | Reactor cooling water valve                              | Sticking   |
| DV(15)               | Condenser cooling water valve                            | Sticking   |
| DV(16)               | Unknown  | Unknown    |
| DV(17)               | Unknown  | Unknown    |
| DV(18)               | Unknown  | Unknown    |
| DV(19)               | Unknown  | Unknown    |
| DV(20)               | Unknown  | Unknown    |

Downs and Vogel (1993) provided the nonlinear process simulation model coded in FORTRAN. The nonlinear state space model is in the following form:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{d})\end{aligned}\tag{5.2}$$

where  $\mathbf{x}$  is 50×1 state vector,  $\mathbf{u}$  is 12×1 manipulated variable vector,  $\mathbf{d}$  is 20×1 disturbance vector, and  $\mathbf{y}$  is 41×1 measurement vector. In this research, we used the code provided by Ricker (1996), who converted the FORTRAN code into MATLAB MEX function such that it can be called directly from MATLAB. The nonlinear model serves

as the “real” process in the chapter. We use it to validate the dynamic performance of the selected control structures and tunings.

Our proposed control structure design method needs a linear state space model of the following form:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} + \mathbf{Wd} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du} + \mathbf{Vd}\end{aligned}\tag{5.3}$$

The linear state space model of Tennessee Eastman problem is obtained by numerically differentiating the nonlinear process model. Downs and Vogel (1993) provided the steady-state values for all variables in the base case. We found the steady-state values of the state variables reported by Downs and Vogel can be further improved; this is achieved by setting the derivatives to zero, setting the values of  $\mathbf{u}$ , and solving the nonlinear algebraic equations using MATLAB function *fsolve()*. Then, we perturb one-at-a-time the elements of state variables  $\mathbf{x}$  and manipulated variables  $\mathbf{u}$  at the magnitude of  $10^{-6}$ . The simulation model does not provide handles to adjust disturbance magnitude continuously; instead, we can only switch on or off of certain disturbance. Therefore, the linearized model for disturbance is not very accurate. The linearized model is used in Section 5.6 to select the safety loops.

## 5.2 Published Designs

There have been many different control structure designs published for the Tennessee Eastman problem. The structures differ greatly in how the production rate is controlled, and how the liquid level and pressure are regulated. Most published structures control the separator and stripper levels by manipulating the liquid streams leaving the vessels. Buckley (1974) called this “*material balance controlled in the direction of flow*”. For example, McAvoy and co-workers (McAvoy and Ye, 1994; McAvoy *et al.*, 1995) proposed a control structure that

- Control all the flow valves with the flowrate measurements on the same stream. The flowrate setpoints can be used as manipulated variables.
- Control the production rate by manipulating the feed C.

- Control the pressure by the reactor temperature setpoint.
- Control the reactor level by feed E.

Another approach was developed by Ricker (1996), who developed a decentralized control structure, which demonstrated better performance than more complex nonlinear model predictive controllers he presented in previous papers (Ricker and Lee, 1995a, 1995b). The features of the decentralized control structure are

- The production rate is controlled by all feed streams through ratio controllers.
- The pressure is controlled by the small purge flow rate.
- The reactor level is controlled by the temperature setpoint of the separator that manipulates the condenser cooling water flow.

In contrast, Luyben (1996) chose to control the separator and stripper liquid levels by streams going into the vessels, which is called “*material balance controlled in direction opposite to flow*” by Buckley (1974). The control structure features

- The production rate is set directly by the stripper bottom liquid flowrate.
- The reactor pressure is controlled by the feed C that has largest flowrate in all feeds.
- The reactor level is controlled by feed D and E. Feed D and E are kept in ratio that is determined by product G and H ratio.

Although these control structures are very different, they are all organized in hierarchical structure using cascade control. The lower level controllers compensate fast and frequent disturbances. The upper level controllers handle material balance and quality.

There is no absolutely superior choice from these control structures because they are designed with different control objectives in mind. McAvoy *et al.* (1995) focused on reducing the variability in the feed streams. Ricker (1996) attempted to satisfy the economic objectives. Luyben (1996) emphasized the tight control of the production rate. So the first step of our design procedure is to clarify the control objectives, which is done in the next section.

### 5.3 Define Control Objectives

Downs and Vogel (1993) described many different control objectives for the Tennessee Eastman problem. The control objectives that we emphasize in this research are listed in Table 5.5.

Table 5.5 Control Objectives

| Category                                | Objective  |
|---|--|
| Safety                                  | The unstable levels, i.e. separator level and stripper level, must be kept in the safe ranges  |
|   | Reactor pressure is strictly below high limit  |
| Production rate and product quality     | Production rate should be tightly controlled   |
|   | Product G/H ratio is kept at (near) its specified value  |
| Smooth operation                        | Maintain process variables at desired values with respect to the disturbances  |
| Integral Controllability with integrity | The feedback system remains stable with respect to the gains of any combination of controllers reduce to zero, effectively turning these controllers on manual. This objective is enforced in some cases, and not in others. |
| Expected disturbance                    | Step change of A/C composition ratio in C feed   |

The production rate should have very low variability because it is fed directly to a downstream process that is sensitive to feed fluctuations (Downs and Vogel, 1993). Therefore, our regulatory design pairings are selected to maintain the desired production rate very close to its set point. We recognize that this design requires manipulating feed rates to achieve the desired production; however, moderate fluctuation of feed rates is allowed for this process (Downs and Vogel, 1993).

## 5.4 Selection of Controlled Variables

The Tennessee Eastman problem has more measurements than manipulated variables. It is neither possible nor necessary to control all the measurements. We select controlled variables based on the control objectives we choose earlier. The high priority safety controlled variables are selected first, followed by the controlled variables for product quality, smooth operation and disturbance rejection. The selection procedure is based on the control objectives and process insight.

Integrating liquid levels are open-loop unstable; therefore, they must be controlled for safety. There are three liquid levels in the Tennessee Eastman problem, which are reactor level PV(8), separator level PV(12) and stripper level PV(15). McAvoy (1998) analyzed the Tennessee Eastman model and found the reactor level is non-integrating, i.e., it is open-loop stable. Therefore, there are only two levels, separator level PV(12) and stripper level PV(15) that have to be controlled for safety reasons. Another important safety variable is reactor pressure PV(7), which is operating very close to its high limit. Any constraint violation will cause a process shutdown through a safety interlock system (not included in this simulation).

The material balance is essential for the successful operation of the chemical plant. In addition, the process should be operated so that each individual component is self-regulatory (and stable). Ricker (1996) gave a detailed analysis for component inventories, which included the following observations

- The concentration of inert B is self-regulating, but without control, accumulation of inert B in the vapor phase causes the pressure to increase rapidly and violate the pressure constraint.
- Byproduct F is self-regulating as long as the reactor temperature is 120~130 °C.
- Liquid products G and H are self-regulating through the liquid level control loops.

- Reactants D and E are used to control the products G and H ratio. In the base case, the reactants A and C are in excess, therefore, reactants D and E are self-regulating.
- By stoichiometry, reactants A and C are consumed in the same molar rates in the reactions that produce the products G and H no matter the product G/H ratio. So the reactant A/C ratio should be maintained at the required value, which is nearly 1:1 but is affected by the side reaction of A with E to form F.

Based on this analysis, we choose inert B in the purge flow and reactant A/C ratio in the reactor feed flow to regulate the component inventories.

In order to achieve smooth operation and disturbance rejection, the reactor level and the separator temperature need to be regulated. The reactor level is part of inventory control, and closely coupled with reactor pressure, reactor temperature and reaction rate. The separator temperature controls the separation, which affects the impurity in the product (Luyben, 1996).

The nine selected controlled variables are summarized as the following:

Table 5.6 Controlled variables

| Control Objective   | Controlled variable    | Measurement   | Variable name                       |
|---------------------|------------------------|---------------|-------------------------------------|
| Safety              | $L_{\text{separator}}$ | PV(12)        | Separator level                     |
|                     | $L_{\text{stripper}}$  | PV(15)        | Stripper level                      |
|                     | $P_{\text{reactor}}$   | PV(7)         | Reactor pressure                    |
| Product quality     | $F_{\text{product}}$   | PV(17)        | Production rate                     |
|                     | $X_{G/H}$              | PV(40)/PV(41) | Product G/H ratio                   |
| Smooth operation    | $L_{\text{reactor}}$   | PV(8)         | Reactor level                       |
|                     | $T_{\text{separator}}$ | PV(11)        | Separator temperature               |
| Component inventory | $X_{A/C}$              | PV(23)/PV(25) | Component A/C ratio in reactor feed |
|                     | $X_B$                  | PV(30)        | Component B in purge flow           |

## 5.5 Selection of Manipulated Variables

There are 12 manipulated variables in Tennessee Eastman problem. The agitator speed MV(12), which changes the heat transfer coefficient, has a very similar effect as

the reactor cooling water outlet temperature setpoint (McAvoy and Ye, 1994), therefore, we fix the agitator speed at its maximum value. Based on optimal steady-state study done by Ricker (1995), the compressor recycle valve MV(5) is fixed at its minimum value to maximize the conversion in the base case. Ricker (1995) also concluded that the stripper steam valve MV(9) should be set to zero to minimize energy consumption. After this analysis, there are nine flow valves remaining free to manipulate.

The flow sheet (Figure 5.1) shows that there is a measurement close to each control valve. By closing these flow and temperature loops 7 out of 20 disturbances listed by Downs and Vogel (1993) can be rejected (McAvoy and Ye, 1994). We concur with McAvoy, who follows engineering practice by placing flow controllers on all streams and inner temperature secondary loops for heat exchangers; advantages and design criteria for cascade control are explained by Marlin (2000). Table 5.7 lists seven flow loops and two temperature loops that we close. The PI controller tunings for the loops listed in Table 5.7 use the values from McAvoy and Ye (1994). The setpoints of these nine inner loops will be the manipulated variables for higher-level controllers. The following discussion is based on the system with these nine inner loops closed.

Table 5.7 Inner loops

| Measurement | Manipulated variable | Setpoint variable      | Variable Name                                       |
|-------------|----------------------|------------------------|---|
| PV(2)       | MV(1)                | $F_D$                  | D feed flow setpoint                                |
| PV(3)       | MV(2)                | $F_E$                  | E feed flow setpoint                                |
| PV(1)       | MV(3)                | $F_A$                  | A feed flow setpoint                                |
| PV(4)       | MV(4)                | $F_C$                  | C feed flow setpoint                                |
| PV(10)      | MV(6)                | $F_{\text{purge}}$     | Purge flow setpoint                                 |
| PV(14)      | MV(7)                | $F_{\text{separator}}$ | Separator flow setpoint                             |
| PV(17)      | MV(8)                | $F_{\text{product}}$   | Product flow setpoint                               |
| PV(21)      | MV(10)               | $T_{\text{reaCW}}$     | Reactor cooling water outlet temperature setpoint   |
| PV(22)      | MV(11)               | $T_{\text{condCW}}$    | Condenser cooling water outlet temperature setpoint |

McAvoy and Ye (1994) did a simplified overall material balance that revealed

- The product G/H ratio directly relates to D/E feed ratio



- The reactant A/C ratio should keep constant about 50/50 no matter what product mode is.

Therefore, manipulating all feeds individually actually introduces disturbances in the material balances. We should use feed ratios instead of individual feed flows as manipulated variables, which again follows industry conventions. We choose to manipulate feed by adjusting D/E feed ratio, A/C feed ratio, E/C feed ratio and C feed. The all manipulated variables we choose are listed in Table 5.8.

Table 5.8 Manipulated variables

| Manipulated variable   | Variable name                                |
|------------------------|--|
| $F_{D/E}$              | D/E feed ratio                               |
| $F_{A/C}$              | A/C feed ratio                               |
| $F_{E/C}$              | E/C feed ratio                               |
| $F_C$                  | C feed flowrate setpoint                     |
| $F_{\text{purge}}$     | Purge flowrate setpoint                      |
| $F_{\text{separator}}$ | Separator pot flowrate setpoint              |
| $F_{\text{product}}$   | Stripper product flowrate setpoint           |
| $T_{\text{reaCW}}$     | Reactor cooling water temperature setpoint   |
| $T_{\text{condCW}}$    | Condenser cooling water temperature setpoint |

## 5.6 Selection of Safety Loops

The safety loops include unstable levels and variables that must be tightly controlled to avoid equipment damage and plant shutdown. In this case study, separator level, stripper level and reactor pressure must be controlled for safety.

The separator level and striper level reflect material inventory; therefore, they are also part of material balance control. Many published control structures use one or all feed flow rates to adjust the production rate. Due to long resident time the tight production rate control is difficult to achieve. We emphasize the tight control on the production rate in this study. So the production rate  $F_{\text{product}}$  is directly set by the stripper product flowrate controller. Any production rate adjustment will be propagated through the system in the direction opposite to the flow until it reaches the feed flow rate.

Dictated by be this material balance strategy, the separator level and stripper level are controlled by upper streams that flow into the vessels, i.e., the separator level is controlled by condenser cooling water temperature setpoint, and the stripper level is controlled by the separator feed flowrate setpoint.

The reactor pressure is influenced by all the vapor flow handles, i.e.  $F_{D/E}$ ,  $F_{A/C}$ ,  $F_{E/C}$ ,  $F_C$  and  $F_{\text{purge}}$ .  $F_{D/E}$  is reserved for controlling product G/H ratio because material balance relations (McAvoy and Ye, 1994). There are four candidates left to control the reactor pressure. The selection criterion is the integrity among these three safety loops, which should regulate properly no matter what happens to other loops. This can be tested by Block Relative Gain (BRG). Positive determinant of the BRG is a necessary condition for integrity (Grosdidier and Morari, 1987). So we should choose a reactor pressure control loop that makes the determinant of BRG for three safety loops positive. Table 5.9 lists determinant of BRG for three safety loops with different reactor pressure pairings.

Table 5.9 BRG for different reactor pressure pairings

|                                 | Pairing 1 | Pairing 2 | Pairing 3 | Pairing 4          |
|---------------------------------|-----------|-----------|-----------|--------------------|
| $P_{\text{reactor}}$            | $F_{A/C}$ | $F_{E/C}$ | $F_C$     | $F_{\text{purge}}$ |
| $\det(\text{BRG}_{3 \times 3})$ | -0.7      | -0.026    | 2.2       | -0.082             |

Based on the BRG criteria, we choose C feed flowrate setpoint to control the reactor pressure (Table 5.10). The safety loops have the same sign for their controller gains no matter the automatic/manual status of the other loops. Table 5.10 also lists the PI controller tunings, i.e. proportional gain  $K_C$  and integral time  $T_I$ , for the safety loops.

Table 5.10 Safety loops

| Controlled variable    | Manipulated variable   | $K_C$ | $T_I$ (minute) |
|------------------------|------------------------|-------|----------------|
| $P_{\text{reactor}}$   | $F_C$                  | 0.01  | 3.3            |
| $L_{\text{separator}}$ | $T_{\text{condCW}}$    | -0.5  | 16.6           |
| $L_{\text{stripper}}$  | $F_{\text{separator}}$ | 0.5   | 16.6           |

## 5.7 Selection of Remaining Loop Pairings

Closing the production rate loop and the three safety loops leaves us with a 5x5 system (Table 5.11). We consider the dynamic response with respect to the disturbance DV(1), i.e. A/C ratio step change in feed C and B component constant. The control structure design problem is formulated for this 5x5 system as described in Chapter 3 and solved using the methodology presented in Chapter 4.

Table 5.11 Design 5x5 system

| Controlled variable    | Manipulated variable |
|------------------------|----------------------|
| $X_{G/H}$              | $F_{D/E}$            |
| $L_{\text{reactor}}$   | $F_{E/C}$            |
| $X_{A/C}$              | $F_{A/C}$            |
| $X_B$                  | $F_{\text{purge}}$   |
| $T_{\text{separator}}$ | $T_{\text{reacCW}}$  |

The nominal model used in the control structure design problem is the linearized process model presented in Section 5.1 combined with base-layer PI control loops, i.e. the inner loops listed in Table 5.7 and three safety loops listed in Table 5.10. There is one mismatched model considered in the control structure design problem. Downs and Vogel (1993) did not provide model uncertainty information; therefore, we make up the mismatched model as the nominal model with ten seconds time delay added into each controlled variable response. Recall that the main purpose of the mismatched model is to moderate the optimal controller tunings so that the feedback action is not too aggressive.

Figure 5.2 shows the three main parts of the solving strategy, which will be discussed in more detail in the following sections.

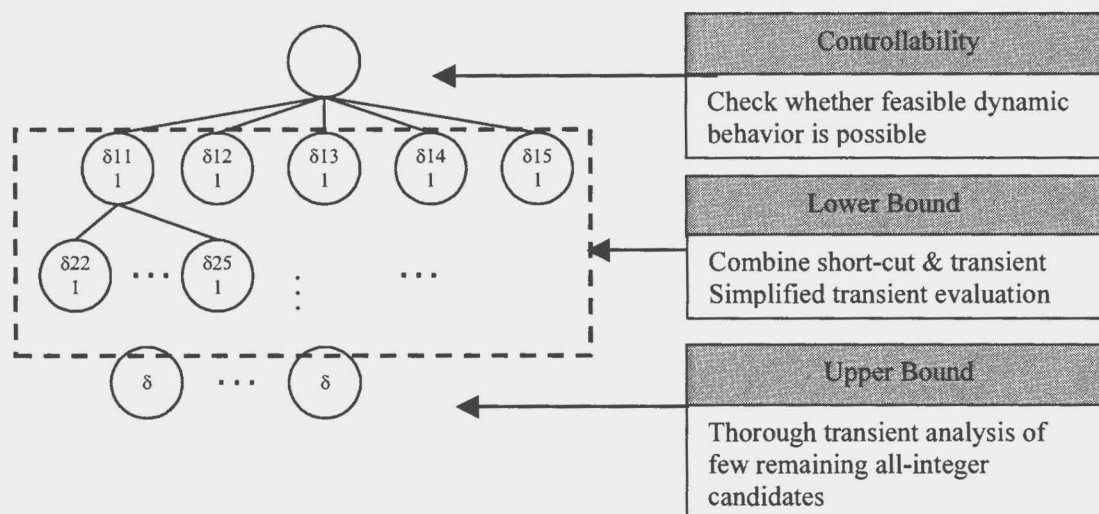


Figure 5.2 Branch and bound solving strategy

### 5.7.1 Controllability Check

The first part checks the controllability of the formulation, which determines whether a feasible solution exists for the problem with the most general feedback control. The problem is simplified by removing all controller equations and structural (loop pairing) constraints. We test whether it is possible for the plant to achieve the performance specification under the best possible control, i.e. optimal open-loop control.

The variable constraints for controlled variables and manipulated variables are listed in Table 5.12.

Table 5.12 Variable constraints for controllability check

|    | Variable Name                  | Low limit | Steady state value | High limit |
|----|--------------------------------|-----------|--------------------|------------|
| CV | $X_{G/H}$ (mass ratio %)       | 45        | 50                 | 55         |
|    | $L_{\text{reactor}}$ (%)       | 65        | 75                 | 85         |
|    | $X_{A/C}$ (mole ratio)         | 0.5       | 1.22               | 2          |
|    | $X_B$ (mol%)                   | 0         | 13.8               | 15         |
|    | $T_{\text{separator}}$ (°C)    | 75        | 80.1               | 85         |
| MV | $F_{D/E}$ (kg $h^{-1}$ ratio)  | 0.7       | 0.813              | 1          |
|    | $F_{E/C}$ (kg $h^{-1}$ /kscmh) | 400       | 477                | 600        |
|    | $F_{A/C}$ (kscmh ratio)        | 0         | 0.026              | 0.1        |
|    | $F_{\text{purge}}$ (kscmh)     | 0         | 0.337              | 0.85       |
|    | $T_{\text{reacCW}}$ (°C)       | 60        | 94.6               | 100        |

The optimization problem has a feasible solution; therefore, the process system is controllable. In addition, we observe that all controlled variables are closely controlled near their setpoints, and the settling time is very short, which means that multiloop control might exist that satisfies all performance specifications.

### 5.7.2 Heuristic Search Parameters

There are three heuristic parameters that guide the branch and bound search, which can have a big impact on the screening efficiency. These parameters are the branching sequence of the controlled variables, the branching sequence of the manipulated variables and the search strategy. Please refer to Section 4.3.2 for detailed discussion on how to choose these heuristic parameters. We use process knowledge to determine these parameters using the following guidelines.

The branching sequence of the controlled variables determines which controlled variable should be paired first. We would look at differences in the gain and dynamics of different pairings when applying this heuristic. The controlled variables that have very dynamic behavior among possible pairings should be paired first. For example, the product G/H ratio only has strong relation with feed D/E ratio based on material balance analysis. Other manipulated variables have very little direct impact on the product G/H ratio. So we choose to pair the product G/H ratio first and expect that many pairings will

give poor performances except the pairing using feed D/E ratio. By this heuristic, we can potentially reduce the branch and bound tree size efficiently at the very top level.

Table 5.13 The branching sequence of the controlled variables

| Controlled variable  | Branching sequence |
|----------------------|--------------------|
| $X_{G/H}$            | 1                  |
| $L_{\text{reactor}}$ | 2                  |
| $X_{A/C}$            | 3                  |
| $X_B$                | 4                  |
| $T_{\text{sep}}$     | 5                  |

The branching sequence of the manipulated variables is used to determine which pairing among all possible pairings we should evaluate first. Again, we would look at differences in the gain and dynamics of different pairings when applying this heuristic. The pairing that has bigger gain and shorter time constants, potentially giving the better performance, should be evaluated first. For example, the pairing of the product G/H ratio and feed D/E should be evaluated first because based on the material balance analysis in Section 5.5 it is the only logical choice and should give the best performance.

Table 5.14 Manipulated variable priority

| Controlled variable    | Manipulated variable |           |           |                    |                    |
|------------------------|----------------------|-----------|-----------|--------------------|--------------------|
|                        | $F_{D/E}$            | $F_{E/C}$ | $F_{A/C}$ | $F_{\text{purge}}$ | $T_{\text{reaCW}}$ |
| $X_{G/H}$              | 1                    | 2         | 3         | 4                  | 5                  |
| $L_{\text{reactor}}$   | 2                    | 1         | 3         | 4                  | 5                  |
| $X_{A/C}$              | 5                    | 4         | 2         | 1                  | 3                  |
| $X_B$                  | 5                    | 4         | 3         | 1                  | 2                  |
| $T_{\text{separator}}$ | 5                    | 4         | 3         | 2                  | 1                  |

When the screening procedure starts we don't have a valid upper bound, which means we cannot trim (terminate) the branches. Therefore, we choose depth-first search strategy to go down the branch and bound tree fast to find an all-integer solution.

### 5.7.3 Integrate with Integrity Requirement

Integrity is always a good property to have if it does not cause too much deterioration in dynamic performance. In this research we are interested in the feedback system that can be stabilized with feedback controllers with integral mode and any combination of controllers can be reduced to zero gain.

