ISBN 978-82-326-4260-1 (printed version)
ISBN 978-82-326-4261-8 (electronic version)
ISSN 1503-8181
ITK Report: 2019-12-W

NTNU
Norwegian University of
Science and Technology

Jonatan Ralf Axel Klemets

Doctoral Thesis

Doctoral theses at NTNU, 2019:332

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

NTNU

Doctoral theses at NTNU, 2019:332

Jonatan Ralf Axel Klemets

Topics in the optimal operation of process plants

NTNU
Norwegian University of
Science and Technology

Jonatan Ralf Axel Klemets

Topics in the optimal operation of process plants

Thesis for the degree of Philosophiae Doctor

Trondheim, August 2019

Norwegian University of Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-326-4260-1 (printed version)
ISBN 978-82-326-4261-8 (electronic version)
ISSN 1503-8181
ITK Report: 2019-12-W

MILJØMERKET
241 TRYKKSAK 731

Doctoral theses at NTNU, 2019:332
Printed by Skipnes Kommunikasjon as

Summary

An increasingly competitive global market, together with stricter environmental and safety regulations make it necessary for chemical process plants to operate close to its optimum. As a result, there has been a growing interest in online optimization methods, e.g., model predictive control (MPC), real-time optimization (RTO), and economic MPC (EMPC). However, implementing such techniques remains challenging, mainly due to the computational complexity and lack of accurate dynamic models. Another approach is to use simple control structures that keep specific controlled variables (CVs) at a constant value, also known as self-optimizing control [122]. The central idea of self-optimizing control is to select CVs such that in the presence of disturbances, the loss is minimized by holding them at constant set-points. Besides using single measurements, selecting linear combinations of measurements as CVs will further improve the self-optimizing control performance.

Using all measurements available will, in theory, give the smallest loss, but increases the risk of getting sensor failures and makes implementing the control structure more difficult. Instead, it is preferable to find an optimal measurement subset using the branch and bound method derived in [26] or the mixed-integer quadratic programming (MIQP) approach in [149]. However, when using decentralized control, it is often desirable to impose some structural constraints on the CVs. E.g., by only combining manipulated variables (MVs) with CVs associated with certain units or parts of the process. Unfortunately, when structural constraints are included, it makes the underlying optimization problem non-convex and thus, finding the optimal solution is difficult. In the first part of this thesis, an alternating direction method of multipliers (ADMM) algorithm is proposed for incorporating structural constraints on the CVs. The resulting algorithm is computationally very efficient and able to find local solutions that give similar or better performance when compared to existing methods.

Self-optimizing control focuses on the steady-state operation, and therefore, the CVs are typically calculated using only steady-state models of the process. As a consequence, little attention has been put on the dynamic performance when selecting measurement combinations, where the general approach is to first compute the optimal CVs and then design their respective controllers. The optimal measurement combinations, can often (especially if many measurements are used) result in very dynamically complex systems, that makes designing the feedback controllers difficult. If dynamic models of the process are available, then it should be possible to also consider how using a measurement combination as CVs will affect the dynamics of the system. In the second part of the thesis, PI controllers and measurement combinations are simultaneously obtained with the aim to find an optimal trade-off between minimizing the steady-state loss and the transient response for the resulting closed-loop system. A solution can be found by solving a bilinear matrix inequality (BMI), which becomes a linear matrix inequality (LMI) by specifying a stabilizing state feedback gain. The resulting control structures were evaluated on several case studies that consisted of different distillation column models. The simulations showed that the resulting control structures could give comparable performance to model predictive controllers (MPC) as long the parameters for the PI controllers and the CVs had been chosen appropriately.

In the ideal case, it would be sufficient to only use self-optimizing control variables with feedback controllers, since the operation would remain near-optimal without needing to change the set-points despite there being disturbances present. However, self-optimizing control alone is unlikely going to achieve truly optimal operation, and will probably require the inclusion of some online optimization method. Common for most of these algorithms (e.g., RTO, EMPC) is that they require more information about the current states and disturbances of the process. Measuring all the relevant states and disturbances is in general not possible, and thus, they must be estimated using appropriate state estimators. For chemical processes and other large-scale systems, using centralized state estimators are in general not favorable due to the high computational complexity. In addition, developing and maintaining a set of local models will, in general, be a lot easier compared to using a single global model. Therefore, it would be preferable to decompose them into multiple different local estimators, that uses a local model and the locally available measurements. From the different local estimates, it should be possible to reconstruct a more accurate global state vector using some appropriate fusion method. However, most existing fusion methods are limited to fusing only two state vectors of the same size, where each locally computed estimate refers to the same states. However, chemical processes are usually composed of different units where the dynamic model for each unit contain their own set of states, with some being shared between each other. Therefore, for the last part of this thesis, a fusion algorithm is proposed that is able to fuse multiple state estimates, where parts of the local state vectors are overlapping each other. The resulting algorithm was able to provide a fused estimate with a lower estimation error compared to existing fusion methods.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU).

The doctoral work has been performed at the Department of Engineering Cybernetics with Professor Morten Hovd (the Department of Engineering Cybernetics) as main supervisor and Adjunct Professor Krister Forsman (Department of Chemical Engineering) as a co-supervisor.

Acknowledgments

First of all, I would like to give my sincerest gratitude to my supervisor, Morten Hovd, who gave me the opportunity to work under his supervision. His generous support and patience have been of tremendous importance, especially for the first half of my studies when I was still trying to find my scientific feet. Without his guidance and encouragement, I would not have been able to complete this work. I would also like to thank my co-supervisor, Krister Forsman, who allowed me to visit him at Perstorps chemical plant in Warrington. Being able to visit a chemical plant early on gave me a better understanding of the scope and some of the challenges faced in process control. This experience was very useful, considering me having had little prior knowledge of chemical plants when I started.

Next, I would like to thank all my colleagues at the Department of Engineering Cybernetics for all their support during my PhD studies. I want to thank my office mates, Mikkel, and Waseem, as well as everyone else in the Lyapunov room. You always had time to discuss and help me solve any issue I had. I am also very grateful to Tove and the rest of the administrative staff, who were always extremely helpful. A special thanks go to the cageball team. I have had a great time playing with you whenever I was able to.

Finally, I would not have been able to complete this thesis without the support from my parents, brothers, sisters, niece, nephews and the rest of my family. I am especially grateful to my parents, who have always been there for me, even though my career pursuits have resulted in less frequent visits.

Jonatan Klemets,
Trondheim, August 2019

Contents

List of figures

List of tables

Abbreviations

ADMM	Alternating direction method of multipliers
BMI	Bilinear matrix inequality
BNB	Branch and bound
CI	Covariance intersection
CSTR	Continuous stirred tank reactor
CV	Controlled variable
EI	Ellipsoidal intersection
EMPC	Economic model predictive control
ESC	Extremum seeking control
GSVD	Generalized singular value decomposition
LEA	Largest ellipsoid algorithm
LMI	Linear matrix inequality
LTI	Linear time-invariant
MIQP	Mixed-integer programming
MPC	Model predictive control
MV	Manipulated variable
NCO	Necessary conditions of optimality
PB3	Partial bidirectional branch and bound
PID	Proportional integral derivative
RGA	Relative gain array
RMSE	Root mean squared error
RTO	Real-time optimizing
SDP	Semi-definite programming
SOC	Self-optimizing control
SOF	Static output feedback
TPM	Throughput manipulator
UKF	Unscented Kalman filter

Part I

Introduction and preliminaries

Chapter 1

Introduction

The following chapter defines the motivation and scope of the research conducted in this thesis. In addition, it provides an overview of the thesis, the main contributions by the author, and a list of the publications written during the course of the PhD studies.

1.1 Motivation and scope

The ever-increasing competitive pressure in the global markets results in the need for continuously improving the performance of chemical processes. Operating the process close to its economically optimal operating point is thus essential. To achieve optimal operation of a chemical plant, the goal is to maximize the profitability while ensuring the plant is kept within acceptable operating regions that satisfy environmental, safety and product requirements. Unfortunately, changes in the optimal operating conditions are unavoidable due to external disturbances, equipment wear and fouling that affects the process operation, as well as changes in the prices for the raw materials and products. This leads to higher demands on the control system, and therefore, efficient design for the control structures and the individual controllers is essential for improving the operation of chemical plants. Typically, these control structures consist of a large number of simple feedback controllers that can be combined with more advanced online optimization-based algorithms to improve the profitability of the plant.

Optimal operation of processing plants is a challenging area where the continuous changes in the operating conditions require control structures that are both robust and capable of adapting to the new operating conditions. As a result, much research has gone into developing new advanced control algorithms with increased complexity. These controllers are often added on top of the already existing control structure, which further increases their complexity. However, before trying to implement advanced controllers, it may be worth investigating, whether the existing control structure is used to its fullest potential.

Self-optimizing control offers a simple feedback strategy, which involves choosing appropriate CVs that when controlled at a constant set-point, can give an acceptable loss compared to the truly optimal operation. The research on self-optimizing control is mainly focused on reducing the economic loss when the process is at steady-state. However, driving the process to the desired operating point requires implementing feedback controllers, that are capable of keeping the plant stable when external disturbances are present. Therefore, both the controllers and the selected CV will influence the robustness and controllability of the plant. A poorly chosen CV can impose limitations on the achievable control performance, which in turn may restrict the application of self-optimizing control to processes with low dynamic complexity.

In this thesis, the main focus has been on the transient behavior when selecting the self-optimizing control variables, where the primary objective has been to develop algorithms that simultaneously compute the measurement combinations together with PI controllers. The resulting control structure should be capable of providing near-optimal behavior, both in terms of the economic (steady-state) and the dynamic performance for the resulting closed-loop system, thus, reducing the need for having to implement more advanced controllers.

A common requirement for most advanced controllers is their need for more information about the current process states and disturbances. However, measuring all the relevant states and disturbances is in general not possible, and therefore, they have to be estimated. For large-scale process plants, this can be challenging due to high computational complexity and the varying degree of the availability and the accuracy of existing process models. Therefore, this thesis proposes a method for fusing multiple and partially overlapping state estimates that can be computed by local estimators for different subsystems.

1.2 Thesis overview

The thesis is structured into five different parts, with a total of twelve chapters that are organized as follows:

Part I – Introduction and preliminaries

Chapter 1 provides the introduction for this thesis and outlines the motivation and scope of the conducted research.
Chapter 2 gives the commonly recommended systematic procedure for control structure design in plantwide control. It also provides a summary of existing control strategies for achieving optimal operation in process plants.

Part II – Measurement selection in self-optimizing control

Chapter 3 reviews the previous work on self-optimizing control together with the null space method and exact local method that are used to find optimal linear measurement combinations. This is an important chapter for this thesis, since the main contributions in both Part II and III are based on the principles of self-optimizing control.
Chapter 4 covers some of the previous work made on optimal measurement selection in self-optimizing control that involves using mixed-integer quadratic programming (MIQP) or a branch and bound algorithm. An alternative approach is also proposed where the re-weighted l_1 norm is used to find an approximate solution.

The re-weighted l_1 norm is less accurate than the existing methods but possesses some properties that makes it easy to combine with other optimization problems, which will later be exploited in Chapter 8 of the thesis.
Chapter 5 describes the controlled variable (CV) selection problem when the optimal measurement combination is constrained to a particular structure, e.g., a block diagonal or a decentralized structure. Being able to find the optimal measurement combinations with the desired structure, can be of great significance to improve the dynamical controllability or to give the CVs a more intuitive physical meaning. Therefore, the first real contributions for the thesis is introduced in this chapter where an ADMM algorithm is proposed to calculate the optimal measurement combination with the predefined structural constraints. The algorithm were able to obtain better solutions within a given time frame when compared to two existing convex approximation methods and a commercial MINLP solver.

Part III – Accounting for the dynamics in self-optimizing control

Chapter 6 highlights the dynamic effects for the resulting system when measurement combinations are chosen as the CVs. These are first illustrated through an example to show that a significant reduction in the interactions between the CVs can be achieved if they are selected appropriately. The dynamic effects of the measurement combinations are further discussed by investigating how they influence the system poles and zeros.
Chapter 7 formulates several optimization problems for finding the H_2 or the H_∞ optimal static output feedback (SOF) controller. The resulting SOF gain can be transformed into decentralized PI controllers that are controlling a measurement combination if one of the proposed methods for augmenting the system matrices is used. This makes it possible to simultaneously compute the measurement combinations and the feedback controllers, which will be crucial in the next chapter.
Chapter 8 presents the main contribution for this thesis, where several algorithms are proposed that simultaneously calculates the measurement combinations and decentralized PI controllers while taking both the dynamic and steady-state performance into account. The algorithms are based on the SOF methods that were presented in the previous chapter and required solving a semi-definite programming (SDP) problem that contains a bilinear matrix inequality (BMI). The BMI makes the optimization problems non-convex, and they are, therefore, solved using an iterative procedure. The resulting algorithm can be combined with the re-weighted l_1 norm to penalize the number of measurements used or the ADMM algorithm to impose some structural constraints on the measurement combination. The obtained control structures are evaluated through simulations on several case studies and showed results that were comparable to using more advanced controllers.

Part IV – Hierarchical decentralized state estimation

Chapter 9 motivates the necessity for decentralized state estimation and reviews some of the most common fusion methods when fusing two state vectors that refer to the same states.
Chapter 10 aims to expand on the previously discussed fusion methods by proposing a new algorithm that can fuse multiple state vectors, where some of the states

overlap each other. This constitutes the third and final contribution for this thesis, where simulations show that the proposed fusion algorithm gives a smaller estimation error compared to the other methods.

Part V – Closing remarks

Chapter 11 concludes the thesis by summarizing the main contributions and their simulation results.
Chapter 12 discusses possible future research directions for the presented work.

1.3 Thesis contributions

This thesis provides three main contributions that has been split into two different research areas:

1. Self-optimizing control:
    a) Impose structural constraints on the measurement selection matrix.
    b) Improve the dynamics of the self-optimizing control variable.
2. Decentralized state fusion for multiple and partially overlapping state vectors.

The first contribution for this thesis was the development of an ADMM algorithm that can be used in self-optimizing control to find the optimal measurement combination with the desired structural constraints. The resulting algorithm was later combined with a branch and bound method to search for the optimal measurement subset that gives the optimal linear measurement combination with the specified structure.

The second and the primary contribution for this thesis was the development of several algorithms that computes the feedback controllers and the controlled variables (CVs) simultaneously, where the CVs consists of linear measurement combinations. Therefore, it becomes possible to search for controllers and CVs that give the optimal trade-off between the dynamic and the economic steady-state performance.

The third and final contribution was the development of a method for fusing multiple state vectors that can be of different size and partially overlap each other. This is a common setup in process plants since the different units can have their own estimator where only some of their states (e.g., flows, temperature, and concentrations) are shared between them. Thus, the purpose of the proposed fusion algorithm is to collect state estimates from different local estimators and reconstruct a global state vector with an improved accuracy.

1.4 Publications

The work for this thesis has resulted in the following peer reviewed publications:

- J. R. A. Klemets and M. Hovd. An iterative LMI approach to controller design and measurement selection in self-optimizing control. *Proceedings of the 2017 Asian Control Conference (ACC)*, pages 2849-2854, 2017.
  (Chapter 7 and 8).
- J. R. A. Klemets and M. Hovd. Controller design and sparse measurement selection in self-optimizing control. *Proceedings of the 10th IFAC International Symposium on Advanced Control of Chemical Processes (ADCHEM)*, pages 458-463, 2018.
  (Chapter 7 and 8).
- J. R. A. Klemets and M. Hovd. Hierarchical decentralized state estimation with unknown correlation for multiple and partially overlapping state vectors. *Proceedings of the 2nd IEEE Conference on Control Technology and Applications (CCTA)*, pages 508-514, 2018.
  (Chapter 9 and 10).
- J. R. A. Klemets and M. Hovd. Accounting for dynamics in self-optimizing control. *Journal of Process Control*, 76:15-26, 2019.
  (Chapter 7 and 8).
- J. R. A. Klemets and M. Hovd. An ADMM algorithm for incorporating structural constraints in self-optimizing control. *Proceedings of the 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS)*, pages 64-69, 2019.
  (Chapter 5).

Chapter 2

Control structure design

Control structure design is a crucial part when it comes to achieving optimal operation of process plants. Control structure design deals with selecting the manipulated variables (MVs) and the controlled variables (CVs) and determines the pairing between these variables. Although the interest surrounding control structure design in literature has increased, it still constitutes only a small fraction compared to, e.g., the research on controller design. However, there are strong arguments that can be made for control structure design being more important in practice. This was addressed by Foss in [44], where he criticized the control community for the gap between theory and practice. Since then, Morari and co-authors have in [97], [95], [96] introduced interesting theories on hierarchical control, multilevel optimization and on control structure design. More recently, systematic procedures for control structure design, including what to control and how to pair the variables have been introduced in [124], and [36], where it is usually referred to as plantwide control.

2.1 Plantwide control

Chemical process plants are typically operated with the aid of a multilayer hierarchical control structure, consisting of several layers that address different time scales [124], [36]:

- Scheduling (weeks),
- Site-wide optimization (days),
- Local optimization (hours),
- Supervisory (predictive, advanced) control (minutes),
- Regulatory control (seconds).

These different layers are illustrated in Figure 2.1. From each layer, the set-points for the CVs are sent from the upper layer downwards to the lower layer. The economic optimization is usually located in an upper layer and uses, e.g., real-time optimization (RTO) [101] to compute and send the optimal set-points to the lower layers. The role of the lower layers are to drive the process to the desired set-points using, e.g., model predictive control (MPC) or other low-level controllers (typically PID Controllers).

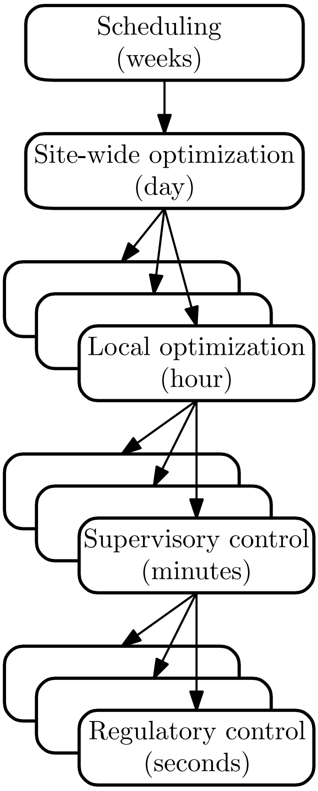


Figure 2.1: Typical control hierarchy in a chemical plant, adopted from [124].

The structural and strategic decisions which are involved in the control system design of a complete chemical plant is often referred to as plantwide control [124], [36]. A review on plantwide control is given in [82], where Larsson and Skogestad propose to decompose the problem into a top-down and analysis and bottom-up part:

I. Top-down analysis The top-down analysis focuses on steady-state economics, where an economic optimization problem is formulated. Optimization is performed both at the nominal operating point and for situations where important disturbances are present. Based on the optimization results, a self-optimizing analysis is done for finding the active constraint regions and selecting the best CVs in different operational regions. For the top-down analysis, usually, only a steady-state model of the process is required. The top-down design can be summarized as:

1. Define operational objectives (optimal operation):
    a) Identify a scalar cost function J (to be minimized).
    b) Define operational constraints.
2. Find optimal regions of active constraints:

   a) Identify steady-state degrees of freedom and the expected disturbances.
   b) Optimize (off-line analysis) the operation using the available degrees of freedom for the nominal case and with the expected disturbances.
   c) Find the optimal regions of active constraints.

3. Choose candidate measurements that will minimize the economic loss. This is further discussed in Chapter 3 on self-optimizing control.
4. Select location of throughput manipulator (TPM).

II. Bottom-up part The bottom-up analysis focuses on the dynamic control of the process. Dynamic models of the process are necessary to validate the implementation of the proposed CVs from the top-down analysis. The bottom-up design can be summarized as:

1. Select the structure of the regulatory control layer:
   a) Select "stabilizing" CVs.
   b) Choose inputs and pairings for the CVs that need stabilizing.
   c) Stabilize the process.
2. Select the structure of supervisory control, which should:
   a) Control the primary CVs.
   b) Supervise the regulatory layer.
   c) Perform switching between CVs for different regions.
3. Select the structure of (or need for) optimization layer (RTO) which should:
   a) Identify active constraints (identify regions).
   b) Update set-points for the CVs (if necessary).

One of the main focuses on this thesis involves choosing candidate measurements with the aim of both minimizing the economic loss and improving the regulatory control performance. Therefore, step 3 in the top-down analysis and the pairing problem in step 1 for the bottom-up analysis will implicitly be solved simultaneously in Chapter 8.

2.2 Common methods for achieving optimal operation

A short overview is given of different methods which aim to obtain optimal operation of process systems. This is only a brief review, and for more information, the reader is referred to the listed references and the references therein.

2.2.1 Extremum seeking control

Extremum seeking control (ESC) dates back to the 1920s [133] and has since had many successful reported implementations made on different process control applications [35]. The purpose of ESC is to find the optimal set-points (and MVs) that either maximizes or minimizes an objective function. To find the optimal operating point, the system is perturbed using an external excitation signal to compute the gradient [79]. Therefore, in ESC both the identification of process states and obtaining optimal inputs are combined [8]. Typically, this method assumes that the objective function, or at least explicit information on the structure of the objective function (how it depends on the states and disturbances) can be directly measured. In either case, a fundamental requirement is that the disturbance space has to be convex to guarantee that the estimation of the unknown parameters converge to their true value. This, together with the assumption of state feedback and slow speed of convergence are some of the major disadvantages for this method. In addition, the introduced excitation signal to the plant can often be seen as undesirable

2.2.2 Necessary conditions of optimality tracking

Necessary conditions of optimality (NCO) tracking, updates the inputs to keep an analytical gradient for the Lagrange associated with the optimal process operation at zero [126], [53]. Thus, NCO tracking resembles ESC in the respect that it does not require a model, but rather a measurement of the cost function. Similar to ESC, it suffers some of the same disadvantages, which can make it difficult to implement in practice. Mainly, the slow convergence speed caused by, e.g., inaccurate gradient information is an issue. However, one advantage of NCO tracking over extremum seeking control is the possibility to track and adapt to changes in the active constraints [127].

2.2.3 Real-time optimization

Real-time optimizing control (RTO), is an on-line optimization method that aims to compute the optimal set-point using a process model and the available measurements [31]. RTO is a popular method that is well suited to use for chemical process plants [100]. Typically, a steady-state process model is utilized even though dynamic versions of the RTO-framework are also starting to become more popular in the literature, e.g., [70]. However, RTO often requires a large and detailed model of the plant, which makes it challenging to develop dynamic models that accurately represent the process. Steady-state models, on the other hand, are often more readily available and are also computationally cheaper to solve compared to dynamic models.

When using conventional (steady-state) RTO, it is necessary to combine it with several other methods to accurately update the model and predict potential disturbances that are acting on the process. These typically, first require steady-state detection and data reconciliation, followed with parameter estimation and a model update. Since data reconciliation and parameter estimation are using steady-state models, it becomes necessary to first use a steady-state detection method to ensure that the steady-state condition has been fulfilled. Otherwise, the updated model provided for the steady-state optimization won't be accurate, and the resulting set-point will become suboptimal. This poses a major drawback for RTO since if a disturbance occurs, it requires the controllers at the regulatory layer to first settle the process at the current set-points before they can receive updated set-points from the RTO. As a consequence, the time required for the RTO to find the new optimal operating-point is usually several hours. Therefore, to improve the performance of RTO, the time required to update the model has to be reduced, in which data reconciliation methods are likely to play a vital role.

2.2.4 Model predictive control

Model predictive control (MPC) has received much attention in the control community and has become the most popular advanced control technology in the chemical processing industries [112]. MPC is a real-time optimization method, where an optimization problem is solved on-line using a dynamic process model together with some constraints. A sequence of control actions is chosen over a specified horizon window that will give the optimal trajectory according to the future prediction. From the computed control action sequence, only the first sample is implemented on the process after which the model is updated, and a new control action sequence is computed. The main advantages of MPC is its ability to control multi-input multi-output (MIMO) systems while handling process constraints. Traditionally, linear process models are used but implementation with nonlinear models, i.e., nonlinear MPC is becoming more popular, see e.g., [112], [92], [3] for numerous industrial installations on both linear and nonlinear MPCs. However, when formulating the optimization problem in MPC, it is important to ensure that it can be solved within the required time frame. Thus, using too large and complex model or a too long prediction horizon can be an issue. However, MPC is mainly used for set-point control, where an optimization layer (e.g., an RTO) gives the optimal set-points to use for varying disturbances. Thus, the economic objective is not being considered by the MPC.

2.2.5 Economic model predictive control

Recently there has been an increasing interest in economic model predictive control (EMPC) [39], which attempts to integrate the economic optimization and process control performance together. The central idea behind EMPC is to incorporate the control objective with the economic objective. As a result, the controller is simultaneously controlling the process while ensuring that it gives the optimal economic performance. This is a major advantage compared to, e.g., RTO since it doesn't have to wait until the process has reached steady-state before it starts driving the process to its optimal operating points. However, despite the recent advancements, there are still challenges when it comes to implementation in real processes, mainly due to the computational complexity and requirement for accurate dynamic models of the process.

2.2.6 Self-optimizing control

Self-optimizing control [122] refers to the strategy for finding CVs that when kept at constant set-points result in near-optimal operation of the process despite being impacted by unmeasured disturbances. The self-optimizing control variables are typically computed off-line using a steady-state model of the process after which simple feedback controllers can be used to keep the CVs at the nominal set-point. The ideal CV would be the gradient of the cost function since keeping it at zero would result in the optimal operation of the process. However, for most practical applications, it is not possible to measure the gradient. Instead, it is common to use a CV that consists of a function of the available measurements, typically a linear measurement combination, which will be discussed in more detail in Chapter 3.

The main advantage of self-optimization control is the simplicity of its control structure, which makes it possible to be directly implemented at the lower regulatory control layer. Therefore, it is able to drive the process to the desired operating point at a much faster time scale than the previously discussed methods. However, self-optimal control can suffer from plant-model mismatch since the CVs are typically computed using a local approximation of the process plant. Furthermore, self-optimizing control is only able to provide near-optimal operation under the assumption that the active constraints remain the same. Otherwise, new CVs have to be computed for every possible active constraint region.

Self-optimizing control is often presented as an alternative to the other methods (e.g., RTO, ESC, and NCO) for achieving optimal steady-state operation. However, this can be misleading as pointed out in [64], where it was argued that self-optimizing control should be viewed as a complement rather than a competitor to the other existing methods. The reason is that self-optimizing control operates at a faster time scale in a different control layer, which makes it possible to combine with the other strategies as demonstrated in, e.g., [130] and [77].

2.3 Conclusion

This chapter describes the typical control hierarchy for chemical process plants together with the systematic procedure for control structure design in plantwide control. Furthermore, various strategies for achieving optimal operation are briefly mentioned. One of these methods is self-optimizing control, which is used to develop CVs that try to incorporate the economic objective into the regulatory layer. However, more details on self-optimizing control will be given in the next chapter.

Part II

Measurement selection in self-optimizing control

Chapter 3

Self-optimizing control

In Section 2.2, some common optimization methods were briefly covered that all aimed to achieve optimal process operation. These consisted of both model-based optimization methods (RTO, and EMPC), and model-free methods (NCO, and ESC) but in general, their purpose was to find and update the optimal set-points based on the disturbances that were present. Another approach is to use simple control structures that keep specific controlled variables (CVs) at a constant value, also known as self-optimizing control [122].

Self-optimizing control dates back to 1980, where Morari and co-authors in [95] introduced the idea of using feedback to control and automatically drive the process to the optimal steady-state. The key concept of self-optimizing control is, therefore, to find CVs that when kept at constant set-points, indirectly keep the inputs optimal (indirect optimizing control) [122]. More precisely, the aim is to select CVs rather than determining optimal set-points.

For specified disturbances (d), the optimization problem can be formulated as

$$\min_{x,u} \; J\Big(x, u, d\Big) \tag{3.1}$$

$$\text{subject to} \;\; f(x, u, d)) = 0 \tag{3.2}$$

$$g_i(x, u, d)) \leq 0 \quad \forall i \tag{3.3}$$

$$y = f_y(x, u, d) \tag{3.4}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, and $d \in \mathbb{R}^{n_d}$ are the states, inputs, and disturbances, respectively. The equality constraints are represented by $f(\cdot)$ and contain the steady-state model equations; the inequality constraints in $g(\cdot)$ define the constraints on the operation, and the available measurements are given by $y \in \mathbb{R}^{n_y}$. Typically, J defines the economic cost of the process and can often be expressed as

$$J = \text{feed cost} + \text{utilities cost} - \text{product value}.$$

However, other objectives, such as energy efficiency and indirect control [62] are also possible. The solution to the optimization problem usually results in some of the constraints being active, i.e., $g_i(x, u, d) = 0$. To achieve optimal operation at steady-state, the variables related to the active constraints should be controlled and kept as close as possible to their optimal set-points. Stabilizing the plant and controlling the active constraints, therefore, requires a corresponding number of degrees of freedom. This results in a reduced space optimization problem:

$$\min_{u} J^{*}\Big(u, d\Big). \tag{3.5}$$

Here, the model equations and active constraints, are implicitly included in J^{*}. What remains is to determine which of the unconstrained variables should be kept constant by using the remaining degrees of freedom. Therefore, the goal is to minimize the constrained cost function (J), using the available degrees of freedom (u), in order to find the optimal operating point for the process.

Instead of evaluating a control structure using the reduced objective in (3.5), [125], and [56] proposed to quantify the performance by designing the loss function:

$$L(u, d) = J(u, d) - J_{opt}(d), \tag{3.6}$$

Here, the loss L is defined as the difference between the actual value of a given cost function and the truly optimal value (accounting for the correct value of the disturbance), i.e., truly optimal operation is achieved when $L = 0$. However, in general, $L > 0$ and thus a smaller value for the loss function, L implies that the plant is operating closer to its optimum. Based on the loss formulation in (3.6), Skogestad defines the concept of self-optimizing control as:

Self-optimizing control is when we can achieve an acceptable loss with constant set-point values for the controlled variables (without the need to reoptimize when disturbances occur) [122].

An illustration of the loss between two candidate control structures is shown in Figure 3.1, where it can be seen that c_1 gives a smaller loss compared to c_2 for some disturbance d. Hence, the control structure represented by c_1 would be preferable to c_2. With some candidate control structures being preferable to other, Skogestad proposed in [122], [125], four main requirements that good self-optimizing control variables should satisfy:

1. The CV should be easy to control, that is, the inputs u should have a significant effect on c.
2. The optimal value of c should be insensitive to disturbances.
3. The CV should be insensitive to noise.
4. In the case of several CVs, the variables should not be closely correlated.

Initially, brute-force methods were used to find the self-optimizing CVs [83], where the idea was simply to evaluate all possible CV candidates when accounting for the disturbances and measurement noise. The brute force approach requires that a solution is found for each possible set of CVs. In the case of selecting single measurements, it would require

$$C_{n_y}^{n_u} = \binom{n_y}{n_u} = \frac{n_y!}{(n_y - n_u)!n_u!}, \tag{3.7}$$

possible CVs to be evaluated. Therefore, the number of possible control structures will grow rapidly with the number of measurements n_y.

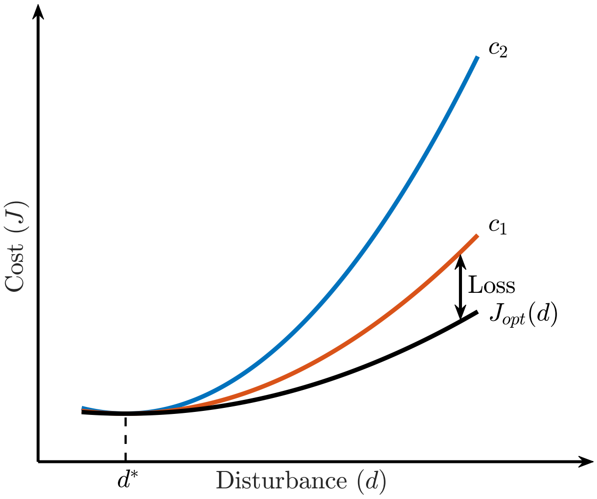


Figure 3.1: The cost for keeping two CVs (c_1 and c_2) at constant set-point compared to the cost for truly optimal operation (J_{opt}) when a disturbance occurs, adopted from [122].

With the brute-force evaluation quickly becoming intractable, most of the self-optimizing control methods are based on local analysis. The local methods use a steady-state model that has been linearized around the nominal operating point. The motivation for using local methods is that a candidate CVs should be able to perform well around the nominal operating point where the process is expected to operate most of the time; otherwise, the CVs can be excluded [65]. Introducing the deviation variables $\Delta u = u - u^*$ and $\Delta d = d - d^*$, then a Taylor expansion around the nominal operating point gives

$$J(u,d) = J(u^*,d^*) + \begin{bmatrix} J_u^* & J_d^* \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix}^T \begin{bmatrix} J_{uu}^* & J_{ud}^* \\ J_{ud}^* & J_{dd}^* \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix}. \tag{3.8}$$

Here, $J_u = \frac{\partial J}{\partial u}$ and $J_d = \frac{\partial J}{\partial d}$ denotes the derivatives of the cost function in (3.5) with respect to u, and d. $J_{uu} = \frac{\partial^2 J}{\partial u^2}$, $J_{ud} = \frac{\partial^2 J}{\partial u \partial d}$ and $J_{uu} = \frac{\partial^2 J}{\partial u^2}$ denote the second derivative with respect to u and d. Taking the gradient of (3.8) with respect to Δu and equating it to zero gives

$$\frac{\partial J}{\partial u} = J_u + J_{uu}\Delta u + J_{ud}\Delta d = 0. \tag{3.9}$$

Since (3.8) has been approximated around the nominal operating point it results in $J_u = 0$ and thus, (3.9) yields

$$\frac{\partial J}{\partial u} = \begin{bmatrix} J_{uu} & J_{ud} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} = 0. \tag{3.10}$$

From which the optimal input $\Delta u^*(d)$ is

$$\Delta u^*(d) = -J_{uu}\Delta u + J_{ud}\Delta d. \tag{3.11}$$

Combining (3.8) with the optimal input (3.11), Alstad showed in [4] that the loss can be expressed as

$$L = (\Delta u - \Delta u^*(d))^T J_{uu} (\Delta u - \Delta u^*(d)) \tag{3.12}$$

or alternatively,

$$L = \left\| J_{uu}^{-1/2} \begin{bmatrix} J_{uu} & J_{ud} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} \right\|_2^2, \tag{3.13}$$

where (3.11) has been inserted in (3.12).

Linearizing the measurements around the nominal point gives,

$$\Delta y = G^y \Delta u + G_d^y \Delta d + n, \tag{3.14}$$

where $n \in \mathbb{R}^{n_y}$ is the measurement noise; $G^y = \frac{\partial y}{\partial u} \in \mathbb{R}^{n_y \times n_u}$ and $G_d^y = \frac{\partial y}{\partial d} \in \mathbb{R}^{n_y \times n_u}$ are Jacobian matrices that represent the gain for the available measurements from the inputs and disturbances respectively. Choosing CVs from the measurements y yields

$$\Delta c = H \Delta y \tag{3.15}$$

$$= H G^y \Delta u + H G_d^y \Delta d + Hn, \tag{3.16}$$

where $H \in \mathbb{R}^{n_u \times n_y}$ is referred to as the measurement selection matrix and has the following property,

$$HH^T = I \tag{3.17}$$

when only single measurements are selected as CVs.

To quantify the loss for a range of disturbances, diagonal scaling matrices $W_d \in \mathbb{R}^{n_d \times n_d}$, and $W_n \in \mathbb{R}^{n_y \times n_y}$ are introduced to represent the disturbance and noise such that

$$\Delta d = W_d d', \tag{3.18}$$

$$n = W_n n', \tag{3.19}$$

where d', and n' are the scaled disturbances and measurement noise respectively. Using the scaling matrices for the given values of d', and n' together with CVs represented by the measurement selection matrix H, the input Δu can be obtained from (3.16):

$$\Delta u = (HG^y)^{-1} (\Delta c - HG_d^y W_d d' - HW_n n'). \tag{3.20}$$

By controlling the selected CVs at constant values (i.e., keeping $\Delta c = H \Delta y$ at zero) then (3.20) can be simplified to

$$\Delta u = -(HG^y)^{-1} H (HG_d^y W_d d' + W_n n'). \tag{3.21}$$

Combining Δu from (3.21) and the optimal input from (3.11) into the loss function (3.12) results in the loss expression derived in [56]:

$$L = \frac{1}{2} \left\| J_{uu}^{1/2} (HG^y)^{-1} HY \begin{bmatrix} d' \\ n' \end{bmatrix} \right\|_2^2, \tag{3.22}$$

where $Y := \begin{bmatrix} FW_d & W_n \end{bmatrix}$. Here, F represents the sensitivity matrix for the optimal deviations in the measurements with respect to changes in the disturbances:

$$F = \frac{\partial y^{opt}}{\partial d}. \tag{3.23}$$

The matrix F can be obtained analytically:

$$F = G_d^y - G^y J_{uu}^{-1} J_{ud}. \tag{3.24}$$

However, it is often easier to compute numerically by optimizing the nonlinear steady-state model of the plant for selected disturbances.

With the expectation operator denoted $\mathbb{E}[\cdot]$, and assuming the disturbances d and measurement noise n are independent and uniformly distributed in the sets $d \in \mathcal{D}$, and $n \in \mathcal{N}$. Then, the worst case and average loss were derived in [56], and [76] respectively and are given by

$$L_{worst} = \max_{d \in \mathcal{D}, n \in \mathcal{N}} L = \frac{1}{2} \left\| J_{uu}^{1/2} (HG^y)^{-1} HY \right\|_2^2, \tag{3.25}$$

$$L_{avg} = \mathop{\mathbb{E}}_{d \in \mathcal{D}, n \in \mathcal{N}} [L] = \frac{1}{2} \left\| J_{uu}^{1/2} (HG^y)^{-1} HY \right\|_F^2. \tag{3.26}$$

The authors of [76] proved that obtaining the H that minimizes the average loss in (3.26) is super-optimal and hence, the same H also minimizes the worst case loss in (3.25). However, the opposite isn't necessarily true. Therefore, only the minimization of the Frobenius norm will be considered in this thesis, where the goal is to choose the selection matrix H such that the loss in (3.26) gets minimized.

3.1 Optimal measurement combination

Rather than selecting single measurements for the unconstrained optimization problem in (3.5), a further reduction in loss can be obtained by selecting the CVs as optimal linear measurement combinations, resulting in the control structure seen in Figure 3.2. Two local methods for finding the optimal measurement combinations are the null space method and the exact local method.

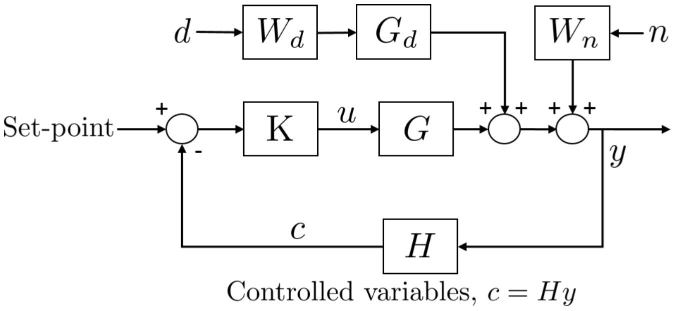


Figure 3.2: Block diagram of the self-optimizing control structure.

3.1.1 The null space method

Under the assumption that $n_y \geq n_u + n_d$ independent measurements are available and that the implementation error can be neglected, then [5] proposed the null space method for selecting a measurement combination that will result in zero local loss.

Theorem 3.1. ***Null space method*** *[5]. Assuming that there are n_u independent unconstrained variables u; n_d independent disturbances d; n_y independent measurements y. Moreover, if there is no implementation error (i.e., $W_n = 0$) and that $n_y \geq n_u + n_d$ independent measurements are available then selecting $H \in \mathbb{R}^{n_u \times n_y}$ as the left null space of F, $H \in \mathcal{N}(F^T)$ such that*

$$HF = 0, \tag{3.27}$$

then the loss in (3.26), and (3.25) will be zero ($L_{avg} = L_{worst} = 0$) as long as HG^y is non-singular.

Proof. When there is no measurement noise ($W_n = 0$) such that $Y = \begin{bmatrix} FW_d & 0 \end{bmatrix}$, then the loss in (3.26) is given by

$$L_{avg} = \frac{1}{2}\left\| J_{uu}^{1/2}(HG^y)^{-1}HY \right\|_F^2 \tag{3.28}$$

$$= \frac{1}{2}\left\| J_{uu}^{1/2}(HG^y)^{-1}H \begin{bmatrix} FW_d & 0 \end{bmatrix} \right\|_F^2 \tag{3.29}$$

$$= \frac{1}{2}\left\| J_{uu}^{1/2}(HG^y)^{-1}HFW_d \right\|_F^2. \tag{3.30}$$

If H is chosen such that $HF = 0$ then it is easy to see that (3.30) becomes zero. An alternative proof can be found in [5]. □

3.1.2 The exact local method

The null space method's underlying assumption of no implementation error is clearly unrealistic in practice. The method also requires the number of measurements to exceed the sum of the number of inputs and the number of disturbances ($n_y \geq n_u + n_d$), which can become very large. Therefore, it would be preferable to find a solution which gives an optimal trade-off between rejecting disturbances and implementation errors by minimizing (3.26) with respect to H:

$$\min_H \frac{1}{2}\left\| J_{uu}^{1/2}(HG^y)^{-1}HY \right\|_F^2. \tag{3.31}$$

At first glance, this seems like a nonlinear optimization problem. However, an important observation was discovered in [6], which found that (3.31) can be recast as a convex optimization problem.

Theorem 3.2. ***Exact local method*** *[6] If H is a full matrix (with no structural constraints) then the problem in (3.31) can be formulated as a convex constrained optimization problem:*

$$\min_{H} \frac{1}{2}\left\|HY\right\|_F^2 \tag{3.32}$$

$$\textit{subject to } \; HG^y = J_{uu}^{1/2} \tag{3.33}$$

Proof. From the original problem in (3.31), it can be shown that the optimal solution for H is non-unique and for any non-singular matrix $Q \in \mathbb{R}^{n_u \times n_u}$,

$$\hat{H} = Q^{-1}H \tag{3.34}$$

results in the same loss. This can be shown by [65]:

$$\begin{aligned} L_{avg} &= \frac{1}{2}\left\|J_{uu}^{1/2}(\hat{H}G^y)^{-1}\hat{H}Y\right\|_F^2 \\ &= \frac{1}{2}\left\|J_{uu}^{1/2}(Q^{-1}HG^y)^{-1}Q^{-1}HY\right\|_F^2 \\ &= \frac{1}{2}\left\|J_{uu}^{1/2}(HG^y)^{-1}QQ^{-1}HY\right\|_F^2 \\ &= \frac{1}{2}\left\|J_{uu}^{1/2}(HG^y)^{-1}HY\right\|_F^2. \end{aligned} \tag{3.35}$$

The non-uniqueness of H can be used to add the constraint in (3.33), which guarantees that the first part in (3.31) becomes $J_{uu}^{1/2}(HG^y)^{-1} = I$. Hence, the nonlinear optimization in (3.31) can be recast as the convex optimization problem in (3.32) and (3.33). □

Theorem 3.3. ***Analytical solution*** *[6] Assuming that YY^T is non-singular then the analytical solution to (3.32), and (3.33) is*

$$H^T = (YY^T)^{-1}G^y\left(G^{y^T}(YY^T)^{-1}G^y\right)^{-1}J^{1/2}. \tag{3.36}$$

Proof. The proof is given in [6]. □

The requirement that YY^T is non-singular will always be satisfied as long as the measurement noise is nonzero.

Remark. Some additional insight was given by [149], who noted that J_{uu} is not needed for finding the optimal H in (3.32), and (3.33). This means J_{uu} can be replaced with any non-singular matrix $Q \in \mathbb{R}^{n_u \times n_u}$ and still give the optimal H. This may simplify the calculations, as J_{uu} can be difficult to obtain numerically. However, J_{uu} would still be required to find the correct numerical value of the loss. As a consequence, the analytical solution in (3.36) can be simplified.

Theorem 3.4. ***Simplified analytical solution*** *[149] Assuming that YY^T is non-singular then a simplified analytical solution to (3.32)–(3.33) can be given by*

$$H^T = (YY^T)^{-1} G^y Q, \tag{3.37}$$

where $Q \in \mathbb{R}^{n_u \times n_u}$ is any non-singular matrix, e.g., $Q = I$.

Proof. The proof is given in [149]. □

3.2 Conclusion

This chapter gives a review of self-optimizing control, where the aim is to find CVs that when kept at constant set-point give near-optimal operation. A loss function is commonly used to quantify the performance for the selected CVs, where the loss is defined as the difference between the actual value of the operational cost compared to the truly optimal operation. To simplify the evaluation of the candidate CVs, it is common to use a local approximation of the loss function that can be obtained by first linearizing the process model and the cost function. The loss can further be reduced by using the null space method or the exact local method. These two methods compute the optimal linear measurement combinations from the local models that can then be used as CVs.

Recently, an approach for finding the global approximation of the CVs have been proposed in [145]. However, the global solution requires matrices that have to be constructed using optimization data from the whole operation space. Therefore, it is considered out of the scope of this thesis.

Chapter 4

Selecting a measurement subset

The lowest steady-state loss can be achieved when the measurement combination H is computed using all available measurements. However, for most practical cases, this is not desirable as it leads to overly complex control structures and increases the likelihood of getting sensor failures. Besides, often, there exists a subset of the available measurements that can be used without any significant reduction in the steady-state performance [65].

Finding the best measurement subset is a combinatorial optimization problem, and the loss has to be evaluated at every possible measurement combination. Selecting the best subset consisting of n measurements would, therefore, require that

$$C_{n_y}^{n} = \binom{n_y}{n} = \frac{n_y!}{(n_y - n)!n!}, \tag{4.1}$$

possible CV combination to be evaluated. To solve this problem, [75] developed a tailor-made branch and bound algorithm. Another approach was presented in [149], where the combinatorial problem was formulated using mixed-integer quadratic programming (MIQP) that can be solved with standard MIQP solvers. These two approaches will be summarized in the two upcoming sections. An alternative method will also be proposed, where sparsity is promoted by using the re-weighted l_1 norm.

4.1 Mixed-integer quadratic programming

Yelchuru and Skogestad proposed in [149] to find the best measurement subset by solving a MIQP problem. MIQP requires that the quadratic problem in (3.32), and (3.33) to be vectorized since it can often be inconvenient for numerical software such as Matlab to deal with matrix formulations. The measurement selection matrix,

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n_y} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n_y} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n_u,1} & h_{n_u,2} & \cdots & h_{n_u,n_y} \end{bmatrix} \tag{4.2}$$

is vectorized by stacking the rows of H, forming a column vector $h_\delta \in \mathbb{R}^{n_u n_y \times 1}$:

$$h_\delta = \begin{bmatrix} h_{1,1} & \cdots & h_{1,n_y} & h_{2,1} & \cdots & h_{2,n_y} & h_{n_u,1} & \cdots & h_{n_u,n_y} \end{bmatrix}^T. \tag{4.3}$$

The matrices G^y, $J_{uu}^{1/2}$, and Y are vectorized in a similar fashion to become $G_\delta^y \in \mathbb{R}^{n_u n_y \times n_u n_y}$, $J_\delta \in \mathbb{R}^{n_u n_u \times 1}$, and $Y_\delta \in \mathbb{R}^{n_u n_y \times n_u n_y}$, respectively. An equivalent vectorized quadratic problem of (3.32), and (3.33) can then be reconstructed:

$$\min_{h_\delta} \frac{1}{2}\left(h_\delta^T Y_\delta h_\delta\right) \tag{4.4}$$

$$\text{subject to } h_\delta G_\delta^y = J_\delta \tag{4.5}$$

To select the best subset, the MIQP formulation in [149] takes advantage of the property that controlling a subset is equivalent to setting the entire column that is not included in the subset to zero. Therefore, the following binary variables were introduced

$$\sigma_\delta = \begin{bmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_{n_y} \end{bmatrix}^T, \qquad \sigma_j \in \{0, 1\}, \tag{4.6}$$

to represent whether a measurement is included in the subset or not. If a measurement j is available in the selected subset, then $\sigma_j = 1$ and the jth column of H will have at least one non-zero element. Otherwise, $\sigma_j = 0$ and all the values in the jth column of H will be zero, which means that its corresponding measurement is discarded from the obtained subset. The binary variable σ_δ will be incorporated in (4.4) as constraints to restrict the number of measurements that will be used in H. The binary constraints can be written as

$$\mathcal{P}\sigma_\delta = n, \tag{4.7}$$

where $\mathcal{P} = 1_{1\times n_y}^T$ is a n_y dimensional matrix of ones, and n is the number of measurement that should be included in the obtained subset. The resulting mixed integer constraints on the column of H are formulated using the standard "big-M" formulation [61]. The problem of selecting the optimal measurement subset can then be written as

$$\min_{h_\delta, \sigma_\delta} \frac{1}{2}\left(h_\delta^T Y_\delta h_\delta\right) \tag{4.8}$$

$$\text{subject to } h_\delta G_\delta^y = J_\delta \tag{4.9}$$

$$\mathcal{P}\sigma_\delta = n \tag{4.10}$$

$$\begin{bmatrix} -m \\ -m \\ \vdots \\ -m \end{bmatrix} \sigma_j \leq \begin{bmatrix} h_{1,j} \\ h_{2,j} \\ \vdots \\ h_{n_u,j} \end{bmatrix} \leq \begin{bmatrix} m \\ m \\ \vdots \\ m \end{bmatrix} \sigma_j, \quad \forall j \in 1, 2, \cdots, n_y. \tag{4.11}$$

Here, m is a positive constant that is used in the big-M constraints to ensure that whenever σ_j is zero, the corresponding column in H will also be zero. In addition, the constraints in (4.11) are also used to bound the decision variables on H.

