Lucas Ferreira Bernardino

Optimal operation with changing control objectives

Doctoral thesis
for the degree of Ph.D. in Chemical Engineering

Trondheim, May 2024

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

NTNU

Norwegian University of Science and Technology

Doctoral thesis
for the degree of Ph.D. in Chemical Engineering

Faculty of Natural Sciences
Department of Chemical Engineering

ISBN 978-82-326-7984-3 (printed version)
ISBN 978-82-326-7983-6 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2024:196

Printed by NTNU-trykk

Abstract

This thesis proposes advances in control structure design to solve a steady-state optimization problem in real-time through feedback control. This is especially beneficial when a real-time optimization layer is not implemented, or its updates are too slow. However, when disturbances act upon the system and change the set of optimally active constraints, a different control structure becomes necessary to achieve optimal operation. This thesis addresses this challenge by proposing systematic methods for designing control structures that automatically switch control objectives when necessary. In Chapter 2, a decentralized control structure is proposed for processes with few constraints, and optimal operation is attained with PID controllers and min/max selectors. In Chapter 3, a region-based MPC is proposed, with different tracking objectives for each set of active constraints, and an active set detection block that switches the control objectives being tracked.

The ideal controlled variables for the unconstrained degrees of freedom are closely related to the steady-state cost gradient with respect to the inputs. In Chapter 4, a cost gradient estimation method is proposed based on the self-optimizing control methods for controlled variable design, resulting in a static linear combination of the available measurements. This estimated cost gradient can be used in a variety of methods for feedback optimizing control.

Chapter 5 presents a case with several constraints, to exemplify the challenges that decentralized region-based control faces in such cases. In Chapter 6, we further compare the region-based control approach to primal-dual feedback optimizing control, which is another approach to solving optimization problems through feedback.

Chapter 7 presents a problem of optimal inventory management of units in series subject to bottlenecks, which is an inherently dynamic problem of maximizing production. The bidirectional inventory control structure solves this problem optimally, as long as minimum flow constraints are not relevant. The control structure is extended to deal with minimum flow constraints by introducing additional feed-

back control loops, and simulations illustrate the trade-off between satisfying the different constraints.

In Chapter 8, motivated by the usefulness of the mathematical framework of self-optimizing control, an alternate formulation for the existing exact local method is presented, together with an explicit expression for the optimal measurement combination when rejecting disturbances is to be prioritized over measurement error, which is the idea of the nullspace method. The obtained expressions are valid regardless of the number of measurements or the presence of measurement error.

Overall, this thesis was able to provide guidelines on how to propose optimal control structures for cases when a steady-state model of the system is available for design, dealing with changes in active constraints during operation. Most methods described in this thesis require an estimate of the cost gradient, but this limitation was addressed by the proposed gradient estimation method. This method does not assume an accurate dynamic model is available, and it is not based on persistent input excitation, making it suitable for a wider class of applications. In a broader sense, this thesis illustrates how the use of logic elements for switching controlled variables is the exact solution for these optimal operation problems.

Acknowledgements

First, I would like to thank my supervisor, Professor Sigurd Skogestad, for introducing me to the topic of self-optimizing control, for all the discussions, and for the support during all these years. You shaped the way I now view the field of process control. It has been an honor learning from you. I would also like to thank my co-supervisors, Professor Johannes Jäschke and Professor Dinesh Krishnamoorthy, for the fruitful discussions throughout this work and for their support in the work we carried out together.

I thank the PhD evaluation committee, Professor Manfred Morari, Professor Nitin Kaistha, and Professor Idelfonso Nogueira, for taking the time to read and provide valuable feedback on this thesis. It is truly an honor to be evaluated by such a distinguished committee.

I thank all my colleagues from the PSE group of the Chemical Engineering Department: Adriana, Allyne, Cristina, David, Dinesh, Risvan, Saket, Adriaen, Carol, Caroline Nakama, Esmé, Evren, Halvor, José Otávio, Leonardo, Lucas Cammann, Mandar, Marius, Rafael, Simen, Timur, Zawadi, Robert, Niloufar, Andrea, Bahareh, Fabienne, Fernando, Haakon, Carine, Erbet, Igor, Patrick, and Vinicius (I hope I did not forget anyone!). It was a pleasure sharing the PhD experience with all of you, with many social activities to boot. I extend this thanks to the other colleagues from the department, from NTNU and outside, especially to Andressa, Aline, Joakim, Julian, Katharina, Maria, and Tamires, who made this experience in Trondheim all the more pleasant.

I especially thank Adriaen and Bicheng for starting L.A.B. with me, a source of great joy in the last few years. I extend this gratitude to Önder, who officially brought me back to music, and to Brian, Lorena, and everyone I played with and could share my music with.

I wholeheartedly thank Allyne, Carol (a.k.a. Bonita), Fabiana, and Otávio. You have been with me through thick and thin on this adventure far from Brazil ever since my master's studies (or before), and this bond only deepened during the PhD. You truly are the family I chose and built during this time, and I will carry you in my heart, always. Thank you for everything.

I thank my family in Brazil for their unwavering support in my pursuit of this degree. Even though we are far away, your contribution to me being where I am today is invaluable. I love you, and I hope to carry the best of you in me on my way forward.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The field of process systems engineering relies on process modeling, design, optimization, and control to improve the operational performance of process systems. Each of these core activities is fundamental for a good process operation, and process modeling is the initial step that informs control and optimization techniques. Some processes have an inherent connection between optimization and control, in the sense that dynamic aspects greatly influence economic performance, such as batch processes or processes with cycling. In such cases, there is a big investment in modeling to accurately describe the system, and the control and optimization tools needed to guarantee optimal operation must make use of all this information, usually in the form of dynamic optimization schemes (Ellis et al. 2014).

Most process systems implemented in practice, however, are designed to operate at a steady state, which greatly simplifies the efforts needed to guarantee optimal operation. Even though dynamic optimization solutions should also give good solutions to these problems, their complexity would lead to unnecessary investments in obtaining accurate models, which would not necessarily pay off with the resulting performance increase. In such cases, a decomposed, hierarchical approach would be sufficient to attain acceptably small economic losses.

The most common form of a hierarchical control framework is given by an optimization layer, a supervisory control layer, and a regulatory control layer, see Figure 1.1 (Skogestad 2000). The layers depend on a time scale separation between them for a good dynamic performance, such that the interaction between the layers is minimized. In this sense, a reasonable assumption in the design and tuning of these layers is that the lower layers are at steady state and that the upper layers have no effect on the process dynamics. The regulatory layer serves the pur-

pose of process stabilization, the optimization layer calculates setpoints based on a steady-state description of the system (consisting of a nonlinear process model, operational cost, and constraints), and the supervisory control layer acts as an intermediate between these layers, dealing with multivariable dynamics, constraints satisfaction, and helping on the implementation of the optimal policy. The supervisory layer has a great degree of flexibility in its implementation and functions, and for this reason, this thesis will focus on the analysis and design of this layer.

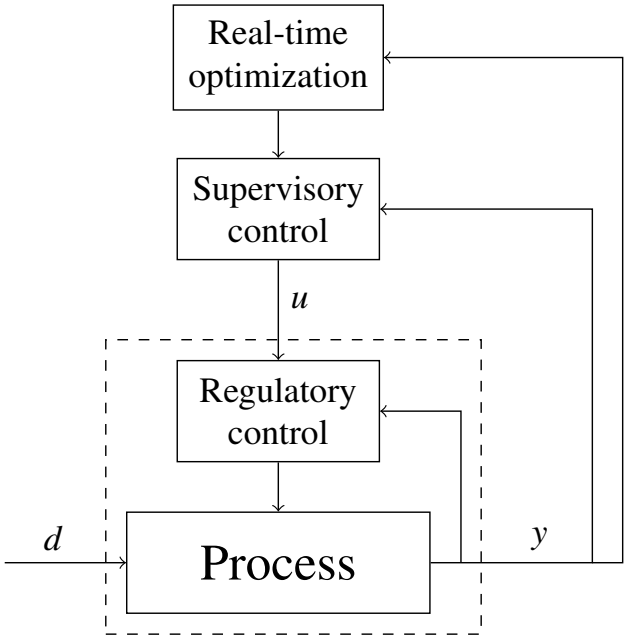


Figure 1.1: Typical hierarchy in process control implementations (Skogestad 2000). In this thesis, the focus is on the supervisory control layer.

The academic community has placed a great focus on model predictive control (MPC) as the main tool for supervisory control. Indeed, MPC is the most successful multivariable controller that can additionally deal with input and output constraints, nonlinear models, and predicted setpoint or disturbance changes in its formulation. However, there are still open issues when it comes to control of uncertain systems, whether this uncertainty comes from stochasticity or modeling errors, and these open issues are intimately related to the proper incorporation of feedback into MPC (Mayne 2014). In most cases of modeling errors, offset-free MPC can be successfully implemented with good closed-loop performance (Morari and Maeder 2012, Pannocchia et al. 2015), even when a simplified linear dynamic model is used, and this is, therefore, the standard form of practical MPC implementation. However, the general case of the separation principle does not

hold for linear or nonlinear systems (Doyle 1978, Freeman 1995), which means that there are fundamental limitations to the use of MPC with state estimators for feedback control.

While MPC is useful for cases where multivariable control is necessary, it should not be considered the only possible solution for supervisory control. For sufficiently decoupled systems, the use of decentralized control gives good performance, and its simplicity of tuning, implementation, and interpretability justifies its use over MPC. Even if the control loops have some interaction, they can be decoupled in time with the sequential design of the loops (Skogestad and Postlethwaite 2005), if the response time for some loops is not critical.

In this sense, considering that good performance can be obtained by a given feedback controller, it is often more important to make a good choice of controlled variables (CVs), such that the closed-loop system tracks good objectives. This is the idea of self-optimizing control, with "self-optimizing" being used in the sense that the chosen controlled variables (CVs), when kept at constant setpoints, drive the system to the optimal operating point (Morari et al. 1980, Skogestad 2000), or at least close to it. This choice of CVs is done such that there is no need for coordination by a higher optimization layer, which means that the real-time optimization (RTO) layer can be completely removed for simple problems. While complex problems can still benefit from this RTO layer, the supervisory layer is designed to give a small economic loss, which improves performance in the fast time scales, i.e. when RTO updates are not yet available.

The choice of good self-optimizing CVs has been extensively studied (Jäschke et al. 2017), and most design methods are based on the assumption that the set of active constraints is constant and directly controlled, with some important exceptions being the works of Manum and Skogestad (2012), Ye et al. (2023), which did not exhaust the topic. Additionally, Reyes-Lúa et al. (2019), Krishnamoorthy and Skogestad (2019) have discussed the use of logic control elements to deal with changes in active constraints, with guidelines that apply when the operating regions are known beforehand.

1.1 Scope of the thesis

The main challenge that this thesis aims to tackle is the design of self-optimizing control structures for systems that are subject to changes in active operational constraints. This arises when sustained disturbances affect the system, changing the way it should operate at steady state. This thesis will mostly focus on the steady-state optimal operation problem, except for Chapter 7, which presents a dynamic optimal operation problem.

When a new constraint is reached, it is intuitive that these should also be controlled directly, in order to remain feasible and reject the disturbances as quickly and effectively as possible. On the other hand, it is less obvious when to stop controlling a constraint, but controlling the constraints when they are not optimally active is too conservative and can lead to considerable economic losses. In other words, it is not immediately clear what unconstrained degrees of freedom are to be controlled when constraint control should be given up. A flexible control structure, with changing control objectives, is a solution that allows for controlling the constraints only when necessary.

For this reason, this thesis studies how to best implement switching control structures for optimal operation. This problem has two fundamental parts, those being the control strategy (decentralized (e.g. PID) or centralized (e.g. MPC) control, along with the control objectives) and the element for switching between objectives. While there is some experience with the use of logic control elements for switching in decentralized control (Reyes-Lúa and Skogestad 2019) and with defining controlled variables for different operating regions (Jäschke and Skogestad 2012), this thesis aims to expand on these topics by proposing complete control structures that can be applied to generic systems. Furthermore, because steady-state optimality conditions usually involve driving a projection of the cost gradient to zero (when the Karush–Kuhn–Tucker conditions apply), estimating this cost gradient plays a vital role in the course of this thesis.

1.2 Thesis structure

The main chapters of this thesis are based on self-contained manuscripts, which have been or will be submitted to peer-reviewed publication.

In Chapter 2, assuming knowledge of the cost gradient from a model-based estimator, the main contribution is the development of a simple decentralized control framework that operates optimally over all possible constraint regions. This framework can be applied to systems where the number of constraints is smaller than the number of operational degrees of freedom.

The general multivariable case is addressed in Chapter 3, where the main contribution is a methodology for applying MPC with different tracking objectives for each set of active constraints. It consists of using a cost gradient estimate, which here is a static linear combination of measurements, for both active set detection and reference tracking.

In Chapter 4, self-optimizing control methods for the design of CVs are used to obtain the best cost gradient estimate to be controlled for minimizing economic loss under disturbances and measurement bias. This expands previous results that were

valid for systems with a sufficient number of perfect measurements (used in the previous chapter), and the result is now applicable to the case with measurement noise and changing active constraints.

In Chapter 5 a heat exchanger network with more constraints than inputs is considered. We show how decentralized, region-based control can be used to operate such a system optimally. Here, the gradient estimate is based on an analytic derivation for heat exchanger systems.

We exemplify how a primal-dual feedback optimizing control structure can be used for optimal operation in Chapter 6. It can be implemented under the assumption that gradients are measured and that a time-scale separation is imposed between the primal and dual control layers.

Chapter 7 is fundamentally different from the preceding chapters in that dynamic optimal operation is the key issue. Here, the case of interest is the optimal inventory management subject to unit bottlenecks solved through feedback control. The bidirectional inventory control is the basis for solving this problem, and an extra control layer is proposed to deal with minimum flow constraints. The similarity with the previous chapters is in the switching control of constraints to achieve optimal operation, but in this case, there are no unconstrained degrees of freedom.

Some notes on self-optimizing control methods are presented in Chapter 8, with a simpler derivation and alternate formulation for the existing exact local method, and an explicit expression for the case where disturbance rejection is to be prioritized, which is valid regardless of the number of measurements.

In Chapter 9 the thesis is concluded with some final remarks and perspectives for future work on the discussed topics.

1.3 List of publications

1.3.1 Conference papers

- L. F. Bernardino, D. Krishnamoorthy, and S. Skogestad. Comparison of simple feedback control structures for constrained optimal operation. *IFAC-PapersOnLine*, 55(7):883–888, 2022a **(Chapter 6)**

- L. F. Bernardino, D. Krishnamoorthy, and S. Skogestad. Optimal operation of heat exchanger networks with changing active constraint regions. In *Computer Aided Chemical Engineering*, volume 49, pages 421–426. Elsevier, 2022b **(Chapter 5)**

- L. F. Bernardino and S. Skogestad. Bidirectional inventory control with

optimal use of intermediate storage and minimum flow constraints. *IFAC-PapersOnLine*, 56(2):2665–2670, 2023a **(Chapter 7)**

- L. F. Bernardino and S. Skogestad. Decentralized control for optimal operation under changing active constraints. In *Computer Aided Chemical Engineering*, volume 52, pages 1699–1704. Elsevier, 2023b

1.3.2 Journal papers

- L. F. Bernardino and S. Skogestad. Decentralized control using selectors for optimal steady-state operation with changing active constraints. *Journal of Process Control*, 137:103194, 2024c **(Chapter 2)**

- L. F. Bernardino and S. Skogestad. Optimal switching of MPC cost function for changing active constraints. *Submitted to Journal of Process Control*, 2024b **(Chapter 3)**

- L. F. Bernardino and S. Skogestad. Optimal measurement-based cost gradient estimate for real-time optimization. *Submitted to Computers & Chemical Engineering*, 2024a **(Chapter 4)**

1.3.3 Other conference presentations

- L. F. Bernardino, D. Krishnamoorthy, and S. Skogestad. Gaussian process based grey-box modeling of heat exchanger networks. In *Computer Aided Process Engineering Forum*, 2020.

- L. F. Bernardino, D. Krishnamoorthy, and S. Skogestad. Supervisory control for optimal operation under changing active constraints. In *23rd Nordic Process Control Workshop*, 2022.

- L. F. Bernardino and S. Skogestad. Bidirectional inventory control with optimal use of intermediate storage and minimum flow constraints. In *Foundations of Computer Aided Process Operations / Chemical Process Control*, 2023.

- L. F. Bernardino and S. Skogestad. Decentralized control for optimal operation under changing active constraints. In *24rd Nordic Process Control Workshop*, 2023.

- L. F. Bernardino and S. Skogestad. Real-time optimization with changing active constraints solved through decentralized feedback control. In *2023 AIChE Annual Meeting*. AIChE, 2023c

Chapter 2

Decentralized control using selectors for optimal steady-state operation with changing active constraints

This chapter has been published as a full paper:

> L. F. Bernardino and S. Skogestad. Decentralized control using selectors for optimal steady-state operation with changing active constraints. *Journal of Process Control*, 137:103194, 2024c

2.1 Introduction

The integration of optimization and control is very important when designing the control system for a process. The main objective of the control system is to keep the process stable and operating at the economically optimal operating point. Although these two objectives can be assessed simultaneously, for example, using economic model predictive control (EMPC) (Ellis et al. 2014), a simpler, and in most cases equally optimal[1], approach is to decompose the system hierarchically into an optimization and a control layer as shown in Figure 2.1, where setpoints

[1] In fact, in some cases a decomposed approach with separate optimization and control layers may be better performing economically than EMPC, because the control layer may be tuned to be fast, whereas this is likely difficult to achieve with a centralized solution like EMPC. This may give economic benefits, especially for fast-changing disturbances.

CV^{sp} are used to connect the two layers. The setpoints may need to be updated due to disturbances that affect the process economics. In the standard implementation in Figure 2.1, the real-time optimization (RTO) and setpoints update is performed on a slow time scale based on a detailed nonlinear process model and the estimated states of the process. In most cases, the RTO layer is static.

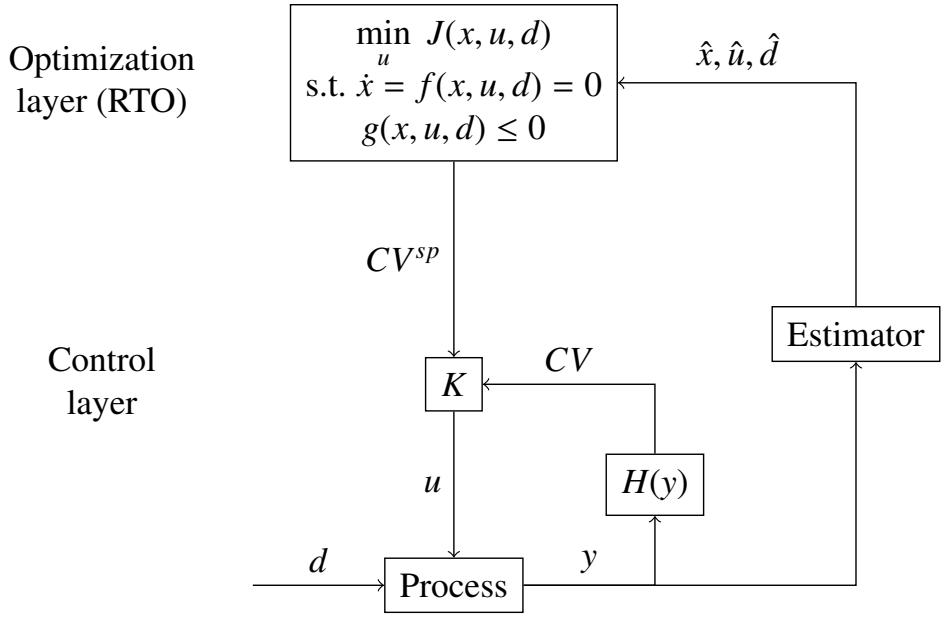


Figure 2.1: Standard optimizing control implementation with separate layers for real-time optimization (RTO) and control (K, which can be e.g. MPC or PID). J denotes the (economic) cost function to be minimized, f the process model, g the process constraints, x the model states, d the disturbances, and u the process inputs (MVs).

Based on the concept of Morari et al. (1980) of feedback optimizing control, the aim of the current paper is to move the real-time optimization, or at least parts of it, into the control layer. A recent review on this topic is given in Krishnamoorthy and Skogestad (2022), where the authors state some of the challenges with RTO implementation, including the cost of developing the model, the uncertainty related to the model and its parameters (or disturbances), and human aspects related to the maintenance of an optimization layer in addition to the already existing digital control system (DCS). The importance of feedback optimizing control lies in being able to reject disturbances that affect economic performance in a simple manner, without relying on an upper optimization layer that may sometimes not even exist. To that end, an appropriate selection of the controlled variables (CVs) for the control layer is important. This is the main idea of self-optimizing control (Skogestad 2000). It is particularly important to include the active constraints as

CVs, that is, the constraints that are optimally at their limiting value (Maarleveld and Rijnsdorp 1970, Morari et al. 1980). If information about the cost gradient is available, the optimal CVs are the active constraints plus the reduced cost gradients, and by controlling these at a constant setpoint of zero we may eliminate the optimization layer Jäschke and Skogestad (2012). This choice of CVs is valid if the set of active constraints does not change in the considered operating region.

Dealing with changes in active constraints has been a concern in previous works. For example, Cao (2004) implemented a cascade control structure with selectors to avoid constraint violation by the lower self-optimizing layer, and Graciano et al. (2015) applied MPC with zone control to the same end. A global self-optimizing control method for changing active constraints has been proposed by Ye et al. (2017), where the goal is to minimize the average loss obtained with a single set of CVs. However, in a new active constraint region, not only do the active constraints change, but the directions related to the reduced cost gradient change accordingly. This means that to eliminate the RTO layer one needs to change the control layer in Figure 2.1 during operation, both in terms of the selected CVs and the corresponding feedback controller K. With this perspective, Manum and Skogestad (2012) has considered a centralized, steady-state analysis on switching control structures, with different CVs for each region.