There are many different definitions of integrity. The one most relevant to this research is Integral Controllability with Integrity (ICI) for Decentralized control, which says that the feedback system can be stabilized with feedback controller with integral mode and any combination of controllers can be reduced to zero gain. However, the necessary and sufficient conditions for integrity are difficult to test. The positive RGA of the overall system is only a necessary condition for integrity. The much tighter necessary condition requires RGA of the paired loops to be positive for all possible sub-matrices within the original system to ensure the integrity for the system (Campo and Morari, 1994). The number of sub-matrices needed to test all possible controls structures grows exponentially with the system dimension. Therefore, we can only test the tighter necessary condition for integrity for relatively small systems (for example, not bigger than 6x6).

As discussed in Chapter 4 integrity constraints, such as positive relative gain (RGA) can be easily integrated into the solving procedure at each branch and bound node. We have chosen to evaluate the integrity constraints at each node prior to solving the mathematical programming problem for computation speed. Therefore, we solve the loop-pairing problem with the positive RGA test for all sub-matrices at each branch and bound node. Figure 5.3 shows branch and bound tree structure with the positive RGA requirement required for all sub-matrices.

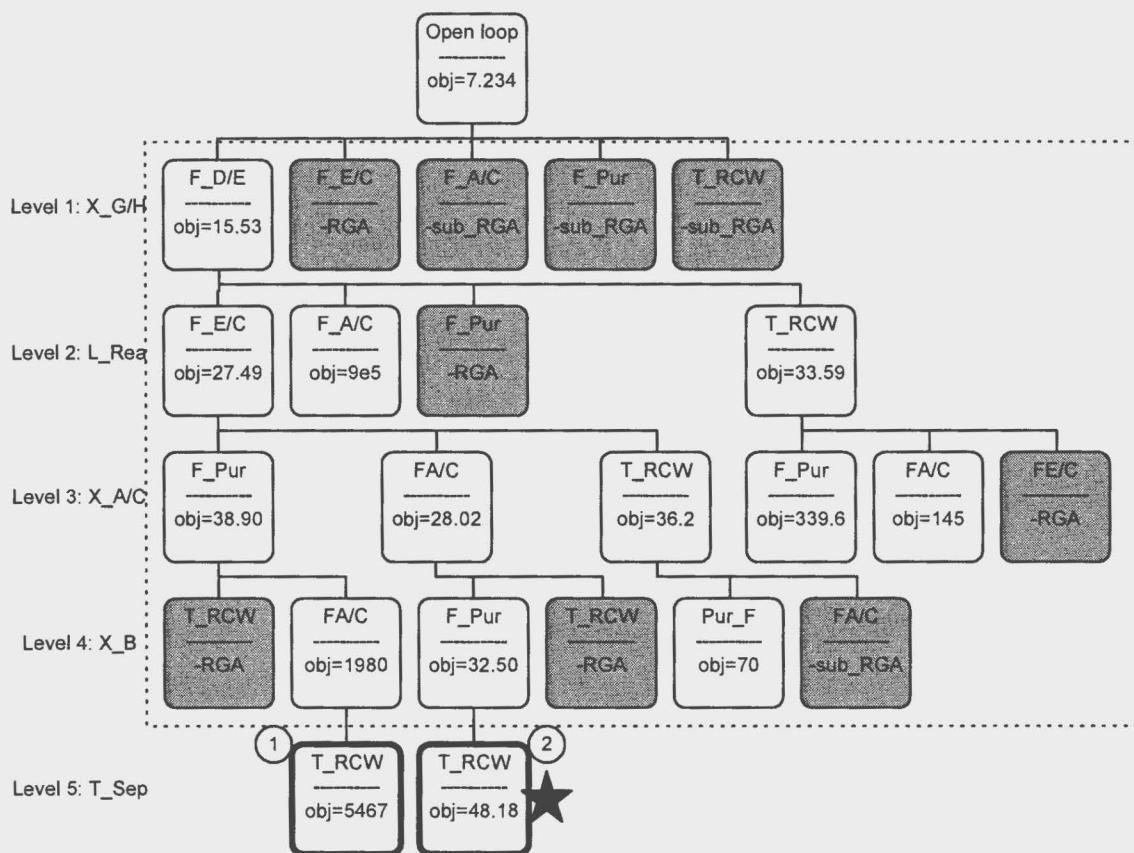


Figure 5.3 Branch &amp; bound tree with dynamic performance and integrity

(Shaded nodes are terminated for integrity; star is the optimal solution.)

In the top tree node the controllability of the loop-pairing problem is tested by solving a relaxed open-loop control problem. The intermediate nodes within the dashed line box form the tree structure for loop pairing screening procedure. The nodes at the bottom of the branch and bound tree with thicker borders represent the all-integer cases with detailed dynamic transient evaluation for candidate pairings by solving the non-linear dynamic models with actuator limits; this is a non-convex NLP. The circled number on these nodes denotes the order of all-integer calculation.

Using the previously described branching sequence of the controlled variables and the manipulated variables, we reach the node #1 at the bottom of the branch and bound tree first. By solving the node #1 a valid upper bound of objective function of value 5467 is found. Then, we backtrack in the branch and bound tree to reach the node #2, which



generates a better upper bound of objective function, 48.18. Using the upper bound generated by the node #2 the branch and bound tree is pruned. No pairing better than the node #2 can be found. So, the best pairing is node #2 defined in Table 5.15.

Table 5.15 Best pairing for 5x5 system with integrity

| Controlled variable    | Manipulated variable |
|------------------------|----------------------|
| $X_{G/H}$              | $F_{D/E}$            |
| $L_{\text{reactor}}$   | $F_{E/C}$            |
| $X_{A/C}$              | $F_{A/C}$            |
| $X_B$                  | $F_{\text{purge}}$   |
| $T_{\text{separator}}$ | $T_{\text{reaCW}}$   |

The shadowed nodes in Figure 5.3 are pruned by the positive RGA requirement without dynamic performance evaluation. The requirement of positive RGA for all sub-matrix enable us to prune branches at the top level of the tree, e.g. pairing  $X_{G/H}$ -  $T_{\text{reaCW}}$ , which eliminates all branches beneath it. And we only need to evaluate two candidate structures using the detailed transient with optimized tuning via an NLP.

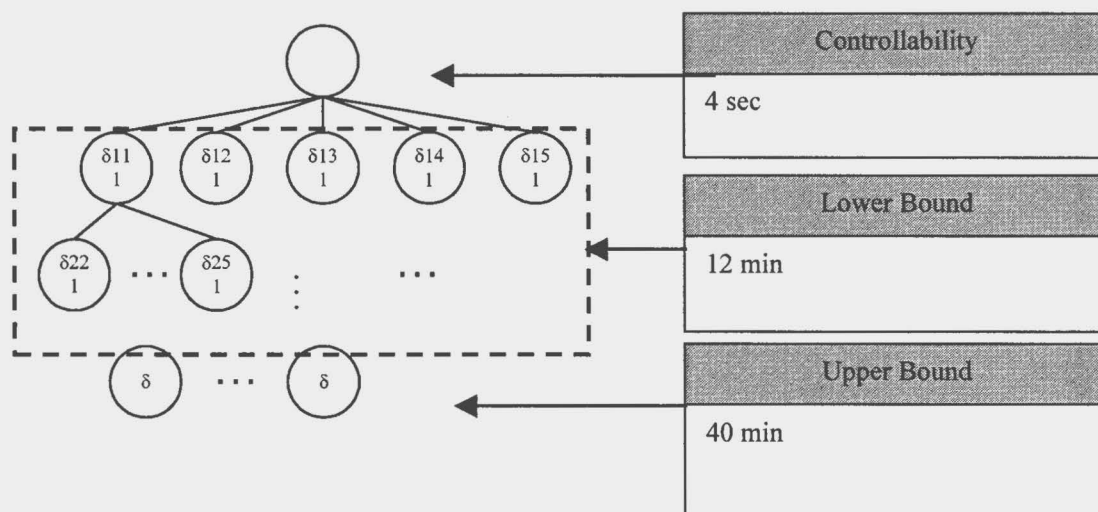


Figure 5.4 Computing time of the loop pairing problem requiring integrity

Figure 5.4 shows the computing time of the three main parts in the solution strategy for the loop pairing problem requiring integrity. The controllability check, a

convex QP, only takes 4 seconds to solve. The number of the intermediate nodes is reduced substantially by requiring positive RGA for all possible sub-matrices. The algorithm requires only 12 minutes to evaluate all relaxed nodes. The detailed transient NLP is solved only for two pairing candidates, which takes 40 minutes. The total computing time is less than one hour. Comparing with the solving time listed in Table 4.1, which are more than five days, our method is significantly faster and completed its optimization to an optimal solution.

#### **5.7.4 Paring for Best Dynamic Performance**

The quantitative information of the effect on dynamic performance caused by the integrity requirement can be very helpful for the loop pairing decision. We can solve the loop pairing problem without requiring pairing on positive RGA requirement to determine the difference in dynamic performance with and without the integrity requirement.

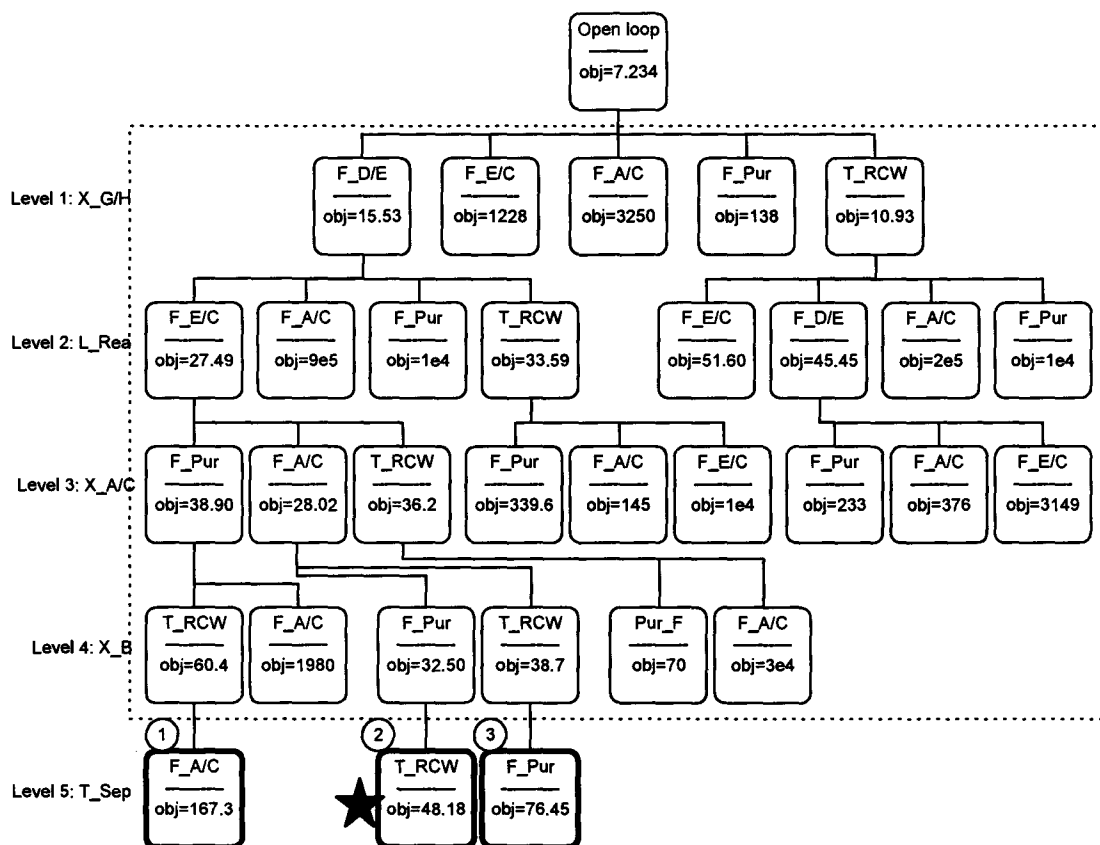


Figure 5.5 Branch and bound tree with dynamic performance only

We solve the loop pairing problem without the positive RGA requirement and generate the branch and bound tree shown in Figure 5.5. The branch and bound tree is bigger than the loop pairing problem with integrity requirement. However, the best pairing for dynamic performance (node #2) is the same with the integrity requirement (Table 5.15). This result proves the best pairing with integrity also gives the best dynamic performance for this example (but, not in all control designs).

The best pairing is physically sound and very close to the control structure published by Luyben (1996), who considered similar control objectives. Based on the material balance analysis we did before, the ratio of feed D and feed E is the most dominant “handle” for controlling the ration of product G and H. The reactor liquid phase is mostly made of product G and H. By adjusting the ratio of feed E and feed C, we control the amount of product G and H produced in the reactor and control the reactor

level. The ratio of feed A and feed C has direct impact on the ratio of component A and C in the reactant. Component B leaves the process only from purge flow; therefore, we use purge flow to control concentration of component B. The reactor cooling water temperature controls the heat removal in the reactor, which indirectly affects the temperature of the separator.

The best pairing with optimal tuning has been simulated with the linearize model. The dynamic transient is shown in Figure 5.6. The system settles after 10 hours and smooth transient within all variable bounds.

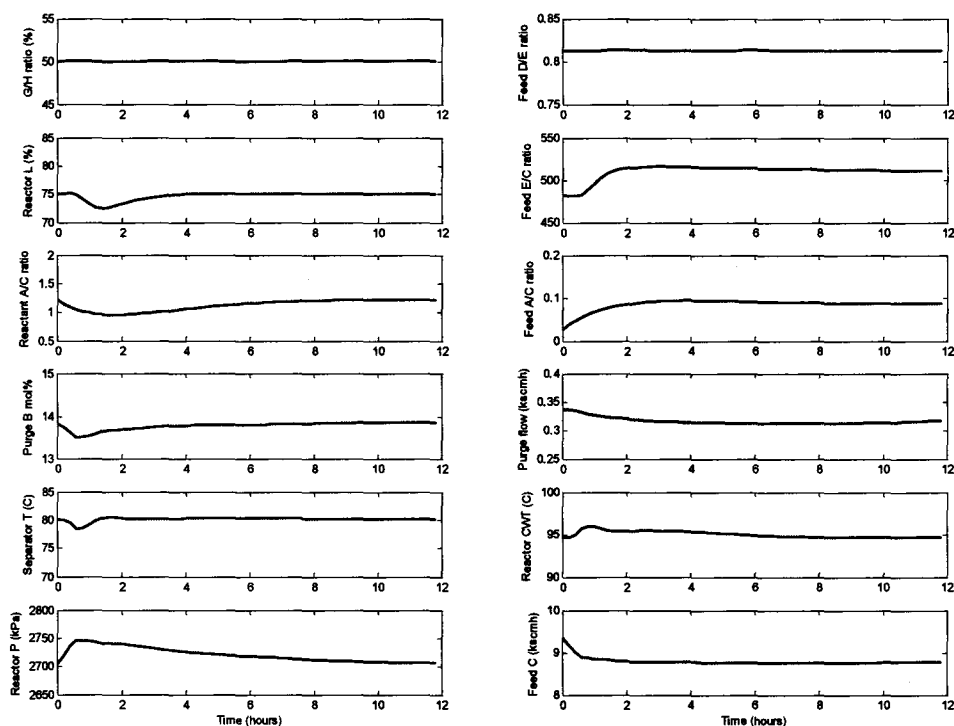


Figure 5.6 The disturbance response for the best pairings with linear system models (without noise)

The design procedure uses a linearized process model for many preliminary decisions. Therefore, we perform a final test with nonlinear process model controlled by

the best pairings and optimal tuning. Figure 5.7 shows the dynamic transient with the nonlinear process model with many higher frequency disturbances as given by Downs and Fogel. The transient is very close to the dynamic response with linear model. So we can conclude that the best control structure design based on the linear model is also adequate for nonlinear model.

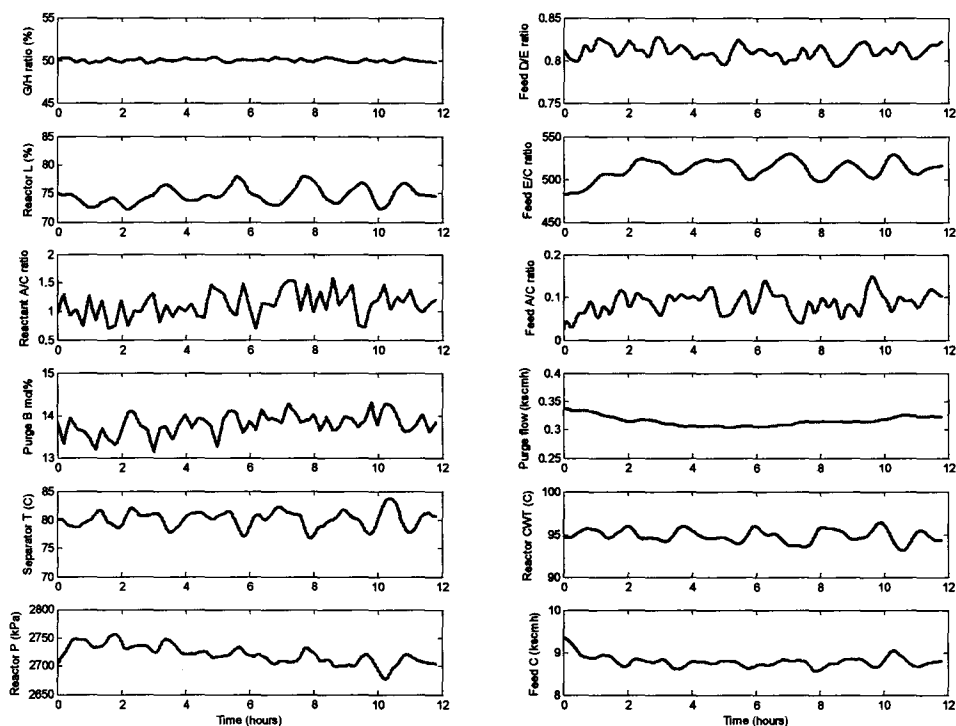


Figure 5.7 The disturbance response for the best pairings with nonlinear process model.

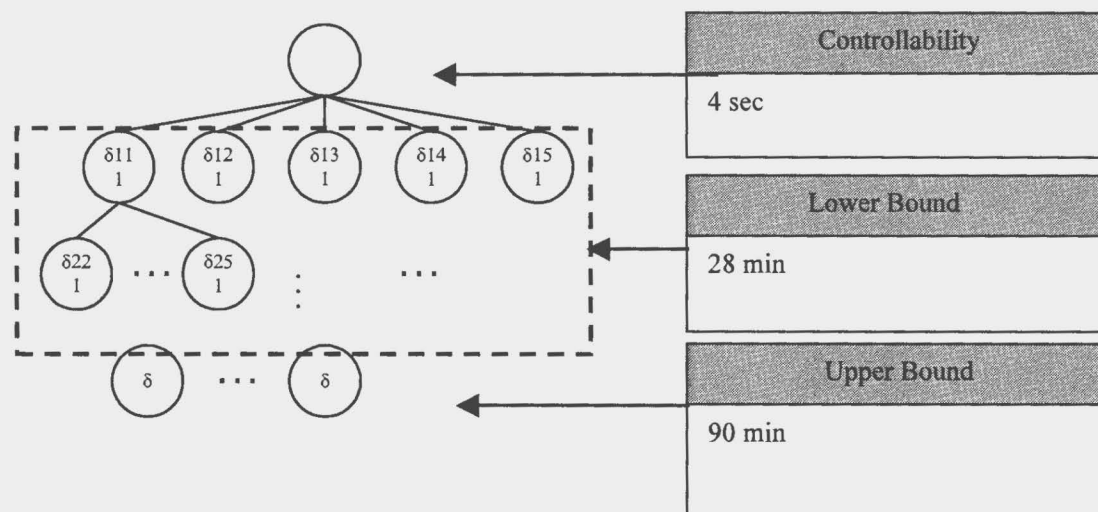


Figure 5.8 Computing time of the loop pairing problem requiring dynamic performance only

Figure 5.8 shows the computing time for three major parts in the solving strategy without requiring integrity. The total computing time is three hours, which is longer than the loop pairing problem with the integrity requirement because we evaluate more intermediate nodes and final pairing candidates. Comparing with the solving time listed in Table 4.1, which are more than five days, our method is significantly faster and completed its optimization to an optimal solution.

## 5.8 Problem Complexity Analysis

We have solved the 5x5 loop pairing problem derived from the Tennessee Eastman problem in less than one hour. The result is promising, and we will estimate the computing time for bigger system by scaling from the 5x5 Tennessee Eastman problem. Two major scaling factors for the control structure design problem are the system dimension  $n$  that defines the number of the controlled variables and manipulated variables (square system), and the time horizon  $t_{\text{final}}$  that we evaluate the dynamic transients. Since the system dimension effects the number of all possible pairings, which

has major impact on the solving time, Table 5.16 only lists the scaling data with respect to the system dimension changes.

Table 5.16 Scaling for problem size and solving time

| System dimension ( $n$ )   | 5               | 6      | 7               |
|--|-----------------|--------|-----------------|
| Time horizon (in sample periods) ( $t_{\text{final}}$ )  | 60              | 60     | 60              |
| # Continuous variables $O(n \cdot t_{\text{final}})$   | 6000            | 7200   | 8400            |
| # Integer variables ( $n^2$ )  | 25              | 36     | 49              |
| # Possible pairings ( $n!$ )   | 120             | 720    | 5040            |
| # Possible intermediate nodes of B&B tree<br>( $n + n \cdot (n-1) + \dots + n \cdot (n-1) \cdot \dots \cdot 2$ )                           | 205             | 1236   | 8659            |
| # Actual B&B tree nodes (10% of total nodes)   | 21              | 124    | 866             |
| # Grid points for each PI controller tuning parameter ( $m$ )  | 3               | 3      | 3               |
| Solving time of each Incremental tuning $O(m^2 \cdot n \cdot t_{\text{final}})$  | 40 sec          | 48 sec | 56 sec          |
| intermediate nodes Retune all loops $O(m^{2n} \cdot n \cdot t_{\text{final}})$   | Not recommended |        |                 |
| Sub-matrices RGA test $O(2^n \cdot n!)$  | 2 sec           | 24 sec | Not recommended |
| # full integer candidates (1.6% of total pairings)   | 2               | 11     | 80              |
| Solving time of the full transient evaluation $O(n \cdot t_{\text{final}})$  | 20 min          | 24 min | 28min           |
| Total solving time   | 50 min          | 6 hr   | 50 hr           |
| (# Actual nodes $\times$ solving time of intermediate node + # full integer candidates $\times$ solving time of full transient evaluation) |                 |        |                 |

Table 5.16 Scaling for problem size and solving time (continued)

| System dimension ( $n$ )   | 8               | 9       | 10              |
|--|-----------------|---------|-----------------|
| Time horizon (in sample periods) ( $t_{\text{final}}$ )  | 60              | 60      | 60              |
| # Continuous variables $O(n \cdot t_{\text{final}})$   | 9600            | 10800   | 12000           |
| # Integer variables ( $n^2$ )  | 64              | 81      | 100             |
| # Possible pairings ( $n!$ )   | 40320           | 362880  | 3628800         |
| # Possible intermediate nodes of B&B tree<br>( $n + n \cdot (n-1) + \dots + n \cdot (n-1) \cdot \dots \cdot 2$ )                           | 69280           | 623529  | 6235300         |
| # Actual B&B tree nodes (10% of total nodes)   | 6928            | 62353   | 623530          |
| # Grid points for each PI controller tuning parameter<br>( $m$ )   | 3               | 3       | 3               |
| Solving time of each incremental tuning $O(m^2 \cdot n \cdot t_{\text{final}})$  | 64 sec          | 72 sec  | 80 sec          |
| intermediate nodes Retune all loops $O(m^{2n} \cdot n \cdot t_{\text{final}})$   | Not recommended |         |                 |
| Sub-matrices RGA test $O(2^n \cdot n!)$  | Not recommended |         |                 |
| # full integer candidates (1.6% of total pairings)   | 645             | 5806    | 58060           |
| Solving time of the full transient evaluation $O(n \cdot t_{\text{final}})$  | 32 min          | 36 min  | 40min           |
| Total solving time   | 19 day          | 197 day | Not recommended |
| (# Actual nodes $\times$ solving time of intermediate node + # full integer candidates $\times$ solving time of full transient evaluation) |                 |         |                 |

The number of all possible pairings and the size of the branch and bound tree grow exponentially with the system dimension. But the actual number of calculated nodes depends on the heuristic searching parameters. Pruning branches at the top levels of branch and bound tree can significantly reduce the number of the nodes. For instance, the 5x5 Tennessee Eastman problem has 205 possible branch and bound nodes; however, only 21 (10%) of them are calculated in the algorithm. Table 5.16 lists the estimated number of the actual calculated nodes based on the percentage from the 5x5 Tennessee Eastman problem, 10% of all possible nodes. The actual number can vary from case to case.