Therefore, if the value of m is chosen to be small, it will reduce the computational load required to find a solution. However, when m is chosen to be too small, it can result in some of the constraints to become active and as a consequence will generate a suboptimal solution. Selecting an appropriate value for m is thus, not straightforward and typically, requires using an iterative approach until no changes are seen in the final solution.

The main advantage of formulating the measurement subset selection problem as an MIQP is that it allows for standard MIQP solvers to be used, e.g., [149] used CPLEX (International Business Machines, 2014) [63] to solve the problem in (4.9)–(4.11).

4.2 Branch and bound method

The branch and bound (BAB) method take advantage of the monotonicity of the loss function with respect to the number of measurements. This means whenever a measurement is removed, it will result in a loss that is larger or equal to the loss for the original set of measurements. Similarly, when adding measurements to a subset, it should only be able to yield an improvement in the loss. Here, the general idea of the bidirectional branch and bound is presented, which was introduced by Cao and Kariwala in [27], for selecting individual measurements based on the minimum singular value criterion. They expanded their work to use the BAB approach for finding the optimal subset of measurement combinations for minimizing the worst-case loss in [74], and the average loss in [75]. For more details on BAB algorithms, the reader is referred to [27], [74], and [75], as well as the corresponding publicly available Matlab code in [24], [25], and [26].

The general principle of the BAB method is to divide (branch) the measurement selection problem into smaller subproblems. If the estimated loss for a subproblem is larger than a previously computed upper bound B, then the subproblem under consideration can be discarded (pruned) since it will not be able to provide the optimal solution. On the other hand, if the subproblem can't be pruned, it will be divided into smaller subproblems that need further evaluation. The procedure is repeated until there are no subproblems left to consider.

Let X_m denote the index set that describes the selected measurement subset, where the value of the subscript m indicates how many measurements are included in X. G^y_X denotes the gain matrix corresponding to the selected measurements, i.e., the rows of G^y corresponding to the indices in X. Similarly, Y_X denotes the matrix consisting of the rows in Y corresponding to the index set X. Finally, let H^*_X denote the optimal measurement selection matrix that gives the smallest loss for the subset of measurements given by the index set X. Then, for given noise and disturbances that are normally distributed, the optimal average loss is a function of the index set X:

$$L(X_m) = \frac{1}{2}\left\| J_{uu}^{1/2}(H^*_X G^y_X)^{-1} H^*_X Y_X \right\|_F^2. \tag{4.12}$$

Defining X_n as an n-element subset selected from X_m. The subset selection problem for finding the globally optimal n-element subset X^*_n that minimizes the loss can be formulated as

$$L(X_n^*) = \min_{X_n \subset X_m} L(X_n). \tag{4.13}$$

Using the monotonicity properties, an efficient BAB algorithm can be developed for selecting the best subset of measurements without having to evaluate every possible n-element measurement combination. Therefore, a BAB approach can be an efficient method for finding the globally optimal subset, while significantly reducing the number of subsets that needs to be evaluated. A downward or an upwards BAB algorithm can be implemented, where measurements are gradually removed or added to the subsets until the optimal subset has been found.

4.2.1 Downward branch and bound

The monotonicity property implies that if one index set X_n is contained in another index set X_m ($X_n \subseteq X_m$), then the optimal loss of the superset must be less or equal to the optimal loss of the subset [65]:

$$L(X_m) \leq L(X_n). \tag{4.14}$$

Thus, to present the downward BAB approach, let $\underline{L}_n(X_s)$, $s > n$, be a downwards lower bound on L over all n-element subsets of X_s:

$$\underline{L}_n(X_s) \leq \min_{X_n \subseteq X_s} L(X_n). \tag{4.15}$$

Further, let B be a known upper bound to the minimum loss for the case when n variables are selected, such that

$$B \geq L_{avg}(X_n^*), \tag{4.16}$$

then this upper bound can be used to remove (prune) sets of measurements, i.e.,

$$L(X_n) > L(X_n^*), \quad \forall X_n \subseteq X_s \quad \text{if } \underline{L}_n(X_s) > B. \tag{4.17}$$

The expression in (4.17) indicates that none of the subsets of X_s can be the optimal subset. Therefore, X_s and all its subsets can be discarded without further evaluation, since none of them will be able to produce a smaller loss. The condition in (4.17) is referred to as the "pruning condition" and is used to systematically eliminate suboptimal subset without the need for further evaluation.

The search tree is, typically, constructed asymmetrically to avoid subset redundancy, i.e., the nodes on the same level contain a different number of branches. Figure 4.1 shows an example of a downwards solution tree, where a 2-element subset is selected from a set of 6-elements. The top node represents the set of all available measurements, whereas the bottom nodes are the terminal nodes for all possible 2-element subsets. Starting from the top node, at each level, one measurement gets removed from its parent set, where the numbers at the different nodes denote which measurement has been discarded. To evaluate all possibilities for selecting 2 out of 6 measurements, would require evaluating $C_6^2 = 15$ measurement combinations.

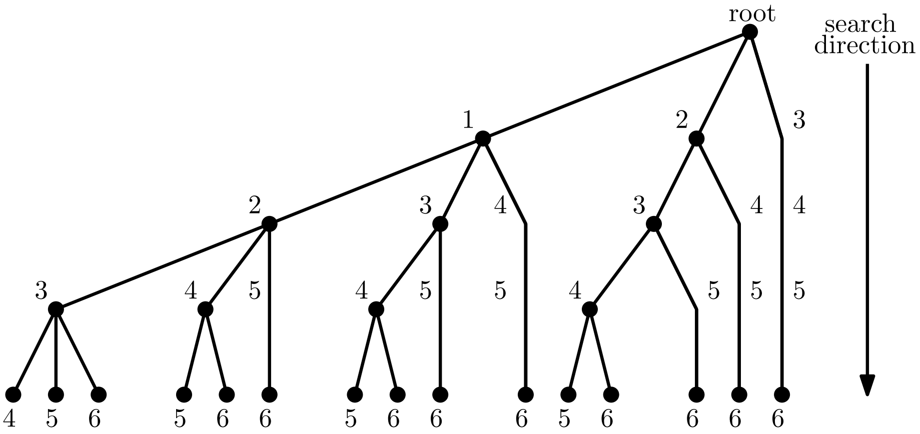


Figure 4.1: Downwards solution tree, adopted from [27].

Using the pruning condition in (4.17), the tree structure can be used to organize the search for the best combination. E.g., let's assume that the node furthest to the right has been evaluated with the resulting loss becoming the current best estimate of the upper bound B. Then, if the loss obtained from the two nodes at the first level are all higher ($L > B$), then those nodes and all their subnodes can be discarded. Thus, a solution has been found that required only evaluating 3 out the 15 possible measurement combinations. [65]

4.2.2 Upward branch and bound

It is possible to implement an upward BAB approach for finding optimal measurement subset, even though it is a lot less popular in the research literature compared to the downwards BAB. Contrary to a downward search, an upward search begins with an empty set and then gradually expands to supersets measurement-by-measurement, until the required subset size has been reached.

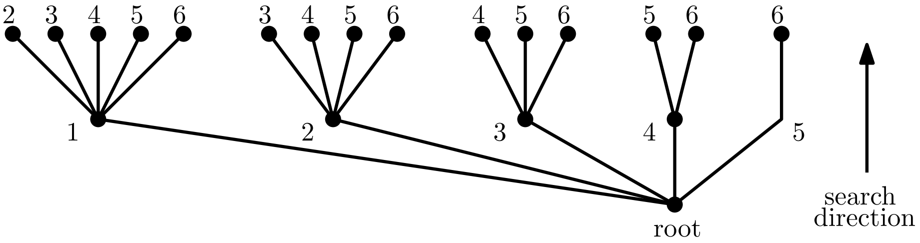


Figure 4.2: Upwards solution tree, adopted from [27].

The upward BAB takes advantage of the upwards monotonicity property that implies if an index set X_n is a superset of another index set X_s ($X_n \supseteq X_s$) then the optimal loss of the subset must be less or equal to the optimal loss of the superset:

$$L(X_s) \geq L(X_n). \tag{4.18}$$

Thus, to present the principle of upward BAB, let B be an upper bound of $L(X_n^*)$ as defined in (4.16) and $\underline{L}_n(X_s)$, $s < n$, be an upwards lower bound of L over all n-element supersets of X_s:

$$\underline{L}_n(X_s) \leq \min_{X_n \supseteq X_s} L(X_n). \tag{4.19}$$

Then,

$$L(X_n) > L(X_n^*), \quad \forall X_n \supseteq X_s \quad \text{if } \underline{L}_n(X_s) > B. \tag{4.20}$$

If the condition in (4.20) holds, then it means that none of the supersets for X_s can be globally optimal. Therefore, X_s and all its supersets can be pruned without further consideration.

The upward BAB method can also be represented using an asymmetric tree, where an example for selecting 2 elements from a 6-element set is shown in Figure 4.2. In the upward tree, the bottom node consists of an empty set, whereas the top nodes are the terminal nodes which represent all possible 2-element supersets. Every node (superset) is obtained by adding one additional measurement to its parent set until the top level is reached. Therefore, the sizes of the nodes get increased when going upwards in the search direction, where the numbers at the nodes in Figure 4.2 represent the added measurements.

4.2.3 Bidirectional pruning

In general, when trying to find optimal subsets from large measurement sets, the downwards pruning based BAB methods tend to be more efficient when the size of the desired subset is large. However, the upwards pruning based BAB methods tend to be more favorable when selecting subsets consisting of only a few variables. Therefore, more sophisticated bidirectional pruning methods was developed in [27], [74], and [75], which prune both from the top and the bottom of the tree, thus making them efficient at handling both cases. In addition, to combining upwards and the downwards pruning, the authors of [75] also developed fast pruning methods to improve the efficiency of discarding suboptimal nodes. However, when searching for measurement combinations, upwards pruning can only be applied when the number of elements for the node under consideration is greater than $n - n_u$. Thus, Kariwala and Cao referred to it as a partial bidirectional branch and bound (PB3) algorithm [74].

4.2.4 Bidirectional branching

Besides effective pruning methods, the PB3 algorithm further improves its performance by using effective branching rules. These rules are used to determine the way in which a node should be divided into multiple subnodes. The purpose of using effective branching rules is to try discard as many of the non-optimal subnodes as possible. By accounting for the advantages and disadvantages in upwards and downwards branching, Cao and Kariwala proposed using a novel bidirectional branching strategy in [27].

Instead of branching all possible nodes, the bidirectional branching approach only produces two branches at a time. These two branches consist of an upward branch and a downward branch of which the one with fewer terminal nodes (n-element subsets) will be evaluated first. This allows for the sub- or supernodes to be ordered in a way that the maximum number of nodes can be pruned that is associated with suboptimal nodes. The decision of which two nodes to branch and the order in which their subnodes should be evaluated is made on a best-first basis. This means that the most promising candidates are evaluated first, which increases the likelihood of being able to quickly discard suboptimal nodes and thus, the number of nodes that need to be evaluated can significantly be reduced.

4.3 Re-weighted l_1 norm

An alternative approach, for finding the optimal measurement subset is to solve a multi-objective optimization problem, that gives the optimal trade-off between steady-state loss and the number of measurements used. If a column-wise sparsity promoting function is included in (3.32) and (3.33), the optimization problem can be formulated as,

$$J = \min_{H} \frac{1}{2}\left\|HY\right\|_F^2 + \lambda \; card(H) \tag{4.21}$$

$$\text{subject to } \; HG^y = J_{uu}^{1/2}. \tag{4.22}$$

By specifying a scalar value for λ, there will be a trade-off between the steady-state loss and the cardinality of H, where the cardinality of the measurement matrix H is defined:

$$card(H) := \text{the number of non-zero columns of } H.$$

The cardinality function is non-convex and non-smooth, that still makes the optimization formulation in (4.21) a combinatorial problem.

To address this issue, several convex relaxations like the l_1 norm and the weighted l_1 norm have been proposed [23]. By using the weighted l_1 norm, the cardinality function can be approximated to:

$$card(H) \approx \sum_{i,j} W_{i,j} \|H_{i,j}\|_1 \tag{4.23}$$

The authors of [23] noted that if the weights $W_{i,j}$ are chosen to be inversely proportional to the l_1 norm, then there is an exact correspondence between the l_1 norm and the cardinality function. However, this requires *a priori* knowledge of the H matrix, and therefore, a re-weighted scheme needs to be implemented, where the weights are updated after every iteration (k) as,

$$W_{i,j}^{k+1} = \frac{1}{\|H_{i,j}^k\|_1 + \kappa} \tag{4.24}$$

where $1 >> \kappa > 0$ ensures the update is well-defined.

The weighted l_1 norm in (4.23) promotes element-wise sparsity. However, it can easily be modified to promote column (or row) sparsity as, e.g., shown in [7] by revising it as,

$$card(H) \approx \sum_{i,j} W_j \|H_{i,j}\|_1 \tag{4.25}$$

with the update rule:

$$W_j^{k+1} = \frac{1}{\sum_i \|H_{i,j}^k\|_1 + \kappa} \tag{4.26}$$

For a given value of λ, the iterative procedure, described in Algorithm 1 can be used to find a subset of the available measurements. By varying the value of λ, there will be a trade-off between the steady-state loss and the number of measurements used. Additional weights may also be included to penalize certain control structures, e.g., if some measurement links are unattractive due to high implementation cost.

Algorithm 1 Measurement selection using the re-weighted l_1 norm.

```
Initialize: For $k = 0$ obtain $H^k$ from (3.36 ) and compute $W$ using (4.26).
1: Set $k \leftarrow k+1$ and for the obtained $W$ solve:
       $H^k = \arg\min_H \frac{1}{2}\left\|HY\right\|_F^2 + \lambda \sum_{i,j} W_j \left\|H_{i,j}\right\|_1$
       subject to $HG^y = J_{uu}^{1/2}$
2: If $\|H^{k-1} - H^k\|_2 < \epsilon$ go to step 3, else update $W$ using (4.26) and repeat step 1 and 2.
3: Remove the measurements that correspond to the zero columns in $H$.
```

While the convergence properties for the re-weighted l_1 norm are still not clearly understood, numerical experiments have shown it to be a very efficient method for promoting sparsity, which also is demonstrated through simulations in Section 4.4.

4.4 Case study: Binary distillation column

In this example, the proposed Algorithm 1 is applied to the "column A" distillation column model [121], where a binary mixture is separated that has a relative volatility of 1.5. The distillation column has 41 stages, which includes the reboiler and the condenser. The stages are counted from the bottom with the reboiler as stage 1 and with the feed at stage 21. For the distillation column, the feed is assumed to be given. Thus, it has four degrees of freedom; bottoms flow rate (B_F), distillate flow rate (D_F), reflux flow rate (L_R), and vapor boilup (V_B). The distillate boilup and bottom flow rate are used to stabilize the two liquid levels in the condenser and the reboiler. This results in the LV configuration shown in Figure 4.3, where the two remaining degrees of freedom are:

$$u = \begin{bmatrix} L_R & V_B \end{bmatrix}^T. \tag{4.27}$$

The objective is to get a top product with 99% light component (1% heavy) and a bottom product with 1% light component, i.e., the cost function is

$$J = \left(\frac{x_H^{top} - x_H^{top,s}}{x_H^{top,s}} \right)^2 + \left(\frac{x_L^{btm} - x_L^{btm,s}}{x_L^{btm,s}} \right)^2, \tag{4.28}$$

where the specifications for the top and bottom products are denoted with the superscript s.

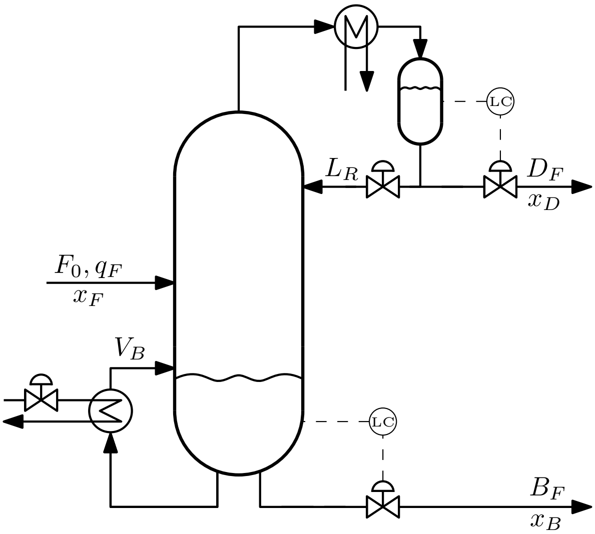


Figure 4.3: Binary distillation column with LV configurations, adopted from [121].

As composition often is difficult to measure, they will be controlled indirectly using the temperatures inside the column as in [62]. It is assumed that the temperatures $T_i(°C)$ on each stage i can be calculated using the linear function [121],

$$T_i = 0x_{L,i} + 10x_{H,i} \tag{4.29}$$

with an accuracy of $\pm 0.5°C$.

The main disturbances considered are changes in feed flow rate (F_0), feed composition (x_F) and feed liquid fraction (q_F), where F_0 and x_F can vary between 1 ± 0.2, and 0.5 ± 0.1 respectively. The nominal value for q_F is 1.0, with a lower bound of 0.9.

4.4.1 Measurement selection

Finding the optimal subset of measurements that minimizes the steady-state loss for the binary distillation column example has previously been studied in [74], and [149]. The problem was solved in [75] with a partial branch and bound (PB3) method, while in [149] a MIQP formulation was used. The optimal CVs and their respective steady-state loss when using 2, 3, 4, 5, and 6 measurements obtained using the PB3 approach presented by [75] can be seen in Table 4.1. The optimal CVs are compared to controlled variables computed using the re-weighted l_1 norm proposed in Algorithm 1. To obtain the desired subset size when using the re-weighted l_1 norm approach requires that appropriate values of λ have been chosen. Since there is no way of knowing what value of λ causes the desired trade-off between the loss and the no. of measurements, these values are obtained using trial and error. Where larger values of λ will reduce the size of the measurement subset at the expanse of increasing the loss. In Figure 4.4, the effect λ with respect to the measurement subset size is shown, where the resulting measurement combination H for sets of 2, 3, 4, 5 and 6 measurements can be seen in Table 4.1.

Table 4.1: Controlled variables and their respective loss, where the CVs obtained using (PB3) [75], are compared to using Algorithm 1 (WL1). Here, the values $\lambda = 0.085$, $\lambda = 0.072$, $\lambda = 0.05$, $\lambda = 0.04$, and $\lambda = 0.03$ have been used for Algorithm 1 to find subsets of 2, 3, 4, 5, and 6 measurements, respectively.

No. of meas.	Controlled variables (CVs)	Loss (L) $\frac{1}{2}\|HY\|_F^2$
2	$\text{PB3}: c = \begin{bmatrix} T_{12} \\ T_{30} \end{bmatrix}$	0.5478
	$\text{WL1}: c = \begin{bmatrix} T_{12} \\ T_{29} \end{bmatrix}$	0.5530
3	$\text{PB3}: c = \begin{bmatrix} T_{12} + 0.0446T_{31} \\ T_{30} + 1.0216T_{31} \end{bmatrix}$	0.4425
	$\text{WL1}: c = \begin{bmatrix} T_{12} + 15.2104T_{29} \\ T_{13} - 14.0138T_{29} \end{bmatrix}$	0.4605
4	$\text{PB3}: c = \begin{bmatrix} 1.0316T_{11} + T_{12} + 0.0993T_{31} \\ 0.0891T_{11} + T_{30} + 1.0263T_{31} \end{bmatrix}$	0.3437
	$\text{WL1}: c = \begin{bmatrix} 1.0811T_{12} + T_{13} + 0.2075T_{30} \\ 0.1205T_{12} + T_{29} + 1.0859T_{30} \end{bmatrix}$	0.3575
5	$\text{PB3}: c = \begin{bmatrix} T_{12} - 11.6579T_{21} + 23.9043T_{29} + 24.2286T_{30} \\ T_{13} + 11.2776T_{21} - 24.2492T_{29} - 24.5323T_{30} \end{bmatrix}$	0.2860
	$\text{WL1}: c = \begin{bmatrix} T_{11} + 1.6950T_{12} + 13.8901T_{29} + 15.5051T_{30} \\ T_{13} - 8.7428T_{29} - 9.5234T_{30} \end{bmatrix}$	0.3253
6	$\text{PB3}: c = \begin{bmatrix} 1.9015T_{12} + 1.8769T_{13} - T_{21} - 0.0858T_{29} + 0.0636T_{31} \\ 1.8004T_{12} + 1.7724T_{13} - T_{21} + 0.0835T_{30} + 0.1420T_{31} \end{bmatrix}$	0.2482
	$\text{WL1}: c = \begin{bmatrix} 2.0630T_{11} + 2.2804T_{12} + 2.0493T_{13} - T_{21} + 0.3015T_{30} \\ 2.2675T_{11} + 2.5064T_{12} + 2.2524T_{13} - T_{21} - 0.3238T_{29} \end{bmatrix}$	0.2632

Algorithm 1 promotes sparsity, using the weighted l_1 norm as a convex relaxation for the cardinality of the H matrix. As a consequence, it can't guarantee that it converges to the optimal measurement subset and therefore, it gives subsets with a slightly larger steady-state loss compared to the ones obtained in [75] and [149]. This is also demonstrated in Figure 4.5, where the loss with respect to the

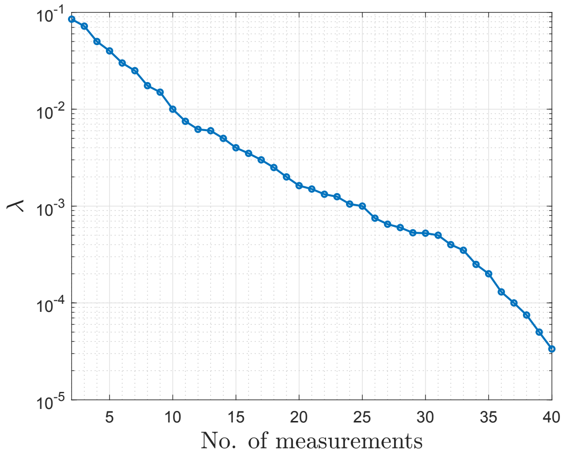


Figure 4.4: λ vs. no. of measurements for the binary distillation column when using the re-weighted l_1 norm.

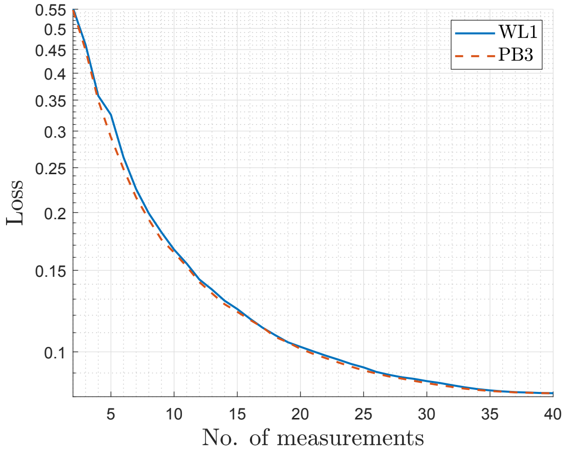


Figure 4.5: Steady-state loss vs. the number of measurements for the partial bidirectional branch and bound (PB3) method and the re-weighted l_1 norm algorithm (WL1) implemented on the binary distillation column.

number of measurements for the re-weighted l_1 norm is compared to the bidirectional branch and bound (PB3) method [75]. The PB3 method is able to give a slightly smaller loss but the CPU time required to find a solution is less affected by the desired subset size when using the re-weighted l_1 norm algorithm, as seen in Figure 4.6. Thus, it might be preferable to use the re-weighted l_1 norm approach when dealing with very large measurement sets. However, this assumes that the appropriate values of λ are known *a priori*.

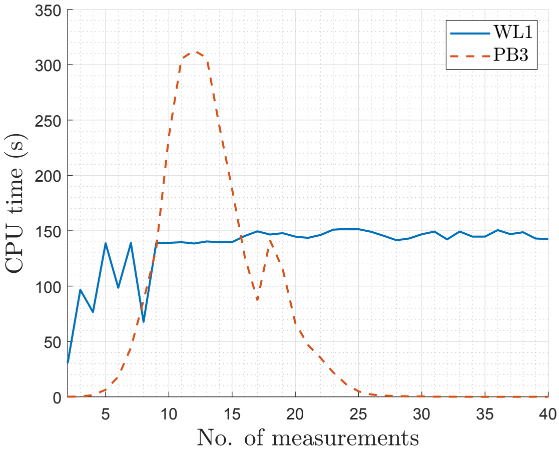


Figure 4.6: CPU time vs. the number of measurements for the partial bidirectional branch and bound (PB3) method and the re-weighted l_1 norm algorithm (WL1) implemented on the binary distillation column.

4.5 Conclusion

Selecting the best subset of size n from n_y measurements with the aim to minimize the loss in (3.26) is a combinatorial optimization problem. The problem has been solved in [75] by using a branch and bound algorithm, whereas an MIQP approach was proposed in [149]. This chapter gives a brief overview of these two methods, after which an alternative approach is investigated. The alternative approach recursively solves a re-weighted l_1 norm optimization problem, where a penalty parameter is included to penalize the number of measurements used. The re-weighted l_1 norm is used as a convex relaxation to the non-convex measurement selection problem and, therefore, the proposed algorithm can't guarantee a globally optimal solution as it converges to a local minimum.

A non-trivial example of a distillation column demonstrates that good results can be found both with respect to the (near) optimality of the steady-state loss, and the CPU time required to find the solution. The existing PB3 algorithm from [75] is capable of finding solutions with a lower loss compared to the re-weighted l_1 norm. However, the re-weighted l_1 norm seems to be less sensitive to the size of the optimization problem. Therefore, it can be used to provide an easily computable (and early) upper bound for the PB3 algorithm, which for large problems, can reduce the computational demands. Another potential advantage of the proposed re-weighted l_1 norm algorithm is the ability to include additional weights that penalize certain control structures, e.g., by including the cost of implementing certain measurements. Thus, making it more flexible and easier to combine with other algorithms, which will be demonstrated in Chapter 8.

Chapter 5

Structural constraints

The lowest steady-state loss can be achieved when the measurement combination H is a full matrix. However, in many practical cases, it may be preferable to impose certain structural constraints on the measurement combinations. These structural constraints may be needed to avoid pairing MVs to CVs that are located far apart, causing long time delay and thus, reducing the dynamic controllability. Furthermore, combining measurements that are of similar type (e.g., combining several temperature measurements) may be preferred by the operators as it has a more intuitive physical meaning. Additionally, when using decentralized control, it is often desirable to impose some structural constraints on the CVs, e.g., by only combining MVs with CVs associated with certain units or parts of the process.

These structural constraints on the measurement combination matrix H should be included in (3.31), such that the optimization problem becomes:

$$\min_{H} \frac{1}{2}\left\| J_{uu}^{1/2}(HG^{y})^{-1}HY \right\|_{F}^{2} \tag{5.1}$$

$$\text{subject to} \;\; H \in \mathcal{S}, \tag{5.2}$$

where $\mathcal{S}$ denotes the structural constraints that are imposed on H. Unfortunately, it is not possible to reformulate (5.1) and (5.2) such that they become a convex problem as in (3.32) and (3.33). This is due to there not being enough degrees of freedom to make $HG^{y} = J_{uu}^{1/2}$ when H is forced to have a certain structure.

To address this problem, a generalized singular value decomposition (GSVD) approach was proposed by [59] whereas in [147], and [148] two convex approximation methods for solving (5.1) and (5.2) were suggested. However, none of these methods are able to guarantee a global optimum, but rather an upper bound for the loss when using structural constraints. Here, an alternative approach is proposed, where an alternating direction method of multipliers (ADMM) is used for imposing structural constraints on the measurement combination H. However, first a brief summary of the two approximation methods is given since they seem to able to give a tighter upper bound on the loss compared to the GSVD method [146]. Therefore, they will be used as references when evaluating the proposed ADMM method.

5.1 Approximation methods

Based on the observation in Theorem 3.2, which stated that the optimal solution to the measurement selection matrix H is non-unique, Yelchuru and Skogestad derived two approximation methods for solving (5.1), and (5.2). Both these have based on numerical experiments shown to provide good upper bounds for the loss when there are structural constraints imposed on H. An overview of these two approaches will be given below, but for a more detailed description the reader is referred to [147], [148], and [146].

5.1.1 Approximation method 1

The optimal solution to (5.1) and (5.2) is non-unique. Therefore, if H is a solution, then there exists $\hat{H} = QH$, which gives the same loss as long as the non-singular matrix $Q \in \mathbb{R}^{n_u \times n_u}$ preserves the desired structure on QH. E.g., if the desired structure for H is block diagonal

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & 0 & 0 \\ 0 & 0 & 0 & h_{2,4} & h_{2,5} \end{bmatrix}, \quad \text{then} \quad Q = \begin{bmatrix} q_{1,1} & 0 \\ 0 & q_{2,2} \end{bmatrix} \quad \text{gives}$$
$$QH = \begin{bmatrix} q_{1,1}h_{1,1} & q_{1,1}h_{1,2} & q_{1,1}h_{1,3} & 0 & 0 \\ 0 & 0 & 0 & q_{2,2}h_{2,4} & q_{2,2}h_{2,5} \end{bmatrix}.$$

Similarly, if H is constrained to a triangular structure, e.g.,

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} & h_{1,5} \\ 0 & 0 & 0 & h_{2,4} & h_{2,5} \end{bmatrix}, \quad \text{then} \quad Q = \begin{bmatrix} q_{1,1} & q_{1,2} \\ 0 & q_{2,2} \end{bmatrix} \quad \text{gives}$$
$$QH = \begin{bmatrix} q_{1,1}h_{1,1} & q_{1,1}h_{1,2} & q_{1,1}h_{1,3} & q_{1,1}h_{1,4} + q_{1,2}h_{2,4} & q_{1,1}h_{1,5} + q_{1,2}h_{2,5} \\ 0 & 0 & 0 & q_{2,2}h_{2,4} & q_{2,2}h_{2,5} \end{bmatrix}.$$

Denoting the available non-zero elements in Q as n_{nz}, then these additional degrees of freedom can be used to match the n_{nz} elements of HG^y to the elements of $J_{uu}^{1/2}$. However, this requires that there are no zero elements in J_{uu} to avoid getting the solution $H = 0$.

To take advantage of the additional n_{nz} degrees of freedom, a binary vector $\beta_\delta = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_{n_u n_u} \end{bmatrix}$ was introduced in [147], where $\beta_l \in 0, 1$ corresponds to the elements of HG^y. For a column-wise vectorization of HG^y (i.e., $vec(HG^y)_l$), each element l is associated with the binary variable β_l. Whenever the lth element of $vec(HG^y)$ gets matched with the lth element of $vec(J_{uu}^{1/2})$, the corresponding β_l is set to 1, otherwise, it remains 0. Thus, the non-zero elements in Q are used to match between n_u and n_{nz} elements of $vec(HG^y)$ to $vec(J_{uu}^{1/2})$, where $\sum_{l=1}^{n_u n_u} \beta_l$ should be kept between n_u and n_{nz}. The remaining unmatched elements in $vec(HG^y)$ are allowed to vary between $-b$ and b, which are formulated using the big-M approach.

The resulting a MIQP can then be formulated as

$$\min_{H,\beta_\delta} \frac{1}{2}\left\| HY \right\|_F^2 \tag{5.3}$$

$$\text{subject to } \; H \in \mathcal{S} \tag{5.4}$$

$$-b(1-\beta_l) \leq vec(HG^y - J^{1/2})_l \leq b(1-\beta_l), \;\; \forall l = 1, 2, ..., n_u n_u \tag{5.5}$$

$$n_u \leq \sum_{l=1}^{n_u n_u} \beta_l \leq n_{n_z} \tag{5.6}$$

$$n_{u_k} \leq \sum_{p=0}^{n_u - 1} \sum_{j=\sum_k n_{u_{k-1}}+1}^{\sum_k n_{u_k}} \beta_l n_u p + j \leq n_{nz_k}, \;\; \forall k = 1, 2, ..., \text{number of blocks} \tag{5.7}$$

where n_{u_k}, and n_{nz_k} are the number of inputs, and the number of non-zeros in Q in block k respectively.

The above approximation method was in [146] referred to as a convex approximation method. However, this is incorrect, since an optimization formulation involving binary variables can't be convex. In addition, using the big-M formulation may not be numerically robust, especially for large values on M.

The optimization in (5.3)−(5.7) can be extended to find subsets of measurements with a specified structure on H by first vectorizing (5.3), (5.4); and then including the big-M constraints from (4.11).

5.1.2 Approximation method 2

For the second method, the equality constraint ($HG^y = J_{uu}^{1/2}$) in (3.33) is replaced with the element-wise inequality constraint, $HG^y \leq J_{uu}^{1/2}$. The approximation can then be written as:

$$\min_{H} \frac{1}{2}\left\| HY \right\|_F^2 \tag{5.8}$$

$$\text{subject to } \; HG^y \leq J_{uu}^{1/2} \tag{5.9}$$

$$H \in \mathcal{S} \tag{5.10}$$

The relaxation of the constraint in (5.9) provides H with additional degrees of freedom that ensures a feasible solution exists when there are structural constraints on H. While there is no available proof, numerical experiments have shown that replacing the constraint ($HG^y = J_{uu}^{1/2}$) in (3.33) with $HG^y \leq J_{uu}^{1/2}$ in (5.9) results in the same solution when there are no structural constraints on H [148]. However, to avoid getting $H = 0$, $J_{uu}^{1/2}$ has to have some negative elements in each row. If there are no negative elements in $J_{uu}^{1/2}$, then it can be modified such that

$$\hat{J}_{uu} = Q_u^T J_{uu} Q_u, \tag{5.11}$$

where $Q_u \in \mathbb{R}^{n_u \times n_u}$ is any non-singular matrix that makes $\hat{J}_{uu}$ contain some negative elements.

The problem formulation in (5.8)–(5.10) can easily be extended to find a subset of n variables with the desired structure on H by adding the big-M constraints from (4.11) and vectorizing the problem (5.8)–(5.10).

5.2 ADMM for structured measurement combinations

The two approximation methods are only able to guarantee an upper bound on the loss when the measurement combination matrix H is forced to have a particular structure. Therefore, an alternating direction method of multipliers (ADMM) algorithm is proposed for incorporating structural constraints on the CVs. The hope that it may be able to provide better results compared to the other existing methods when obtaining structured measurement combinations.

5.2.1 Alternating direction method of multipliers

ADMM has extensively been studied since it was introduced in the 1970s [46] and has shown to be a simple, yet a robust algorithm that is well suited for distributed convex optimization in large-scale problems. However, it can also be extended to non-convex problems, see [21] for a survey of ADMM and its different applications. Recently, there has also been an increasing interest in using ADMM for sparsity promoting functions [88], [34].

There are many ways of describing the construction of the ADMM since it is closely related to several other methods. However, the procedure given in, e.g., [21] is one of the more common procedures of explaining ADMM and will thus, be used here. This entails first, giving a short review over some of its precursors.

Dual ascent algorithm

Consider the following convex optimization problem

$$\min_{x} f(x) \tag{5.12}$$

$$\text{subject to } \; Ax = b \tag{5.13}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, and $f(\cdot)$ is a convex function. The Lagrangian for the above problem is equivalent to

$$\mathcal{L}(x,y) = f(x) + y^T(Ax - b), \tag{5.14}$$

with $y \in \mathbb{R}^m$ being the dual variable. The dual function is given by,

$$g(y) = \inf_{x} \mathcal{L}(x,y) = -f^*(-A^T y) - b^T y, \tag{5.15}$$

where f^* is known as the *conjugate* of the function f [20]. The resulting dual problem can then be formulated as

$$\max_{y} g(y). \tag{5.16}$$

Assuming that strong duality holds, then the optimal values for both the dual and the primal problems should be the same. Therefore, from the optimal dual point y^*, it is possible to recover the optimal primal point x^* as

$$x^* = \arg\min_x \mathcal{L}(x, y^*). \tag{5.17}$$

In the dual ascent algorithm, the dual problem is solved using a gradient ascent method. Denoting $\nabla g(y)$ as the gradient of $g(y)$ Assuming $g(\cdot)$ is differentiable, the gradient $\nabla g(y)$ can be evaluated as follows:

$$\nabla_y g(y) = \nabla_y \big(\inf_x \mathcal{L}(x, y)\big) = \nabla\big(\mathcal{L}(x^*, y)\big) = \nabla_y f(x^*) + \nabla_y y^T \big(Ax - b\big) = Ax - b. \tag{5.18}$$

The dual ascent algorithm consists of iteratively updating the following steps

$$x^{k+1} := \arg\min_x \mathcal{L}(x, y^k), \tag{5.19}$$

$$y^{k+1} := y^k + \eta^k \big(Ax^{k+1} - b\big), \tag{5.20}$$

where $\eta^k > 0$ is a scalar step size and the superscript k is the iteration counter. The algorithm is called dual ascent since if η^k is chosen appropriately, then the dual function will increase after each iteration such that $g(y^{k+1}) > g(y^k)$.

Dual decomposition

A great benefit of the dual ascent method is that for certain cases is it can be used for decentralized implementation. E.g., assuming that the cost function $f(\cdot)$ is separable, i.e.,

$$f(x) = \sum_{i=1}^{N} f_i(x_i), \tag{5.21}$$

where x can be split into N different sub vectors such that $x = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}$. Decomposing the matrix $A = \begin{bmatrix} A_1 & \cdots & A_N \end{bmatrix}$ such that $A = \sum_{i=1}^{N} A_i x_i$, then the Lagrangian can be written as

$$\mathcal{L}(x, y) = \sum_{i=1}^{N} \mathcal{L}_i(x_i, y) = \sum_{i=1}^{N} \Big(f_i(x_i) + y^T A_i x_i - \frac{1}{N} y^T b\Big). \tag{5.22}$$

Thus, the Lagrangian can also be separable in x, which means that the step in (5.19) can be split into N different problems that can be solved in parallel. The resulting algorithm is referred to as dual decomposition and is formulated as

$$x^{k+1} := \arg\min_x \mathcal{L}_i(x_i, y^k), \tag{5.23}$$

$$y^{k+1} := y^k + \eta^k \big(Ax^{k+1} - b\big), \tag{5.24}$$

where the x minimization problem has been split into N sub problems that can be solved independently.

Augmented Lagrangian and the method of multipliers

The dual ascent method requires that the function $f(\cdot)$ in (5.12) to be strictly convex and finite. This restriction motivated the development of the augmented Lagrangian method, which offers a more robust alternative that is able to yield convergence without less strict assumptions on $f(\cdot)$. The augmented Lagrangian function for (5.12) is defined as

$$\mathcal{L}_\rho(x,y) := f(x) + y^T\big(Ax - b\big) + \frac{\rho}{2}\big\|Ax - b\big\|_2^2, \tag{5.25}$$

where $\rho > 0$ is a positive scalar value called the penalty parameter. The augmented Lagrangian can be seen as the Lagrangian associated with the following problem:

$$\min_x f(x) + \frac{\rho}{2}\big\|Ax - b\big\|_2^2 \tag{5.26}$$

$$\text{subject to } \; Ax = b. \tag{5.27}$$

The solution to this problem is clearly equivalent to the original problem described in (5.12), and (5.13) since the added penalty term $\big(\rho/2\big)\big\|Ax - b\big\|_F^2$ is zero for any feasible solution to x. Applying the dual ascent approach to the modified problem yields

$$x^{k+1} := \arg\min_x \mathcal{L}_\rho(x, y^k), \tag{5.28}$$

$$y^{k+1} := y^k + \eta^k\big(Ax^{k+1} - b\big), \tag{5.29}$$

which is known as the method of multipliers. The benefit of including the quadratic penalty term is that the algorithm is able to converge under much milder conditions, compared to the dual descent. However, the improved convergence properties come at a cost. The main disadvantage is that with the introduced penalty term, it is no longer possible to split the Lagrangian function and, thus, the method of multipliers can not be used for decomposition. This drawback can be overcome using ADMM.

Alternating direction method of multipliers

Alternating direction of multipliers (ADMM) is an algorithm that solves optimization problems by partitioning the decision variables into two groups. The algorithm solves problems of the form

$$\min_{x,z} \; f(x) + g(z) \tag{5.30}$$

$$\text{subject to } \; Ax + Bz = c, \tag{5.31}$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$, and $c \in \mathbb{R}^p$. The functions $f(\cdot)$ and $g(\cdot)$ are assumed to be a convex and separable across the decision variable x and z. The ability to split the objective into two parts (x and z) is what constitutes the main difference between the initial problem in (5.12)–(5.13) and the above problem.

Similar, to the method of multipliers the augmented Lagrangian for the problem in (5.30)–(5.31) can be formulated as

$$\mathcal{L}_\rho(x, z, y) = f(x) + g(z) + y^T\big(Ax + Bz - c\big) + \frac{\rho}{2}\big\|Ax + Bz - c\big\|_2^2. \tag{5.32}$$

The ADMM algorithm solves the problem in (5.30)–(5.31) by iteratively solving,

$$x^{k+1} := \arg\min_x \mathcal{L}_\rho(x, z^k, y^k), \tag{5.33}$$

$$z^{k+1} := \arg\min_z \mathcal{L}_\rho(x^{k+1}, z, y^k), \tag{5.34}$$

$$y^{k+1} := y^k + \rho\big(Ax^{k+1} + Bz^{k+1} - c\big), \tag{5.35}$$

where $\rho > 0$. The ADMM algorithm closely resembles dual ascent and the method of multipliers. Fist, it minimizes (5.32) with respect to x in (5.33), and then it minimizes (5.32) with respect to z in (5.34). A dual update is performed in (5.35), using a step size equal to the augmented Lagrangian parameter ρ. Thus, the optimal x^* and z^* are solved in an alternating fashion, hence the name alternating direction.

The ADMM can be expressed more conveniently by combining the linear and quadratic terms for the augmented Lagrangian in (5.32) and changing the dual variable y to the scaled dual variable $\Lambda = \rho^{-1}y$. By defining the residual,

$$r := Ax + Bz - c, \tag{5.36}$$

the augmented Lagrangian in (5.32) can be rewritten to

$$\mathcal{L}_\rho(x, z, \Lambda) = f(x) + g(z) + \frac{\rho}{2}\big\|r\big\|_2^2 + \rho\Lambda^T r. \tag{5.37}$$

Since

$$\frac{\rho}{2}\big\|r + \Lambda\big\|_2^2 = \frac{\rho}{2}\big(r + \Lambda\big)^T\big(r + \Lambda\big) = \frac{\rho}{2}\big\|r\big\|_2^2 + \rho\Lambda^T r + \frac{\rho}{2}\big\|\Lambda\big\|_2^2 \tag{5.38}$$

and, thus,

$$\frac{\rho}{2}\big\|r\big\|_2^2 + \rho\Lambda^T r = \frac{\rho}{2}\big\|r + \Lambda\big\|_2^2 - \frac{\rho}{2}\big\|\Lambda\big\|_2^2. \tag{5.39}$$

The term $\big(\frac{\rho}{2}\big\|r\big\|_2^2 + \rho\Lambda^T r\big)$ in (5.37) can be replaced with (5.39), resulting in the scaled Lagrangian:

$$\mathcal{L}_\rho(x, z, \Lambda) = f(x) + g(z) + \frac{\rho}{2}\big\|Ax + Bz - c + \Lambda\big\|_2^2 - \big\|\Lambda\big\|_2^2 \tag{5.40}$$

The solution for minimizing $\mathcal{L}_\rho(x, z, \Lambda)$ with respect to x and z is independent on the final term $\|\Lambda\|_2^2$. Therefore, the iterations can be written as

$$x^{k+1} := \arg\min_x f(x) + \frac{\rho}{2}\big\|Ax + Bz^k - c + \Lambda\big\|_2^2, \tag{5.41}$$

$$z^{k+1} := \arg\min_z g(z) + \frac{\rho}{2}\big\|Ax^{k+1} + Bz - c + \Lambda\big\|_2^2, \tag{5.42}$$

$$\Lambda^{k+1} := \Lambda^k + Ax^{k+1} + Bz^{k+1} - c. \tag{5.43}$$

The scaled form of the ADMM (5.41)−(5.43) is equivalent to its unscaled form in (5.33)−(5.35), but the formulas for the scaled form are often shorter and more convenient to work with. In particular, it can be observed that ρ no longer appears when updating the step size in (5.43), which is one of the main reason it is preferred over the unscaled version.

5.2.2 The ADMM algorithm for imposing structural constraints

As previously mentioned, it is not possible to reformulate (5.1) and (5.2) such that it becomes a convex optimization problem as in (3.32) and (3.33). This is due to there not being enough degrees of freedom to make $HG^y = J_{uu}^{1/2}$ when H is forced to have a specific structure. To solve this problem two convex relaxation methods were proposed in [147], and [148] that finds an upper bound for the loss when structural constraints are imposed on the measurement matrix H. Here, an alternative approach is suggested where the aim is to further reduce the loss for measurement combinations with structural constraints by adapting the ADMM algorithm presented in Section 5.2.1 .

Lemma 5.1. *Under the assumption that $\hat{H}$ is the optimal measurement combination that minimizes (5.1) with the structural constraints (5.2) where $\hat{H}G^y$ is full rank. Then, there exists a non-singular matrix $Q \in \mathbb{R}^{n_u \times n_u}$ and a full matrix $H \in \mathbb{R}^{n_u \times n_y}$ such that*

$$H = Q\hat{H}, \tag{5.44}$$

where H satisfies

$$HG^y = J_{uu}^{1/2}. \tag{5.45}$$

Proof. From Theorem 3.2, it was shown that the optimal solution for H is non-unique, and for a non-singular matrix $Q \in \mathbb{R}^{n_u \times n_u}$

$$H = Q\hat{H} \tag{5.46}$$

gives the same loss. Setting $Q = J_{uu}^{1/2}\big(\hat{H}G^y\big)^{-1}$ then

$$HG^y = Q\hat{H}G^y = J_{uu}^{1/2}. \tag{5.47}$$

Thus, while there may not be enough degrees of freedom to make $HG^y = J_{uu}^{1/2}$ when structural constraints are imposed on $\hat{H}$, it is possible to find a non-singular matrix Q such that $Q\hat{H}G^y = J_{uu}^{1/2}$. □

Using Lemma 5.1, the nonlinear problem (5.1) and (5.2) is reformulated as shown in the following theorem.

Theorem 5.2. *If for some structural constraints $\mathcal{S}$, there exists measurement combinations $\hat{H} \in \mathbb{R}^{n_u \times n_y}$, $H \in \mathbb{R}^{n_u \times n_y}$; and a non-singular matrix $Q \in \mathbb{R}^{n_u \times n_u}$,* *where both $\left(HG^y\right)$, and $\left(Q\hat{H}G^y\right)$ are full rank. Then the optimization problem in (5.1) and (5.2) can be reformulated as follows:*

$$\min_{H,Q,\hat{H}} \frac{1}{2}\left\|HY\right\|_F^2 \tag{5.48}$$

$$\textit{subject to } HG^y = J_{uu}^{1/2} \tag{5.49}$$

$$H = Q\hat{H} \tag{5.50}$$

$$\hat{H} \in \mathcal{S} \tag{5.51}$$

Proof. When there are no structural constraints (H is a full matrix), it was shown in Theorem 3.2 that minimizing the loss in (5.1) is equivalent of solving (5.48) subject to (5.49). Furthermore, Lemma 5.1, implies that by imposing structural constraints on $\hat{H}$, it is possible to find a non-singular matrix Q such that (5.50) and (5.51) are satisfied with H, and $\hat{H}$ resulting in the same loss. □

The inclusion of the additional decision variables Q and $\hat{H}$ gives H enough degrees of freedom to satisfy the constraint in (5.49), while Q can be computed such that $Q^{-1}H$ (and $\hat{H}$) has the desired structure. Unfortunately, this is still a non-convex problem due to the bilinear constraints in (5.50) and can, therefore, be difficult to solve. However, numerical experiments indicate that this problem translates well to using alternating direction method of multipliers (ADMM).

The augmented Lagrangian associated with (5.48) and the constraint (5.50) can be formulated as

$$\mathcal{L}_\rho(H, Q, \hat{H}, \Lambda) = \frac{1}{2}\left\|HY\right\|_F^2 + \frac{\rho}{2}\left\|H - Q\hat{H} + \Lambda\right\|_F^2, \tag{5.52}$$

where $\Lambda \in \mathbb{R}^{n_u \times n_y}$ is the dual variable (Lagrange multiplier), and ρ is a positive scalar. The ADMM algorithm solves the problem in (5.48)–(5.51), by iteratively solving,

$$\hat{H}^{k+1} := \arg\min_{H,\hat{H}} \mathcal{L}_\rho(H, Q^k, \hat{H}, \Lambda^k), \tag{5.53}$$

$$\text{subject to } (5.49), \text{and } (5.51)$$

$$H^{k+1}, Q^{k+1} := \arg\min_{H,Q} \mathcal{L}_\rho(H, Q, \hat{H}^{k+1}, \Lambda^k), \tag{5.54}$$

$$\text{subject to } (5.49)$$

$$\Lambda^{k+1} := \Lambda^k + H^{k+1} - Q^{k+1}\hat{H}^{k+1} \tag{5.55}$$

until it converges.

There are two major benefits for using ADMM on (5.48)–(5.51). First, it temporarily relaxes the equality constraints in (5.50), thus, allows for more flexibility when searching for the optimal solution. Secondly, by separating $\hat{H}$ from Q, both the step in (5.53) and the step in (5.54) become convex quadratic optimization problems with equality constraints. Therefore, there exist analytical solutions to both these steps for improved computational efficiency.