However, the implementation of such a region-based control strategy quickly becomes impractical. This is because the number of active constraint regions grows exponentially with the number of constraints. Let n_u denote the number of process inputs or manipulated variables (MVs) and n_g the number of independent constraints. The upper bound on the number of active constraint regions is 2^{n_g}, which is reached when all constraint combinations are feasible (Reyes-Lúa and Skogestad 2019). In each region, we ideally need a new controller K, and if we want to use decentralized control then we need to design n_u single-input single-output (SISO) controllers in each region. For example, with $n_g = 4$ and $n_u = 5$, there could be up to $2^4 = 16$ constraint regions, which may require the tuning of $2^{n_g} \cdot n_u = 16 \cdot 5 = 80$ SISO controllers. Even though some CVs are reused between regions, the number of necessary SISO loops will be high.

The key contribution of this paper is to propose a simple and generic region-based control structure with only $n_g + n_u$ SISO controllers, as represented in Figure 2.2, with the same set of unconstrained variables (CV^0 and CV^{0g}) in *all* operating regions[2]. Considering the previous example, this structure would have only $4+5 = 9$ SISO controllers. In the paper, we show that the unconstrained variables are ob-

[2]The superscript 0 is used to indicate the unconstrained case and g the constrained case. The superscript $0g$ indicates unconstrained cases that are associated with switching constraints.

tained from n_u projections of the full cost gradient with respect to the inputs, $\nabla_u J$. This leads to n_u gradient controllers (K^0 and K^{0g}) and n_g constraint controllers (K^g). However, at any given time, only a subset with n_u of the $n_u + n_g$ controller outputs is implemented as process inputs, with the switching logic choosing between the controller outputs u^{0g} and u^g.

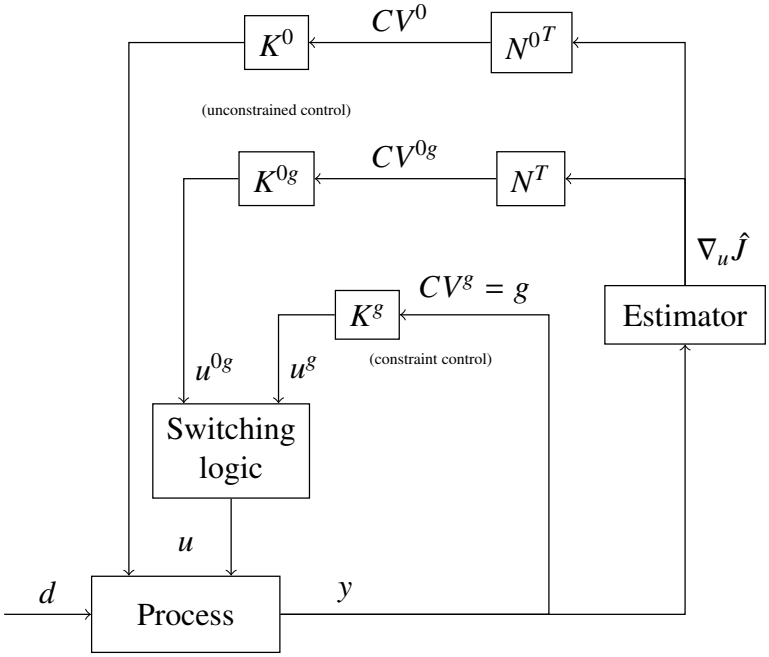


Figure 2.2: Proposed optimizing control implementation, assuming $n_u \geq n_g$. The controllers K^0, K^{0g} and K^g are usually single-variable PID controllers. The projection (null-space) matrices N^0 and N are defined in Equation (2.7) and Equation (2.8), respectively. There is no CV^0, N^0, and K^0 if $n_u = n_g$. Note that the optimization layer in Figure 2.1is eliminated, and an estimate $\nabla_u \hat{J}$ of the cost gradient is needed. The switching logic takes care of the change between active constraint regions. In this paper, this logic is decentralized to n_g individual blocks, see Figure 2.3, which can be implemented as min or max selectors according to Theorem 2.3.

The second key contribution of this paper is to show that the switching logic in Figure 2.2 can be effectively implemented using n_g min or max selectors, which are well-known advanced control elements and commonly used in practical control applications. An important decision is to pair each constraint to an MV, but this pairing problem is not addressed in this paper (the interested reader is referred to Skogestad and Postlethwaite (2005)). The main assumptions in this work are that we have at least as many MVs as constraints ($n_u \geq n_g$), and that an estimator for the unconstrained cost gradient $\nabla_u J$ is available. In terms of cost gradient es-

timation, there are several methods available (see Krishnamoorthy and Skogestad (2022)), and in this work, we use the simple model-based approach of dynamic state estimation and model linearization proposed by Krishnamoorthy et al. (2018).

Selectors have been used in industry to switch between CVs since the 1940s (Skogestad 2023). Selectors are also used in academic case studies on optimal operation (Reyes-Lúa and Skogestad 2019, Krishnamoorthy and Skogestad 2019). In these case studies, a control structure is proposed for the nominal operating region, with added logic elements and control loops to deal with the neighboring regions. However, the treatment of the unconstrained degrees of freedom is not clear. Krishnamoorthy and Skogestad (2020) proposes a framework for constraint handling using min and max selectors, focusing on systems with a single MV, and therefore not considering the changes of reduced gradients for the unconstrained variables. To the best of the authors' knowledge, even though a general scheme for the paradigm of region-based control is proposed in the review paper by Krishnamoorthy and Skogestad (2022), a systematic procedure for designing a decentralized control structure for optimal operation of generic multivariable systems has not yet been explored, as well as whether there are any fundamental limitations for the design of such systems. In this work, we explore these topics, and we describe a class of multivariable systems for which a decentralized control structure is always possible.

2.2 Decentralized control framework for optimal operation

We consider a generic, steady-state optimization problem given by:

$$\begin{aligned} \min_{u} \quad & J(u,d) \\ \text{s.t.} \quad & g(u,d) \leq 0 \end{aligned} \tag{2.1}$$

Here, $J : \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}$ is a scalar cost function to be minimized, $g : \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_g}$ is the function that returns the vector of inequality constraints, $u \in \mathbb{R}^{n_u}$ is the vector of decision variables (MVs), and $d \in \mathbb{R}^{n_d}$ is the vector of disturbances. Note that the states x (see Figure 2.1) have been formally eliminated from the equations, such that J and g are functions only of the independent variables u and d. Introduce the Lagrange function $\mathcal{L}(u, \lambda, d) = J(u, d) + \lambda^T g(u, d)$. Then, for a given value of d, define u^* as the solution of Equation (2.1), which satisfies the Karush–Kuhn–Tucker (KKT) conditions (Nocedal and Wright 2006):

$$\nabla_u \mathcal{L}(u^*, \lambda^*, d) = \nabla_u J(u^*, d) + (\nabla_u g(u^*, d))^T \lambda^* = 0 \tag{2.2a}$$
$$g(u^*, d) \leq 0 \tag{2.2b}$$
$$\lambda^* \geq 0 \tag{2.2c}$$
$$\lambda_i^* g_i(u^*, d) = 0, \qquad i = 1, \cdots, n_g \tag{2.2d}$$

Here, λ is the vector of Lagrange multipliers associated with the inequality constraints, and λ^* is its optimal value. We remark that the KKT conditions only imply that the solution is a stationary point, and they are also satisfied by local minima or maximum and saddle points. We do not address these issues in this work, and we consider that the optimization problem in Equation (2.1) is convex. While these optimization problems can be efficiently solved using numerical methods, we here focus on how to solve these problems with feedback control. For this, we rewrite the KKT conditions as control objectives, which allows us to embed the optimization into the control layer design.

The set of active constraints $\mathcal{A}$ is defined as the set that satisfies $g_i(u^*, d) = 0$ for $i \in \mathcal{A}$. For convenience, define $g_{\mathcal{A}} : \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_a}$ as the function that returns the active constraints. Define the matrix:

$$G^g = \nabla_u g(u, d) \tag{2.3}$$

as the gradient of the constraints with respect to the MVs, and the matrix $G^g_{\mathcal{A}} = \nabla_u g_{\mathcal{A}}(u, d)$ as the gradient of the active constraints with respect to the MVs. If the set of active constraints $\mathcal{A}$ is known, Jäschke and Skogestad (2012) prove that optimality can be attained by controlling to zero the active constraints and the associated reduced cost gradient. Their result is given by the following theorem:

Theorem 2.1 (**Optimal controlled variables).** *Consider the optimization problem in Equation* (2.1)*, where we assume that linear independence constraint qualification (LICQ) holds. We assume that the set of optimally active constraints $\mathcal{A}$ is known. Let $N_{\mathcal{A}} \in \mathbb{R}^{n_u \times (n_u - n_a)}$ be a basis for the nullspace of $G^g_{\mathcal{A}}$ such that:*

$$G^g_{\mathcal{A}} N_{\mathcal{A}} = 0 \tag{2.4}$$

Further, define the reduced cost gradient as:

$$\nabla_{u,\mathcal{A}} J(u, d) = N_{\mathcal{A}}^T \nabla_u J(u, d) \tag{2.5}$$

Then controlling $g_{\mathcal{A}}(u, d) = 0$ and $\nabla_{u,\mathcal{A}} J(u, d) = 0$ results in optimal steady-state operation.

Proof. (Jäschke and Skogestad 2012) If the active constraints $\mathcal{A}$ are known, the necessary optimality conditions (2.2) are equivalent to:

$$\begin{cases} \nabla_u \mathcal{L}(u^*, d) = \nabla_u J(u^*, d) + \left(G^g_{\mathcal{A}}\right)^T \lambda^*_{\mathcal{A}} = 0 \\ g_{\mathcal{A}}(u^*, d) = 0 \end{cases} \tag{2.6}$$

where $\lambda^*_{\mathcal{A}} > 0$ is the optimal vector of Lagrange multipliers for the active constraints. Premultiplying $\nabla_u \mathcal{L}(u^*, d)$ by $N^T_{\mathcal{A}}$ leads to:

$$N^T_{\mathcal{A}} \nabla_u \mathcal{L}(u^*, d) = N^T_{\mathcal{A}} \nabla_u J(u^*, d) + \left(G^g_{\mathcal{A}} N_{\mathcal{A}}\right)^T \lambda^*_{\mathcal{A}} = 0$$

Since by definition $G^g_{\mathcal{A}} N_{\mathcal{A}} = 0$, the optimality conditions are equivalent to $g_{\mathcal{A}}(u^*, d) = 0$ and $N^T_{\mathcal{A}} \nabla_u J(u^*, d) = 0$, which are n_u equations that fully determine u^* because $N_{\mathcal{A}}$ is full rank, and the associated optimal Lagrange multiplier can always be found as $\lambda^*_{\mathcal{A}} = -\left(G^g_{\mathcal{A}} (G^g_{\mathcal{A}})^T\right)^{-1} G^g_{\mathcal{A}} \nabla_u J(u^*, d)$. Therefore, enforcing $g_{\mathcal{A}}(u^*, d) = 0$ and $N^T_{\mathcal{A}} \nabla_u J(u^*, d) = 0$ leads to satisfying (2.6), which is equivalent to satisfying (2.2). □

In terms of feedback control, Theorem 2.1 says that $g_{\mathcal{A}}$ and $\nabla_{u,\mathcal{A}} J$ (both with setpoints 0) are the steady-state optimal CVs for a given operating region where the active constraints do not change. Here, the reduced cost gradient $\nabla_{u,\mathcal{A}} J = N^T_{\mathcal{A}} \nabla_u J$ is defined as the gradient in the unconstrained directions as given by the nullspace $N_{\mathcal{A}}$ of the active constraints (Jäschke and Skogestad 2012). If the system is to operate at another active constraint region, however, the CVs need to change, and if shifts in operating regions happen in real-time, the control system needs to automatically detect these region switches. The main idea of this work is to design a decentralized control structure, see Figure 2.2, for all possible active constraint regions of the optimization problem in Equation (2.1). The main assumption for guaranteeing the existence of this decentralized control structure is as follows:

Assumption 2.1. *The matrix G^g is always full row rank, and the number of constraints is not greater than the number of MVs, that is,* $\text{rank}(G^g) = n_g$*, and* $n_u \geq n_g$*.*

This not only guarantees LICQ for any set of constraints that may be optimally active, but it also guarantees the existence of decoupled CVs for optimal operation, as shown in the next theorem. For use in the next theorem, define N^0 as an orthonormal basis of the nullspace of G^g, that is:

$$G^g N^0 = 0 \tag{2.7}$$

The matrix N^0 represents the unconstrained directions that are never in conflict with constraint control. Note here that N^0 is an empty matrix (nonexistent) if we have as many constraints as inputs ($n_u = n_g$). Further define G^g_{-i} as the matrix containing all but the i-th row of G^g, and define:

$$N = \begin{bmatrix} N_1 & \cdots & N_{n_g} \end{bmatrix} \tag{2.8}$$

as a matrix of n_g columns, where each column N_i is a unitary vector such that:

$$\begin{bmatrix} G^g_{-i} \\ N^{0T} \end{bmatrix} N_i = 0 \tag{2.9}$$

Each vector N_i represents the direction that may conflict with the corresponding constraint g_i, as shown next.

Theorem 2.2 (Optimal switching between CVs). *Given that Assumption 2.1 holds and that the active constraint index set is $\mathcal{A}$, the following control strategy allows for optimal operation:*

- *If $n_u > n_g$, which means N^0 is non-empty, control $CV^0 = N^{0T}\nabla_u J(u,d) = 0$;*
- *For $i = 1, 2, \cdots, n_g$, if $i \in \mathcal{A}$, control $CV^g_i = g_i(u,d) = 0$; otherwise, control $CV^{0g}_i = N^T_i \nabla_u J(u,d) = 0$.*

Proof. To prove Theorem 2.2, it is sufficient to prove that the controlled variables are equivalent to the necessary first-order optimality conditions. Firstly, it is useful to note that, due to its construction, $G^g N_i = (G^g_i N_i)\hat{e}_i$, with G^g_i being the i-th row of G^g, and $\hat{e}_i$ being the i-th unit vector from the standard basis. Additionally, if the active constraint set is $\mathcal{A}$, and the inactive constraint set is $\mathcal{I} = \{1, \ldots, n_g\} - \mathcal{A}$, the optimality conditions can be written as:

$$\begin{cases} \nabla_u \mathcal{L}(u^*, d) = \nabla_u J(u^*, d) + G^{gT}\lambda^* = 0 \\ g_i(u^*, d) = 0, \quad i \in \mathcal{A} \\ \lambda^*_i = 0, \quad i \in \mathcal{I} \end{cases}$$

Let $N_{\mathcal{I}}$ be the matrix with columns equal to N_i for $i \in \mathcal{I}$. Then, premultiplying $\nabla_u \mathcal{L}$ by $\begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix}^T$ leads to:

$$\begin{aligned}\begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix}^T \nabla_u \mathcal{L} &= \begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix}^T \nabla_u J + \left(G^g \begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix} \right)^T \lambda^* \\ &= \begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix}^T \nabla_u J + \left(\begin{bmatrix} G^g N_{\mathcal{I}} & 0 \end{bmatrix} \right)^T \lambda^*\end{aligned}$$

Here, $\left(G^g N_{\mathcal{I}}\right)^T \lambda^* = 0$, because $G^g N_i = (G^g_i N_i)\hat{e}_i$, and, from the optimality conditions, $\hat{e}_i^T \lambda^* = \lambda_i^* = 0$ for $i \in \mathcal{I}$. Therefore, the optimality conditions become

$$\begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix}^T \nabla_u \mathcal{L} = \begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix}^T \nabla_u J = 0,$$

which are the CVs proposed in addition to $g_i(u,d) = 0$ for $i \in \mathcal{A}$. Similarly to Theorem 2.1, this fully defines the operational degrees of freedom, and a suitable vector of Lagrange multipliers can be found. □

Note that the matrix

$$N_{(\mathcal{A})} = \begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix} \tag{2.10}$$

used in the proof of Theorem 2.2 is a particular parametrization of the nullspace matrix $N_{\mathcal{A}}$ from Theorem 2.1, and therefore both results are equivalent for a given active constraint region. In Theorem 2.2 however, we specify an ideal association between CVs such that the handling of region switching may be done in a decentralized fashion, avoiding changes in the rest of the control structure. For instance, if the i-th constraint changes from inactive to active, only the corresponding unconstrained degree of freedom $CV_i^{0g} = N_i^T \nabla_u J$ will become uncontrolled, and the remaining CVs are kept unaltered. In addition, the matrix $N_{(\mathcal{A})} = \begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix}^T$ is designed to be full row rank, and therefore all operational degrees of freedom are filled for any active set $\mathcal{A}$. The choice of building the vectors N_i unitary and orthogonal to N^0 is purely for the uniqueness of the solution, as one could propose another projection $N_i' = \alpha N_i + N^0 w$ for any nonzero scaling factor α and any vector w, and optimal operation would still be attained, as ${N^0}^T \nabla_u J$ is always optimally zero.

Theorem 2.2 states a general set of feedback control objectives to attain optimal operation. It does not specify the type of controller to be used, and one may apply these results to obtain optimal operation with conventional tracking MPC with switching objectives to eliminate the RTO layer. This would be useful for cases where decentralized control performs poorly, but one still wishes to propose a simple control layer. In this work, however, we choose to explore the implications of this result for decentralized control, which is often more easily implemented in practice.

Pairing of MVs and CVs. It should be noted that Theorem 2.2 makes no distinction about the pairing between MVs and CVs, and it is left for the practitioner to make this pairing taking into account controllability and performance aspects. However, the theorem states the optimal association between CVs for region switching, which means that the control of $CV_i^g = g_i$ and $CV_i^{0g} = N_i^T \nabla_u J$ must be performed by the same MV in the case of a decentralized framework. From now on, it is considered that the MVs are ordered such that u_i is used to control the pair $CV_i^g = g_i$ and CV_i^{0g} for $i \leq n_g$. These considerations lead to the control structure presented in Figure 2.3. Here, $\nabla_u \hat{J}$ represents the estimate of the cost gradient ($\nabla_u J$), K_i^g represent the individual constraint controllers, K_i^{0g} represent the individual gradient controllers that are conditionally active (i.e. only one of K_i^g and K_i^{0g} is active at any given time), and K_i^0 represent the individual gradient controllers that are always active. It is important that the controllers K_i^g and K_i^{0g} include anti-windup action so that the integral modes in the inactive controllers do not grow indefinitely.

We finally focus on the applicability of min/max selectors as the logic elements to switch between active constraint regions, which were left undetermined in Figure 2.3 as "select" blocks. These selectors are applied on the controller outputs u_i^g and u_i^{0g} associated with the controlled variables g_i and CV_i^{0g}, respectively, resulting in the process input (MV) u_i to be applied to the system. This methodology was adopted in Krishnamoorthy and Skogestad (2020) for optimal operation in the scalar case, i.e. with a single MV, where it was concluded that a constraint with a positive gain ($G^g > 0$) requires a min selector, whereas a negative gain ($G^g < 0$) requires a max selector. In the next theorem, we present similar results for the multivariable case.