The calculation at each intermediate node is a combination of the test of shortcut metrics and the grid search for best tuning. The shortcut metrics, such as RGA, can be calculated before solving the optimization problem. If only overall RGA is tested the testing time can be ignored. We recommend the test of all possible sub-matrices RGA for only those systems with dimension less than six because the testing time grows with the



system dimension exponentially and exceeds the computing time of tuning problem for system with dimension bigger than six.

The incremental tuning strategy (keeping the tuning of the previous loops unchanged and only tuning the new loop) is the only tractable tuning strategy for systems with dimension bigger than five.

Although the number of possible pairings is very large, our experience indicates that only few promising pairing candidates need to be solved through the difficult non-convex full transient problem. For instance, the 5x5 Tennessee Eastman problem has 120 possible pairings; however, only 2 (1.6%) of them need to solve the full transient problem. Table 5.16 list the estimated number of the full integer candidates that needs to be solved for the full transient problem base on the percentage from the 5x5 Tennessee Eastman problem, 1.6% of the all possible pairs. The actual number can vary from case to case.

The total solving time scaled from 5x5 Tennessee Eastman problem grows fast with the system dimension. For some problems, the method might become intractable for system with dimension bigger than eight. However, we expect that as the problem grows, the percentage of loop pairings with “reasonable” performance, which must be evaluated in either the lower bound or upper bound sections, *will reduce significantly*. This expectation is based on the very small gains and slow dynamics between manipulated and controlled variables that are “far apart” in large process systems, which involve many process units. Therefore, we anticipate that the number of nodes evaluated would grow much less rapidly than shown in the conservative analysis above. As a result, the tractable extension to larger systems is likely, but certainly not guaranteed.

To investigate the change in percentage of all-integer nodes needing full evaluation, we look at two different cases for the Tennessee Eastman problem. Table 5.17 compares the number of pairings with potentially good performance for two different systems, a 5x5 reduced Tennessee Eastman problem with safety-related loops are pre-selected using heuristic rules (refer to Section 5.6), and a 10x10 Tennessee Eastman problem including all manipulated variables except agitator speed and

compressor recycle flow. We would like to select pairings with integrity and relative small loop interactions. Therefore, shortcut metrics rule,  $RGA > 0.1$ , is used as criterion for pairings with potential good performance. Table 5.17 shows that the number of pairings with potential good performance only increases from 2 to 3 when the system dimension doubled. This implies that we might solve 10x10 Tennessee Eastman problem within reasonable time with proper heuristic tuning parameters.

Table 5.17 Compare the number of pairings with potential good performance for different system dimension

| Dimension | # possible pairings | # pairing with<br>RGA > 0.1 | % pairing with<br>RGA > 0.1 in all possible<br>pairing |
|-----------|---------------------|-----------------------------|--|
| 5x5       | 120                 | 2                           | 1.6%   |
| 10x10     | 3628800             | 3                           | 0.00008%   |

Table 5.18 Relative Gain Array of the 5x5 reduced Tennessee Eastman problem

|                        | $F_{D/E}$ | $F_{E/C}$ | $F_{A/C}$ | $F_{\text{purge}}$ | $T_{\text{reactor CW}}$ |
|------------------------|-----------|-----------|-----------|--------------------|-------------------------|
| $X_{G/H}$              | 1.9872    | -0.9926   | 0.0025    | 0.0024             | 0.0005                  |
| $L_{\text{reactor}}$   | -0.6218   | 1.2132    | 0.0159    | -0.0075            | 0.4002                  |
| $X_{A/C}$              | 0.0051    | -0.0018   | 0.9228    | 0.0021             | 0.0717                  |
| $X_B$                  | -0.0002   | 0.0011    | 0.0036    | 1.0024             | -0.0068                 |
| $T_{\text{separator}}$ | -0.3703   | 0.7801    | 0.0552    | 0.0007             | 0.5343                  |

Table 5.19 Relative Gain Array of the 10x10 Tennessee Eastman problem

|                        | $F_D$    | $F_E$    | $F_A$    | $F_C$    | $F_{\text{purge}}$ | $F_{\text{separator}}$ | $F_{\text{stripper}}$ | $F_{\text{steam}}$ | $T_{\text{reactor CW}}$ | $T_{\text{condenser CW}}$ |
|------------------------|----------|----------|----------|----------|--------------------|------------------------|-----------------------|--------------------|-------------------------|---------------------------|
| $X_{G/H}$              | 0.993325 | 0.098782 | -0.00036 | -0.09671 | 0.002536           | -0.00489               | -4.9E-12              | 0.000228           | 0.003505                | 0.003587                  |
| $L_{\text{reactor}}$   | -0.20256 | 0.680989 | -0.02752 | 0.680234 | -0.01505           | 1.208472               | -1.9E-10              | 0.015292           | 0.075642                | -1.4155                   |
| $X_{A/C}$              | 0.005614 | -0.00817 | 0.858484 | 0.027635 | 0.001543           | -0.00173               | -8.8E-11              | -0.00062           | 0.078132                | 0.039114                  |
| $P_{\text{reactor}}$   | 0.108616 | -0.62329 | 0.025277 | -0.47952 | 0.002403           | -0.98461               | 4.31E-10              | -0.02413           | 0.188045                | 2.787213                  |
| $X_B$                  | -0.02088 | -0.01419 | 0.011981 | 0.071058 | 1.018802           | -0.07534               | -1.5E-11              | 0.006684           | 0.017024                | -0.01514                  |
| $L_{\text{stripper}}$  | -1.29902 | -1.05127 | 0.12291  | 2.892055 | -0.03431           | 0.352323               | -2.1E-06              | -0.00031           | -0.00103                | 0.018656                  |
| $F_{\text{product}}$   | 2.42E-05 | 1.84E-05 | -1.6E-06 | -5.2E-05 | 1.05E-06           | 8.11E-06               | 1.000002              | -1.4E-08           | 1.76E-07                | -7E-07                    |
| $T_{\text{stripper}}$  | 0.216598 | 0.289747 | -0.04104 | -0.41495 | 0.003056           | -0.02353               | 2.73E-12              | 0.929742           | 0.038855                | 0.001526                  |
| $L_{\text{separator}}$ | 1.285952 | 1.461368 | -0.06363 | -1.89485 | 0.021991           | 0.198216               | 9.3E-12               | -0.00136           | -0.01336                | 0.005675                  |
| $T_{\text{separator}}$ | -0.08767 | 0.166023 | 0.113913 | 0.215094 | -0.00097           | 0.331082               | -1.1E-10              | 0.074474           | 0.613185                | -0.42513                  |

As a fall-back position for large systems, only the first two segments, controllability and lower bound, could be applied. These nodes require much less computing time than the all-integer nodes. This modified approach would yield candidates with acceptable performance based on the linear dynamic process model and integrity (if desired) but not considering manipulated variables saturation. If not one candidate appeared superior, some engineering judgment would be required when selecting a few candidates for evaluation of the upper bound problem, which is much more computationally intensive.

## 5.9 Summary

In this chapter, we applied our systematic loop pairing strategy to the Tennessee Eastman problem to demonstrate its ability to solve medium size industrial problems. At first the base layer control loops are identified and are closed to reject common disturbances and ensure safety. This step is essential for good control design and is not included in the automated method developed in this research. After the process insight is included in defining the controlled and manipulated variables, a 5x5 system remains for which a proper loop pairing is designed.

We have applied several NLP, MINLP and global solvers to solve this 5x5 loop pairing problem in Chapter 4. The result is none of these solvers can solve a medium size

control structure design problem in a reasonable time (5 days). However, our algorithm finds a physically sound pairing with good performance in less than one hour.

The good performance of this method results from the following.

- An efficient lower bound calculation at intermediate nodes allows screening a large number of pairing in relatively short computing time.
- The branch and bound search takes advantage of search parameters based on process knowledge to guide search process. By properly selecting the pairing sequence of controller variables the algorithm prunes many branches at the top level of the branch and bound tree and reduced the number of actual calculated nodes. The guidelines for selecting the search parameters are generic and should be applicable to many other process control designs.
- Integrating short-cut metrics, i.e. RGA for integrity, in the algorithm further reduces the number of actual calculated nodes and saves solving time.

A scaling study based on the 5x5 Tennessee Eastman pairing problem shows our methodology can handle 8x8 system with proper heuristic search parameters. We expect that even larger problems will be tractable because the percentage of good pairings needing evaluation will decrease with problem size in control design for the process industries.

## Chapter 6

### Selection of Block Centralized Structures

Multivariable Model Predictive Control (MPC) has been widely used in chemical plants. The common plant-wide control structures consist of several multivariable MPC and many single loop PID controllers, which can be described as block centralized structure.

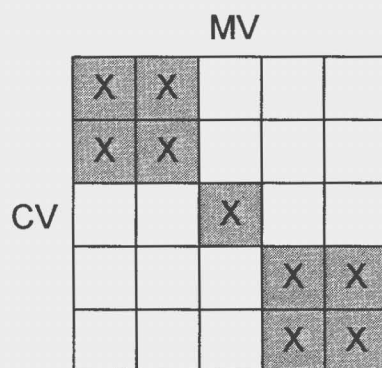


Figure 6.1 Block centralized structure

Figure 6.1 illustrates a block centralized structure as a matrix, in which each row represents a controlled variable and each column represents a manipulated variable. The gray cells represent information feedback from the controlled variables to the corresponding manipulated variables for a specific control design. In other words, the

gray blocks along the diagonal represent single loop or multivariable controllers, which do not exchange information with each other. In this research each block is a MPC controller.

A block centralized structure requires the selected controlled variable and manipulated variable pairings to form blocks along the diagonal after row and column permutations. A block centralized structure is very flexible and can represent various control structures from the multi-loop structure (loop pairing problem) to the fully centralized structure. The loop pairing problem that we have solved in the previous chapters is a special case of block centralized structure, which has only 1x1 blocks on the diagonal.

In this chapter, we will extend the design formulation and the solving technology to accommodate the more general block centralized structure.

## **6.1 Block Centralized Design Problem Definition**

The discussion about the control design objectives, multiple objective optimization and mathematical formulation in Chapter 3 is valid in this chapter. The things that we need to change are the controller equations and the integer structural constraints imposed on the controller structure to represent a block centralized controller.

### **6.1.1 Unconstrained MPC Controller**

Because multivariable MPC is needed if the block size is great than one, the mathematical formulation of the PI controller equations we used in Chapter 3 is inadequate for a block centralized control structure. In this section, we derive a block centralized MPC controller formulation that can be used in the mathematical formulation for the block centralized control structure selection.

Dynamic Matrix Controller (DMC) is the MPC algorithm developed by Cutler and Ramaker (1979) and has been widely used in chemical industry (Morari and Lee, 1999; Ricker, 1991). The open-loop stable process can be represented by Finite Step

Response (FSR) model, which is explained in a SISO system as the following (Garcia *et al.*, 1989):

$$y(k+1) = y_0 + \sum_{j=1}^{k+1} a_j \Delta u(k-j+1) \quad (6.1)$$

where  $a_j$  is the step response coefficient at time step  $j$ ,  $y$  is system output,  $y_0$  is the initial value of output, and  $u$  is input. The overall effect of all the input steps on the output is the sum of each individual effect since we assume that the system is linear. In the following discussion, we assume all variables are deviation variables from steady state values, therefore,  $y_0$  is ignored. Equation (6.1) can be written in matrix form as (6.2a).

$$\begin{bmatrix} a_1 & 0 & 0 & 0 \\ a_2 & a_1 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ a_k & a_{k-1} & \cdots & a_1 \\ \hline a_{k+1} & a_k & \cdots & a_2 \\ a_{k+2} & a_{k+1} & \cdots & a_3 \\ \vdots & \vdots & \ddots & \vdots \\ a_{k+p} & a_{k+p-1} & \cdots & a_{p+1} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline a_1 & 0 & 0 & 0 \\ a_2 & a_1 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ a_p & a_{p-1} & \cdots & a_{p-m+1} \end{bmatrix} \begin{bmatrix} \Delta u(0) \\ \Delta u(1) \\ \vdots \\ \Delta u(k-1) \\ \hline \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+m) \end{bmatrix} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(k) \\ \hline y(k+1) \\ y(k+2) \\ \vdots \\ y(k+p) \end{bmatrix} \quad (6.2a)$$

Assume the current step is  $k$ , Equation (6.2a) can be partitioned in sub-matrices as (6.2b):

$$\begin{bmatrix} \mathbf{A}^c & \mathbf{0} \\ \mathbf{A}^f & \mathbf{A}^p \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}^c \\ \Delta \mathbf{u}^p \end{bmatrix} = \begin{bmatrix} \mathbf{y}^c \\ \mathbf{y}^p \end{bmatrix} \quad (6.2b)$$

where  $\Delta \mathbf{u}^c$  and  $\mathbf{y}^c$  are current and past input and output vectors with dimension of  $k$ ;  $\Delta \mathbf{u}^p$  and  $\mathbf{y}^p$  are the predicted future, with input vector having dimension  $m$  and future output vector having dimension  $p$ .  $\mathbf{A}^c$ ,  $\mathbf{A}^f$  and  $\mathbf{A}^p$  are the dynamic matrices. In the MPC algorithm we are interested in the future prediction of the input and the output and the current and present values are known constants; therefore, equation (6.2b) can be simplified to yield only the effects of future control actions.

$$\mathbf{y}^p = \mathbf{A}^f \Delta \mathbf{u}^c + \mathbf{A}^p \Delta \mathbf{u}^p \quad (6.2c)$$

The objective of MPC algorithm is to minimize the future output  $\mathbf{y}^p$  deviation from the desired trajectory  $\mathbf{y}^{sp}$  by manipulating the future input  $\Delta \mathbf{u}^p$ . An appropriate objective

function is the weighted error squared plus a penalty for changes to the manipulated variables (the move suppression) to provide robustness and desired moderate changes to the manipulated variable.

$$\begin{aligned} \min_{\Delta \mathbf{u}^p} & (\mathbf{y}^{sp} - \mathbf{y}^p)^T \mathbf{Q} (\mathbf{y}^{sp} - \mathbf{y}^p) + (\Delta \mathbf{u}^p)^T \mathbf{R} (\Delta \mathbf{u}^p) \\ \text{s.t.} \quad & \mathbf{y}^p = \mathbf{A}^f \Delta \mathbf{u}^c + \mathbf{A}^p \Delta \mathbf{u}^p + \mathbf{d}^p \end{aligned} \quad (6.3)$$

Equation (6.3) is the mathematical formulation of the unconstrained MPC problem, where  $\mathbf{Q}$  is the weighting for controlled variables,  $\mathbf{R}$  is the move suppression factor, and  $\mathbf{d}^p$  is the future value of the disturbance, which is estimated based on feedback. The unconstrained MPC problem has the analytical solution in equation (6.4) that can be used as controller equation (Marlin 2000).

$$\begin{aligned} \Delta \mathbf{u}^p &= \mathbf{K}^{DMC} (\mathbf{y}^{sp} - \mathbf{A}^f \Delta \mathbf{u}^c - \mathbf{d}^p) \\ \mathbf{K}^{DMC} &= [(\mathbf{A}^p)^T \mathbf{Q} (\mathbf{A}^p) + \mathbf{R}]^{-1} (\mathbf{A}^p)^T \mathbf{Q} \end{aligned} \quad (6.4)$$

The discussion above is based on SISO system. We can extend equation (6.4) to cover MIMO systems by changing the variables as the following.

$$\mathbf{A}^f = \begin{bmatrix} \mathbf{A}_{11}^f & \mathbf{A}_{12}^f & \cdots & \mathbf{A}_{1n}^f \\ \mathbf{A}_{21}^f & \mathbf{A}_{22}^f & \cdots & \mathbf{A}_{2n}^f \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}^f & \mathbf{A}_{n2}^f & \cdots & \mathbf{A}_{nn}^f \end{bmatrix} \quad \mathbf{A}^p = \begin{bmatrix} \mathbf{A}_{11}^p & \mathbf{A}_{12}^p & \cdots & \mathbf{A}_{1n}^p \\ \mathbf{A}_{21}^p & \mathbf{A}_{22}^p & \cdots & \mathbf{A}_{2n}^p \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}^p & \mathbf{A}_{n2}^p & \cdots & \mathbf{A}_{nn}^p \end{bmatrix} \quad (6.5a)$$

$$\mathbf{Q} = \begin{bmatrix} Q_1 & & & \\ & \ddots & & \\ & & Q_1 & \\ & & & Q_2 & \\ & & & & \ddots \\ & & & & & Q_n \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} R_1 & & & \\ & \ddots & & \\ & & R_1 & \\ & & & R_2 & \\ & & & & \ddots \\ & & & & & R_n \end{bmatrix} \quad (6.5b)$$

$$\Delta \mathbf{u}^p = [\Delta u_1(k) \quad \cdots \quad \Delta u_1(k+m) \quad \Delta u_2(k) \quad \cdots \quad \Delta u_2(k+m) \quad \cdots \quad \Delta u_n(k+m)]^T \quad (6.5c)$$

$$\Delta \mathbf{u}^c = [\Delta u_1(1) \quad \cdots \quad \Delta u_1(k-1) \quad \Delta u_2(1) \quad \cdots \quad \Delta u_2(k-1) \quad \cdots \quad \Delta u_n(k-1)]^T \quad (6.5d)$$

$$\mathbf{y}^{sp} = [y_1^{sp}(k) \quad \cdots \quad y_1^{sp}(k+p) \quad y_2^{sp}(k) \quad \cdots \quad y_2^{sp}(k+p) \quad \cdots \quad y_n^{sp}(k+p)]^T \quad (6.5e)$$

$$\mathbf{d}^p = [d_1^p(k) \quad \cdots \quad d_1^p(k+p) \quad d_2^p(k) \quad \cdots \quad d_2^p(k+p) \quad \cdots \quad d_n^p(k+p)]^T \quad (6.5f)$$



Equation (6.5a) extends dynamic matrices to multivariable forms, where  $\mathbf{A}_{ij}^f$  is the dynamic matrix relating the past changes of the  $j^{\text{th}}$  input to the future prediction of the  $i^{\text{th}}$  output, and  $\mathbf{A}_{ij}^p$  is the dynamic matrix relates the future changes of the  $j^{\text{th}}$  input to the prediction of the  $i^{\text{th}}$  output. Controlled variable weighting matrix  $\mathbf{Q}$  is extended by aligning weightings for all controlled variables on the diagonal (6.5b). The same extension is applied to move suppression fact  $\mathbf{R}$ . The vectors are extended to multivariable forms by stacking multiple variables one after another (6.5c-f).

### 6.1.2 Block Centralized MPC Controller

In the block centralized control structure, each block is controlled by a MPC controller, no matter the block size. In this section we show all MPC controllers in the block centralized control structure can be combined into one MPC controller with structured dynamic matrices. (Note that for this block centralized formulation a single loop will be controlled by a single-variable MPC controller and not a PI algorithm.)