Analytical solution to the (5.54) subproblem

After dropping superscripts for notational simplicity, the H, Q minimization step in (5.54) becomes

$$\min_{H,Q} \ = \frac{1}{2}\left\|HY\right\|_F^2 + \frac{\rho}{2}\left\|H - Q\hat{H} + \Lambda\right\|_F^2 \tag{5.56}$$

$$\text{subject to } \ HG^y = J_{uu}^{1/2}. \tag{5.57}$$

Theorem 5.3. *Under the assumption that YY^T is full rank and $\hat{H}$ is of full row rank, then an analytical solution to (5.56) can be obtained for H and Q:*

$$\begin{aligned} H^T &= (\phi^{-1} - \phi^{-1}G^y({G^y}^T\phi^{-1}G^y)^{-1}{G^y}^T\phi^{-1})\chi \\ &\quad + (\phi^{-1}G^y({G^y}^T\phi^{-1}G^y)^{-1})J_{uu}^{1/2}, \end{aligned} \tag{5.58}$$

$$Q^T = (\hat{H}\hat{H}^T)^{-1}\hat{H}(\Lambda^T + H^T), \tag{5.59}$$

where

$$\begin{aligned} \phi &:= YY^T + \rho I - \rho\hat{H}^T(\hat{H}\hat{H}^T)^{-1}\hat{H}, \\ \chi &:= \hat{H}^T(\hat{H}\hat{H}^T)^{-1}\hat{H}\Lambda^T - \Lambda^T. \end{aligned}$$

Proof. Solving (5.56) with respect to Q is an unconstrained quadratic optimization problem, and the optimal solution can be obtained from:

$$\frac{\partial\mathcal{L}}{\partial Q} = (\hat{H}\hat{H}^T)^TQ^T - \hat{H}\Lambda^T - \hat{H}H^T = 0,$$

for which the solution for Q^T is equivalent to (5.59). To find a solution for (5.56) and (5.57) with respect to H, the problem must satisfy the following KKT-conditions [107]:

$$\begin{bmatrix} YY^T + \rho I & -G^y \\ {G^y}^T & 0 \end{bmatrix}\begin{bmatrix} H^T \\ \lambda_{\mathcal{L}}^T \end{bmatrix} = \begin{bmatrix} \rho\hat{H}^TQ^T - \Lambda^T \\ J_{uu}^{1/2} \end{bmatrix}, \tag{5.60}$$

where $\lambda_{\mathcal{L}}$ is the Lagrange multiplier for the constraint in (5.57). Replacing Q with the solution from (5.59), the KKT conditions in (5.60) can be rewritten to:

$$\begin{bmatrix} \phi & -G^y \\ {G^y}^T & 0 \end{bmatrix}\begin{bmatrix} H^T \\ \lambda_{\mathcal{L}}^T \end{bmatrix} = \begin{bmatrix} \chi \\ J_{uu}^{1/2} \end{bmatrix}. \tag{5.61}$$

The optimal H^T can then be found by inverting the KKT-matrix using the Schur complement for the inverse of block partitioned matrices (see, e.g., [91]). □

Analytical solution to the (5.53) subproblem

The solution to (5.53) is identical to solving

$$\min_{H,\hat{H}} \; = \frac{1}{2}\left\|HY\right\|_F^2 + \frac{\rho}{2}\left\|H - Q\hat{H} + \Lambda\right\|_F^2 \tag{5.62}$$

$$\text{subject to } HG^y = J_{uu}^{1/2} \tag{5.63}$$

$$\Gamma_{\mathcal{S}} \; vec(\hat{H}^T) = 0 \tag{5.64}$$

with $vec(\hat{H}^T)$ being the vectorization of $\hat{H}^T$. $\Gamma_{\mathcal{S}}$ is a matrix consisting of only ones and zeros that is the orthogonal complement to the structural constraints imposed on $vec(\hat{H}^T)$. E.g., if the structural constraints for $\hat{H}$ are:

$$\hat{H} \in \mathcal{S} := \begin{bmatrix} \hat{h}_{1,1} & 0 & \hat{h}_{1,3} \\ 0 & \hat{h}_{2,2} & 0 \end{bmatrix},$$

then $\Gamma_{\mathcal{S}}$ should be chosen such that $\Gamma_{\mathcal{S}} \; vec(\hat{H}^T)$ contains the elements that should be set to zero, i.e.,

$$\Gamma_{\mathcal{S}} \; vec(\hat{H}^T) = 0 \equiv \begin{bmatrix} \hat{h}_{1,2} & \hat{h}_{2,1} & \hat{h}_{2,3} \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T.$$

Theorem 5.4. *Under the assumption of Q, and YY^T being full rank, then an analytical solution for (5.62)–(5.64) can be obtained for $\hat{H}$ and H:*

$$vec(\hat{H}^T) = \left(\zeta^{-1} - \zeta^{-1}\Gamma_{\mathcal{S}}^T\left(\Gamma_{\mathcal{S}}\zeta^{-1}\Gamma_{\mathcal{S}}^T\right)^{-1}\Gamma_{\mathcal{S}}\zeta^{-1}\right) \; vec(\theta), \tag{5.65}$$

$$H^T = \Upsilon(\rho\hat{H}^TQ^T - \Lambda^T) + \psi^{-1}G^y(G^{y^T}\psi^{-1}G^y)^{-1}J_{uu}^{1/2}, \tag{5.66}$$

where θ, ψ, and Υ are defined as

$$\theta := \Upsilon\Lambda^TQ - \psi^{-1}G^y(G^{y^T}\psi^{-1}G^y)^{-1}J_{uu}^{1/2}Q - \Lambda^TQ, \tag{5.67}$$

$$\Upsilon := \psi^{-1} - \psi^{-1}G^y(G^{y^T}\psi^{-1}G^y)^{-1}G^{y^T}\psi^{-1}, \tag{5.68}$$

$$\psi := YY^T + \rho I, \tag{5.69}$$

and ζ is given by

$$\zeta = Q^TQ \otimes (\rho\Upsilon - I), \tag{5.70}$$

with $\otimes$ being the Kronecker product.

Proof. The proof for (5.65), and (5.66) follows a similar procedure as in the proof for Theorem 5.3, where the KKT conditions are first formulated to solve (5.62)-–(5.64) with respect to H, and $vec(\hat{H}^T)$. The resulting KKT matrices with respect to H is equivalent to the one in (5.60). Solving (5.60) for H and replacing the results in the KKT conditions for $vec(\hat{H}^T)$ gives,

$$\begin{bmatrix} \zeta & \Gamma_{\mathcal{S}}^T \\ -\Gamma_{\mathcal{S}} & 0 \end{bmatrix} \begin{bmatrix} vec(\hat{H}^T) \\ \lambda_{\mathcal{L}} \end{bmatrix} = \begin{bmatrix} vec(\theta) \\ 0 \end{bmatrix} \tag{5.71}$$

from which the results in (5.65) can be obtained. □

5.2.3 The proposed ADMM algorithm

The resulting ADMM algorithm can be seen in Algorithm 2, where first an initial value for Q (e.g., $Q = I$) needs to be set, together with a positive scalar for the parameter ρ.

Algorithm 2 ADMM for structural constraints.

```
Initialize: For $k = 1$, select a non-singular matrix $Q$, (e.g., $Q^k = I$) and choose a
    sufficiently large positive scalar $\rho$.
1: Calculate $\hat{H}^{k+1}$ using (5.65).
2: Calculate $H^{k+1}$, and $Q^{k+1}$ using (5.58) and (5.59).
3: Update $\Lambda^{k+1}$ using (5.55).
4: If the maximum number of iterations has been reached, or if both $\|H^{k+1} - Q^{k+1}\hat{H}^{k+1}\|_F^2 \le \epsilon_1$ and $\|H^{k+1} - H^k\|_F^2 \le \epsilon_2$ are satisfied then stop. Otherwise,
    set $k$ to $k+1$ and repeat step 1 to 4.
```

It is important to note that for non-convex problems, the ADMM algorithm may not converge to the globally optimal solution; in fact, it may not converge at all. Thus, it should only be considered as a local optimization method. Whereas global convergence of ADMM can be guaranteed for convex problems, this is not the case when dealing with non-convex problems. However, the ADMM algorithm seems to be able to converge in most cases as long as the value of ρ is chosen to be sufficiently large. Furthermore, even when it converges, the final result can depend on how the initial values Q^k and ρ were chosen. However, since all the steps can be solved analytically, and a solution can be obtained relatively fast, it should be easy to try different initial values until a suitable solution has been found. Therefore, the hope is that it may be able to provide better results than other methods when obtaining structured measurement combinations.

5.2.4 Convergence

For convex problems, convergence for ADMM has been proved and illustrated in, e.g., [21] and [38] as long as $\rho > 0$ and is fixed. While it is sufficient to select $\rho > 0$ to guarantee convergence for convex cases, the value of ρ still plays an important role in the performance of the algorithm when implemented on different applications. Where a poorly chosen value for ρ is often associated with slow convergence. Since determining what constitutes good values for ρ are difficult to know *a priori*, several schemes have been proposed that vary $\rho = \rho^k$ across the iterations [58], [144], [49]. In general, these methods work by either increasing or decreasing the value of ρ^k whenever some specified conditions are satisfied, e.g.,

$$\rho^{k+1} = \begin{cases} \tau\rho^k, & \text{if } \|x^{k+1} - y^{k+1}\|_2 < \mu\|y_{k+1} - y_k\|_2 \\ \tau^{-1}\rho^k, & \text{if } \|y^{k+1} - y^k\|_2 < \mu\|x_{k+1} - y_{k+1}\|_2 \\ \rho^k, & \text{otherwise} \end{cases} \tag{5.72}$$

where, $\tau > 1$ and $\mu > 1$ are constants. However, convergence can only be guaranteed if ρ^k will, in the end, be held at a fixed value.

While convergence to a globally optimal solution can be guaranteed for convex problems, the convergence for ADMM when solving non-convex problems remains an open topic of research. Despite the lack of convergence proofs, it has been demonstrated that ADMM works extremely well on many applications with non-convex objectives [88], [21]. The general consensus is that ADMM converges as long as ρ has been chosen sufficiently large. This seemed to be due to the quadratic term $\frac{\rho}{2}\left\|Ax + Bz - c\right\|_2^2$ in (5.32) tend convexify the objective function locally when ρ is selected large enough. For a number of different nonlinear problems, the convergence has recently been proved in [142], and [60], where [142] showed convergence related to nonlinear consensus and sharing problems. A convergence proof for a family of non-convex problems under more general conditions was shown in [60].

These problems are all been dealing with the assumptions that the constraints are linear and thus when the constraints are bilinear as in (5.53)−(5.55), the same convergence proofs may no longer hold. However, recently W. Gao et al. showed in [50] that ADMM employed on non-convex problems with multi-affine constraints also converges to a set of stationary points as long as the penalty parameter ρ has been chosen sufficiently large. Numerical experiments seem to confirm this for the problem in (5.53)−(5.55), as it is able to converge when ρ has been chosen large enough. The smallest required value for ρ to ensure convergence has been determined in [54] for a non-negative matrix factorization problem, and in [131] a rank constrained optimization problem. However, these convergence results can't directly be adapted for the problem in (5.53)−(5.55), due to the additional constraints in (5.49) and (5.51). Thus, determining what constitutes a sufficiently large ρ to solve (5.53)−(5.55) is difficult to know *a priori*. To further improve the ADMM algorithm, a scheme is proposed that automatically updates the penalty parameter ρ such that it converges to a set of stationary points.

5.2.5 Adaptive penalty parameter

The larger the value of the penalty parameter ρ, for the problem in (5.53)−(5.55) the more the algorithm prioritizes satisfying the structural constraints versus reducing the steady-state loss. Therefore, choosing a very large value of ρ will more likely ensure convergence of Algorithm 2, but it may result in a measurement combination H that gives an unnecessary high loss. If $\rho^k \to \infty$, then the optimization problem essentially becomes:

$$\min_{H^{k+1}, Q^{k+1}, \hat{H}^{k+1}} \frac{\rho}{2}\left\|H^{k+1} - Q^{k+1}\hat{H}^{k+1} + \Lambda^k\right\|_F^2 \tag{5.73}$$

$$\text{subject to } H^{k+1}G^y = J_{uu}^{1/2} \tag{5.74}$$

$$\hat{H}^{k+1} \in \mathcal{S} \tag{5.75}$$

For which the optimal solution is

$$(H^{k+1} - Q^{k+1}\hat{H}^{k+1}) \approx -\Lambda^k, \tag{5.76}$$

and thus, updating the dual step in (5.55) becomes

$$\begin{aligned} \Lambda^{k+1} &= \Lambda^k + (H^{k+1} - Q^{k+1}\hat{H}^{k+1}) && (5.77) \\ &\approx \Lambda^k - \Lambda^k = 0 && (5.78) \end{aligned}$$

Therefore, the aim is to try to keep ρ small such that more emphasis is put on minimizing the loss. However, ρ still have to be sufficiently large to ensure that Algorithm 2 converges by forcing the be penalty term Λ^k to decrease such that as $k \to \infty$:

$$\Lambda^{k+1} - \Lambda^k \to 0 \tag{5.79}$$

By excluding ρ from the Lagrangian in (5.52), the following objective is defined:

$$J^k := \frac{1}{2}\left\|H^k Y\right\|_F^2 + \frac{1}{2}\left\|H^k - Q^k\hat{H}^k + \Lambda^{k-1}\right\|_F^2. \tag{5.80}$$

If the value of ρ has been chosen sufficiently large such that for two consecutive iterations the following condition holds,

$$J^{k+1} \leq J^k, \tag{5.81}$$

then the problem in (5.53)−(5.55) should also converge. If the sequence of (5.80) is non-increasing, then Λ^k must also decease over time such that the condition in (5.78) eventually holds, which means,

$$\begin{aligned} H^{k+1} - H^k &\to 0 && (5.82) \\ (H^{k+1} - Q^{k+1}\hat{H}^{k+1}) &\to 0 && (5.83) \end{aligned}$$

as $k \to \infty$. On the other hand, if the sequence of (5.80) keeps increasing, it would imply that Λ^k grows at a rate faster than the loss $\|H^k Y\|_F^2$ is able to decrease. In this case, the value for ρ needs to increase until the condition in (5.81) is satisfied once again.

The resulting update rule for ρ^k is given thus given by

$$\rho^{k+1} = \begin{cases} \tau\rho^k, & \text{if } J^{k-1} - J^k \geq \epsilon \\ \rho^k, & \text{otherwise} \end{cases} \tag{5.84}$$

where $1 >> \epsilon \geq 0$, and $\tau > 1$ (e.g., $\tau = 2$). Note that in contrast to the typical update scheme in (5.72), the value for ρ^k in (5.84) is never allowed to decrease. The proposed ADMM algorithm with an adaptive penalty parameter can be seen in Algorithm 3, where again an initial value for Q^k (e.g., $Q^k = I$) needs to be set, together with an initial positive scalar for the parameter ρ^k. Choosing a small value for the for the initial parameter ρ^k (e.g., $\rho^k = 1$), and then allowing ρ^k to increase until it is sufficiently large such that the problem in (5.53)−(5.55) starts to converge have shown to give good results.

Algorithm 3 ADMM for structural constraints with adaptive penalty parameter

```
Initialize: For $k = 1$, select a non-singular matrix $Q$, (e.g., $Q^k = I$), and choose
    a positive scalar $\rho^k$ (e.g., $\rho^k = 1$).
1: Calculate $\hat{H}^{k+1}$ using (5.65).
2: Calculate $H^{k+1}$, and $Q^{k+1}$ using (5.58) and (5.59).
3: Update $\rho^{k+1}$ according to (5.84).
4: Update $\Lambda^{k+1}$ using (5.55).
5: If the maximum number of iterations has been reached, or if both $\|H^{k+1} - Q^{k+1}\hat{H}^{k+1}\|_F^2 \leq \epsilon_1$ and $\|H^{k+1} - H^k\|_F^2 \leq \epsilon_2$ are satisfied then stop. Otherwise,
    set $k$ to $k+1$ and repeat step 1 to 5.
```

5.2.6 ADMM combined with the branch and bound method

The ADMM algorithm can't guarantee that a globally optimal solution has been found since the underlying problem that it tries to solve in (5.53)−(5.55) is non-convex. However, numerical experiments indicate that it often is able to give solutions that are on pair to using, e.g., global search [136] despite requiring only a fraction of the time. This will be demonstrated in more detail in Section 5.3. In particular, the ADMM algorithm seems to work extremely well when dealing with block diagonal constraints, at which it was repeatedly able to converge to the same solution despite using different initial conditions. This would imply that in most cases, the property of monotonicity also should hold, i.e., the loss can't be reduced by removing elements from the measurement matrix H (setting elements to zero). Therefore, it is possible to combine the proposed ADMM algorithm with a branch and bound type method to search for the optimal subset of measurements when there are structural constraints imposed on the measurement selection matrix H.

When there are no structural constraints, very efficient partially bidirectional branch and bound (PB3) methods, have already been proposed for both the worst case loss in [74], and the average case loss in [75] with publicly available code in [25], and [26] respectively. Therefore, the existing code in [26] will be modified to also account for the structural constraints by having it call the ADMM algorithm. These modifications will briefly be outlined here, whereas, for more details on PB3 algorithm, the reader is referred to [74], and [75].

Let's recall the notation used to describe the branch and bound method in Section 4.2, where X_m denoted an index set. The index set describes a selected measurement subset, with the subscript m indicating how many measurements are included in X. The gain matrix that corresponded to the selected measurement was denoted G_X^y, and similarly, Y_X denoted the Y matrix for the index set X. The optimal measurement combination, obtained using, e.g., (3.36) with index set X is denoted H_X^*, and the resulting average loss is given by the function:

$$L(X_m) = \frac{1}{2}\left\| J_{uu}^{1/2}(H_X^* G_X^y)^{-1} H_X^* Y_X \right\|_F^2. \tag{5.85}$$

Let $\hat{X}_m$ denote an index set for which its optimal measurement combination $\hat{H}_X^*$

contain some structural constraints $\mathcal{S}_X$. The corresponding loss is given by,

$$L(\hat{X}_m) = \frac{1}{2}\left\| J_{uu}^{1/2}(\hat{H}_X^* G_X^y)^{-1}\hat{H}_X^* Y_X \right\|_F^2, \tag{5.86}$$

which is same as in (5.85) but with $\hat{H}_X^*$ obtained using, e.g., the proposed ADMM algorithm for an index set X and the structural constraints $\mathcal{S}_X$. Next, let $\hat{X}_n$ be an n-element subset selected from $\hat{X}_m$. The selection problem for finding the globally optimal n-element subset $\hat{X}_n^*$ that minimizes the loss with the measurement combination $\hat{H}_X$ satisfying the structural constraints can be formulated as

$$L(\hat{X}_n^*) = \min_{\hat{X}_n \subseteq \hat{X}_m} L(\hat{X}_n) \tag{5.87}$$

$$\text{subject to } \hat{H}_X \in \mathcal{S}_X. \tag{5.88}$$

Using the monotonicity properties, the PB3 algorithm presented in [75] can be modified to also account for structural constraint by having it call Algorithm 3 whenever a node is being evaluated. Thus, the best subset of measurements can be found that contains the structural constraints without having to evaluate every possible n-element measurement combination. However, due to the ADMM algorithm being more computationally expensive to evaluate then directly evaluating the loss when there are no structural constraints, the downward and upward pruning schemes are modified with the aim to reduce the number of calls required to the ADMM algorithm.

Downward pruning

The monotonicity property implies that if one index set X_n is contained in another index set X_s ($X_n \subseteq X_s$), then the optimal loss of the superset must be less or equal to the optimal loss of the subset [65]:

$$L(X_s) \leq L(X_n). \tag{5.89}$$

Similarly, if one index set has some structural constraints imposed, then the loss has to be equal or larger compared to when there are no structural constraints:

$$L(X_n) \leq L(\hat{X}_n). \tag{5.90}$$

Thus, to present the modified downward pruning scheme, let $\underline{L}_n(X_s)$, $s > n$, be a downwards lower bound on L over all n-element subsets of X_s:

$$\underline{L}_n(X_s) \leq \min_{X_n \subseteq X_s} L(X_n). \tag{5.91}$$

Further, let B be a known upper bound to the minimum loss for the case when n variables are selected with structural constraints, such that

$$B \geq L(\hat{X}_n^*), \tag{5.92}$$

then this upper bound can be used to remove (prune) sets of measurements, i.e.,

$$L(\hat{X}_n) > L(\hat{X}_n^*), \quad \forall X_n \subseteq X_s \quad \text{if } \underline{L}_n(X_s) > B. \tag{5.93}$$

The expression in (5.93) indicates that none of the subsets of X_s can be the optimal subset. Therefore, X_s and all its subsets can be discarded without further evaluation, because none of them will be able to produce a smaller loss. The pruning condition in (5.93), will be used as a downward pruning condition whenever a node that is under consideration has an index set X_s that is larger than the terminal set X_n. Thus, for index sets X_s with $s > n$, it is not necessary to account for the structural constraints since the monotonicity property implies that if $L(X_s) > B$ with $X_n \subseteq X_s$, then

$$L(\hat{X}_n) \geq L(X_n) > B, \tag{5.94}$$

and $\hat{X}_n$ can't be optimal.

Upward pruning

The upward BAB takes advantage of the upwards monotonicity property that implies if an index set X_n is a superset of another index set X_s ($X_n \supseteq X_s$) then the optimal loss of the subset must be less or equal to the optimal loss of the superset:

$$L(X_s) \geq L(X_n). \tag{5.95}$$

Thus, to present the principle of upward BAB, let B be an upper bound of as defined in (5.92) and $\underline{L}_n(X_s)$, $s < n$, be an upwards lower bound of L over all n-element supersets of X_s:

$$\underline{L}_n(X_s) \leq \min_{X_n \supseteq X_s} L(X_n). \tag{5.96}$$

Then

$$L(\hat{X}_n) > L(\hat{X}_n^*), \quad \forall X_n \supseteq X_s \quad \text{if } \underline{L}_n(X_s) > B. \tag{5.97}$$

Condition (5.97) ensures that none of the supersets of X_s can be globally optimal. Thus, X_s and its supersets can be pruned without further consideration. Similar, to downward pruning, the structural constraints will be ignored unless the node under consideration an index set X_s with an equal size to the desired terminal set ($s = n$). As a consequence, the lower bounds can be conservative due to $\underline{L}_n(X_s) \leq \underline{L}_n(\hat{X}_s)$, which means that some nodes won't be pruned as quickly as they would if the structural constraints were accounted for. However, since the PB3 algorithm will spend most of its time evaluating nodes that won't be able to provide the optimal solution, the more conservative pruning criteria will be used, since it reduces the number of calls required to the ADMM algorithm.

Upper bound

As already mentioned, the structural constraints will only be considered when evaluating the terminal sets X_n, and only after the following condition is satisfied:

$$L(X_n) < B. \tag{5.98}$$

Assuming (5.98) holds, then Algorithm 3 is used to compute $\hat{H}_X$ with the desired structural constraints for the index set X_n. The resulting loss for the index set $\hat{X}_n$ is used to update the upper bound B according to

$$B = \begin{cases} L(\hat{X}_n), & \text{if } L(\hat{X}_n) < B \\ B, & \text{otherwise.} \end{cases} \tag{5.99}$$

As a result, the PB3 scheme can be used together with the ADMM algorithm to find the optimal subset for measurement combinations with structural constraints. Thus, it is not necessary to screen all possible solutions through exhaustive search with the ADMM algorithm.

5.3 Case studies: Convergence of the ADMM algorithm

Here, the proposed ADMM algorithm has been evaluated on two different case studies. The main purpose is to see how Algorithm 3 convergences and how well it is able to find a good local minimum for the non-convex optimization problem in (5.1)–(5.2).

5.3.1 Random process

In the first case study, a random process has been generated. The process consists of 3 inputs, 8 measurement and 4 disturbances with:

$$G^y = \begin{bmatrix} -0.1031 & 0.2609 & -0.8069 \\ -1.0071 & 0.7786 & 0.8094 \\ -1.5647 & -0.1936 & 0.4398 \\ 1.8334 & 0.9947 & -0.6062 \\ 0.0757 & -1.2160 & 1.0620 \\ 1.9456 & -0.8198 & 0.4871 \\ 0.1119 & 0.5672 & 0.2836 \\ 1.6124 & -0.3451 & 0.6946 \end{bmatrix}, G^y_d = \begin{bmatrix} 0.7930 & -0.5819 & 0.7189 & 0.1261 \\ 0.7931 & 0.0451 & 0.2182 & -0.1238 \\ 0.5492 & 0.4076 & 0.0258 & -0.0536 \\ -0.9135 & 0.0030 & -0.0048 & 0.3923 \\ 1.2711 & -0.2075 & 0.0509 & 0.0017 \\ -1.0865 & -1.1671 & 0.4221 & -0.8474 \\ 1.1219 & 0.2733 & -0.7735 & 0.3302 \\ -0.5605 & 0.3294 & 1.4395 & -1.9455 \end{bmatrix},$$

$$J_{uu} = \begin{bmatrix} 29.4885 & -4.5106 & -5.9584 \\ -4.5106 & 17.8899 & -0.8324 \\ -5.9584 & -0.8324 & 31.0935 \end{bmatrix}, \ J_{ud} = \begin{bmatrix} 0.0550 & 0.3736 & -1.2710 & 0.0749 \\ -1.6069 & 1.3093 & -2.2350 & 0.1956 \\ 1.0128 & 1.3157 & -1.9978 & -0.3612 \end{bmatrix},$$

$$W_n = \operatorname{diag}\left(\begin{bmatrix} 0.4940 & 0.5855 & 0.6490 & 0.4489 & 0.1003 & 0.4080 & 0.0983 & 0.7851 \end{bmatrix}\right),$$

$$W_d = \operatorname{diag}\left(\begin{bmatrix} 0.5351 & 0.6475 & 0.5311 & 0.2984 \end{bmatrix}\right).$$

Algorithm 3 is initialized with $\rho^k = 1$, but instead of having $Q^k = I$, 1000 different initial values of Q^k were randomly chosen to help evaluate how well the proposed algorithm is performing. The first three initial values of Q^k were given by,

$$Q_1 = \begin{bmatrix} -1.58 & 0.94 & 0.35 \\ -0.18 & -1.44 & 0.85 \\ 0.11 & 0.14 & -0.50 \end{bmatrix}, Q_2 = \begin{bmatrix} -1.51 & -0.77 & -1.11 \\ -0.55 & -1.35 & -1.01 \\ 1.72 & -0.03 & -1.21 \end{bmatrix}, Q_3 = \begin{bmatrix} 0.01 & 0.59 & 0.75 \\ -0.13 & 0.56 & 0.13 \\ -0.70 & -1.19 & -1.04 \end{bmatrix} \tag{5.100}$$

and will be used to help illustrate some typical behavior that can be expected for the convergence of Algorithm 3. Two different sets of structural constraints are imposed on the measurement combination H; a block diagonal constraint, where only one element is allowed in each column of H; and a mixed structure, which allows for some elements to exist in the same column of H. The resulting loss is compared to the approximation method 2 [148], and Global search [136] from the Matlab Global Optimization Toolbox. Global search searches for the globally optimal solution by exploring a set of local minima that are solved using the local solver fmincon. The final solution from Global search often tend to give very good results, but the algorithm can be computationally demanding for large scale problems.

Block diagonal constraints

A set of block diagonal constraints are imposed the measurement combination H such that it results in the following structure:

$$H \in \mathcal{S} = \begin{bmatrix} 0 & h_{1,2} & 0 & 0 & h_{1,5} & 0 & h_{1,7} & 0 \\ 0 & 0 & h_{2,3} & 0 & 0 & h_{2,6} & 0 & 0 \\ h_{3,1} & 0 & 0 & h_{3,4} & 0 & 0 & 0 & h_{3,8} \end{bmatrix}. \tag{5.101}$$

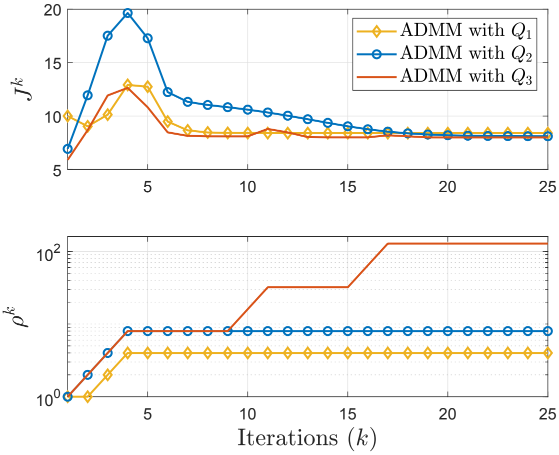


Figure 5.1: Convergence of J^k, and ρ^k given in (5.80), and (5.84), respectively when imposing the block diagonal constraints in (5.101).

The convergence for the function (5.80) together with the evolution of ρ^k can be seen in Figure 5.1 when initializing with the Q matrices given in (5.100). Whenever J^k is increasing, it will cause ρ^k to also increase. Once ρ^k has become large enough, it forces J^k to begin decreasing again until it converges. For all 1000 experiments, Algorithm 3 returned the same (but scaled differently) measurement combination,

$$H = \begin{bmatrix} 0 & 1.0000 & 0 & 0 & -0.5860 & 0 & 1.0374 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 & -2.9920 & 0 & 0 \\ 1.0000 & 0 & 0 & 1.1341 & 0 & 0 & 0 & -0.2867 \end{bmatrix},$$

which resulted in the loss $L = 7.9982$. The exact same results were also provided when using Global search with Matlab global optimization toolbox. Using the approximation method 2 [148] yielded the following measurement combination,

$$H = \begin{bmatrix} 0 & 1.0000 & 0 & 0 & 0.0474 & 0 & 0.0770 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 & -27.6697 & 0 & 0 \\ 1.0000 & 0 & 0 & 8.7718 & 0 & 0 & 0 & 0.9503 \end{bmatrix},$$

for which the loss was 11.4691. Thus, the proposed ADMM algorithm was able to provide a solution that is significantly better than the convex approximation method 2, and that is on par with Global search.

Mixed constraints

Here, a mixed structure is investigated, where the structural constraints imposed on the measurement combination H allows for multiple elements to exist on certain columns of H:

$$H \in \mathcal{S} = \begin{bmatrix} 0 & h_{1,2} & h_{1,3} & 0 & h_{1,5} & 0 & h_{1,7} & h_{1,8} \\ h_{2,1} & 0 & h_{2,3} & 0 & 0 & h_{2,6} & h_{2,7} & 0 \\ h_{3,1} & 0 & 0 & h_{3,4} & 0 & h_{3,6} & 0 & h_{3,8} \end{bmatrix}. \tag{5.102}$$

The smallest loss was obtained using global search, which resulted in the measurement combination,

$$H = \begin{bmatrix} 0 & 1.0000 & 0.3077 & 0 & -0.6240 & 0 & 0.1250 & 0.0044 \\ -17.0222 & 0 & 1.0000 & 0 & 0 & 22.3106 & 31.4103 & 0 \\ 1.0000 & 0 & 0 & 1.9637 & 0 & -0.5584 & 0 & -0.2668 \end{bmatrix},$$

with the loss $L = 4.6620$. The proposed ADMM algorithm was able to achieve the exact same result as Global search for 907 out the 1000 experiments. The average loss for the 1000 simulations was $\bar{L} = 4.8426$ with the largest obtained loss being $\hat{L} = 13.44396$. In contrast, the resulting measurement combination when using the approximation method was

$$H = \begin{bmatrix} 0 & 1.0000 & 0.0580 & 0 & -0.0491 & 0 & 0.4855 & 0.3208 \\ 0.5676 & 0 & 1.0000 & 0 & 0 & -1.3793 & -2.1468 & 0 \\ 1.0000 & 0 & 0 & 9.6080 & 0 & 0.6697 & 0 & 0.9914 \end{bmatrix},$$

which gave the loss $L = 11.2298$. Thus, while in most cases, the proposed ADMM algorithm outperforms the approximation method in terms of finding an optimal solution. There is always the possibility that in certain circumstances, the loss given by Algorithm 3 is higher than the one obtained by the approximation method, due to both methods only being able to guarantee a local solution.

The convergence for the function (5.80) together with the evolution of ρ^k can be seen in Figure 5.2 when the proposed algorithm has been initialized with the different Q matrices given in (5.100). For all the three initial values on Q^k, the ADMM algorithm is almost able to converge after 40 iterations. Due to the relatively quick convergence and computations of H^{k+1}, $\hat{H}^{k+1}$, and Q^{k+1}, it should be possible to alter Algorithm 3 such that it performs several runs using different initial values of Q^k. This would increase the likelihood of getting a result that is close to the globally optimal solution.

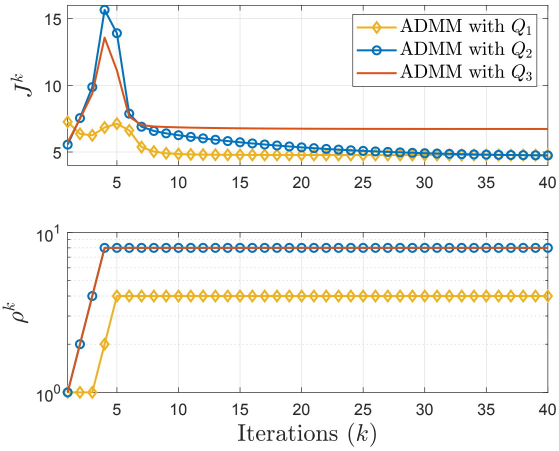


Figure 5.2: Convergence of J^k, and ρ^k given in (5.80), and (5.84), respectively when imposing the mixed constraints in (5.102).

5.3.2 Evaporator

The next case study consists of the evaporation process represented in Figure 5.3. The evaporator was originally treated by [103] and has been modified in [76]. The process uses 2 inputs u, 10 potential measurements y, with the 3 disturbances d being changes in compositions and temperature of the inflows:

$$u = \begin{bmatrix} F_{200} & F_1 \end{bmatrix}^T \tag{5.103}$$

$$y = \begin{bmatrix} p_2 & T_2 & T_3 & F_2 & F_{100} & T_{201} & F_3 & F_5 & F_{200} & F_1 \end{bmatrix}^T \tag{5.104}$$

$$d = \begin{bmatrix} x_1 & T_1 & T_{200} \end{bmatrix}^T \tag{5.105}$$

The economic objective of the evaporator is to maximize the operating profit [\$/h] and has been formulated in [76]:

$$J = 600F_{100} + 0.6F_{200} + 1.009(F_2 + F_3) + 0.2F_1 - 4800F_2. \tag{5.106}$$

The CV selection containing structural constraints has been studied in [59], and [146]. Four different control structures ($\mathcal{S}_1 - \mathcal{S}_4$) were obtained in [146] that had been designed to separate the measurements associated with the condenser and the measurements associated with the evaporator. The control structure uses subsets of 3, 4, 5, and 10 measurements from (5.104) and are given by

$$\mathcal{S}_1 = \begin{bmatrix} 0 & h_{1,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{2,8} & h_{2,9} & 0 \end{bmatrix}, \tag{5.107}$$

$$\mathcal{S}_2 = \begin{bmatrix} 0 & 0 & 0 & h_{1,4} & h_{1,5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{2,8} & h_{2,9} & 0 \end{bmatrix}, \tag{5.108}$$

$$\mathcal{S}_3 = \begin{bmatrix} 0 & 0 & 0 & h_{1,4} & 0 & 0 & h_{1,7} & 0 & 0 & 0 \\ h_{2,1} & 0 & 0 & 0 & 0 & 0 & 0 & h_{2,8} & h_{2,9} & 0 \end{bmatrix}, \tag{5.109}$$

and,

$$\mathcal{S}_4 = \begin{bmatrix} 0 & h_{1,2} & 0 & h_{1,4} & h_{1,5} & 0 & h_{1,7} & 0 & 0 & h_{1,10} \\ h_{2,1} & 0 & h_{2,3} & 0 & 0 & h_{2,6} & 0 & h_{2,8} & h_{2,9} & 0 \end{bmatrix}. \tag{5.110}$$

Using the proposed ADMM algorithm, CVs are found for the above structural constraints. These results are compared to using Global search [136] and approximation method 2 [148]. The resulting measurement combination H when imposing the structural constraints in $\mathcal{S}_4$, were identical for both the ADMM algorithm and global search, which resulted in

$$H = \begin{bmatrix} 0 & 1 & 0 & 136.8399 & -9.8220 & 0 & -2.2268 & 0 & 0 & -5.2358 \\ 1 & 0 & 0.8319 & 0 & 0 & -1.4251 & 0 & 34.1130 & -1.9143 & 0 \end{bmatrix},$$

whereas approximation method 2 yielded

$$H = \begin{bmatrix} 0 & 1 & 0 & 151.2819 & -10.2887 & 0 & -2.6060 & 0 & 0 & -5.0012 \\ 1 & 0 & 0.8295 & 0 & 0 & -2.0407 & 0 & 41.6321 & -2.1988 & 0 \end{bmatrix}.$$

The resulting loss, together with the other obtained control structures, are available in Table 5.1. In addition, the CPU time needed to compute the solutions has also been included in Table 5.1.

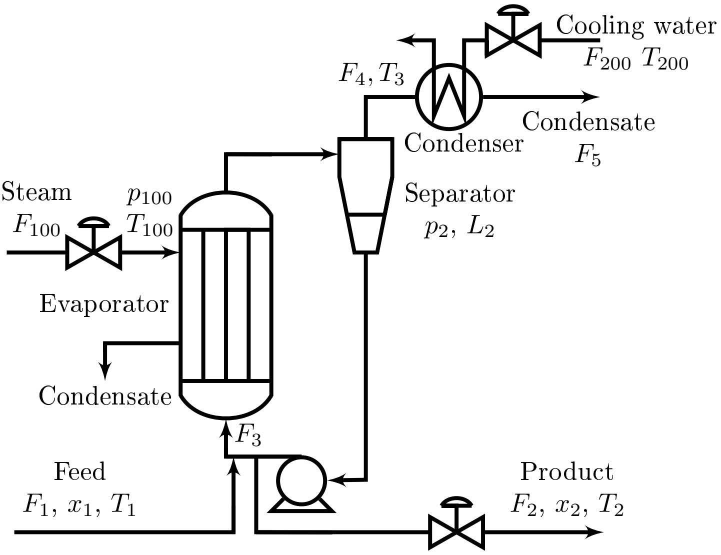


Figure 5.3: The evaporator process flowsheet, adopted from [76].

The results from Table 5.1 shows that for the structural constraints in (5.107)-−(5.110) imposed on the evaporator, both the ADMM algorithm and the Global search are able to find measurement combinations with a smaller loss compared to the approximation method. In particular, for the constraint $\mathcal{S}_3$ in (5.109), the improvement in the loss is of great significance. The final result for the global search and the proposed ADMM algorithm are almost identical; thus, both methods seem able to find good solutions. However, the CPU time required by the ADMM algorithm to compute the measurement combinations is $100 - 1000$ times smaller compared to Global search. Therefore, the ADMM algorithm is a lot more scalable. This is also evident from Table 5.1, where the CPU time rapidly grows when searching for solutions that contain a larger set of measurements. On the other hand, the CPU time for the ADMM algorithm is almost unaffected by the increase in the number of measurements.

Table 5.1: CVs for the Evaporator together with their respective loss and CPU time when using Global search (G. search) [136], approximation method 2 (Method 2) [148], and the proposed ADMM algorithm (ADMM).

		Controlled variables (CVs)	Loss	CPU(s)
S_1 :	G. search :	$c = \left[T_2; \quad F_5 - 0.0800F_{200}\right]$	58.2074	3.4788
	Method 2 :	$c = \left[T_2; \quad F_5 - 0.0386F_{200}\right]$	58.6554	0.0468
	ADMM :	$c = \left[T_2; \quad F_5 - 0.0802F_{200}\right]$	58.2074	0.0312
S_2 :	G. search :	$c = \left[-6.2372F_2 + F_{100}; \quad F_5 - 0.0391F_{200}\right]$	11.9160	3.8532
	Method 2 :	$c = \left[-6.2592F_2 + F_{100}; \quad F_5 - 0.0386F_{200}\right]$	11.9439	0.0156
	ADMM :	$c = \left[-6.2374F_2 + F_{100}; \quad F_5 - 0.0391F_{200}\right]$	11.9160	0.0624
S_3 :	G. search :	$c = \left[P_2 - 6.7287F_5 - 0.0998F_{200}; \quad F_2 - 0.0321F_3\right]$	20.3317	14.4457
	Method 2 :	$c = \left[P_2 + 117.7954F_5 - 4.9451F_{200}; \quad F_2 - 0.0324F_3\right]$	31.7479	0.0468
	ADMM :	$c = \left[P_2 - 6.7287F_5 - 0.0998F_{200}; \quad F_2 - 0.0321F_3\right]$	20.3317	0.0936
S_4 :	G. search :	$c = \left[f(T_2, F_2, F_{100}, F_3.F_1); \quad f(P_2, T_3.T_{201}, F_5, F_{200})\right]$	8.8847	42.3075
	Method 2 :	$c = \left[f(T_2, F_2, F_{100}, F_3.F_1); \quad f(P_2, T_3.T_{201}, F_5, F_{200})\right]$	9.2450	0.1404
	ADMM :	$c = \left[f(T_2, F_2, F_{100}, F_3.F_1); \quad f(P_2, T_3.T_{201}, F_5, F_{200})\right]$	8.8847	0.0624

5.4 Case studies: Subset selection with structural constraints

The main purpose of these case studies is to evaluate how well the proposed ADMM algorithm combined with branch and bound (PB3) [75] performs when selecting measurement subsets with structural constraints imposed on the measurement combination H. The results are compared to the approximation based MIQP methods [147], [148], which were briefly described in Section 5.1.

5.4.1 Evaporator

Here, the evaporator that was described in Section 5.3.2 will again be used as a case study. The process consisted of 2 inputs u, 10 potential measurements y, and 3 disturbances d that were defined in (5.103)−(5.105) with the objective function J being given in (5.106).

The subset selection with structural constraints imposed on the measurement combination has previously been studied in [59], and [146]. Heldt used a generalized singular value decomposition (GSVD) approach in [59] whereas Yelhuru implemented the two approximation methods [147], [148] in his PhD thesis [146] on the evaporator process. The approximation methods were able to get slightly better results compared to the GSVD approach. Out of the two approximation methods, there was no difference in their final result, at least in terms of minimizing the loss [146]. Therefore, the second approximation method is used as a benchmark to compare the performance with the proposed ADMM algorithm combined with the PB3 scheme.

The CVs will be designed to separate the measurements associated with the condenser (P_2; T_3; T_{201}; F_5; F_{200}) and the measurements associated with the evaporator (T_2; F_2; F_{100}; F_3; F_1), such that the resulting structural constraints become:

$$\mathcal{S} = \begin{bmatrix} 0 & h_{1,2} & 0 & h_{1,4} & h_{1,5} & 0 & h_{1,7} & 0 & 0 & h_{1,10} \\ h_{2,1} & 0 & h_{2,3} & 0 & 0 & h_{2,6} & 0 & h_{2,8} & h_{2,9} & 0 \end{bmatrix}. \tag{5.111}$$

Table 5.2: CVs for the Evaporator and their respective loss when using the convex approximation method 2 [148], and the proposed ADMM algorithm with PB3.

No. of meas.	Controlled variables (CVs)	Loss $\frac{1}{2}\|HY\|_F^2$
2	Method 2 : $c = [T_2; \quad T_{201}]$	65.3235
	ADMM : $c = [F_3; \quad F_{200}]$	56.0260
3	Method 2 : $c = [T_2; \quad F_5 - 0.0386F_{200}]$	58.6554
	ADMM : $c = [F_2 - 0.1639F_{100}; \quad F_{200}]$	20.4124
4	Method 2 : $c = [-6.2592F_2 + F_{100}; \quad F_5 - 0.0386F_{200}]$	11.9439
	ADMM : $c = [-6.2376F_2 + F_{100}; \quad F_5 - 0.0391F_{200}]$	11.9160
5	Method 2 : $c = [-6.2592F_2 + F_{100}; \quad 0.0085p_1 + F_5 - 0.0420F_{200}]$	11.9328
	ADMM : $c = [-8.6799F_2 + F_{100} + 0.4043F_1; \quad F_5 - 0.0388F_{200}]$	10.8068
10	Method 2 : $c = [f(T_2, F_2, F_{100}, F_3.F_1); \quad f(P_2, T_3.T_{201}, F_5, F_{200})]$	9.2450
	ADMM : $c = [f(T_2, F_2, F_{100}, F_3.F_1); \quad f(P_2, T_3.T_{201}, F_5, F_{200})]$	8.8847

For the structural constraints in (5.111), subsets of measurement combinations are computing using the ADMM algorithm. These results are compared to using approximation method 2 [148], which involved solving an MIQP using the IBM ILOG Optimizer CPLEX solver [63] in Matlab, with $m = 200$ selected as the value for the big-M constraint. The minimized loss as a function of the number of measurement used can be seen in Figure 5.4 when using these two approaches, with their CVs shown in Table 5.2 that corresponds to subsets of 2−5 measurements. The proposed ADMM algorithm is able to find CVs that provide a significant reduction in loss compared to the approximation method. Especially when choosing a subset of 3 measurements, the improvement in loss is almost threefold.

The CPU times required to compute the solutions are plotted in Figure 5.5. When using ADMM with the exhaustive search, i.e., the ADMM algorithm is used

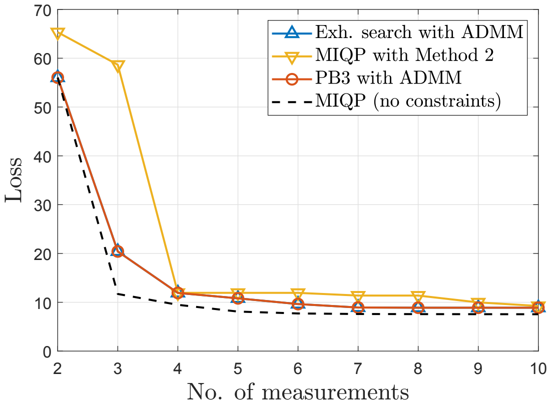


Figure 5.4: The loss vs. the of number measurements when imposing structural constraints on the evaporator case study. The results are obtained using (i) exhaustive search with Algorithm 3, (ii) the convex approximation method 2 in [148], (iii) Algorithm 3 combined with PB3 [75], and (iv) MIQP to find the optimal subsets without constraints [149].

on all possible measurement subsets, then the time it takes to find an optimal solution is almost 10 times higher compared to when combining the ADMM algorithm with PB3. The two cases where MIQP is used requires less time than then PB3 with ADMM. However, since the size of the evaporator case study is relatively small and both the approximation method and the proposed ADMM algorithm with PB3 are able to obtain their solution almost under 1 second, then the small differences in the CPU times become less important.

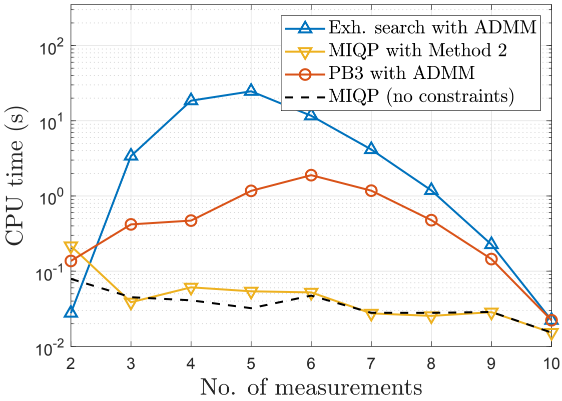


Figure 5.5: The CPU time vs. the number of measurements when imposing structural constraints on the evaporator case study. The results are obtained using (i) exhaustive search with Algorithm 3, (ii) approximation method 2 in [148], (iii) Algorithm 3 combined with PB3 [75], and (iv) MIQP to find the optimal subsets without constraints [149].

5.4.2 Distillation column

Next, the ADMM algorithm with the PB3 method was evaluated on the binary distillation column model [121]. The column is described in more detail in Section 4.4, where the objective is to minimize the deviations in the top product composition (x_D) and the bottom product composition (x_B) by using the reflux flow rate (L_R) and the vapor boilup rate (V_B) as the inputs. There are 41 potential temperature measurements $T_i(^oC)$ avaialble, where i denotes the different stages inside the column. Therefore, the aim is to obtain the optimal subset of the available measurements when structural constraints have been imposed on the measurement selection matrix H.

Table 5.3: CVs for the distillation column and their respective loss when using approximation method 1 [147], the BARON solver from [78], and the ADMM algorithm with PB3.

	No. of meas.	Controlled variables (CVs)	Loss $\frac{1}{2}\|HY\|_F^2$
Block diagonal structure		Method 1 : $c = [T_{29};\quad T_{12}]$	0.5530
	2	BARON : $c = [T_{30};\quad T_{12}]$	0.5478
		ADMM : $c = [T_{30};\quad T_{12}]$	0.5478
		Method 1 : $c = [T_{30} + 0.989T_{31};\quad T_{12}]$	0.4427
	3	BARON : $c = [T_{30} + 0.997T_{31};\quad T_{12}]$	0.4427
		ADMM : $c = [T_{30} + 0.997T_{31};\quad T_{12}]$	0.4427
		Method 1 : $c = [T_{30} + 0.989T_{31};\quad T_{11} + 0.992T_{12}]$	0.3441
	4	BARON : $c = [T_{30} + 0.996T_{31};\quad T_{11} + 0.998T_{12}]$	0.3441
		ADMM : $c = [T_{30} + 0.996T_{31};\quad T_{11} + 0.998T_{12}]$	0.3441
		Method 1 : $c = [f(T_{21} \cdots T_{41});\quad f(T_1 \cdots T_{20})]$	0.1046
	41	BARON : $c = [f(T_{21} \cdots T_{41});\quad f(T_1 \cdots T_{20})]$	0.1039
		ADMM : $c = [f(T_{21} \cdots T_{41});\quad f(T_1 \cdots T_{20})]$	0.1039
Triangular structure		Method 1 : $c = [T_{30};\quad T_{12}]$	0.5478
	2	BARON : $c = [T_{30};\quad T_{12}]$	0.5478
		ADMM : $c = [T_{30};\quad T_{12}]$	0.5478
		Method 1 : $c = [T_{30} + 0.989T_{31};\quad T_{11} + 0.737T_{30} + 0.781T_{31}]$	0.4640
	3	BARON : $c = [T_{30} + 1.022T_{31};\quad T_{12} + 0.974T_{30} + 1.040T_{31}]$	0.4425
		ADMM : $c = [T_{30} + 1.022T_{31};\quad T_{12} - 0.164T_{30} - 0.123T_{31}]$	0.4425
		Method 1 : $c = [T_{30} + 0.989T_{31};\quad T_{10} + 1.006T_{11} + 0.690T_{30} + 0.774T_{31}]$	0.3529
	4	BARON : $c = [T_{29} + 1.137T_{31};\quad T_{12} + 0.916T_{13} + 0.571T_{29} + 0.934T_{31}]$	0.3565
		ADMM : $c = [T_{30} + 1.023T_{31};\quad T_{11} + 0.998T_{12} - 0.306T_{30} - 0.215T_{31}]$	0.3439
		Method 1 : $c = [f(T_{21} \cdots T_{41});\quad f(T_1 \cdots T_{41})]$	0.0937
	41	BARON : $c = [f(T_{21} \cdots T_{41});\quad f(T_1 \cdots T_{41})]$	0.0881
		ADMM : $c = [f(T_{21} \cdots T_{41});\quad f(T_1 \cdots T_{41})]$	0.0881

Similar to the evaporator case study, selecting measurement subsets with structural constraints have been studied in [146] when using the MIQP with the two approximation methods in [147], and [148]. The first approximation method was shown in [146] to outperform the second one for this particular example, and thus, it will be used for comparison to the proposed ADMM algorithm combined with the PB3 method. In addition, the results will also be compared to the general purpose solver BARON [78], which is considered one of the leading commercial global solvers for solving mixed-integer nonlinear programming (MINLP) optimization problems [84], [102].