Theorem 2.3 (Decentralized control. Applicability of min/max selectors). *In addition to Assumption 2.1, assume that the Hessian of the cost function with respect to the inputs is constant and positive definite, that is, $\nabla_u J(u,d) = J_{uu}u + J_{u,m}(d)$, with $J_{uu} > 0$ and arbitrary $J_{u,m}(d)$, and that G^g is constant. Consider the control structure in Figure 2.3 and Theorem 2.2, and assume that every possible subsystem is stable.*

Let u_i^{0g} denote the value of u_i that controls $CV_i^{0g} = N_i^T \nabla_u J(u,d) = 0$, and let u_i^g denote the value of u_i that controls $g_i(u,d) = 0$. For a given active set $\mathcal{A}$, the associated nullspace of the active constraint gain matrix G_A^g is$N_{(\mathcal{A})} = \begin{bmatrix} N_{\mathcal{I}} & N^0 \end{bmatrix}$. Define the scaled projection matrix $P_{\mathcal{A}}$ and the transformed constraint gain matrix $G_{P_{\mathcal{A}}}^g$ as:

$$P_{\mathcal{A}} = N_{(\mathcal{A})} \left(N_{(\mathcal{A})}^T J_{uu} N_{(\mathcal{A})} \right)^{-1} N_{(\mathcal{A})}^T \tag{2.11}$$

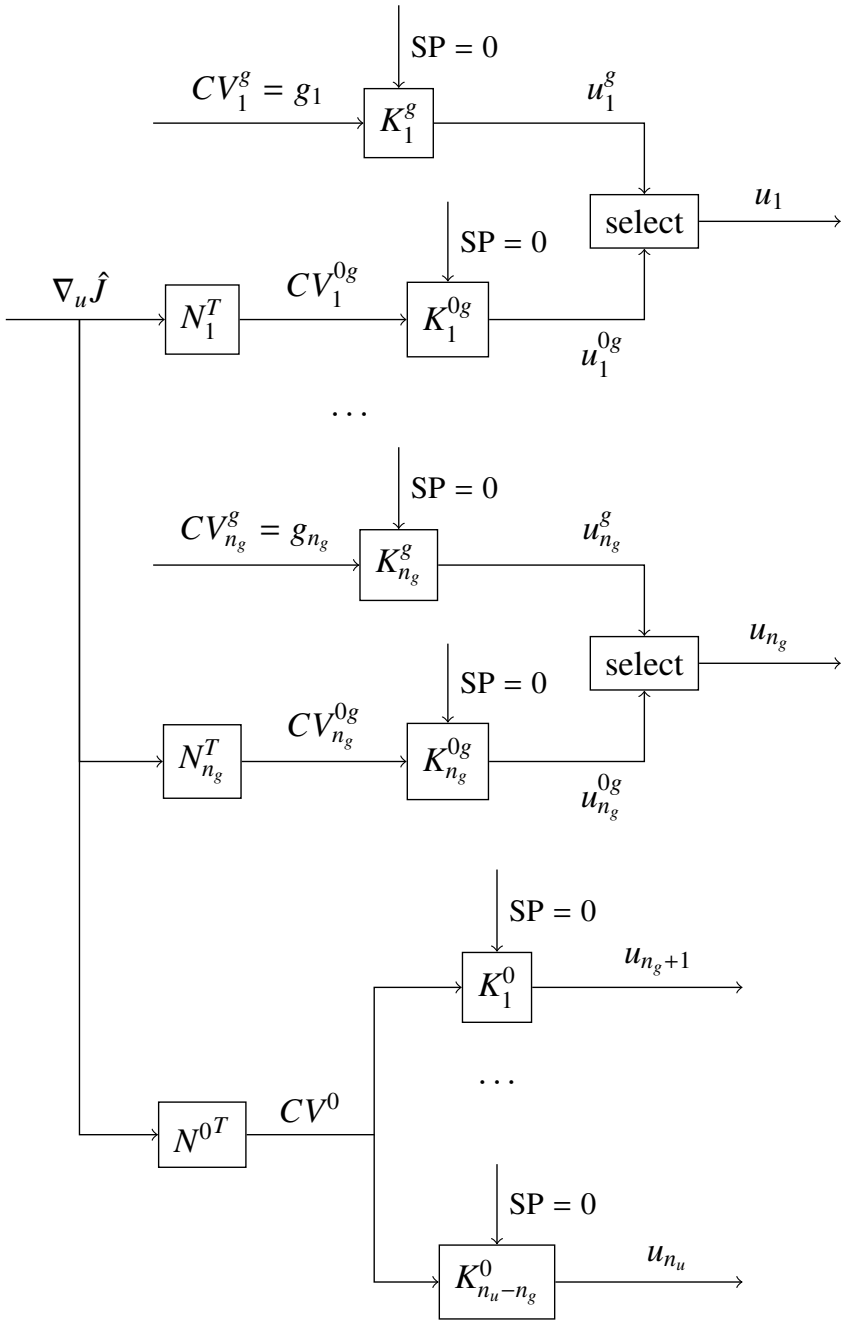


Figure 2.3: Decentralized control structure for optimal operation according to Theorem 2.2. The "select" blocks are usually max or min selectors (see Theorem 2.3).

$$G^g_{P_\mathcal{A}} = G^g P_\mathcal{A} \tag{2.12}$$

The optimal input is given by $u_i^* = \min(u_i^{0g}, u_i^g)$ *if the* i*-th diagonal element of the transformed gain matrix is positive (*$(G^g_{P_\mathcal{A}})_{ii} > 0$*) for any active set* $\mathcal{A}$ *that does not include* i*. Conversely, the optimal input is given by* $u_i^* = \max(u_i^{0g}, u_i^g)$ *if* $(G^g_{P_\mathcal{A}})_{ii} < 0 \; \forall \mathcal{A} \not\ni i$*.*

Proof. See 2.A □

It is worth noting that the i-th row of $G^g_{P_\mathcal{A}}$ is identically zero for $i \in \mathcal{A}$, since $P_\mathcal{A}$ involves a projection to the nullspace of the active constraints. If $(G^g_{P_\mathcal{A}})_{ii}$ changes sign for different active sets, a single type of selector would not account for all theoretical regions. The single-input case of Theorem 2.3 can be easily verified by writing $u^g - u^{0g} = -\frac{1}{J_{uu}} G^g \lambda^i$. As $J_{uu} > 0$ for a convex optimization problem, $G^g > 0$ leads to $u^* = \min(u^{0g}, u^g)$, and $G^g < 0$ leads to $u^* = \max(u^{0g}, u^g)$, which is equivalent to the result in Krishnamoorthy and Skogestad (2020).

Can some of the assumptions in Theorem 2.3 be removed? According to Theorem 2.3, the use of max- (or min-) selectors in Figure 2.3 assumes that $(G^g_{P_\mathcal{A}})_{ii}$ remains positive (or negative) for any active set $\mathcal{A}$ that does not include i. This is to rule out cases where the steady-state gain for control of the constraint g_i changes sign, as this would lead to instability with integral action in the controller. In other words, this is to rule out interacting processes where $u_i^* = \min(u_i^{0g}, u_i^g)$ for a given active set $\mathcal{A}$, and $u_i^* = \max(u_i^{0g}, u_i^g)$ for another. However, it is not clear whether this is a restriction in practice. Thus, it is possible that the assumption about no sign change for the diagonal elements $(G^g_{P_\mathcal{A}})_{ii}$ is not needed. This is left as an open research issue.

Cascade implementation. It is anyway possible to avoid this restriction by using the cascade switching implementation in Figure 2.4. That is, for this implementation the simple selector logic is always optimal without the assumption about the sign of $(G^g_{P_\mathcal{A}})_{ii}$ in Theorem 2.3. In the cascade implementation in Figure 2.4, the constraints are always controlled in the lower layer, and the optimal constraint setpoint g_i^{sp} will either be the value $g_i^{sp,0}$ that controls $CV_i^{0g} = 0$ or the constraint's limit value itself, such that $g_i^{sp} = \min(g_i^{sp,0}, 0)$ leads to optimal operation. This result can also be obtained by rewriting Theorem 2.3 in terms of the transformed inputs $v = \begin{bmatrix} g_1 & \cdots & g_{n_g} & u_{n_g+1} & \cdots & u_{n_u} \end{bmatrix}^T$, where it can be verified that the condition $(G^{g,v}_{P_\mathcal{A}})_{ii} > 0$ is always satisfied. This result is presented in 2.B.

The idea of using cascade control for self-optimizing control and constraint satisfaction has been previously proposed in Cao (2004). There, the cost gradient

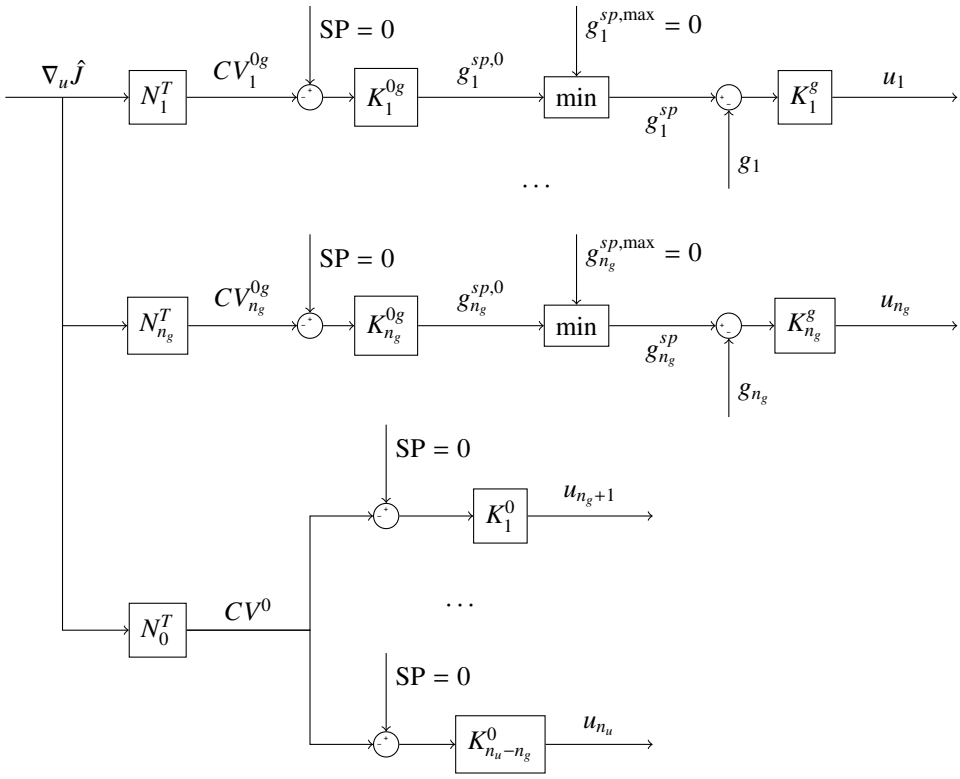


Figure 2.4: Decentralized control structure for optimal operation, using an alternative cascade implementation.

is controlled in the unconstrained case, while the lower layer keeps the system feasible by saturating the setpoint from the upper layer. This approach ensures feasibility and self-optimizing behavior at the unconstrained region, but optimality at all active constraint regions is only ensured by carefully selecting the CVs at the upper layer, which is the main idea of the present work. In addition, even though the cascade structure will always operate optimally at steady state, it requires that the outer controllers K_i^{0g} are sufficiently slower than the inner controllers K_i^g, and therefore the generic structure in Figure 2.3 offers more flexibility in terms of loop tuning and implementation of further control overrides. The simulations presented in this paper are for the implementation in Figure 2.3.

2.3 Case study 1 - Toy example

In this section, to illustrate the implementation of the proposed control structure, we consider a linear process with a quadratic cost function and 2 linear constraints. The process has 2 dynamic states x, 3 inputs (MVs) u, and 2 disturbances d. The linear state-space model is:

$$\dot{x} = \underbrace{\begin{bmatrix} -\frac{1}{\tau_1} & 0 \\ 0 & -\frac{1}{\tau_2} \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} \frac{0.2}{\tau_1} & 0 & 0 \\ 0 & \frac{0.2}{\tau_2} & 0 \end{bmatrix}}_{B} u + \underbrace{\begin{bmatrix} \frac{1}{\tau_1} & 0 \\ 0 & \frac{1}{\tau_2} \end{bmatrix}}_{B_d} d \tag{2.13}$$

with $\tau_1 = 1$ and $\tau_2 = 2$. It is assumed that both states are measured, that is, $y = Cx + Du$ with $C = I$ and $D = 0$.

The steady-state optimization problem in terms of the states is:

$$\begin{aligned} \min_{u} \quad & \frac{1}{2} x^T \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} x + \frac{1}{2} u^T \begin{bmatrix} 1 & -0.1 & -0.2 \\ -0.1 & 0.8 & -0.1 \\ -0.2 & -0.1 & 0.3 \end{bmatrix} u \\ \text{s.t.} \quad & \begin{cases} g_1 = x_1 - 0.8 x_2 \leq 0 \\ g_2 = u_1 + u_2 + u_3 \leq 0 \end{cases} \end{aligned} \tag{2.14}$$

At steady state, the states can be eliminated to give the following static optimization problem:

$$\begin{aligned} \min_{u} \quad & J(u,d) = \frac{1}{2}\, u^T \begin{bmatrix} 1.04 & -0.1 & -0.2 \\ -0.1 & 1.2 & -0.1 \\ -0.2 & -0.1 & 0.3 \end{bmatrix} u + u^T \begin{bmatrix} 0.2 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix} d \\ \text{s.t.} \quad & g(u,d) = \underbrace{\begin{bmatrix} 0.2 & -0.16 & 0 \\ 1 & 1 & 1 \end{bmatrix}}_{G^g} u + \begin{bmatrix} 1 & -0.8 \\ 0 & 0 \end{bmatrix} d \leq 0 \end{aligned} \tag{2.15}$$

For given disturbances d, we can solve the problem in Equation (2.15) to find the optimal steady-state inputs u^* and the active set $\mathcal{A}$. From this, we can graphically represent the active constraint regions as a function of the two disturbances as shown in Figure 2.5. Note that this is done for visualization purposes only and is not a part of the proposed method. In fact, for the proposed method we do not need to know what the disturbances are; what is needed is measured or estimated values for the constraints g and the unconstrained cost gradient $\nabla_u \hat{J}$. We see in Figure 2.5 that all $2^{n_g} = 2^2 = 4$ combinations of constraints are possible. Each region has a specific set of CVs for optimal operation, namely the active constraints and the corresponding reduced gradients, as given in Theorem 2.2.

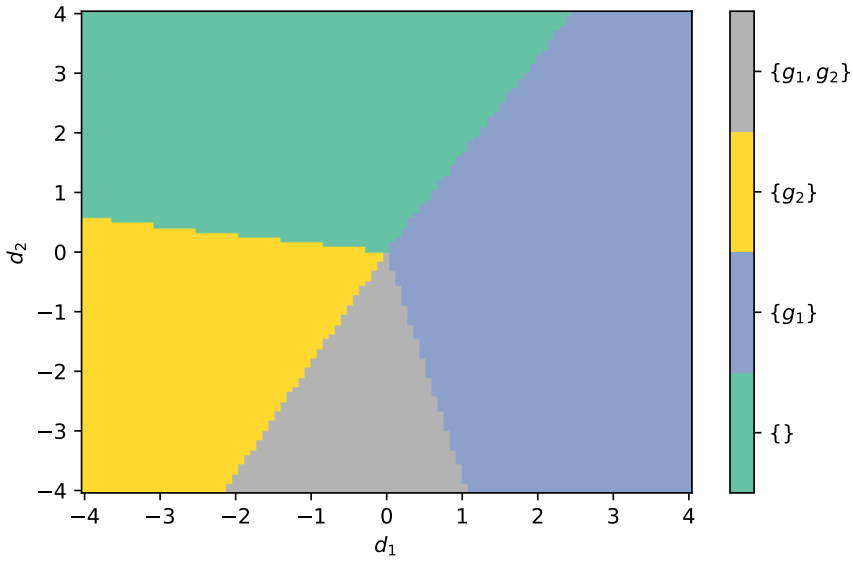


Figure 2.5: Active constraint regions for case study 1 as a function of disturbances.

We have $n_u = 3$ and $n_g = 2$, so with the proposed method, we need to design $n_u + n_g = 5$ SISO controllers with $n_g = 2$ selectors to obtain optimal steady-state

operation. Since $n_u > n_g$, we always have $n_u - n_g = 1$ unconstrained degree of freedom corresponding to the controlled variable $CV^0 = N^{0T}\nabla_u \hat{J}$. From the nullspace of the full G^g matrix, we find that this direction is given by

$$N^0 = \begin{bmatrix} -0.36214 & -0.45268 & 0.81482 \end{bmatrix}^T.$$

In addition, there are two unconstrained directions related to the two constraints. We have that $CV_1^{0g} = N_1^T \nabla_u \hat{J}$ should be controlled when g_1 is not active, and $CV_2^{0g} = N_2^T \nabla_u \hat{J}$ should be controlled when g_2 is not active. These directions are:

$$\begin{bmatrix} 1 & 1 & 1 \\ -0.36214 & -0.45268 & 0.81482 \end{bmatrix} N_1 = 0 \Longrightarrow N_1 = \begin{bmatrix} -0.73179 \\ 0.67952 \\ 0.052271 \end{bmatrix}$$

$$\begin{bmatrix} 0.2 & -0.16 & 0 \\ -0.36214 & -0.45268 & 0.81482 \end{bmatrix} N_2 = 0 \Longrightarrow N_2 = \begin{bmatrix} 0.50902 \\ 0.63627 \\ 0.57971 \end{bmatrix}$$

For designing a decentralized control structure, a pairing between the constraints and the MVs must be performed. From the steady-state gain matrix G^g we see that u_3 should not be used to control g_1 (because of zero gain). Otherwise, there are no clear restrictions, and g_1 is arbitrarily paired to u_1, and g_2 is paired to u_2. We must require that the corresponding unconstrained optimal CVs are paired accordingly, meaning that CV_1^{0g} is paired to u_1, CV_2^{0g} is paired to u_2, and CV^0 is paired to u_3.

$\mathcal{A}$	$(G^g_{P_\mathcal{A}})_{11}$	$(G^g_{P_\mathcal{A}})_{22}$
$\{\}$	0.201	1.443
$\{1\}$	-	1.801
$\{2\}$	0.155	-

Table 2.1: Diagonal elements of $G^g_{P_\mathcal{A}}$ for all relevant sets $\mathcal{A}$ for case study 1.

For selector design, Table 2.1 shows the transformed constraint gains calculated using Equation (2.12) for all active constraint sets, and we verify that the gains are always positive for both constraints. This means that selectors are possible for both control loops and that both selectors should be “min"-selectors. The resulting control structure is shown in Figure 2.6.

The cost gradient is estimated through a relinearization of the dynamic model at

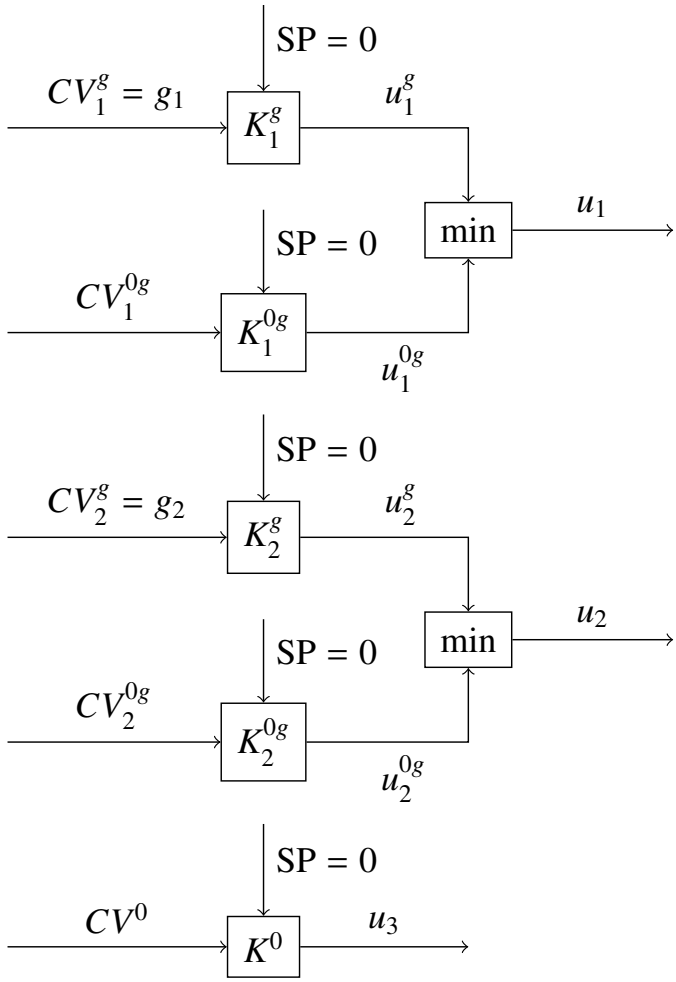


Figure 2.6: Decentralized control structure for case study 1.

the current estimated state to obtain the following linear model:

$$\begin{cases} \dot{x} = Ax + Bu \\ J = C_J x + D_J u \end{cases} \tag{2.16}$$

where, by setting $\dot{x} = 0$, the estimated steady-state cost gradient becomes (Krishnamoorthy et al. 2018)

$$\nabla_u \hat{J} = -C_J A^{-1} B + D_J \tag{2.17}$$

For state estimation, the model is augmented to include the disturbances as integrating states, according to:

$$\begin{cases} \begin{bmatrix} \dot{x} \\ \dot{d} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \\ y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} \end{cases} \tag{2.18}$$

To estimate the states, a continuous-time Kalman filter is implemented with this augmented model, and the estimated state $\hat{x}$ and current input u are used to evaluate the matrices in Equation (2.16) at all times, leading to the estimated cost gradient $\nabla_u \hat{J}$ in (2.17) . The matrices A, B, B_d, C, and D are as defined in Equation (2.13), and C_J and D_J are calculated from Equation (2.14) to give:

$$C_J = \hat{x}^T \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$$

$$D_J = u^T \begin{bmatrix} 1 & -0.1 & -0.2 \\ -0.1 & 0.8 & -0.1 \\ -0.2 & -0.1 & 0.3 \end{bmatrix}$$

We emphasize that analytical expressions for these derivatives are available due to the simplicity of this case study, and we encourage the use of automatic differentiation tools to obtain these matrices in more realistic case studies.

The constraint controllers K_1^g and K_2^g were designed according to the SIMC rules (Skogestad 2003) with the choice $\tau_{C,1} = 0.1$ s and $\tau_{C,2} = 0.01$ s. In terms of gradient control, we assume that the effect of the inputs on the estimated $\nabla_u \hat{J}$ is that of a pure gain process, neglecting any dynamics associated with the gradient estimation, and therefore the gradient controllers K_1^{0g}, K_2^{0g}, and K^0 become pure integral controllers. These were tuned according to the SIMC rules with $\tau_C = 0.5$ s. All controllers linked to selectors are implemented with anti-windup action

based on the back-calculation strategy (Åström and Rundqwist 1989, Skogestad 2023), with a tracking time of $\tau_T = 0.01$ s. The resulting controller tunings are summarized in Table 2.2.

Controller	Parameter	Value
K_1^g	K_c	50
	τ_I	1.0
K_2^g	K_I	100
K_1^{0g}	K_I	-2.382
K_2^{0g}	K_I	3.055
K^0	K_I	5.523

Table 2.2: PI controller tunings for case study 1. Note that $K_I = K_c/\tau_I$ is the integral gain. The first four controllers have anti-windup with tracking time $\tau_T = 0.01$ s

The closed-loop simulations are shown in Figure 2.7. To validate the optimality of the control structure, the disturbances were changed stepwise every 15 seconds (see lower left plots) to make the system operate in all four active constraint regions (see lower right plot). It can be seen that constraint changes are effectively handled, giving up the corresponding gradient projection when a constraint becomes active, and that operation is driven to the optimal steady state for all disturbances.