Assume a block centralized control structure has  $q$  blocks. Equation (6.6) shows the variables partitioned into sub-vectors or sub-matrices according the blocks.

$$\begin{aligned} \Delta \mathbf{u}^p &= \begin{bmatrix} \Delta \mathbf{u}^{p_1} \\ \Delta \mathbf{u}^{p_2} \\ \vdots \\ \Delta \mathbf{u}^{p_q} \end{bmatrix} & \Delta \mathbf{u}^c &= \begin{bmatrix} \Delta \mathbf{u}^{c_1} \\ \Delta \mathbf{u}^{c_2} \\ \vdots \\ \Delta \mathbf{u}^{c_q} \end{bmatrix} & \mathbf{y}^{sp} &= \begin{bmatrix} \mathbf{y}^{sp_1} \\ \mathbf{y}^{sp_2} \\ \vdots \\ \mathbf{y}^{sp_q} \end{bmatrix} & \mathbf{d}^p &= \begin{bmatrix} \mathbf{d}^{p_1} \\ \mathbf{d}^{p_2} \\ \vdots \\ \mathbf{d}^{p_q} \end{bmatrix} \\ \mathbf{Q} &= \begin{bmatrix} \mathbf{Q}_1 & & & \\ & \mathbf{Q}_2 & & \\ & & \ddots & \\ & & & \mathbf{Q}_q \end{bmatrix} & \mathbf{R} &= \begin{bmatrix} \mathbf{R}_1 & & & \\ & \mathbf{R}_2 & & \\ & & \ddots & \\ & & & \mathbf{R}_q \end{bmatrix} \end{aligned} \quad (6.6)$$

For block  $j$ , the MPC controller equation is:

$$\begin{aligned} \Delta \mathbf{u}^{p_j} &= \mathbf{K}^{DMC_j} (\mathbf{y}^{sp_j} - \mathbf{A}_{jj}^f \Delta \mathbf{u}^{c_j} - \mathbf{d}^{p_j}) \\ \mathbf{K}^{DMC_j} &= [(\mathbf{A}^{p_j})^T \mathbf{Q}_j (\mathbf{A}^{p_j}) + \mathbf{R}_j]^{-1} (\mathbf{A}^{p_j})^T \mathbf{Q}_j \end{aligned} \quad (6.7)$$

We can combine all MPC controllers into one matrix form as shown in the following.

$$\begin{bmatrix} \Delta \mathbf{u}^{p_1} \\ \Delta \mathbf{u}^{p_2} \\ \vdots \\ \Delta \mathbf{u}^{p_q} \end{bmatrix} = \begin{bmatrix} \mathbf{K}^{DMC_1} & & & \\ & \mathbf{K}^{DMC_2} & & \\ & & \ddots & \\ & & & \mathbf{K}^{DMC_q} \end{bmatrix} \cdot \left( \begin{bmatrix} \mathbf{y}^{sp_1} \\ \mathbf{y}^{sp_2} \\ \vdots \\ \mathbf{y}^{sp_q} \end{bmatrix} - \begin{bmatrix} \mathbf{A}^f_1 & & & \\ & \mathbf{A}^f_2 & & \\ & & \ddots & \\ & & & \mathbf{A}^f_q \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}^c_1 \\ \Delta \mathbf{u}^c_2 \\ \vdots \\ \Delta \mathbf{u}^c_q \end{bmatrix} - \begin{bmatrix} \mathbf{d}^{p_1} \\ \mathbf{d}^{p_2} \\ \vdots \\ \mathbf{d}^{p_q} \end{bmatrix} \right) \quad (6.8)$$

$$\begin{bmatrix} \mathbf{K}^{DMC_1} & & & \\ & \mathbf{K}^{DMC_2} & & \\ & & \ddots & \\ & & & \mathbf{K}^{DMC_q} \end{bmatrix} = \begin{bmatrix} [(\mathbf{A}^{p_1})^T \mathbf{Q}_1 (\mathbf{A}^{p_1}) + \mathbf{R}_1]^{-1} (\mathbf{A}^{p_1})^T \mathbf{Q}_1 & & & \\ & \ddots & & \\ & & [(\mathbf{A}^{p_q})^T \mathbf{Q}_q (\mathbf{A}^{p_q}) + \mathbf{R}_q]^{-1} (\mathbf{A}^{p_q})^T \mathbf{Q}_q & \end{bmatrix}$$

After simple matrix manipulations we get a simplified equation for all MPC controllers within the block centralized structure as:

$$\begin{aligned} \Delta \mathbf{u}^p &= \mathbf{K}^{DMC_{block}} \cdot (\mathbf{y}^{sp} - \mathbf{A}^f_{block} \Delta \mathbf{u}^c - \mathbf{d}^p) \\ \mathbf{K}^{DMC_{block}} &= [(\mathbf{A}^{p_{block}})^T \mathbf{Q} (\mathbf{A}^{p_{block}}) + \mathbf{R}]^{-1} (\mathbf{A}^{p_{block}})^T \mathbf{Q} \\ \mathbf{A}^f_{block} &= \begin{bmatrix} \mathbf{A}^f_1 & & & \\ & \mathbf{A}^f_2 & & \\ & & \ddots & \\ & & & \mathbf{A}^f_q \end{bmatrix} \\ \mathbf{A}^{p_{block}} &= \begin{bmatrix} \mathbf{A}^{p_1} & & & \\ & \mathbf{A}^{p_2} & & \\ & & \ddots & \\ & & & \mathbf{A}^{p_q} \end{bmatrix} \end{aligned} \quad (6.9)$$

Equation (6.9) is identical to the MPC controller equation (6.4) except the dynamic matrix  $\mathbf{A}^f_{block}$  and  $\mathbf{A}^{p_{block}}$  are different from equation (6.5a). The dynamic matrices  $\mathbf{A}^f_{block}$  and  $\mathbf{A}^{p_{block}}$  are not full matrices but have the same matrix structure as the block centralized control structure. We emphasize that the block centralized controller is not in

general equivalent to the fully centralized controller because of the zeros in the controller equation (6.9) where the true plant would have causal models.

The conclusion is that all unconstrained MPC controllers in the block centralized control structure are equivalent to a special unconstrained, fully-centralized MPC controllers with the block centralized structure imposed on the dynamic matrix  $\mathbf{A}^f$  and  $\mathbf{A}^p$ . Therefore, the integer structural constraints are imposed on the dynamic matrices  $\mathbf{A}^f$  and  $\mathbf{A}^p$  in our mathematical formulation to generate the block centralized structure controller.

### 6.1.3 Block Centralized Structural Constraints

In Chapter 3 the integer structural constraints for multi-loop structure are relatively simple because only SISO controllers are allowed. From the block centralized structure point of view, the size of each block must be 1x1. For a general block centralized structure, the size of each block becomes a variable that can be adjusted to achieve better performance. Therefore, more complex structural constraints are needed.

First, we introduce two parameters, the maximum number of the blocks  $q$  and the maximum block size  $z$ , which define the complexity of the control structure. If we want the final control structure close to multi-loop structure, we can increase  $q$  and decrease  $z$  which will require many, small blocks. When  $z$  is equal to one and  $q$  is equal to the dimension of the system inputs (or dimension of the outputs, which is the same) a multiloop design will be required.

Three binary matrices, i.e. the block structure matrix  $\delta$  with dimension of  $n \times n$ , the row block matrix  $\alpha$  with dimension of  $n \times q$ , and column block matrix  $\beta$  with dimension of  $q \times n$ , are needed to address the block centralized structure constraints. Equations (6.10a-c) defines the elements in binary matrices  $\delta$ ,  $\alpha$  and  $\beta$ .

$$\delta_{ij} = \begin{cases} 1 & j\text{th MV is paired with } i\text{th CV} \\ 0 & \text{Otherwise} \end{cases} \quad (6.10a)$$

$$\alpha_{ib} = \begin{cases} 1 & i\text{th row of } \delta \text{ belongs to } b\text{th block} \\ 0 & \text{Otherwise} \end{cases} \quad (6.10b)$$

$$\beta_{bj} = \begin{cases} 1 & j\text{th column of } \delta \text{ belongs to } b\text{th block} \\ 0 & \text{Otherwise} \end{cases} \quad (6.10c)$$

Figure 6.2 illustrates the relationship of the binary matrices  $\delta$ ,  $\alpha$  and  $\beta$ . Each row of  $\alpha$  is associated with corresponding row of  $\delta$ , and records the controlled variable belongs to which block. Each column of  $\beta$  is associated with corresponding column of  $\delta$ , and records the manipulated variable belongs to which block.

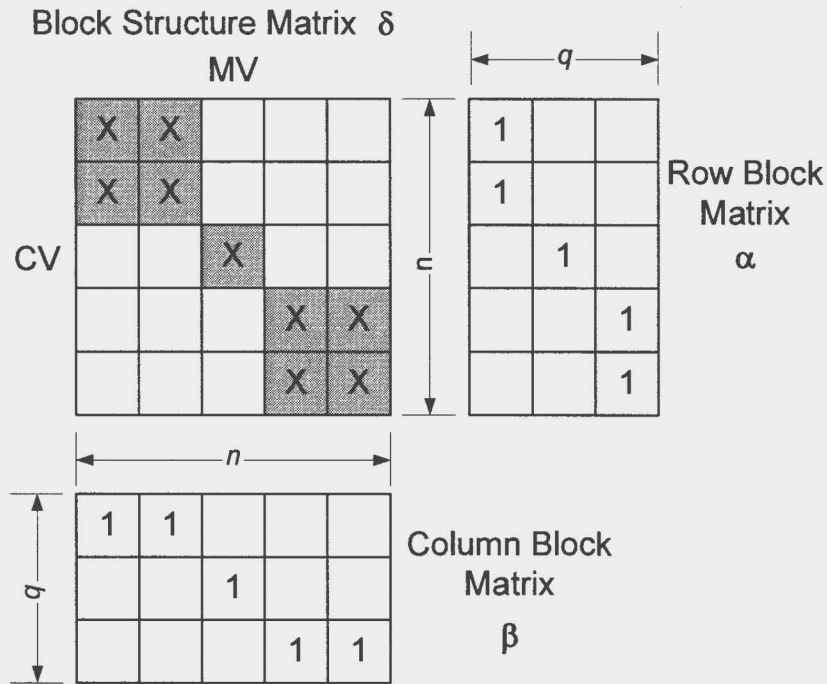


Figure 6.2 Relationship of binary matrices  $\delta$ ,  $\alpha$  and  $\beta$

The block centralized structure of  $\delta$  can be modeled using the following constraints.

$$\sum_{b=1}^q \alpha_{ib} = 1 \quad \forall i = 1 \cdots n \quad (6.11a)$$

$$\sum_{b=1}^q \beta_{bj} = 1 \quad \forall j = 1 \cdots n \quad (6.11b)$$

$$\sum_{i=1}^n \alpha_{ib} = s_b \leq z \quad \forall b = 1 \cdots q \quad (6.11c)$$

$$\sum_{j=1}^n \beta_{bj} = s_b \leq z \quad \forall b = 1 \cdots q \quad (6.11d)$$

$$\sum_{b=1}^q s_b = n \quad (6.11e)$$

$$\delta_{ij} = \sum_{b=1}^q \alpha_{ib} \cdot \beta_{bj} \quad \forall i, j \quad (6.11f)$$

Constraint (6.11a) states that one row (controlled variable) can belong to only one block; similarly, constraint (6.11b) states one column (manipulated variable) can belong to only one block. Constraints (6.11c) and (6.11d) limit the maximum block size to  $z$ , where  $s$  is the vector of the block sizes. Constraint (6.11e) ensures all block sizes sum up to system dimension. Constraints (6.11f) forces  $\delta_{ij}$  to zero when  $i$ th row (controlled variable) and  $j$ th column (manipulated variable) do not belong to the same block. Constraints (6.11f) forms the block centralized structure of  $\delta$  by forcing all off-diagonal elements (marked as blank cells in Figure 6.2) to zero.

#### 6.1.4 Block Centralized Design Problem Formulation

We can combine the control design objectives, multiple objective optimization and mathematical formulation in Chapter 3 with the unconstrained block centralized MPC controller equation and the block centralized structural constraints in the following mathematical formulation for block centralized control structure design problem.

$$\min_{\delta, \alpha, \beta, \mathbf{R}, s_k^U(t), s_k^L(t)} \sum_k \sum_{t=1}^{tf} [\mathbf{e}_k(t)^T \mathbf{Q} \mathbf{e}_k(t)] \quad (6.12a)$$

s.t.

$$\begin{aligned} \mathbf{x}_k(t+1) &= \mathbf{A}_k \mathbf{x}_k(t) + \mathbf{B}_k \mathbf{u}_k(t) + \mathbf{W}_k \mathbf{d}(t) \\ \mathbf{y}_k(t) &= \mathbf{C}_k \mathbf{x}_k(t) + \mathbf{V}_k \mathbf{d}(t) + \mathbf{N}(t) \\ \mathbf{e}_k(t) &= \mathbf{sp}(t) - \mathbf{y}_k(t) \end{aligned} \quad (6.12b)$$

$$\begin{aligned}
\mathbf{u}_k(t) &= \mathbf{u}_k(t-1) + \Delta \mathbf{u}_k(t) - \mathbf{s}_k^U(t) + \mathbf{s}_k^L(t) \\
\mathbf{u}^L(t) &\leq \mathbf{u}_k(t) \leq \mathbf{u}^U(t) \\
[u_j(t) - u_j^L] \cdot s_j^L(t) &= 0 \\
[u_j(t) - u_j^U] \cdot s_j^U(t) &= 0 \\
\mathbf{s}_k^U(t) &\geq 0 \\
\mathbf{s}_k^L(t) &\geq 0
\end{aligned} \tag{6.12c}$$

$$\Delta \mathbf{u}_k^p(t) = \mathbf{K}^{DMC} \cdot (\mathbf{y}^{sp}(t) - \mathbf{A}^f_{block} \Delta \mathbf{u}_k^c(t) - \mathbf{d}_k^p(t)) \tag{6.12d}$$

$$\begin{aligned}
\mathbf{K}^{DMC} &= \left[ (\mathbf{A}^p_{block})^T \mathbf{Q} (\mathbf{A}^p_{block}) + \mathbf{R} \right]^{-1} (\mathbf{A}^p_{block})^T \mathbf{Q} \\
\mathbf{A}^f_{block} &= \begin{bmatrix} \mathbf{A}_{11}^f \delta_{11} & \mathbf{A}_{12}^f \delta_{12} & \cdots & \mathbf{A}_{1n}^f \delta_{1n} \\ \mathbf{A}_{21}^f \delta_{21} & \mathbf{A}_{22}^f \delta_{22} & \cdots & \mathbf{A}_{2n}^f \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}^f \delta_{n1} & \mathbf{A}_{n2}^f \delta_{n2} & \cdots & \mathbf{A}_{nn}^f \delta_{nn} \end{bmatrix} \\
\mathbf{A}^p_{block} &= \begin{bmatrix} \mathbf{A}_{11}^p \delta_{11} & \mathbf{A}_{12}^p \delta_{12} & \cdots & \mathbf{A}_{1n}^p \delta_{1n} \\ \mathbf{A}_{21}^p \delta_{21} & \mathbf{A}_{22}^p \delta_{22} & \cdots & \mathbf{A}_{2n}^p \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{n1}^p \delta_{n1} & \mathbf{A}_{n2}^p \delta_{n2} & \cdots & \mathbf{A}_{nn}^p \delta_{nn} \end{bmatrix}
\end{aligned} \tag{6.12e}$$

$$\begin{aligned}
\delta_{ij} &= \sum_{b=1}^q \alpha_{ib} \cdot \beta_{bj} \quad \forall i, j \\
\sum_{b=1}^q \alpha_{ib} &= 1 \quad \forall i = 1 \cdots n \\
\sum_{b=1}^q \beta_{bj} &= 1 \quad \forall j = 1 \cdots n \\
\sum_{i=1}^n \alpha_{ib} &= s_b \leq z \quad \forall b = 1 \cdots q \\
\sum_{j=1}^n \beta_{bj} &= s_b \leq z \quad \forall b = 1 \cdots q \\
\sum_{b=1}^q s_b &= n
\end{aligned} \tag{6.12f}$$

Although equation (6.12d) is unconstrained MPC equation, the closed loop response is constrained by equation (6.12c). Manipulated variables will be clamped to their upper or lower bounds if the calculated values exceed the constraints. The

formulation guarantees that the manipulated variables do not exceed the constraints at the current step; however, it does not take into account of the constraints in the prediction horizon. Therefore, the performance is suboptimal when compared with the constrained MPC. In some constrained cases the suboptimality can be serious. So the global optimality is not guaranteed. The advantage of this formulation is that it avoids solving a QP at each time step as a constrained MPC. The weighting matrix  $\mathbf{Q}$  in equation (6.12d) is selected basing on the relative importance of the controlled variables; and is not a decision variable of problem (6.12). The move suppression matrix  $\mathbf{R}$  is a decision variable of problem (6.12).

## 6.2 Tailored Solving Strategy

The block centralized structure design problem (6.12) is a very challenging MINLP and more difficult than the loop pairing problem to solve. The difficulties are two-fold. First, the possible number of block centralized structures is greater than the multi-loop structure and grows exponentially with the system dimension. Kariwala *et al.* (2003) gave an approximation for the number of the structure alternatives for block centralized structure as (6.13).

$$N(n) \approx n!^{1.52} \quad (6.13)$$

Second, the mathematical formulation of the block centralized structure design problem is more complex than the loop pairing problem. The integer structural constraints (6.12f) has bilinear terms involving binary variables. The MPC controller gain  $\mathbf{K}^{DMC}$  calculation (6.12d) is also highly nonlinear.

We extend the solving methodology developed in Chapter 4, which takes advantage of the special problem structure to guide the branch & bound search and uses control knowledge to efficiently generate a reasonably tight lower bound at each node.

### 6.2.1 Branch Strategy on Blocks

As we discussed in Chapter 4, values of zero for the branching binary variables, i.e.  $\alpha$  and  $\beta$ , do not select a clear block structure. Therefore, instead of branching on the binary variables directly, we branch on the logic decision to determine a block structure.

The block structure decision can be broken into two steps: what is the block size partition  $s$ , and which controlled variables and manipulated variables are allocated in each block. These two decisions are explored by two tree structures.

In the first step, we enumerate all possible block size partitions  $s$  with respect to the maximum number of the blocks  $q$  and the maximum block size  $z$ . At this step we determine how many controllers we must consider and how big each controller is. Since there is no one size distribution that is inherently better than others, we enumerate all possible block size distributions and compare their dynamic performances.

$$s = \left\{ (s_1, \dots, s_b) \mid \sum_{i=1}^b s_i = n, \quad z \geq s_1 \geq s_2 \geq \dots \geq s_b \geq 1, \quad b \leq q \right\} \quad (6.14)$$

Mathematically, we enumerate the partitions of a position integer  $n$  (6.14), which are finite non-increasing sequence of positive integers  $s_i$  sums up to  $n$ . The block size partition  $s$  can be enumerated by the tree structure in Figure 6.3, which shows the tree for  $n = 4$ ,  $q = 4$  and  $z = 4$ .

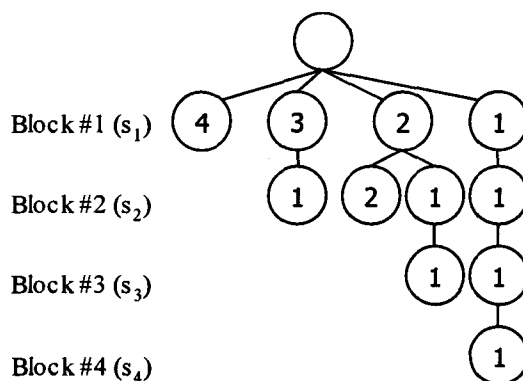


Figure 6.3 Decision tree for block size partition

In Figure 6.3 each tree level represents a block and each node represents a block size. The branching rules are the following:



- Each node branches all possible block sizes decreasingly.
- The lower level node must be not bigger than the upper node.
- Branching ends when the sum of all block sizes in that branch equals to the system dimension  $n$ .

Therefore, all possible block size partitions in Figure 6.3 are (4), (3,1), (2,2), (2,1,1), (1,1,1,1).

In the next step, for each block size partition, we find the controlled variable and manipulated variable allocation in each block that gives the best performance. This can be done in a modified branch & bound algorithm that is similar to the loop pairing problem in Chapter 4. The levels of the branch & bound tree now represent the blocks (controllers) and each node represents one possible controlled variable and manipulated variable allocation for this block. In another words, we enumerate all possible solutions of  $\alpha_{ib}$  and  $\beta_{bj}$  for  $b$ th block with respect to equation (6.11a~d) on  $b$ th level of the branch and bound tree.

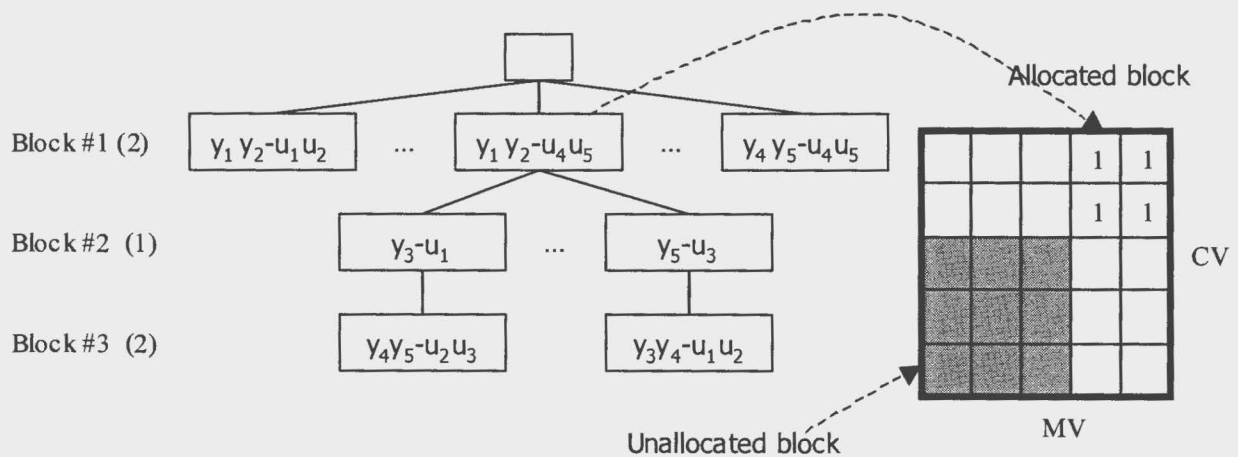


Figure 6.4 Branch and bound tree for block centralized structure

Figure 6.4 shows the branch and bound tree for 5x5 system with block size partition of (2, 1, 2). The text in each branch and bound tree node represents an allocation of controlled variables and manipulated variables. For instance,  $[y_1 y_2 - u_1 u_2]$  means that the 1<sup>st</sup> and 2<sup>nd</sup> controlled variables and the 1<sup>st</sup> and 2<sup>nd</sup> manipulated variables are allocated in block #1, which is a two-variable centralized MPC.

The loop pairing problem is a special case of the block centralized structure problem, which has only one possible block size partition, all block sizes are one. Each CV is allocated to a different block; therefore, the levels of the branch and bound tree directly relate to CV. The branch and bound tree in Figure 6.4 degenerates to the one in Figure 4.5 for loop pairing problem.

## 6.2.2 Generate Lower Bounds Using Control Knowledge

A fast calculation of the lower bound for the achievable performance is needed for the bounding procedure on intermediate nodes. We use the same idea for relaxation as for loop pairing problem in Chapter 4. At each intermediate node of the branch & bound tree some controlled variables and manipulated variable are allocated to controllers (blocks) while others are not (Figure 6.4). The controller equations for the unallocated manipulated variables are removed to relax the problem. Without the controller equations, the values of the manipulated variables can be determined by the optimization to improve the objective function in the feasible region. As a result, the transient of manipulated variables are calculated for these “relaxed” variables using open-loop dynamic optimization as part of the solution at each node.