In [146], the approximations methods were used to find measurement combination with block diagonal and triangular structures. In particular, the block diagonal structure is often desirable in many practical situations due to dynamic reasons. Similarly, a triangular structure can also be dynamically beneficial, since it can reduce the time delays by, e.g., including the bottom tray temperatures for the reflux flow rate L_R. Therefore, these two structural constraints will also be considered here, with the block diagonal constraints being

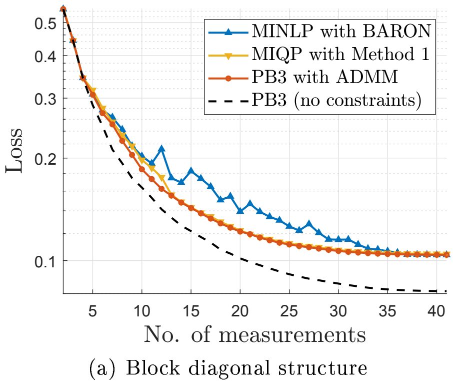
$$\mathcal{S}_{Blk} = \begin{bmatrix} 0 & \cdots & 0 & h_{1,21} & \cdots & h_{1,41} \\ h_{2,1} & \cdots & h_{2,20} & 0 & \cdots & 0 \end{bmatrix}, \tag{5.112}$$

and the triangular constraints given by

$$\mathcal{S}_{Tri} = \begin{bmatrix} 0 & \cdots & 0 & h_{1,21} & \cdots & h_{1,41} \\ h_{2,1} & \cdots & h_{2,20} & h_{2,21} & \cdots & h_{2,41} \end{bmatrix}. \tag{5.113}$$

The resulting CVs for 2, 3, 4, and 41 measurements together with their respective loss, are shown in Table 5.3 for both block diagonal and triangular structural constraints. The obtained loss for 3, and 4 measurements when using the approximation method are higher for the triangular structure compared to the block diagonal structure. This was also mentioned in [146] since it contradicts the monotonicity property, in which adding more elements should not increase the loss. Therefore, the optimal solution for the triangular structure needs to be at least as good as the block diagonal structure when the number of measurements is the same. A similar behavior can also be seen in Figure 5.6 where the loss has been plotted against the number of measurements, for both the block diagonal and the triangular constraints. For the triangular structure in Figure 5.6(b), the loss is occasionally increasing when using the approximation method despite there being an increase in the number of measurements. However, this is not the case when using the proposed ADMM algorithm with PB3, which is able to provide solutions that are monotonically decreasing as more measurements are added with a loss that is less or equal to the convex approximation method.

The computational times required to solve the problems are shown in Figure 5.7, where the maximum time limit had been set to 3600 seconds. This is why the ADMM algorithm is able to provide a better solution than BARON, despite BARON being a global solver. However, even for the cases where the maximum time limit has been reached, the ADMM algorithm with PB3 was able to give good solutions. This is due to the branching strategy used by the PB3 method puts a higher priority on evaluating solutions that are estimated to give good results. The proposed ADMM algorithm with PB3 is able to find solutions in a time frame that is comparable to the approximation method. It works especially well when searching for a subset of measurements that is smaller than 27 for the triangular structure. One reason for this may be due to the loss for the triangular structure is smaller and a lot closer to the case when there are no constraints. The pruning

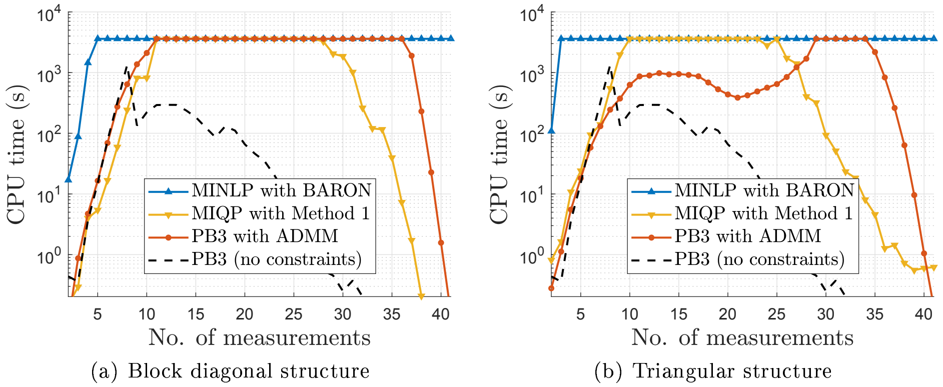


(a) Block diagonal structure

(b) Triangular structure

Figure 5.6: The loss vs. the number of measurements when imposing block diagonal and triangular structural constraints on the distillation column. The results are obtained using (i) the BARON solver from [78], (ii) approximation method 1 in [147], (iii) Algorithm 3 combined with PB3 [75], and (vi) PB3 to find the optimal subsets without constraints [75].

conditions used when combining the ADMM algorithm with the PB3 method are a bit conservative since the structural constraints are only being considered when evaluating the terminal nodes. This would imply that the closer the loss is between the optimal constrained and unconstrained measurement combinations, the more effective will the pruning of suboptimal nodes be.



(a) Block diagonal structure

(b) Triangular structure

Figure 5.7: The CPU time vs. the number of measurements when imposing block diagonal and triangular structural constraints on the distillation column. The results are obtained using (i) the BARON solver from [78], (ii) approximation method 1 in [147], (iii) Algorithm 3 combined with PB3 [75], and (vi) PB3 to find the optimal subsets without constraints [75].

5.5 Conclusion

In this chapter, an ADMM algorithm for incorporating structural constraints in self-optimizing control variables has been investigated. It was demonstrated in two different case studies that the ADMM algorithm contained very good convergence properties, both in terms of obtaining good solutions and the CPU time required to find the solutions.

To find the optimal measurement subsets, the proposed ADMM algorithm was combined with the partial bidirectionally branch and bound (PB3) method presented in [26]. The resulting scheme was able to find measurement subsets with a smaller loss compared to the methods in [147], and [148] when implemented on case studies, consisting of an evaporator and a distillation column. It may thus, serve as an alternative to the approaches in [59], [147] and [148] when trying to compute CVs with specified structures.

Furthermore, the proposed algorithm seemed capable of computing solutions that were on par with the commercial MINLP solver BARON [78], but with a significant reduction in the computational time. Thus, while the solution from the proposed algorithm can't be guaranteed to be globally optimal, it can be used to initialize BARON or other global solvers to reduce their CPU time.

Part III

Accounting for the dynamics in self-optimizing control

Chapter 6

Dynamic considerations for linear measurement combinations

Self-optimizing control focuses on finding controlled variables (CVs), that when kept at constant set-points results in near optimal steady-state economic operation in spite of disturbances. Besides using single measurements as the CVs, the null space, or the exact local method can be used to select linear combinations of measurements ($c = Hy$) that can further improve the self-optimizing control performance.

The null space method selects linear measurement combinations H, such that H satisfies $HF = 0$, with F being the sensitivity matrix given in (3.24). When using the exact local method, the matrix H can be obtained from equation (3.36). In both cases, the resulting measurement combination matrix H can be multiplied with any non-singular matrix Q such that $\hat{H} = Q^{-1}H$, without having any effect on the steady-state loss. Therefore, when multiple unconstrained manipulated variables (MVs) are available, both the null space and the exact local offers sets, consisting of an infinite number of possibilities for choosing the measurement combinations. This property also holds for single input systems, when using the null space method due to the non-uniqueness of the solution to $HF = 0$. All solutions give the same loss from a steady-state perspective, but the dynamic response depends on how the measurement combinations are selected. Thus, poorly chosen linear combinations can lead to complex dynamic behavior for which designing feedback controllers could be challenging.

6.1 Dynamic effects when selecting measurement combinations

To better illustrate the dynamic effects when selecting measurement combinations, let's consider an example of the same binary distillation column that was used for the case study in Section 4.4. The objective is to minimize the deviations in the product compositions using a linear combination of the temperature measurements inside the column. Let the available temperature measurements be,

$$y = \begin{bmatrix} T_{11} & T_{12} & T_{14} & T_{30} & T_{31} \end{bmatrix}^T \tag{6.1}$$

with the sensitivity matrix:

$$F = \begin{bmatrix} 3.0644 & 4.0279 & 6.1584 & 3.0958 & 2.3518 \\ 0.2172 & 0.2850 & 0.4384 & 0.2572 & 0.1946 \end{bmatrix}^T . \tag{6.2}$$

Using the null space method, two possible candidates are considered $c_1 = H_1 y$ and $c_2 = H_2 y$, where H_1 and H_2 are given by

$$H_1 = \begin{bmatrix} 0.1569 & -0.0126 & -0.0625 & 0.5869 & -0.7917 \\ 0.6795 & -0.6884 & 0.1185 & -0.1437 & 0.1724 \end{bmatrix}, \tag{6.3}$$

$$H_2 = \begin{bmatrix} 0.1938 & 0.7602 & -0.6154 & -0.0139 & 0.0752 \\ 0.9004 & -0.3855 & -0.1973 & 0.0269 & -0.0318 \end{bmatrix}. \tag{6.4}$$

Both the linear combinations H_1, and H_2 satisfies $HF = 0$, and thus, they are equivalent when only considering the steady-state loss. Therefore, it would be preferable to select the CV that is easier to control.

One way of analyzing the two different control structure is to use the relative gain array (RGA) [22]. Let $G(s)$ represent the transfer function for a continuous process with an equal number of outputs and inputs. The RGA is then defined as

$$RGA(s) := G(s) \times G(s)^{-T}, \tag{6.5}$$

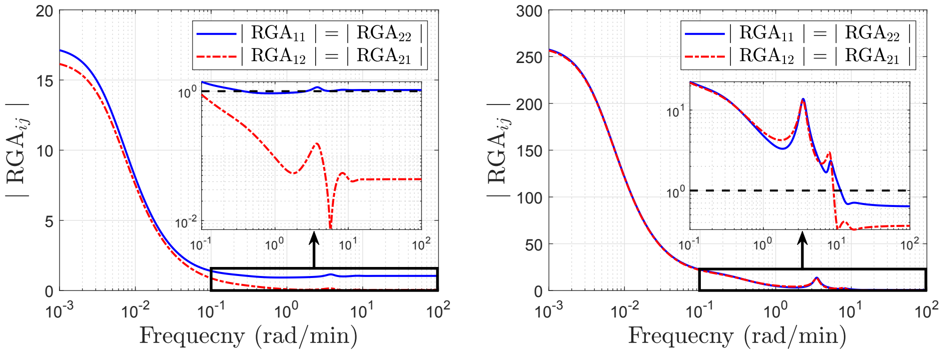
where the symbol × denotes the element-by-element multiplication. For decentralized control, the RGA can be used to determine the pairing of the inputs and outputs using the following rules [125]:

- **Pairing rule 1:** *Prefer pairings such that the rearranged system, with the selected pairings along the diagonal, has an RGA close to identity at the frequencies around the closed-loop bandwidth.*
- **Pairing rule 2:** *Avoid (if possible) pairing on negative steady-state RGA elements.*

The two proposed measurement combinations, given in (6.3) and (6.4) results in the following steady-state RGA matrices:

$$RGA_{c_1}(0) = \begin{bmatrix} 17.44 & -16.44 \\ -16.44 & 17.44 \end{bmatrix}, \quad RGA_{c_2}(0) = \begin{bmatrix} 262.24 & -261.24 \\ -261.24 & 262.24 \end{bmatrix}. \tag{6.6}$$

Both the RGA matrices have negative elements in the off-diagonal, which according to the second pairing rule suggests a diagonal pairing of the inputs and the outputs is preferred. Next, the RGA elements are plotted in Figure 6.1 for the c_1, and c_2 with respect to the frequency. After about 0.1 rad/min, the diagonal elements for $c_1 = H_1 y$ in Figure 6.1(a) start to become close to one, with the off-diagonal elements becoming close to zero. Thus, at frequencies between 0.1 rad/min and 100 rad/min, the RGA matrix for the measurement combination in (6.3) is close to the identity. The RGA elements for the second measurement combination (6.4)



(a) RGA elements for $c_1 = H_1 y$.

(b) RGA elements for $c_2 = H_2 y$.

Figure 6.1: Magnitude of RGA elements for the binary distillation column

is shown in Figure 6.1(b). Here, there are large values for both the diagonal and off-diagonal elements (with the off-diagonal corresponding to negative values) in the RGA matrix for frequencies smaller than 10 rad/min. According to the first pairing rule, it is desirable to have an RGA close to the identity matrix around frequencies at the closed-loop bandwidth. Therefore, based on Figure 6.1, it would suggest that using the measurement combination in (6.3) is preferable to the one in (6.4) as it would likely result in a lot less interaction between the CVs. Especially, if the closed-loop bandwidth is placed between 0.1 rad/min and 100 rad/min since in that frequency region, the RGA in Figure 6.1(a) will closely resemble the identity matrix.

6.2 Measurement combinations effect on poles and zeros

As illustrated in the previous section, the choice of the measurement combinations can significantly alter the dynamic behavior of the resulting system. Therefore, similar to the work in [4], the measurement combinations effect on the poles and zeros for the underlying plant will be discussed here.

Defining a system described by a continuous state-space model as,

$$\dot{x}(t) = A_c x(t) + B_c u(t) \tag{6.7}$$
$$y(t) = C_c x(t) + D_c u(t) \tag{6.8}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, and $y \in \mathbb{R}^{n_y}$, are the states, inputs, and measurements, respectively. When using a measurement combination $\hat{H}$ ($\hat{H} = Q^{-1}H$) is used, then the measurements y in (6.8) can be replaced with the CV:

$$c(t) = \hat{H}C_c x(t) + \hat{H}D_c u(t) \tag{6.9}$$
$$= Q^{-1}HC_c x(t) + Q^{-1}HD_c u(t). \tag{6.10}$$

This results in a square plant which can be represented by the system matrices in (6.7), and $c \in \mathbb{R}^{n_u}$ in (6.10) as the new measurements. The measurement selection matrices $\hat{H} \in \mathbb{R}^{n_u \times n_y}$, and $H \in \mathbb{R}^{n_u \times n_y}$ gives the same steady-state loss as long $Q \in \mathbb{R}^{n_u \times n_u}$ is a non-singular matrix. Instead of the state-space model in (6.7) and (6.10), a transfer function $G(s) \in \mathbb{R}^{n_u \times n_u}$ can also be used to describe the system when using measurement combinations, which can be given by

$$G(s) = Q^{-1}HC_c(sI - A_c)^{-1}B_c + Q^{-1}HD_c. \tag{6.11}$$

The poles determine the behavior and stability of the system and are the values for s at which the denominator of a transfer function $G(s)$ becomes zero. For a state-space model described in (6.7), and (6.10), the poles can be found by computing the eigenvalues of the matrix A_c. [125]

Theorem 6.1. *The location of the poles for the system given by (6.7), and (6.10) are independent of the static non-singular matrix Q, and the linear measurement combination H.*

Proof. The location of the poles is equivalent to the eigenvalues of the system matrix A_c in (6.7). It then becomes evident that matrices H, and Q have no effect on the system poles since they don't influence the matrix A_c. □

The zeros for a transfer function $G(s)$ are the frequencies where the numerator of $G(s)$ becomes zero, i.e., it can be argued that zeros are the values for s in which the transfer function $G(s)$ loses rank. The location of the zeros plays a critical role on the dynamic behavior of the system, where, e.g., zeros placed at the right hand plane (RHP) will pose limitations on the achievable performance for the control system that is being implemented. [125]

Theorem 6.2. *The location of the zeros to the system given by (6.7), and (6.10) is dependent on the measurement combination H, but are unaffected by the choice of Q as long as Q is a static matrix that is of full rank.*

Proof. Defining the matrices $N \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$, and $M(s) \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$:

$$N := \begin{bmatrix} I & 0 \\ 0 & Q^{-1} \end{bmatrix}, \quad M(s) := \begin{bmatrix} sI - A_c & -B_c \\ HC_c & HD_c \end{bmatrix}. \tag{6.12}$$

The state-space model in (6.7) and (6.10) can be written as

$$\begin{bmatrix} 0 \\ c \end{bmatrix} = \begin{bmatrix} sI - A_c & -B_c \\ Q^{-1}HC_c & Q^{-1}HD_c \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}, \tag{6.13}$$

which is equivalent to

$$\begin{bmatrix} 0 \\ c \end{bmatrix} = NM(s) \begin{bmatrix} x \\ u \end{bmatrix}, \tag{6.14}$$

where the zeros are the values for s in which the polynomial system matrix $NM(s)$ loses rank [125]. Both N and $M(s)$ are square matrices, with N always being a full rank matrix as long as Q is non-singular. The rank for the multiplication $NM(s)$ is then given by [51]:

$$\operatorname{rank}\big(NM(s)\big) = \operatorname{rank}\big(M(s)\big). \tag{6.15}$$

Therefore, the rank and hence, the location of the zeros are determined by the matrix $M(s)$, which is dependent on the measurement combination H. □

When the system in (6.7) and (6.10) have multiple inputs u, then the resulting system poles and zeros will have directions associated to them. The pole directions can be obtained directly from the transfer function matrix $G(s)$ at the different poles p_i:

$$G(p_i)u_{p,i} = \infty, \quad y_{p,i}^{\mathcal{H}}G(p_i) = \infty. \tag{6.16}$$

Here, $u_{p,i}$, and $y_{p,i}^{\mathcal{H}}$ are defined as the i:th input pole direction, and i:th the output pole direction respectively with $\mathcal{H}$ being defined as the complex conjugate transpose. The pole directions provide an indication on how much the i:th mode is excited for each input and output [125].

Similarly, the i:th input zero direction, and the i:th output zero direction, defined as $u_{z,i}$, and $y_{z,i}^{\mathcal{H}}s$ respectively can be obtained from:

$$G(z_i)u_{z,i} = \infty, \quad y_{z,i}^{\mathcal{H}}G(z_i) = \infty. \tag{6.17}$$

The output zero directions $y_{z,i}^{\mathcal{H}}$, are typically of greater interest than the input zero directions $u_{z,i}$ since they give information about which output may be difficult to control. Thus, the achievable control performance will be limited by the CV that corresponds to non-zero elements in the zero output direction. [125]

Theorem 6.3. *The pole and zero directions for the system given by (6.7), and (6.10) are dependent on both the non-singular matrix Q and on the linear measurement combination H.*

Proof. The pole and zero directions are given in (6.16), and (6.17), respectively, where they both are influenced by the transfer function $G(s)$. Since $G(s)$, given in (6.11) is dependent on both Q and H, it can be concluded that they both will influence the pole and zero directions for the system. □

To summarize, neither the measurement combination H nor the non-singular matrix Q will change the location of the system poles. However, the choice of H will influence the location of the zeros; and both Q and H will affect the pole and zero directions for the system.

6.3 Controller design for measurement combinations

In Section 6.1, and 6.2, it was demonstrated that the selection of measurement combinations will influence the dynamic behavior of the system. As a consequence, they can have a significant impact on the closed-loop performance, where the resulting control configuration is illustrated in Figure 6.2. For systems with multiple unconstrained degrees of freedom, the non-singular matrix Q can be chosen with great flexibility as it doesn't influence the steady-state performance for which most of the economics are based on. Therefore, the matrix Q was treated in [4], as a static compensator, which can be used to try to decouple the system interactions at some desired frequency range. A typical stepwise approach for designing the closed-loop system shown in Figure 6.2 can then be given by:

Step 1 Obtain the optimal measurement combination H that minimizes the steady-state loss.

Step 2 Find a non-singular matrix Q such that $Q^{-1}H$ improves the controllability of the system by, e.g., reducing the interactions between the resulting CVs.

Step 3 Design the feedback controllers K.

The first step can be solved using, e.g., the null space method in (3.27) or for the exact local method in (3.36) to obtain a measurement combination H. If the process is mainly perturbed by slow varying disturbances, then for the second step, it may be sufficient to use steady-state decoupling by setting $Q = G(0)$. A decentralized controller can then be designed such that the closed-loop bandwidth lies in the lower frequency domain. In case it is necessary to have a controller with a faster response time, then Q should be chosen such that the interactions between the CVs are reduced by, e.g., having the corresponding RGA be close to the identity matrix around the desired bandwidth frequency. Alternatively, instead of the RGA, some gramian based input-output pairing techniques could be used, such as the participation matrix [30], the Hankel interaction index array [143], or the Σ_2 interaction measure [19]. Finally, the feedback controller K needs to be tuned for the CVs, after which dynamic simulations would typically be necessary to validate the proposed control structure. If the result is not acceptable, the above three-step procedure has to be repeated, which may require finding a different measurement combination. As a consequence, selecting measurement combinations and designing feedback controller can in certain cases, become a very tedious task since it may require a lot of "trial and error".

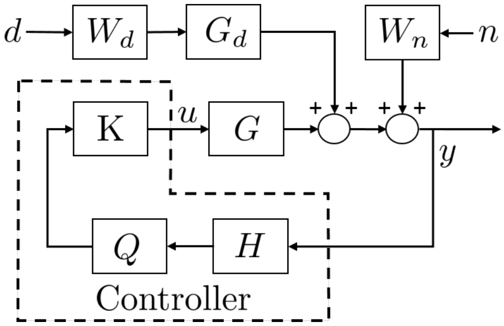


Figure 6.2: Closed-loop system that controls a measurement combination.

The null space method and exact local method are steady-state only when selecting measurement combinations H. Therefore, they don't account for the potential dynamic complexity that these measurement combinations can add to the system. It should be possible to use the matrix Q to reduce the complexity of the system by, e.g., decreasing the interactions around some frequency range. The interactions can be determined using, e.g., the RGA, but it can be difficult for the designer to specify the frequency ranges. More importantly, even if the desired closed-loop bandwidth is known, there exists no simple method of choosing Q such that the interactions are improved around these frequencies. In addition, it may not even be possible to design controllers that achieve acceptable performance for the desired bandwidth. This holds in particular if the initially chosen measurement combinations H have imposed severe limitations on the achievable control performance due to RHP zeros. Similarly, gramian based methods for interaction measurements don't consider the final closed-loop performance [9]. Furthermore, the choice of Q is only relevant when there are multiple unconstrained degrees of freedom. For single input systems, the value on Q becomes static gain only and doesn't have any effect on the dynamics of the resulting system.

6.4 Conclusion

The profit for process plants is mainly determined by the steady-state economics, and thus, little emphasis has been put on the dynamic performance when selecting the measurement combination H. However, as demonstrated in this chapter, CVs consisting of measurement combinations can give rise to complex dynamic behavior which imposes limitations on the achievable performance.

It is in general, assumed that the plant has been stabilized before determining the self-optimizing control variables. Thus, as long as the feedback controllers K are not tuned too aggressively for the self-optimizing control variables, they should be able to maintain the stability of the plant. It can, therefore, be argued that using the above three-step procedure is sufficient to achieve a stable plant with acceptable dynamic and steady-state performance. However, not fully utilizing the additional tuning variables given by the matrices Q, and H when designing the closed-loop system is clearly suboptimal. In particular, if a large improvement in the dynamic response can be achieved for the same steady-state performance without needing to increase the complexity of the control structure. In the two upcoming chapters, a method is presented for selecting the measurement combination and designing the feedback controllers. Instead of choosing K, Q, and H separately, the proposed method computes them simultaneously by trying to solve a static output feedback control problem.

Chapter 7

Static output feedback control

When tuning feedback controllers, the aim is often to minimize some common control performance criterion, such as the integrated absolute error (IAE). However, in self-optimizing control, minimizing IAE may not be ideal. Instead, the self-optimizing control variables should ideally, when subjected to disturbances, drive the process to the new optimal operating point while minimizing deviations in variables with large economic impact (e.g., the active constraints). Therefore, it might be better to recast it as an optimization problem for finding, e.g., the H_2 or the H_∞ optimal static output feedback (SOF) controller that minimizes the deviations in a specified performance output. Contrary to full state-feedback or full-order controllers, which can be solved using linear matrix inequalities (LMI), structured SOF generally results in bilinear matrix inequalities (BMI) and remains an open problem [132], [114]. They are often solved to a local optimum by iteratively fixing some variables and solving the resulting LMI.

In this chapter, formulations are given that can be used for finding discrete or continuous-time SOF controllers that minimize either the H_2 or the H_∞ norm for the resulting closed-loop system. In addition, two ways of augmenting the system matrices are presented, such that the resulting SOF controller can be transformed into decentralized PI controllers that control linear measurement combinations. These formulations are based on the two-step procedure for SOF controller design, introduced in [109], [37], [94], [2], and [98]. There an optimization problem is formulated that contains a BMI, which can be made into an LMI if initialized with a stabilizing state feedback controller. The advantage of these methods is that they make it possible to decouple the controller parameters and the measurement combination when solving the optimization problem. Therefore, these SOF methods will be expanded on in Chapter 8, where the goal will be to design feedback controllers and select the self-optimizing control variables simultaneously.

7.1 Static output feedback control in continuous-time

Consider a system described by the continuous linear time-invariant state-space model,

$$\dot{x}(t) = A_c x(t) + B_c u(t) + E_c w(t), \tag{7.1}$$

$$y(t) = C_c x(t) + D_c w(t), \tag{7.2}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $w \in \mathbb{R}^{n_w}$, and $y \in \mathbb{R}^{n_y}$ are the states, inputs, disturbances, and measurements, respectively. The aim is to find a controller K,

$$u(t) = Ky(t) = K\big(C_c x(t) + D_c w(t)\big) \tag{7.3}$$

such that resulting closed-loop system

$$\dot{x}(t) = \big(A_c + B_c K C_c\big)x(t) + \big(E_c + B_c K D_c\big)w(t) \tag{7.4}$$

is stable and minimizes either the H_2 or the H_∞ norm. In the upcoming sections, two approaches are presented that tries to find the H_2 or the H_∞ optimal SOF controller. Both these methods heavily rely on the following projection lemma (also known as the elimination lemma [120]):

Lemma 7.1. *(Projection lemma [111]) Given a symmetric matrix $\Psi \in \mathbb{R}^{n\times n}$ and two matrices $U \in \mathbb{R}^{n\times l}$ and $V \in \mathbb{R}^{m\times n}$, there exists a matrix $\Xi \in \mathbb{R}^{l\times m}$ that satisfies*

$$U^T \Xi Y + Y^T \Xi U + \Psi \prec 0 \tag{7.5}$$

iff the following projection inequalities are satisfied:

$$U^{\perp T} \Psi U^{\perp} \prec 0, \tag{7.6}$$

$$V^{\perp T} \Psi V^{\perp} \prec 0. \tag{7.7}$$

Here, $U^\perp$ and $V^\perp$ are arbitrary matrices whose columns form a basis of the null spaces of U and V, respectively.

7.1.1 H_2 optimal control in continuous-time

Consider the following generalized extension of the system in (7.1) and (7.2) with $D_c = 0$:

$$Process: \begin{cases} \dot{x}(t) = A_c x(t) + B_c u(t) + E_c w(t), \\ z(t) = C_2 x(t) + D_2 u(t), \\ y(t) = C_c x(t), \end{cases} \tag{7.8}$$

where $z(t) \in \mathbb{R}^{n_z}$ is the performance output vector. The process model maps the exogenous disturbance inputs $w(t) \in \mathbb{R}^{n_w}$ and the control inputs $u(t)$ to the performance output $z(t)$ and the measured outputs $y(t)$ as shown in Figure 7.1.

Let $T_{w,z}$ represent the closed-loop system for (7.8) from the disturbance input w to the performance output z. The H_2 optimal control problem is to design the controller K such that $T_{w,z}$ is internally stable, and the H_2 norm $\|T_{w,z}\|_2$ is minimized. An upper bound for $\|T_{w,z}\|_2$ can be found using Lemma 7.2 [115], [37]:

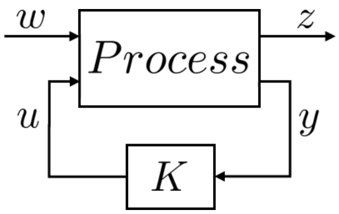


Figure 7.1: Control configuration for static output feedback control.

Lemma 7.2. *Suppose that the SOF controller K is chosen such that, the closed-loop system $T_{w,z}$ is asymptotically stable. Then for a given scalar γ, the inequality $\|T_{w,z}\|_2^2 < \gamma^2$ holds if there exists a symmetric and positive definite matrix $P \in \mathbb{R}^{n_x \times n_x}$ such that:*

$$trace(E_c^T P E_c) < \gamma^2 \tag{7.9}$$

$$A_{cl}^T P + P A_{cl} + C_{cl}^T C_{cl} \prec 0 \tag{7.10}$$

$$P \succ 0 \tag{7.11}$$

where A_{cl}, and C_{cl} are the closed-loop matrices that are defined as

$$A_{cl} := A_c + B_c K C_c, \tag{7.12}$$

$$C_{cl} := C_2 + D_2 K C_c. \tag{7.13}$$

Lemma 7.2 implies that the upper bound on $\|T_{w,z}\|_2$ can be reduced by minimizing trace$(E_c^T P E_c)$ with respect to K and P while satisfying the matrix inequalities in (7.10) and (7.11).

By introducing slack variables, [109] proposed a new parameterization method for finding the H_2 optimal SOF controller. Defining the linear function:

$$M_2(P) := \begin{bmatrix} 0 & B_c^T P \\ P B_c & A_c^T P + P A_c \end{bmatrix} + \begin{bmatrix} D_2^T \\ C_2^T \end{bmatrix} \begin{bmatrix} D_2 & C_2 \end{bmatrix}. \tag{7.14}$$

Then the following theorem gives a necessary and a sufficient condition for determining a stabilizing controller that minimizes the H_2 norm for the closed-loop system.

Theorem 7.3. *There exists a non-singular matrix $\Phi_1 \in \mathbb{R}^{n_u \times n_u}$, and a matrix $\Phi_2 \in \mathbb{R}^{n_u \times n_y}$ such that $K = \Phi_1^{-1} \Phi_2$ is a stable H_2 optimal SOF controller, iff there exists a stabilizing state feedback matrix $K_{SF} \in \mathbb{R}^{n_u \times n_x}$ and a positive definite matrix $P = P^T \in \mathbb{R}^{n_x \times n_x}$ that gives the optimal solution for the non-convex optimization problem:*

$$\Phi_1, \Phi_2 = \arg \min_{K_{SF}, P, \Phi_1, \Phi_2} trace(E_c^T P E_c) \tag{7.15}$$

$$subject\ to \quad P \succ 0 \tag{7.16}$$

$$M_2(P) + N_2 \prec 0 \tag{7.17}$$

where N_2 is defined in (7.18) with the symbol $He\{X\}$ indicating $X^T + X$:

$$N_2 := He\left\{\begin{bmatrix} I \\ -K_{SF}^T \end{bmatrix} \begin{bmatrix} \Phi_1 & -\Phi_2 C_c \end{bmatrix}\right\}. \tag{7.18}$$

Proof. Lemma 7.2 states that the closed-loop system $\|T_{w,z}\|_2^2$ has an upper bound that is given by trace$(E_c^T P E_c)$ if there exists a matrix $P = P^T \succ 0$ satisfying the inequality,

$$\begin{aligned}\left(A_c{+}B_c\Phi_1^{-1}\Phi_2C_c\right)^T P + P\left(A_c + B_c\Phi_1^{-1}\Phi_2C_c\right) \\ + \left(C_2 + D_2\Phi_1^{-1}\Phi_2C_c\right)^T\left(C_2 + D_2\Phi_1^{-1}\Phi_2C_c\right) \prec 0,\end{aligned} \tag{7.19}$$

where controller K has been replaced with $\Phi_1^{-1}\Phi_2$. Based on the proof given in [37], the above expression can be formulated as:

$$\begin{bmatrix} C_c^T\Phi_2^T\Phi_1^{-T} & I \end{bmatrix} M_2(P) \begin{bmatrix} \Phi_1^{-1}\Phi_2C_c \\ I \end{bmatrix} \prec 0 \tag{7.20}$$

$$\begin{bmatrix} I \\ -C_c^T\Phi_2^T\Phi_1^{-T} \end{bmatrix}^{\perp} M_2(P) \begin{bmatrix} I \\ -C_c^T\Phi_2^T\Phi_1^{-T} \end{bmatrix}^{\perp T} \prec 0. \tag{7.21}$$

According to Lemma 7.1, the above expressions can be shown to be equivalent to,

$$M_2(P) + He\left\{\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \begin{bmatrix} I & -\Phi_1^{-1}\Phi_2C_{yx} \end{bmatrix}\right\} \prec 0, \tag{7.22}$$

with the matrices $Z_1 \in \mathbb{R}^{n_u \times n_u}$ and $Z_2 \in \mathbb{R}^{n_x \times n_u}$. Factorizing Z_1 gives,

$$M_2(P) + He\left\{\begin{bmatrix} I \\ Z_2Z_1^{-1} \end{bmatrix} \begin{bmatrix} Z_1 & -Z_1\Phi_1^{-1}\Phi_2 \end{bmatrix}\right\} \prec 0, \tag{7.23}$$

which is equal to (7.17) with $Z_1 = \Phi_1$ and $K_{SF}^T = -Z_2Z_1^{-1}$. Finally, it can be shown that pre- and post- multiplying (7.17) with $\begin{bmatrix} K_{SF}^T & I \end{bmatrix}$ and its transpose, respectively satisfies:

$$\begin{bmatrix} K_{SF}^T & I \end{bmatrix} M_2(P) \begin{bmatrix} K_{SF} \\ I \end{bmatrix} \prec 0 \tag{7.24}$$

$$\begin{aligned}(A_c{+}B_cK_{SF})^T P + P(A_x + B_cK_{SF}) \\ + \left(C_2 + D_2K_{SF}\right)^T\left(C_2 + D_2K_{SF}\right) \prec 0.\end{aligned} \tag{7.25}$$

□

The optimization problem given in (7.15)–(7.17) is non-convex due to the BMI in (7.17), and thus, an optimal global solution can't be guaranteed. However, if an initial stable state feedback controller K_{SF} has been chosen, then the problem becomes convex and can be solved using standard semi-definite programming (SDP) solvers. The obtained solution will be dependent on the choice of K_{SF}, and it is thus, often recommended to select K_{SF} as the H_2 optimal state feedback gain since it often tends to give satisfactory solutions [109]. If the final result is not considered acceptable or if a feasible solution can't be found, then it is recommended to try several different stabilizing state feedback gains using, e.g., the method in [11].

7.1.2 H_∞ optimal control in continuous-time

Let the state-space model of the system be given by

$$Process: \begin{cases} \dot{x}(t) = A_c x(t) + B_c u(t) + E_c w(t), \\ z(t) = C_\infty x(t) + D_\infty u(t) + E_\infty w(t), \\ y(t) = C_c x(t) + D_c w(t). \end{cases} \tag{7.26}$$

The H_∞ optimal control problem consists of minimizing the H_∞ norm of the closed-loop system from exogenous disturbance signals $w(t)$ to the controlled output signals $z(t)$ [47], [125]. Defining the closed-loop matrix as

$$T_{w,z} := \left[\begin{array}{c|c} A_{cl} & B_{cl} \\ \hline C_{cl} & D_{cl} \end{array}\right] \tag{7.27}$$

with the closed loop matrices:

$$A_{cl} = A_c + B_c K C_c, \tag{7.28}$$
$$B_{cl} = E_c + B_c K D_c, \tag{7.29}$$
$$C_{cl} = C_\infty + D_\infty K C_c, \tag{7.30}$$
$$D_{cl} = D_\infty + D_\infty K D_c. \tag{7.31}$$

The objective is to find the controller K such that $\|T_{w,z}\|_\infty$ is minimized. The H_∞ norm has several interpretations regarding performance. One is that it minimizes the peak of the singular value of $T_{w,z}(j\omega)$. Alternatively, from a time-domain interpretation, it can be considered as the worst-case H_2 norm [125]:

$$\|T_{w,z}\|_\infty = \max_{w(t)\neq 0} \frac{\|z(t)\|_2}{\|w(t)\|_2} \tag{7.32}$$

Finding an H_∞ optimal controller can be difficult and therefore, in practice, it is often easier to design a suboptimal one, by obtaining the minimum upper bound γ that satisfies $\|T_{w,z}\|_\infty < \gamma$. The upper bound γ can be given by the well-known Bounded real lemma for continuous-time systems.

Lemma 7.4. *(Bounded real lemma for continuous-time systems [139]), $T_{w,z}$ is asymptotically stable and $\|T_{w,z}\|_\infty < \gamma$ iff there exists a symmetric and positive definite matrix $P \in \mathbb{R}^{n_x \times n_x}$ such that the following inequality holds:*

$$\begin{bmatrix} A_{cl}^T P + P A_{cl} + C_{cl}^T C_{cl} & P B_{cl} + C_{cl}^T D_{cl} \\ B_{cl}^T P + D_{cl}^T C_{cl} & D_{cl}^T D_{cl} - \gamma^2 I \end{bmatrix} \prec 0 \tag{7.33}$$

The parametrization for H_2 static output feedback (SOF) control that was introduced in [109] can be extended to a parametrization for the H_∞ SOF case as given in [10] and [37]. Defining the linear function:

$$\begin{aligned} M_\infty(P,\gamma) := &\begin{bmatrix} I & 0 & 0 \\ A_c & E_c & B_c \end{bmatrix}^T \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ A_c & E_c & B_c \end{bmatrix} \\ &+ \begin{bmatrix} C_\infty & E_\infty & D_\infty \\ 0 & I & 0 \end{bmatrix}^T \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} C_\infty & E_\infty & D_\infty \\ 0 & I & 0 \end{bmatrix}. \end{aligned} \tag{7.34}$$

Then, an H_∞ optimal solution for K, while ensuring a stable closed-loop system for (7.27) can be obtained from the following theorem.

Theorem 7.5. *There exists an SOF controller $K = \Phi_1^{-1}\Phi_2$, where $\Phi_1 \in \mathbb{R}^{n_u \times n_u}$ $\Phi_2 \in \mathbb{R}^{n_u \times n_y}$ that gives a stable closed-loop system and minimizes γ while achieving $\|T_{w,z}\|_\infty \leq \gamma$, if there exists matrices $K_1 \in \mathbb{R}^{n_u \times n_x}$, $K_2 \in \mathbb{R}^{n_u \times n_x}$, and a positive definite matrix $P = P^T \in \mathbb{R}^{n_x \times n_x}$ that solves the following non-convex optimization problem:*

$$\Phi_1, \Phi_2 = \arg \min_{K_1, K_2, P, \Phi_1, \Phi_2} \gamma^2 \tag{7.35}$$

$$\textit{subject to } P \succ 0 \tag{7.36}$$

$$M_\infty(P, \gamma) + N_\infty \prec 0 \tag{7.37}$$

where N_∞ is defined in (7.38) with the symbol $He\{X\}$ indicating $X^T + X$:

$$N_\infty := He\left\{ \begin{bmatrix} K_1 \\ K_2 \\ -I \end{bmatrix} \begin{bmatrix} \Phi_2 C_c & \Phi_2 D_c & \Phi_1 \end{bmatrix} \right\} \tag{7.38}$$

Proof. Similar to the proof in [10], the inequality in (7.33) can be rewritten as:

$$\begin{bmatrix} I & 0 & C_c^T K^T \\ 0 & I & D_c^T K^T \end{bmatrix} M_\infty(P, \gamma) \begin{bmatrix} I & 0 & C_c^T K^T \\ 0 & I & D_c^T K^T \end{bmatrix}^T \prec 0. \tag{7.39}$$

Replacing K with $\Phi_1^{-1}\Phi_2$, then according to Lemma 7.1, equation (7.39) is equivalent to

$$M_\infty(P, \gamma) + He\left\{ \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} \begin{bmatrix} -\Phi_1^{-1}\Phi_2 C_c & -\Phi_1^{-1}\Phi_2 D_c & I \end{bmatrix} \right\} \prec 0, \tag{7.40}$$

with the matrices $Z_1 \in \mathbb{R}^{n_x \times n_u}$, $Z_2 \in \mathbb{R}^{n_x \times n_u}$ and, $Z_3 \in \mathbb{R}^{n_u \times n_u}$. Factorizing Z_3 gives,

$$M_\infty(P, \gamma) + He\left\{ \begin{bmatrix} Z_1 Z_3^{-1} \\ Z_2 Z_3^{-1} \\ I \end{bmatrix} \begin{bmatrix} -Z_3 \Phi^{-1}\Phi_2 C_c & -Z_3 \Phi_1^{-1}\Phi_2 D_c & Z_3 \end{bmatrix} \right\} \prec 0. \tag{7.41}$$

Finally, defining $K_1 := Z_1 Z_3^{-1}$, $K_2 := Z_2 Z_3^{-1}$, and $Z_3 := -\Phi_1$ gives the expression in (7.37). □

The optimization problem in (7.35)–(7.37) include a BMI, which makes it non-convex and thus, an optimal global solution can't be guaranteed. By initializing with some stabilizing state feedback gains K_1 and K_2, the problem becomes convex, and a suboptimal solution can be obtained. The initialization variables K_1 and K_2 can be interpreted as a suboptimal solution to a convex, full information H_∞ control problem (with $C_c = \begin{bmatrix} I & 0 \end{bmatrix}^T$ and $D_c = \begin{bmatrix} 0 & I \end{bmatrix}^T$) and can, e.g., be obtained using the method described in [37].

7.2 Static output feedback control in discrete-time

Consider a system described by the discrete linear time-invariant state-space model,

$$x_{k+1} = A_d x_k + B_d u_k + E_d w_k, \tag{7.42}$$

$$y_k = C_d x_k + D_d w_k, \tag{7.43}$$

where $x_k \in \mathbb{R}^{n_x}$, $y_k \in \mathbb{R}^{n_y}$, $u_k \in \mathbb{R}^{n_u}$, $w_k \in \mathbb{R}^{n_w}$ are the states, measurements, inputs, and disturbances, respectively. The SOF control problem is to find the controller K,

$$u_k = K y_k = K\Big(C_d x_k + D_d w_k\Big), \tag{7.44}$$

such that the closed-loop system

$$x_{k+1} = \big(A_d + B_d K C_d\big) x_k + \big(E_d + B_d K D_d\big) w_k \tag{7.45}$$

is stable and possesses the desired performance characteristics.

7.2.1 H_2 optimal control in discrete-time

Let, $D_d = 0$, then a generalized extension of the system in (7.42) and (7.43) is given by,

$$Process : \begin{cases} x_{k+1} = A_d x_k + B_d u_k + E_d w_k, \\ z_k = C_2 x_k + D_2 u_k, \\ y_k = C_d x_k, \end{cases} \tag{7.46}$$

where $z_k \in \mathbb{R}^{n_z}$ is the performance output vector. The resulting plant model shown in Figure 7.1, maps the exogenous disturbance inputs $w_k \in \mathbb{R}^{n_w}$ and the control inputs u_k to the performance output z_k and the measured outputs y_k.

Let $T_{w,z}$ represent the closed-loop system for (7.46) that uses the feedback controller K. The aim is to design the controller K such that $T_{w,z}$ is internally stable, and the H_2 norm of the signals from w_k to z_k gets minimized. An upper bound for the H_2 norm of the closed-loop system can be obtained by using the following lemma [98], [32].

Lemma 7.6. *Suppose that the SOF controller K is chosen such that, the closed-loop system $T_{w,z}$ is asymptotically stable. Then for a given scalar γ, the inequality $\|T_{w,z}\|_2^2 < \gamma^2$ holds if there exists a positive definite matrix $P = P^T \in \mathbb{R}^{n_x \times n_x}$ such that:*

$$trace(E_d^T P E_d) < \gamma^2, \tag{7.47}$$

$$A_{cl}^T P A_{cl} - P + C_{cl}^T C_{cl} \prec 0, \tag{7.48}$$

$$P \succ 0, \tag{7.49}$$

where A_{cl}, and C_{cl} are the closed-loop matrices that are defined as

$$A_{cl} := A_d + B_d K C_d, \tag{7.50}$$

$$C_{cl} := C_2 + D_2 K C_d. \tag{7.51}$$

As in the continuous case, Lemma 7.6 implies that the upper bound on $\|T_{w,z}\|_2$ can be reduced by minimizing trace$(E_d^T P E_d)$ with respect to K and P while satisfying the matrix inequalities in (7.48) and (7.49). Based on the two-step procedure presented in [98] and [140], a SOF controller that minimizes the upper bound for $\|T_{w,z}\|_2$ can be found using the following theorem.

Theorem 7.7. *There exists a SOF controller $K = \Phi_1^{-1}\Phi_2$, where $\Phi_1 \in \mathbb{R}^{n_u \times n_u}$, $\Phi_2 \in \mathbb{R}^{n_u \times n_y}$ of which Φ_1 has to be non-singular that gives a stable closed-loop system and minimizes the H_2 norm, if there exists a stabilizing state feedback gain $K_{SF} \in \mathbb{R}^{n_u \times n_x}$, a positive definite matrix $P = P^T \in \mathbb{R}^{n_x \times n_x}$, and matrices, $Z_1 \in \mathbb{R}^{n_x \times n_x}$ and $Z_2 \in \mathbb{R}^{n_z \times n_z}$ that solves the non-convex optimization problem:*

$$\Phi_1, \Phi_2 := \arg \min_{K_{SF}, P, \Phi_1, \Phi_2, Z_1, Z_2} \mathit{trace}(E_d^T P E_d) \tag{7.52}$$

$$\textit{subject to} \quad P \succ 0 \tag{7.53}$$

$$W_2 \prec 0 \tag{7.54}$$

where

$$W_2 := \begin{bmatrix} -P & * & * & * \\ Z_1^T A_d + Z_1^T B_d K_{SF} & P - Z_1 - Z_1^T & * & * \\ Z_2^T C_2 + Z_2^T D_2 K_{SF} & 0 & I - Z_2 - Z_2^T & * \\ \Phi_2 C_d - \Phi_1 K_{SF} & B_d^T Z_1 & D_2^T Z_2 & -\Phi_1 - \Phi_1^T \end{bmatrix}. \tag{7.55}$$

Proof. Defining,

$$S := \begin{bmatrix} I & 0 & 0 & \Phi_1^{-1}(\Phi_2 C_2)^T \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \end{bmatrix}, \tag{7.56}$$

and multiplying the expression in (7.55) with S from the left and S^T from the right gives:

$$\begin{bmatrix} -P & A_{cl}^T Z_1 & C_{cl}^T Z_2 \\ Z_1^T A_{cl} & P - Z_1 - Z_1^T & 0 \\ Z_2^T C_{cl} & 0 & I - Z_2 - Z_2^T \end{bmatrix} \prec 0. \tag{7.57}$$

Finally, by multiplying (7.57) with $\begin{bmatrix} I & A_{cl}^T & C_{cl}^T \end{bmatrix}$ from the left and its transpose from the right side gives

$$A_{cl}^T P A_{cl} - P + C_{cl}^T C_{cl} \prec 0,$$

which is equal to (7.48) in Lemma 7.6. □

The inequality in (7.54) makes the problem given in (7.52)–(7.54) non-convex. However, the problem can be made convex if an initial stable state feedback controller K_{SF} has been chosen for the inequality in (7.54).

7.2.2 H_∞ optimal control in discrete-time

Consider the discrete-time LTI system where the performance output vector $z_k \in \mathbb{R}^{n_z}$ has been added to the system in (7.112) and (7.113):

$$Process: \begin{cases} x_{k+1} &= A_d x_k + B_d u_k + E_d w_k, \\ z_k &= C_\infty x_k + D_\infty u_k + E_\infty w_k, \\ y_k &= C_d x_k + D_d w_k. \end{cases} \tag{7.58}$$

The H_∞ optimal control problem then consists of minimizing the H_∞ norm of the closed-loop system from the exogenous disturbance signals w_k to the performance output signals z_k [47], [125]. Defining the closed-loop matrices,

$$A_{cl} = A_d + B_d K C_d, \tag{7.59}$$

$$B_{cl} = E_d + B_d K D_d, \tag{7.60}$$

$$C_{cl} = C_\infty + D_\infty K C_d, \tag{7.61}$$

$$D_{cl} = E_\infty + D_\infty K D_d, \tag{7.62}$$

where K is the static output feedback controller. When a state feedback controller is used, then KC_d, KD_d, KC_d, and KD_d in (7.59)–(7.62) are replaced with the state feedback gain K_{SF}. If the resulting closed-loop system is defined as

$$T_{w,z} := \left[\begin{array}{c|c} A_{cl} & B_{cl} \\ \hline C_{cl} & D_{cl} \end{array}\right], \tag{7.63}$$

then the objective is to find the static output feedback controller K such that $\|T_{w,z}\|_\infty$ is minimized. Next, let's recall the well-known Bounded real lemma for discrete-time systems.