2.4 Case study 2 - Williams-Otto reactor

The control structure proposed in Section 2.2 depends on using projection matrices. These are constant only when the constraints are linear in the MVs. We now consider a nonlinear case study where this assumption is not satisfied and one may expect economic losses in some regions. The case study is based on the process described by Williams and Otto (1960) and studied in (Krishnamoorthy and Skogestad 2020), see Figure 2.8. It consists of a continuously stirred reactor tank with perfect level control, in which A and B are mixed, generating the main product P, the less interesting product E and the undesired byproduct G. The reactions and reaction rates are given by:

$$\begin{aligned} A + B &\xrightarrow{k_1} C & k_1 &= k_{0,1} e^{-E_1/T_r} \\ C + B &\xrightarrow{k_2} P + E & k_2 &= k_{0,2} e^{-E_2/T_r} \\ P + C &\xrightarrow{k_3} G & k_3 &= k_{0,3} e^{-E_3/T_r} \end{aligned}$$

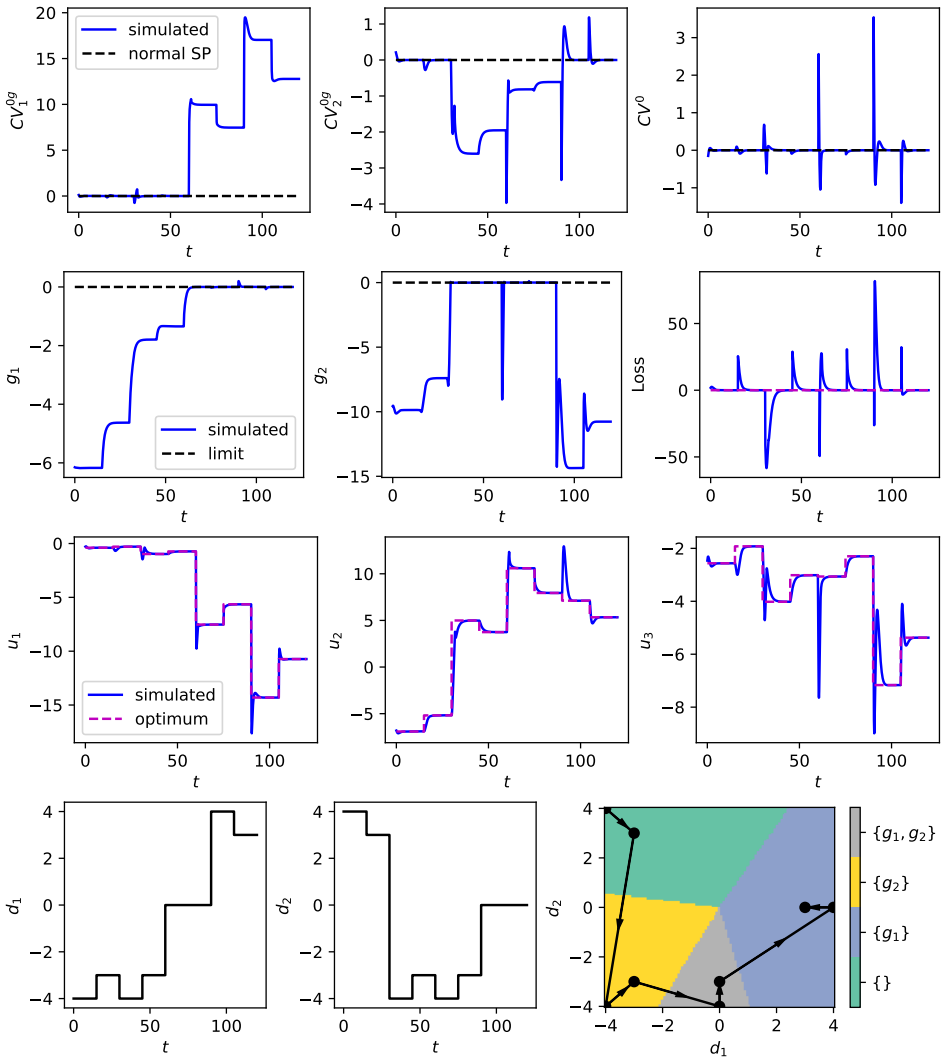


Figure 2.7: Closed-loop simulation results for case study 1.

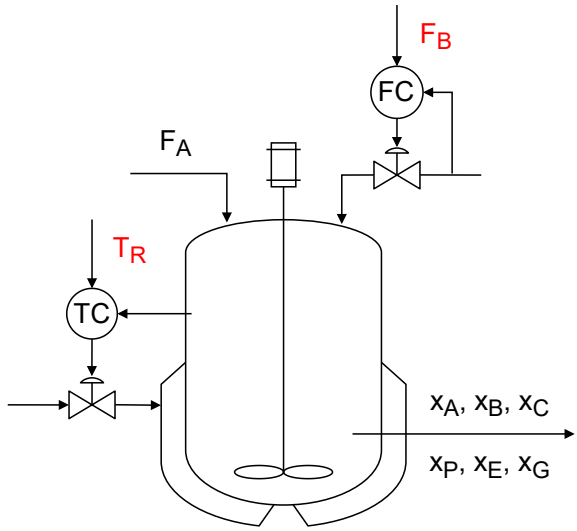


Figure 2.8: Schematic representation of Williams-Otto reactor, with MVs in red.

The component mass balances for the six components give the following set of ODEs:

$$\frac{dx_A}{dt} = \frac{F_A}{W} - \frac{(F_A + F_B)x_A}{W} - k_1 x_A x_B \tag{2.19a}$$

$$\frac{dx_B}{dt} = \frac{F_B}{W} - \frac{(F_A + F_B)x_B}{W} - k_1 x_A x_B - k_2 x_C x_B \tag{2.19b}$$

$$\frac{dx_C}{dt} = -\frac{(F_A + F_B)x_C}{W} + 2k_1 x_A x_B - 2k_2 x_C x_B - k_3 x_P x_C \tag{2.19c}$$

$$\frac{dx_P}{dt} = -\frac{(F_A + F_B)x_P}{W} + k_2 x_C x_B - 0.5 k_3 x_P x_C \tag{2.19d}$$

$$\frac{dx_E}{dt} = -\frac{(F_A + F_B)x_E}{W} + 2k_2 x_C x_B \tag{2.19e}$$

$$\frac{dx_G}{dt} = -\frac{(F_A + F_B)x_G}{W} + 1.5 k_3 x_P x_C \tag{2.19f}$$

The model parameters for this case study are summarized in Table 2.3. The economic cost J includes the cost of reactants p_A and p_B and the selling price of products p_P and p_E, and the operational constraints are related to maximum allowed values for x_A and x_E. The steady-state optimization problem becomes

$$\begin{aligned} \min_u J &= p_A F_A + p_B F_B - (F_A + F_B)\left[p_P(1+\Delta p_P)x_P + p_E x_E\right] \\ \text{s.t.} \quad & g_1 = x_E - 0.30 \leq 0 \\ & g_2 = x_A - 0.12 \leq 0 \end{aligned} \tag{2.20}$$

The degrees of freedom (MVs) are $u = \begin{bmatrix} F_B & T_r \end{bmatrix}^T$, and the disturbances are $d = \begin{bmatrix} F_A & \Delta p_P \end{bmatrix}^T$, where Δp_P is the relative change in the price of the main product, as defined in Equation (2.20).

Parameter	Value
W	2105 kg
$k_{0,1}$	1.6599×10^{-6} kg/s
$k_{0,2}$	7.2117×10^{-8} kg/s
$k_{0,3}$	2.6745×10^{-12} kg/s
E_1	6666.7 K
E_2	8333.3 K
E_3	11111 K
p_A	79.23 \$/kg
p_B	118.34 \$/kg
p_P	1043.38 \$/kg
p_E	20.92 \$/kg

Table 2.3: Model parameters for case study 2.

The active constraint regions as a function of the two disturbances are shown in Figure 2.9. In contrast to the previous case study, the lines delimiting each region are not straight. This alone should not affect the optimality of the proposed framework, as the optimality only requires that the constraints are linear in the MVs. However, since the latter does not hold for the case study, the use of constant projection matrices will lead to some economic loss.

We have $n_u = n_g = 2$ so Assumption 2.1 is satisfied. With the proposed method we need to design $n_u + n_g = 4$ SISO controllers with $n_g = 2$ selectors to obtain optimal steady-state operation. To obtain the gain matrix G^g from the MVs to the constraints, we need to linearize the steady-state model of the constraints. In the following simulations, we use the linearization performed at the nominal operating point presented in Table 2.4, leading to fixed CVs for operation in all regions. This linearization gives:

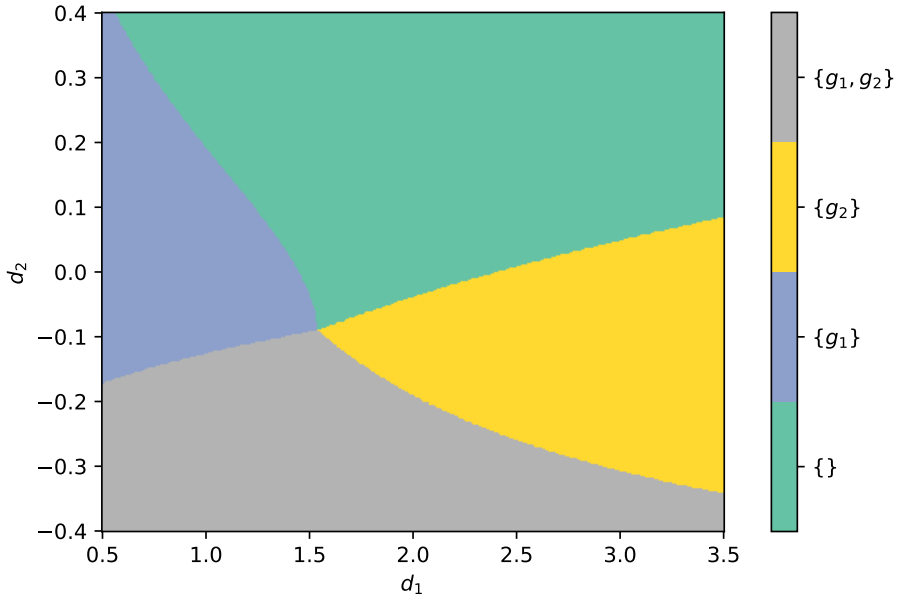


Figure 2.9: Active constraint regions for case study 2 as a function of disturbances.

$$G^g = \begin{bmatrix} -0.1045 & 0.003268 \\ -0.04379 & -0.00241 \end{bmatrix}$$

Since $n_u = n_g$, the system has no completely unconstrained degrees of freedom, so there are no variables CV^0 that are always controlled. The gradient projections N_1 and N_2 for the two potentially unconstrained degrees of freedom become:

$$\begin{bmatrix} -0.04379 & -0.00241 \end{bmatrix} N_1 = 0 \Longrightarrow N_1 = \begin{bmatrix} -0.05499 \\ 0.9985 \end{bmatrix}$$

$$\begin{bmatrix} -0.1045 & 0.003268 \end{bmatrix} N_2 = 0 \Longrightarrow N_2 = \begin{bmatrix} 0.03126 \\ 0.9995 \end{bmatrix}$$

For MV-CV pairing, we choose $u_1 = F_B$ for controlling g_1 and $CV_1^{0g} = N_1^T \nabla_u \hat{J}$, and $u_2 = T_r$ controlling g_2 and $CV_2^{0g} = N_2^T \nabla_u \hat{J}$. This pairing choice was made based on the steady-state RGA for constraint control, which gives $\lambda = 0.638$ for the chosen pairing. For designing the selectors according to Theorem 2.3, a local analysis of the transformed constraint gains given in Equation (2.12) was made at the nominal point and is summarized in Table 2.5. The projected gains are

Variable	Value
F_A	0.5 kg/s
Δp_P	0
F_B	1.4587 kg/s
T_r	342.537 K
x_A	0.0712 kg/kg
x_B	0.4107 kg/kg
x_C	0.0173 kg/kg
x_P	0.1246 kg/kg
x_E	0.3 kg/kg
x_G	0.0762 kg/kg

Table 2.4: Nominal operating point for case study 2.

negative for both constraints regardless of the active set, which means that both selectors should be “max" selectors. The resulting control structure is presented in Figure 2.10.

$\mathcal{A}$	$(G^g_{P_\mathcal{A}})_{11}$	$(G^g_{P_\mathcal{A}})_{22}$
$\{\}$	-6.01×10^{-4}	-0.0279
$\{1\}$	-	-0.0287
$\{2\}$	-5.05×10^{-4}	-

Table 2.5: Diagonal of $G^g_{P_\mathcal{A}}$ for all relevant sets $\mathcal{A}$ for case study 2.

To tune the controllers, we obtained the following transfer functions from the MVs to the constraints (with time in hours):

$$G_{11}(s) = \frac{-0.1045}{0.225s + 1}, \quad G_{22}(s) = \frac{-0.00241}{0.072s + 1}$$

Based on this, the PI controllers for the constraints were tuned using the SIMC rules (Skogestad 2003) with $\tau_{C,1} = 0.005$ h and $\tau_{C,2} = 0.01$ h. Similar to the previous case study, the method for gradient estimation is again considered to be instantaneous with respect to the inputs, meaning that the gradient controllers become integral controllers, tuned using the SIMC rules with $\tau_C = 0.05$ h. All four controllers were implemented with anti-windup action based on back-calculation with a tracking time of $\tau_T = 0.01$ h. The controller tunings are summarized in Table 2.6.

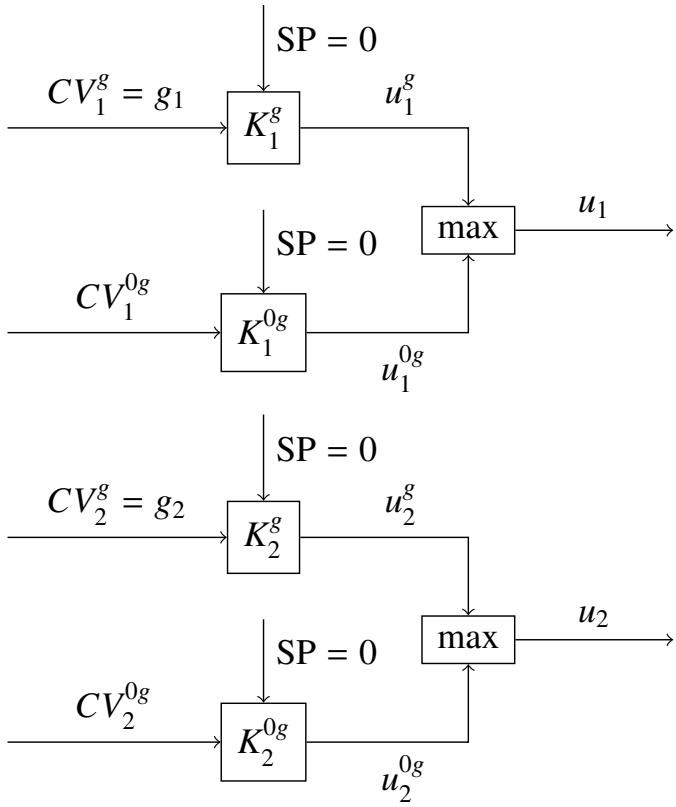


Figure 2.10: Complete control structure for case study 2.

Controller	Parameter	Value
K_1^g	K_c	-430.6
	τ_I	0.225 h
K_2^g	K_c	-2988
	τ_I	0.072 h
K_1^{0g}	K_I	-1.833
K_2^{0g}	K_I	202.5

Table 2.6: PI controller tunings for case study 2.

Closed-loop dynamic simulations are presented in Figure 2.11. The disturbances were changed so that all four active constraint regions were explored. Since we consider that the cost gradient $\nabla_u \hat{J}$ is an available measurement (from an estimator with a perfect model), operation in the fully unconstrained region (from $t = 21$ h to $t = 27$ h) is optimal at steady state, which can be seen by the input values converging to the exact steady-state optimal value. Since we assume that the constraints are directly measured (which is a mild assumption), the same logic applies to the fully constrained region from $t = 9$ h to $t = 15$ h. In addition, operation is optimal at the nominal point by design. In the two remaining partly constrained regions, the system does not converge exactly to the steady-state optimum, but the constraints are always satisfied (except for short dynamic transients, which may be avoided by introducing a back-off for the constraints).

It is interesting to note that for the third set of disturbances ($d = [1.0,\ -0.2]^T$ from $t = 6$ h to 9 h), the second constraint ($g_2 = 0$) is not controlled, even though it should be optimally controlled together with $g_1 = 0$. Instead, the selector logic results in the control of $CV_2^{0g} = 0$, which can be done without violation of g_2, that is, constraint g_2 is "over-satisfied". The reason for this non-optimal operation is that the selected value for projection matrix N_2 is not optimal in this operating region.

The steady-state economic loss is better visualized as a function of the disturbances in Figure 2.12. The highest losses are observed around where we ideally should switch between the partly constrained and the fully constrained regions. The optimal switch between these regions (black lines) does not coincide with the actual switch obtained with the selectors (blue lines). Economic loss is observed before the optimal switch due to the inaccuracy of the projection matrices. For the same reason, and because this further leads to suboptimal performance of the selectors, economic loss is also seen between the optimal and actual switch of CVs.

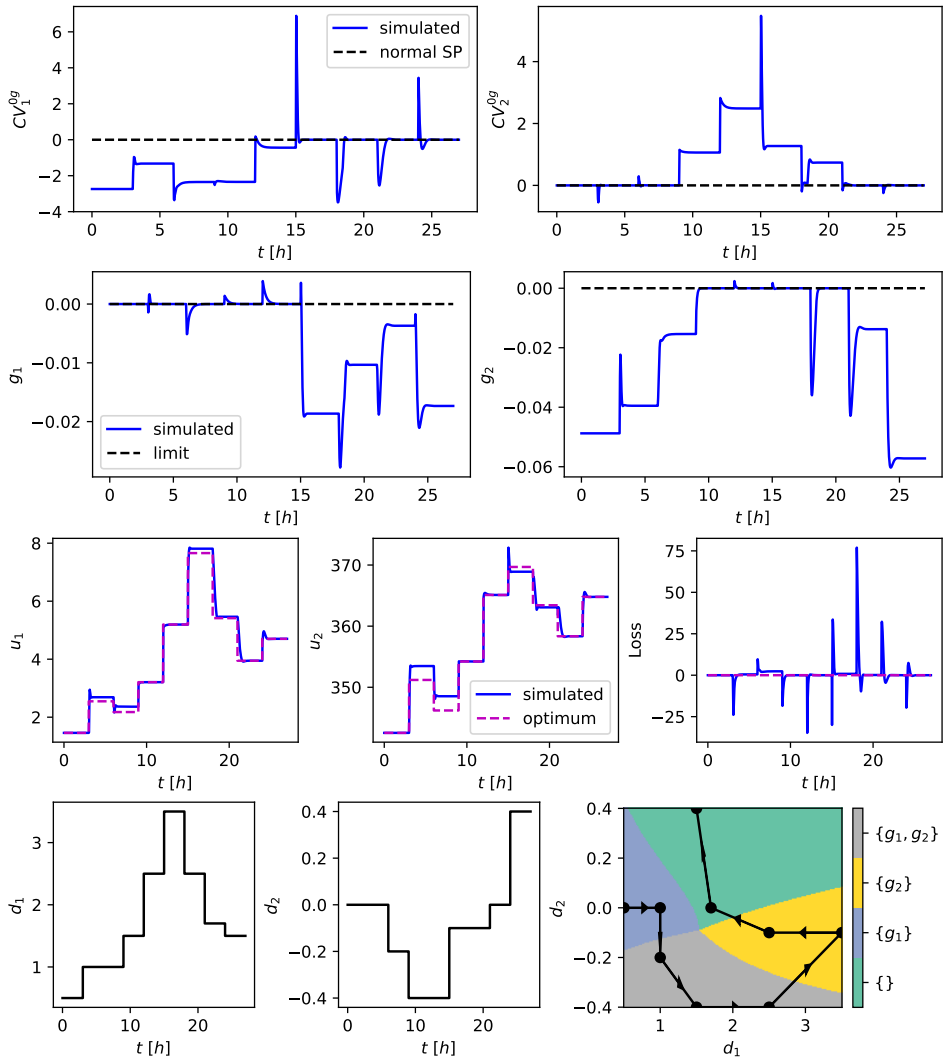


Figure 2.11: Closed-loop simulation results for case study 2.

However, the optimal switch between the fully unconstrained and the partly constrained regions coincides with the actual switch between the corresponding CVs. This happens because, before the switch, the full cost gradient $\nabla_u \hat{J}$ is controlled to zero, leading to zero economic loss, and the constraint becomes active immediately at the switch. Therefore, at this switch, the economic loss is zero, and it continuously grows as the system moves further into the partly constrained region.

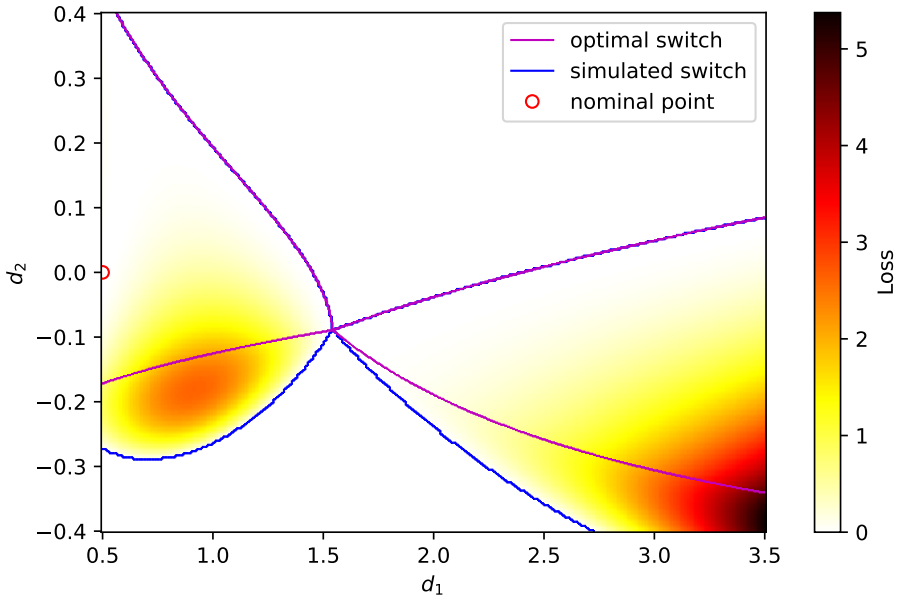


Figure 2.12: Steady-state closed-loop economic loss for case study 2.