The relaxed problem we solve at each intermediate node is equation (6.15).

$$\min_{\mathbf{R}, \mathbf{u}_k^{unalloc}(t)} \sum_k \sum_{t=1}^{tf} [\mathbf{e}_k(t)^T \mathbf{Q} \mathbf{e}_k(t)] \quad (6.15a)$$

s.t.

$$\begin{aligned} \mathbf{x}_k(t+1) &= \mathbf{A}_k \mathbf{x}_k(t) + \mathbf{B}_k \mathbf{u}_k(t) + \mathbf{W}_k \mathbf{d}(t) \\ \mathbf{y}_k(t) &= \mathbf{C}_k \mathbf{x}_k(t) + \mathbf{V}_k \mathbf{d}(t) + \mathbf{N}(t) \end{aligned} \quad (6.15b)$$

$$\begin{aligned} \mathbf{e}_k(t) &= \mathbf{sp}(t) - \mathbf{y}_k(t) \\ \mathbf{u}_k(t) &= \begin{bmatrix} \mathbf{u}_k^{alloc}(t) \\ \mathbf{u}_k^{unalloc}(t) \end{bmatrix} \quad \mathbf{y}_k(t) = \begin{bmatrix} \mathbf{y}_k^{alloc}(t) \\ \mathbf{y}_k^{unalloc}(t) \end{bmatrix} \end{aligned} \quad (6.15c)$$

$$\begin{aligned} \Delta \mathbf{u}_k^{p, alloc}(t) &= \mathbf{K}^{DMC} \cdot \left( \mathbf{y}^{sp, alloc}(t) - \mathbf{A}^{f, block} \Delta \mathbf{u}_k^{c, alloc}(t) - \mathbf{d}_k^{p, alloc}(t) \right) \\ \mathbf{K}^{DMC} &= \left[ (\mathbf{A}^{p, block})^T \mathbf{Q} (\mathbf{A}^{p, block}) + \mathbf{R} \right]^{-1} (\mathbf{A}^{p, block})^T \mathbf{Q} \end{aligned} \quad (6.15d)$$

$$\begin{aligned} \mathbf{u}_k^{alloc}(t) &= \mathbf{u}_k^{alloc}(t-1) + \Delta \mathbf{u}_k^{alloc}(t) \\ \mathbf{u}^L(t) &\leq \mathbf{u}_k^{unalloc}(t) \leq \mathbf{u}^U(t) \end{aligned} \quad (6.15e)$$

The controller variables and manipulated variables are divided into two sets in (6.15c). For controlled variables and manipulated variables that are allocated to blocks, i.e.  $\mathbf{u}_k^{alloc}(t)$  and  $\mathbf{y}_k^{alloc}(t)$ , the block centralized MPC controller equation (6.15d) is used to enforce the feedback control laws. For controlled variables and manipulated variables that have not allocated to any block yet, i.e.  $\mathbf{u}_k^{unalloc}(t)$  and  $\mathbf{y}_k^{unalloc}(t)$ , only upper and lower bounds (6.15e) are enforced.

In order to simplify the relaxed problem, we also remove the complementarity constraints for the MV saturation for the manipulated variables defined in the allocated blocks; thus, the relaxed problem does not enforce manipulated variable bounds for allocated manipulated variables, a factor that contributes to the solution being a looser lower bound than preferred on performance.

Problem (6.15) has two sets of decision variables, i.e. the unallocated manipulated variables  $\mathbf{u}_k^{unalloc}(t)$  and the move suppression factor  $\mathbf{R}$ . For any given  $\mathbf{R}$  problem (6.15) is a simple QP. We can view the optimum of problem (6.15) as a function of  $\mathbf{R}$ . Therefore, problem (6.15) is solved with  $\mathbf{R}$  as the optimization degree of freedom, which is a non-convex optimization problem.

Since it is non-convex, we solve problem (6.15) with multiple initial guesses. We can pick a fast and a slow move suppression factor for each manipulated variable based on process knowledge; then, we use all possible combinations of fast and slow move suppression factors as initial guesses of  $\mathbf{R}$ . For large size problem we can use several known operating points as the initial guesses of  $\mathbf{R}$ . In this way, we are likely to find a global solution; however, we cannot guarantee global optimality.

MATLAB optimization toolbox functions *fmincon()* and *fminsearch()* are used to find the optimal  $\mathbf{R}$  (for the “newest” block, i.e., the block added to the structural decisions at this node). *fmincon()* uses a gradient-based local NLP algorithm. The gradient is generated using finite difference, which sometimes is not very reliable. *fminsearch()* uses gradient-free Nelder-Mead algorithm; however the convergence is slow. In order to overcome the gradient issue and achieve reasonable solving speed, we iterate between *fmincon()* and *fminsearch()*, i.e. the problem is solved by *fmincon()* first;

then the solution is used as the initial guess and the problem is solved again by *fminsearch()*; if the two solutions are close enough then the optimum is found, otherwise, the problem is solved again by *fmincon()* with initial guess from *fminsearch()* solution.

### 6.2.3 Integrate Transients with Shortcut Metrics

Similarly to Chapter 4, the shortcut metrics can be integrated into the solving strategy the same way as in the loop pairing problem. We evaluate these shortcut metrics as constraints (necessary conditions) before solving the lower bound problem. If any of these shortcut metrics fails, we can prune the branch without solving the optimization problem.

There are some shortcut metrics can be applied to block centralized structure. We use Block Relative Gain (BRG) to test integrity of block centralized structure. The Block Relative Gain is discussed in Appendix B, and a brief introduction of BRG is given here. BRG is the extension of the concept of the relative gain array (RGA) to block centralized structure. Manousiouthakis, *et al.* (1986) defined BRG as the ratio of the open-loop block gain matrix and the closed-loop block gain matrix with all other variables are perfectly controlled. BRG measures the interaction among the blocks (multivariable controllers). The properties of RBG used in this research is:  $\det(\text{BRG}) > 0$  is the necessary condition for *integral controllable with integrity* (ICI) (Kariwala *et al.*, 2003). (However, the system not being ICI can also have  $\det(\text{BRG}) > 0$ .) We require all acceptable design candidates to have a positive determinant of its BRG.

## 6.3 Case Study

The fired heater case study has been introduced in Chapter 4. In this chapter it is used to demonstrate the procedure of selecting a block centralized control structure.

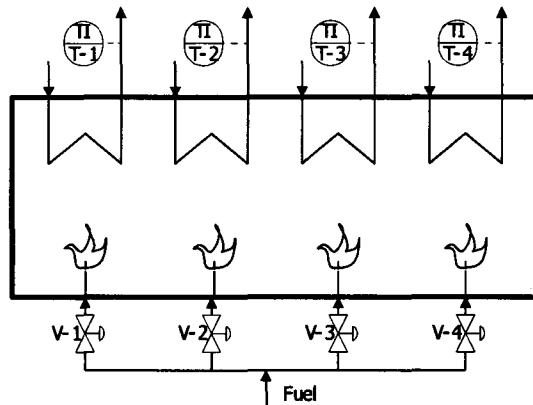


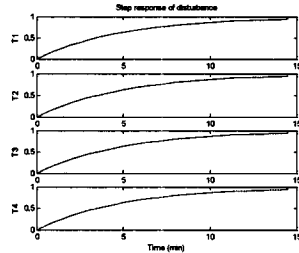
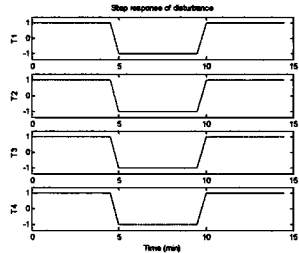
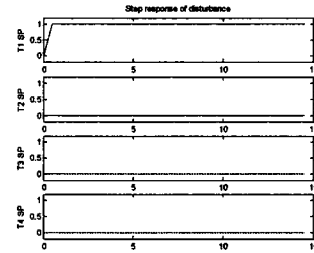
Figure 6.5 Fired Heater

We choose two parameters, the maximum number of the blocks and the maximum block size, both equal to the system dimension, four. Therefore, the block centralized structure can vary from diagonal structure (loop pairing) to fully centralized structure. The resulting structure is the best of all possible block structures.

The dynamic transient is evaluated under realistic scenario settings with:

- 10% measurement noise added for all four temperatures.
- Model uncertainty addressed by a second mismatched model with 50% gain increase of all process transfer functions and 0.5 minute delay added for each inputs.
- Rosenbrock (1974) did not specify the engineering unit for all variables, nor variable limits. Therefore, we treat all variables as dimensionless deviation variables in this case study. All manipulated variables are limited within  $[-1, +1]$  to avoid extreme behavior.
- Three different disturbances (Table 6.1) are considered. Disturbance #1 simulates slow fuel composition drift, disturbance #2 mimics switching the fuel types, and disturbance #3 is the temperature setpoint change of coil number 1.

Table 6.1 Testing disturbances

| Disturbance #1  | Disturbance #2  | Disturbance #3   |
|---|---|--|
| Slow change disturbance in fuel composition                                       | Fast change disturbance in fuel composition                                       | Temperature setpoint change for coil 1   |
|  |  |  |

Integrity is included in the design objective, which requires the selected control structures to have positive determinant of BRG for all blocks. Requiring positive determinant of BRG also speeds up the solving process significantly as shown in the following section.

6.3.1 Solving Process

The design problem is solved three times with respect to three different disturbances. The solving process is the same for three different disturbances. The discussion in this section is based on disturbance #1.

As discussed in Section 6.2.1 we first enumerate all possible block size partitions in Figure 6.6.

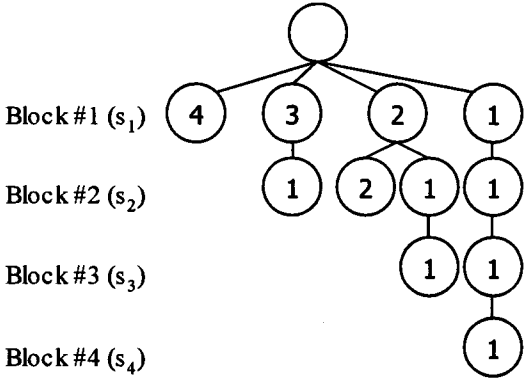


Figure 6.6Block size partitions for 4×4 system

There are five possible block size partitions for a  $4 \times 4$  system, which are (4), (3,1), (2,2), (2,1,1) and (1,1,1,1). For each block size partition, the best variable allocation is found by a modified branch and bound strategy. All of these are evaluated in the solving procedure. We will discuss the block size partition (2,1,1) as an example to show the procedure.

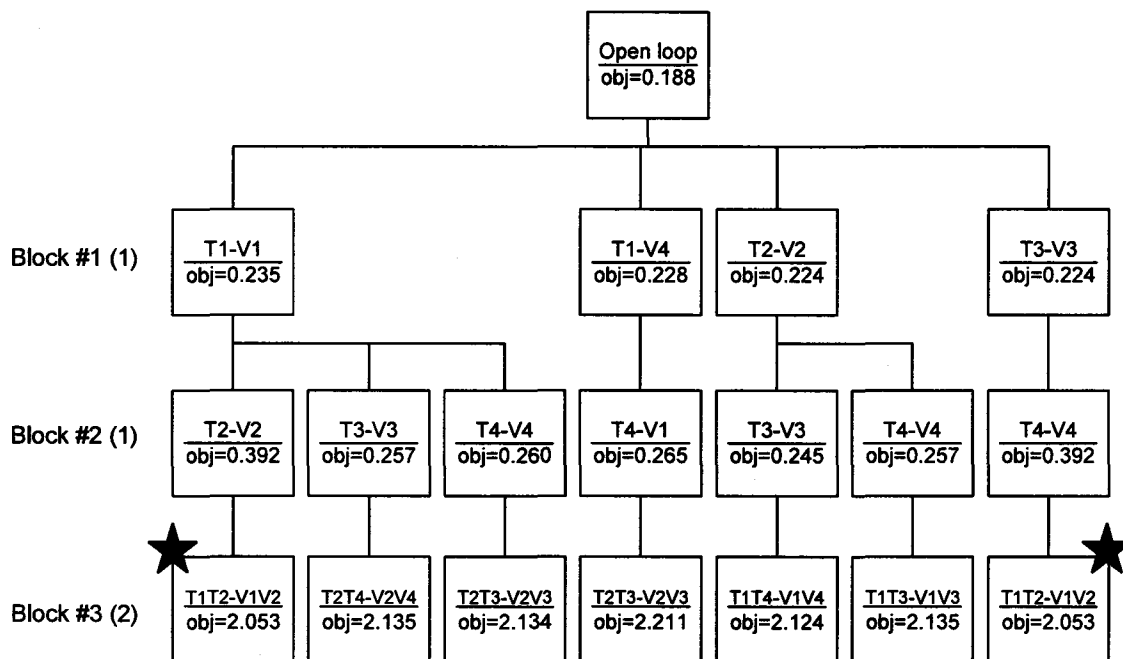


Figure 6.7 Variable allocation tree for block size partition (2,1,1)

Figure 6.7 is the branch and bound tree used to select variable allocation for block size partition (2,1,1). The best variable allocations are ([T1-V1], [T2-V2], [T3-T4-V3-V4]) and ([T1-T2-V1-V2], [T3-V3], [T4-T4]). Due to the space limitation the nodes eliminated by position determine of BRG requirement are not shown in Figure 6.7, but summarized in Table 6.2.

Table 6.2 The nodes with negative determinant of BRG in block size partition (2,1,1)

| Block | # of possible nodes | # of nodes with negative determinant of BRG | % of nodes with negative determinant of BRG |
|-------|---------------------|---|---|
| #1    | 16                  | 12  | 75%   |
| #2    | 36                  | 25  | 69%   |
| #3    | 11                  | 4   | 36%   |

We can eliminate 75% of the nodes on the first level of the branch and bound tree for Block #1, 69% of the nodes on the second level of the branch and bound tree for Block #2, and 36% of the nodes on the third level of the branch and bound tree by testing the determinant of BRG without evaluating dynamic transient. The calculation of the determinant of BRG is much faster than evaluating dynamic transient, therefore, we can solve the structure design problem in relatively short time. The total solve time is listed in Table 6.3.

Table 6.3 Total solving time for the block centralized design of the fired heater

|                                |           | Solving Time  |
|--------------------------------|-----------|---------------|
| Enumerate block size partition |           | $\cong 0$ sec |
| Variable allocation            | (4)       | 213 sec       |
|                                | (3,1)     | 1381 sec      |
|                                | (2,2)     | 1559sec       |
|                                | (2,1,1)   | 1821 sec      |
|                                | (1,1,1,1) | 724 sec       |
| Total solving time             |           | 1.6 hr        |

### 6.3.2 Results Analysis

The design problem has been solved three times with respect to three different disturbances. Table 6.4 lists the best block structure for five possible block size partitions for each disturbance, where **R** is the move suppression factor giving the minimum objective value.



Table 6.4a Block centralized structures for the three disturbances

| Block size partition | Variable allocation                 |                       | Disturbance #1                          |
|----------------------|-------------------------------------|-----------------------|---|
| (4)                  | [T1T2T3T4-<br>V1V2V3V4]             | <b>R</b><br>Objective | diag(0.678,0.828,0.828, 0.678)<br>1.906 |
| (3,1)                | [T1T2T3-V1V2V3],<br>[T4-V4]         | <b>R</b><br>Objective | diag(0.655,0.833,0.916, 0.996)<br>1.993 |
| (2,2)                | [T1T2-V1V2],<br>[T3T4-V3V4]         | <b>R</b><br>Objective | diag(0.699,1.019,1.019,0.699)<br>1.983  |
| (2,1,1)              | [T1-V1],[T2-V2],<br>[T3T4-V3V4]     | <b>R</b><br>Objective | diag(0.935,1.650,0.859,0.582)<br>2.053  |
| (1,1,1,1)            | [T1-V1],[T2-V2],<br>[T3-V3],[T4-V4] | <b>R</b><br>Objective | diag(0.589,2.079,2.079, 0.589)<br>2.204 |

Table 6.4b Block centralized structures for the three disturbances (cont.)

| Block size partition | Variable allocation                 |                       | Disturbance #2                            |
|----------------------|-------------------------------------|-----------------------|---|
| (4)                  | [T1T2T3T4-<br>V1V2V3V4]             | <b>R</b><br>Objective | diag(0.478,0.573,0.573,0.478)<br>124.642  |
| (3,1)                | [T1T2T3-V1V2V3],<br>[T4-V4]         | <b>R</b><br>Objective | diag(0.495,0.638,0.709, 0.506)<br>125.123 |
| (2,2)                | [T1T2-V1V2],<br>[T3T4-V3V4]         | <b>R</b><br>Objective | diag(0.505,0.654,0.654,0.505)<br>125.145  |
| (2,1,1)              | [T1-V1],[T2-V2],<br>[T3T4-V3V4]     | <b>R</b><br>Objective | diag(1.329,0.434,0.661,0.549)<br>126.052  |
| (1,1,1,1)            | [T1-V1],[T2-V2],<br>[T3-V3],[T4-V4] | <b>R</b><br>Objective | diag(0.311,70.64,70.89,0.311)<br>111.304  |

Table 6.4c Block centralized structures for the three disturbances (cont.)

| Block size partition | Variable allocation                 |                       | Disturbance #3                          |
|----------------------|-------------------------------------|-----------------------|---|
| (4)                  | [T1T2T3T4-<br>V1V2V3V4]             | <b>R</b><br>Objective | diag(0.733,1.130,1.274,1.700)<br>7.687  |
| (3,1)                | [T1T2T3-V1V2V3],<br>[T4-V4]         | <b>R</b><br>Objective | diag(0.749,1.261,2.285,0.798)<br>7.754  |
| (2,2)                | [T1T2-V1V2],<br>[T3T4-V3V4]         | <b>R</b><br>Objective | diag(0.859,1.450,0.513, 0.966)<br>7.956 |
| (2,1,1)              | [T1-V1],[T2-V2],<br>[T3T4-V3V4]     | <b>R</b><br>Objective | diag(1.107,0.489,1.938,2.753)<br>8.840  |
| (1,1,1,1)            | [T1-V1],[T2-V2],<br>[T3-V3],[T4-V4] | <b>R</b><br>Objective | diag(1.101,0.469,3.470,3.064)<br>8.921  |

The gray rows in Table 6.4 mark the best block centralized structure for different disturbances. The results show that the best structure changes with the disturbance. For the single setpoint change (#3) and slow disturbance (#1) fully centralized structure gives the best performance. But for a fast disturbance (#2), diagonal structure is the best.

If we do not consider model mismatch, the best controller is the fully centralized controller with zero move suppression factor, which inverses the process model and cancel all the interaction among variables. However, this inverse based controller is very sensitive to model mismatch. We must increase the move suppression factor to achieve certain level of robustness. The block centralized controller cannot completely cancel all the interaction and the closed loop system exhibits certain directionality. If the directionality of the closed loop system aligns with the directionality of the disturbance, the remaining interaction among variables is favorable to compensate the disturbance effects. This is the reason why the diagonal structure outperforms the fully centralized structure for disturbance #2.

Disturbance #1 has very slow effect and does not exhibit strong directionality; therefore, the fully centralized structure is the best structure. Disturbance #3 (setpoint change) prefers no interaction among variables, i.e. all variables remain constant except the variable that has the setpoint change. Therefore, the fully centralized controller is the best choice.

Manousiouthakis *et al.* (1986) used the same fired heater as example to demonstrate the control structure design procedure using BRG. The conclusion they drew is the control structure ([T1T2-V1V2], [T3T4-V3V4]) has better dynamic performance than the more centralized structure ([T1T2T3-V1V2V3], [T4-V4]) for setpoint changes. However, our result shows the opposite. The different results are caused by different setpoint changes, i.e. Manousiouthakis *et al.* tested y4 setpoint change, but we test y1 setpoint change. The structure ([T1T2T3-V1V2V3], [V4-V4]) is more favorable for y1 setpoint change than y4 setpoint change because the 3x3 multivariable controller [T1T2T3-V1V2V3] cancels most of interaction between y1 and other controlled variables, therefore y1 setpoint change has less impact on other controlled variables. This

is another good example showing how the disturbance directionality affects the control performance and therefore, affects the choice of the controller structure.

The different values of the objective among the different structures are small for this case study. For instance, the objective value of the diagonal structure (2.204) is only 15% worse than the objective value of the fully centralized structure (1.906) for disturbance #1 (Table 6.4a). The engineers can use process knowledge to tradeoff the control system complexity with the performance loss and select the simplest controller with acceptable performance.

Figures 6.8-6.10 are the close-loop simulation of the best block centralized structure with the nominal mode. The measurement noises are not included in the simulation in order to clearly illustrate the directionality of the responses.

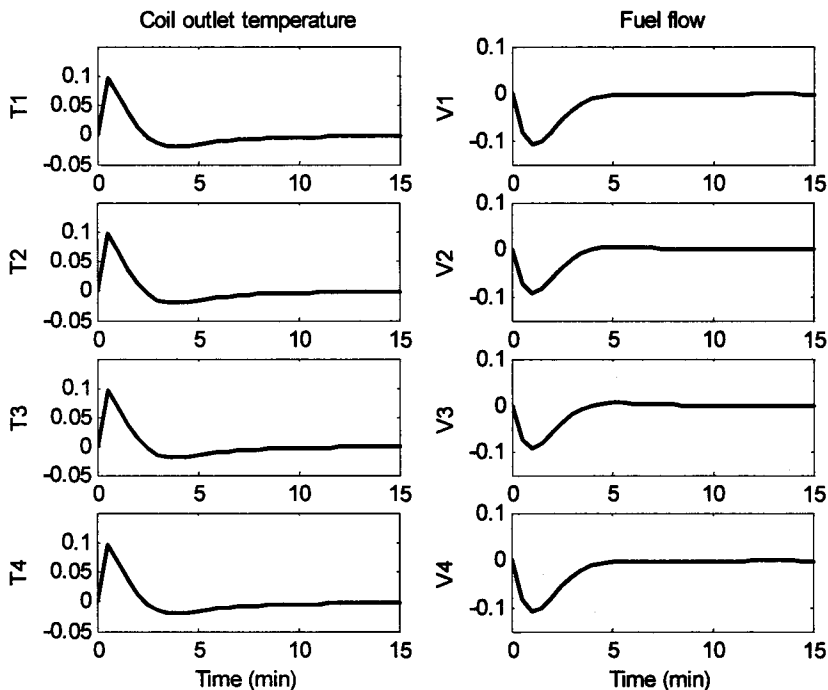


Figure 6.8 Dynamic response of disturbance #1 with fully centralized structure

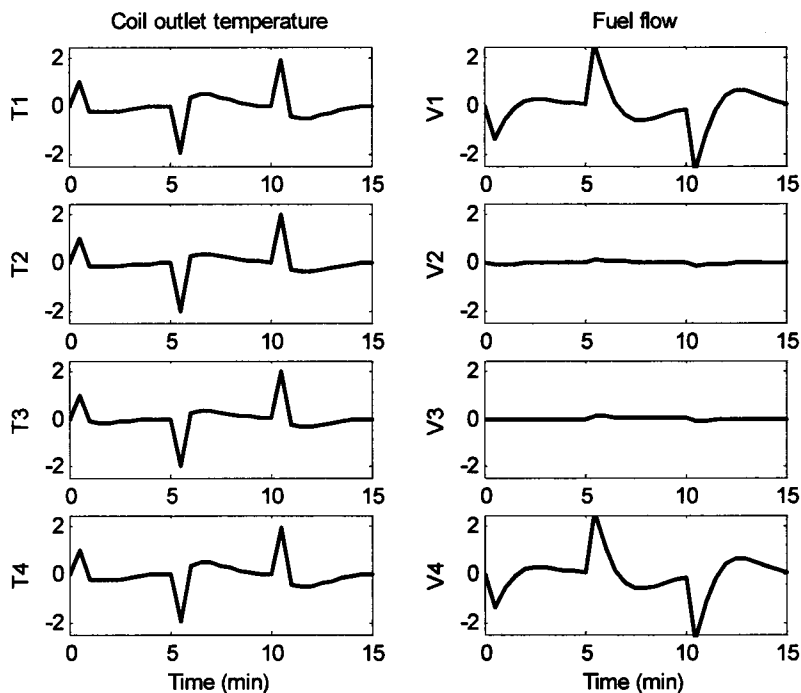


Figure 6.9 Dynamic response of disturbance #2 with diagonal structure

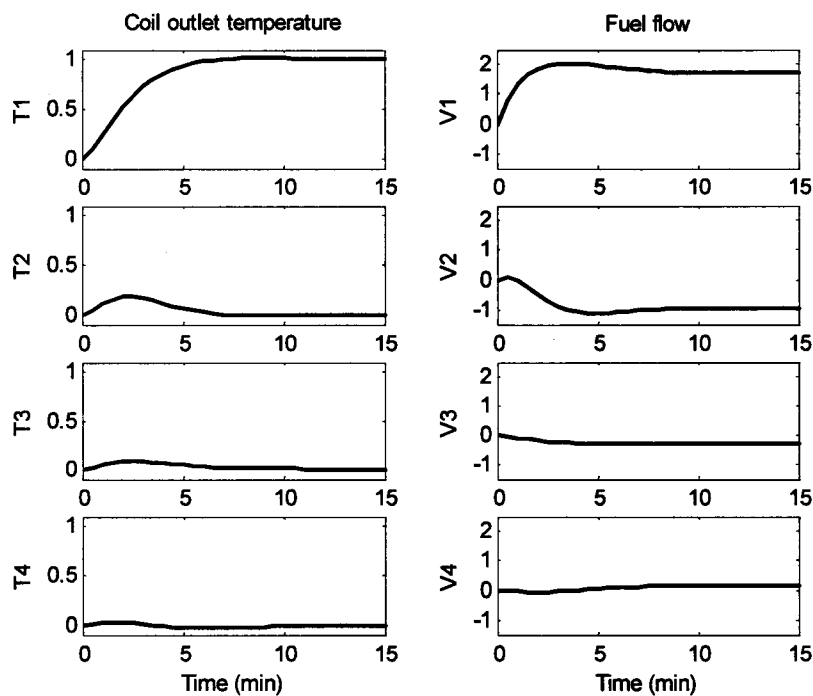


Figure 6.10 Dynamic response of disturbance #3 with fully centralized structure

### 6.3.3 Modified Process

For an additional study, we modify the plant by building an insulating wall between burners #2 and #3, which separates the fire box into two independent parts. Figure 6.11 shows the modification of the fired heater. The 4x4 plant model (6.16) now has a block centralized structure because there is no interaction between coils in the different fire boxes.