Lemma 7.8. *(Bounded real lemma for discrete-time systems [47]), $T_{w,z}$ is asymptotically stable and $\|T_{w,z}\|_\infty < \gamma$ iff there exists a symmetric and positive definite matrix $P \in \mathbb{R}^{n_x \times n_x}$ such that the following inequality holds:*

$$\begin{bmatrix} A_{cl}^T P A_{cl} - P & A_{cl}^T P B_{cl} & C_{cl}^T \\ B_{cl}^T P A_{cl} & B_{cl}^T P B_{cl} - \gamma^2 I & D_{cl}^T \\ C_{cl} & D_{cl} & -I \end{bmatrix} \prec 0. \tag{7.64}$$

Designing an H_∞ optimal controller can be challenging, and therefore it is common to instead find a suboptimal one, by obtaining the minimum upper bound γ that satisfies $\|T_{w,z}\|_\infty < \gamma$. Based on Lemma 7.1 and Lemma 7.8, an H_∞ optimal solution for K that ensures a stable closed-loop and a minimum upper bound γ for $\|T_{w,z}\|_\infty$ can be obtained from the following theorem.

Theorem 7.9. *There exists a SOF controller $K = \Phi_1^{-1}\Phi_2$, where $\Phi_1 \in \mathbb{R}^{n_u \times n_u}$, $\Phi_2 \in \mathbb{R}^{n_u \times n_y}$ of which Φ_1 has to be non-singular that gives a stable closed-loop system and minimizes γ while achieving $\|T_{w,z}\|_\infty \le \gamma$, if there exists a stabilizing state feedback gain $K_{SF} \in \mathbb{R}^{n_u \times n_x}$, a positive definite matrix $P = P^T \in \mathbb{R}^{n_x \times n_x)}$,* *and matrices Z_1, $Z_2 \in \mathbb{R}^{n_x \times n_x}$ that solves the following non-convex optimization problem:*

$$\begin{aligned}
\Phi_1, \Phi_2 &:= \arg \min_{K_{SF}, Z_1, Z_2, P, \Phi_1, \Phi_2} \gamma^2 && (7.65)\\
&\textit{subject to } P \succ 0 && (7.66)\\
&W_\infty + V_\infty \prec 0 && (7.67)
\end{aligned}$$

where W_∞ is defined

$$W_\infty := \begin{bmatrix} \Theta_{11} & * & * & * & * \\ \Theta_{21} & \Theta_{22} & * & * & * \\ B_d^T Z_2^T & B_d^T Z_1^T & 0 & * & * \\ \Theta_{41} & 0 & D_\infty & -I & * \\ E_d^T Z_2^T & E_d^T Z_1^T & 0 & E_\infty^T & -\gamma^2 I \end{bmatrix}, \tag{7.68}$$

with

$$\begin{aligned}
\Theta_{11} &= -P + (A_d + B_d K_{SF})^T Z_2^T + Z_2 (A_d + B_d K_{SF}),\\
\Theta_{21} &= -Z_2^T + Z_1 (A_d + B_d K_{SF}),\\
\Theta_{22} &= P - Z_2 - Z_2^T,\\
\Theta_{41} &= C_\infty + D_\infty K_{SF},
\end{aligned}$$

and V_∞ is given by

$$V_\infty := \begin{bmatrix} 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ \Phi_2 C_{yx} - \Phi_1 K_{SF} & 0 & -\Phi_1 - \Phi_1^T & * & * \\ 0 & 0 & 0 & 0 & * \\ 0 & 0 & D_d^T \Phi_2^T & 0 & 0 \end{bmatrix}. \tag{7.69}$$

Proof. Similar to the proof in [94], the expression in (7.67) can be rewritten as

$$U \Phi_1 V^T + V \Phi_1^T U^T + W_\infty \prec 0, \tag{7.70}$$

where V and U are defined as

$$\begin{aligned}
V &:= \begin{bmatrix} \Phi_1^{-1} \Phi_2 C_d - K_{SF} & 0 & -I & 0 & \Phi_1^{-1} \Phi_2 D_d \end{bmatrix}^T, && (7.71)\\
U &:= \begin{bmatrix} 0 & 0 & I & 0 & 0 \end{bmatrix}^T. && (7.72)
\end{aligned}$$

Choosing the matrices,

$$V^\perp = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ \Phi_1^{-1} \Phi_2 D_d & 0 & 0 & \Phi_1^{-1} \Phi_2 C_d - K_{SF} \\ 0 & 0 & I & 0 \\ I & 0 & 0 & I \end{bmatrix}, \tag{7.73}$$

$$U^\perp = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}, \tag{7.74}$$

whose columns form the null spaces of V, and U, respectively. Then, according to Lemma 7.1, the expression in (7.70) is equivalent to

$$\begin{bmatrix} A_{cl}^T Z_1^T + Z_1 A_{cl} - P & * & * & * \\ -Z_1^T + Z_2 A_{cl} & P - Z_2 - Z_2^T & * & * \\ C_{cl} & 0 & -I & * \\ B_{cl}^T Z_1^T & B_{cl}^T Z_2^T & D_{cl}^T & -\gamma^2 I \end{bmatrix} \prec 0, \tag{7.75}$$

which is a multiplication of W_∞ in (7.70) by $V^{\perp T}$ on the left and $V^\perp$ on the right. Replacing K in (7.59)–(7.62), with $\Phi_1^{-1}\Phi_2$, guarantees the stability of the closed-loop system when using the static output feedback controller. Multiplying W_∞ in (7.70) by $U^{\perp T}$ on the left and $U^\perp$ on the right gives the second condition in Lemma 7.1 and ensures stability for the system when using the state feedback controller K_{SF}. The resulting expression is the same as in (7.75), but with K_{SF} replacing the static output feedback controllers for A_{cl}, B_{cl}, C_{cl}, and D_{cl}. Finally, multiplying (7.75) with β^T on the left and β on the right, where

$$\beta := \begin{bmatrix} I & 0 & 0 \\ A_{cl} & B_{cl} & 0 \\ 0 & 0 & I \\ 0 & I & 0 \end{bmatrix}, \tag{7.76}$$

results in the Bounded real lemma, given in Lemma 7.8. □

The optimization problem in (7.65)–(7.67) requires solving a BMI, which is NP-hard and thus, difficult to solve. However, by specifying a stable state feedback gain K_{SF}, it becomes a convex LMI, from which a local optimum can be found. The final SOF controller can be obtained from $K = \Phi_1^{-1}\Phi_2$.

7.3 Static output feedback control with application to PI control

The proportional integral (PI) controller is, by far the most commonly used controller in the process industries due to its simplicity and robust performance [13]. With progress in numerical methods, new convex optimization methods have been developed for designing controllers. However, for restricted-order controllers (e.g., PI/PID controller) the optimization problems tend to become non-convex in the controller parameter space. They are usually solved by employing heuristics or intelligent methods [66], [48]. A loop shaping method was proposed in [73] by specifying bounds on the phase and gain margins.

In this section, two methods of augmenting the system matrices are proposed for finding PI controllers together with a measurement combination H when using the prescribed formulations for SOF controllers. The two different ways of augmenting the process are based on whether a PI controller is in its ideal or parallel form is used.

Although most PI controllers used in commercial implementation are in its ideal form, the parallel form can often be seen for algorithms that try to compute the optimal controller parameters. The reason for this is that keeping the variables independent is in general advantageous when solving optimization problems. From a practical perspective, it does not matter which form is used since it is always possible to switch from one to the other. However, for this application, both forms can be used, where the final controller type depends on how the system matrices are augmented.

7.3.1 Parallel form

PI controllers in their parallel form can be formulated in either continuous-time or discrete-time and are given by,

Continuous-time **Discrete-time**

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau, \qquad u_k = K_p e_k + K_i \sum_{n=0}^{k-1} e_n. \tag{7.77}$$

Here, $u(t), u_k \in \mathbb{R}^{n_u}$ represent the control variable, and $e(t), e_k \in \mathbb{R}^{n_u}$ are the error values. The parameters $K_p \in \mathbb{R}^{n_u \times n_u}$ and $K_i \in \mathbb{R}^{n_u \times n_u}$ are diagonal matrices, representing the proportional and integral gains, respectively:

$$K_p := \begin{bmatrix} k_{p,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & k_{p,n_u} \end{bmatrix}, \quad K_i := \begin{bmatrix} k_{i,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & k_{i,n_u} \end{bmatrix}. \tag{7.78}$$

For discrete-time systems, the sampling time is assumed to be included in K_i, i.e., K_i will then vary depending on the sampling time.

When a measurement combination $H \in \mathbb{R}^{n_u \times n_y}$ is used, then the error values are defined as $e(t) := r(t) - Hy(t)$, and $e_k := r_k - Hy_k$, where $r(t), r_k \in \mathbb{R}^{n_u}$ are the reference inputs, and $y(t), y_k \in \mathbb{R}^{n_y}$ are the available measurements. Setting the reference values to $r(t), r_k = 0$ (using deviation variables), then the error values become $e(t) = -Hy(t)$ and $e_k = -Hy_k$ for the continuous-time and discrete-time case, respectively. The PI controllers in (7.77) can then be expressed as

$$u(t) = -K_p H y(t) - K_i H \int_0^t y(\tau)\, d\tau, \quad u_k = -K_p H y_k - K_i H \sum_{n=0}^{k-1} y_n. \tag{7.79}$$

To include the integrating (or the summation) states from the PI controllers, the original state vectors can be replaced by the augmented state vectors $\bar{x}(t), \bar{x}_k \in \mathbb{R}^{(n_x+n_u)}$:

$$\bar{x}(t) := \begin{bmatrix} x(t) \\ q(t) \end{bmatrix}, \qquad \bar{x}_k := \begin{bmatrix} x_k \\ q_{k-1} \end{bmatrix}, \tag{7.80}$$

Here, $q(t), q_{k-1} \in \mathbb{R}^{n_u}$ are defined,

$$q(t) := \int_0^t K_i H y(\tau) d\tau, \qquad q_k := \sum_{n=0}^{k} K_i H y_n, \tag{7.81}$$

and thus, the PI controllers can be represented as

$$u(t) = -\Big(K_p H y(t) + q(t)\Big), \qquad u_k = -\Big(K_p H y_k + q_{k-1}\Big). \tag{7.82}$$

Defining $K_{PI} \in \mathbb{R}^{2n_u \times 2n_u}$, and $\mathcal{I} \in \mathbb{R}^{2n_u \times n_u}$,

$$K_{PI} = \begin{bmatrix} K_p & 0 \\ 0 & K_i \end{bmatrix}, \ \mathcal{I} = \begin{bmatrix} I \\ I \end{bmatrix}, \tag{7.83}$$

such that the SOF controller $K \in \mathbb{R}^{2n_u \times 2n_u}$ becomes

$$K = -K_{PI}\mathcal{I}\hat{H}. \tag{7.84}$$

Then the augmented control input vectors $\bar{u}(t), \bar{u}_k \in \mathbb{R}^{2n_u}$ will be given by the control laws:

$$\bar{u}(t) = -Ky(t) = -\begin{bmatrix} K_p H y(t) \\ K_i \hat{H} y(t) \end{bmatrix}, \qquad \bar{u}_k = -Ky_k = -\begin{bmatrix} K_p H y_k \\ K_i H y_k \end{bmatrix}. \tag{7.85}$$

The closed-loop system for the continuous-time process model in (7.1)–(7.2) with the PI controllers in (7.82) can be formulated using the augmented model,

$$\dot{\bar{x}}(t) = \bar{A}_c \bar{x}(t) + \bar{B}_c \bar{u}(t) + \bar{E}_c w(t), \tag{7.86}$$

$$y(t) = \bar{C}_c \bar{x}(t) + D_c w(t), \tag{7.87}$$

where the augmented system matrices are:

$$\bar{A}_c = \begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix}, \bar{B}_c = \begin{bmatrix} B_c & 0 \\ 0 & I \end{bmatrix}, \bar{E}_c = \begin{bmatrix} E_c \\ 0 \end{bmatrix}, \bar{C}_c = \begin{bmatrix} C_c & 0 \end{bmatrix}. \tag{7.88}$$

Using the proposed control law for $\bar{u}(t)$ in (7.85), which is equivalent to

$$\bar{u}(t) = -\begin{bmatrix} K_p \hat{H} C_c x(t) + K_p H D_c w_k \\ K_i H C_c x(t) + K_i \hat{H} D_c w_k \end{bmatrix}, \tag{7.89}$$

on the augmented process model in (7.86) and (7.87) gives

$$\begin{bmatrix} \dot{x}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} A_c - B_c K_p \hat{H} C_c & B_c \\ K_i H C_c & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} E_c - B_c K_p H D_c \\ K_i H D_c \end{bmatrix} w(t). \tag{7.90}$$

The expression in (7.90) then becomes equivalent to

$$\dot{\bar{x}}(t) = \big(\bar{A}_c + \bar{B}_c K \bar{C}_c\big)\bar{x}(t) + \big(\bar{E}_c + \bar{B}_c K \bar{D}_c\big) w(t), \tag{7.91}$$

which resembles the closed-loop system seen in (7.4), where the controller K is given by (7.84).

For a discrete-time system, the augmented state-space model for the process in (7.42)–(7.43) becomes

$$\bar{x}_{k+1} = \bar{A}_d \bar{x}_k + \bar{B}_d \bar{u}_k + \bar{E}_d w_k, \tag{7.92}$$

$$y_k = \bar{C}_d \bar{x}_k + \bar{D}_d w_k, \tag{7.93}$$

with the augmented state-space models:

$$\bar{A}_d = \begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix}, \bar{B}_u = \begin{bmatrix} B_d & 0 \\ 0 & I \end{bmatrix}, \bar{E}_d = \begin{bmatrix} E_d \\ 0 \end{bmatrix}, \bar{C}_d = \begin{bmatrix} C_d & 0 \end{bmatrix}. \tag{7.94}$$

Inserting $\bar{u}_k = -K y_k$ in (7.92), where K is the controller in (7.104), and $\bar{y}_k$ is the augmented measurement vector from (7.93), will result in

$$\begin{bmatrix} x_{k+1} \\ q_k \end{bmatrix} = \begin{bmatrix} A_d - B_d K_p H C_d & -B_d \\ K_i H C_d & I \end{bmatrix} \begin{bmatrix} x_k \\ q_{k-1} \end{bmatrix} + \begin{bmatrix} E_d - B_d H D_d \\ K_i H D_d \end{bmatrix} w_k, \tag{7.95}$$

which gives the closed-loop system in (7.45) with augmented matrices:

$$\bar{x}_{k+1} = \big(\bar{A}_d + \bar{B}_d K \bar{C}_d\big)\bar{x}_k + \big(\bar{E}_d + \bar{B}_u K \bar{D}_d\big) w_k. \tag{7.96}$$

A measurement combination H that is being controlled by decentralized PI can, therefore, be formulated as the SOF controller in (7.84) using either the augmented system matrices in (7.88) for continuous-time systems or the ones in (7.94) for discrete-time systems.

7.3.2 Ideal form

In their ideal form, PI controllers in continuous-time and discrete-time can be formulated as

Continuous-time: **Discrete-time:**

$$u(t) = K_p\Big(e(t) + K_i \int_0^t e(\tau)d\tau\Big), \qquad u_k = K_p\Big(e_k + K_i \sum_{n=0}^{k-1} e_n\Big), \tag{7.97}$$

where the variables have been defined for (7.77), with the controller gains given in (7.78). When a measurement combination H is controlled using deviation variables, the PI controllers in (7.97) can be formulated as

$$u(t) = -K_p\Big(Hy(t) + K_i \int_0^t Hy(\tau)d\tau\Big), \quad u_k = -K_p\Big(Hy_k + K_i \sum_{n=0}^{k-1} Hy_n\Big). \tag{7.98}$$

From the above formulation, it can be seen that this description is overparameterized, where K_p can be considered as a simple scaling value for H. Therefore, K_p can be selected to be any non-zero value, e.g., setting $K_p = I$ gives:

$$u(t) = -\Big(Hy(t) + K_i \int_0^t Hy(\tau)d\tau\Big), \qquad u_k = -\Big(Hy_k + K_i \sum_{n=0}^{k-1} Hy_n\Big). \tag{7.99}$$

Defining the auxiliary state vectors $q(t) \in \mathbb{R}^{n_u}, q_k \in \mathbb{R}^{n_u}$ as the integral term for the summation term in (7.98):

$$q(t) = \int_0^t Hy(\tau)d\tau, \qquad q_k = \sum_{n=0}^{k} Hy_n \tag{7.100}$$

such that the PI controllers can be represented as

$$u(t) = -\Big(Hy(t) + K_i q(t)\Big), \qquad u_k = -\Big(Hy_k + K_i q_k\Big). \tag{7.101}$$

By introducing the augmented state vectors $\bar{x}(t), \bar{x}_k \in \mathbb{R}^{(n_x+n_u)}$

$$\bar{x}(t) := \begin{bmatrix} x(t) \\ q(t) \end{bmatrix}, \qquad \bar{x}_k := \begin{bmatrix} x_k \\ q_{k-1} \end{bmatrix}, \tag{7.102}$$

the augmented measurement vectors $\bar{y}(t), \bar{y}_k \in \mathbb{R}^{(n_y+n_u)}$

$$\bar{y}(t) := \begin{bmatrix} y(t) \\ q(t) \end{bmatrix}, \qquad \bar{y}_k := \begin{bmatrix} y_k \\ q_{k-1} \end{bmatrix}, \tag{7.103}$$

and using the SOF controller $K \in \mathbb{R}^{2n_u \times 2n_u}$,

$$K = -\begin{bmatrix} H & 0 \\ 0 & K_i \end{bmatrix}, \tag{7.104}$$

then the augmented control input vectors $\bar{u}(t), \bar{u}_k \in \mathbb{R}^{2n_u}$ are given by

$$\bar{u}(t) := -K\bar{y}(t) = -\begin{bmatrix} Hy(t) \\ K_i q(t) \end{bmatrix}, \qquad \bar{u}_k := -K\bar{y}_k = -\begin{bmatrix} Hy_k \\ K_i q_{k-1} \end{bmatrix}. \tag{7.105}$$

The closed-loop system for the continuous-time process model in (7.1)–(7.2) with the PI controllers in (7.101) can be given by the augmented model

$$\dot{\bar{x}}(t) = \bar{A}_c \bar{x}(t) + \bar{B}_c \bar{u}(t) + \bar{E}_c w(t), \tag{7.106}$$

$$\bar{y}(t) = \bar{C}_c \bar{x}(t) + \bar{D}_c w(t), \tag{7.107}$$

where the augmented system matrices are:

$$\bar{A}_c = \begin{bmatrix} A_c & 0 \\ 0 & 0 \end{bmatrix}, \bar{B}_c = \begin{bmatrix} B_c & B_c \\ I & 0 \end{bmatrix}, \bar{E}_c = \begin{bmatrix} E_c \\ 0 \end{bmatrix}, \bar{C}_c = \begin{bmatrix} C_c & 0 \\ 0 & I \end{bmatrix}, \bar{D}_c = \begin{bmatrix} D_c \\ 0 \end{bmatrix}. \tag{7.108}$$

Using the proposed control law for $\bar{u}(t)$ in (7.105), which is equivalent to

$$\bar{u}(t) = -\begin{bmatrix} HC_c x(t) + HD_c w_k \\ K_i q(t) \end{bmatrix}, \tag{7.109}$$

then it can easily be shown that the augmented process model in (7.106) and (7.107) becomes:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} A_c - B_c \hat{H} C_c & -B_c K_i q(t) \\ HC_c & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} E_c - B_c \hat{H} D_c \\ HD_c \end{bmatrix} w(t). \tag{7.110}$$

The expression in (7.110) is equivalent to

$$\dot{\bar{x}}(t) = \big(\bar{A}_c + \bar{B}_c K \bar{C}_c\big)\bar{x}(t) + \big(\bar{E}_c + \bar{B}_c K \bar{D}_c\big) w(t), \tag{7.111}$$

which corresponds to a closed-loop system in (7.4) that uses the augmented state-space model matrices in (7.108) with the PI controller in (7.101).

Similarly, the augmented model for the discrete-time process in (7.42)–(7.43), can be given by,

$$\bar{x}_{k+1} = \bar{A}_d \bar{x}_k + \bar{B}_d \bar{u}_k + \bar{E}_d w_k, \tag{7.112}$$

$$\bar{y}_k = \bar{C}_d \bar{x}_k + \bar{D}_d w_k, \tag{7.113}$$

with the augmented system matrices:

$$\bar{A}_d = \begin{bmatrix} A_d & 0 \\ 0 & I \end{bmatrix}, \bar{B}_d = \begin{bmatrix} B_d & B_d \\ I & 0 \end{bmatrix}, \bar{E}_d = \begin{bmatrix} E_d \\ 0 \end{bmatrix}, \bar{C}_d = \begin{bmatrix} C_d & 0 \\ 0 & I \end{bmatrix}, \bar{D}_d = \begin{bmatrix} D_d \\ 0 \end{bmatrix}. \tag{7.114}$$

Using the controller from (7.104) with the augmented measurements in (7.103) such that $\bar{u}_k = -K\bar{y}_k$, and inserting it in (7.112)–(7.113) gives

$$\begin{bmatrix} x_{k+1} \\ q_k \end{bmatrix} = \begin{bmatrix} A_d - B_d H C_d & -B_d K_i q(t) \\ H C_d & I \end{bmatrix} \begin{bmatrix} x_k \\ q_{k-1} \end{bmatrix} + \begin{bmatrix} E_d - B_d H D_d \\ H D_d \end{bmatrix} w_k, \tag{7.115}$$

which is equivalent to the closed-loop system in (7.45) with augmented matrices:

$$\bar{x}_{k+1} = \big(\bar{A}_d + \bar{B}_d K \bar{C}_d\big)\bar{x}_k + \big(\bar{E}_d + \bar{B}_d K \bar{D}_d\big) w_k. \tag{7.116}$$

Therefore, ideal PI controllers that are controlling a measurement combination H can be expressed as the SOF controller in (7.104) by using the augmented system matrices in (7.108) for continuous-time. Alternatively, in case the process is represented with discrete-time models, then the augmented matrices (7.114) can be used to formulate the SOF controller as PI controllers in their ideal form.

7.4 Conclusion

In the previous chapter, the dynamic effects of using measurement combinations as CVs were investigated. It was shown that the location of the system zeros, together with the pole and zero directions are dependent on how the measurement combinations are selected. As a consequence, the interactions and the achievable control performance will be partly determined by the chosen CVs. However, the behavior of the resulting closed-loop system will be dependent on both the measurement combination and the feedback controllers and, thus, ideally, they should be chosen simultaneously. Therefore, in this chapter, SOF control methods were presented for both continuous-time and discrete-time systems that either minimizes the H_2 norm or the H_∞ norm of the resulting closed-loop system. To have the computed SOF gain include both the feedback controllers and measurement combinations, methods for augmenting the system matrices were presented. The resulting feedback controller will consist of a decentralized PI controller, in its ideal or parallel form, depending on how the system matrices are augmented.

The proposed augmented system matrices in (7.88), (7.94), (7.108), and (7.114) differs from how state-space models typically are augmented with PI controllers

[113], [87]. E.g., the length of the augmented control input vectors has been doubled by separating the effects of the proportional and the integral part from each other. The benefit of this approach is that it will allow for decoupling the measurement combination and the controller parameters from each other in the optimization problem. Therefore, it will significantly reduce the complexity of the optimization problem, which will be advantageous when the aim is to simultaneously design the controllers and self-optimizing control variables in the next chapter.

Chapter 8

Controller design for self-optimizing control variables

The main focus of self-optimizing control is to select CVs, that often, consists of linear measurement combinations, such that when controlled at constant set-point leads to an improvement in the economic steady-state behavior of a process plant. However, as demonstrated in Chapter 6, the chosen measurement combinations will also heavily influence the dynamic behavior of the resulting closed-loop system. Thus, when determining these measurement combinations, it would be advantageous to consider both the steady-state the dynamic behavior for the resulting plant especially, if the robustness and transient response for the closed-loop system can be improved without sacrificing the economic performance.

Typically, the self-optimizing control variables are controlled by the remaining degrees of freedom, once the plant first has been stabilized, and all the active constraints are being controlled. Although it is economically optimal to operate the plant as close as possible to its active constraints, it is usually necessary to employ some "back off" to avoid dynamic and steady-state problems. "Back off" is the difference between the optimal set-point and the actual set-point that has been estimated based on the information of the disturbances and the expected control performance [52], [12]. Therefore, the self-optimizing control variables should preferably, when subjected to disturbances, drive the process to the new optimal operating point while minimizing deviations in the active constraints (i.e., reducing the "back off") or in other variables with large economic impact. If there are large deviations from the optimal operation, caused by, e.g., external disturbances, it would evidently result in an economic loss and could violate some of the operating constraints. Therefore, both the steady-state (economic) objective and the dynamic performance of the process should be considered when designing the control system.

The resulting closed-loop system is not just dependent on the measurement combination, but also on the feedback controllers. In this chapter, several optimization problems are presented with the aim to simultaneously obtain a measurement combination together with the tuning parameters for decentralized PI controllers, thus making it possible to account for both the dynamic and the steady-state performance of the resulting control system.

8.1 PI controllers with measurement combinations

In the previous chapter, four different optimization problems were formulated for finding SOF controllers K that were based on the two-step procedures given in [109], [37], [94], [2], and [98]. For continuous-time systems, Theorem 7.3, and Theorem 7.5 gave SOF controllers that minimized the H_2 norm, or the H_∞ norm for the closed-loop system, respectively. If the process model is given in discrete-time, then Theorem 7.7 can be used to find the H_2 optimal SOF controller, whereas solving the optimization problem in Theorem 7.9 will result in SOF controller that minimizes the upper bound of the H_∞ norm of the closed-loop system. In all these cases, the resulting SOF controller can be obtained from

$$K = \Phi_1^{-1}\Phi_2, \tag{8.1}$$

where Φ_1 and Φ_2 are decision variables that will be computed when solving one of the optimization problems in Theorem 7.3, 7.5, 7.7 or 7.9. In Section 7.3.1, two approaches were proposed for augmenting the system matrices such that a SOF gain K could be represented by a measurement combination H that is being controlled using decentralized PI controllers. Therefore, by replacing the original state-space models with these augmented matrices, it becomes possible to simultaneously compute the controllers and the self-optimizing control variables when using one of the SOF techniques from Chapter 7.

8.1.1 Decentralized PI controllers in their parallel form

If the PI controllers are in their parallel form, then the state-space model can be augmented as in (7.88), or as in (7.94), depending on whether it is in continuous-time or discrete-time. The resulting SOF controller can then be given as in (7.84):

$$K = -K_{PI}\mathcal{I}H, \tag{8.2}$$

where K_{PI} and $\mathcal{I}$ are defined in (7.83). Let X be a diagonal matrix that represents the inverse of the PI parameters in K_{PI}, i.e., $X = K_{PI}^{-1}$. Defining Φ_1 and Φ_2 as

$$\Phi_1 := X, \;\; \Phi_2 := \mathcal{I}H, \tag{8.3}$$

then the SOF controller in (8.2) will correspond to a measurement combination H and decentralized PI controllers (in their parallel form) that are given by $K_{PI} = X^{-1}$. Hence, for continuous-time systems, Theorem 7.3, and Theorem 7.5 can be modified as described in the following theorem:

Theorem 8.1. *When the continuous-time system model is replaced with the augmented matrices in (7.88), then there exists decentralized PI controllers in their parallel form and a measurement combination $H \in \mathbb{R}^{n_u \times n_y}$ that gives a stable closed-loop system and minimizes the upper bound of the H_2 norm or the H_∞ norm, if there exist a diagonal matrix $X \in \mathbb{R}^{2n_u \times 2n_u}$ that is non-singular and a positive definite matrix $P = P^T \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$ together with either the stabilizing state feedback gain $K_{SF} \in \mathbb{R}^{2nu\times(nx+nu)}$ for the H_2 optimal control problem, or the matrices $K_1, K_2 \in \mathbb{R}^{2n_u\times(n_x+n_u)}$ for the H_∞ optimal control problem as the* *solution to one of the following non-convex optimization problems:*

Continuous-time H_2 optimal:

$$\min_{K_{SF},P,X,H} \; trace(\bar{E}_d^T P \bar{E}_d) \tag{8.4}$$

subject to

$$P \succ 0 \tag{8.5}$$

$$X = diag(x_1 \dots x_{2nu}) \tag{8.6}$$

$$M_2 + N_2 \prec 0 \tag{8.7}$$

Continuous-time H_∞ optimal:

$$\min_{K_1,K_2,P,X,H} \; \gamma^2 \tag{8.8}$$

subject to

$$P \succ 0 \tag{8.9}$$

$$X = diag(x_1 \dots x_{2nu}) \tag{8.10}$$

$$M_\infty + N_\infty \prec 0 \tag{8.11}$$

The matrices M_2, N_2, M_∞, and N_∞ are given by (7.14) ,(7.18), (7.34), and (7.38), respectively, where Φ_1 and Φ_2 in N_2 and N_∞ have been replaced by the definition in (8.3).

Proof. The proofs that were given for Theorem 7.3 and Theorem 7.5 ensures that the closed-loop system with the resulting SOF controller $K = X^{-1}\mathcal{I}H$ is stable and minimizes the upper bound for the H_2 norm or the H_∞ norm. The parameters for the decentralized PI controllers can then be obtained from $K_{PI} = X^{-1}$ as long as X is a diagonal and a non-singular matrix. □

Similarly, Theorem 7.7, and 7.9 can be adapted to find a measurement combination H, and discrete-time PI controllers in their parallel form that ensures a stable closed-loop with an H_2, or an H_∞ optimal solution.

Theorem 8.2. *When the discrete-time system model is replaced with the augmented matrices in (7.94), then there exist decentralized PI controllers in their parallel form together with a measurement combination $H \in \mathbb{R}^{n_u \times n_y}$ that gives a stable closed-loop system and minimizes the upper bound of the H_2 norm or the H_∞ norm, if there exists a stabilizing state feedback gain $K_{SF} \in \mathbb{R}^{2n_u \times (n_x+n_u)}$, a non-singular diagonal matrix $X \in \mathbb{R}^{n_u \times n_u}$, a positive definite matrix $P = P^T \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$, and matrices Z_1, $Z_2 \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$ that solves one of the following non-convex optimization problems:*

Discrete-time H_2 optimal:

$$\min_{\substack{K_{SF},Z_1,Z_2,\\ P,X,,H}} \; trace(\bar{E}_d^T P \bar{E}_d) \tag{8.12}$$

subject to

$$P \succ 0 \tag{8.13}$$

$$X = diag(x_1 \dots x_{2nu}) \tag{8.14}$$

$$W_2 \prec 0 \tag{8.15}$$

Discrete-time H_∞ optimal:

$$\min_{\substack{K_{SF},Z_1,Z_2\\ P,X,H}} \; \gamma^2 \tag{8.16}$$

subject to

$$P \succ 0 \tag{8.17}$$

$$X = diag(x_1 \dots x_{2nu}) \tag{8.18}$$

$$W_\infty + V_\infty \prec 0 \tag{8.19}$$

The matrices W_2, W_∞, and V_∞ are given by (7.55), (7.68), and (7.69), respectively, where Φ_1 and Φ_2 in W_2 and V_∞ have been replaced by the definition in (8.3).

Proof. The proofs that were given for Theorem 7.7 and Theorem 7.9 ensures that the closed-loop system with the resulting SOF controller $K = X^{-1}\mathcal{I}H$ is stable and minimizes an upper bound for the H_2 norm or the H_∞ norm. The parameters for the decentralized PI controllers can then be obtained from $K_{PI} = X^{-1}$ as long as X is a diagonal and non-singular matrix. □

8.1.2 Decentralized PI controllers in their ideal form

When the augmented process models described in (7.108) for continuous-time systems or (7.114) for discrete-time systems are used, then PI controllers in their ideal form can be represented by the SOF controller given in (7.104):

$$K = -\begin{bmatrix} H & 0 \\ 0 & K_i \end{bmatrix}. \tag{8.20}$$

Since the steady-state loss for using $Q^{-1}H$ will be equivalent to H, the SOF controller in (8.20) can be replaced with

$$K = -\begin{bmatrix} Q^{-1}H & 0 \\ 0 & K_i \end{bmatrix}, \tag{8.21}$$

as long as $Q \in \mathbb{R}^{n_u \times n_u}$ is non-singular. Thus, by redefining Φ_1, and Φ_2, such that

$$\Phi_1 := \begin{bmatrix} Q & 0 \\ 0 & X_1 \end{bmatrix}, \quad \Phi_2 := \begin{bmatrix} H & 0 \\ 0 & X_2 \end{bmatrix}, \tag{8.22}$$

where $X_1, X_2 \in \mathbb{R}^{n_u \times n_u}$ are diagonal matrices, of which X_1 is non-singular, then the SOF controller in (8.2) can be expressed as the measurement combination $\hat{H} = Q^{-1}H$ and the PI controller with tuning parameters:

$$K_p = I, \quad K_i = X_1^{-1}X_2. \tag{8.23}$$

Theorem 8.3. *When the continuous-time system model is replaced with the augmented matrices in (7.108), then there exists decentralized PI controllers in their ideal form and a measurement combination $H \in \mathbb{R}^{n_u \times n_y}$ that gives a stable closed-loop system and minimizes the upper bound of the H_2 norm or the H_∞ norm, if there exists a positive definite matrix $P = P^T \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$, a matrix $Q \in \mathbb{R}^{n_u \times n_u}$, and diagonal matrices $X_1, X_2 \in \mathbb{R}^{n_u \times n_u}$ out of which X_1 ,and Q are non-singular. Together with either the stabilizing state feedback gain $K_{SF} \in \mathbb{R}^{2nu\times(nx+nu)}$ for the H_2 optimal control problem or the matrices $K_1, K_2 \in \mathbb{R}^{2n_u\times(n_x+n_u)}$ for the H_∞ optimal control problem, a SOF controller can be obtained by solving one of the following non-convex optimization problems:*

Continuous-time H_2 optimal:

$$\min_{\substack{K_{SF},X_1,X_2,\\ P,Q,H}} \quad trace(\bar{E}_c^T P \bar{E}_c) \tag{8.24}$$

subject to

$$P \succ 0 \tag{8.25}$$

$$X_1 = diag(x_{1_1} \cdots x_{1_{nu}}) \tag{8.26}$$

$$X_2 = diag(x_{2_1} \cdots x_{2_{nu}}) \tag{8.27}$$

$$M_2 + N_2 \prec 0 \tag{8.28}$$

Continuous-time H_∞ optimal:

$$\min_{\substack{K_1,K_2,X_1,X_2,\\ P,Q,H}} \quad \gamma^2 \tag{8.29}$$

subject to

$$P \succ 0 \tag{8.30}$$

$$X_1 = diag(x_{1_1} \cdots x_{1_{nu}}) \tag{8.31}$$

$$X_2 = diag(x_{2_1} \cdots x_{2_{nu}}) \tag{8.32}$$

$$M_\infty + N_\infty \prec 0 \tag{8.33}$$

The matrices M_2, N_2, M_∞, and N_∞ are given by (7.14) ,(7.18), (7.34), and (7.38), respectively, where Φ_1 and Φ_2 in N_2 and N_∞ have been replaced by the definition in (8.22).

Proof. The proofs that were given for Theorem 7.3 and Theorem 7.5 ensures that the closed-loop system with the resulting SOF controller K (given in (8.21)) is stable and minimizes the upper bound for the H_2 norm or the H_∞ norm. The parameters for the decentralized PI controllers and the measurement combination can then be obtained from $K_p = I$, $K_i = X_1^{-1} X_2$, and $\hat{H} = Q^{-1} H$, respectively. □

Alternatively, if the process models are expressed in discrete-time, then Theorem 7.7, and 7.9 can be modified to find a measurement combination $\hat{H}$, and discrete-time PI controllers in their ideal form that is given by an H_2, or an H_∞ optimal SOF controller that ensures stability for the closed-loop system.

Theorem 8.4. *When the discrete-time system model is replaced with the augmented matrices in (7.94), then there exist decentralized PI controllers in their ideal form together with a measurement combination $H \in \mathbb{R}^{n_u \times n_y}$ that gives a stable closed-loop system and minimizes the upper bound of the H_2 norm or the H_∞ norm, if there exists a stabilizing state feedback gain $K_{SF} \in \mathbb{R}^{2n_u \times (n_x + n_u)}$, a positive definite matrix $P = P^T \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$, matrices Z_1, $Z_2 \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)}$, a matrix $Q \in \mathbb{R}^{n_u \times n_u}$, and diagonal matrices $X_1, X_2 \in \mathbb{R}^{n_u \times n_u}$ of which X_1 and Q are non-singular that solves one of the following non-convex optimization problems:*

Discrete-time H_2 optimal:

$$\min_{\substack{K_{SF},Z_1,Z_2,\\ P,Q,,H}} \quad trace(\bar{E}_d^T P \bar{E}_d) \tag{8.34}$$

subject to

$$P \succ 0 \tag{8.35}$$

$$X_1 = diag(x_{1_1} \cdots x_{1_{nu}}) \tag{8.36}$$

$$X_2 = diag(x_{2_1} \cdots x_{2_{nu}}) \tag{8.37}$$

$$W_2 \prec 0 \tag{8.38}$$

Discrete-time H_∞ optimal:

$$\min_{\substack{K_{SF},Z_1,Z_2,\\ P,Q,,H}} \quad \gamma^2 \tag{8.39}$$

subject to

$$P \succ 0 \tag{8.40}$$

$$X_1 = diag(x_{1_1} \cdots x_{1_{nu}}) \tag{8.41}$$

$$X_2 = diag(x_{2_1} \cdots x_{2_{nu}}) \tag{8.42}$$

$$W_\infty + V_\infty \prec 0 \tag{8.43}$$

The matrices W_2, W_∞, and V_∞ are given by (7.55), (7.68), and (7.69) respectively, where Φ_1, and Φ_2 in W_2 and V_∞ have been replaced by the definition in (8.22).

Proof. The proofs that were given for Theorem 7.7 and Theorem 7.9 ensures that the closed-loop system with the resulting SOF controller K (given in (8.21)) is stable and minimizes the upper bound for the H_2 norm or the H_∞ norm. The parameters for the decentralized PI controllers and the measurement combination can then be obtained from $K_p = I$, $K_i = X_1^{-1}X_2$, and $\hat{H} = Q^{-1}H$, respectively. □

8.2 Controlling optimal measurement combinations

The optimization problems, given in Theorem 8.3 and Theorem 8.4 requires solving a BMI, which is NP-hard and, thus, finding the globally optimal solution is difficult. However, by specifying stable state feedback gains, the BMIs become LMIs, which turns them into convex optimization problems that can be solved using standard SDP solvers. An improved solution can be obtained by using an iterative algorithm to find a local optimum, following the procedure described in, e.g., Algorithm 4 for continuous-time systems.

Algorithm 4 H_2 or H_∞ optimal control in continuous-time.

```
Initialize: For $k = 0$ choose a stabilizing state feedback gain $K_{SF}$ for the $H_2$
    control problem, or the stabilizing gains $K_1$ and $K_2$ for the $H_\infty$ control problem.
1: Set $k \leftarrow k+1$ and for the fixed gain $K_{SF}$, or for the fixed gains $K_1$ and $K_2$ solve:
    Continuous-time $H_2$ optimal:
        $J_1^k = \min_{\Phi_1,\Phi_2,P} \operatorname{trace}(\bar{E}_c^T P \bar{E}_c)$
        subject to $P \succ 0$
            $M_2 + N_2 \prec 0$
            $\Phi_2 \in$ Controller constraints
            $\Phi_1, \Phi_2 \in$ SOC constraints
    Continuous-time $H_\infty$ optimal:
        $J_1^k = \min_{\Phi_1,\Phi_2,P} \gamma^2$
        subject to $P \succ 0$
            $M_\infty + N_\infty \prec 0$
            $\Phi_2 \in$ Controller constraints
            $\Phi_1, \Phi_2 \in$ SOC constraints
2: Fix $\Phi_1$ and $\Phi_2$ at the values obtained in step 1 and solve the resulting LMI:
    Continuous-time $H_2$ optimal:
        $J_2^k = \min_{K_{SF},P} \operatorname{trace}(\bar{E}_c^T P \bar{E}_c)$
        subject to $P \succ 0$
            $M_2 + N_2 \prec 0$
    Continuous-time $H_\infty$ optimal:
        $J_2^k = \min_{K_1,K_2,P} \gamma^2$
        subject to $P \succ 0$
            $M_\infty + N_\infty \prec 0$
3: If $J_2^k - J_1^k < \epsilon$ stop, else update $K_{SF}$, or $K_1$ and $K_2$, and repeat step 1 and 2.
```

Similarly, a local optimum for discrete-time systems can be found by using the iterative algorithm given in Algorithm 5 after it has been initialized with a stabilizing state feedback gain. Note that the decision variable Φ_2 is available in both step 1 and step 2 for discrete-time systems, whereas it is only available in step 1 when solving for continuous-time systems. These additional degrees of freedom is likely to be beneficial for Algorithm 5, since having more decision variables allows more flexibility when searching for the local optimum.

Algorithm 5 H_2 or H_∞ optimal control in discrete-time.

```
Initialize: For $k = 0$ choose a stabilizing state feedback gain $K_{SF}$.
1: Set $k \leftarrow k + 1$ and for the fixed state feedback gain $K_{SF}$ solve one of:
    Discrete-time $H_2$ optimal:
        $J_1^k = \min_{Z_1, Z_2, \Phi_1, \Phi_2, P} \operatorname{trace}(\bar{E}_d^T P \bar{E}_d)$
        subject to $P \succ 0$
            $W_2 \prec 0$
            $\Phi_1, \Phi_2 \in$ Controller constraints
            $\Phi_2 \in$ SOC constraints
    Discrete-time $H_\infty$ optimal:
        $J_1^k = \min_{Z_1, Z_2, \Phi_1, \Phi_2, P} \gamma^2$
        subject to $P \succ 0$
            $W_\infty + V_\infty \prec 0$
            $\Phi_1, \Phi_2 \in$ Controller constraints
            $\Phi_2 \in$ SOC constraints
2: Fix $\Phi_1$, $Z_2$, and $Z_2$ at the values obtained in step 1 and solve the LMI:
    Discrete-time $H_2$ optimal:
        $J_2^k = \min_{K_{SF}, \Phi_2, P} \operatorname{trace}(\bar{E}_d^T P \bar{E}_d)$
        subject to $P \succ 0$
            $W_2 \prec 0$
            $\Phi_2 \in$ Controller constraints
            $\Phi_2 \in$ SOC constraints
    Discrete-time $H_\infty$ optimal:
        $J_2^k = \min_{K_{SF}, \Phi_2, P} \gamma^2$
        subject to $P \succ 0$
            $W_\infty + V_\infty \prec 0$
            $\Phi_2 \in$ Controller constraints
            $\Phi_2 \in$ SOC constraints
3: If $J_2^k - J_1^k < \epsilon$ stop, else update $K_{SF}$ and repeat step 1 and 2.
```

The matrices M_2, N_2, M_∞, and N_∞ in Algorithm 4 are given by (7.14), (7.18), (7.34), and (7.38), respectively, whereas for Algorithm 5 the matrices W_2, W_∞, and V_∞ are described in (7.55), (7.68), and (7.69), respectively.

To have the resulting controllers consist of decentralized PI controllers, the process matrices in M_2, N_2, M_∞, and N_∞ should be replaced with one of the augmented models in (7.88), or (7.108). Similarly, the augmented matrices in (7.94), and (7.114) should replace the original state-space models in W_2, W_∞, and V_∞. Which set of augmented matrices to use, depends on the structure of the PI controllers, where (7.88), and (7.94) corresponds to PI controllers in their parallel form and (7.108), (7.114) coincides with PI controllers in their ideal form. Furthermore, the decision variables Φ_1, and Φ_2 should be replaced accordingly,

Parallel form

$$\Phi_1 := X, \quad \Phi_2 := \mathcal{I} H, \tag{8.44}$$

Ideal form

$$\Phi_1 := \begin{bmatrix} Q & 0 \\ 0 & X_1 \end{bmatrix}, \quad \Phi_2 := \begin{bmatrix} H & 0 \\ 0 & X_2 \end{bmatrix}, \tag{8.45}$$

where $X \in \mathbb{R}^{2n_u \times 2n_u}$, $X_1 \in \mathbb{R}^{n_u \times n_u}$, $X_2 \in \mathbb{R}^{n_u \times n_u}$, $Q \in \mathbb{R}^{n_u \times n_u}$, and $H \in \mathbb{R}^{n_u \times n_y}$ become the new decision variables for the optimization problem with $\mathcal{I}$ being defined in (7.83). To ensure that the resulting controllers correspond to decentralized PI controllers, the variables X, X_1, and X_2 have to be diagonal matrices, which should be included as the controller constraints in Algorithm 4 and Algorithm 5.

The measurement combination H has been separated from the other decision variables, which provides the possibility of directly incorporating self-optimizing control (SOC) properties on the resulting closed-loop system by, e.g., using the null space or the exact local method. If the null space method described in Section 3.1.1 is used, then the expression $HF = 0$ can simply be included as the SOC constraints. Unfortunately, using the exact local method is not as straightforward, since then the optimal measurement combination is restricted by the optimization problem given in (3.32), and (3.33). However, if there are multiple unconstrained MVs available ($n_u > 1$), the measurement combination can be multiplied with a non-singular matrix without increasing the steady-state loss. Therefore, if an optimal measurement combination H_{opt} has been obtained *a priori* by using, e.g., (3.37), then the measurement matrices C_c and D_c in (7.2) (or C_d and D_d in (7.43)) can be replaced with C_H and D_H that are defined as:

$$C_H = H_{opt}C_c, \quad D_H = H_{opt}D_c. \tag{8.46}$$

In this case, the size of the decision variable H in (8.44) and (8.45) goes from $H \in \mathbb{R}^{n_u \times n_y}$ to $H \in \mathbb{R}^{n_u \times n_u}$.

When Algorithm 4 or Algorithm 5 is initialized with stabilizing feedback gains such that step 1 gives a feasible solution for the first iteration, then they will generate a sequence of non-increasing solutions such that:

$$J_1^{k+1} \leq J_2^k \leq J_1^k, \qquad \forall k.$$

Thus, the solution will converge to a local minimum. If one of the described methods of augmenting the system models have been used, then the final controller parameters can be obtained from:

Parallel form

$$\begin{bmatrix} K_p & 0 \\ 0 & K_i \end{bmatrix} = X^{-1}, \tag{8.47}$$

Ideal form

$$\begin{bmatrix} K_p & 0 \\ 0 & K_i \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & X_1^{-1}X_2 \end{bmatrix}. \tag{8.48}$$

The final measurement combination $\hat{H}$ is given by

$$\hat{H} = H, \tag{8.49}$$

$$\hat{H} = Q^{-1}H \tag{8.50}$$

when using the null space method or alternatively,

$$\hat{H} = HH_{opt}, \tag{8.51}$$

$$\hat{H} = Q^{-1}HH_{opt} \tag{8.52}$$

when H_{opt} have been computed *a priori* using the exact local method.

Next, the effectiveness of the two proposed algorithms will be validated by application to two different distillation column models. The resulting optimization problems were solved in Matlab, using the YALMIP toolbox [90] together with the SDP solver MOSEK [99].

8.2.1 Case study: Binary distillation column

The proposed method was applied to the "column A" distillation column model [121], described in more detail in Section 4.4. The distillation column uses an LV configuration where the two available degrees of freedom are:

$$u = \begin{bmatrix} L_R & V_B \end{bmatrix}^T . \tag{8.53}$$

The objective is to minimize deviations in the top product composition x_D and the bottom product composition x_B, which has been set to 99% and 1%, respectively. These compositions will be controlled indirectly using the temperature measurements $T_i({}^oC)$ inside the column, where the different stages are denoted with i. The main disturbances considered are changes in feed flow rate (F_0), feed composition (x_F), and feed liquid fraction (q_F):

$$d = \begin{bmatrix} F_0 & x_F & q_F \end{bmatrix}^T . \tag{8.54}$$

The controlled variable $c_{ref,1}$ will be used as a reference for the simulations, where the distillate composition x_D and the bottom composition x_B are being controlled directly. The PI controllers presented in [121] are used for $c_{ref,1}$, which has been demonstrated to have good disturbance rejection properties.

Table 8.1: Controlled variables and PI parameters.

Controlled variables (CVs)	PI Parameters
$c_{ref,1} = \begin{bmatrix} x_D \\ -x_B \end{bmatrix}$	$k_p = 26.1, k_i = 6.94$ $k_p = 37.5, k_i = 11.3$
$c_{ref,2} = \begin{bmatrix} 0.037\ T_{12} - 0.645\ T_{30} - 0.657\ T_{31} \\ 1.250\ T_{12} - 0.205\ T_{30} - 0.154\ T_{31} \end{bmatrix}$	$k_p = 0.59, k_i = 0.07$ $k_p = 0.73, k_i = 0.09$
$c_{EL} = \begin{bmatrix} 0.267\ T_{12} - 0.511\ T_{30} - 0.510\ T_{31} \\ 1.000\ T_{12} - 0.302\ T_{30} - 0.264\ T_{31} \end{bmatrix}$	$k_p = 2.12, k_i = 0.46$ $k_p = 1.32, k_i = 0.29$
$c_{NS,1} = \begin{bmatrix} 0.348\ T_{11} - 0.248\ T_{12} + 0.594\ T_{30} - 0.811\ T_{31} \\ 1.000\ T_{11} - 0.755\ T_{12} - 0.146\ T_{30} + 0.182\ T_{31} \end{bmatrix}$	$k_p = 17.0, k_i = 1.66$ $k_p = 19.9, k_i = 2.81$
$c_{NS,2} = \begin{bmatrix} 0.080\ T_{11} - 0.007\ T_{12} - 0.032\ T_{14} + 0.301\ T_{30} - 0.406\ T_{31} \\ 0.987\ T_{11} - 1.000\ T_{12} + 0.172\ T_{14} - 0.209\ T_{30} + 0.250\ T_{31} \end{bmatrix}$	$k_p = 36.0, k_i = 3.89$ $k_p = 46.2, k_i = 10.1$
$c_{NS,3} = \begin{bmatrix} 0.472\ T_{11} - 0.465\ T_{12} + 0.078\ T_{14} - 0.089\ T_{30} + 0.121\ T_{31} - 0.396\ T_{37} \\ 0.783\ T_{11} - 0.655\ T_{12} + 0.044\ T_{14} + 0.602\ T_{30} - 0.872\ T_{31} + 1.000\ T_{37} \end{bmatrix}$	$k_p = 31.1, k_i = 10.8$ $k_p = 35.1, k_i = 6.17$

Exact local method

The selection of controlled variables using indirect control of the compositions for the distillation column example has previously been investigated in [146]. Based on the exact local method, a mixed-integer quadratic programming (MIQP) approach was used to select the best subsets of the available measurements $(T_1 \cdots T_{41})$. This resulted in the following controlled variable when using three measurements:

$$c_{ref,2} = H_{opt} \begin{bmatrix} T_{12} & T_{30} & T_{31} \end{bmatrix} , \tag{8.55}$$

where the optimal measurement combination was given by

$$H_{opt} = \begin{bmatrix} -0.0369 & 0.6449 & 0.6572 \\ -1.2500 & 0.2051 & 0.1537 \end{bmatrix} .$$

To illustrate the ease of controlling $c_{ref,2}$, the author of [146] implemented two decentralized PI controllers, that were tuned using the SIMC method [123]. However, it should be possible to find a different measurement combination $\hat{H} = Q^{-1}H_{opt}$ together with PI controllers that further improves the transient response, without affecting the steady-state loss.