2.5 Discussion

2.5.1 Steady-state cost gradient estimation

The results in this paper assume the availability of the steady-state cost gradient $\nabla_u J$ during operation. This can be fulfilled through model-based estimation, model-free estimation, or a combination of both methods (Krishnamoorthy and Skogestad 2022).Model-free methods usually depend on the perturbation of the inputs, and when the constraints are being controlled the perturbation can be done in their setpoints instead. In the presented case studies, we used a model-based approach, where a Kalman Filter was used to estimate the current dynamic state x and disturbance d with an augmented model (2.18), and then setting $\dot{x} = 0$ in the linearized model (2.16) leads to the gradient estimate (2.17).

Because the cost gradient $\nabla_u J$ is, by definition, a steady-state variable, it is not well-defined during a dynamic transition, and any gradient estimator must make some steady-state assumption or prediction. The necessity of estimating the cost gradient is related to ensuring exact optimality. In practice, one would wish to use an approximation of the cost gradient that is more easily implementable, even if that means accepting some economic loss. In that sense, data-driven approaches for this estimation would be appealing, as well as self-optimizing control methods that provide an approximation for the cost gradient through a static combination of measurements (Jäschke and Skogestad 2012).

However, a simpler approach is to use a static estimation of $\nabla_u J$ directly based on the measurements y. In another paper (Bernardino and Skogestad 2024a), we prove the optimality of a simple linear steady-state gradient estimate of the form

$$\nabla_u \hat{J} = Hy - c_s$$

where y are the measurements, and the constant vector c_s and the constant matrix H are obtained using the "exact local method" of self-optimizing control. In addition, a correction of c_s from a more accurate gradient estimator may be applied on a slower time scale, for example, using a model-based approach like RTO or a data-based perturbation method like extremum-seeking control.

2.5.2 Handling of constraints

Constraints in process systems are usually measured or estimated, and our approach is optimal for such cases, as control loops are implemented to handle these constraints. In MPC applications, where the problem is formulated as a dynamic trajectory optimization, it is common that process constraints are posed as constraints on the dynamic states, but this is not how our approach handles this issue. Rather, our method is focused on process constraints that may become active at steady state and influence process economics.

2.5.3 Updating of projection matrices

In the simulations presented in this paper, we assumed that a linearization of the constraints at a nominal operating point would be sufficiently accurate for capturing the transitions between active constraint regions. This simplification was primarily made to ensure a control structure that can be easily implemented, and it led to acceptable results even for a nonlinear case study (see Figure 2.12). However, it is possible to enhance economic performance by updating the projection matrices N and N^0 during operation. To accomplish this, an accurate estimator for the complete constraint gradient matrix G^g is required, and typically such an estimator is only available at a time scale similar to that of RTO. However, our

primary objective is to achieve acceptable economic losses in fast time scales, and this could be accomplished by using constant projection matrices.

2.5.4 Controller tuning

Even though the proposed control structure only has $n_u + n_g$ SISO controllers to be designed, the tuning of these controllers may prove to be challenging. This is because these controllers must work in many different regions (up to 2^{n_g} theoretical regions), and the interaction between loops will change depending on which controllers are active. The pairing between MVs and CVs should consider this, and the tuning for the loops should be robust in the sense that acceptable performance is attained for every operation mode. This issue was not noticed in the case studies in this work, but it is easy to see that it may arise in practice.

2.5.5 Limitations for systems with many constraints

In this paper, we consider a class of problems with $n_u \geq n_g$, so it is possible to devise a simple, decentralized control structure. There is a particular case of systems with more constraints than inputs that can fit into the framework proposed in this work. That would be the case where the constraints can be arranged into n_g groups, where each group is comprised of constraints that have parallel gain vectors with respect to the inputs, i.e. the constraints g_i and g_j would belong to the same group if $\nabla_u g_i = \alpha \nabla_u g_j$ for some nonzero α. In practice, this would represent a process variable with lower and upper bounds, or constraints of similar nature caused by different factors, e.g. a maximum processing rate due to upstream or downstream conditions. Each of these groups has a unique characteristic direction in terms of the rows of G^g, which can be used to calculate the gradient projections with the methodology described in Theorem 2.2. Each of these groups should then be organized internally following the single-variable methodology described by Krishnamoorthy and Skogestad (2020). As the methodology devised in this paper mitigates the correlation between each group of constraints, the gradient projections that serve as unconstrained CVs remain constant with respect to changes in the remaining loops, and therefore no additional logic is required in the implementation of max/min selectors for constraint handling.

The main case not covered by the present methodology is when there are $n_g > n_u$ *independent* constraints that may become active, expressly violating Assumption 2.1. In this case, considering that some pairing between MVs and constraints is done, the first problem that arises is the possibility of constraints paired with the same MV becoming active at the same time, requiring that one or several constraints become controlled by other MVs. A heat exchanger case with $n_g = 3 > n_u = 2$ was studied in Bernardino et al. (2022a), where it was shown that a region-

based approach similar to the one studied in the present paper fails to achieve optimality for some disturbance scenarios, whereas the primal-dual approach always reaches the optimal steady state. To achieve optimality with a region-based approach, an adaptive pairing strategy may be used, as described for this case study in Bernardino et al. (2022b). This gives optimal operation for all disturbances, but the adaptive pairing becomes quite complicated (see Figure 2 in (Bernardino et al. 2022b)).

In this sense, general strategies for switching pairings require more complex logic, and currently, there is no systematic arranging of classical control logic blocks that can account for that. On top of that, even if conflicting constraints are not an issue for the considered operating window, i.e. constraints paired to the same MV do not become active at the same time for the considered disturbances, the design of controllers for the unconstrained degrees of freedom becomes more complicated. As the constraints are assumed to be independent, the gradient projections optimally controlled to zero will be different when each of them is active. This entails that the remaining control loops have to change depending on which controller related to this MV is active. Therefore, proposing decentralized control structures for the optimal operation of systems with more constraints than MVs inevitably leads to complex and interacting control loops, and centralized strategies such as the primal-dual feedback optimizing control presented in Krishnamoorthy (2021) or MPC become more appealing.

For the same reason, the proposed framework has limitations in optimal dealing with input saturation. In real systems, every MV has physical bounds $u^{lb} \leq u \leq u^{ub}$ in addition to the process constraints g. Therefore, every physical system in a way has more constraints than MVs, and one must identify the constraints that are more likely to become active if one follows Theorem 2.2 for designing a control structure. The choice of not pairing an MV that may saturate with an important CV, in this case, an economic constraint, agrees with the rule of thumb “pair an MV that may saturate with a CV that may be given up" (Skogestad 2004), as the gradient projections paired to that MV should by design be given up in case of MV saturation.

The proposed framework attains optimal operation in a wide operating range, by enforcing optimality conditions at steady state for all possible active set combinations for a maximum of n_u independent constraints. It should also be emphasized that less frequent constraints can still be dealt with in the current framework by the implementation of more selectors, even if $n_g > n_u$, bearing in mind that steady-state optimal operation will not be guaranteed when those become active due to changes in the unconstrained CVs. However, violation of such infrequent constraints would be prevented, which is the main goal of such additional control

loops.

2.5.6 Stability and optimal convergence of selectors

The control strategy proposed in this work relies on switching blocks to perform optimal operation in different operating regions. Analyzing the stability of switched systems is more complex, as the stability of a switched system may not necessarily match that of its corresponding continuous subsystems (Liberzon and Morse 1999). In Theorem 2.3, we assume that each subsystem within the switching system is stable, which is a condition already present and well described when using decentralized control in multivariable systems. By ensuring that every subsystem is stable, the overall stability of the switching system can be guaranteed. This can be achieved by enforcing a sufficiently large average dwell time (Lin and Antsaklis 2009), which is a practical and easily implementable solution.

The conditions for implementing min/max selectors to detect switches in active constraints optimally are outlined in Theorem 2.3. This theorem is based on a local analysis of the optimization problem and is rigorously applicable to problems with a constant positive definite Hessian J_{uu} and constant constraints gain G^g. A relevant case in practice is that of linear economic objectives, for which J_{uu} is positive semidefinite, but this case is always solved by active constraint control, as there are no unconstrained degrees of freedom to be determined. While the presented proof does not address generic nonlinear optimization problems, it provides a useful local test that can eliminate certain impossible configurations resulting from the chosen MV-CV pairings or the formulation of the optimal operation problem itself. If the conditions specified in Theorem 2.3 are not satisfied, we recommend utilizing the cascade framework presented in Figure 2.4.

The condition derived in Theorem 2.3 for applicability of selectors would only be violated by highly interacting systems, where the sign of the transformed constraint gain ($G^g_{P_\mathcal{A}}$) would change depending on the active loops. This condition is conjectured to be associated with the decentralized integral controllability (DIC) of each potential subsystem (Lee and Edgar 2002). In our study, we could not find an example of a linear system with a convex objective function and without DIC that does not satisfy the conditions stated in the selector theorem. This further suggests a connection between these concepts and that the DIC conditions possibly imply the applicability of selectors. The link between the conditions of Theorem 2.3 and controllability aspects remains an open challenge that requires further investigation.

2.5.7 Other switching approaches

In this work, we have proposed the use of selectors in the controller outputs for detecting switches in active constraints. However, other strategies for adaptively controlling constraints in the context of optimal operation have been proposed. Manum and Skogestad (2012) studied the problem of active constraint switching in self-optimizing control by tracking the self-optimizing CVs in neighboring regions, where the switching happens when there is a change of sign in the monitored variable. In the notation herein presented, this would be equivalent to the following switching logic:

- If $CV_i^{0g} = N_i^T \nabla_u J$ is being controlled to zero, a change of sign in g_i means that the i-th constraint became active, as this sign change corresponds to constraint violation;
- Conversely, if g_i is being controlled to zero, a change of sign in $CV_i^{0g} = N_i^T \nabla_u J$ means that the i-th constraint became inactive, as this sign change corresponds to a change in the objective function slope.

The problem with implementing such logic lies in the resulting dynamics of the control system. As this logic implies that the reference variable is perfectly controlled for accurate detection, the logic should operate in a slower time scale than that of the closed-loop system, which would in turn result in undesired behavior, especially constraint violation. Operating the switching logic in fast time scales could in turn lead to the appearance of limit cycles, due to self-sustained switching between control loops.

We have also presented the cascade control structure in Figure 2.4 as an alternative switching strategy. A similar idea has been proposed by Cao (2004) to promote self-optimizing operation at the nominal region while sub-optimally coping with constraint satisfaction. There are however some disadvantages to this approach related to the limitations that the cascade structure imposes. If constraint control is slow, controlling the corresponding gradient projection becomes unnecessarily slow. Moreover, even though constraint control may help with decoupling the system, it may also cause the opposite problem, and the interaction between loops may impose limitations on the performance of the upper layer. Therefore, the use of a cascade framework for optimal operation may be beneficial, but the improvement that it may bring must be assessed for each particular case study.

Recently, the work of Ye et al. (2023) has tackled the problem of changing active constraints by embedding the switching constraints into the CV design, generating

a single nonlinear CV. Because the resulting CV design problem was deemed intractable in most cases, a neural network was used to approximate the behavior of this theoretical CV. It is interesting to note that the switching behavior still happens in the designed CV, with the exact ideal CV being in general non-smooth. This is expected because of the nature of the problem, and although neural networks can approximate these variables, the interpretability of the resulting CV is lost, and constraint control must be explicitly performed elsewhere. In the present work, we deal with the switching explicitly, controlling the constraints directly when it is optimal.

2.6 Conclusion

We propose a simple framework for decentralized optimizing control with changing active constraints. The starting point is that at steady state, optimal economic operation in a given active constraint region $\mathcal{A}$ is achieved by keeping the controlled variables $CV = \left[g_{\mathcal{A}}; N_{\mathcal{A}}^T \nabla_u J(u^*, d)\right]$ at constant setpoints $CV^{sp} = 0$ (Theorem 1 Jäschke and Skogestad (2012)). Here $g_{\mathcal{A}}$ denotes the set of active steady-state constraints, and $N_{\mathcal{A}}^T \nabla_u J(u^*, d)$ is the reduced steady-state cost gradient for the remaining unconstrained degrees of freedom.

There are some degrees of freedom in the choice of the directions in the unconstrained nullspace $N_{\mathcal{A}}$ and to implement constraint switching in a simple manner, these should be chosen in accordance with the constraint directions. The main contribution of this paper is to prove in Theorem 2.2 for the case with $n_u \geq n_g$, that we should control the unconstrained variables $CV^0 = N^{0T} \nabla_u J$ (which are not affected by the constraints), and in addition, depending on whether the constraint g_i is active or not, either control the constraint $CV_i^g = g_i$ or the associated unconstrained variable $CV_i^{0g} = N_i^T \nabla_u J$, where N^0 and N_i are calculated according to (2.7)-(2.9). This can be implemented with the simple control structure in Figure 2.2. Furthermore, Theorem 2.3 shows that the switching can be performed with min/max selectors, which leads to the simple control structure in Figure 2.3. Here, no centralized supervisor is needed to determine the active constraints, as the switching logic uses local feedback controllers.

Acknowledgements

The authors thank Johannes Jäschke for fruitful discussions.

Funding

This work was funded by the Research Council of Norway through the IKTPLUSS programme (project number 299585).

2.A Proof of Theorem 2.3

Proof. Define a set $A \subset \{1, 2, \cdots, n_g\}$ and an index i such that $i \notin A$, and another set A^* such that $A^* = A \cup \{i\}$. We prove the theorem by comparing the solution of the optimization problems with active sets A^* and A, i.e. what is the effect of controlling $g_i = 0$ instead of the corresponding unconstrained degree of freedom $CV_i^{0g} = N_i^T \nabla_u J = 0$ for arbitrary A and i.

The optimality conditions for A^* as the active set are given by:

$$\begin{cases} \nabla_u \mathcal{L} = J_{uu} u^{A^*} + J_{u,m}(d) + (G^g_{A^*})^T \lambda^{A^*} = 0 \\ g_{A^*}(u^{A^*}, d) = G^g_{A^*} u^{A^*} + g^m_{A^*}(d) = 0 \end{cases}$$

in which the constraint-related matrices are partitioned with relation to the active set A and the remaining index i as $G^g_{A^*} = \begin{bmatrix} (G^g_A)^T & (G^g_i)^T \end{bmatrix}^T$ and $g^m_{A^*}(d) = \begin{bmatrix} g^m_A(d)^T & g^m_i(d)^T \end{bmatrix}^T$.

Eliminating u^{A^*} from the first equation gives:

$$u^{A^*} = -J_{uu}^{-1} J_{u,m}(d) - J_{uu}^{-1} (G^g_{A^*})^T \lambda^{A^*}$$

Substituting this equation into the second optimality condition gives the following expression for λ^{A^*}:

$$(G^g_{A^*} J_{uu}^{-1} (G^g_{A^*})^T) \lambda^{A^*} = g^m_{A^*}(d) - G^g_{A^*} J_{uu}^{-1} J_{u,m}(d)$$

The variables can be partitioned as follows:

$$\begin{bmatrix} G^g_A J_{uu}^{-1} (G^g_A)^T & G^g_A J_{uu}^{-1} (G^g_i)^T \\ G^g_i J_{uu}^{-1} (G^g_A)^T & G^g_i J_{uu}^{-1} (G^g_i)^T \end{bmatrix} \begin{bmatrix} \lambda^{(A)} \\ \lambda^i \end{bmatrix} = \begin{bmatrix} g^m_A(d) - G^g_A J_{uu}^{-1} J_{u,m}(d) \\ g^m_i(d) - G^g_i J_{uu}^{-1} J_{u,m}(d) \end{bmatrix}$$

The same procedure with A as the active set leads to:

$$\begin{cases} u^A = -J_{uu}^{-1} J_{u,m}(d) - J_{uu}^{-1} (G^g_A)^T \lambda^A \\ (G^g_A J_{uu}^{-1} (G^g_A)^T) \lambda^A = g^m_A(d) - G^g_A J_{uu}^{-1} J_{u,m}(d) \end{cases}$$

The term $(G^g_{A^*})^T \lambda^{A^*} = (G^g_A)^T \lambda^{(A)} + (G^g_i)^T \lambda^i$ can be expressed in terms of the solution for A, as follows:

$$\begin{aligned}\lambda^{(A)} &= (G_A^g J_{uu}^{-1}(G_A^g)^T)^{-1}\left(g_A^m(d) - G_A^g J_{uu}^{-1} J_{u,m}(d) - G_A^g J_{uu}^{-1}(G_i^g)^T\lambda^i\right)\\ &= \lambda^A - (G_A^g J_{uu}^{-1}(G_A^g)^T)^{-1} G_A^g J_{uu}^{-1}(G_i^g)^T\lambda^i\\ \implies (G_{A^*}^g)^T\lambda^{A^*} &= (G_A^g)^T\lambda^A + \left(I - (G_A^g J_{uu}^{-1}(G_A^g)^T)^{-1} G_A^g J_{uu}^{-1}\right)(G_i^g)^T\lambda^i\end{aligned}$$

Therefore, u^{A^*} can be expressed in terms of u^A as follows:

$$\begin{aligned}u^{A^*} &= -J_{uu}^{-1} J_{u,m}(d) - J_{uu}^{-1}(G_{A^*}^g)^T\lambda^{A^*}\\ &= u^A - J_{uu}^{-1}\left(I - (G_A^g J_{uu}^{-1}(G_A^g)^T)^{-1} G_A^g J_{uu}^{-1}\right)(G_i^g)^T\lambda^i\end{aligned}$$

The transformation $P_{\mathcal{A}} = J_{uu}^{-1}\left(I - (G_A^g J_{uu}^{-1}(G_A^g)^T)^{-1} G_A^g J_{uu}^{-1}\right)$ is equivalent to a scaled projection to the nullspace of G_A^g, $N_{(\mathcal{A})}$ (Nocedal and Wright 2006), according to the identity:

$$P_{\mathcal{A}} = N_{(\mathcal{A})}\left(N_{(\mathcal{A})}^T J_{uu} N_{(\mathcal{A})}\right)^{-1} N_{(\mathcal{A})}^T = J_{uu}^{-1}\left(I - (G_A^g J_{uu}^{-1}(G_A^g)^T)^{-1} G_A^g J_{uu}^{-1}\right)$$

$P_{\mathcal{A}}$ is therefore positive semidefinite. The effect of inclusion of an arbitrary constraint g_i is therefore given by:

$$u^{A^*} - u^A = -P_{\mathcal{A}}(G_i^g)^T\lambda^i \tag{2.21}$$

We can see that the i-th component of the vector $P_{\mathcal{A}}(G_i^g)^T$ dictates the steady-state behavior of the i-th MV when the i-th constraint becomes active and the system is operating at the active set A. Following the notation introduced in the statement of Theorem 2.3 and in Figure 2.3, we have $u_i^g = (u^{A^*})_i$ and $u_i^{0g} = (u^A)_i$, since these are the MV values such that $g_i = 0$ and $CV_i^{0g} = N_i^T\nabla_u J = 0$, respectively. This means that, for $(P_{\mathcal{A}}(G_i^g)^T)_i > 0$, $u_i^g - u_i^{0g} < 0$ when $\lambda^i > 0$ and consequently u_i^g should be selected, and $u_i^g - u_i^{0g} > 0$ when $\lambda^i < 0$ and consequently u_i^{0g} should be selected, meaning that we must use $u_i^* = \min(u_i^{0g}, u_i^g)$ for guaranteeing optimality. Similar analysis can be performed for $(P_{\mathcal{A}}(G_i^g)^T)_i < 0$, leading to $u_i^* = \max(u_i^{0g}, u_i^g)$. Since this analysis is performed for arbitrary i and $\mathcal{A} \not\ni i$, guaranteeing that $(P_{\mathcal{A}}(G_i^g)^T)_i = (G_{P_{\mathcal{A}}}^g)_{ii}$ has the same sign for any possible $\mathcal{A}$ is sufficient for guaranteeing the theorem statement, which completes the proof. □

2.B Optimality of min selectors in cascade structure

In this section, we restate Theorem 2.3 for the cascade case illustrated in Figure 2.4, proving its optimality. In this control structure, the manipulated variables

as seen from the higher layer can be represented at steady-state as:

$$v = \begin{bmatrix} g_1 \\ \vdots \\ g_{n_g} \\ u_{n_g+1} \\ \vdots \\ u_{n_u} \end{bmatrix} \implies \Delta v = \left[\begin{array}{c|c} G^g & \\ \hline 0_{(n_u-n_g)\times n_g} & I_{n_u-n_g} \end{array}\right] \Delta u = W \Delta u$$

With this change of variables, the transformed optimization problem becomes:

$$\begin{aligned} \min_{v} \quad & J(v,d) \\ \text{s.t.} \quad & v_i \leq 0, \quad i = 1, \cdots, n_g \end{aligned} \tag{2.22}$$

Note that we must require that W, the Jacobian for the change of variables, is full rank, such that the optimality conditions for Equation (2.22) and Equation (2.1) are equivalent. This is a mild assumption related to the steady-state controllability of the constraints in the lower layer with the chosen pairing, and it results in a transformed Hessian $J_{vv} = W^T J_{uu} W$ that is also positive definite. Also, for the transformed problem, we have the gain matrix $G^{g,v}$ from the transformed inputs to the constraints:

$$G^{g,v} = \begin{bmatrix} I_{n_g} & 0_{n_g \times (n_u - n_g)} \end{bmatrix}$$

This allows us to write the optimal CVs in terms of the projection matrices N_i and N^0 for the transformed problem as:

$$N^0 = \begin{bmatrix} 0_{n_g \times (n_u - n_g)} \\ I_{n_u - n_g} \end{bmatrix}$$

$$N_i = \hat{e}_i, \quad i = 1, \cdots, n_g$$

The procedure for obtaining the difference in the optimal solution for neighboring regions presented in 2.A is also valid for the transformed problem, and we must therefore analyze the sign of the diagonal of the matrix product $G^{g,v} P_{\mathcal{A}}$. Recall that $P_{\mathcal{A}} = N_{(\mathcal{A})} \left(N_{(\mathcal{A})}^T J_{vv} N_{(\mathcal{A})} \right)^{-1} N_{(\mathcal{A})}^T$, where the matrix $N_{(\mathcal{A})}^T J_{vv} N_{(\mathcal{A})}$ is positive definite, being here a principal submatrix of J_{vv} which selects the rows and

columns with indexes not in $\mathcal{A}$. Therefore, $P_{\mathcal{A}}$ becomes a positive seimdefinite $n_u \times n_u$ matrix, where the diagonal elements $P_{\mathcal{A}_{ii}}$ are zero for $i \in \mathcal{A}$ and positive for $i \notin \mathcal{A}$.