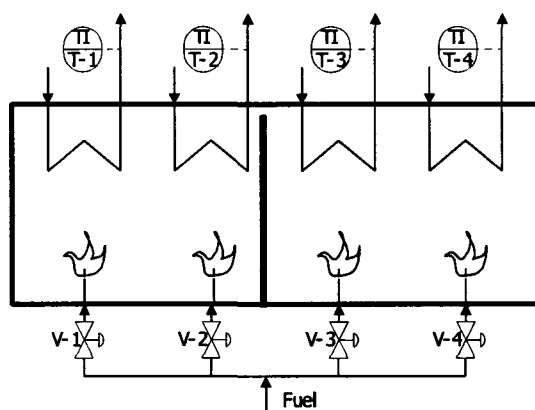


Figure 6.11 Modified fired heater

$$G(s) = \begin{bmatrix} \frac{1}{1+4s} & \frac{0.7}{1+5s} & 0 & 0 \\ \frac{0.6}{1+5s} & \frac{1}{1+4s} & 0 & 0 \\ 0 & 0 & \frac{1}{1+4s} & \frac{0.6}{1+5s} \\ 0 & 0 & \frac{0.7}{1+5s} & \frac{1}{1+4s} \end{bmatrix} \quad (6.16)$$

We solved the same design problem with the modified plant model, and the results are listed in Table 6.5.

Table 6.5a Block centralized structures for the three disturbances

| Block size partition | Variable allocation                 |                       | Disturbance #1                         |
|----------------------|-------------------------------------|-----------------------|--|
| (4)                  | [T1T2T3T4-V1V2V3V4]                 | <b>R</b><br>Objective | diag(0.579,0.544,0.544,0.579)<br>1.897 |
| (3,1)                | [T1T2T3-V1V2V3],<br>[T4-V4]         | <b>R</b><br>Objective | diag(0.579,0.544,0.540,0.850)<br>1.964 |
| (2,2)                | [T1T2-V1V2],<br>[T3T4-V3V4]         | <b>R</b><br>Objective | diag(0.579,0.544,0.544,0.579)<br>1.897 |
| (2,1,1)              | [T1-V1],[T2-V2],<br>[T3T4-V3V4]     | <b>R</b><br>Objective | diag(0.850,0.540,0.544,0.579)<br>1.964 |
| (1,1,1,1)            | [T1-V1],[T2-V2],<br>[T3-V3],[T4-V4] | <b>R</b><br>Objective | diag(0.851,0.540,0.540,0.851)<br>2.032 |

Table 6.5b Block centralized structures for the three disturbances (cont.)

| Block size partition | Variable allocation                 |                       | Disturbance #2                           |
|----------------------|-------------------------------------|-----------------------|--|
| (4)                  | [T1T2T3T4-V1V2V3V4]                 | <b>R</b><br>Objective | diag(0.406,0.383,0.383,0.406)<br>124.716 |
| (3,1)                | [T1T2T3-V1V2V3],<br>[T4-V4]         | <b>R</b><br>Objective | diag(0.406,0.382,0.380,0.514)<br>125.325 |
| (2,2)                | [T1T2-V1V2],<br>[T3T4-V3V4]         | <b>R</b><br>Objective | diag(0.406,0.383,0.383,0.406)<br>124.716 |
| (2,1,1)              | [T1-V1],[T2-V2],<br>[T3T4-V3V4]     | <b>R</b><br>Objective | diag(0.514,0.340,0.382,0.406)<br>125.325 |
| (1,1,1,1)            | [T1-V1],[T2-V2],<br>[T3-V3],[T4-V4] | <b>R</b><br>Objective | diag(0.514,0.398,0.398,0.514)<br>125.94  |

Table 6.5c Block centralized structures for the three disturbances (cont.)

| Block size partition | Variable allocation                 |                       | Disturbance #3                         |
|----------------------|-------------------------------------|-----------------------|--|
| (4)                  | [T1T2T3T4-V1V2V3V4]                 | <b>R</b><br>Objective | diag(0.717,1.045,2.083,1.933)<br>7.545 |
| (3,1)                | [T1T2T3-V1V2V3],<br>[T4-V4]         | <b>R</b><br>Objective | diag(0.717,1.045,2.083,2.083)<br>7.545 |
| (2,2)                | [T1T2-V1V2],<br>[T3T4-V3V4]         | <b>R</b><br>Objective | diag(0.717,1.045,2.083,1.933)<br>7.545 |
| (2,1,1)              | [T1-V1],[T2-V2],<br>[T3T4-V3V4]     | <b>R</b><br>Objective | diag(0.985,0.481,2.034,2.034)<br>8.065 |
| (1,1,1,1)            | [T1-V1],[T2-V2],<br>[T3-V3],[T4-V4] | <b>R</b><br>Objective | diag(0.985,0.481,2.034,2.034)<br>8.065 |

Gray rows in Table 6.5 mark the best block centralized structure for different disturbances. The results show the fully centralized structure always gives the best performance. We note that the two designs, the fully centralized and the 2x2 block structure ([T1T2-V1V2], [T3T4-V3V4]), have exactly the same move suppression factor and the same objective value no matter the disturbances. This is because the plant structure is identical to control structure ([T1T2-V1V2], [T3T4-V3V4]). Therefore, the “ideal” plant-wide control structure is the inherent plant structure, which is not necessary always the fully centralized structure. For example, most of chemical plants have several processing units in serial connected by few intermediate flows, which can be abstracted as block centralized structure.

Figures 6.12-6.14 are the close-loop simulations for the structure ([T1T2-V1V1], [T3T4-V3V4]) with nominal plant model. The measurement noises are not included in the simulation in order to clearly illustrate the directionality of the responses.

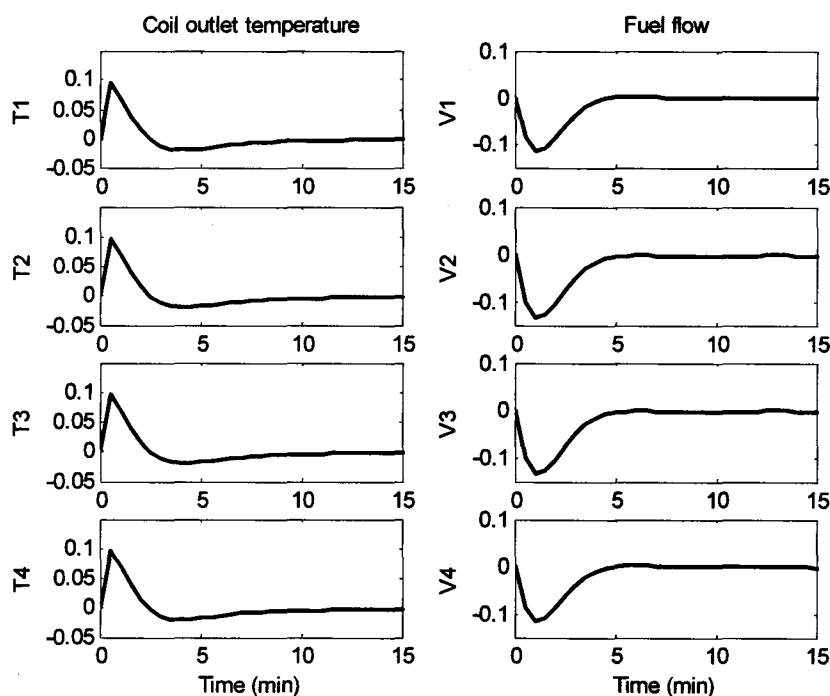


Figure 6.12 Dynamic response of disturbance #1 with structure ([T1T2-V1V1],[T3T4-V3V4]) for modified process

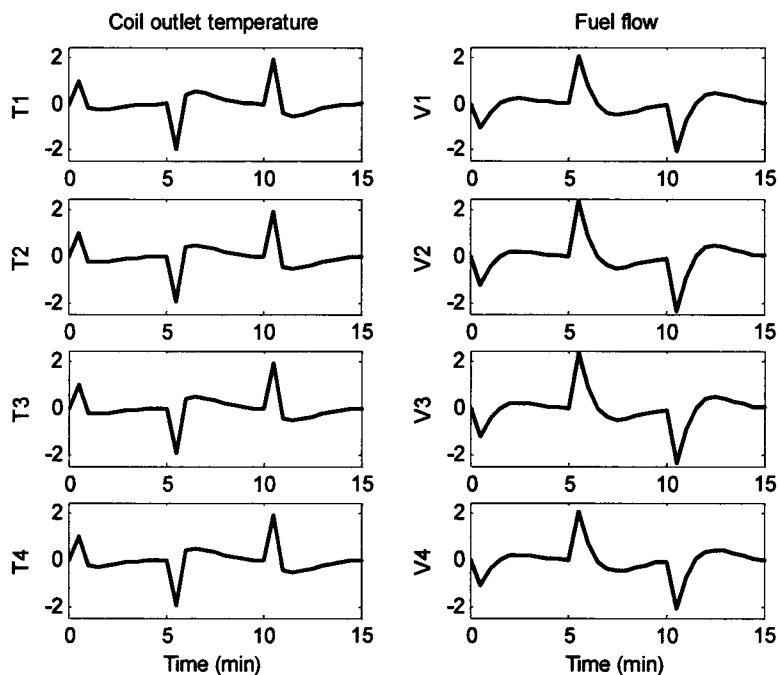


Figure 6.13 Dynamic response of disturbance #2 with structure ([T1T2-V1V1],[T3T4-V3V4]) for modified process

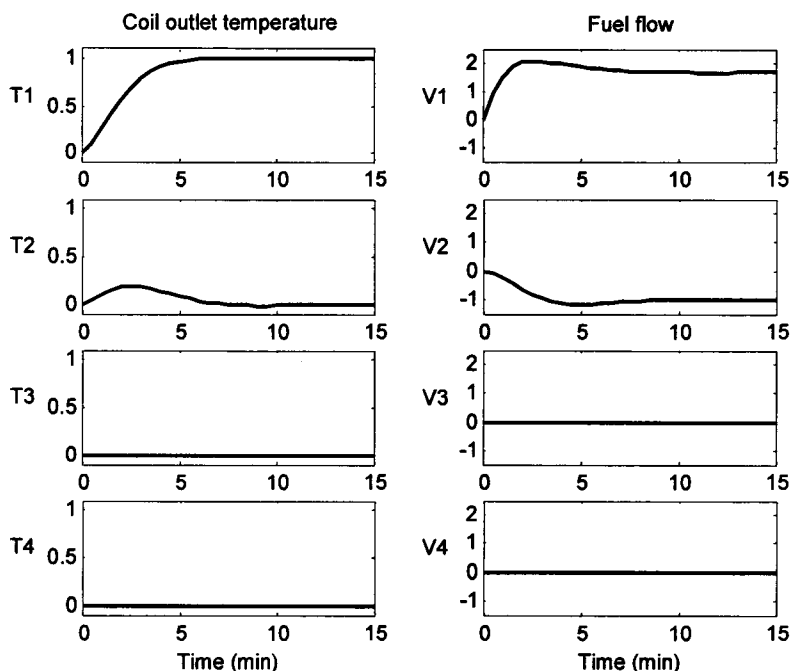


Figure 6.14 Dynamic response of disturbance #3 with structure ([T1T2-V1V1],[T3T4-V3V4]) for modified process



## 6.4 Summary

In this chapter, the control structure design methodology is extended to handle the block centralized structure. First, the unconstrained block centralized MPC equation is derived as the controller equation in the optimization formulation. We find all of the MPC controllers in the design problem can be represented by one MPC controller with a structured model. Then, a set of integer constraints is developed to enforce the block centralized structure on the controller model.

A two-step solving process is developed to solve this complex MINLP. At step one, all possible block size partitions are enumerated; then for each block size partition, the best variable allocation for each block is found by a modified branch and bound method (similar to Chapter 4) at step two. Comparing all the results gives us the best block centralized structure.

The complexity of the block centralized structure design problem increases significantly from the loop pairing problem presented in Chapter 3, 4 and 5. Table 6.6 shows the differences of the number of possible control structures and solving time between the loop pairing problem and the block centralized structure design problem for fired heater case study.

Table 6.6 Comparison of loop pairing problem and block centralized structure problem for fired heater case study

|                             | # of possible structures | Solving time |
|-----------------------------|--------------------------|--------------|
| Loop pairing                | 24                       | 1 min        |
| Block centralized structure | 77                       | 1.6 hr       |

There are two factors that make the block centralize structure design problem difficult to solve. First, the block centralize structure allows much more possible structures, while loop pairing is just a subset of it. The formulation has to have more complex integer constraints to represent the block centralize structure. Second, MIMO controller, such as MPC, is required to enforce feedback control in the block centralize

structure design problem. Optimal tuning of MIMO controller is much more difficult than optimal single loop tuning used in loop pairing. We have to simplify the formulation by using linearized plant model, and unconstrained DMC with output clamping to address saturation behavior. The non-convex optimal tuning problem is solved by local NLP solver with multiple starting points. We note that the global optimality of the result is not guaranteed.

The fired heater case study shows:

- Integrating shortcut metrics as design objective can significantly reduce the number of possible structures and reduce the solving time.
- The best control structure depends on the disturbance directionality. The fully-centralized structure is not always the best choice.
- The “ideal” plant-wide control structure is the inherent plant structure, which can be the block centralized structure matches the plant structure.

## **Chapter 7 Conclusion**

This chapter summarizes the work presented in this thesis, reiterates the important insights arising out of this work, and suggests possible areas for future research.

### **7.1 Summary**

This research focuses on the selection of control structure based on the closed-loop dynamic performance. Additional design criterion, such as RGA, are integrated into one rigorous mathematic formulation so we can compare and trade off different control structure designs with respect to multiple control objectives systematically. A novel solving strategy is developed to handle this difficult mathematic problem by utilizing control concepts to guide the branching directions and generate sound lower bounds with low computing demands. We can solve a control structure design problem with respect to realistic scenario settings for medium size systems (5x5) in a reasonable time (less than one hour).

In Chapter 3, we developed a mathematic formulation that includes all important control objectives, such as robustness (by multiple process models), noise, saturation (by complementarity constraints), equipment capacity (by variable bounds) and optional design criteria (e.g. RGA for integrity), into one multiple objective optimization problem. The goal is to select the best dynamic control system structure based on a realistic

scenario. A small example demonstrated that the incomplete scenario definition led to an incorrect design.

The mathematic formulation we developed is a challenging MINLP. Several off-the-shelf NLP, MINLP and global solvers are tested on a medium size control structure design problem; however, none of these can solve it in a reasonable time. Therefore, in Chapter 4 we developed a specialized solving method for the loop pairing problem. The solving method is a specialized branch and bound algorithm that integrates with control knowledge and uses process insight to guide the searching process. The controllability is checked at the beginning to verify the existence of a satisfactory solution. The transition responses are evaluated only when an integer variable is set to one, which generates block structures that have clear physical meaning. The lower bound of the objective value is derived from the open-loop controller concept based on the block structures. The tuning grid search is implemented at intermediate branch and bound nodes to explore large tuning ranges. All these tailored solving strategies speed up the solving procedure significantly.

The solving strategy for the loop pairing problem is applied to the Tennessee Eastman problem to demonstrate its ability to solve medium size industrial problems in Chapter 5. At first the base layer control loops are identified and are closed to reject common disturbances and ensure safety. Then a 5x5 loop pairing problem is solved using the proposed solving strategy. The algorithm finds a physically sound pairing with good performance in less than one hour compared with other solvers cannot find a solution in 5 days. A complexity study based on the 5x5 Tennessee Eastman pairing problem shows our methodology can handle 8x8 system with proper heuristic search parameters. We expect that even larger problems will be tractable because the percentage of good pairings needing evaluation will decrease with problem size for process plants.

The loop-pairing method is also extended to handle non-square systems. An example problem is solved to demonstrate the critical importance of evaluating the transient response when designing control systems. The method developed here performed much better than a previously published heuristic approach.

Chapter 6 extends the control structure design methodology to handle the block centralized structure, which represents the practical plant-wide control structure. A two-step solving strategy is developed to solve this complex MINLP. At step one, all possible block size partitions are enumerated; then for each block size partition, the best variable allocation of each block is found by a modified branch and bound method (similar to Chapter 4) at step two. Then, the best block structure is selected from all possible block size partitions. The case study demonstrates the best control structure changes with the disturbances considered in the scenario. Also, the case study shows a properly designed block structure, as simple as diagonal structure, can achieve most of the performance of a fully-centralized multivariable controller in some cases.

## **7.2 Contributions**

The main contributions of this research are:

- A rigorous mathematical formulation for control structure design problem that includes full closed-loop transient analysis with the additional integrity requirement. The multi-objective framework is extendable so that different control performance objectives can be easily added. Unique process requirements and engineer inputs can be taken into account as additional constraints.
- A novel solving strategy that makes this challenging MINLP computationally tractable. This is made possible by limiting the research scope to linear plant model and linear controller algorithms. A tailored Branch and Bound algorithm is developed to take advantage of the special problem structure by using control knowledge to generate valid lower bounds efficiently. There are heuristic parameters designed in the algorithm that can guide the solving process by using prior knowledge of the processes. A complexity study shows the methodology can solve up to an  $8 \times 8$  design problem. Considering the percentage of good pairings

needing evaluation will decrease with problem size even larger problems will be tractable.

- A general approach that addresses the common control structures in process industries, such as square and non-square SISO loop pairing using PID controller and block centralized structure using MPC. Various case studies demonstrate the usefulness of this research.

Some limitations of this research:

- The research scope does not include hybrid control systems (with continuous and discrete variables), or highly nonlinear processes or nonlinear controller algorithms.
- Although the results are uniformly good on problems addressed, the results can be certified to be only local optima. The methods developed here do not provide a guarantee of global optimality.
- This research does not explore the implementation details of decentralized control system, such as how to coordinate different controllers.

## **7.3 Future Works**

This thesis emphasizes the rigorous mathematical framework to solve the control structure design problem with respect to dynamic performance. The proposed framework has been demonstrated to solve medium size system efficiently. However, several research opportunities remains.

### **7.3.1 Global Optimal Tuning for Multiple-Loop and Multiple-MPC Controller**

Multiple-loop controller tuning problem is a non-convex problem with potentially many local solutions. We expect that the control structure is a more dominant factor than the controller parameter tunings on dynamic performance. Therefore, we use a coarse grid search to find correct parameter magnitude and sign to differentiate possible control

structures. The method developed here performed much better than a previously published heuristic approach. However, an example problem is solved to demonstrate the critical importance of evaluating the transient response when designing control systems.

An efficient global solving method for multiple loop controller tuning problem is needed to provide a global optimum of the control structure design methodology, also is crucial for the implementation of the designed control system.

Although it is a common practice to have multiple MPC coexist in one chemical plant, little research has been performed on the tuning of multiple, interacting MPC. In Chapter 6 we formulate the multiple MPC tuning problem as a non-convex optimization problem and iterate between gradient-based and gradient-free solver to avoid trapping in local solution. However, a global optimum is not guaranteed. More fundamental research is needed to understand the mechanism of how multiple MPC interact with each other and how to find a global solution efficiently.

### 7.3.2 Additional Decentralized Controller Algorithms

The controller algorithms considered in this research are conventional PI controller and MPC. Each controller is completely autonomic without any information exchange. The new decentralized MPC algorithm, such as agent-based algorithm, is an active research area. Venkat *et. al.* (2006) proved the stability and performance equivalency between fully-centralized MPC and decentralized MPC with full information exchange. Olvera (Decentralized MPC, Ph.D. Thesis in progress, McMaster University, Hamilton, Ontario, 2008) proposed a decentralized MPC algorithm with partial information exchange that does not require iterative solving.

The control structure design formulation proposed in this research can be modified to accommodate new MPC algorithms. However, the solving strategy needs to be modified to take into account for information exchanged among controllers.

### 7.3.3 Control Structure with Hierarchy

The plant-wide control system can be decomposed hierarchically into several layers. The SISO PID loops implemented in DCS, called base layer, is at the bottom of the control hierarchy. The advanced process control (APC) layer, which consists of MIMO MPC, is on top of the base layer. These MPC normally use the set points of the base layer loops as their manipulated variables. Clearly, The base layer control structure design has impact on APC layer design. The propose method can solve the loop pairing problem for the base layer or the block centralized structure problem for APC layer respectively. A new methodology that can solve base layer and APC layer control structure design problem simultaneously may potentially improve the overall control system performance.

### 7.3.4 Software Structure

The research code is developed in MATLAB that calls several solvers interfaced with GAMS and AMPL. MATLAB exchanges information with the solvers through disk files. Although the solving strategy is completely implemented using this software structure, the efficiency is low. Coding in C++ or FORTRAN language and calling solvers using their native C++ or FORTRAN interfaces could significantly improve the solving speed. All information exchanges could in computer memory, which is several orders of magnitude faster than disk file.