Using Algorithm 4, PI controllers in their parallel form and a new CV were found with a measurement combination that satisfies $\hat{H} = Q^{-1}H_{opt}$. The resulting CV (denoted c_{EL}) from Algorithm 4 and the other referenced CVs can be seen in Table 8.1 together with their PI parameters.

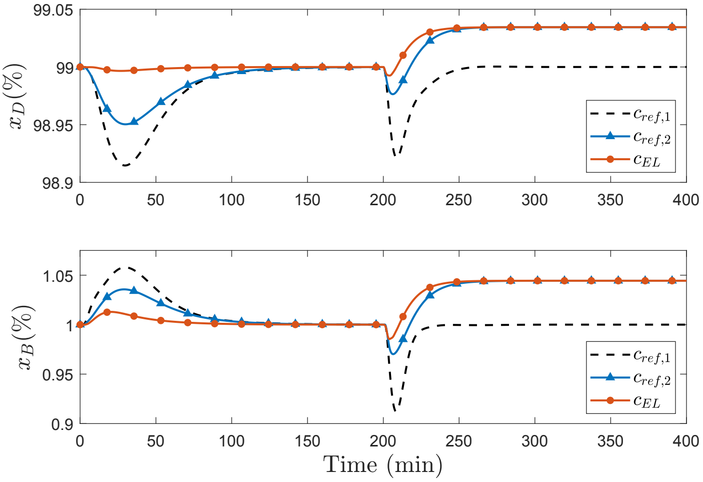


Figure 8.1: Deviations in x_D and x_B for a +10% filtered $(\frac{1}{25s+1})$ step change in F_0 after 1 min and a −20% filtered $(\frac{1}{10s+1})$ step change in q_F at 200 min.

Dynamic simulations were performed to demonstrate the improvements in the transient response when using the newly computed c_{EL} compared to the existing $c_{ref,2}$ from [146]. The disturbances were a +10% filtered $(\frac{1}{25s+1})$ step change in F at 1 min and a −20% filtered $(\frac{1}{10s+1})$ step change in q_F after 200 min. The simulation results can be seen in Figure 8.1, which shows a clear improvement in the transient behavior for the proposed c_{EL} compared to $c_{ref,2}$. Furthermore, both c_{EL} and $c_{ref,2}$ have the same steady-state loss since their respective x_D and x_B converge to the same values.

Null space method

The null space method requires that the number of independent measurements is greater or equal the sum of the number inputs and disturbances, i.e., $n_y \geq n_u + n_d$. Out of the three disturbances F_0, x_F and q_F, changes in feed flow F have no steady-state effect on the cost, and thus, four independent measurements are required to satisfy the null space condition. When more measurements are added, it has the potential to reduce the effect of measurement noise, which can thus, further decrease the steady-state loss [62]. This often comes at the expense of an increase in the complexity of the control structure and can make the controller design more challenging. However, it also has the potential to further improve the dynamic response when using one of the proposed algorithms, since extra measurements will be treated as additional degrees of freedom in the optimization problem.

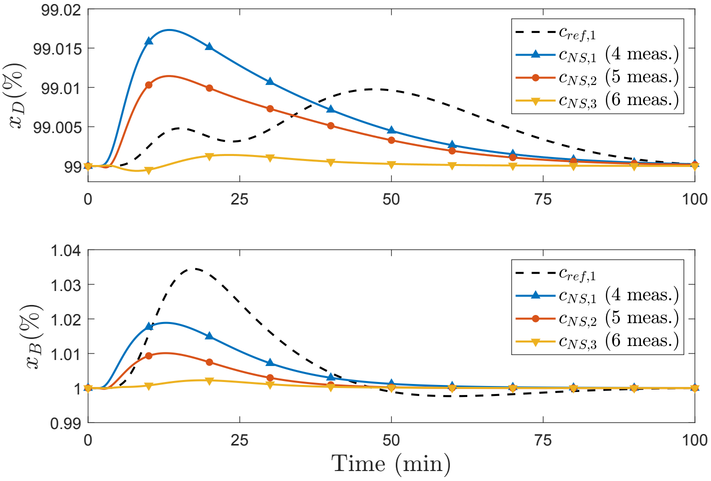


Figure 8.2: Deviations in x_D and x_B for a +20% filtered ($\frac{1}{15s+1}$) step change in x_F after 1 min.

To illustrate this, the following CVs are used,

$$\begin{aligned} c_{NS,1} =& H_1 \begin{bmatrix} T_{11} & T_{12} & T_{30} & T_{31} \end{bmatrix}^T, \\ c_{NS,2} =& H_2 \begin{bmatrix} T_{11} & T_{12} & T_{14} & T_{30} & T_{31} \end{bmatrix}^T, \\ c_{NS,3} =& H_3 \begin{bmatrix} T_{11} & T_{12} & T_{14} & T_{30} & T_{31} & T_{37} \end{bmatrix}^T, \end{aligned}$$

where the measurements $\{T_{11}\}$, $\{T_{11}, T_{14}\}$, and $\{T_{11}, T_{14}, T_{37}\}$ have been added to the set given in (8.55) for $c_{NS,1}$, $c_{NS,2}$, and $c_{NS,3}$, respectively. Thus, the goal is to determine the measurement combinations H_1, H_2, and H_3, together with their controller the parameters. The resulting CVs ($c_{NS,1}$, $c_{NS,2}$, and $c_{NS,3}$) are computed by solving the H_2 optimal control problem for Algorithm 4 when using PI controllers in their parallel form. The resulting CVs and their controller parameters are shown in Table 8.1.

As already mentioned, adding measurements to the optimization problems has the potential to improve the dynamic response, since extra measurements are treated as additional degrees of freedom in the optimization problem. This is confirmed by the simulation shown in Figure 8.2 and 8.3, where the CVs with more measurements are able to achieve better disturbance rejection.

More measurements can thus, easily be implemented using the proposed method to further improve the dynamic performance. However, the increased risks of sensor failures and the cost of the additional measurements have to be weighed against the improved dynamic behavior.

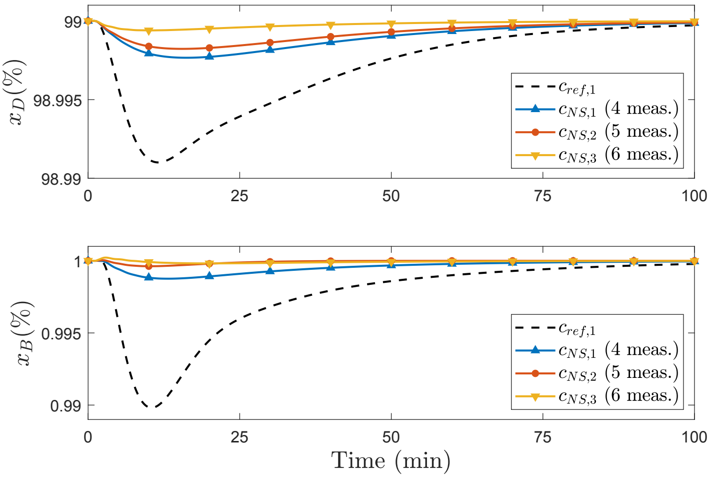


Figure 8.3: Deviations in x_D and x_B for a -5% filtered $(\frac{1}{25s+1})$ step change in q_F after 1 min.

8.2.2 Case study: Kaibel distillation column

A Kaibel distillation column [71] is a thermally coupled distillation column that can separate four products using a single condenser and a single reboiler (see Figure 8.4). To achieve the same four product streams from a single feed stream when using conventional binary columns, it would require a setup consisting of three different binary columns. A Kaibel distillation column is an extension to the Petlyuk distillation column [110] and has together with other diving wall columns, to a great extent been studied in the literature [138]. In comparison to the traditional configurations, the Kaibel column has the potential to give up to 40% reduction in energy consumption, as well as significantly reducing the capital investment cost and the physical space required in the process plant. However, the energy savings are only achieved if the distillation column operates close to its optimal value, which remains a challenge as it is a highly interactive multivariable system that is difficult to control.

The Kaibel column is dived into seven sections, and each section consists of several stages. For the simulated model, there are a total of 64 stages, as shown in Figure 8.4, where the stages are numbered with the prefractionator section first. The ternary feed is located between stage 12 and 13 and consists of three components with the mole fractions z_D, z_{S_1}, and z_{S_2}. Four product streams are drawn off, where the light component z_D dominates the distillate stream (D_F), component z_{S_1} and z_{S_2} dominate in the side-streams $(S_1$, and $S_2)$ and the remaining heavy components z_B dominates the bottom stream (B_F). For a more detailed description of the process model and its nominal values, the reader is referred to [80].

The distillate flow rate (D_F) and bottom flow rate (B_F) are used to stabilize the levels in the condenser and the reboiler, respectively. Furthermore, vapor boilup V_B and the vapor split R_V will be kept constant as it has been shown to be difficult to control in practice. Instead, they will be treated as disturbances. Therefore, the

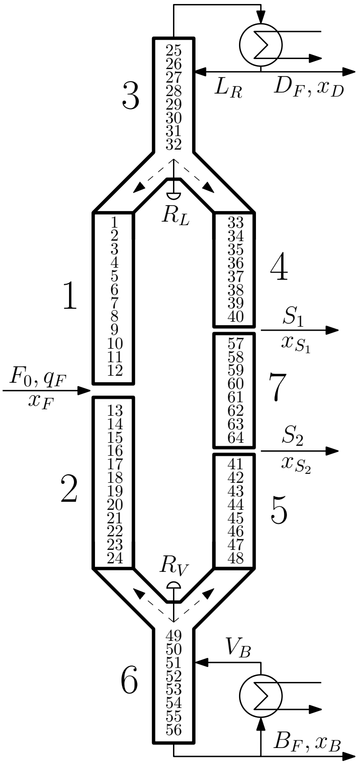


Figure 8.4: Kaibel distillation column, adopted from [129].

remaining degrees of freedom u and disturbances d are:

$$u = \begin{bmatrix} R_L & L_R & S_1 & S_2 \end{bmatrix}^T \tag{8.56}$$

$$d = \begin{bmatrix} V_B & R_V & F_0 & z_D & z_{S_1} & z_{S_2} & q_F \end{bmatrix}^T. \tag{8.57}$$

For the Kaibel column, the objective is to optimize the product distribution for given feed rate F_0 and boilup rate V_B. Assuming equal value of the products and that only the main components in each product stream are of value, then the objective is equivalent to minimizing the sum of the impurity flows. The cost function J can then be written as [129]:

$$J = D_F(1 - x_D) + S_1(1 - x_{S_1}) + S_2(1 - x_{S_2}) + B_F(1 - x_B). \tag{8.58}$$

Control structure design

The control structure design for the Kaibel column model has previously been studied in [80], [129], and [81]. Most of the work has mainly been focused on the steady-state operation of the column; however, different control structures was presented in [81].

Using the temperatures at stages 17, 30, 49, and 59 as the available measurements,

$$y = \begin{bmatrix} T_{17} & T_{30} & T_{49} & T_{59} \end{bmatrix}^T, \tag{8.59}$$

decentralized PI control was designed for the Kaibel column in [81], where the PI controllers were tuned using the SIMC tuning rules [123]. Controlling the temperatures in (8.59) had been shown to possess good self-optimizing control properties based on the singular value method [56]. However, due to the high level of process interactions, it was difficult to achieve good control performance using only decentralized PI controllers. Therefore, the authors of [81] proposed using an MPC to better counteract the interactions and obtain less total impurity flow. These two control structures will be used for comparison in the dynamic simulation and are denoted MPC and PI.

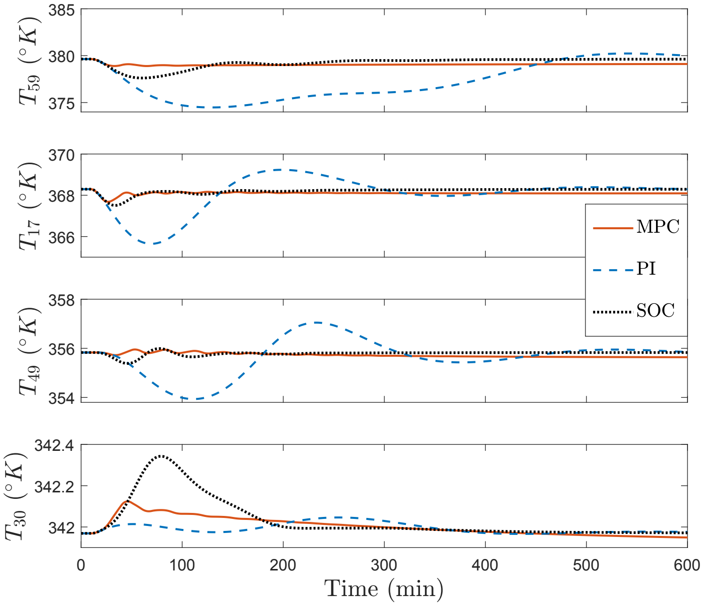


Figure 8.5: Temperature changes in the Kaibel column for a step disturbance of +10% in F_0.

The decentralized PI controllers in [81] are controlling each of the measurements in (8.59) individually. However, since the number of control inputs is equal to the number of measurements ($n_u = n_y$), it is possible to choose a controlled variable $c = \hat{H}y$ with any measurement combination $\hat{H}$ (assuming $\hat{H}$ is invertible) and still get the same the steady-state loss. This can be demonstrated by selecting $Q = \hat{H}$, which according to (3.34), gives $H = Q^{-1}\hat{H} = I$. Therefore, for the same measurements given in (8.59), H_∞ optimal PI controllers in their ideal form together with a new measurement combination was computed using Algorithm 5.

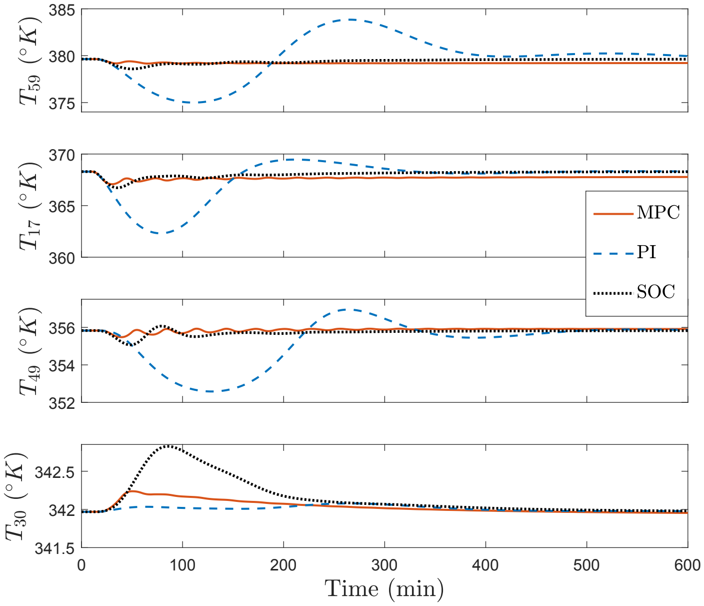


Figure 8.6: Temperature changes in the Kaibel column for a step disturbance of +20% in z_{S_1}.

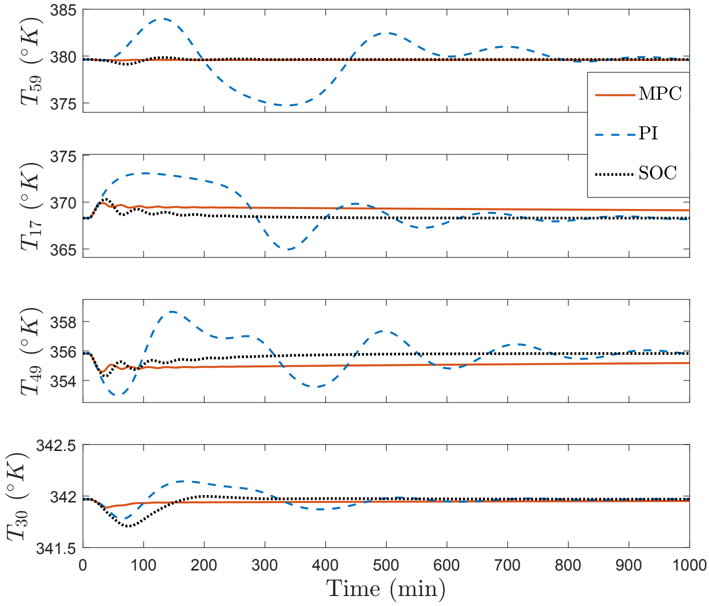


Figure 8.7: Temperature changes in the Kaibel column for a step disturbance of +10% in R_V.

This resulted in the following PI parameters,

$$K_p = I, \quad K_i = \begin{bmatrix} 0.0074 & 0 & 0 & 0 \\ 0 & 0.0149 & 0 & 0 \\ 0 & 0 & 0.0073 & 0 \\ 0 & 0 & 0 & 0.0088 \end{bmatrix}, \tag{8.60}$$

and the measurement combination given by

$$\hat{H} = \begin{bmatrix} 0.0209 & -0.0013 & -0.0002 & -0.0096 \\ 0.0171 & 0.0292 & -0.0009 & 0.0132 \\ -0.0237 & 0.0274 & 0.0019 & -0.0247 \\ 0.0230 & 0.0052 & -0.0270 & 0.0224 \end{bmatrix}. \tag{8.61}$$

The proposed control structure is denoted SOC and should give a better dynamic performance compared to the decentralized PI controllers in [81].

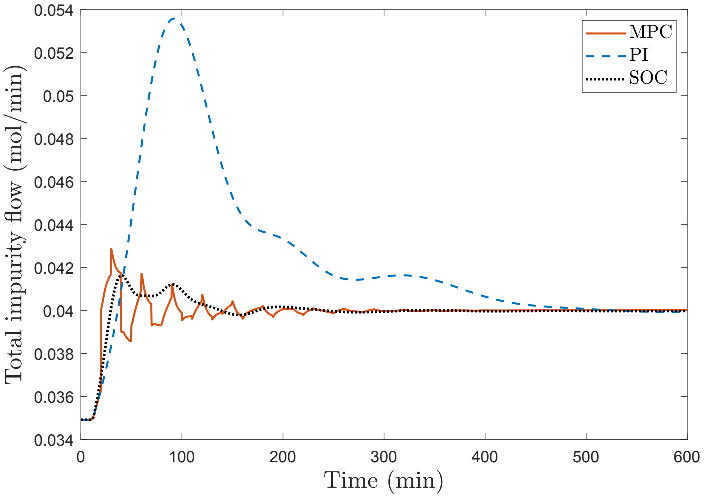


Figure 8.8: Impurity flow (8.58) for a step disturbance of +10% in F_0.

Dynamic simulation

Dynamic simulations were performed on the nonlinear model of the Kailbel distillation column. In the Figures 8.5, 8.6, and 8.7, changes in the controlled temperatures T_{17}, T_{30}, T_{49}, and T_{59} are simulated for step changes in the disturbances F_0, z_{S_1}, and R_V. Compared to the decentralized PI controllers, there is a significant improvement in the transient response when using the proposed measurement combination SOC. Furthermore, in the resulting impurity flows shown in the Figures 8.8, 8.9, and 8.10, the proposed control structure seems to be on par with the MPC.

These results are particularly interesting, as it would suggest that by properly selecting controlled variables (measurement combinations) together with well-tuned PI controllers, it might be possible to compete with a control structure consisting of an MPC combined with real-time optimization.

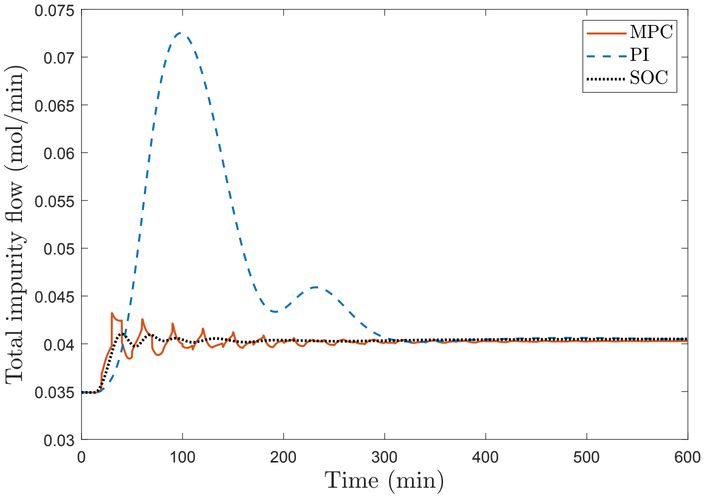


Figure 8.9: Impurity flow (8.58) for a step disturbance of +20% in z_{S_1}.

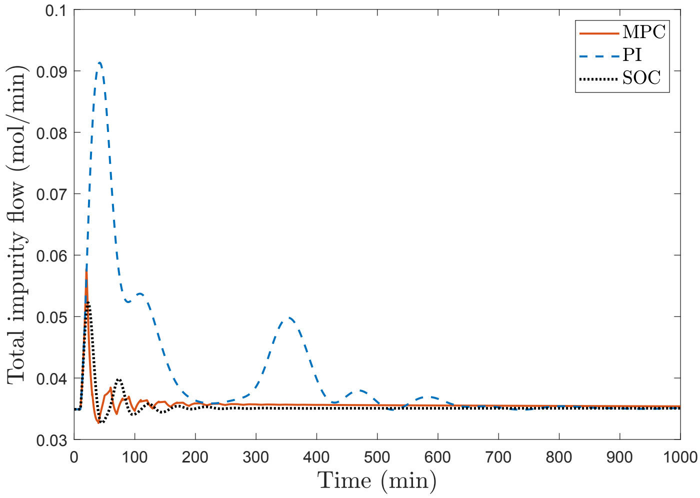


Figure 8.10: Impurity flow (8.58) for a step disturbance of +10% in R_V.

8.3 Sparse measurement selection

The previous section introduced Algorithm 4, and Algorithm 5 that aimed to find the H_2, or the H_∞ optimal PI controllers together with the measurement combination, H by iteratively solving two LMIs. In these two algorithms, the measurement combination H was be computed using all the available measurements that have been included in the problem formulation. It was also demonstrated by the case study in Section 8.2.1 that including more measurements, will improve the dynamic behavior of for the closed-loop system, at least when using the null space method.

Thus, from a theoretical perspective, using all available measurements would be optimal as it would provide the optimization problem with the largest number of decision variable when it searches for the optimum. However, in process plants, there can be hundreds and sometimes over thousands of measurements available. Including them all in the measurement combination would clearly be impractical, as it would lead to unnecessary complex control structures and increase the risk of getting control complications caused by sensor failures. Instead, the goal would be to find a smaller measurement subset that when used together with the obtained matrix H and feedback controllers will give a dynamic performance comparable to using all measurements. Therefore, the problem of finding the optimal measurement subset should ideally, also be included when designing the closed-loop system.

In Section 4.3, a convex relaxation approach for finding measurement subsets was proposed to deal with the combinatorial nature of the optimization problem. This method included the column-wise sparsity promoting re-weighted l_1 norm,

$$W_j^{k+1} = \frac{1}{\sum_i |H_{i,j}^k| + \kappa}, \tag{8.62}$$

where the weights are being updated iteratively with κ being a small positive value to ensure the update is well-defined. These weights, together with the penalty term λ, can then be included in the objective functions for Algorithm 4, and Algorithm 5 to penalize the number of measurements used in the resulting control structure.

Algorithm 6 shows an example of how the sparsity promoting function can be incorporated when computing H_∞ optimal PI controllers, together with measurement combinations that satisfy the null space condition ($HF = 0$). Here, it is assumed that a continuous-time model is being been used that has been augmented as described in (7.88), which will result in the PI controllers given in (8.47). However, a similar procedure can be used for both types of PI controllers, that uses either continuous-time or discrete-time model with one of the H_2 norm, or the H_∞ norm as the performance measure for the closed-loop system.

By varying the value λ, there will be a trade-off between the H_∞ performance for the resulting closed-loop system and the number of measurements used. Additional weights may also be included to penalize certain control structures, e.g., if some measurement links are unattractive due to high implementation cost. When the re-weighted l_1 norm has been included in the optimization problem, then the convergence is unknown. However, numerical experiments (e.g., [41], and [7]) indicate that it tends to converge to a local minimum.

Algorithm 6 H_∞ optimal PI controller design and sparse measurement selection for the null space method.

```
Initialize: For $k = 0$ choose stabilizing state feedback gains $K_1$ and $K_2$.
1: Set $k \leftarrow k + 1$ and for the fixed state feedback gains $K_1^k$ and $K_2^k$ solve:
       $J_1^k := \min_{P,X,H} \gamma^2$    (8.63)
       subject to $HF = 0$, (8.9), (8.10), and (8.11)
2: Fix $X^k$, and $H^k$ at the values obtained in step 1 and solve the LMI:
       $J_2^k := \min_{K_1,K_2,P} \gamma^2$    (8.64)
       subject to (8.9), (8.10), and (8.11)
3: If $J_2^k - J_1^k < \epsilon_1$ proceed to step 4, else repeat step 1 and 2.
4: Set $k = 0$, and compute $W^{k+1}$ from (8.62) by using the last $H^k$ that was
   calculated from the previous three steps when using all available measurements.
5: For $k \leftarrow k + 1$, fix $K_1$, and $K_2$ at the values obtained in step 2 and solve:
       $H^k = \arg\min_{P,X,H} \gamma^2 + \lambda \sum_{i,j} W_j^k |H_{i,j}^k|$    (8.65)
       subject to $HF = 0$, (8.9), (8.10), and (8.11)
6: If $\|H^{k-1} - H^k\| < \epsilon_2$ go to step 7, else use (8.62) to update $W^{k+1}$ and repeat
   step 5.
7: Remove the measurements that correspond to the zero columns in $H^k$ and for
   the reduced measurement subset repeat step 1−3.
```

To evaluate Algorithm 6, a process model consisting of a Petlyuk distillation column was used as a case study. Here, the aim was to find an optimal subset of measurement that satisfies the null space condition, together with decentralized PI controllers. The optimization problems have been formulated by the YALMIP toolbox [90] in Matlab and solved using the solver MOSEK [99].

8.3.1 Petlyuk distillation column

The Petlyuk distillation column, often referred to as the divided wall column, offers an appealing alternative for separating ternary mixtures. In comparison to the traditional configuration, where two columns are used in series, the Petlyuk column is capable of saving up to 30% in both capital and energy costs according to [134]. However, despite the potential benefits of Petlyuk columns, only a few implementations exist in the industry due to their operational challenges.

The Petlyuk distillation column consists of six sections as illustrated in Figure 8.11, with the model information seen in Table 8.2. Each section is arranged in the same column shell with eight stages for each section. The ternary feed is located between section 1 and 2, where its composition x_F consists of the components A, B, and C with the mole fractions z_A, z_B, and z_C. Three product streams are drawn off in the sections 3–6, where the light component A dominates the distillate stream (D_F), component B_F dominates in the side-stream (S_F) and the heavy component C dominates the bottom stream (B_F). For a more detailed description of the model, the reader is referred to [4].

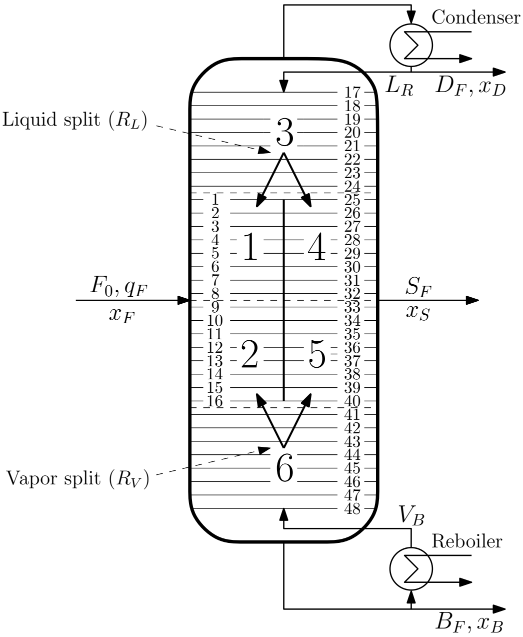


Figure 8.11: Petlyuk distillation column, adopted from [4].

The distillate flow rate (D_F) and bottom flow rate (B_F) are used to stabilize the levels in the condenser and the reboiler, respectively. Furthermore, the vapor split R_V will be kept constant as it is difficult to use in practice, and [55] found that good self-optimizing control can still be achieved when R_V is fixed.

Therefore, the remaining available degrees of freedom are,

$$u = \begin{bmatrix} L_R & V_B & S_F & R_L \end{bmatrix}^T \tag{8.66}$$

corresponding to the reflux, boilup, side-stream flow and liquid split, respectively.

For the Petlyuk column, three product specifications should be kept during operation; distillate purity ($x_{A,D}$), bottom purity ($x_{C,B}$) and side-stream purity ($x_{B,S}$), where $x_{i,j}$ denotes the mole fraction of component i in stream j. The operational objective is to minimize energy cost (V_B) while maintaining the product purity specifications.

Table 8.2: Data for the Petlyuk column

Column data	
Relative volatilities	$\alpha_R = \begin{bmatrix} 9 & 3 & 1 \end{bmatrix}^T$
Liquid time constant	$\tau_L = 0.063$ min
Holdup top and bottom	$M_B = M_D = 20M_i$
Holdup stages	$M_i = 1$ kmol
Boiling points A,B,C	$T_B = \begin{bmatrix} 299.3 & 342.15 & 399.3 \end{bmatrix}^T$ K
Antoine's parameters	$\begin{bmatrix} 2.86 & -1143 & -0.349 \end{bmatrix}$
Feed	
Flow	$F_0 = 1$ kmol/min
Composition	$z_A = z_B = z_C = 0.33$
Liquid fraction	$q_F = 0.477$
Product specifications	$x_{A,D} = x_{B,S} = x_{C,D} = 97\% \pm 1\%$
Measurement delays	
Compositions	5 min
Temperatures	1 min

Previous control structures

The control structure design for the Petlyuk column model has previously been studied by [4], [55] and [152]. Most of the work has mainly been focused on the steady-state operation of the column. However, decentralized PI controllers were designed by [4] for CVs consisting of three composition measurements and a linear combination of temperatures obtained using the null space method. The Petlyuk column was also investigated by [152]. For the same CVs, the authors proposed using sparse PI controllers (with partial interactions between the controllers), that further improved the transient performance. These control structures will be used for comparison in the dynamic simulations and are denoted $c_{ref,D}$, and $c_{ref,S}$, with the subscripts, D and S representing the decentralized and sparse control structures, respectively.

Indirect Control

Indirect control is when the primary variables are kept close to their desired value by controlling secondary variables at constant set-points [62]. The proposed control structures in $c_{ref,D}$ and $c_{ref,S}$, both use the measurements of the three product compositions to keep the products at their targeted values. While this ensures that the purity specifications are kept, the long delays associated with the composition measurements imposes limitations on the closed-loop performance and makes fast control difficult. Therefore, an indirect control structure is proposed, that uses only the temperature measurements.

Perfect indirect control can be achieved using the null space method assuming, $n_y \geq n_d + n_u$. In this example, only changes in z_A, z_B, and q_F will be considered, and thus, at least seven independent measurements are required. Using Algorithm 6, two control structures were computed, using 48 (all), and 7 temperature mea-

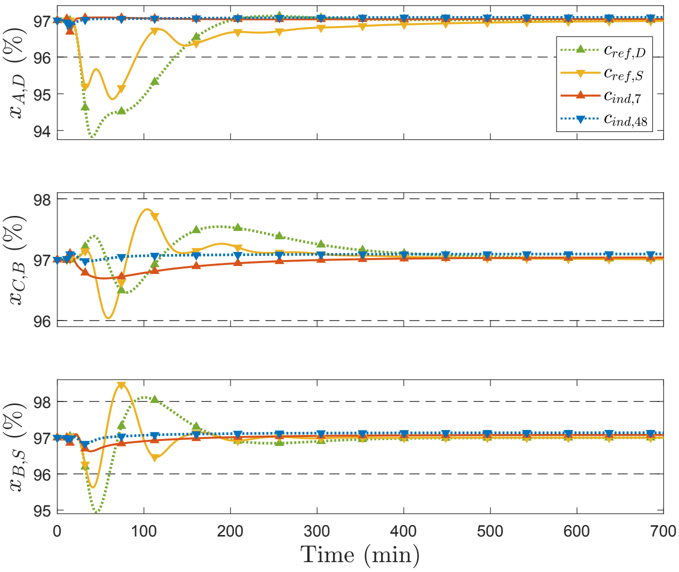


Figure 8.12: Deviations in $x_{A,D}$, $x_{C,B}$, and $x_{B,S}$ in the Petlyuk column for a step disturbance of -10% in z_A.

surements. These are denoted $c_{ind,48}$, and $c_{ind,7}$, where $c_{ind,7}$ resulted in

$$c_{ind,7} = \begin{bmatrix} -0.191T_4 + 0.044T_9 + T_{19} - 0.004T_{24} - 0.433T_{38} + 0.297T_{41} + 0.066T_{47} \\ 0.267T_4 - 0.069T_9 - T_{19} - 0.078T_{24} + 0.571T_{38} - 0.329T_{41} - 0.306T_{47} \\ -0.535T_4 + 0.189T_9 + T_{19} + 0.262T_{24} - 0.694T_{38} + 0.443T_{41} + 0.064T_{47} \\ 0.217T_4 + 0.009T_9 - 0.869T_{19} - 0.187T_{24} + T_{38} - 0.631T_{41} - 0.510T_{47} \end{bmatrix},$$

with T_i being the temperature at stage i. The PI controllers for $c_{ind,7}$ are given by

$$K_p = \begin{bmatrix} 0.4745 & 0 & 0 & 0 \\ 0 & 0.1908 & 0 & 0 \\ 0 & 0 & 0.1763 & 0 \\ 0 & 0 & 0 & 0.0495 \end{bmatrix}, K_i = \begin{bmatrix} 0.0035 & 0 & 0 & 0 \\ 0 & 0.1563 & 0 & 0 \\ 0 & 0 & 0.0012 & 0 \\ 0 & 0 & 0 & 0.0007 \end{bmatrix}.$$

for the proportional gain and the integral gain, respectively.

Dynamic simulation

Dynamic simulations were performed on the nonlinear model of the Petlyuk distillation column. In the Figures 8.12, 8.13, and 8.14, the proposed $c_{ind,48}$, and $c_{ind,7}$ are compared to $c_{ref,D}$, and $c_{ref,S}$ for step disturbances in z_A, z_B, and q_F. Since the sensitivity matrix F is computed using a local linear model, it causes some steady-state loss for $c_{ind,48}$ and $c_{ind,7}$. However, there is a significant improvement in the transient behavior compared to $c_{ref,D}$ and $c_{ref,S}$. The steady-state loss can also easily be removed by, e.g., including an outer loop that adjusts the set-points, using a slow integral action from the composition measurements. As expected, using all 48 temperature measurements gives a better response compared to using 7, since Algorithm 6 gives a trade-off between the dynamic performance (H_∞ norm) and the number of measurements used.

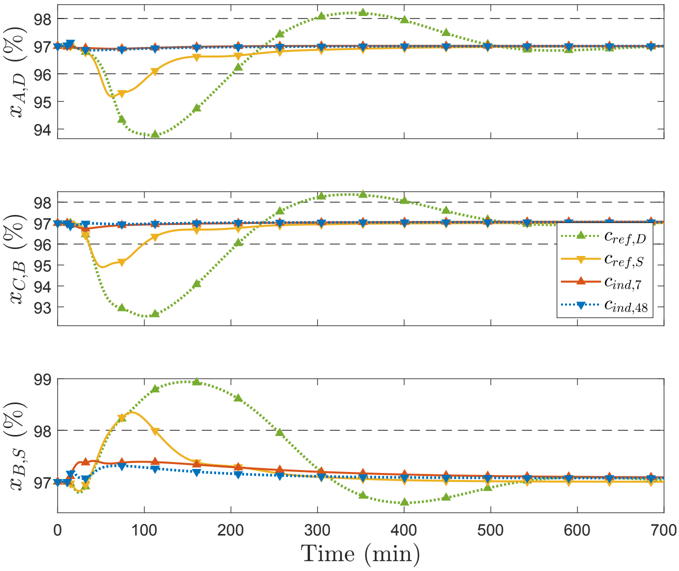


Figure 8.13: Deviations in $x_{A,D}$, $x_{C,B}$, and $x_{B,S}$ in the Petlyuk column for a step disturbance of +10% in z_B.

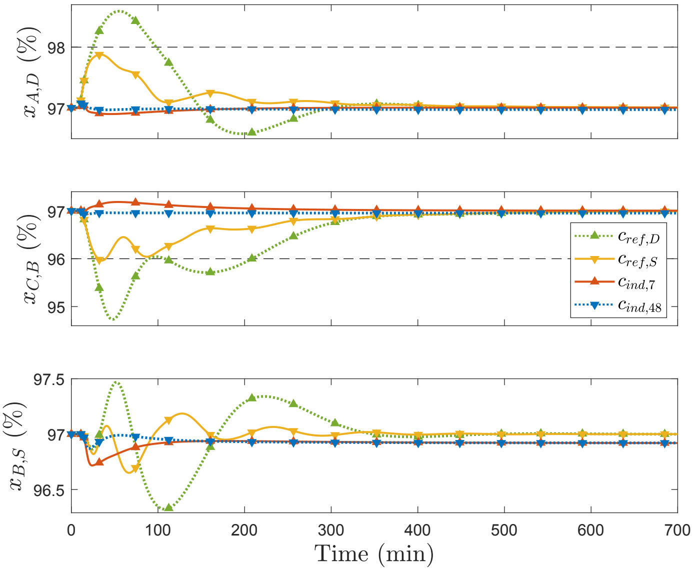


Figure 8.14: Deviations in $x_{A,D}$, $x_{C,B}$, and $x_{B,S}$ in the Petlyuk column for a step disturbance of +10% in q_F.

8.4 Trade-off between dynamic and steady-state performance

The optimal measurement combinations H for the null space method is non-unique since there is an infinite number of solutions that satisfies the condition $HF = 0$. This flexibility makes it possible to find a CV that improves the dynamic behavior of the closed-loop system while still achieving the optimal economic performance at steady-state. However, the solutions to the null space method are only optimal under the assumption that there is no measurement noise, which is unrealistic in practice. Therefore, it is often preferred to use the exact local method instead when computing the self-optimizing control variables, since it accounts for both the disturbances and the measurement errors. Unfortunately, the exact local method doesn't offer the same flexibility as the null space method when selecting the optimal measurement combinations. Instead, the optimal self-optimizing control variable is found from solving the optimization problem given in (3.32)−(3.33):

$$\min_{H} \frac{1}{2}\left\| HY \right\|_F^2 \tag{8.67}$$

$$\text{subject to } \; HG^y = J_{uu}^{1/2} \tag{8.68}$$

If an optimal measurement combination that has been obtained using the exact local method is multiplied with any non-singular matrix Q, it will result in a different measurement combination $\hat{H}$ such that,

$$\hat{H} = Q^{-1}H, \tag{8.69}$$

where $\hat{H}$ will give the same steady-state loss as the original measurement combination H. Therefore, the matrix Q can be used to improve the controllability of the self-optimizing control variable and hence give a better dynamic performance for the resulting closed-loop system, which was previously demonstrated in Section 8.2.1 and 8.2.2. However, the optimal self-optimizing control variable is only non-unique if the number of unconstrained MVs available is greater than one. Furthermore, while Algorithm 4 and 5 can be used to maintain the optimal steady-state solution, it may be beneficial to choose a measurement combination with a larger steady-state loss if it would provide a significant improvement in the closed-loop performance. That is, if the dynamic improvements resulted in better disturbance rejection, it could allow for a reduction in the "back off" applied to the active constraints, and as a consequence further increase the profitability.

Therefore, in this section, an iterative LMI algorithm is proposed that solves a Pareto optimization problem that gives a trade-off between minimizing the steady-state loss and the dynamic performance. Here, the augmented models in (7.108), or (7.114) are used to get PI controllers in their ideal form since both Q, and H are included as decision variables for the resulting SOF controller in (8.45). Thus, it becomes possible to compute them simultaneously, where the choice of H will impact both the dynamic and steady-state performance, whereas Q can be used to improve the level of interactions between the CVs.

In Algorithm 7, an iterative procedure is described for finding PI controllers and a measurement combination $\hat{H}$ that makes a trade-off between the steady-state loss and the dynamic behavior. Here, the H_∞ norm is used as the dynamic performance measurement for the closed-loop system with the process models being given in discrete-time. However, it is also possible to modify the algorithm such that it uses continuous-time models or such that the H_2 norm is minimized instead. In addition, similar to Algorithm 6, the sparsity promoting weighted l_1 norm has also been included in the objective, and thus, it attempts to solve a multi-objective optimization problem that includes the steady-state loss, the transient response, and the number of measurements used.

Algorithm 7 Trade-off between steady-state loss, dynamic performance, and the number of measurement used.

```
Initialize: For $k = 0$ choose a stabilizing state feedback gain $K_{SF}$ and obtain $H$
    using, e.g., (3.36) for all available measurements.
1: Set $k \leftarrow k + 1$ and compute $W^k$ using (8.62).
2: For the fixed $K_{SF}$, and $W^k$ solve the LMI:

   $$J_1^k = \min_{\substack{Z_1,Z_2,P\\X_1,X_2,Q,H}} \gamma^2 + \alpha\frac{1}{2}\left\|HY\right\|_F^2 + \lambda\sum_{i,j} W_j^k \left\|H_{i,j}\right\|_1 \qquad (8.70)$$
   $$\text{subject to } \; HG^y = J_{uu}^{1/2}, (8.40), (8.41), (8.42), \text{and } (8.43)$$

3: Update $W^k$ using (8.62) with the $H$ obtained from step 2.
4: Fix $Q$, $X_1$, $Z_1$, and $Z_2$ at the values obtained in step 2 and solve the LMI:

   $$J_2^k = \min_{\substack{K_{SF},P\\X_2,H}} \gamma^2 + \alpha\frac{1}{2}\left\|HY\right\|_F^2 + \lambda\sum_{i,j} W_j^k \left\|H_{i,j}\right\|_1 \qquad (8.71)$$
   $$\text{subject to } \; HG^y = J_{uu}^{1/2}, (8.40), (8.42), \text{and } (8.43)$$

5: If $\|H^{k-1} - H^k\|_2 < \epsilon_1$, and $J_1^k - J_2^k < \epsilon_2$ go to step 6, else update $K_{SF}$ and
   repeat step 1 to 5.
6: Remove the measurements that correspond to the zero columns in $H$.
```

Varying the values of the scalars α and λ will give different trade-offs between the H_∞ performance, the steady-state loss, and the number of measurements used. The final controller parameters can be obtained from $K_p = I$, and $K_i = X_1^{-1}X_2$, whereas measurement combination will be given by $\hat{H} = Q^{-1}H$. The convergence of Algorithm 7 follows the convergence properties of previous algorithms. When $\lambda = 0$. the convergence is monotonically decreasing:

$$J_1^{k+1} \leq J_2^k \leq J_1^1, \qquad \forall k.$$

If the re-weighted l_1 norm is incorporated (λ is non-zero), then the convergence is unknown, but the numerical experiments given in, e.g., [41], and [7] indicate that it tends to converge to a local minimum.

The algorithm was evaluated, on the binary distillation column that has previously been used as the case study in Section 8.2.1. However, instead of computing controllers and measurement combinations for a given set of measurements, here, the goal is to design different control structures that make a trade-off between the steady-state loss, the transient behavior, and the number of measurements used. These optimization problems were solved in Matlab with the solver MOSEK [99] after they been formulated with the YALMIP toolbox [90].

8.4.1 Case study: Binary distillation column

The binary distillation column model [121] will be used to evaluate Algorithm 7, which was also used as a case study in Section 8.2.1. The distillation column has two unconstrained MVs available that consist of the liquid recycle flow rate L_R and the vapor boilup rate V_B:

$$u = \begin{bmatrix} L_R & V_B \end{bmatrix}^T \tag{8.72}$$

The objective is to minimize deviations in the top product composition x_D and the bottom product composition x_B by controlling the temperature measurements $T_i({}^oC)$ inside the column, where the different stages are denoted with i. The main disturbances considered are changes in feed flow rate (F_0), feed composition (x_F) and feed liquid fraction (q_F):

$$d = \begin{bmatrix} F_0 & x_F & q_F \end{bmatrix}^T . \tag{8.73}$$

Table 8.3: Controlled variables for the binary distillation column.

Controlled variables (CVs)
$c_{A,4} = \begin{bmatrix} -0.970T_{11} - 0.923T_{12} - 0.200T_{30} - 0.297T_{31} \\ -1.093T_{11} - 1.040T_{12} - 0.227T_{30} - 0.336T_{31} \end{bmatrix}$
$c_{B,4} = \begin{bmatrix} -0.479T_{11} - 0.479T_{12} + 1.110T_{32} + 0.852T_{33} \\ -0.782T_{11} - 0.801T_{12} - 1.265T_{32} - 1.147T_{33} \end{bmatrix}$
$c_{A,7} = \begin{bmatrix} -0.118T_{11} - 0.128T_{12} - 0.123T_{13} - 0.015T_{21} + 0.125T_{28} + 0.139T_{29} + 0.126T_{30} \\ -0.204T_{11} - 0.220T_{12} - 0.209T_{13} + 0.090T_{21} + 0.062T_{28} + 0.054T_{29} + 0.030T_{30} \end{bmatrix}$
$c_{B,7} = \begin{bmatrix} -0.272T_{11} - 0.276T_{12} - 0.267T_{13} + 0.251T_{25} + 0.645T_{32} + 0.530T_{33} + 0.416T_{34} \\ -0.536T_{11} - 0.528T_{12} - 0.505T_{13} + 0.244T_{25} - 1.055T_{32} - 0.971T_{33} - 0.784T_{34} \end{bmatrix}$

Dynamic and steady-state performance

First, PI controllers and the matrix Q are computed for the optimal measurement combination when using 4 measurements that have been listed in Table 4.1. Here, Algorithm 5 is used to optimize the dynamic performance of the closed-loop system without changing the steady-state loss. The resulting CV is denoted $c_{A,4}$, and can be seen in Table 8.3 with its controller parameters given in Table 8.4. However, it should be possible to find a different control structure that still uses 4 measurements but gives an improvement in the transient response at the expense of the steady-state loss. Therefore, Algorithm 7 is used to find another CV with 4 measurements, where both the steady-state and dynamic performance have been included in the optimization problem. The resulting CV should thus, improve the controllability

Table 8.4: Control structures and their dynamic and steady-state performance for the binary distillation column.

CV	$\frac{1}{2}\|HY\|_F^2$	MV	k_p	k_i	$\|T_{w,z}\|_\infty$
$c_{A,4}$	0.344	L_T V_B	1.0 1.0	0.164 0.173	0.144
$c_{B,4}$	0.383	L_T V_B	1.0 1.0	0.601 0.736	0.070
$c_{A,7}$	0.222	L_T V_B	1.0 1.0	0.170 0.206	0.341
$c_{B,7}$	0.271	L_T V_B	1.0 1.0	0.624 0.731	0.066

properties at the expense of an increase in the steady-state loss. Similarly, the CVs $c_{A,7}$ and $c_{B,7}$ are computed when using 7 measurements. The CV $c_{A,7}$ puts more emphasis on the steady-state loss (large α), while $c_{B,7}$ priorities the dynamic performance of the resulting closed-loop system (small α). All the CVs $c_{A,n}$, and $c_{B,n}$ (n defines the number of measurements) can be seen in Table 8.3, whereas their controllers and steady-state loss are shown in Table 8.4. In addition, the resulting H_∞ norms of the closed-loop system ($\|T_{w,z}\|_\infty$) is also given in Table 8.4, which is used as the measurement of their dynamic performance.

Out of the three disturbances F_0, x_F, and q_F, changes in feed flow F_0 have no steady-state effect on the cost. Therefore, the dynamic performance for changes in F_0 is only considered, as it makes it easier to compare the transient responses. The simulation for a step change in F_0 can be seen in Figure 8.15 and shows that a better response is achieved for $c_{B,4}$ compared to $c_{A,4}$. This is expected since the measurement combination for $c_{B,4}$ was computed using Algorithm 7 that makes a trade-off between minimizing the H_∞ norm and steady-state loss, whereas the measurement combination for $c_{A,4}$ had been obtained *a priori*, where only the steady-state loss was considered. It is interesting to note that using 7 measurements ($c_{A,7}$ and $c_{B,7}$), gives the best and the worst dynamic response depending on whether the focus lies on minimizing the H_∞ norm of the closed-loop system or the steady-state performance. This would suggest that the dynamic considerations when selecting measurement combinations become more crucial, the more measurements are used. Furthermore, $c_{B,7}$ also has the second lowest steady-state loss, and thus, it can be seen that a small sacrifice in the steady-state loss might give a significant improvement in the control performance.