Finally, we can see that $(G^{g,v}P_{\mathcal{A}})_{ii} > 0$ for $i \notin \mathcal{A}$, since the first n_g elements of the diagonal of $G^{g,v}P_{\mathcal{A}}$ are the same as those of $P_{\mathcal{A}}$. It follows that, according to Equation (2.21) and the results here obtained, the optimal solution is given by $v_i^* = \min(v_i^{0g}, 0)$ for $i = 1, \cdots, n_g$, which completes the proof.

Chapter 3

Region-based model predictive control for self-optimizing operation

This chapter has been submitted as a full paper:

> L. F. Bernardino and S. Skogestad. Optimal switching of MPC cost function for changing active constraints. *Submitted to Journal of Process Control*, 2024b

3.1 Introduction

Model predictive control (MPC) denotes a class of control strategies based on the online optimization of the predicted dynamic trajectory of the system (Rawlings 2000). It is a valuable tool for process control, being able to deal with multivariable interactions and constraint satisfaction. In practice, MPC is usually implemented as a supervisory control layer above the plant regulatory layer, where stability is assessed, and is subordinate to a real-time optimization (RTO) layer, which updates plant operation based on economics, as presented in Figure 3.1. It is possible to combine the RTO and control (MPC) layers into one; this is commonly known as Economic MPC (Ellis et al. 2014) and is not considered in this paper.

The steady-state economic optimization of the plant, solved at the RTO layer, can be defined as the following constrained optimization problem:

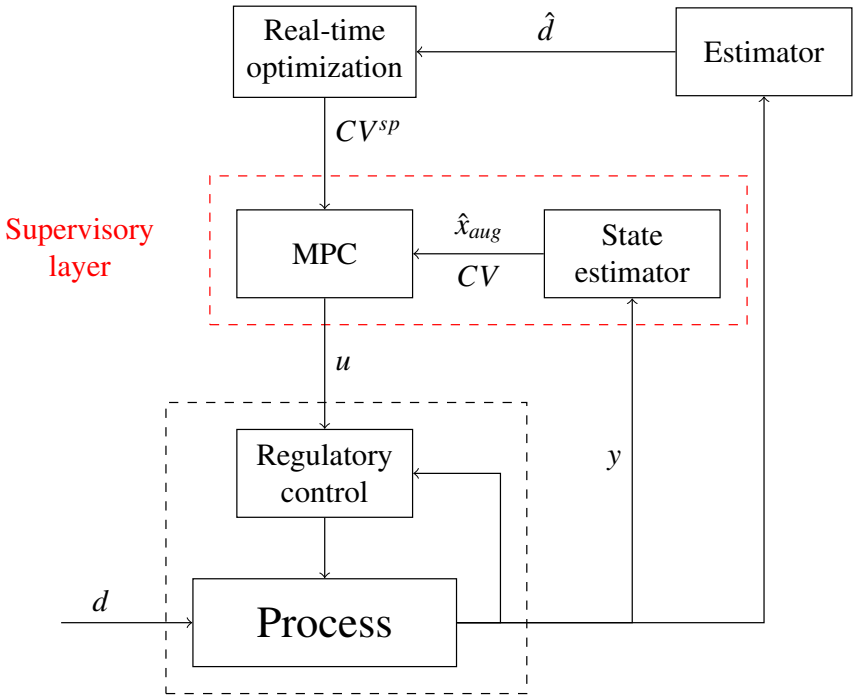


Figure 3.1: Typical hierarchical control structure with standard setpoint-tracking MPC in the supervisory layer. The cost function for the RTO layer is J^{ec} and the cost function for the MPC layer is J^{MPC}. With no RTO layer (and thus constant setpoints CV^{sp}), this structure is not economically optimal when there are changes in the active constraints. For smaller applications, the state estimator may be used also as the RTO estimator, provided there is an accurate dynamic model.

$$\begin{aligned} \min_{u} \quad & J^{ec}(u,d) \\ \text{s.t.} \quad & g(u,d) \leq 0 \end{aligned} \tag{3.1}$$

where $u \in \mathbb{R}^{n_u}$ is the vector of inputs or manipulated variables (MVs), $d \in \mathbb{R}^{n_d}$ is the vector of disturbances, J^{ec} is the scalar economic cost function, and $g(u,d) \in \mathbb{R}^{n_g}$ is the vector of inequality constraints. Note that the model equations and correspondent states have been formally eliminated from the formulation. The set of active constraints $\mathcal{A}$ is defined for the optimal solution u^* as the set for which $g_i(u^*,d) = 0$ with $i \in \mathcal{A}$.

Solving the problem in (3.1) results in the optimal plant inputs u^*, but as shown in Figure 3.1 the RTO layer implements the optimization results in the form of setpoint updates CV^{sp} to the MPC layer. We refer to this type of implementation as setpoint-tracking MPC, or standard MPC. As discussed in more detail later, see Equation (3.5), standard MPC uses a cost function of the form:

$$J^{MPC} = \sum_{k=1}^{N} ||CV_k - CV^{sp}||_Q^2 + ||\Delta u_k||_R^2$$

where the first term penalizes setpoint deviations and the last term penalizes dynamic input changes.

Standard MPC has two main elements: a state estimator and an open-loop moving horizon optimizer (which is often referred to simply as MPC). The state estimator ensures feedback, correcting the internal model according to the measurements, and the MPC uses that information to calculate the input sequence that drives the internal model to the desired operating point. The MPC problem can accommodate constraint satisfaction, either as direct constraints in the optimization problem or through the use of penalty terms. Additionally, one may consider a target calculation block, which ensures that the setpoint that the MPC tracks is feasible at steady state (Rawlings 2000).

In most standard MPC implementations, the CVs are selected based on process intuition, and not in a systematic manner. In this context, self-optimizing control (SOC) provides useful tools for systematic selection of CVs, having optimal steady-state operation as the main goal (Skogestad 2000, Jäschke et al. 2017). This gives the controlled variables:

$$CV = Hy$$

where y denotes the available process measurements (including selected inputs and measured disturbances) and H is a selection or combination matrix. Most

SOC approaches for CV selection assume that the steady-state active constraint set $\mathcal{A}$ is constant (Jäschke et al. 2017).

Graciano et al. (2015) implemented MPC using nominal self-optimizing CVs, that is, with the nominally active constraints. This can reject disturbances in fast timescales and minimize the nominal economic loss without the intervention of the RTO layer, at the same time avoiding violation of constraints. This is relatively simple to implement, but it cannot be regarded as self-optimizing in a broad sense, because the optimal approach is to use different self-optimizing CVs for each set of active steady-state constraints (Jäschke and Skogestad 2012).

This work proposes a framework for self-optimizing control under changing active constraints, which we label "region-based MPC", see Figure 3.2. Here, the self-optimizing CVs tracked by MPC are a function of the detected active constraint set. The constraint switching is based on the work of Woodward et al. (2010) and on the self-optimizing CVs here used. In three case studies, we show that standard MPC with a single (nominal) set of CVs leads to economic loss when there are changes in active constraints during operation, and we show that the proposed MPC framework attains steady-state optimal operation if the design conditions of SOC are met, and near-optimal operation in a broader sense.

The rest of the paper is organized as follows. In Section 3.2 we present some basic notions of MPC implementation. In Section 3.3 we describe the control structure proposed in this work, and the results of its application in some case studies are presented in Section 3.4. Based on these results and the theoretical aspects of the control structure, we discuss the proposed framework in Section 3.5, and the paper is then concluded in Section 3.6.

3.2 Standard MPC implementation

We first briefly discuss the standard MPC implementation represented in Figure 3.1, which includes a state estimator and an open-loop optimizer (MPC block). For the estimator, consider the following dynamic model used as an internal model for MPC:

$$\begin{cases} \dfrac{dx}{dt} = f(x, u, d_c) \\ y = h(x, u, d_c) \end{cases} \tag{3.2}$$

Here, $x \in \mathbb{R}^{n_x}$ represents the vector of dynamic states, $u \in \mathbb{R}^{n_u}$ the vector of inputs (MVs), $y \in \mathbb{R}^{n_y}$ the vector of measurements and $d_c \in \mathbb{R}^{n_{d_c}}$ a vector of model disturbances.

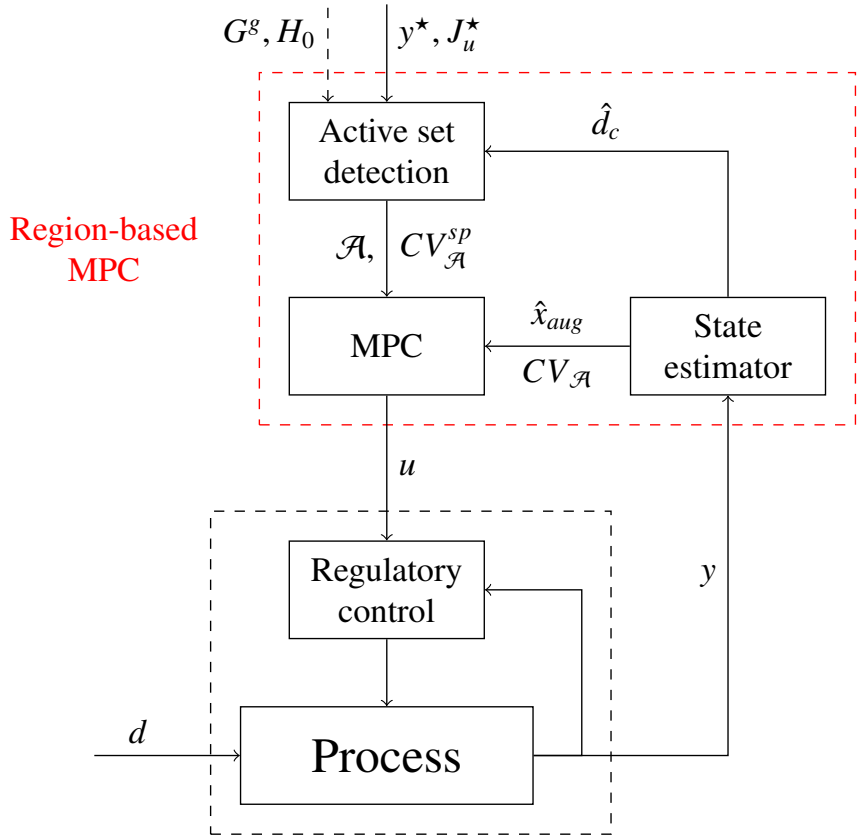


Figure 3.2: Proposed region-based MPC structure with active set detection and change in controlled variables. The possible updates from an upper RTO layer (y^*, J_u^* etc.) are not considered in the present work. Even with no RTO layer (and thus with constant setpoints $CV_{\mathcal{A}}^{sp}$, see (3.14) and (3.15), in each active constraint region), this structure is potentially economically optimal when there are changes in the active constraints.

There is usually one d_c for each controlled variable, which is used to account for uncertainty, for example, related to the "true" disturbances d, measurement bias, or model parameter changes. Note that d_c does not need to have a physical interpretation and is used mainly to include integral action in MPC. In other words, to attain offset-free control, the internal model is augmented with the integrating states d_c:

$$x_{aug} = \begin{bmatrix} x \\ d_c \end{bmatrix} \tag{3.3}$$

For a linear internal model, the number of additional integrating states must be at least the number of controlled variables, and it need not be greater than the number of measurements (Maeder et al. 2009). The dynamic model considered by the state estimator is therefore of the form:

$$\begin{cases} \dfrac{d\hat{x}_{aug}}{dt} = \dfrac{d}{dt}\begin{bmatrix} \hat{x} \\ \hat{d}_c \end{bmatrix} = \begin{bmatrix} f(\hat{x}, u, \hat{d}_c) \\ 0 \end{bmatrix} + \omega \\ \hat{y} = h(\hat{x}, u, \hat{d}_c) + \nu \end{cases} \tag{3.4}$$

where $\omega \sim \mathcal{N}(0, Q^e)$ and $\nu \sim \mathcal{N}(0, R^e)$ are the random variables present in most state estimation frameworks, and Q^e and R^e are the corresponding tuning parameters (Simon 2006). The estimated states $\hat{x}$ and $\hat{d}_c$ are then used to solve a moving-horizon optimization problem, which results in the next control action to be implemented. A simple discretized MPC optimization problem can be of the form:

$$\begin{aligned} \min_{u_k, x_k} \quad & J^{MPC} = \sum_{k=1}^{N} ||CV_k - CV^{sp}||_Q^2 + ||\Delta u_k||_R^2 \\ \text{s.t.} \quad & x_k = \phi(x_{k-1}, u_{k-1}, \hat{d}_c) \\ & y_k = h(x_k, u_k, \hat{d}_c) \\ & CV_k = Hy_k \\ & \Delta u_k = u_k - u_{k-1} \\ & x_0 = \hat{x} \\ & y_{min} \leq y_k \leq y_{max} \\ & x_{min} \leq x_k \leq x_{max} \\ & u_{min} \leq u_k \leq u_{max} \\ & -\Delta u_{max} \leq \Delta u_k \leq \Delta u_{max} \end{aligned} \tag{3.5}$$

Here, x_k denotes the state at the k-th time step, and $\phi(x_{k-1}, u_{k-1}, \hat{d}_c)$ is the result of the integration of the dynamic model (3.2) from t_{k-1} to $t_k = t_{k-1} + \Delta t$ with $u = u_{k-1}$, $d_c = \hat{d}_c$, and the initial condition as the previous state $x(t_{k-1}) = x_{k-1}$. The objective function J^{MPC} aims to minimize the tracking error $CV - CV^{sp}$ while penalizing large input changes Δu_k. N is the number of prediction steps, and Q and R are tuning matrices.

The output, state, and input constraints in (3.5) can be used to embed the RTO constraint $g(u, d) \leq 0$ (3.1) in the MPC time scale. State constraints are not needed if we assume that the constraints $g(u, d)$ are measured (or estimated) and included as elements in the measurements vector y. Without loss of generality, we will assume that the economic constraints can be estimated from the dynamic model as:

$$g = h_g(x, u, d_c) \tag{3.6}$$

We remark that an MPC in the form of Equation (3.5) has no stability guarantees, but it can converge if the prediction horizon N is large enough (the reader is referred to Mayne (2014) for an in-depth review of MPC formulations).

The focus of the present work is the case where the original setpoints CV^{sp} must be given up due to constraints becoming active at steady state. In theory, the RTO layer may update the setpoints, but in most cases there is no RTO layer, so the setpoints are constant. Standard MPC satisfies the constraints, but it is suboptimal in terms of steady-state economic performance, and we propose a better way of dealing with economic constraints on standard setpoint-tracking MPC, without the need for RTO updates.

3.3 Region-based MPC framework

The structure of the proposed region-based MPC scheme is summarized in Figure 3.2. The state estimator, also present in standard MPC, serves as the feedback element for MPC as well as for the active set detection block, which is the new element of the framework when compared to standard MPC. The detected active set $\mathcal{A}$ along with the setpoints, is sent to the MPC block, which uses a different set of CVs for each active set $\mathcal{A}$.

We next describe how the CVs for each active set are determined, and how the set of active constraints can be estimated online.

3.3.1 Controlled variables for MPC

The controlled variables for each region are defined such that the steady-state economic problem (3.9) is solved by feedback control, meaning that we adjust u to keep $CV = CV^{sp}$. These CVs are defined as:

$$CV_{\mathcal{A}} = \begin{bmatrix} g_{\mathcal{A}} \\ c_{\mathcal{A}} \end{bmatrix} \tag{3.7}$$

Here, $g_{\mathcal{A}}$ denotes the active constraints, and $c_{\mathcal{A}}$ denotes the unconstrained CVs for optimal operation. The control action calculation for the proposed region-based MPC is very similar to that of Equation (3.5), but the objective function changes for each $\mathcal{A}$ according to:

$$J_{\mathcal{A}}^{MPC} = \sum_{k=1}^{N} ||CV_{\mathcal{A}} - CV_{\mathcal{A}}^{sp}||_{Q_{\mathcal{A}}}^2 + ||\Delta u_k||_{R_{\mathcal{A}}}^2 \tag{3.8}$$

where $Q_{\mathcal{A}}$ and $R_{\mathcal{A}}$ are tuning parameters that can be chosen independently for each active set $\mathcal{A}$.

The unconstrained controlled variables $c_{\mathcal{A}}$ should be selected to minimize the steady-state cost, given that the active constraints $g_{\mathcal{A}}$ are being controlled. For that, we follow Halvorsen et al. (2003) and we consider a local QP approximation of the economic optimization problem of the form:

$$\begin{aligned} \min_{\Delta u} \quad & J^{ec} = J^{ec\star} + \begin{bmatrix} \Delta u^T & \Delta d^T \end{bmatrix} \begin{bmatrix} J_u^\star \\ J_d^\star \end{bmatrix} \\ & + \frac{1}{2} \begin{bmatrix} \Delta u^T & \Delta d^T \end{bmatrix} \begin{bmatrix} J_{uu} & J_{ud} \\ J_{ud}{}^T & J_{dd} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} \end{aligned} \tag{3.9a}$$

$$\text{s.t.} \quad g = g^\star + G^g \Delta u + G_d^g \Delta d \leq 0 \tag{3.9b}$$

Here, $\Delta d = d - d^\star$ and $\Delta u = u - u^\star$ represent the disturbances and inputs as their deviation from their reference values $d^\star$ and $u^\star$ respectively, and $J^{ec\star}$, $J_u^\star$, and J_{uu} represent respectively the cost function, its gradient, and its Hessian with respect to the inputs, evaluated at steady state at the reference point. Additionally, the measurements y can be locally represented by a linear steady-state model of the form:

$$\Delta y = G^y \Delta u + G_d^y \Delta d \tag{3.10}$$

The linearized expression (3.9b) for the constraints is not used by Halvorsen et al. (2003), but it is needed here because we consider changes in active constraints. The unconstrained CVs $c_{\mathcal{A}}$ are defined in terms of the CVs for the fully unconstrained region, c_0, which is itself a static linear combination of the measurements:

$$c_0 = H_0 y \tag{3.11}$$

Assuming that there are enough measurements ($n_y = n_u + n_d$), we can find an analytical expression for H_0 based on the nullspace method that is given by (Alstad et al. 2009, Jäschke and Skogestad 2011):

$$H_0 = \begin{bmatrix} J_{uu} & J_{ud} \end{bmatrix} \begin{bmatrix} G^y & G_d^y \end{bmatrix}^\dagger \tag{3.12}$$

which is used to obtain an optimal first-order estimate of the cost gradient J_u (Jäschke and Skogestad 2011):

$$\hat{J}_u = c_0 - c_0^{sp} = H_0(y - y^\star) + J_u^\star \tag{3.13}$$

The setpoint for the unconstrained CV, $c_0^{sp} = H_0 y^\star - J_u^\star$, is calculated based on the reference steady state and the corresponding value of the cost gradient. If the reference steady state is an optimal operating point at a fully unconstrained region, $J_u^\star$ is zero. However, if the reference steady state is not optimal, or if the system is operating at a constrained region, it is nonzero.

For the constrained case, the ideal self-optimizing variable is the reduced gradient $N_{\mathcal{A}}^T J_u$ (Jäschke and Skogestad 2012), and the CVs related to the unconstrained degrees of freedom are given by:

$$c_{\mathcal{A}} = N_{\mathcal{A}}^T H_0 y$$

where $N_{\mathcal{A}}$ is a projection matrix, defined as a basis for the nullspace of the active constraints gradient, i.e. $G_{\mathcal{A}}^g N_{\mathcal{A}} = 0$. Because $c_{\mathcal{A}}$ is a linear combination of the unconstrained CVs c_0, the corresponding setpoint will be given by $c_{\mathcal{A}}^{sp} = N_{\mathcal{A}}^T c_0^{sp}$. The full set of CVs for each active constraint region $\mathcal{A}$ then becomes:

$$CV_{\mathcal{A}} = \begin{bmatrix} g_{\mathcal{A}} \\ c_{\mathcal{A}} \end{bmatrix} = \begin{bmatrix} g_{\mathcal{A}} \\ N_{\mathcal{A}}^T H_0 y \end{bmatrix} \tag{3.14}$$

with the corresponding setpoints being:

$$CV_{\mathcal{A}}^{sp} = \begin{bmatrix} 0 \\ N_{\mathcal{A}}^T(H_0 y^\star - J_u^\star) \end{bmatrix} \tag{3.15}$$

This choice of CVs minimizes the steady-state economic loss around the reference point $(u^\star, d^\star)$. Furthermore, even if the new operating point is such that the active set $\mathcal{A}$ is different than that of the reference point, the use of $CV_{\mathcal{A}}$ minimizes the steady-state loss, as long as the approximations in Equations (3.9) and (3.10) hold. We now discuss how to detect the active set using the available measurements, so as to select the correct controlled variables.

3.3.2 Active constraint set detection

In the previous section, we estimated the cost gradient J_u as a function of the available measurements y. The same idea is now applied to active set detection. Woodward et al. (2010) describes an active set detection algorithm for a feedback optimizing strategy that only depends on the current value of the cost gradient J_u, the constraints g, and the constraints gradient G^g. Here, we adapt this strategy so that the method depends directly on the available measurements. In summary, we assume g to be directly measured, J_u is estimated using Equation (3.13), and G^g is assumed constant at its nominal value from Equation (3.9b).