Another approach is to use more alternative optimization software, such as Xpress-Optimizer developed by Dash Optimization (Guéret, *et. al.*, 2002). The mixed integer solver in Xpress-Optimizer allows inserting custom callback functions into each branch and bound node, in which the solving strategy developed in this thesis could be implemented.

The design software can use an iterative approach. Since solving complementarity constraints for saturation is a challenge for general NLP solver, user may want to solve a simplified formulation, which does not include complementarity constraints. The



simplified formulation can be solved faster than the full formulation. If user is not satisfied with the result or the dynamic transient is unrealistic, he/she can solve the full formulation again to get more accurate result.

## Nomenclature

|            |   |
|------------|---|
| <b>a</b>   | Step response coefficient                             |
| <b>A</b>   | State coefficient matrix of linear state space model  |
|            | Dynamic matrix  |
| <b>B</b>   | Input coefficient matrix of linear state space model  |
| <b>C</b>   | Output coefficient matrix of linear state space model |
| <b>c</b>   | Complementarity constraints                           |
| <b>CV</b>  | Controlled variable                                   |
| <b>d</b>   | Disturbances vector of linear state space model       |
| <b>d</b>   | Number of disturbances                                |
| <b>DV</b>  | Disturbance variable                                  |
| <b>det</b> | Determinant   |
| <b>e</b>   | Control error vector                                  |
| <b>F</b>   | Flow  |
| <b>G</b>   | Process transfer function matrix                      |
| <b>g</b>   | Inequality constraints                                |
| <b>h</b>   | Equality constraints                                  |
| <b>J</b>   | Objective function                                    |
| <b>K</b>   | Controller transfer function matrix                   |

|                    |   |
|--------------------|---|
| $\mathbf{K}_C$     | Proportional gain matrix                              |
| $\mathbf{K}_I$     | Integral gain matrix                                  |
| $\mathbf{K}^{DMC}$ | Unconstrained DMC controller gain                     |
| $L$                | Level   |
| $M$                | Large positive number                                 |
| $MV$               | Manipulated variable                                  |
| $m$                | Number of manipulated variables                       |
|                    | Input horizon   |
| $\mathbf{N}$       | Output white noise vector of linear state space model |
| $n$                | Number of controlled variables                        |
| $n_c$              | Number of complementarity constraints                 |
| $n_g$              | Number of inequality constraints                      |
| $n_h$              | Number of equality constraints                        |
| $n_J$              | Number of objective functions                         |
| $n_k$              | Number of mismatch models                             |
| $n_l$              | Number of predetermined controller gains              |
| $p$                | Output horizon  |
| $PV$               | Process variable                                      |
| $\mathbf{Q}$       | Weighting matrix for controlled variables             |
| $q$                | Maximum number of blocks                              |
| $\mathbf{R}$       | Move suppression matrix                               |
| $r$                | Number of states of linear state space model          |
| $\mathbf{s}$       | Slack variable vector                                 |
| $\mathbf{sp}$      | Set point vector                                      |
| $T$                | Temperature   |
| $t$                | Time step   |
| $\Delta T$         | Controller execution time                             |
| $T_I$              | Integral time   |
| $tf$               | Time horizon  |

|              |  |
|--------------|--|
| $U$          | Utility function   |
| $\mathbf{u}$ | Manipulated variable vector                                      |
| $\mathbf{V}$ | Output disturbance coefficient matrix of liner state space model |
| $V$          | Valve  |
| $v$          | Number of predetermined gain values                              |
| $\mathbf{W}$ | State disturbance coefficient matrix of linear state space model |
| $w$          | Weighting of objective functions                                 |
| $X$          | Component concentration  |
| $\mathbf{x}$ | State vector of linear state space model                         |
|              | Optimization variable vector                                     |
| $\mathbf{y}$ | Controlled variable vector                                       |
|              | Optimization variable vector                                     |
| $\mathbf{z}$ | Optimization variable vector                                     |
| $z$          | Maximum block size   |

## Greek letters

|               |   |
|---------------|---|
| $\alpha$      | Binary matrix represents row block structure    |
| $\beta$       | Binary matrix represents column block structure |
| $\delta$      | Binary matrix represents control structure      |
| $\varepsilon$ | Detune factor matrix                            |
| $\phi$        | Objective function                              |
| $\Lambda$     | Relative gain matrix                            |
| $\lambda$     | Lagrange multiplier                             |
| $\lambda$     | Relative gain                                   |
| $\mu$         | Central path parameter                          |
| $\pi$         | Slack variable                                  |
| $\theta$      | Lagrange multiplier                             |
| $\rho$        | Lagrange multiplier                             |

$\psi$  Slack variable

## Subscripts

|               |                                    |
|---------------|------------------------------------|
| <i>air</i>    | Air                                |
| <i>b</i>      | Block                              |
| <i>block</i>  | Block centralized structure        |
| <i>cat</i>    | Catalyst                           |
| <i>reaCW</i>  | Reactor cooling water              |
| <i>condCW</i> | Condenser cooling water            |
| <i>d</i>      | Disturbance                        |
| <i>i</i>      | Index of Controlled variables      |
| <i>j</i>      | Index of manipulated variables     |
| <i>k</i>      | Index of mismatch models           |
| <i>l</i>      | Index of predetermined gain values |
| <i>q</i>      | Index of objective functions       |
| <i>rgn</i>    | Regenerator                        |
| <i>ris</i>    | Riser                              |

## Superscripts

|                |                                   |
|----------------|-----------------------------------|
| <i>alloc</i>   | Allocated variables               |
| <i>unalloc</i> | Have not been allocated variables |
| <i>c</i>       | Current                           |
| <i>f</i>       | Future                            |
| <i>L</i>       | Lower bound                       |
| <i>U</i>       | Upper bound                       |
| <i>p</i>       | Prediction                        |
| <i>sp</i>      | Set point                         |

## Reference

- Adjiman, C. S., I. P. Androulakis, and C. A. Floudas. "Global Optimization of Mixed-Integer Nonlinear Problems." *AIChE Journal* 46 (2000): 1769-1797.
- Arbel, A., Z. Huang, R. H. Rinard, and R. Shinnar. "Dynamics and Control of Fluidized Catalytic Crackers. 1. Modeling of the Current Generation of FCC's." *Industrial & Engineering Chemistry Research* 34 (1995): 1228-1243.
- Arbel, A., R. H. Rinard, and R. Shinnar. "Dynamics and Control of Fluidized Catalytic Crackers. 3. Designing the Control System: Choice of Manipulated and Measured Variables for Partial Control." *Industrial & Engineering Chemistry Research* 35 (1996): 2215-2233.
- Arkun, Y. and J. Downs. "A General Method to Calculate Input-Output Gains and the Relative Gain Array for Integrating Processes." *Computers & Chemical Engineering* 14 (1990): 1101-1110.
- Astrom, K. J., P. Albertos, and J. Quevedo. "PID Control." *Control Engineering Practice* 9, no. 11 (2001): 1159-1161.
- Astrom, K. J., and T. Hagglund. "Automatic Tuning of Simple Regulators with Specification on Phase and Amplitude Margins." *Automatica* 20 (1984): 645-651.

- Baker, R., and C. L. E. Swartz. "Rigorous Handling of Input Saturation in the Design of Dynamically Operable Plants." *Industrial and Engineering Chemistry Research* 43, no. 18 (2004): 5880-5887.
- Baker, Rhoda. "Interior Point Methods in Integrated Design and Control." Ph.D. diss., McMaster University, 2006.
- Bansal, V., V. Sakizlis, R. Ross, J. D. Perkins, and E. N. Pistikopoulos. "New Algorithms for Mixed-Integer Dynamic Optimization." *Computers & Chemical Engineering* 27 (2003): 647-668.
- Bennett, S. *A History of Control Engineering, 1930-1955*. Stevenage, Herts., U.K.: P. Peregrinus on behalf of the Institution of Electrical Engineers, London, 1993.
- Bristol, E. H. "On a New Measure of Interaction for Multivariable Process Control." *IEEE Transactions on Automatic Control* 11, no. 1 (1966): 133-134.
- Brogan, W. L. *Modern Control Theory*. 3rd Edition ed. Englewood Cliffs, N.J.: Prentice Hall, 1991.
- Buckley, P. S. *Techniques of Process Control*. New York: Wiley, 1964.
- Buckley, Page S. "Material Balance Control in Recycle Systems." *Instrumentation Technology* 21, no. 5 (1974): 29-34.
- Campo, P. J., and M. Morari. "Achievable Closed-Loop Properties of Systems Under Decentralized Control: Conditions Involving the Steady-State Gain." *IEEE Transactions on Automatic Control* 39, no. 5 (1994): 932-943.
- Cao, Yi, and Diane Rossiter. "Input Pre-Screening Technique for Control Structure Selection." *Computers & Chemical Engineering* 21, no. 6 (1997): 563-569.

- Chang, Jin Wen, and Cheng Ching Yu. "Relative Disturbance Gain Array." *AIChE Journal* 38, no. 4 (1992): 521-534.
- Chiu, Min Sen, and Yaman Arkun. "Decentralized Control Structure Selection Based on Integrity Considerations." *Industrial & Engineering Chemistry Research* 29, no. 3 (1990): 374-382.
- Cohen, G. H., and G. A. Coon. "Theoretical Considerations of Retarded Control." *Transactions of ASME* 75 (1953): 827.
- Cutler, C., and B. Ramaker. "Dynamic Matrix Control - A Computer Control Algorithm." *AIChE National Meeting* (1979).
- Decroocq, D. *Catalytic Cracking of Heavy Petroleum Fractions*. Houston: Gulf Publishing, 1984.
- Douglas, J. M. *Conceptual Design of Chemical Processes*. New York: McGraw-Hill, 1988.
- Downs, J. J. "Distillation Control in a Plantwide Control Environment." In *Practical Distillation Control*. Springer, 1992, Chapter 20.
- Downs, J. J., and E. F. Vogel. "Plant-Wide Industrial Process Control Problem." *Computers & Chemical Engineering* 17, no. 3 (1993): 245-255.
- Dukelow, S. G. *The Control of Boilers*. Research Triangle Park, N.C.: ISA Press, 1986.
- Duran, M. A., and I. E. Grossmann. "An Outer Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs." *Mathematical Programming* 36, no. 3 (1986): 307-339.



- Ferris, M. C. *MATLAB and GAMS: Interfacing Optimization and Visualization Software*. Computer Sciences Department, University of Wisconsin, Madison, Wisconsin: Mathematical Programming Technical Report, 1998.
- Fletcher, R., S. Leyffer, D. Ralph, and S. Scholtes. *Local Convergence of SQP Methods for Mathematical Programs with Equilibrium Constraints*. University of Dundee, 2002.
- Floudas, C. A., A. Aggarwal, and A. R. Ciric. "Global Optimum Search for Nonconvex NLP and MINLP Problems." *Computers & Chemical Engineering* 13, no. 10 (1989): 1117-1132.
- Foss, C. S. "Critique of Chemical Process Control Theory." *AIChE Journal* 19, no. 2 (1973): 209-214.
- Garcia, C. E., D. M. Prett and M. Morari. "Model Predictive Control: Theory and Practice – a Survey." *Automatica* 25, no. 3 (1989): 335-348.
- Gary, J. H., and G. E. Handwerk. *Petroleum Refining Technology and Economics*. New York: Marcel Dekker, Inc., 1984.
- Grosdidier, P., and M. Morari. "A Computer Aided Methodology for the Design of Decentralized Controllers." *Computers & Chemical Engineering* 11, no. 4 (1987): 423-433.
- Grosdidier, P., M. Morari, and B. R. Holt. "Closed-Loop Properties from Steady-State Gain Information." *Industrial & Engineering Chemistry, Fundamentals* 24, no. 2 (1985): 221-235.
- Guéret, Christelle, Christian Prins, and Marc Sevaux. *Applications of Optimization with Xpress-MP*. Dash Optimization Ltd., 2002.

- Harris, T. J., and J. F. MacGregor. "Design of Multivariable Linear-Quadratic Controllers using Transfer Functions." *AIChE Journal* 33, no. 9 (1987): 1481-1495.
- Horn, Roger, and Charles Johnson. *Topics in Matrix Analysis*. Cambridge: Cambridge University Press, 1994.
- Hovd, M. and S. Skogestad. "Pairing Criteria for Decentralized Control of Unstable Plant." *Industrial and Engineering Chemistry Research* 33, no. 9 (1994): 2134-2139.
- Kariwala, Vinay, J. Fraser Forbes, and Edward S. Meadows. "Block Relative Gain: Properties and Pairing Rules." *Industrial and Engineering Chemistry Research* 42, no. 20 (2003): 4564-4574.
- Kookos, I. K., and J. D. Perkins. "Regulatory Control Structure Selection of Linear Systems." *Computers & Chemical Engineering* 26, no. 6 (2002): 875-887.
- Kookos, I. K., and J. D. Perkins. "Heuristic-Based Mathematical Programming Framework for Control Structure Selection." *Industrial & Engineering Chemistry Research* 40 (2001): 2079-2088.
- Lee, J. H., and M. Morari. "Robust Control Structure Selection and Control System Design Methods Applied to Distillation Column Control." *Proceedings of the 29th Conference on Decision and Control* (1990): 2041-2046.
- Lipták, B. G. *Optimization of Industrial Unit Processes*. 2nd Edition ed. Boca Raton, Fla: CRC Press, 1999.
- Luyben, W. L. "Simple Method for Tuning SISO Controllers in Multivariable Systems." *Industrial & Engineering Chemistry Process Design and Development* 25 (1986): 654-660.
- Luyben, W. L., B. D. Tyreus, and M. L. Luyben. *Plantwide Process Control*. New York: McGraw-Hill, 1998.

- Luyben, William L. "Simple Regulatory Control of the Eastman Process." *Industrial & Engineering Chemistry Research* 35, no. 10 (1996): 3280-3289.
- Manousiouthakis, V., R. Savage, and Y. Arkun. "Synthesis of Decentralized Process Control Structures using the Concept of Block Relative Gain." *AIChE Journal* 32, no. 6 (1986): 991-1003.
- Marlin, T. E., and M. Young. "Integrating the Effects of Process Controllers with Steady-State, Equation-Based Simulation." *Chemical Engineering Communications* 165 (1998): 67-87.
- Marlin, T. E. *Process Control: Designing Process and Control System for Dynamic Performance*. New York: McGraw-Hill, 2000.
- Marlin, T. E. "Engineering Economics and Problem Solving (CHEM ENG 4N04) Course Note." Course Note, 2002.
- McAvoy, T. J., N. Ye, and C. Gang. "Improved Base Control for the Tennessee Eastman Problem." *Proceedings of the American Control Conference* 1 (1995): 240-244.
- McAvoy, T. *Interaction Analysis*. ISA Monograph, 1983.
- McAvoy, T., Y. Arkun, R. Chen, D. Robinson, and P. D. Schnelle. "A New Approach to Defining a Dynamic Relative Gain." *Preprints of 6th IFAC Symposium on Dynamic and Control of Process System* (2001): 481-485.
- McAvoy, T. J., and N. Ye. "Base Control for the Tennessee Eastman Problem." *Computers & Chemical Engineering* 18, no. 5 (1994): 383-413.
- McAvoy, Thomas J. "Synthesis of Plantwide Control Systems using Optimization." *Industrial and Engineering Chemistry Research* 38, no. 8 (1999): 2984-2994.

- "Methodology for Screening Level Control Structures in Plantwide Control Systems." *Computers & Chemical Engineering* 22, no. 11 (1998): 1543-1552.
- Morari, M., and J. H. Lee. "Model Predictive Control: Past, Present and Future." *Computers and Chemical Engineering* 23, no. 4-5 (1999): 667-682.
- Morari, M., S. Skogestad, and D. E. Rivera. "Implications of Internal Model Control for PID Controllers." *Proceedings of American Control Conference* (1984): 661.
- Nett, C. N. "Decentralized Control System Design for a Variable-Cycle Gas Turbine Engine." *Proceedings of the IEEE Conference on Decision and Control Including the Symposium on Adaptive Processes* 2 (1989): 1301.
- Nett, C. N., and V. Manousiouthakis. "Euclidean Condition and Block Relative Gain: Connections, Conjectures, and Clarifications." *IEEE Transactions on Automatic Control* 32, no. 5 (1987): 405-407.
- Neumaier, A., O. Shcherbina, W. Huyer, and T. Vinko. "A Comparison of Complete Global Optimization Solvers." *Mathematical Programming* 103, no. 2 (2005): 335-356.
- Niederlinski, A. "A Heuristic Approach to the Design of Linear Multivariable Interacting Control Systems." *Automatica* 7, no. 6 (1971): 691-701.
- Qin, S. J., and T. A. Badgwell. "A Survey of Industrial Model Predictive Control Technology." *Control Engineering Practice* 11, no. 7 (2003): 733-764.
- Quesada, I., and I. E. Grossmann. "An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problem." *Computers & Chemical Engineering* 16 (1992): 937-947.

- Raghuathan, A. U., and L. T. Biegler. "Mathematical Programs with Equilibrium Constraints (MPECs) in Process Engineering." *Computers & Chemical Engineering* 27 (2003): 1381-1392.
- Rawlings, J. B., and K. Muske. "The Stability of Constrained Receding Horizon Control." *IEEE Transactions on Automatic Control* 38, no. 10 (1993): 1512-1516.
- Ricker, N. L. "Model Predictive Control: State of the Art." *Chemical Process Control – CPC IV* (1991): 271-296.
- Ricker, N. L. "Optimal Steady-State Operation of the Tennessee Eastman Challenge Process." *Computers & Chemical Engineering* 19, no. 9 (1995): 949-959.
- Ricker, N. L., and J. H. Lee. "Nonlinear Model Predictive Control of the Tennessee Eastman Challenge Process." *Computers & Chemical Engineering* 19, no. 9 (1995a): 961-981.
- Ricker, N. L., and J. H. Lee. "Nonlinear Modeling and State Estimation for the Tennessee Eastman Challenge Process." *Computers & Chemical Engineering* 19, no. 9 (1995b): 983-1005.
- Ricker, N. L. "Decentralized Control of the Tennessee Eastman Challenge Process." *Journal of Process Control* 6, no. 4 (1996): 205-221.
- Robinson, D., R. Chen, T. McAvoy, and P. D. Schnelle. "An Optimal Control Based Approach to Designing Plantwide Control System Architectures." *Journal of Process Control* 11, no. 2 (2001): 223-236.
- Rosenbrock, H. H. *Computer-Aided Control System Design*. New York: Academic Press, 1974.

- Ryoo, H. S., and N. V. Sahinidis. "Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design." *Computers & Chemical Engineering* 19, no. 5 (1995): 551-566.
- Sahinidis, N. V. "BARON: A General Purpose Global Optimization Software Package." *Journal of Global Optimization* 8, no. 2 (1996): 201-205.
- Shinskey, F. G. *Process-Control Systems : Application, Design, and Adjustment*. New York: McGraw-Hill, 1988.
- Shinskey, F. G. *Distillation Control : For Productivity and Energy Conservation*. New York: McGraw-Hill, 1984.
- Shridhar, R., and D. J. Cooper. "A Tuning Strategy for Unconstrained Multivariable Model Predictive Control." *Industrial & Engineering Chemistry Research* 37 (1998): 4003-4016.
- Skogestad, S., and M. Morari. "Effect of Disturbance Directions on Closed-Loop Performance." *Industrial & Engineering Chemistry Research* 26 (1987a): 2029-2035.
- Skogestad, S., and M. Morari. "Implications of Large RGA Elements on Control Performance." *Industrial & Engineering Chemistry Research* 26 (1987b): 2323-2330.
- Skogestad, Sigurd, Petter Lundstrom, and Elling W. Jacobsen. "Selecting the Best Distillation Control Configuration." *AIChE Journal* 36, no. 5 (1990): 753-764.
- Stanley, G., M. Marino-Galarraga, and T. J. McAvoy. "Short-Cut Operability Analysis: 1. the Relative Disturbance Gain." *Industrial & Engineering Chemistry, Process Design and Development* 24, no. 4 (1985): 1181-1188.

- Steuer, R. E. *Multiple Criteria Optimization: Theory, Computation, and Application*. New York: John Wiley & Sons, 1986.
- Trierweiler, J. O., and L. A. Farina. "RPN Tuning Strategy for Model Predictive Control." *Journal of Process Control* 13 (2003): 591-598.
- Trierweiler, J. O., and S. Engell. "The Robust Performance Number: A New Tool for Control Structure Design." *Computers & Chemical Engineering* 21, no. Supplement (1997): S409-S414.
- Turton, R., R. Bailie, W. Whiting, and J. Sheiwitz. *Analysis, Synthesis, and Design of Chemical Processes*. Upper Saddle River, N.J.: Prentice Hall, 1998.
- Venkat, Aswin N., Ian A. Hiskens, James B. Rawlings, and Stephen J. Wright. "Distributed Output Feedback MPC for Power System Control." *Proceedings of the IEEE Conference on Decision and Control* (2006): 4038-4045.
- Vlachos, C., D. Williams, and J. B. Gomm. "Genetic Approach to Decentralised PI Controller Tuning for Multivariable Processes." *IEE Proceedings - Control Theory and Applications* 146, no. 1 (1999): 58-64.
- Wachter, A. W. "Interior Point Method for Large-Scale Nonlinear Programming with Applications in Process System Engineering." Ph.D. diss., Carnegie Mellon University, 2002.
- Wang, P., and T. J. McAvoy. "Synthesis of Plantwide Control Systems using a Dynamic Model and Optimization." *Industrial & Engineering Chemistry Research* 40 (2001): 5732-5742.
- Woods, D. R. *Cost Estimation for the Process Industries*. McMaster University, Hamilton, Canada: unpublished, 1992.

Zhuang, M., and D. P. Atherton. "PID Controller Design for a TITO System." *IEE Proceedings - Control Theory and Applications* 141, no. 2 (1994): 111-120.

Ziegler, J. G., and N. B. Nichols. "Optimum Settings for Automatic Controllers." *Transactions of ASME* 64 (1942): 759-768.



## Appendix A

### Integrity

A feedback system with integrity has the property that it achieves “good” performance when a partial failure occurs (e.g. a sensor fails or a controller is placed on manual) or limitation (e.g., valve saturation) is encountered. Subsequently, the affected control loop is automatically turned off or manually switched to manual. Since “good” performance is not easy to be quantified most rigorous definitions of integrity are stability based (Marlin, 2000).

The basic assumptions used in various integrity definitions are

- The closed-loop system we study is in Figure A.1:

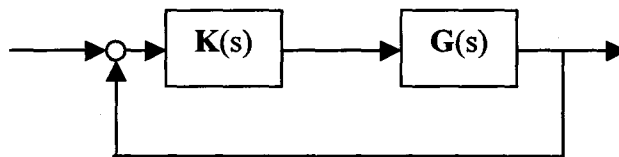


Figure A.1 The closed-loop system under integrity study

- $G(s)$  is  $n \times n$  open-loop stable process. An open-loop unstable process is always unstable upon the loss of all feedback controllers; therefore, it does not have integrity.