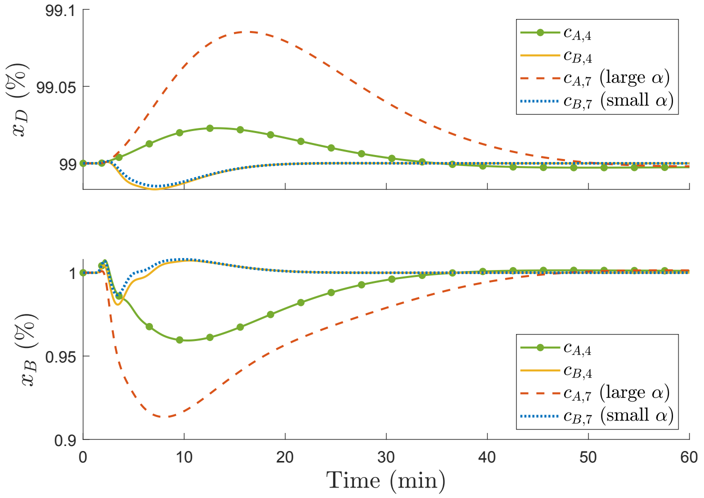


Figure 8.15: Distillate and bottom compositions changes in the binary distillation column for a step disturbance of -10% in F_0.

8.5 Structured measurements

In the previous section, an algorithm was proposed that aims to find the optimal trade-off between dynamic and steady-state performance by simultaneously computing the linear measurement combinations and the PI controllers. So far, it has been assumed that there are no structural constraints on the measurement combination H, i.e., that H is a full matrix. However, as discussed in Chapter 5, in many practical cases, it can be preferable to impose some structural constraints on the measurement combination H. The main reasons were to improve the dynamic controllability, to combine measurements that are of similar type, or to avoid pairing MVs and CVs that are located far apart. Although the dynamic controllability becomes less of an issue when using, e.g., Algorithm 7, the other two reasons may still require the use of a structured H. In particular, the second reason could make it easier to convince operators to implement these types of control structures in their plants, since combining measurements of similar type give the CVs a more intuitive meaning.

Contrary to using a full H matrix, the optimal measurement combinations when structural constraints are imposed can be unique. E.g., if a block diagonal constraint is used, where only one element is allowed in each column of H, then the optimal solution is unique when not considering scaling of the matrix H. Therefore, the relationship between the dynamic and the steady-state performance can become more significant for structured measurement combinations. To incorporate structural constraints in the measurement combination, an ADMM algorithm was proposed in Section 5.2.2 when only considering the steady-state loss. Here, an augmented Lagrangian associated with the original objective function (HY) and the bilinear constraint ($H = Q\hat{H}$) was used to formulate the optimization problem:

$$\min_{H,Q,\hat{H}} \frac{1}{2}\left\|HY\right\|_F^2 + \frac{\rho}{2}\left\|H - Q\hat{H} + \Lambda\right\|_F^2 \tag{8.74}$$

$$\text{subject to } \; HG^y = J_{uu}^{1/2} \tag{8.75}$$

$$\hat{H} \in \mathcal{S} \tag{8.76}$$

where Λ is the dual variable and ρ is positive scalar. The above optimization problem was solved by using the iterative procedure described in (5.53)−(5.55).

Since Algorithm 7 and the ADMM algorithm are both solved iteratively, it becomes fairly straightforward to combine them such that is possible to find a local solution that trades-off the steady-state loss against the dynamic performance for the resulting closed-loop system. An example of how this can be done is demonstrated in Algorithm 8, where similar to Algorithm 7 the H_∞ norm has been used to measure the dynamic performance with the resulting PI controllers being given in their discrete-time ideal form.

Algorithm 8 Trade-off between steady-state loss and dynamic performance for structured measurement combinations.

```
Initialize: For $k = 0$ choose a stabilizing state feedback gain $K_{SF}$ and a measurement combination matrix $\hat{H}^k$.
1: Set $k \leftarrow k + 1$, and for the fixed $K_{SF}$ and $\hat{H}^k$ solve the LMI:
```

$$J_1^k = \min_{\substack{Z_1,Z_2,P\\X_1,X_2,Q,H}} \gamma^2 + \alpha\frac{1}{2}\left\|HY\right\|_F^2 + \frac{\rho}{2}\left\|H - Q\hat{H} + \Lambda^k\right\|_F^2 \tag{8.77}$$

$$\text{subject to } \; HG^y = J_{uu}^{1/2}, (8.40), (8.41), (8.42), \text{and } (8.43).$$

```
2: Fix $Q$, $X_1$, $Z_1$, and $Z_2$ at the values obtained in step 1 and solve the LMI:
```

$$J_2^k = \min_{P,X_2,\hat{H},H} \gamma^2 + \alpha\frac{1}{2}\left\|HY\right\|_F^2 + \frac{\rho}{2}\left\|H - Q\hat{H} + \Lambda^k\right\|_F^2 \tag{8.78}$$

$$\text{subject to } \; HG^y = J_{uu}^{1/2}, \hat{H} \in \mathcal{S}, (8.40), (8.42), \text{and } (8.43).$$

```
3: Update $\Lambda^{k+1} = \Lambda^k + H^k - Q^k\hat{H}^k$
4: If $\|H^{k-1} - H^k\|_2 < \epsilon_1$, and $\|H^k - Q^k\hat{H}^k\|_2 < \epsilon_2$ stop, else repeat step 1 to 4.
```

The value of the scalar α determines how much the steady-state loss should be prioritized over the dynamic performance. The final controller parameters are obtained from $K_p = I$, and $K_i = X_1^{-1}X_2$, with the measurement combination given by $\hat{H}$, where $\mathcal{S}$ are the structural constraints that have been imposed on $\hat{H}$.

The convergence of Algorithm 8 is not completely known since, for ADMM algorithms, convergence to a globally optimal solution is only guaranteed for convex-problems. However, for non-convex problems, an ADMM algorithm should converge as long as ρ has been chosen to be sufficiently large [50]. Thus, if the algorithm has been initialized with a state feedback gain K_{SF} that gives a feasible solution in step 1, then with sufficiently large ρ, then Algorithm 8 should converge to a local minimum.

8.5.1 Case study: Binary distillation column

Here, the binary distillation column model described in Section 8.2.1 will again be used as a case study to evaluate Algorithm 8. The purpose is to find control structures that give the optimal trade-off between the steady-state loss and the dynamic performance when there are structural constraints are imposed on the measurement combination. The problems were formulated with the YALMIP toolbox [90] in Matlab and solved using the solver MOSEK [99].

The objective for the distillation column is to minimize the deviations in the top product composition x_D and the bottom product composition x_B. The available inputs u are given by

$$u = \begin{bmatrix} L_R & V_B \end{bmatrix}^T, \tag{8.79}$$

where L_R and V_B are liquid recycle flow and the vapor boilup rate, respectively. The main disturbances consist of changes in feed flow rate (F_0), feed composition (x_F), and feed liquid fraction (q_F):

$$d = \begin{bmatrix} F_0 & x_F & q_F \end{bmatrix}^T. \tag{8.80}$$

To evaluate the algorithm, the following set of temperature measurements has been selected:

$$y = \begin{bmatrix} T_5 & T_7 & T_{10} & T_{18} & T_{20} & T_{30} & T_{33} & T_{35} \end{bmatrix}^T. \tag{8.81}$$

Thus, the goal is to find PI controllers and measurement combinations H for the measurements in (8.81), where some structural constraints have been imposed on H. These will be block diagonal constraints, where the four temperature measurements closest to the bottom $\{T_5, T_7, T_{10}, T_{18}\}$ will be separated from the four

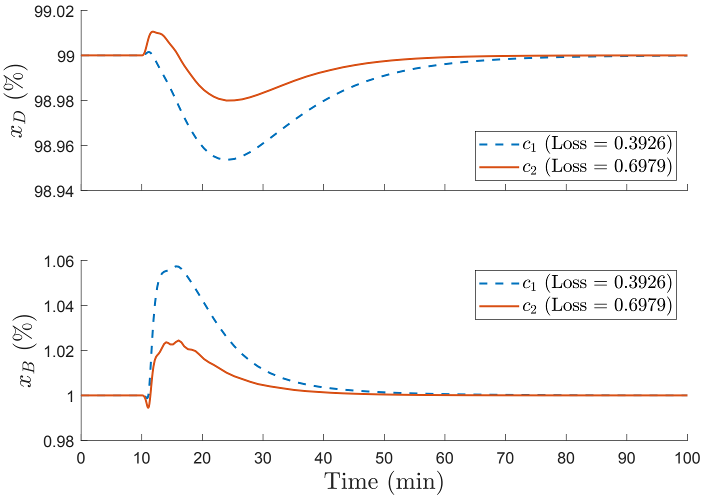


Figure 8.16: Deviations in x_D and x_B, for a +10% step change in F_0 after 10 min.

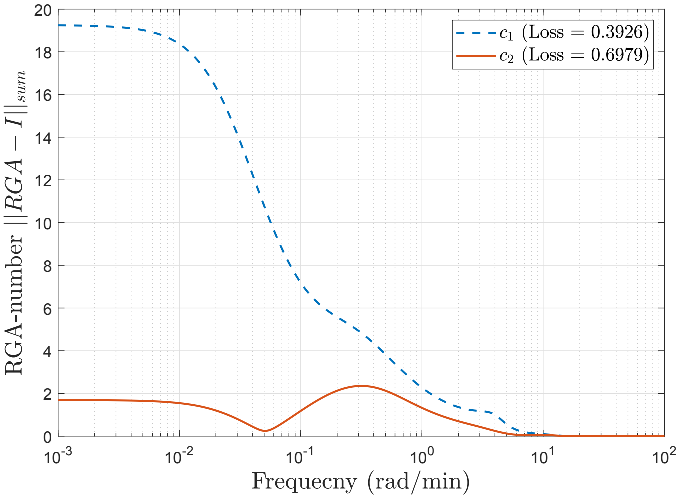


Figure 8.17: RGA-number for the CVs: $c_1 = H_1y$ and $c_2 = H_2y$.

measurements at the top of the column $\{T_{20}, T_{30}, T_{33}, T_{35}\}$. Two different CVs were computed $c_1 = H_1y$, and $c_2 = H_2y$, where c_1 puts a higher priority on minimizing the steady-state loss at the expense of the dynamic performance. Using Algorithm 8, the measurement combination matrix H_1 for c_1 was given by

$$H_1 = \begin{bmatrix} 0.000 & 0.000 & 0.000 & 0.000 & -0.613 & 1.220 & 0.954 & 0.702 \\ -0.237 & -0.357 & -0.546 & -0.080 & 0.000 & 0.000 & 0.000 & 0.000 \end{bmatrix}$$

with the following PI controllers:

$$Kp_1 = I, \quad Ki_1 = \begin{bmatrix} 0.0836 & 0 \\ 0 & 0.4390 \end{bmatrix}.$$

The second CV denoted c_2, puts more emphasis on the dynamic behavior and should thus, give a CV with a higher steady-state loss compared to c_1. This resulted in the following measurement combination,

$$H_2 = \begin{bmatrix} 0.000 & 0.000 & 0.000 & 0.000 & -2.363 & 2.022 & 1.576 & 1.159 \\ -0.427 & -0.649 & -1.012 & -0.308 & 0.000 & 0.000 & 0.000 & 0.000 \end{bmatrix},$$

where the PI parameters were given by,

$$Kp_2 = I, \quad Ki_2 = \begin{bmatrix} 0.1060 & 0 \\ 0 & 0.3878 \end{bmatrix}.$$

A simulation of a step change in the feed flow can be seen in Figure 8.16, where it can be seen that c_2 gives a better response but at the cost of a higher steady-state loss. In addition, the controllability of the two CVs (c_1, and c_2) was analyzed by investigating the RGA [22] with respect to the frequency. However, plotting the magnitudes of the RGA element (as was done in Section 6.1) can sometimes be misleading since it doesn't distinguish between positive and negative values. Instead, the RGA-number will be used, which is given by [125]:

$$\text{RGA-numnber} = \|RGA - I\|_{sum}. \tag{8.82}$$

Here, it is preferable to select CVs that have a small RGA-number at the crossover frequency. The RGA-number for the two CVs can be seen in Figure 8.17 and shows that c_2, which was computed with more emphasis on the dynamic behavior, has a lower RGA-number for the whole frequency range. Thus, the CV given by c_2 is also likely to be more robust against uncertainties in the dynamic process model, since the level of interactions are not as sensitive to the closed-loop bandwidth. However, these improvements in the dynamic performance need to be weighed against the increase in the steady-state loss to determine whether they are worth it or not.

8.6 Conclusion

In this chapter, the transient behavior when selecting the self-optimizing control variable has been considered. The main objective has been to find measurement combinations together with PI controllers that minimize the dynamic impact from disturbances while still achieving an acceptable steady-state economic loss based on the self-optimizing control conditions. First, an algorithm was proposed that finds the controllers and the CVs that optimizes the dynamic performance while still maintaining the optimal steady-state behavior when using either the null space or the exact local method. The second algorithm, incorporated the sparsity promoting weighted l_1 norm into the objective function to penalize the number of measurements used. Thus, it searches for a measurement subset and a control structure that will minimize the dynamic impact disturbances have on the closed-loop system while satisfying the null space condition as the self-optimizing control criteria. The third algorithm that was developed computes the measurement combination together with PI controllers and gives the optimal trade-off between dynamic and steady-state performance. A penalty function can also be included that penalizes the number of measurement used and attempts to find a subset for the desired trade-off. In the final algorithm, the PI controllers and CVs are computed when some structural constraints have been imposed on the measurement combinations. These problems may appear in situations where it is desirable to combine measurements that are of similar type or if the inputs and measurements that are associated with the same units of the process should be used in the same control loop.

All the proposed algorithms are trying to solve an optimization problem that includes a BMI, and thus, they are non-convex. The focus of this work has been on simultaneous control structure and controller design, and not on the solution of BMIs. Therefore, the use of global optimization solvers for the resulting BMI problems [43] have not been investigated. Instead, the algorithms use an iterative procedure to solve the problem. However, several non-trivial examples demonstrate that good results are found both with respect to the (near) optimality of the steady-state solution and the control performance of the resulting closed-loop system.

The examples used to evaluate the algorithms consisted of different distillation column models. Distillation columns are dynamically interactive and often ill-conditioned; thus, they work as great case studies for control structure design, when using the proposed methods. In particular, the Petlyuk and the Kaibel distillation columns are interesting problems, since divided wall columns are known to be challenging to control when using decentralized strategies, where, e.g., [1] suggested that an MPC should be used to achieve good controllability. However, as illustrated in Section 8.3.1, and 8.2.2, when using the proposed algorithms, it was possible to find decentralized control structures with comparable performance to an MPC. Therefore, it seems to indicate that well-tuned decentralized PI controllers can give comparable results to more advanced control structures if the controlled variables (measurement combinations) are chosen properly.

It is possible that better performance can be achieved if more advanced controller formulations are used instead of decentralized PI controllers. The main difference would be how the controller description is included in the formulation of the augmented system matrices described in Section 7.3. One important requirement is that the controllers incorporate some integral action to ensure that the self-optimizing control criterion is met at steady-state. However, the CVs (measurement combinations) obtained using the proposed methods are likely to have smaller interactions between each other, compared to measurement combinations that have been obtained without considering the dynamics. Therefore, using a decentralized PI control structure allows for relatively easy retuning of the controllers, when compared to advanced multi-variable controllers, should future changes in operating conditions make retuning necessary. Thus, the small potential improvements in performance for more advanced control strategies may not be worth their additional complexity.

Part IV

Hierarchical decentralized state estimation

Chapter 9

Decentralized state estimation

The increasing demands for efficient operation and profitability in the industry have led to higher demands on the control systems. The control systems that have been implemented should, ideally, be capable of driving the plant to the new optimal operating point as efficiently as possible, whenever the plant is being affected by disturbances. In the previous chapters, control structures were designed for the lower regulatory control layer that were able to achieve near-optimal operation, both in terms of the economic steady-state and the dynamic performance. These control structures were computed based on the principles of self-optimizing control, where the goal was to find simple control structures consisting of, e.g., decentralized PI controllers that give a performance comparable to more complex control structures. However, for many processes, implementing self-optimizing control on the lower layer controllers may not suffice, especially, if the systems are very nonlinear and there are frequent changes in the active constraints for the optimal operation. Therefore, it becomes necessary to incorporate some more advanced control systems in the upper layers, e.g., real-time optimizers (RTO), model predictive controllers (MPC) and economic MPCs.

To successfully implement such techniques, more frequent information of the states and disturbances is required [150]. Measuring all the important states and disturbances is often too expensive and often not possible. Therefore, they need to be estimated, using appropriate state estimators as, e.g., in [93], and [141]. For chemical processes and other large-scale systems, using centralized state estimators are in general not favorable due to the high computational complexity. Instead, it is preferable to decompose the problem into several subsystems with local estimators, that uses the locally available measurements [137].

In [40], the concepts of distributed and parallel state estimation was introduced, where it was concluded that distributed state estimation is a viable option when estimating the states for complex and large-scale systems. The principle of distributed or decentralized state estimation is to split the state estimation problem into several smaller subsystems that are able to produce their own local state estimate. By collecting the local estimates and fusing them together, it should be possible to obtain a new global state vector with better accuracy compared to the local estimates. Therefore, this chapter will briefly cover some of the most common fusion methods, which will work as the background for the proposed fusion method that will be presented in Chapter 10.

9.1 Fusion strategies

Consider a discrete system described by a nonlinear dynamic model $f(\cdot)$ and n different measurement models $h_j(\cdot)$:

$$x(k+1) = f\big(x_j(k), u(k)\big) + w(k), \tag{9.1}$$

$$y_j(k) = h_j\big(x(k)\big) + v_j(k), \tag{9.2}$$

where $x(k) \in \mathbb{R}^{n_x}$ and $u(k) \in \mathbb{R}^{n_u}$ are the states and inputs, respectively. The process noise $w(k)$ and the measurement noise $v_j(k)$ are uncorrelated zero mean white Gaussian noise with the respective covariance matrices $Q_w(k)$, and $R_{v,j}(k)$. Assume that there exists a different local estimator for every measurement model $y_j(k)$ that tries to estimate the same state vector using the model given in (9.1). The local estimators can be of different type, but they all produce a local estimate together with a covariance matrix that can be represented by $N(\hat{x}_j, P_j)$. The goal is to fuse together the local estimates to obtain a new covariance matrix and a global state vector with better accuracy compared to the local state estimates.

9.1.1 Naive Fusion

A state vector and its covariance matrix can be represented as an information vector and an information matrix, also known as their canonical form [16]. The canonical form is commonly used in distributed and decentralized state estimation, since fusing estimates then becomes equivalent to adding the information matrices and vectors together. The information matrix and information vector are defined as:

$$\Omega(k|k) := P(k|k)^{-1}, \tag{9.3}$$

$$\xi(k|k) := P(k|k)^{-1}\hat{x}(k|k), \tag{9.4}$$

where $\hat{x}(k|k)$ is the state estimate, and $P(k|k)$ is its covariance matrix at time k.

The corresponding fused information matrix and information vector can then be given by

$$\Omega(k|k) = \sum_{j=1}^{n} \Omega_j(k|k) \tag{9.5}$$

$$\xi(k|k) = \sum_{j=1}^{n} \xi_j(k|k) \tag{9.6}$$

where Ω_j and ξ_j are the information matrix and information vector from the jth subsystem, respectively. The global covariance matrix and state vector can then simply be obtained from:

$$P(k|k) = \Omega(k|k)^{-1} \tag{9.7}$$

$$\hat{x}(k|k) = P(k|k)\xi(k|k). \tag{9.8}$$

The global estimate in (9.8) and its covariance matrix in (9.7) are optimal if there is no correlation between the local estimates. This assumption rarely holds and has, therefore, been referred to as the "naive approach" [28]. A typical cause of correlation is that the local estimates are affected by the same process noise, e.g., a temperature measured directly and the estimated temperature obtained by another node from a pressure measurement combined with a reaction model will be correlated [67]. If the cross-correlations are ignored, it will result in overconfident and inconsistent state estimates. This will make the state estimates less accurate and could even lead to divergence if the result is fed back to local estimators.

9.1.2 State fusion under known correlation

The Bar-Shalom Campo (BC) formula [15] is a well-known method for incorporating the known cross-correlation when fusing state estimates. After dropping $(k|k)$ for ease of notation, the BC formula can be given by

$$P_{BC} = P_1 - (P_1 - P_{12})(P_1 + P_2 - P_{12} - P_{21})^{-1}, \tag{9.9}$$

$$\begin{aligned}\hat{x}_{BC} &= (P_2 - P_{21})(P_1 + P_2 - P_{12} - P_{21})^{-1}\hat{x}_1 \\ &\quad + (P_1 - P_{12})(P_1 + P_2 - P_{12} - P_{21})^{-1}\hat{x}_2,\end{aligned} \tag{9.10}$$

where P_{12} and P_{21} constitute the cross-correlations. From a maximum likelihood sense, this formula results in consistent fusion. However, keeping track on and maintaining these cross-correlations is expensive, especially for large-scale systems. Instead, different suboptimal strategies are often used, that gives a fused solution without the need to maintain the cross-correlations.

9.1.3 State fusion under unknown correlation

Several methods exist for fusing estimates with unknown correlations. One of the most popular methods used is known as covariance intersection (CI) [69]. It determines the fused estimate by multiplying the information matrices and vectors with a scalar weight, $\omega \in \mathbb{R}_{[0,1]}$:

$$P_{CI} = (\omega \cdot \Omega_1 + (1 - \omega) \cdot \Omega_2)^{-1}, \tag{9.11}$$

$$\hat{x}_{CI} = P_{CI}(\omega \cdot \xi_1 + (1 - \omega) \cdot \xi_2). \tag{9.12}$$

Numerous approaches for determining the weight ω exists, e.g., [29] and [45] but in general, they attempt to minimize the trace or the determinant of P_{CI}. The CI method guarantees that the fused estimate is consistent as long as the local estimates are consistent. It does so by overestimating the covariances in all directions, thus ensuring the fused covariance is larger than for the worst-case cross-correlation scenario. This is also one of its drawbacks, as it does not necessarily reduce the estimation error.

Contrary to CI, which attempts to obtain a minimum overestimation of the intersection region between covariances, the ellipsoidal intersection (EI) [118] and the largest ellipsoid algorithm (LEA) [18] aim to find the maximum ellipsoid inside the region of the intersection.

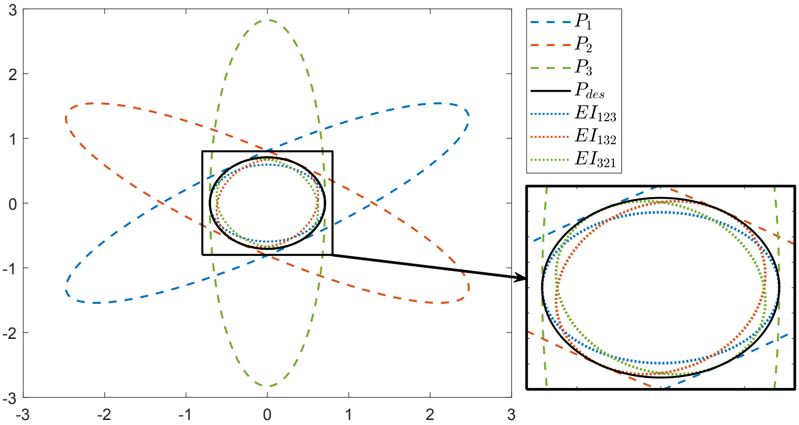


Figure 9.1: Covariance ellipses P_1, P_2, and P_3; the desired ellipse P_{des}; and the fused covariances EI_{123}, EI_{132}, and EI_{321} when fusing P_1, P_2, and P_3 with ellipsoidal intersection in different orders.

EI ensures that the uncertainty decreases for the fused estimate (i.e., $P_{EI} \preceq P_1$, and $P_{EI} \preceq P_2$), and gives a reduction in the estimation error compared to using CI. It does so by calculating the mutual covariance P_m and the mutual mean x_m, yielding:

$$P_{EI} = (P_1^{-1} + P_2^{-1} - P_m^{-1})^{-1}, \tag{9.13}$$

$$\hat{x}_{EI} = P_{EI}(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2 - P_m^{-1}x_m). \tag{9.14}$$

The EI and LEA methods are similar since they both aim to obtain the maximum ellipsoid inside the region of the intersection through eigenvalue (or singular value) decomposition. However, EI uses eigenvalue decomposition to find the mutual information (P_m and x_m), whereas LEA computes a transformation matrix T such that,

$$\tilde{P}_1 = TP_1T^T, \tag{9.15}$$

$$\tilde{P}_2 = TP_2T^T, \tag{9.16}$$

becomes diagonal matrices with $\tilde{P}_1$ being equal to the identity matrix. The transformation matrix T can be found by first taking the singular value composition (svd) of P_1,

$$[U_1, D_1] = \text{svd}(P_1), \tag{9.17}$$

and then compute the svd for:

$$[U_2, D_2] = \text{svd}(D_1^{-\frac{1}{2}}U_1^TP_2U_1D_1^{-\frac{T}{2}}). \tag{9.18}$$

The diagonal matrix $\tilde{P}_2$ can then be obtained from

$$\tilde{P}_2 = D_2 = U_2^TD_1^{-\frac{1}{2}}U_1^TP_2U_1D_1^{-\frac{T}{2}}U_2, \tag{9.19}$$

whereas $\tilde{P}_1$ is given by

$$\tilde{P}_1 = U_2^T D_1^{-\frac{1}{2}} U_1^T P_1 U_1 D_1^{-\frac{T}{2}} U_2. \tag{9.20}$$

Replacing P_1 with $U_1 D_1 U_1^T$ in (9.20) gives

$$\tilde{P}_1 = U_2^T D_1^{-\frac{1}{2}} U_1^T U_1^T D_1 U_1 U_1 D_1^{-\frac{T}{2}} U_2, \tag{9.21}$$

and because U_1 and U_2 are unitary matrices, multiplying them by their respective conjugate transposes yields the identity matrix:

$$\begin{aligned} \tilde{P}_1 &= U_2^T D_1^{-\frac{1}{2}} D_1 D_1^{-\frac{T}{2}} U_2, && (9.22)\\ &= U_2^T U_2, && (9.23)\\ &= I. && (9.24) \end{aligned}$$

Thus, by defining the transformation matrix T as

$$T := U_2^T D_1^{-\frac{1}{2}} U_1^T, \tag{9.25}$$

the diagonal matrices $\tilde{P}_1$ and $\tilde{P}_2$ can be given by (9.15) and (9.16), respectively. The intersection ellipsoid can then be calculated from,

$$P_{LEA} = T^{-1} \tilde{P}_{LEA} T^{-T}, \tag{9.26}$$

where $[\tilde{P}_{LEA}]_{j,j} = \min([\tilde{P}_1]_{i,i}, [\tilde{P}_2]_{j,j})$.

The similarity between EI and LEA was also addressed in [105]. However, the fused mean for LEA is not adapted to the fused covariance, which is a significant difference compared to EI.

9.2 State fusion for more than two local estimates

The framework of the three fusion methods (CI, EI, and LEA) have been devised for fusing two estimates only. When extended to multiple estimates, the general recommendation is to sequentially apply these fusion methods [33], [89]. However, as demonstrated in [14] when using CI for three or more estimates, it tends to overestimate the covariance, and therefore, the minimum overestimate no longer holds. On the other hand, using a sequential approach for EI, and LEA, may lead to an underestimate of the uncertainty, that is, result in a covariance that is smaller than the maximum ellipsoid inside the intersection region as seen in Figure 9.1. Furthermore, the order of the sequence of which the fusion is done affects the final estimates as also illustrated in Figure 9.1, where the three covariances P_1, P_2, and P_3 end up with a different fused covariance depending on the sequential order.

9.3 Conclusion

This chapter covered some of the popular approaches for fusing state estimates that have been computed by different local state estimators. When the cross-correlation between the local estimates is known, then the Bar-Shalom Campo (BC) formula can be used to reconstruct a state estimate that is optimal in the sense of maximum likelihood. However, keeping track of the cross-correlations is difficult for large-scale systems, and instead, it is preferable to use fusion methods that don't account for the correlations between the local estimates.

Naive fusion offers a simple and straightforward way of computing a global state vector, but can give inconsistent estimates and does not necessarily reduce the estimation error. Covariance intersection (CI) can guarantee consistency for the fused estimate by multiplying all the local estimates with a scalar weight. The weights decide to what extent the different local estimates should be emphasized when they are fused together. However, since these weights are scalars, it means that every estimate in the local state vectors will be scaled equally, despite that the accuracy of the different estimates can vary within each state vector. As a consequence, CI is unlikely going to be able to reduce the estimation error for all the state estimates. This issue is partly solved when using ellipsoidal intersection (EI), which improves the accuracy in all direction. Unfortunately, EI is restricted to fusing two state vectors and requires the use of a sequential approach when extended to multiple estimates, from which the fusion order will influence the final result.

Motivated by the advantages and disadvantages of the above fusion methods, an alternative approach for computing weights for the local estimates will be presented in the next chapter. However, instead of having the weights be scalars as in CI, the proposed fusion method aims to find weight matrices that will, similarly to EI, be able to reduce the estimation error for all estimates, but where the final result is independent of any sequential order.

Chapter 10

State fusion for partially overlapping state estimates

The previous chapter covered some of the common state fusion methods when fusing state estimates that have been obtained using different state estimators. However, these approaches were restricted to problems that involved fusing only two estimates that where both trying to estimate the same state vector. Therefore, they are not particularly well suited for decentralized state estimation in chemical processes or other large-scale systems, since in general, they would require fusing several local estimates of different size and with only partially overlapping state vectors.

Instead, when estimating the states for complex and large-scale systems, the concepts of distributed and parallel state estimation should be used that was introduced in [40]. Here, information can be shared between all the subsystems or only with neighboring subsystems. From the shared information, it should be possible to obtain a global estimate. Two main approaches exist for dealing with the distributed/decentralized state estimation problem:

One approach is to have every subsystem estimate the entire state vector using a global model for each local estimator. A global estimate is then received by fusing all the local estimates in a centralized fusion layer as in [151]. The main drawback of this approach is that it is a full order problem, and each subsystem must have access to a full dynamical model of the system. Therefore, this approach is poorly suited for most chemical processes and other large-scale systems.

In the second approach, the different subsystems compute only a partial estimate using a local model and then transmits the information to its neighbors. This approach is also referred to as partition-based state estimation [42]. It is in general, more preferable for large-scale systems as in [137] since it results in low order estimation problems that rely only on local dynamic and measurement models. What information that is shared between the neighboring subsystems vary depending on the distributed estimation scheme, but can be the measurements, state estimates, and corresponding covariances. The main drawback with this method is that it requires good performance for the communication between the local estimators, where corrupted information, information delays, and transmission frequency limitations may cause problems.

Consensus algorithms for distributed state estimation based on Kalman filters have recently received much attention [108], [86], [85]. In general, they belong to the second approach, where each node (local estimator) computes an estimate and shares it with its neighbors. The estimates by the different nodes converge to the same state estimate using a consensus algorithm. The Kalman based consensus filters have mostly been applied to target tracking [72], where each local estimator tries to estimate the same state vector. However, a consensus based method for merging partially overlapping state estimates was also proposed in [128].

One aspect most of the distributed state estimation methods have in common is that they are typically designed using a top-down approach, i.e., they start with a global model and then decompose it into smaller subsystems. The local estimators for the decomposed subsystems then often employ the same state estimation algorithm.

However, a major issue in process control is that most chemical plants, together with their different units, are unique. Therefore, most models have to be obtained using empirical modeling, requiring excitation of the process through, e.g., system identification. These experiments are often time-consuming and expensive since it interferes with the process operation. As a consequence, the accuracy of the models varies greatly, where some parts of the process can have very accurate nonlinear models, while other parts have basic linear models. Thus, using the same estimator algorithm for every subsystem would be suboptimal. Instead, it would be more practical to use a bottom-up approach, where each local estimator is independently designed using a local model and the local measurements. The local estimates can be sent to a fusion center, from which a more accurate global state estimate can be recovered. Therefore, in this chapter, a fusion method is proposed that aims to fuse together several local estimates that are estimating different parts of the process.

10.1 Hierarchical state estimation

Hierarchical state estimation can be done by computing the global information matrix, and vector from n different subsystems,

$$\Omega(k|k) = \Omega(k|k-1) + \sum_{i=1}^{n} \Big(\Omega_i(k|k) - \Omega_i(k|k-1)\Big), \tag{10.1}$$

$$\xi(k|k) = \xi(k|k-1) + \sum_{i=1}^{n} \Big(\xi_i(k|k) - \xi_i(k|k-1)\Big), \tag{10.2}$$

where the information matrix, and vector were defined in (9.3), and (9.4) respectively. Here, each local estimator is assumed to have the global model available and tries to estimate the whole state vector using the locally available measurements. This can essentially be interpreted as having a centralized estimator that computes a global *a priori* estimate $\xi(k|k-1)$, $\Omega(k|k-1)$, and then receives the additional information containing the measurements from each local estimator. This approach has been shown by [57] to be equivalent to using a centralized state estimator, but the requirement of a global model makes it impractical for large-scale systems.

Instead, a more desirable approach would be to have the local estimators only estimate a part of the global state vector and then use a fusion center to reconstruct a global state estimate.

Let the nonlinear dynamic model of a large-scale process be given by the discrete-time system:

$$x(k+1) = f\big(x(k), u(k)\big) + w(k), \tag{10.3}$$

$$y(k) = h\big(x(k)\big) + v(k), \tag{10.4}$$

where $x(k) \in \mathbb{R}^{n_x}$ and $u(k) \in \mathbb{R}^{n_u}$ are the states and inputs, respectively. The process noise $w(k)$ and the measurement noise $v(k)$ are uncorrelated zero mean white Gaussian noise with the respective covariance matrices $Q_w(k)$, and $R_v(k)$. Decomposing the system into n subsystems of a lower order where index i identifies the ith subsystem represented as:

$$x_i(k+1) = f_i\big(x_i(k), u_i(k), X_i(k)\big) + w_i(k) \tag{10.5}$$

$$y_i(k) = h_i\big(x_i(k), X_i(k)\big) + v_i(k) \tag{10.6}$$

Here the vector $X_i(k)$ represents the states which are shared among the neighboring subsystems, but for which the model equations are unknown to subsystem i.

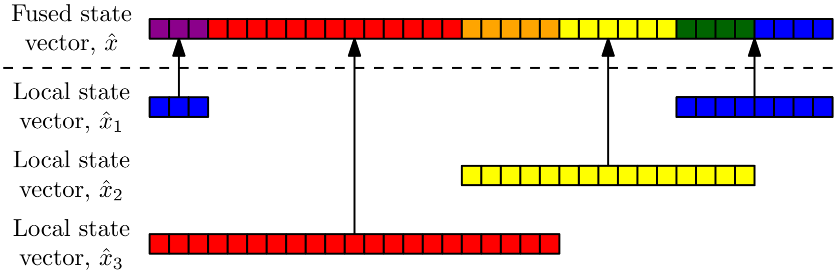

Fused state vector, $\hat{x}$
Local state vector, $\hat{x}_1$
Local state vector, $\hat{x}_2$
Local state vector, $\hat{x}_3$

Figure 10.1: Fusion of partially overlapping local state estimates.

Each subsystem uses a decentralized state estimator, that can operate independently, i.e., it does not require any information from any of the other subsystems. For the local estimator to operate without the need to communicate with its neighbors, the local model in (10.5) is augmented with integrators that represent the interacting states $X_i(k)$. Thus, the augmented model for the local estimators becomes:

$$\bar{x}_i(k+1) = \begin{bmatrix} x_i(k+1) \\ X_i(k+1) \end{bmatrix} = \begin{bmatrix} f_i(x_i(k), u_i(k), X_i(k)) \\ X_i(k) \end{bmatrix} \tag{10.7}$$

It is assumed that the augmented state vector $\bar{x}_i(k)$ is observable using the locally available measurements $y_i(k)$. The local estimators can be of different type, but they all produce a local estimate together with a covariance matrix that can be represented by $N(\hat{\bar{x}}_i, \bar{P}_i)$. Each of the local estimates $\hat{\bar{x}}_i$ represents a smaller part of the global state vector in (9.1) where some parts of the local estimates overlap with each other as in Figure 10.1. Therefore, the objective is to collect and fuse the local estimates together in a fusion center, using a hierarchical decentralized structure, as shown in Figure 10.2. However, the role of the fusion center is not only to assemble the global state vector, but also to get a more accurate state estimate.

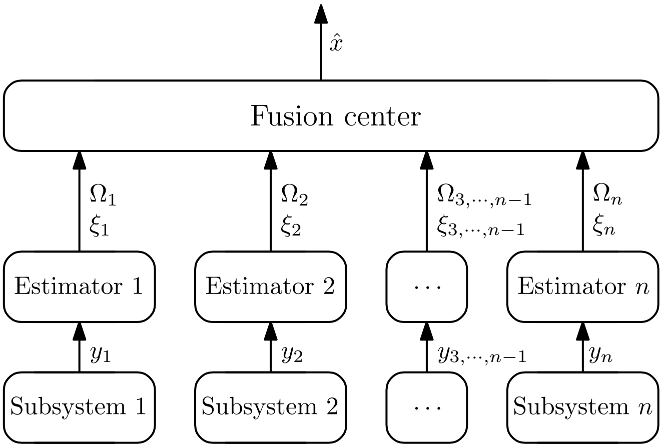


Figure 10.2: The hierarchical decentralized state estimator.

Therefore, instead of having each local estimator compute both the *a posteriori* and the *a priori* estimate of the entire global state vector as in (10.1) and (10.2), it would be preferable to fuse only the *a posteriori* information matrices and vectors directly from the local models:

$$\Omega(k|k) = \sum_{i=1}^{n} \Gamma_i \Omega_i(k|k) \Gamma_i^T \tag{10.8}$$

$$\xi(k|k) = \sum_{i=1}^{n} \Gamma_i \xi_i(k|k) \tag{10.9}$$

Here, Γ_i is a transformation matrix, that maps the local state vectors $\hat{x}_i(k|k)$ to the global state vector $\hat{x}(k|k)$, i.e. $\hat{x}_i(k|k) = \Gamma_i^T \hat{x}(k|k)$.

Similar, to the "naive" approach given by (9.5) and (9.6), the global estimate from (10.8) and (10.9) is optimal if there is no correlation between the local estimates. However, as mentioned in Section 9.1.1, this is an unrealistic assumption, and in most practical cases will result in overconfident and inconsistent state estimates. For the proposed hierarchical decentralized state estimator (see Figure 10.2), consistency will not be a major issue, since the fused estimates have no influence on the local estimates. However, the cross-correlations should still be considered in order to improve the accuracy of the fused estimate.

10.2 State fusion with unequal state vectors

Compared to fusing state estimates that correspond to the same state vector, research regarding fusing partial overlapping state vectors is fairly limited. A consensus based method for fusing the overlapping parts have been proposed in [128]. In [119], an empirical method was investigated, where the EI method was extended to deal with partially overlapping state vectors. However, both these methods are better suited to fully distributed state estimation, where the local estimates are shared between its neighbors without any fusion center.

A more suitable technique for the desired hierarchical fusion structure was proposed in [104]. There, fusing unequal state vectors is treated as a weighted least squares (WLS) problem, where the fused state estimate becomes:

$$\hat{x}_{WLS} = K_e \begin{bmatrix} \hat{x}_1^T & \cdots & \hat{x}_n^T \end{bmatrix}^T \tag{10.10}$$

The gain K_e for unknown cross-correlations can be computed:

$$K_e = (H^T \Omega_{WLS} \Gamma)^{-1} H^T (\Omega_{WLS})^{-1}. \tag{10.11}$$

$$\text{where } \Gamma := \begin{bmatrix} \Gamma_1^T \\ \vdots \\ \Gamma_n^T \end{bmatrix}, \text{ and } \Omega_{WLS} := \begin{bmatrix} w_1\Omega_1 & & \\ & \ddots & \\ & & w_n\Omega_n \end{bmatrix}.$$

The scalar weights ω_i can be obtained using, e.g., CI such that $\omega_i \geq 0$, and $\sum_{i=1}^{n} \omega_i = 1$. However, no more information was given on how to compute ω_i. Furthermore, CI is not particularly well suited for dealing with partially overlapping state estimates. For instance, if the goal is to minimize the trace of the fused covariance, then using larger weights for the local estimates of lower dimensions would in general, be beneficial, even if these estimates are inaccurate.

10.3 Proposed fusion method

Motivated by some of the shortcomings when fusing multiple and partially overlapping state estimates, a method is proposed that tries to address some of these issues. The proposed method is inspired by EI [118] and LEA [18], as it tries to obtain a fused covariance that corresponds to the maximum ellipsoid inside the region of intersection. Therefore, similar to EI, it does not guarantee consistency [106] unless the local estimates are weakly correlated [89]. However, instead of using a sequential approach, the aim is to compute static weight matrices W_i such that the global state estimate becomes,

$$\hat{x} = \Omega^{-1}\xi, \tag{10.12}$$

where,

$$\Omega = \sum_{i=1}^{n} \Gamma_i W_i \Omega_i W_i^T \Gamma_i^T, \tag{10.13}$$

$$\xi = \sum_{i=1}^{n} \Gamma_i W_i \Omega_i W_i^T \hat{x}_i. \tag{10.14}$$

The weights W_i in (10.13), and (10.14) will be obtained by solving two semi-definite programming (SDP) problems.

Lemma 10.1. *[17] An inner ellipsoidal approximation of the intersection for the ellipsoids $\Gamma_1 \Omega_1 \Gamma_1^T, ..., \Gamma_n \Omega_n \Gamma_n^T$ can be obtained by solving the following linear matrix inequality (LMI):*

$$\max_{Z,t_i} \quad det(Z) \tag{10.15}$$

$$\textit{subject to:} \qquad t_i \geq 0, \qquad\qquad \forall i \tag{10.16}$$

$$\begin{pmatrix} I & 0 & \Gamma_i \Omega_i^{\frac{1}{2}} \Gamma_i^T Z \\ 0 & 1 - t_i & 0 \\ Z \Gamma_i \Omega_i^{\frac{1}{2}} \Gamma_i^T & 0 & t_i I \end{pmatrix} \succeq 0, \ \forall i \tag{10.17}$$

The approximation of the intersection Ω_0 is obtained from:

$$\Omega_0 = Z^{-2}. \tag{10.18}$$

The next step is to find weight matrices W_i such that (10.13) becomes equal (or close) to Ω_0. This can be achieved by finding transformation matrices T_i that manipulate the orientation of Ω_i such that they all align with Ω_0.

The singular value composition of Ω_0 is given by

$$[U_0, D_0] = \text{svd}(\Omega_0), \tag{10.19}$$

from which, a transformation matrix can be defined,

$$T_0 = D_0^{-\frac{1}{2}} U_0^T \tag{10.20}$$

that makes $T_0 \Omega_0 T_0^T = I$. The following transformation matrices can be computed for $i = 1...n$:

$$[U_i, D_i] = \text{svd}(T_0 \Gamma_i \Omega_i \Gamma_i^T T_0^T), \tag{10.21}$$

$$T_i = U_i^T D_0^{-\frac{1}{2}} U_0^T. \tag{10.22}$$

The transformation matrices, T_i can be used to bring Ω_0 and Ω_i within a space where they have compatible orientations:

$$\tilde{\Omega}_0 = T_i \Omega_0 T_i^T = I, \tag{10.23}$$

$$\tilde{\Omega}_i = T_i \Gamma_i \Omega_i \Gamma_i^T T_i^T, \tag{10.24}$$

where $\tilde{\Omega}_0$ and $\tilde{\Omega}_i$ are diagonal matrices. Next, the goal is to find some diagonal scaling weights $\tilde{W}_i$, such that the corresponding ellipsoid when transformed back to original space becomes:

$$\Omega_0 = \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i \tilde{\Omega}_i T_i^{-T} \tag{10.25}$$

$$= \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i \Gamma_i \Omega_i \Gamma_i^T T_i^T T_i^{-T} \tag{10.26}$$

$$= \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i \Gamma_i \Omega_i \Gamma_i^T. \tag{10.27}$$

Theorem 10.2. *The weights $\tilde{W}_i$ can be obtained by solving:*

$$\max_{\tilde{W}_i} \; det\Big(\sum_{i=1}^{n} T_i^{-1}\tilde{W}_i T_i \Gamma_i \Omega_i \Gamma_i^T\Big) \tag{10.28}$$

$$\textit{subject to:} \qquad \tilde{W}_i \geq 0, \qquad\qquad \forall i \tag{10.29}$$

$$\Big\| \sum_{i=1}^{n} T_i^{-1}\tilde{W}_i T_i \Big\|_2 \leq 1 \tag{10.30}$$

Proof. From (10.27), $(T_i^{-1}\tilde{W}_i T_i)$ can be considered as a weight matrix that when multiplied with $\Gamma_i\Omega_i\Gamma_i^T$ results in:

$$(T_i^{-1}\tilde{W}_i T_i)\Gamma_i\Omega_i\Gamma_i^T = \Gamma_i\Omega_i\Gamma_i^T(T_i^{-1}\tilde{W}_i T_i)^T \succeq 0. \tag{10.31}$$

The symmetry is due to $\tilde{W}_i$, and $\tilde{\Omega}_i$ in (10.25) being diagonal matrices, and the constraint in (10.29) guarantees positive semi-definiteness. The upper bound placed on $\tilde{W}_i$ through (10.30) ensure that:

$$\sum_{i=1}^{n} T_i^{-1}\tilde{W}_i T_i \Gamma_i\Omega_i\Gamma_i^T \preceq \Omega_0. \tag{10.32}$$

Therefore, the weights $\tilde{W}_i$ are optimal if $\sum_{i=1}^{n} T_i^{-1}\tilde{W}_i T_i = I$, which results in:

$$\sum_{i=1}^{n} T_i^{-1}\tilde{W}_i T_i \Gamma_i\Omega_i\Gamma_i^T = \Omega_0. \tag{10.33}$$

□

Remark Ideally, the constraint in (10.30) could be replaced with the expression $\sum_{i=1}^{n} T_i^{-1}\tilde{W}_i T_i = I$. However, due to possible round-off error and other numerical issues when computing T_i, it can give poor results. Therefore, the relaxed constraint in (10.30) is preferred and should result in $\sum_{i=1}^{n} T_i^{-1}\tilde{W}_i T_i \approx I$.

Because $\tilde{W}_i$ are diagonal matrices, (10.25) can be written as

$$\Omega_0 = \sum_{i=1}^{n} T_i^{-1}\tilde{W}_i^{\frac{1}{2}}\tilde{\Omega}_i\tilde{W}_i^{\frac{T}{2}}T_i^{-T}, \tag{10.34}$$

$$= \sum_{i=1}^{n} T_i^{-1}\tilde{W}_i^{\frac{1}{2}}T_i\Gamma_i\Omega_i\Gamma_i^T T_i^T\tilde{W}_i^{\frac{T}{2}}T_i^{-T}. \tag{10.35}$$

The weights in the original space then become

$$W_i = \Gamma_i^T T_i^{-1}\tilde{W}_i^{\frac{1}{2}}T_i\Gamma_i, \quad \forall i. \tag{10.36}$$

The procedure for computing the weights W_i is summarized in Algorithm 9.

Algorithm 9 Computation of the fusion weights in hierarchical state estimation.

```
Initialize: Collect the information matrices $\Omega_i$ from all the $n$ different local estimators together with their transformation matrices $\Gamma_i$ that map the local estimates to the global state vector.
1: Compute the maximum ellipsoid inside the region of intersection $\Omega_0 = Z^{-2}$, where $Z$ is computed by solving problem given in Lemma 10.1:
       $Z = \arg\max_{Z,t_i} \ \det(Z)$
       subject to:  $t_i \geq 0, \quad \forall i$
       $\begin{pmatrix} I & \mathbf{0} & \Gamma_i \Omega_i^{\frac{1}{2}} \Gamma_i^T Z \\ \mathbf{0} & 1 - t_i & \mathbf{0} \\ Z \Gamma_i \Omega_i^{\frac{1}{2}} \Gamma_i^T & \mathbf{0} & t_i I \end{pmatrix} \succeq 0, \ \forall i$
2: Calculate the transformation matrices $T_i$ from (10.19) that aligns $\Omega_0$ with $\Omega_i$ for all $i$:
       $T_i = U_i^T D_0^{-\frac{1}{2}} U_0^T,$
   where $D_0$, $U_i$, and $U_i$ is obtain using (10.20)–(10.22):
       $[U_0, D_0] = \operatorname{svd}(\Omega_0),$
       $T_0 = D_0^{-\frac{1}{2}} U_0^T,$
       $[U_i, D_i] = \operatorname{svd}(T_0 \Gamma_i \Omega_i \Gamma_i^T T_0^T).$
3: Compute the diagonal weight matrices $\tilde{W}_i$ by solving the optimization problem given in (10.28)–(10.30):
       $\tilde{W}_i = \arg\max_{\tilde{W}_i} \det\left( \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i \Gamma_i \Omega_i \Gamma_i^T \right)$
       subject to:  $\tilde{W}_i \geq 0, \quad \forall i$
       $\left\| \sum_{i=1}^{n} T_i^{-1} \tilde{W}_i T_i \right\|_2 \leq 1$
4: Calculate the weights $W_i$ in their original space from (10.36):
       $W_i = \Gamma_i^T T_i^{-1} \tilde{W}_i^{\frac{1}{2}} T_i \Gamma_i, \quad \forall i.$
```

The global state estimate can then be obtained from (10.12), (10.13), and (10.14). Alternatively, (10.10), and (10.11) can be used, where $\omega_i \Omega_i$ in Ω_{WLS} can be replaced with $W_i \Omega_i W_i^T$.