In order to estimate J_u, which is the steady-state cost gradient, we use the value of the measurements at the expected steady state where the CVs are driven to their setpoints, which we call y^{ss}. This expected steady state can be determined using Equation (3.2), leading to:

$$\begin{cases} 0 = f(x^{ss}, u^{ss}, \hat{d}_c) \\ CV_{\mathcal{A}}(x^{ss}, \hat{d}_c) = CV_{\mathcal{A}}^{sp} \end{cases} \tag{3.16}$$

The steady-state measurements are then obtained as $y^{ss} = h(x^{ss}, u^{ss}, \hat{d}_c)$, along with the corresponding predicted constraint values from Equation (3.6) as $g^{ss} = h_g(x^{ss}, u^{ss}, \hat{d}_c)$. With these values, we are ready to apply the method from Woodward et al. (2010). The algorithm is summarized in Algorithm 1, and we shall explain its main steps.

The algorithm begins in step 1 by finding the expected steady-state measurements y^{ss} by solving Equation (3.16), and with it we find the predicted cost gradient $\hat{J}_u$ through Equation (3.13). Then, we include in step 2 the constraints predicted to be violated, i.e. $g_i^{ss} \geq 0$, into the estimated active set. With this augmented active set $\mathcal{A}$, we solve in step 3 the following optimization problem:

Algorithm 1 Active set estimation based on the MPC internal model, adapted from Woodward et al. (2010)

1: $\hat{J}_u \leftarrow H_0(y^{ss} - y^\star) + J_u^\star$ ▷ from (3.13)
2: $\mathcal{A}^k \leftarrow \mathcal{A}^{k-1} \cup \{\, i \mid g_i^{ss} \geq 0 \,\}$
3: $\delta u^* \leftarrow$ solution of (3.17)
4: $\mathcal{A}^k \leftarrow \{\, i \in \mathcal{A}^k \mid G_i^g \delta u^* = 0 \,\}$
5: **if** $n\left(\mathcal{A}^k\right) > n_u$ **then** ▷ too many active constraints
6: Find $\mathcal{A}' \subset \mathcal{A}^k \mid g^{ss}(CV_{\mathcal{A}'} = CV_{\mathcal{A}'}^{sp}) \leq 0$ ▷ re-solve (3.16)
7: $\mathcal{A}^k \leftarrow \mathcal{A}'$
8: **end if**

$$\delta u^* = \arg\min_{\delta u} \; -\delta u^T \delta u \quad \text{s.t.} \quad \begin{cases} G_{\mathcal{A}}^g \delta u \leq 0 \\ \delta u^T \delta u = -\delta u^T \hat{J}_u \end{cases} \tag{3.17}$$

With this problem, we wish to find the largest projection of the negative of the estimated cost gradient, $-\hat{J}_u$, onto the feasible directions, i.e. directions that do not violate $G_{\mathcal{A}}^g \delta u \leq 0$. The solution δu^* therefore dictates the best feasible descent direction for improving the economic cost function. In step 4, the inactive constraints at the solution ($G_i^g \delta u^* < 0$) are then removed from the active set $\mathcal{A}$, as controlling these constraints would hinder economic improvement.

Because this method does not account for infeasibility, an additional step to obtain the active set $\mathcal{A}$ sent to the controller is necessary. If the active set resulting from the previous operations has more than n_u elements, it is deemed infeasible, because the controller cannot track more than n_u variables with the available inputs. One must then pick a subset that is predicted to be feasible ($g_i^{ss} \leq 0 \; \forall i$) by evaluating the corresponding expected steady states with Equation (3.16), as represented in step 6. With this, the controller always tries to control a feasible set of active constraints. One may also add a constraint priority list, such that, when operation is infeasible for all candidate active sets, the less important constraint is given up.

The problem given in Equation (3.17) is presented as an NLP due to the quadratic equality constraint, but the work of Woodward et al. (2010) solves this problem using a specific algorithm. Here, we simply solve the optimization problem directly. To prevent premature switching due to the estimator dynamics, the estimated $\mathcal{A}$ is only used for switching CVs after N_{sw} time steps where the estimated $\mathcal{A}$ is different than the one being implemented in the controller, and N_{sw} becomes a tuning parameter that improves switching performance.

3.4 Case studies

For the following case studies, consider the continuous-discrete time objective function for MPC:

$$J_{\mathcal{A}}^{MPC} = \int_0^{N\Delta t} ||CV_{\mathcal{A}} - CV_{\mathcal{A}}^{sp}||_{Q_{\mathcal{A}}}^2 \, dt + \sum_{k=1}^{N} ||\Delta u_k||_{R_{\mathcal{A}}}^2 \tag{3.18}$$

This formulation is used for convenience, as the integration of the MPC objective function and the internal model are done together using orthogonal collocation. The resulting problem is solved using CasADi/IPOPT (Andersson et al. 2019).

3.4.1 Case study 1 - toy example

In this example, we illustrate the optimality of the proposed methodology for systems with quadratic cost function and linear dynamic and constraint models, which is sufficient for the exactness of the methodology described in Section 3.3, and the economic improvement when compared to a standard implementation of self-optimizing MPC.

The hypothetical system considered here has 2 dynamic states x and 3 MVs u, with economic objectives and constraints being represented by the following optimization problem:

$$\begin{aligned} \min_{u} \quad & \frac{1}{2} x^T \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} x + \frac{1}{2} u^T \begin{bmatrix} 1 & -0.1 & -0.2 \\ -0.1 & 0.8 & -0.1 \\ -0.2 & -0.1 & 0.3 \end{bmatrix} u \\ \text{s.t.} \quad & \begin{cases} g_1 = x_1 - 0.8 x_2 \leq 0 \\ g_2 = u_1 + u_2 + u_3 \leq 0 \end{cases} \end{aligned} \tag{3.19}$$

The dynamic states x are affected by the MVs u and the disturbances d according to the following linear state-space dynamic model:

$$\begin{cases} \dot{x} = \begin{bmatrix} -\frac{1}{\tau_1} & 0 \\ 0 & -\frac{1}{\tau_2} \end{bmatrix} x + \begin{bmatrix} \frac{0.2}{\tau_1} & 0 & 0 \\ 0 & \frac{0.2}{\tau_2} & 0 \end{bmatrix} u + \begin{bmatrix} \frac{1}{\tau_1} & 0 \\ 0 & \frac{1}{\tau_2} \end{bmatrix} d \\ y = \begin{bmatrix} g_1 \\ g_2 \\ x_2 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 & -0.8 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} u \end{cases} \tag{3.20}$$

with $\tau_1 = 1$ and $\tau_2 = 2$. For the vector of measurements y, as discussed in Section 3.3, we follow the convention of considering the constraints as direct measurements, figuring in the first two rows of the measurement vector. The remaining measurements are chosen with the goal of satisfying a sufficient number of independent measurements ($n_y = n_u + n_d$).

Using the information above, we may eliminate the state variables x, since at steady state, we have that:

$$x = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \end{bmatrix} u + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} d \tag{3.21}$$

We then write the steady-state optimization problem in the standard form:

$$\begin{aligned} \min_{u} \quad & J = \frac{1}{2}\, u^T \begin{bmatrix} 1.04 & -0.1 & -0.2 \\ -0.1 & 1.2 & -0.1 \\ -0.2 & -0.1 & 0.3 \end{bmatrix} u + u^T \begin{bmatrix} 0.2 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix} d \\ \text{s.t.} \quad & g = \begin{bmatrix} 0.2 & -0.16 & 0 \\ 1 & 1 & 1 \end{bmatrix} u + \begin{bmatrix} 1 & -0.8 \\ 0 & 0 \end{bmatrix} d \leq 0 \end{aligned} \tag{3.22}$$

along with the steady-state expression for the measurements:

$$y = \begin{bmatrix} 0.2 & -0.16 & 0 \\ 1 & 1 & 1 \\ 0 & 0.2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} u + \begin{bmatrix} 1 & -0.8 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} d \tag{3.23}$$

for which the matrices presented in Equations (3.9) and (3.10) are recognizable. This problem has two inequality constraints and three MVs, so optimal operation has always between one and three unconstrained DOFs. As the system has only two disturbances, we can graphically illustrate the active constraint regions as in Figure 3.3, where we can see all possible combinations of active constraints. This map of disturbances is not used in the method, and it is only made with the goal of visualizing the optimal operation mode for each disturbance.

For implementing a standard self-optimizing MPC controller with dynamic constraint handling, we follow the strategy described by Graciano et al. (2015). We design the self-optimizing CVs $c = Hy$ at the unconstrained region, choosing as the reference steady state the optimal value for $d^\star = \begin{bmatrix} -4; & +4 \end{bmatrix}$. We can therefore use the matrix $H = H_0$ as defined in Equation (3.12), leading to:

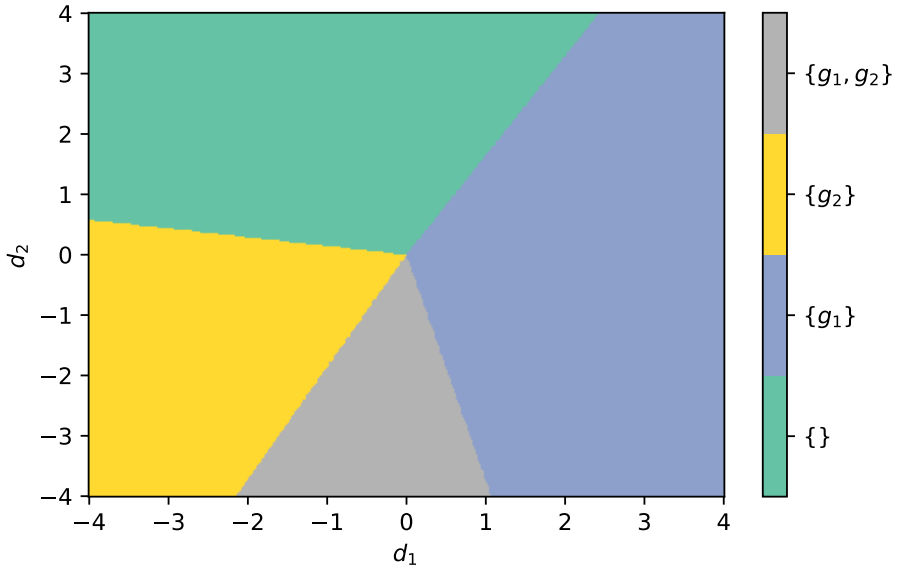


Figure 3.3: Active constraint regions for case study 1 as a function of disturbances

$$H_0 = \begin{bmatrix} 0.2 & 1 & 0.16 & -1.1 & -1.2 \\ 0 & -0.1 & 2 & 0.9 & 0 \\ 0 & -0.2 & 0 & 0.1 & 0.5 \end{bmatrix}$$

The measurements at the optimal operating point for $d^\star$ are $y^\star = [-6.17 \quad -9.86 \quad 2.62 \quad -6.91 \quad -2.56]^T$, and the corresponding setpoint is $c^{sp} = H_0 y^\star = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. The implementation of the region-based MPC controller leads to c_0 being this same set of CVs, and it depends additionally on the calculation of projection matrices $N_\mathcal{A}$ for the constrained regions. These are presented in Table 3.1. In terms of the tuning of the controllers, these are presented in Table 3.2, with the standard MPC using only the tuning corresponding to the fully unconstrained region ($\mathcal{A} = \{\}$).

The process was simulated at two different points inside each region. Results of the closed-loop simulation of both control strategies are given in Figure 3.4. The three CVs $c = H_0 y$ obtained in the unconstrained region ($\mathcal{A} = \{\}$) are not suitable for the constrained regions in terms of optimal operation. This is most easily seen by comparing the inputs (u_1, u_2, u_3) from standard MPC (green) with the optimal inputs (magenta). The inputs obtained with the proposed region-based MPC (blue) are optimal at steady state in all four regions, and the switching of CVs is seen to be smooth.

$\mathcal{A}$	$N_{\mathcal{A}}^T$
$\{\}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$\{1\}$	$\begin{bmatrix} 0.625 & 0.781 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$\{2\}$	$\begin{bmatrix} -0.577 & 0.789 & -0.211 \\ -0.577 & -0.211 & 0.789 \end{bmatrix}$
$\{1,\ 2\}$	$\begin{bmatrix} -0.362 & -0.453 & 0.815 \end{bmatrix}$

Table 3.1: Optimal gradient projections for case study 1

Parameter	$\mathcal{A}$	Value
$Q_{\mathcal{A}}$	$\{\}$	diag($[1, 1, 100]$)
	$\{1\}$	diag($[1, 1, 1]$)
	$\{2\}$	diag($[1, 1, 1]$)
	$\{1,\ 2\}$	diag($[1, 1, 1]$)
$R_{\mathcal{A}}$		diag($[0.01, 0.01, 0.01]$)
N		30
Δt		0.333
Q^e		diag($[0.05, 0.05, 1, 1]$)
R^e		diag($[0.01, 0.01, 0.01, 0.01, 0.01]$)

Table 3.2: Tuning of controllers and estimator for case study 1

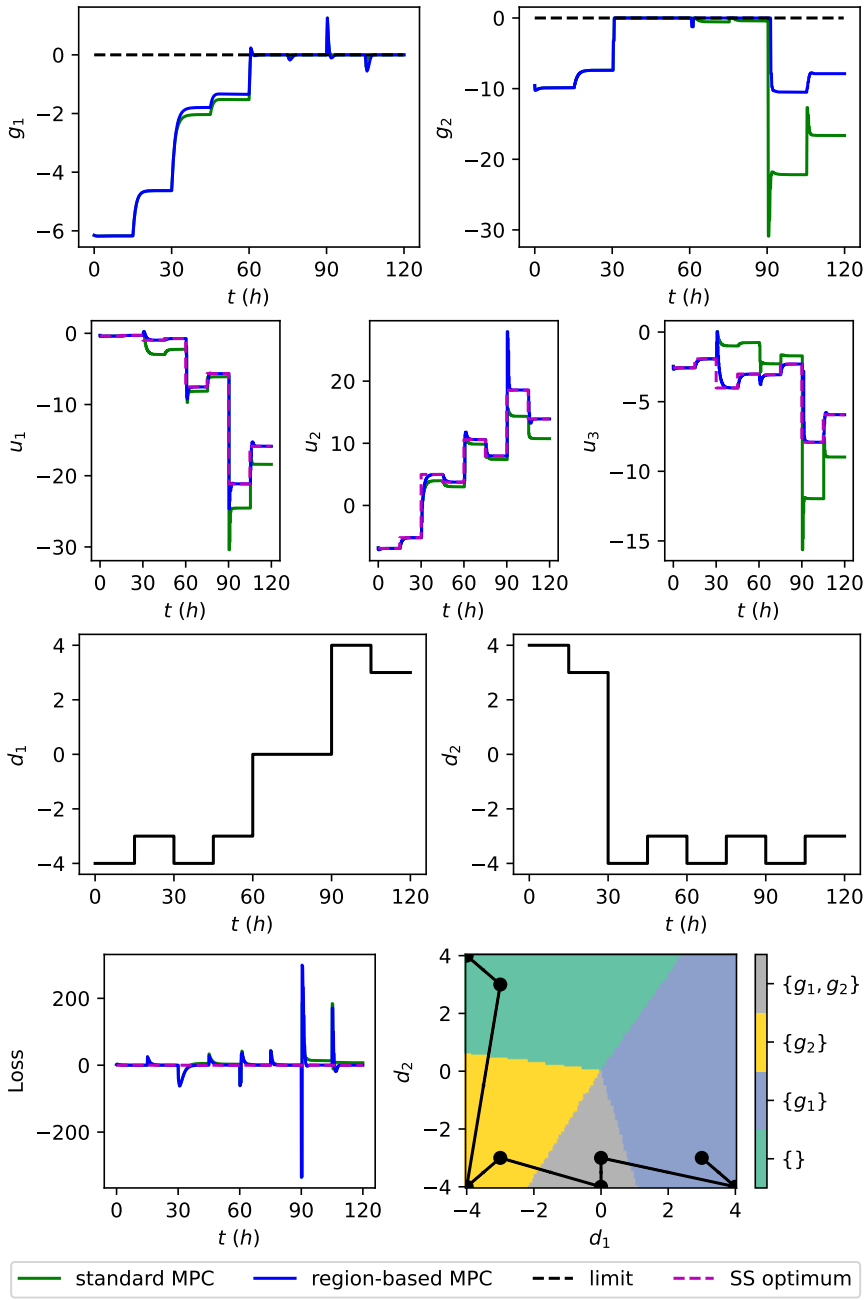


Figure 3.4: Dynamic simulation results for case study 1 - comparison between standard MPC (green) and the proposed region-based MPC (blue)

The economic loss of standard MPC is shown in more detail in Figure 3.5. It can be seen that the loss is nonzero whenever the system leaves the unconstrained region. Note that the lines delimiting the operating regions (blue) do not coincide with the optimal boundaries (magenta) at the partly constrained regions, because the use of fixed CVs is not optimal. Therefore, standard MPC does not guarantee control of the correct active constraints.

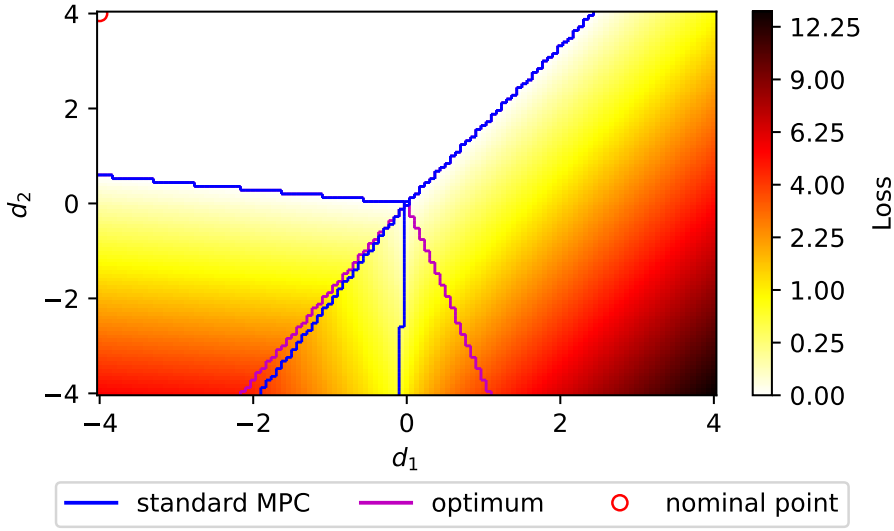


Figure 3.5: Economic loss of standard MPC for case study 1 as a function of disturbances. Magenta lines delimit optimal active constraint regions, blue lines delimit operating regions of standard MPC. Region-based MPC attains zero loss for all disturbance values.

It is worth mentioning that the proposed method does not rely on RTO updates for dealing with changes in active constraints. Instead, it relies on a switching logic for the CVs solely based on measurements and the nominal plant behavior, and on the self-optimizing property of the chosen CVs, which means that no setpoint updates are required for the system to operate optimally.

3.4.2 Case study 2 - Williams-Otto reactor

This case study is based on the process described by Williams and Otto (1960), see Figure 3.6. It consists of a continuously stirred reactor tank with perfect level control, in which A and B are mixed, generating the main product of interest P, along with the less interesting product E and the undesired byproduct G. The three reactions are:

$$
\begin{aligned}
A + B &\xrightarrow{k_1} C & k_1 &= k_{0,1} e^{-E_1/T_r} \\
C + B &\xrightarrow{k_2} P + E & k_2 &= k_{0,2} e^{-E_2/T_r} \\
P + C &\xrightarrow{k_3} G & k_3 &= k_{0,3} e^{-E_3/T_r}
\end{aligned}
$$

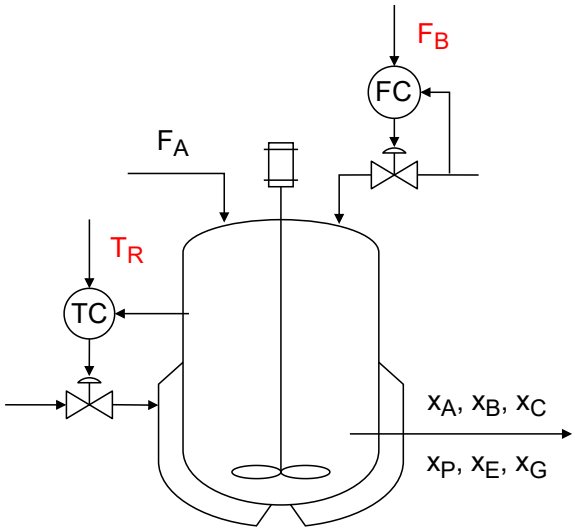


Figure 3.6: Schematic representation of Williams-Otto reactor, with MVs in red

The component mass balances result in the following system of ODEs:

$$\frac{dx_A}{dt} = \frac{F_A}{W} - \frac{(F_A + F_B)x_A}{W} - k_1 x_A x_B \tag{3.24a}$$

$$\frac{dx_B}{dt} = \frac{F_B}{W} - \frac{(F_A + F_B)x_B}{W} - k_1 x_A x_B - k_2 x_C x_B \tag{3.24b}$$

$$\frac{dx_C}{dt} = -\frac{(F_A + F_B)x_C}{W} + 2k_1 x_A x_B - 2k_2 x_C x_B - k_3 x_P x_C \tag{3.24c}$$

$$\frac{dx_P}{dt} = -\frac{(F_A + F_B)x_P}{W} + k_2 x_C x_B - 0.5 k_3 x_P x_C \tag{3.24d}$$

$$\frac{dx_E}{dt} = -\frac{(F_A + F_B)x_E}{W} + 2k_2 x_C x_B \tag{3.24e}$$

$$\frac{dx_G}{dt} = -\frac{(F_A + F_B)x_G}{W} + 1.5 k_3 x_P x_C \tag{3.24f}$$

Here, x_i represents the mass fraction of component i. The model parameters for this case study are summarized in Table 3.3. The economic optimization problem

to be considered is:

$$\begin{aligned} \min_{u} J &= p_A F_A + p_B F_B - (F_A + F_B)\left[p_P(1+\Delta p_P)x_P + p_E x_E\right] \\ \text{s.t.} \quad & x_E \leq 0.30 \\ & x_A \leq 0.12 \end{aligned} \tag{3.25}$$

Parameter	Value
W	2105 kg
$k_{0,1}$	1.6599×10^{-6} kg/s
$k_{0,2}$	7.2117×10^{-8} kg/s
$k_{0,3}$	2.6745×10^{-12} kg/s
E_1	6666.7 K
E_2	8333.3 K
E_3	11111 K
p_A	79.23 \$/kg
p_B	118.34 \$/kg
p_P	1043.38 \$/kg
p_E	20.92 \$/kg

Table 3.3: Model parameters for case study 2

The available DOFs for operation are $u = \begin{bmatrix} F_B & T_R \end{bmatrix}^T$, namely the mass inflow of pure B and the reactor temperature, and the considered disturbances are $d = \begin{bmatrix} F_A & \Delta p_P \end{bmatrix}^T$, namely the mass inflow of pure A and the relative variation of the price p_P. Similar to case study 1, we can visualize the active constraint regions as a function of the two disturbances, as shown in Figure 3.7.