- All controlled variables have type 1 closed-loop behavior, which requires offset-free control at steady state (controller with integral action is required).
- $\mathbf{K}(s)$  is  $n \times n$  controller with an integrator in each channel in the form

$$\mathbf{K}(s) = \boldsymbol{\varepsilon} \mathbf{C}(s) \frac{1}{s} \quad (\text{A.1})$$

Where  $\mathbf{C}(s)$  is  $n \times n$  diagonal or block diagonal transfer function matrix and  $\boldsymbol{\varepsilon}$  is the tuning matrix.

The following are integrity definitions (Campo and Morari, 1994):

- The process  $\mathbf{G}(s)$  has *Integral Stabilizability (IS)* if there exists a controller  $\mathbf{K}(s)$  that stabilizes  $\mathbf{G}(s)$ .
- The process  $\mathbf{G}(s)$  has *Integral Controllability (IC)* if there exists a controller  $\mathbf{K}(s)$  that stabilizes  $\mathbf{G}(s)$  for all tuning matrix  $\boldsymbol{\varepsilon}$  defined as

$$\boldsymbol{\varepsilon} \in \{\alpha \mathbf{I} \mid \alpha \in (0,1]\} \quad (\text{A.2})$$

IC implies all control loops can be simultaneously detuned by a same factor, and the closed-loop system will remain stable.

- The process  $\mathbf{G}(s)$  has *Integral Controllability with Integrity (ICI)* if there exists a controller  $\mathbf{K}(s)$  that stabilizes  $\mathbf{G}(s)$  for all tuning matrix  $\boldsymbol{\varepsilon}$  defined as

$$\boldsymbol{\varepsilon} \in \{\text{diag}(\varepsilon_i) \mid \varepsilon_i \in \{0,1\}, i = 1, \dots, n\} \quad (\text{A.3})$$

ICI implies arbitrary combination of controllers can be brought in or out of service (each  $\varepsilon_i$  can be either 0 or 1), and the remaining closed-loop subsystem will remain stable without changes in tuning.

- The process  $\mathbf{G}(s)$  has *Decentralized Integral Controllability (DIC)* if there exists a controller  $\mathbf{K}(s)$  that stabilizes  $\mathbf{G}(s)$  for all tuning matrix  $\boldsymbol{\varepsilon}$  defined as

$$\boldsymbol{\varepsilon} \in \{\text{diag}(\varepsilon_i) \mid \varepsilon_i \in [0,1], i = 1, \dots, n\} \quad (\text{A.4})$$

DIC implies each controller can be detuned and/or brought in or out of service independently, and the remaining closed-loop subsystem will remain stable without retuning.

In real chemical plant operation, it is not uncommon for one or more controllers to be out of service due to routine sensor calibration and control valve maintenance. It is difficult to predict all possible operational subsystems at the control system design stage. Therefore, Integral Controllability with Integrity (ICI) is included in our control system design criteria to guarantee the closed-loop stability under all possible operating conditions.

The challenge of using ICI is that there is no easily verified condition to ensure it exists. A commonly used criterion, positive values on the diagonal (loop pairing) elements of Relative Gain Array (RGA) for the overall process, is a necessary condition for ICI. The tighter necessary condition for ICI involves testing for positive values of the diagonal elements of RGA for all possible subsystems of the process. In general, the necessary and sufficient condition for ICI is unknown. The next section will discuss the detailed properties of RGA.

## Appendix B

### Relative Gain Array (RGA)

*Relative Gain Array (RGA)* is defined as the ratio between the open-loop gain and the gain of the same input/output pair when all other loops are perfectly controlled at their set points in steady state (Bristol, 1966). It was introduced as a measurement of the interaction between different control loops.

$$\lambda_{ij} = \frac{\left. \frac{\partial CV_i}{\partial MV_j} \right|_{MV_k = \text{const}, k \neq j}}{\left. \frac{\partial CV_i}{\partial MV_j} \right|_{CV_k = \text{const}, k \neq i}} \quad (\text{B.1})$$

Where  $\lambda_{ij}$  is the relative gain between  $i$ th controlled variable  $CV_i$  and  $j$ th manipulated variable  $MV_j$ .

For open-loop stable system RGA can be easily calculated from the open-loop process gain matrix  $\mathbf{G}(0)$  as the following

$$\mathbf{RGA} = \mathbf{G}(0) \otimes [\mathbf{G}(0)^{-1}]^T \quad (\text{B.2})$$

where  $\otimes$  is the element-by-element product.

For an integrating system some of the open-loop process gains are infinite. Therefore, equation (B.2) is not applicable. If  $\mathbf{G}(s)$  is  $n \times n$  integrating system and it can be written as the following:

$$\mathbf{G}(s) = \begin{bmatrix} \mathbf{G}_N(s) \\ 1/s\mathbf{G}_I(s) \end{bmatrix} \quad \mathbf{G}_I \in \mathbf{R}^{(n-k) \times n}(s) \quad (\text{B.3})$$

where  $\mathbf{G}_N(s)$  is the non-integrating part of transfer function and  $1/s\mathbf{G}_I(s)$  is the integrating part of transfer function.  $\mathbf{G}_I(s)$  is the transfer function part that does not have integrator.

Arkun and Downs (1990) proved that

$$\text{RGA}[\mathbf{G}(s)] = \text{RGA} \begin{bmatrix} \mathbf{G}_N(s) \\ \mathbf{G}_I(s) \end{bmatrix} \quad (\text{B.4})$$

By simply set  $s$  to zero, we have

$$\text{RGA}[\mathbf{G}(0)] = \text{RGA} \begin{bmatrix} \mathbf{G}_N(0) \\ \mathbf{G}_I(0) \end{bmatrix} \quad (\text{B.5})$$

Since there is no integrator in  $\mathbf{G}_N(s)$  and  $\mathbf{G}_I(s)$  the open-loop steady state gain matrix  $\mathbf{G}_N(0)$  and  $\mathbf{G}_I(0)$  are finite.

RGA has some algebraic properties (Grosdidier and Morari, 1985):

- The sum of the elements of each row and each column of RGA is always unity.
- Any permutation of rows and columns in the process open-loop process gain matrix  $\mathbf{G}(0)$  results in the same permutation in the RGA.
- RGA is invariant under input and output scaling.
- If the process open-loop process gain matrix  $\mathbf{G}(0)$  is diagonal or triangular, then RGA is identity matrix.

Two reasons make RGA the most recognized shortcut metrics for control structure design. The first one is the easiness of RGA calculation; only steady state gain information is required. The second one is that RGA has rigorous relation with closed-loop properties, such as:

- *Stability* — Pairing a loop with negative relative gain value will cause at least one of three undesired situations: the loop is unstable, the whole closed-loop system is unstable or the closed-loop system without the loop is unstable (Grosdidier and Morari, 1985).
- *Robustness* — The norm of RGA bounds the minimal condition number of process gain matrix from the bottom, which establishes solid link with robustness (Skogestad and Morari, 1987). Pairings with big RGA value make the control system very sensitive to unstructured model mismatch.
- *Integrity* — Pairing loops with positive relative gain values is the necessary condition for Integral Controllability with Integrity (Campo and Morari, 1994).

In this research RGA is used as integrity criterion integrated into the control structure design problem. Two different necessary conditions of Integral Controllability with Integrity (ICI) are built into the algorithm. An easy-to-test necessary condition requires all positive diagonal elements of the pre-calculated overall RGA and the computation load is minimum. Another much tighter necessary condition requires RGA calculation of all possible subsystems at each branch and bound node, and the computation load grows exponentially with the problem size (Please refer to Section 5.8). Campo (1994) stated that the tighter necessary condition is also sufficient condition of ICI for 3×3 and smaller system. In general the tighter necessary condition is not sufficient although no proof of counterexample has been demonstrated.

All the results above are for open-loop stable system. Hovd and Skogestad (1994) presented RGA result of open-loop unstable system. Depending on the difference in the number of unstable poles in the plant, and its diagonal elements Hovd and Skogestad showed that one might want to pair on negative RGA to stabilize the system.

## Appendix C

### Block Relative Gain (BRG)

The Block Relative Gain (BRG) generalizes the concept of RGA to handle multivariable block structures (Manousiouthakis, *et al.*, 1986). BRG measures the interaction among multivariable controllers and will be explained with respect to a block 2x2 process, but it can be generalized to any number of blocks. Consider a square process  $G(s)$  partitioned into the following block structure.

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11}(s) & \mathbf{G}_{12}(s) \\ \mathbf{G}_{21}(s) & \mathbf{G}_{22}(s) \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \quad (\text{C.1})$$

The process is to be controlled by a block centralized controller  $\mathbf{K}(s)$ .

$$\mathbf{K}(s) = \begin{bmatrix} \mathbf{K}_{11}(s) & 0 \\ 0 & \mathbf{K}_{22}(s) \end{bmatrix} \quad (\text{C.2})$$

BRG for pairing  $(\mathbf{y}_1, \mathbf{u}_1)$  is defined as the ratio of the open-loop block gain matrix and the apparent gain matrix of the same pairing when the rest of the process is perfectly controlled to their set points.

$$BRG_{11} = \frac{\left. \frac{\partial y_1}{\partial u_1} \right|_{u_2=0}}{\left. \frac{\partial y_1}{\partial u_1} \right|_{y_2=0}} = G_{11}(0) \cdot [G^{-1}(0)]_{11} \quad (C.3)$$

Where  $[G^{-1}(0)]_{11}$  is the first block within the inverse of the steady state process gain

matrix as  $G^{-1}(0) = \begin{bmatrix} [G^{-1}(0)]_{11} & [G^{-1}(0)]_{12} \\ [G^{-1}(0)]_{21} & [G^{-1}(0)]_{22} \end{bmatrix}$ .

BRG has some algebraic properties (Manousiouthakis, *et al.*, 1986):

- Any permutation of rows and columns in the process open-loop process gain matrix  $G(0)$  results in the same permutation in the BRG.
- BRG is independent of input scaling but dependent on output scaling. However, the diagonal elements of BRG are invariant under input and output scaling.
- The values of the diagonal elements of BRG are equal to the summation of all the relative gain values within the same rows.

Similar to RGA, BRG has rigorous relation with closed-loop properties such as

- *Stability* — Choose a multivariable controller with negative determinant of BRG will cause at least one of three undesired situations: the multivariable control system by itself is unstable, the whole closed-loop system is unstable or the closed-loop system without the multivariable controller is unstable (Grosdidier and Morari, 1987). Therefore, the general loop pairing guideline is to choose multivariable controller with positive determinant of BRG.
- *Robustness* — The spectral radius of any BRG associated with the system is the lower bound of Euclidean condition number of the system (Nett and Manousiouthakis, 1987). In general, a control system with large maximum singular value of BRG is difficult to control.



- *Integrity* — Selecting control with positive determinant of BRG is the *necessary* condition for Integral Controllability with Integrity for block centralized structure (Chiu and Arkun, 1990).

In this research BRG is used as integrity criterion integrated into the block centralized structure design problem. The necessary condition of integrity is built in the algorithm. At each branch and bound node BRG of the block structure candidate is calculated. If the determinant of BRG is negative, the branch is pruned (Please refer to Chapter 6).

## Appendix D

# Interior Point Method with Complementarity Constraints

### D.1 Actuator Saturation as Complementarity Constraints

In this research the control structure design problem includes the actuator saturation formulation (D.1), which allows manipulated variables to saturate at their limits and provides a realistic control performance prediction (Section 3.3.3).

$$u_j(t) = u_j(t-1) + \Delta u_j(t) - s_j^U(t) + s_j^L(t) \quad (\text{D.1a})$$

$$[u_j(t) - u_j^L] \cdot s_j^L(t) = 0 \quad (\text{D.1b})$$

$$[u_j^U - u_j(t)] \cdot s_j^U(t) = 0 \quad (\text{D.1c})$$

$$u_j^L \leq u_j(t) \leq u_j^U \quad (\text{D.1d})$$

$$s_j^U(t) \geq 0 \quad (\text{D.1e})$$

$$s_j^L(t) \geq 0$$

where constraints (D.1b)~(D.1e) are complementarity constraints.

In general, mathematic program with complementarity constraints (MPCC) takes the following form:

$$\begin{aligned}
& \min \phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) \\
& \text{s.t. } \mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{0} \\
& \quad \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \geq \mathbf{0} \\
& \quad \mathbf{c}(\mathbf{x}, \mathbf{y}) = x_i y_i = 0 \quad i = 1 \dots n_c \\
& \quad (\mathbf{x}, \mathbf{y}) \geq \mathbf{0}
\end{aligned} \tag{D.2}$$

where variables  $\mathbf{x}$  and  $\mathbf{y}$  are complementarity pairs, and  $\mathbf{z}$  are variables that do not appear in the complementarity constraints.

MPCC is a special type of problem. Since the inequality conditions cannot be strictly satisfied for any feasible solution, complementarity constraints violate the Mangasarian-Fromovitz constraint qualification. This violation implies unbounded Lagrange multipliers that may cause difficulties with classical NLP solving technologies (Fletcher *et al.*, 2002). Therefore, a specially designed solving technology is needed to handle complementarity constraints explicitly.

## D.2 Interior Point Method for General NLP

Given a general constrained NLP

$$\begin{aligned}
& \min \phi(\mathbf{x}) \\
& \text{s.t. } \mathbf{h}(\mathbf{x}) = \mathbf{0} \\
& \quad \mathbf{g}(\mathbf{x}) \geq \mathbf{0}
\end{aligned} \tag{D.3}$$

The optimality condition (KKT) for (D.3) is:

$$\nabla \phi(\mathbf{x}) - \nabla^T \mathbf{h}(\mathbf{x}) \boldsymbol{\lambda} - \nabla^T \mathbf{g}(\mathbf{x}) \boldsymbol{\theta} = \mathbf{0} \tag{D.4a}$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \tag{D.4b}$$

$$\mathbf{g}(\mathbf{x}) \geq \mathbf{0} \tag{D.4c}$$

$$g_i(\mathbf{x}) \theta_i = 0 \quad i = 1 \dots n_g \tag{D.4d}$$

$$\boldsymbol{\theta} \geq \mathbf{0} \tag{D.4e}$$

Where  $n_g$  is the number of inequality constraints.

The Newton method can be applied to (D.4); however, constraints (D.4d) prevent the application of a pure Newton method because of slow down convergence.

One approach to ensure that the iterates are strictly feasible is to relax the KKT condition solved by Newton method as follows:

$$\nabla \phi(\mathbf{x}) - \nabla^T \mathbf{h}(\mathbf{x})\lambda - \nabla^T \mathbf{g}(\mathbf{x})\theta = \mathbf{0} \quad (\text{D.5a})$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \quad (\text{D.5b})$$

$$\mathbf{g}(\mathbf{x}) \geq \mathbf{0} \quad (\text{D.5c})$$

$$g_i(\mathbf{x})\theta_i = \mu \quad i = 1 \dots p \quad (\text{D.5d})$$

$$\theta \geq \mathbf{0} \quad (\text{D.5e})$$

where  $\mu > 0$  is a parameter that is gradually reduced to zero. It can be shown that a unique solution exists for the set of nonlinear equations (D.5) for every  $\mu > 0$  if and only if a strictly feasible solution exists for the original problem (Wright, 1997). The set of strictly feasible solutions  $\{\mathbf{x}(\mu) \mid \mu > 0\}$  and  $\{\theta(\mu) \mid \mu > 0\}$  is frequently referred to as the central path.

### D.3 Interior Point Method for NLP with Complementarity Constraints

The KKT condition for the problem with complementarity constraints (D.2) can be written as:

$$\nabla_{\mathbf{x}} \phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) - \nabla_{\mathbf{x}}^T \mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{z})\lambda - \nabla_{\mathbf{x}}^T \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{z})\theta - \nabla_{\mathbf{x}}^T \mathbf{c}(\mathbf{x}, \mathbf{y})\rho - \boldsymbol{\pi} = \mathbf{0} \quad (\text{D.6a})$$

$$\nabla_{\mathbf{y}} \phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) - \nabla_{\mathbf{y}}^T \mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{z})\lambda - \nabla_{\mathbf{y}}^T \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{z})\theta - \nabla_{\mathbf{y}}^T \mathbf{c}(\mathbf{x}, \mathbf{y})\rho - \boldsymbol{\psi} = \mathbf{0} \quad (\text{D.6b})$$

$$\nabla_{\mathbf{z}} \phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) - \nabla_{\mathbf{z}}^T \mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{z})\lambda - \nabla_{\mathbf{z}}^T \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{z})\theta = \mathbf{0} \quad (\text{D.6c})$$

$$\mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{0} \quad (\text{D.6d})$$

$$\mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{z}) - \mathbf{u} = \mathbf{0} \quad (\text{D.6e})$$

$$x_i y_i = 0 \quad i = 1 \dots n_c \quad (\text{D.6f})$$

$$\pi_i x_i = 0 \quad i = 1 \dots n_c \quad (\text{D.6g})$$

$$\psi_i y_i = 0 \quad i = 1 \dots n_c \quad (\text{D.6h})$$

$$\theta_i u_i = 0 \quad i = 1 \dots n_g \quad (\text{D.6i})$$

$$(\mathbf{x}, \mathbf{y}, \theta, \boldsymbol{\pi}, \boldsymbol{\psi}, \mathbf{u}) \geq \mathbf{0} \quad (\text{D.6j})$$

The KKT condition (D.6) has four sets of complementarity constraints (D.6f) ~ (D.6i), in which (D.6f) presents in the original problem (D.2), and (D.6g) ~ (D.6i) are for optimality. The conventional interior point strategy designed for general NLP only relaxes the complementarity constraints for optimality. The same strategy can also be applied to the complementarity constraints in the original problem (Baker, 2006).

IPOPT-C (Ragunathan and Biegler, 2003) is an algorithm for solving MPCC based on the interior point code IPOPT (Wachter, 2002), where the interior point method is modified to treat complementarity constraints using the strategy described previously.

IPOPT-C outperforms common NLP solvers, such as MINOS and CONOPT2, when solving MPCC (Baker, 2006). Our experience with IPOPT-C confirms it can solve the control structure design problem with explicit actuator saturation formulation fast and reliably. However, IPOPT-C only uses local information for search and the control structure design problem is non-convex; therefore, global optimality is not guaranteed.

## **Appendix E**

### **Capital Cost Estimation for FCCU Air Blower**

In Section 3.4, Case C requires an air blower with 600% extra capacity beyond its normal operation. The cost of the required air blower with correction for capacity, pressure and inflation is calculated. The cost estimation method can be found in, for example, Woods (1992) and Turton et. al. (1998).

Arbel *et al.* (1996) did not give the capacity data of the air blower in their case study. The capacity data of a typical air blower in Fluidized Catalytic Cracking Unit is found in Decroocq's book (1984), which is a centrifugal blower working at 150 kPa with capacity of 13.5 m<sup>3</sup>/sec.

The cost information for the equipment is given by Woods (1992). The inflation factor is calculated based on Chemical Engineering Plant Cost Index (CEPCI) of January 2008 (Chemical Engineering, 2008).

Table E.1 shows the cost estimation for the extra capacity of the air blower.

Table E.1 Capital cost for the extra capacity of the air blower

|                | Cost<br>10 <sup>3</sup> \$ | Size<br>m <sup>3</sup> /s | Factor | Pressure<br>140- 200 kPa | Include:<br>drive, motor | Inflation<br>factor |
|----------------|----------------------------|---------------------------|--------|--------------------------|--------------------------|---------------------|
| Listed case    | 60                         | 30                        | 0.61   | ×4.5                     | ×1.6                     | ×4.31               |
| Base capacity  | 1,143                      | 13.5                      | √      | 150                      | √                        | √                   |
| Extra capacity | <b>3,412</b>               | 81                        | √      | 150                      | √                        | √                   |

## **Appendix F**

### **Simulation Configurations**

This appendix summarizes the environment and parameters of the case studies in this research. The hardware configuration of the computer used in this research is listed in Table F.1. Table F.2 lists all the software and its version number that is used to solve the control design problem. The common parameters for all the case studies are listed in Table F.3.

Table F.2 Computer Hardware configuration

|               |                       |
|---------------|-----------------------|
| CPU           | Intel Pentium 4, 1GHz |
| Number of CPU | 1                     |
| Memory        | 512 MB                |



Table F.2 Software Version

|                      | Name    | Version  |
|----------------------|---------|----------|
| Operating system     | Windows | 2000 SP4 |
| Modeling environment | MATLAB  | R13      |
|                      | GAMS    | 21.4     |
|                      | AMPL    | Plus 1.6 |
| Solver               | CPLEX   | 9.0      |
|                      | BARON   | 7.2      |
|                      | CONOPT  | 3        |
|                      | SBB     | 21.4     |
|                      | IPOPT   | 2.2.1    |

Table F.3 Simulation parameter

|                                 | Sample<br>time | Time<br>horizon<br>(steps) | CV measurement noise<br>(standard deviation) |      | MV   | Lower<br>bound | Upper<br>bound |
|---------------------------------|----------------|----------------------------|--|------|--|----------------|----------------|
| FCC                             | 2 sec          | 30                         | $T_{\text{rgn}}$ (°F)                        | 0.1  | $F_{\text{air}}$ (lb air/lb feed)                          | 0              | 0.8            |
|                                 |                |                            | $T_{\text{ris}}$ (°F)                        | 0.1  | $F_{\text{cat}}$ (lb cat/lb feed)                          | 0              | 10             |
| Fired heater                    | 0.5 sec        | 30                         | T1   | 0.1  | V1   | -5             | 5              |
|                                 |                |                            | T2   | 0.1  | V1   | -5             | 5              |
|                                 |                |                            | T3   | 0.1  | V1   | -5             | 5              |
|                                 |                |                            | T4   | 0.1  | V1   | -5             | 5              |
| Non-square<br>test system       | 0.5 sec        | 400                        | y1   | 0.1  | u1   | -10            | 10             |
|                                 |                |                            | y2   | 0.1  | u2   | -10            | 10             |
|                                 |                |                            |  |      | u3   | -10            | 10             |
| Tennessee<br>Eastman<br>problem | 0.2 hr         | 60                         | $X_{\text{G/H}}$ (%mol/%mol)                 | 0.5  | $F_{\text{D/E}}$ (kg $\text{h}^{-1}$ /kg $\text{h}^{-1}$ ) | 0              | 1              |
|                                 |                |                            | $L_{\text{reactor}}$ (%)                     | 0.5  | $F_{\text{E/C}}$ (kg $\text{h}^{-1}$ /kscmh)               | 0              | 900            |
|                                 |                |                            | $X_{\text{A/C}}$ (%mol/%mol)                 | 0.25 | $F_{\text{A/C}}$ (kscmh/kscmh)                             | 0              | 0.5            |
|                                 |                |                            | $X_{\text{B}}$ (%mol)                        | 0.3  | $F_{\text{purge}}$ (kscmh)                                 | 0              | 0.6            |
|                                 |                |                            | $T_{\text{separator}}$ (°C)                  | 0.1  | $T_{\text{reacCW}}$ (°C)                                   | 0              | 100            |