An example of a resulting fused covariance ellipse can be seen in Figure 10.3, together with the original and weighted covariances. The individual covariances get scaled, so the uncertainty mainly increases in the inaccurate directions. This forces the fused estimate to put more emphasis on the accurate estimates.

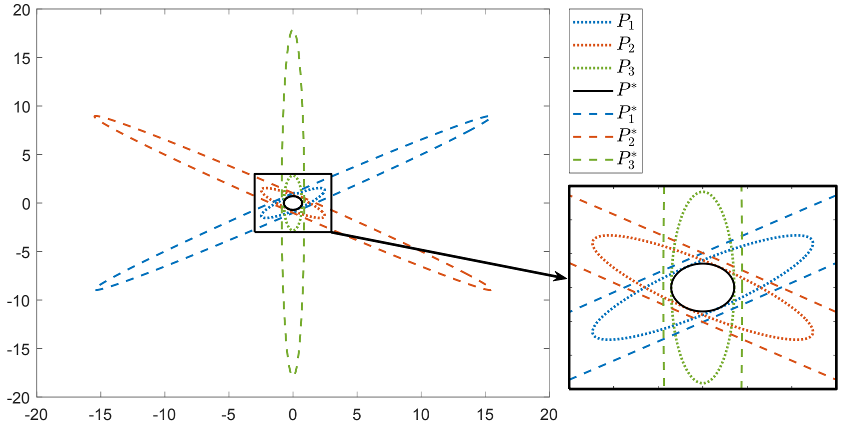


Figure 10.3: Covariance ellipses P_1, P_2, and P_3; the weighted covariances $P_1^* = (W_1\Omega_1 W_1^T)^{-1}$, $P_2^* = (W_2\Omega_2 W_2^T)^{-1}$, $P_3^* = (W_3\Omega_3 W_3^T)^{-1}$; and their fused covariance $P^* = (W_1\Omega_1 W_1^T + W_2\Omega_2 W_2^T + W_3\Omega_3 W_3^T)^{-1}$.

The main drawback with the proposed algorithm is that it requires solving two SDPs, which can be computational demanding, especially for large-scale systems. However, unless there are significant changes in the covariance matrices from the local estimates (due to, e.g., high nonlinearities), updating the weights at every iteration should not be necessary. Instead, the weights can be kept constant until the operating conditions in the process changes.

10.4 Case study: Lorenz attractor

Let's consider three estimators, that tries to estimate the states of the Lorenz attractor, given by the following state-space model:

$$\dot{x}_1 = -10(x_1 - x_2) \tag{10.37}$$

$$\dot{x}_2 = x_1(28 - x_3) - x_2 \tag{10.38}$$

$$\dot{x}_3 = x_1 x_2 - \frac{8}{3}x_3. \tag{10.39}$$

Each estimator uses the locally available measurements y_i, together with an unscented Kalman filter (UKF) [68], where the measurements are given by:

$$y_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, y_2 = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}, y_3 = \begin{bmatrix} x_3 \\ x_1 \end{bmatrix}.$$

The covariance matrices for the process and measurement noise are assumed to be known and are for $i = 1, 2, 3$:

$$Q_{w,i} = \begin{bmatrix} 0.5^2 & 0 & 0 \\ 0 & 0.5^2 & 0 \\ 0 & 0 & 0.5^2 \end{bmatrix}, \quad R_{v,i} = \begin{bmatrix} 0.01^2 & 0 \\ 0 & 0.1^2 \end{bmatrix}$$

The Lorenz attractor is simulated for 1000 samples with a step size of 0.01 using a 4th order Runge-Kutta numerical integration. To improve the result, the estimated states from the local estimators are fused, using a hierarchical structure as in Figure 10.1, but with all states overlapping. The cross-correlations between the estimates are assumed to be unknown, and thus, the estimation error for the fast CI [45], the EI, and the proposed method are compared. The root mean squared error (RMSE) for the respective estimator and the three fusion methods can be seen in Table 10.1, where the subscripts of "EI" denote the sequential order the estimates have been fused.

Table 10.1: RMSE for the Lorenz attractor

	RMSE x_1	RMSE x_2	RMSE x_3
Local Estimator 1	**0.0101**	0.0969	1.7172
Local Estimator 2	1.1101	**0.0101**	0.0992
Local Estimator 3	0.0968	1.4670	**0.0103**
Fast CI	0.1315	0.1036	0.0905
EI_{123}	0.0108	**0.0101**	0.0109
EI_{132}	**0.0101**	0.0106	0.0107
EI_{321}	0.0110	0.0112	**0.0103**
Proposed fusion method (updated W_i)	**0.0101**	**0.0101**	**0.0103**
Proposed fusion method (static W_i)	0.0101	0.0101	0.0104

As seen in Table 10.1, all three fusion methods give an overall reduction in the estimation error compared to the local estimates. However, the proposed method gives a better result compared to the two other methods, where there is little improvement when updating the weights at every iteration compared to using the same (static) weights. It can also be seen that the estimation error for the EI method depends on the order which the local estimates have been fused in.

10.5 Case study: CSTR and flash separator

A chemical process consisting of two continuous stirred tank reactors (CSTRs) and a flash separator connected in series is considered as seen in Figure 10.4. Pure material A is fed into the two CSTRs, in which two reactions occur, i.e., $A \to B$ and $B \to C$. The outlet of the second CSTR is fed into the flash separator at a flow rate F_2. The overhead of the separator is condensed and passed to a downstream unit at flow rate F_r and the bottom product stream is removed at flow rate F_3. Each tank is equipped with a jacket to heat or cool the tank. The dynamic model obtained via mass and energy balances are the same as in [135], but the flows and thus the volumes are assumed to be constant, resulting in:

CSTR 1 (Estimator 1)

$$\dot{x}_{A1} = \frac{F_{10}x_{A10} + F_r x_{Ar} - F_1 x_{A1}}{\rho_{r_1} V_{r_1}} - k_1 e^{\frac{-E_1}{RT_1}} x_{A1} \tag{10.40}$$

$$\dot{x}_{B1} = \frac{F_{10}(1 - x_{A10}) + F_r x_{Br} - F_1 x_{B1}}{\rho_{r_1} V_{r_1}} + k_1 e^{\frac{-E_1}{RT_1}} x_{A1} - k_2 e^{\frac{-E_2}{RT_1}} x_{B1} \tag{10.41}$$

$$\dot{T}_1 = \frac{F_{10}T_{10} + F_r T_3 - F_1 T_1}{\rho_{r_1} V_{r_1}} - \frac{\Delta H_1}{c_p} k_1 e^{\frac{-E_1}{RT_1}} x_{A1} - \frac{\Delta H_2}{c_p} k_2 e^{\frac{-E_2}{RT_1}} x_{B1} + \frac{Q_{r_1}}{\rho_{r_1} c_p V_{r_1}} \tag{10.42}$$

CSTR 2 (Estimator 2)

$$\dot{x}_{A2} = \frac{F_{20}x_{A20} + F_1 x_{A1} - F_2 x_{A2}}{\rho_{r_2} V_{r_2}} - k_1 e^{\frac{-E_1}{RT_2}} x_{A2} \tag{10.43}$$

$$\dot{x}_{B2} = \frac{F_{20}(1 - x_{A20}) + F_1 x_{B1} - F_2 x_{B2}}{\rho_{r_2} V_{r_2}} + k_1 e^{\frac{-E_1}{RT_2}} x_{A2} - k_2 e^{\frac{-E_2}{RT_2}} x_{B2} \tag{10.44}$$

$$\dot{T}_2 = \frac{F_{20}T_{20} + F_1 T_1 - F_2 T_2}{\rho_{r_2} V_{r_2}} - \frac{\Delta H_1}{c_p} k_1 e^{\frac{-E_1}{RT_2}} x_{A2} - \frac{\Delta H_2}{c_p} k_2 e^{\frac{-E_2}{RT_2}} x_{B2} + \frac{Q_{r_2}}{\rho_{r_2} c_p V_{r_2}} \tag{10.45}$$

Separator (Estimator 3)

$$\dot{x}_{A3} = \frac{F_2 x_{A2} - (F_p + F_r) x_{Ar} - F_3 x_{A3}}{\rho_s V_s} \tag{10.46}$$

$$\dot{x}_{B3} = \frac{F_2 x_{B2} - (F_p + F_r) x_{Br} - F_3 x_{B3}}{\rho_s V_s} \tag{10.47}$$

$$\dot{T}_3 = \frac{F_2 T_2 - (F_p + F_r + F_3) T_3}{\rho_s V_s} + \frac{Q_s}{\rho_s c_p V_s} \tag{10.48}$$

with the algebraic equations:

$$x_{Ar} = \frac{\alpha_A x_{A3}}{\bar{x}_3}, \quad x_{Br} = \frac{\alpha_B x_{B3}}{\bar{x}_3}, \tag{10.49}$$

$$\bar{x}_3 = \alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C (1 - x_{A3} - x_{B3}), \tag{10.50}$$

and the parameters given in Table 10.2.

The process is split into three local estimators; one for each of the reactors, and one for the separator. The available measurements y_i, and inputs u_i for the estimators are:

$$y_1 = \begin{bmatrix} x_{A3} \\ x_{B1} \\ T_1 \end{bmatrix}, y_2 = \begin{bmatrix} x_{A1} \\ x_{B2} \\ T_2 \end{bmatrix}, y_3 = \begin{bmatrix} x_{A2} \\ x_{B3} \\ T_3 \end{bmatrix}, \begin{matrix} u_1 = x_{A10} \\ u_2 = x_{A20} \end{matrix}$$

The process and measurement noise are given by:

$$Q_{w,i} = \begin{bmatrix} 0.005^2 & 0 & 0 \\ 0 & 0.005^2 & 0 \\ 0 & 0 & 0.1^2 \end{bmatrix}, \quad R_{v,i} = \begin{bmatrix} 0.0002^2 & 0 & 0 \\ 0 & 0.0001^2 & 0 \\ 0 & 0 & 0.1^2 \end{bmatrix}$$

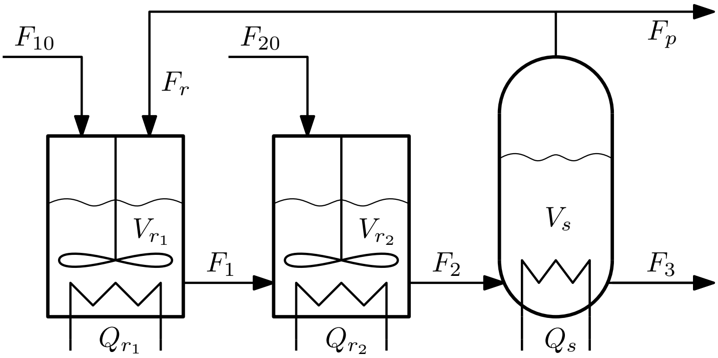


Figure 10.4: Diagram of two CSTR and a flash separator, adopted from [135].

for $i = 1, 2, 3$. All three models are augmented with three states as in (10.7) with x_{Ai}, x_{Bi}, and T_i from the previous estimator. The parameters and the steady-state values are shown in Table 10.2. The process is simulated for 100s with a sampling time of 0.5s, where the inputs u_1 and u_2 are varying with two different sinusoidal signals.

Table 10.2: Parameters for the 2 CSTR and flash separator

$x_{A1} = 0.923$ wt(%)	$T_1 = 315.1$ K	$F_1 = 74.5$ kg/s
$x_{B1} = 0.074$ wt(%)	$T_2 = 315.2$ K	$F_{10} = 8.3$ kg/s
$x_{A2} = 0.919$ wt(%)	$T_3 = 314.9$ K	$F_2 = 75.0$ kg/s
$x_{B2} = 0.081$ wt(%)	$Q_{r_1} = 10.0$ kJ/s	$F_{20} = 0.5$ kg/s
$x_{A3} = 0.806$ wt(%)	$Q_{r_2} = 10.0$ kJ/s	$F_3 = 8.0$ kg/s
$x_{B3} = 0.184$ wt(%)	$Q_s = 10.0$ kJ/s	$F_r = 66.2$ kg/s
$x_{A10} = 0.90$ wt(%)	$V_{r_1} = 89.4$ m^3	$F_p = 0.8$ kg/s
$x_{A20} = 0.80$ wt(%)	$V_{r_2} = 90.0$ m^3	$c_p = 25$ kJ/kg K
$E_1/R = -100$ K	$V_s = 3.27$ m^3	$\alpha_A = 3.5$
$E_2/R = -150$ K	$k_1 = 0.020$ 1/s	$\alpha_B = 1.1$
$\Delta H_1 = -40$ kJ/kg	$k_2 = 0.018$ 1/s	$\alpha_C = 0.5$
$\Delta H_2 = -50$ kJ/kg	$T_{10}, T_{20} = 315$ K	$\rho_{r_1}, \rho_{r_1}, \rho_s = 0.15$ kg/m^3

The hierarchical decentralized state estimator was implemented for the process, where a UKF was used for all the local estimators. The same scaling weights were used during the whole simulation, which had been computed offline using the proposed method. The RMSE for the compositions can be seen in Figure 10.5, where the result is compared to using (10.10), and (10.11) with the scalar weights, $\omega_1 = \omega_2 = \omega_3 = 1/3$.

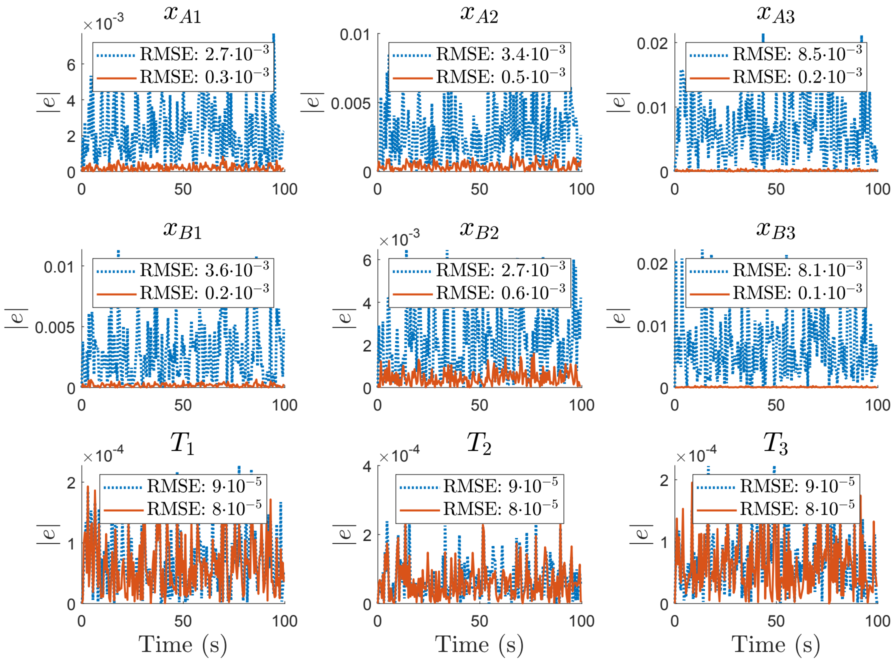


Figure 10.5: RMSE for the 2 CSTR and flash separator process. The dotted blue line is the absolute estimation error when using the CI method with $\omega_1 = \omega_2 = \omega_3 = 1/3$, and the solid red line is the absolute estimation error when using the proposed method.

10.6 Conclusion

A method for fusing multiple and partially overlapping state estimates with unknown cross-correlation has been proposed. The resulting algorithm calculates weight matrices that scales the local estimates, such when fused together gives a more accurate estimate. The fusion method was evaluated through simulations of two different case studies. The first case study consisted of the Lorenz attractor, which involved fusing three local estimates that were trying to estimate the same state vector. The results were compared to using EI and CI and showed a reduction in the estimation error for the proposed algorithm. Furthermore, the accuracy of EI depends on which sequential order the local estimates are fused in, which is not an issue for Algorithm 9 since the weights are computed simultaneously.

In the second example, a decentralized state estimation scheme was implemented on a process that consisted of two reactors and a flash separator. Three different local estimators were implemented that only estimated a subset of the global state vector. The different local estimates where fused together using the proposed algorithm that resulted in a global state vector with a significant reduction in the estimation error compared to CI. Here, only CI was used to compare the results, since EI is not well-suited for the desired hierarchical state estimation structure that includes partially overlapping state vectors.

A centralized estimator should be able to provide more accurate estimates than the proposed decentralized state estimation method since the correlations between the subsystems are unknown. However, the decentralized approach is more scalable, which makes it more favorable for large-scale systems and can significantly reduce the estimation error compared to using only local estimators.

Part V

Closing remarks

Chapter 11

Conclusion

After a brief outline of the thesis, together with some introductory information related to control structure design in Part I, the main contributions for this thesis are given in Part II, III, and IV.

11.1 Measurement selection in self-optimizing control

Part II starts by providing the reader with the fundamentals of self-optimizing control, which includes the null space method and the exact local method. These two methods can be used to find the optimal measurement combinations as CVs when only the steady-state behavior is considered. Next, a brief summary was given of the existing approaches that search for the optimal subset of measurements when a measurement combination is used as the self-optimizing control variable. The existing methods involved solving an MIQP in [149] or using the branch and bound algorithm that was given in [75]. An alternative to the existing methods was also proposed, which involved using the re-weighted l_1 norm as a convex relaxation to the otherwise combinatorial optimization problem. The proposed algorithm was evaluated on a binary distillation column and showed that the re-weighted l_1 algorithm, gave a slightly higher loss compared to using the existing methods. However, by replacing the original combinatorial problem into a convex approximation, it makes the computational demands less sensitive to the size of the problem. Thus, for large scale-scale problems, it can be used to reduce initial set of measurements into a smaller subset, from which one of the existing methods can be used to find the optimal set of measurements. In addition, the re-weighted l_1 norm is a lot more flexible and easier to include in other optimization problems, as was later demonstrated in Part III of the thesis.

The main contribution for Part II was given in Chapter 5, where the issue of incorporating structural constraints on the measurement combination was investigated. Besides finding the optimal subset of measurements, it is often desirable to impose a certain structure on the measurement selection matrix, e.g., to have the input control the measurements that correspond to the same part of the process. However, this results in a nonlinear optimization problem that is difficult to solve. Here, an ADMM algorithm was proposed, which was shown to be capable of finding good solutions to the problem with little computational effort. The ADMM algorithm was also combined with a branch and bound algorithm to find the optimal set measurements when there are structural constraints imposed on the measurement selection matrix. The resulting algorithm was evaluated on several case studies and showed that it was able to find better solutions with a lower loss compared to the other existing methods in [147], [148]. Furthermore, the proposed algorithm seemed capable of computing solutions that were on par with the commercial MINLP solver BARON [78], but with a significant reduction in the computational time.

11.2 Accounting for dynamics in self-optimizing control

The main focus of the thesis is presented in Part III, where the dynamic effects of using measurement combinations as CVs have been considered. First, the impact measurement combinations have on the system poles, zeros, and their directions are discussed and illustrated through an example. This discussion highlights that both the measurement combination and feedback controllers will impact the behavior of the resulting closed-loop system. Therefore, it is suggested that they should ideally, be computed simultaneously, where the measurement selection matrix can be seen as additional tuning parameters that will influence both the dynamic and steady-state performance of the process.

The measurement combination and feedback controllers can be obtained simultaneously by solving a static output feedback (SOF) control problem. Therefore, different approaches for formulating the SOF optimization problem were presented that are based on the two-step procedure given in [109], [37], [94], [2], and [98]. In addition, different ways of augmenting the system matrices where proposed, so that the resulting SOF gain can be transformed into a control structure that consists of decentralized PI controllers that are controlling measurement combinations as the CVs.

In the final chapter for Part III, several algorithms are proposed for simultaneously computing measurement combinations as the self-optimizing control variable together with the tuning parameters for the decentralized PI controllers. These algorithms aim to find the optimal trade-off between economic steady-state and dynamic performance for the resulting control structure. In addition, they can also incorporate the re-weighted l_1 norm to penalize the number of measurements used, or be combined with the ADMM algorithm to impose structural constraints on the measurement combination. The proposed algorithms were evaluated on several case studies and demonstrated that good results could be found both with respect to the (near) optimality of the steady-state solution, and the control performance of the resulting closed-loop system. In particular, the case studies for the divided-wall distillation columns where of great interest, since it has been stated in the literature that they need a model predictive controller (MPC) to achieve good dynamic performance [1]. However, simulations show that a control structure containing decentralized PI controllers and can provide comparable results to an MPC, as long as the CVs and the tuning parameters for the PI controllers have been chosen appropriately.

11.3 Hierarchical decentralized state estimation

The final contribution for this thesis is presented in Part IV, where a fusion method is proposed that is capable of fusing together multiple state vectors that are partially overlapping each other. The resulting fusion algorithm computes different weight matrices that are used to emphasize the more accurate state estimates. The proposed algorithm was evaluated on two different case studies and showed a reduction in the estimation error compared to other existing fusion methods.

Chapter 12

Future work

In this chapter, some of the potential future research direction will be discussed for the three main contributions.

12.1 Measurement selection in self-optimizing control

The ADMM algorithm used for incorporating structural constraints on the measurement combinations was the primary contribution in this part of the thesis. The initial results for the algorithm were promising, but there are still several areas which could be worth further research.

The convergence of the algorithm is dependent on the initial values of ρ, and Q, where especially the value of ρ plays a critical role. The algorithm seems to converge as long as ρ is chosen sufficiently large, however, a proof of its convergence is still lacking. If a convergence proof was formulated, such that a value of ρ that guarantees convergence could be determined initially, then there would be no need to use the proposed procedure for adapting the penalty parameter ρ. This could potentially further reduce the computational time required to find a solution.

The effectiveness of the ADMM algorithm was evaluated on different case studies where it outperformed the other existing methods [147], [148] and the commercial solver BARON [78]. However, whereas the BARON solver is capable of giving the globally optimal solution, the final result for the ADMM algorithm will depend on how it is initialized, which was also demonstrated in Section 5.3.1. Therefore, the ADMM algorithm needs to be evaluated on more case studies, with a focus on large-scale systems to further study its speed and accuracy.

12.2 Accounting for dynamics in self-optimizing control

The proposed method produces a measurement combination and the tuning parameters for decentralized PI controllers. However, it is very likely that better performance can be achieved if more advanced controller formulations are used, e.g., by using sparse or centralized PI controllers [116], [117]. Considering that the proposed method is a quite advanced technique for designing the controllers (and selecting measurement combination), a strong case can be made for also implementing some more advanced and higher order controllers. One important requirement is that the controllers incorporate some integral action to ensure that the self-optimizing control criterion is met at steady-state. Therefore, it may be worth investigating the potential benefits of using more advanced control structures, where the desired controllers don't necessarily have to be limited to PI controllers.

Another potential extension to the presented work would be to include some additional robustness for the resulting control structure. If there exists a range of parameters that can be represented by linear models in a polytope, then it is possible to create what is commonly known in the literature as a polytopic model [98]. Using the polytopic model, it should be possible to extend the proposed algorithms such that the designed controllers and CVs guarantee a stable closed-loop system for the entire parameter space. It is possible to further improve the dynamic performance by designing the controllers such that their tunings vary for different points in the polytopic model. However, more interestingly, the same can also be done for the CVs that consists of measurement combinations.

The self-optimizing control variables are typically, obtained using local models, which gives suboptimal performance when the process drifts away from its nominal operating point. Letting the CVs vary can, therefore, reduce the loss caused by nonlinearities in the process. It may also be possible to automatically update the measurement combinations when there are disturbances present that cause a change in the active constraints for the optimal operation. This would allow the CVs and feedback controllers to change such that the plant could operate close to optimal for multiple active constraint regions.

12.3 Hierarchical decentralized state estimation

The proposed fusion method involves solving two SDPs, which can be computationally demanding, especially for large-scale systems. Therefore, it would be worth investigating whether it is possible to replace the two SDP problem with a simpler procedure that could achieve the same or a similar result while reducing the computational load.

The fused covariance matrix consists of the maximum ellipsoid inside the region of the intersection for the covariance matrices that was computed by the local estimators. As a consequence, the fused estimate is not consistent unless the local estimates are only weakly correlated. Therefore, similar to covariance intersection (CI), it would be preferable if the resulting covariance matrix instead was given by the minimum overestimation of the intersection region between the local covariances.

If the computational demands could be reduced and consistency could be guaranteed, then it would make the resulting fusion method a lot more widely applicable for a vast number of different research areas.

References

[1] T. Adrian, H. Schoenmakers, and M. Boll. Model predictive control of integrated unit operations: Control of a divided wall column. *Chemical Engineering and Processing: Process Intensification*, 43(3):347–355, 2004.

[2] C. M. Agulhari, R. C. L. F. Oliveira, and P. L. D. Peres. Robust H_∞ static output-feedback design for time-invariant discrete-time polytopic systems from parameter-dependent state-feedback gains. *Proceedings of the 2010 American Control Conference*, pages 4677–4682, 2010.

[3] F. Allgöwer, R. Findeisen, and Z. K. Nagy. Nonlinear model predictive control: From theory to application. *Journal of the Chinese Institute of Chemical Engineers*, 35(3):299–315, 2004.

[4] V. Alstad. *Studies on selection of controlled variables.* PhD thesis, Norwegian University of Science and Technology, 2005.

[5] V. Alstad and S. Skogestad. Null space method for selecting optimal measurement combinations as controlled variables. *Industrial & Engineering Chemistry Research*, 46(3):846–853, 2007.

[6] V. Alstad, S. Skogestad, and E. S. Hori. Optimal measurement combinations as controlled variables. *Journal of Process Control*, 19(1):138–148, 2009.

[7] A. Argha, S. W. Su, A. Savkin, and B. Celler. A framework for optimal actuator/sensor selection in a control system. *International Journal of Control.*

[8] K. B. Ariyur and M. Krstić. *Real-Time Optimization by Extremum Seeking Control.* John Wiley & Sons, Inc., Hoboken, New Jersey, 2003.

[9] M. C. Arranz, W. Birk, and G. Nikolakopoulos. A survey on control configuration selection and new challenges in relation to wireless sensor and actuator networks. *Proceedings of the 20th IFAC World Congress*, pages 8810–8825, 2017.

[10] D. Arzelier and D. Peaucelle. An iterative method for mixed H_2/H_∞ synthesis via static output-feedback. *Proceedings of the IEEE Conference on Decision and Control*, pages 3464–3469, 2002.

[11] D. Arzelier, E. N. Gryazina, D. Peaucelle, and B. T. Polyak. Mixed LMI/randomized methods for static output feedback control design. *Proceedings of the 2010 American Control Conference (ACC)*, pages 4683–4688, 2010.

[12] E. B. Aske, S. Skogestad, and S. Strand. Throughput maximization by improved bottleneck control. *Proceedings of the 8th International IFAC Symposium on Dynamics and Control of Process Systems*, pages 63–68, 2007.

[13] K. J. Åström and T. Hägglund. The future of PID control. *Control Engineering Practice*, 9(11):1163–1175, 2001.

[14] M. A. Bakr and S. Lee. Distributed multisensor data fusion under unknown correlation and data inconsistency. *Sensors*, 17(11), 2017.

[15] Y. Bar-Shalom and L. Campo. The effect of the common process noise on the two-sensor fused-track covariance. *IEEE Transactions on Aerospace and Electronic Systems*, 22(6):803–805, 1986.

[16] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*. John Wiley & Sons, Inc, New York, 2001.

[17] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. MPS-SIAM Series on Optimization, Philadelphia, 2001.

[18] A. R. Benaskeur. Consistent fusion of correlated data sources. *Proceedings of the 28th Annual Conference of the IEEE Industrial Electronics Society*, pages 2652–2656, 2002.

[19] W. Birk and A. Medvedev. A note on gramian-based interaction measures. *Proceedings of the 2003 European Control Conference (ECC)*, pages 2625–2630, 2003.

[20] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, U.K., 2004.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[22] E. H. Bristol. On a new measure of interactions for multivariable process control. *IEEE Transactions on Automatic Control*, 11(1):133–134, 1966.

[23] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905, 2008.

[24] Y. Cao. Bidirectional branch and bound minimum singular value solver: V2, 11/11/2007. URL http://www.mathworks.com/matlabcentral/fileexchange/17480.

[25] Y. Cao. Bidirectional branch and bound solvers for worst case loss minimization, 16/09/2009. URL http://www.mathworks.com/matlabcentral/fileexchange/22632.

[26] Y. Cao. Bidirectional branch and bound for average loss minimization, 23/02/2010. URL http://www.mathworks.com/matlabcentral/fileexchange/25870.

[27] Y. Cao and V. Kariwala. Bidirectional branch and bound for controlled variable selection: Part I. Principles and minimum singular value criterion. *Computers & Chemical Engineering*, 32(10):2306–2319, 2008.

[28] K. C. Chang, C.-Y. Chong, and S. Mori. Analytical and computational evaluation of scalable distributed fusion algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 46(4):2022–2034, 2010.

[29] L. Chen, P. O. Arambel, and R. K. Mehra. Fusion under unknown correlation - covariance intersection as a special case. *Proceedings of the 5th International Conference on Information Fusion*, pages 905–912, 2002.

[30] A. Conley and M. E. Salgado. Gramian based interaction measure. *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 5020–5022, 2000.

[31] M. L. Darby, M. Nikolaou, J. Jones, and D. Nicholson. RTO: An overview and assessment of current practice. *Journal of Process Control*, 21(6):874–884, 2002.

[32] M. C. De Oliveira, J. C. Geromel, and J. Bernussou. Extended H_2 and H_∞ norm characterizations and controller parametrizations for discrete-time systems. *International Journal of Control*, 75(9):666–679, 2002.

[33] Z. Deng, P. Zhang, W. Qi, J. Liu, and Y. Gao. Sequential covariance intersection fusion Kalman filter. *Information Sciences*, 189:293–309, 2012.

[34] N. K. Dhingra, M. R. Jovanović, and Z.-Q. Luo. An ADMM algorithm for optimal sensor and actuator selection. *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 4039–4044, 2014.

[35] D. Dochain, M. Perrier, and M. Guay. Extremum seeking control and its application to process and reaction systems: A survey. *Mathematics and Computers in Simulation*, 82(3):369–380, 2011.

[36] J. J. Downs and S. Skogestad. An industrial and academic perspective on plantwide control. *Annual Reviews in Control*, 35(1):99–110, 2011.

[37] Y. Ebihara, D. Peaucelle, and D. Arzelier. *S-variable approach to LMI-based robust control.* Springer-Verlag, London, U.K., 2015.

[38] J. Eckstein and W. Yao. Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives. *Pacific Journal of Optimization*, 11(4):619–644, 2015.

[39] M. Ellis, H. Durand, and P. D. Christofides. A tutorial review of economic model predictive control methods. *Journal of Process Control*, 24(8):1156–1178, 2014.

[40] D. M. Falcao, F. F. Wu, and L. Murphy. Parallel and distributed state estimation. *IEEE Transactions on Power Systems*, 10(2):724–730, 1995.

[41] M. Fardad, F. Lin, and M. R. Jovanović. Sparsity-promoting optimal control for a class of distributed systems. *Proceedings of the 2011 American Control Conference (ACC)*, pages 2050–2055, 2011.

[42] M. Farina, G. Ferrari-Trecate, and R. Scattolini. Moving-horizon partition-based state estimation of large-scale systems. *Automatica*, 46(5):910–918, 2010.

[43] C. A. Floudas. *Deterministic Global Optimization: Theory, Methods and Applications.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[44] A. Foss. Critique of chemical process control theory. *AIChE Journal*, 19: 209–214, 1973.

[45] D. Fränken and A. Hüpper. Improved fast covariance intersection for distributed data fusion. *Proceedings of the 7th International Conference on Information Fusion (FUSION)*, pages 154–160, 2005.

[46] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[47] P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4(4):421–448, 1994.

[48] Z.-L. Gaing. A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Transactions on Energy Conversion*, 19(2):384–391, 2004.

[49] G. Galvan, M. Lapucci, T. Levato, and M. Sciandrone. An alternating augmented Lagrangian method for constrained nonconvex optimization. *Optimization Methods and Software*, 2019.

[50] W. Gao, D. Goldfarb, and F. E. Curtis. ADMM for multiaffine constrained optimization. *arXiv: 1802.09592*, 2018.

[51] J. E. Gentle. *Matrix Algebra: Theory, Computations, and Applications in Statistics.* Springer Science + Business Media, New York, 2007.

[52] M. S. Govatsmark and S. Skogestad. Selection of controlled variables and robust setpoints. *Industrial & Engineering Chemistry Research*, 44(7):2207–2217, 2005.

[53] S. Gros, B. Srinivasan, and D. Bonvin. Optimizing control based on output feedback. *Computers & Chemical Engineering*, 33(1):191–198, 2009.

[54] D. Hajinezhad and Q. Shi. Alternating direction method of multipliers for a class of nonconvex bilinear optimization: convergence analysis and applications. *Journal of Global Optimization*, 70(1):261–288, 2018.

[55] I. J. Halvorsen, M. Serra, and S. Skogestad. Evaluation of self-optimizing control structures for an integrated petlyuk distillation column. *Hungarian Journal of Industrial Chemistry*, 1(28):11–15, 2000.

[56] I. J. Halvorsen, S. Skogestad, J. C. Morud, and V. Alstad. Optimal selection of controlled variables. *Industrial & Engineering Chemistry Research*, 42(14): 3273–3284, 2003.

[57] H. R. Hashemipour, S. Roy, and A. J. Laub. Decentralized structures for parallel Kalman filtering. *IEEE Transactions on Automatic Control*, 33(1): 88–94, 1988.

[58] B. S. He, H. Yang, and S. L. Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and Applications*, 106(2):337–356, 2000.

[59] S. Heldt. Dealing with structural constraints in self-optimizing control engineering. *Journal of Process Control*, 20(9):1049–1058, 2010.

[60] M. Hong, Z.-Q. Luo, and M. Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

[61] J. N. Hooker and M. A. Osorio. Mixed logical-linear programming. *Discrete Applied Mathematics*, 96-97:395–442, 1999.

[62] E. S. Hori, S. Skogestad, and V. Alstad. Perfect steady-state indirect control. *Industrial & Engineering Chemistry Research*, 44(4):863–867, 2005.

[63] IBM. IBM ILOG CPLEX Optimization Studio V12.9.0, 2019. URL http://www.cplex.com.

[64] J. Jäschke and S. Skogestad. NCO tracking and self-optimizing control in the context of real-time optimization. *Journal of Process Control*, 21(10): 1407–1416, 2011.

[65] J. Jäschke, Y. Cao, and V. Kariwala. Self-optimizing control – A survey. *Annual Reviews in Control*, 43:199–223, 2017.

[66] A. Jayachitra and R. Vinodha. Genetic algorithm based PID controller tuning approach for continuous stirred tank reactor. *Advances in Artificial Intelligence*, 2014, 2014.

[67] S. Julier and J. K. Uhlmann. General decentralized data fusion with covariance intersection (CI). In D. Hall and J. Llinas., editors, *Handbook of Multisensor Data Fusion*, chapter 12. CRC Presss, Boca Raton, Florida, 2001.

[68] S. J. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[69] S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. *Proceedings of the 1997 American Control Conference (ACC)*, pages 2369–2373, 1997.

[70] J. V. Kadam, W. Marquardt, M. Schlegel, T. Backx, O. H. Bosgra, P.-J. Brouwer, G. Dünnebier, D. van Hessem, A. Tiagounov, and S. de Wolf. Towards integrated dynamic real-time optimization and control of industrial processes. *Proceedings Foundations of Computer-Aided Process Operations (FOCAPO2003)*, pages 593–596, 2003.

[71] G. Kaibel. Distillation columns with vertical partitions. *Chemical Engineering & Technology*, 10(1):92–98, 1987.

[72] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury. Information weighted consensus filters and their application in distributed camera networks. *IEEE Transactions on Automatic Control*, 58(12):3112–3125, 2013.

[73] A. Karimi, M. Kunze, and R. Longchamp. Robust controller design by linear programming with application to a double-axis positioning system. *Control Engineering Practice*, 15(2):197–208, 2007.

[74] V. Kariwala and Y. Cao. Bidirectional branch and bound for controlled variable selection. Part II: Exact local method for self-optimizing control. *Computers & Chemical Engineering*, 33(8):1402–1414, 2009.

[75] V. Kariwala and Y. Cao. Bidirectional branch and bound for controlled variable selection Part III: Local average loss minimization. *IEEE Transactions on Industrial Informatics*, 6(1):54–61, 2010.

[76] V. Kariwala, Y. Cao, and S. Janardhanan. Local self-optimizing control with average loss minimization. *Industrial & Engineering Chemistry Research*, 47 (4):1150–1158, 2008.

[77] B. D. Keating and A. Alleyne. Combining self-optimizing control and extremum seeking for online optimization with application to vapor compression cycles. *Proceedings of the 2016 American Control Conference (ACC)*, pages 6085–6090, 2016.

[78] M. R. Kilinç and N. V. Sahinidis. Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON. *Optimization Methods and Software*, 33(3):540–562, 2018.

[79] M. Krstić and H.-H. Wang. Stability of extremum seeking feedback for general dynamic systems. *Automatica*, 36(4):595–601, 2000.

[80] M. Kvernland. Model predictive control of a Kaibel distillation column. Master's thesis, Norwegian University of Science and Technology, 2009.

[81] M. Kvernland, I. J. Halvorsen, and S. Skogestad. Model predictive control of a Kaibel distillation column. *Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems (DYCOPS 2010)*, pages 553–558, 2010.

[82] T. Larsson and S. Skogestad. Plantwide control - A review and a new design procedure. *Modeling, Identification and Control*, 21(4):209–240, 2000.

[83] T. Larsson, K. Hestetun, E. Hovland, and S. Skogestad. Self-optimizing control of a large-scale plant: The Tennessee Eastman process. *Industrial & Engineering Chemistry Research*, 40(22):4889–4901, 2001.

[84] T. Lastusilta, M. R. Bussieck, and T. Westerlund. Comparison of some high-performance MINLP solvers. *Chemical Engineering Transactions*, 11: 125–130, 2007.

[85] W. Li, Z. Wang, G. Wei, L. Ma, J. Hu, and D. Ding. A survey on multisensor fusion and consensus filtering for sensor networks. *Discrete Dynamics in Nature and Society*, 2015:1–12, 2015.

[86] W. Li, G. Wei, F. Han, and Y. Liu. Weighted average consensus-based unscented Kalman filtering. *IEEE Transactions on Cybernetics*, 46(2):558–567, 2016.

[87] J. S. Lim and Y. I. Lee. Design of discrete-time multivariable PID controllers via LMI approach. *Proceedings of the International Conference on Control, Automation and Systems*, pages 1867–1871, 2008.

[88] F. Lin, M. Fardad, and M. R. Jovanović. Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Transactions on Automatic Control*, 58(9):2426–2431, 2013.

[89] B. Liu, X. Zhan, and Z. H. Zhu. Multisensor parallel largest ellipsoid distributed data fusion with unknown cross-covariances. *Sensors*, 17(7), 2017.

[90] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. *Proceedings of the 2004 IEEE International Symposium on Computer Aided Control Systems Design (CACSD)*, pages 284–289, 2004.

[91] T.-T. Lu and S.-H. Shiou. Inverses of 2 × 2 block matrices. *Computers & Mathematics with Applications*, 43(1-2):119–129, 2002.

[92] J. M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, 2002.

[93] M. Mansouri, H. Nounou, and M. Nounou. State estimation of a chemical reactor process model - A comparative study. *Proceedings of the 10th International Multi-Conference on Systems, Signals & Devices (SSD13)*, pages 1–6, 2013.

[94] C. F. Morais, M. F. Braga, R. C. L. F. Oliveira, and P. L. D. Peres. H_∞ static output feedback control of discrete-time Markov jump linear systems with uncertain transition probability matrix. *Proceedings of the 2014 American Control Conference (ACC)*, pages 489–494, 2014.

[95] M. Morari and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes: Part II: Structural aspects and the synthesis of alternative feasible control schemes. *AIChE Journal*, 26(2):232–246, 1980.

[96] M. Morari and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes: Part III: Optimal selection of secondary measurements within the framework of state estimation in the presence of persistent unknown disturbances. *AIChE Journal*, 26(2):247–260, 1980.

[97] M. Morari, Y. Arkun, and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes. Part I: Formulation of the problem. process decomposition and the classification of the control task analysis of the optimizing control structures. *AIChE Journal*, 26(2):220–232, 1980.

[98] H. R. Moreira, R. C. L. F. Oliveira, and P. L. D. Peres. Robust H_2 static output feedback design starting from a parameter-dependent state feedback controller for time-invariant discrete-time polytopic system. *Optimal Control Applications and Methods*, 32(1):1–13, 2011.

[99] MOSEK ApS. The MOSEK optimization toolbox for MATLAB manual. version 8.0, 2018. URL http://docs.mosek.com/8.0/toolbox/index.html.

[100] D. Müller, B. Dercks, E. Nabati, M. Blazek, T. Eifert, J. Schallenberg, U. Piechottka, and K. Dadhe. Real-time optimization in the chemical processing industry. *Chemie Ingenieur Technik*, 89(11):1464–1470, 2017.

[101] M. R. Naysmith and P. L. Douglas. Review of real time optimization in the chemical process industries. *Developments in Chemical Engineering and Mineral Processing*, 3(2):67–87, 1995.

[102] A. Neumaier, O. Shcherbina, W. Huyer, and T. Vinkó. A comparison of complete global optimization solvers. *Mathematical Programming*, 103(2): 335–356, 2007.

[103] R. B. Newell and P. Lee. *Applied process control: A case study.* Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[104] B. Noack, J. Sijs, and U. D. Hanebeck. Fusion strategies for unequal state vectors in distributed Kalman filtering. *Proceedings of the 19th IFAC World Congress*, pages 3262–3267, 2014.

[105] B. Noack, J. Sijs, and U. D. Hanebeck. Algebraic analysis of data fusion with ellipsoidal intersection. *Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 365–370, 2016.

[106] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck. Decentralized data fusion with inverse covariance intersection. *Automatica*, 79:35–41, 2017.

[107] J. Nocedal and S. Wright. *Numerical optimization.* Springer Science + Business Media, New York, 2nd edition, 2006.

[108] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5492–5498, 2007.

[109] D. Peaucelle and D. Arzelier. An efficient numerical solution for H_2 static output feedback synthesis. *Proceedings of the 2001 European Control Conference (ECC)*, pages 3800–3805, 2001.

[110] F. Petlyuk, V. Platonov, and D. Slavinskii. Thermodynamically optimal method for separating multicomponent mixtures. *International Chemical Engineering*, 5(3):555–561, 1965.

[111] G. Pipeleers, B. Demeulenaere, J. Swevers, and L. Vandenberghe. Extended LMI characterizations for stability and performance of linear systems. *Systems & Control Letters*, 58(7):510–518, 2009.

[112] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control. *Control Engineering Practice*, 11(7):773–764, 2003.

[113] D. Rosinová and A. Kozáková. Robust decentralized PID controller design. In R. C. Panda, editor, *Introduction to PID Controllers – Theory, Tuning and Application to Frontier Areas*, chapter 6. IntechOpen, London, U.K., 2012.

[114] M. S. Sadabadi and D. Peaucelle. From static output feedback to structured robust static output feedback: A survey. *Annual Reviews in Control*, 42: 11–26, 2016.

[115] C. W. Scherer and S. Weiland. Linear matrix inequalities in control. In *Lecture Notes for the Dutch Institute of Systems and Control.* 2005.

[116] Y. Shen, W.-J. Cai, and S. Li. Multivariable process control: Decentralized, decoupling, or sparse? *Industrial & Engineering Chemistry Research*, 49(2): 761–771, 2010.

[117] Y. Shen, Y. Sun, and W. Xu. Centralized PI/PID controller design for multivariable processes. *Industrial & Engineering Chemistry Research*, 53(25): 10439–10447, 2014.

[118] J. Sijs and M. Lazar. State fusion with unknown correlation: Ellipsoidal intersection. *Automatica*, 48(8):1874–1878, 2012.

[119] J. Sijs, U. Hanebeck, and B. Noack. An empirical method to fuse partially overlapping state vectors for distributed state estimation. *Proceedings of the 2013 European Control Conference (ECC)*, pages 1615–1620, 2013.

[120] R. E. Skelton, T. Iwasaki, and K. Grigoriadis. *A Unified Approach to Linear Control Design.* CRC Presss, Boca Raton, Florida, 1997.

[121] S. Skogestad. Dynamics and control of distillation columns: A tutorial introduction. *Chemical Engineering Research and Design*, 75(6):539–562, 1997.

[122] S. Skogestad. Plantwide control: The search for the self-optimizing control structure. *Journal of Process Control*, 10(5):487–507, 2000.

[123] S. Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 13(4):291–309, 2003.

[124] S. Skogestad. Control structure design for complete chemical plants. *Computers & Chemical Engineering*, 28(1-2):219–234, 2004.

[125] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design.* John Wiley & Sons, Inc., Hoboken, New Jersey, 2nd edition, 2005.

[126] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki. Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. *Computers & chemical engineering*, 27(1):27–44, 2003.

[127] B. Srinivasan, L. T. Biegler, and D. Bonvin. Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function. *Computers & Chemical Engineering*, 32(3):572–579, 2008.

[128] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic. Consensus based overlapping decentralized estimator. *IEEE Transactions on Automatic Control*, 54(2):410–415, 2009.

[129] J. Strandberg. *Optimal Operation of Dividing Wall Columns.* PhD thesis, Norwegian University of Science and Technology, 2011.

[130] J. Straus, D. Krishnamoorthy, and S. Skogestad. On combining self-optimizing control and extremum-seeking control – Applied to an ammonia reactor case study. *Journal of Process Control*, 78:78–87, 2019.

[131] C. Sun and R. Dai. A customized ADMM for rank-constrained optimization problems with approximate formulations. *Proceedings of the 56th Annual Conference on Decision and Control (CDC)*, pages 3769–3774, 2017.

[132] V. L. Syrmos, C. T. Abdallah, P. Dorato, and K. Grigoriadis. Static output feedback – A survey. *Automatica*, 33(2):125–137, 1997.

[133] Y. Tan, W. H. Moase, C. Manzie, D. Nešić, and I. M. Y. Mareels. Extremum seeking from 1922 to 2010. *Proceedings of the 29th Chinese control conference*, pages 14–26, 2010.

[134] C. Triantafyllou and R. Smith. The design and optimisation of fully thermally coupled distillation columns. *Chemical Engineering Research and Design: Transactions of the Institution of Chemical Engineers*, 70(A2):118–132, 1992.

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

[136] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí. Scatter search and local NLP solvers: A multistart framework for global optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007.

[137] R. Vadigepalli and F. J. Doyle. A distributed state estimation and control algorithm for plantwide processes. *IEEE Transactions on Control Systems Technology*, 11(1):119–127, 2003.

[138] R. C. van Diggelen, A. A. Kiss, and A. W. Heemink. Comparison of control strategies for dividing-wall columns. *Industrial & Engineering Chemistry Research*, 49(1):288–307, 2010.

[139] J. G. VanAntwerp and R. D. Braatz. A tutorial on linear and bilinear matrix inequalities. *Journal of Process Control*, 10(4):363–385, 2000.

[140] A. N. Vargas, L. Acho, G. Pujol, R. C. L. F. Oliveira, J. B. do Val, and P. L. Peres. Robust H_2 static output feedback to control an automotive throttle valve. *Proceedings of the 2014 American Control Conference (ACC)*, pages 3141–3146, 2014.

[141] C. Venkateswarlu. Advances in monitoring and state estimation of bioreactors. *Journal of Scientific & Industrial Research*, 63(6):491–498, 2004.

[142] Y. Wang, W. Yin, and J. Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78:29–63, 2019.

[143] B. Wittenmark and M. E. Salgado. Hankel-norm based interaction measure for input-output pairing. *Proceedings of the 15th IFAC World Congress*, pages 429–434, 2002.

[144] B. Wohlberg. ADMM penalty parameter selection by residual balancing. *arXiv:1704.06209*, 2017.

[145] L. Ye, Y. Cao, and X. Yuan. Global approximation of self-optimizing controlled variables with average loss minimization. *Industrial & Engineering Chemistry Research*, 54:12040–12053, 2015.

[146] R. Yelchuru. *Quantitative methods for controlled variables selection.* PhD thesis, Norwegian University of Science and Technology, 2012.

[147] R. Yelchuru and S. Skogestad. Optimal controlled variable selection for individual process units in self optimizing control with MIQP formulation. *Proceedings of the 2011 American Control Conference (ACC)*, pages 342–347, 2011.

[148] R. Yelchuru and S. Skogestad. Optimal controlled variable selection with structural constraints using MIQP formulations. *Proceedings of the 18th IFAC World Congress*, pages 4977–4982, 2011.

[149] R. Yelchuru and S. Skogestad. Convex formulations for optimal selection of controlled variables and measurements using mixed integer quadratic programming. *Journal of Process Control*, 22(6):995–1007, 2012.

[150] X. Zhu and S. Wang. State estimation for nonlinear systems by using dynamic optimization. *Proceedings of the 2005 IEEE International Conference on Networking, Sensing and Control*, pages 918–922, 2005.

[151] Y. Zhu, Z. You, J. Zhao, K. Zhang, and X. R. Li. The optimality for the distributed Kalman filtering fusion with feedback. *Automatica*, 37(9):1489–1493, 2001.

[152] D. Zumoffen, M. Gonzalo, and M. Basualdo. Improvements on multivariable control strategies tested on the Petlyuk distillation column. *Chemical Engineering Science*, 93:292–306, 2013.

ISBN 978-82-326-4260-1 (printed version)
ISBN 978-82-326-4261-8 (electronic version)
ISSN 1503-8181
ITK Report: 2019-12-W

NTNU
Norwegian University of
Science and Technology

Doctoral theses at NTNU, 2019:332

NTNU

Jonatan Ralf Axel Klemets

Doctoral Thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

Doctoral theses at NTNU, 2019:332

Jonatan Ralf Axel Klemets

Topics in the optimal operation of process plants

NTNU
Norwegian University of
Science and Technology