We choose to scale the constraints relative to the maximum optimal constraint value in the disturbance window shown in Figure 3.7. This gives the following scaled problem:

$$\begin{aligned} \min_{u} J &= p_A F_A + p_B F_B - (F_A + F_B)\left[p_P(1+\Delta p_P)x_P + p_E x_E\right] \\ \text{s.t.} \quad & g_1 = \frac{x_E - 0.30}{0.0287329} \leq 0 \\ & g_2 = \frac{x_A - 0.12}{0.0714527} \leq 0 \end{aligned} \tag{3.26}$$

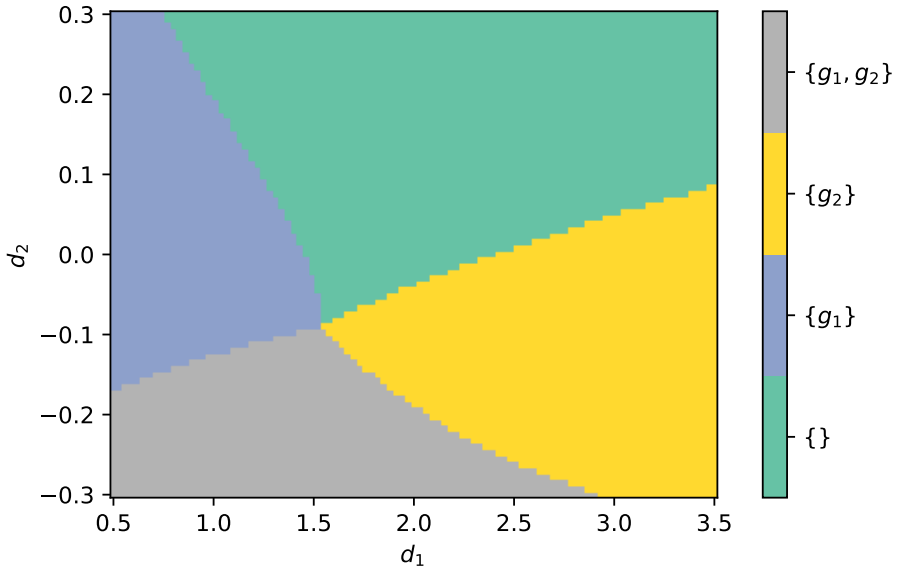


Figure 3.7: Active constraint regions for case study 2 as a function of disturbances

The measurements are the two constraints, the fraction of component P and the price of P, that is, $y = \begin{bmatrix} g_1 & g_2 & x_p & \Delta p_P \end{bmatrix}^T$. To design the region-based MPC, we must first obtain the matrices J_{uu}, J_{ud}, G^y, G^y_d, and G_u, which depend on the operating point. One simple approach is to calculate those matrices at the design point and keep them constant during operation. We shall consider this strategy using two design points, in order to evaluate the effect of nonlinearity on the proposed framework.

For the MPC controllers, we use a linear approximation of the dynamic model at the design point, and estimate disturbances and additional integrating states using a linear Kalman filter that ensures zero offset (Pannocchia et al. 2015). The use of a mismatch between the internal linear MPC model and the true nonlinear plant model is to show that the steady-state economic performance is a result of a correct choice of CVs, and not necessarily of a correct dynamic process model.

The first set of simulations refers to a linearization at the vertex between the regions ($d^\star = [1.54265 - 0.0891]$), where the resulting H_0 is given by:

$$H_0 = \begin{bmatrix} -42.0785 & -36.0878 & 1153.68 & -126.066 \\ 3.42257 & -0.370313 & -232.921 & 0.369661 \end{bmatrix}$$

and the corresponding gradient projections and controller tunings are given in

Tables 3.4 and 3.5, respectively.

$\mathcal{A}$	$N_{\mathcal{A}}^T$
$\{\}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\{1\}$	$\begin{bmatrix} 0.1208 & 0.9927 \end{bmatrix}$
$\{2\}$	$\begin{bmatrix} -0.0849 & 0.9964 \end{bmatrix}$
$\{1,\ 2\}$	-

Table 3.4: Optimal gradient projections for example 2 - linearization at vertex

Parameter	$\mathcal{A}$	Value
$Q_{\mathcal{A}}$	$\{\}$	diag($[0.01, 1.0]$)
	$\{1\}$	diag($[30.0, 1.0]$)
	$\{2\}$	diag($[30.0, 1.0]$)
	$\{1,\ 2\}$	diag($[3.0, 30.0]$)
$R_{\mathcal{A}}$		diag($[0.5, 0.02]$)
N		60
Δt		0.0333 h
Q^e		diag($[10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}, 8, 8, 0.8, 0.8]$)
R^e		diag($[10^{-12}, 10^{-12}, 10^{-12}, 10^{-12}]$)

Table 3.5: Tuning of controllers and estimator for example 2 - linearization at vertex

The results are shown in Figures 3.8 to 3.10, From the dynamic simulation in Figure 3.8, we see that the behavior of standard MPC and region-based MPC is identical in the unconstrained region (until $t = 4$ h), but the operation at the subsequent region with g_1 active (from $t = 4$ to $t = 8$ h) highlights the difference between the approaches. The region-based MPC framework detects quite accurately the region change and switches the CVs accordingly, whereas standard MPC attempts to track the CVs from the unconstrained region, which is not always optimal. There is some loss with region-based MPC associated with the nonlinearity in the model. For the fully constrained region ($\{g_1, g_2\}$ from $t = 8$ to $t = 12$ h), the two MPC schemes behave similarly, attaining zero steady-state loss by taking all the constraints to their limit values. The region-based MPC attains this through direct constraint control, whereas standard MPC relies on its dynamic constraint handling, which has its own issues regarding stability and performance.

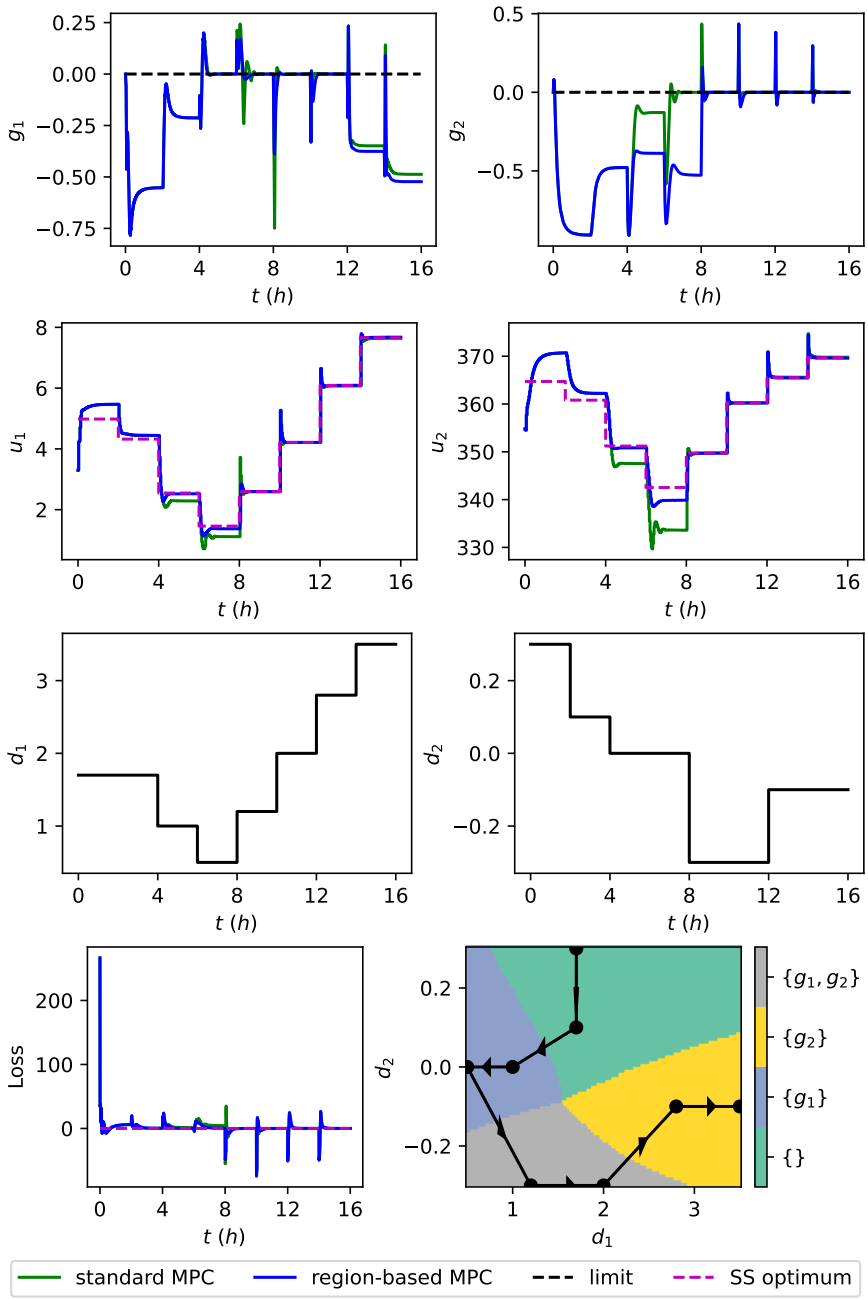


Figure 3.8: Dynamic simulation results for case study 2 - comparison between standard MPC (green) and the proposed region-based MPC (blue) - linearized at vertex

The steady-state behavior for both the region-based MPC and the standard MPC was simulated for the whole domain displayed in Figure 3.7, and the results are presented in Figures 3.9 and 3.10, respectively. Due to the linearization being performed at the vertex between the four regions, the description of the boundaries between the regions is fairly accurate for the region-based MPC in Figure 3.9, which illustrates the local exactness of the method. However, we can see a large economic loss at the unconstrained region as the system moves further from the reference point, and this can be attributed to the errors associated with Equation (3.12) for nonlinear systems. The standard MPC in Figure 3.10 does not reproduce the optimal behavior locally in terms of region boundaries, and it creates an economic loss peak at the region with g_1 active, which is not seen in the same magnitude for the region-based MPC. It can also be seen that there are some disturbance combinations with a smaller economic loss for standard MPC than for region-based MPC, although this is not the general trend. This curious behavior is a combination of the inaccuracy of the CVs calculated locally with the giving up of those CVs by the MPC algorithm, which in itself depends on the tuning parameters of the MPC. This fact is illustrated in Figure 3.11, where a different standard MPC tuning than that of Table 3.5 led to much worse overall performance in the constrained regions. For the proposed region-based MPC this is not an issue because the choice of CVs is consistent with the active constraints, and therefore the control offset will be zero at steady state, making the steady-state performance independent of the dynamic tuning.

We now design the region-based MPC and the standard MPC to operate around $d^\star = [2.0, +0.2]$, which lies in the interior of the unconstrained region. Here, the resulting H_0 is:

$$H_0 = \begin{bmatrix} -46.0296 & -32.6404 & 1577.81 & -96.6946 \\ 5.50426 & -2.61125 & -393.808 & 0.342395 \end{bmatrix}$$

and the corresponding gradient projections and controller tunings are given in Tables 3.6 and 3.7, respectively.

For this linearization, the results are shown in Figures 3.12 to 3.14. In this case, the region-based MPC overall gives a much smaller economic loss on the partly constrained regions compared to standard MPC. Also, the regions obtained in Figure 3.13 are shaped similarly to the optimal regions, which does not happen with the standard MPC in Figure 3.14. Because the linearization of the system happened in the interior of the unconstrained region, the economic loss on that region is smaller when compared to that of Figure 3.9, while not resulting in a larger loss for the remaining regions in the case of the region-based MPC.

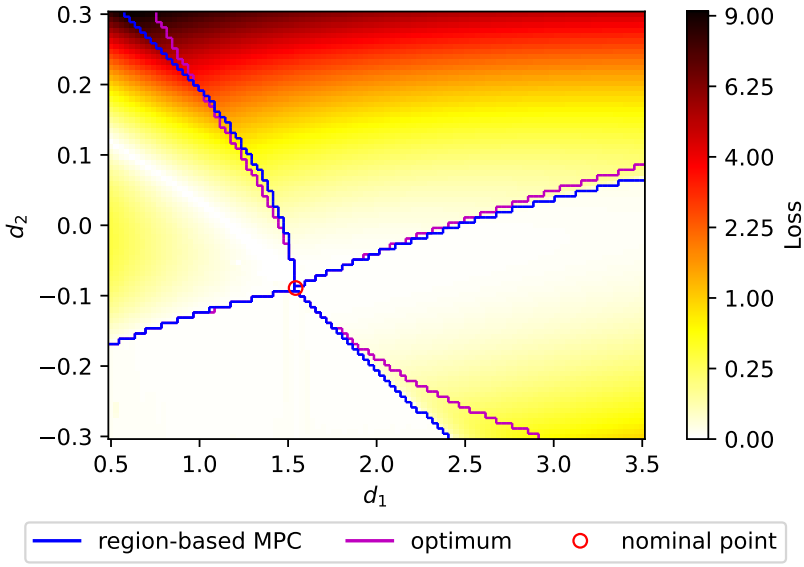


Figure 3.9: Steady-state economic loss for region-based MPC on case study 2 - linearized at vertex

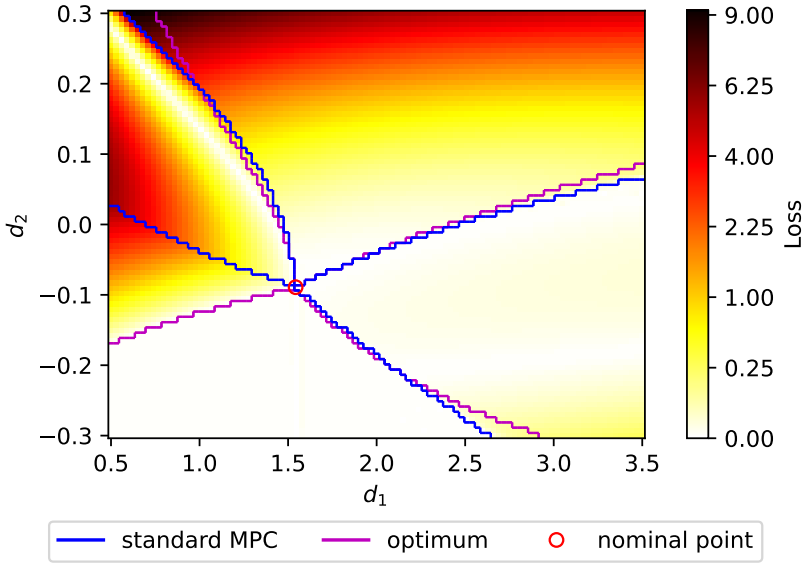


Figure 3.10: Steady-state economic loss for standard MPC on case study 2 - linearized at vertex

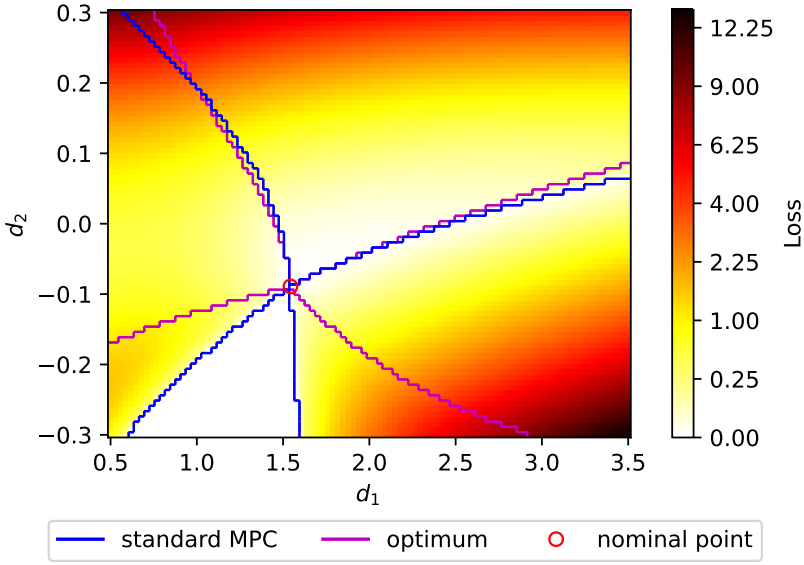


Figure 3.11: Steady-state economic loss for standard MPC on case study 2 - linearized at vertex, $Q = \text{diag}([5 \times 10^{-4},\ 5])$

$\mathcal{A}$	$N_{\mathcal{A}}^T$
$\{\}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\{1\}$	$\begin{bmatrix} 0.1110 & 0.9938 \end{bmatrix}$
$\{2\}$	$\begin{bmatrix} -0.1685 & 0.9857 \end{bmatrix}$
$\{1,\ 2\}$	-

Table 3.6: Optimal gradient projections for example 2 - linearization at $d^\star = [2.0, +0.2]$

Parameter	$\mathcal{A}$	Value
$Q_{\mathcal{A}}$	$\{\}$	$\text{diag}([5 \times 10^{-4},\ 5])$
	$\{1\}$	$\text{diag}([30.0, 1.0])$
	$\{2\}$	$\text{diag}([30.0, 1.0])$
	$\{1,\ 2\}$	$\text{diag}([3.0, 30.0])$

Table 3.7: Tuning of controllers and estimator for example 2 - linearization at $d^\star = [2.0,\ +0.2]$ (omitted parameters are the same as in Table 3.5)

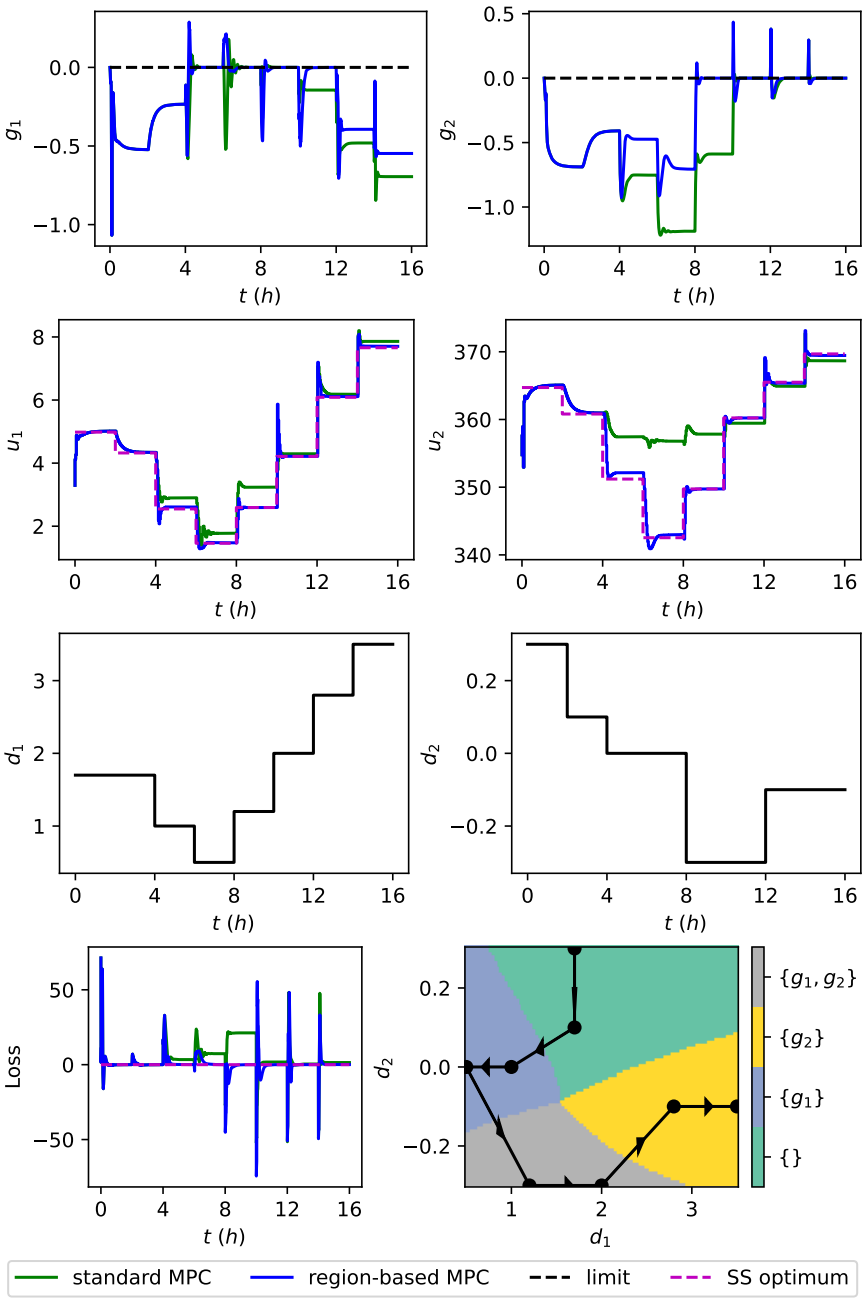


Figure 3.12: Dynamic simulation results for case study 2 - comparison between standard MPC (green) and the proposed region-based MPC (blue) - linearized at $d^\star = [2.0, +0.2]$

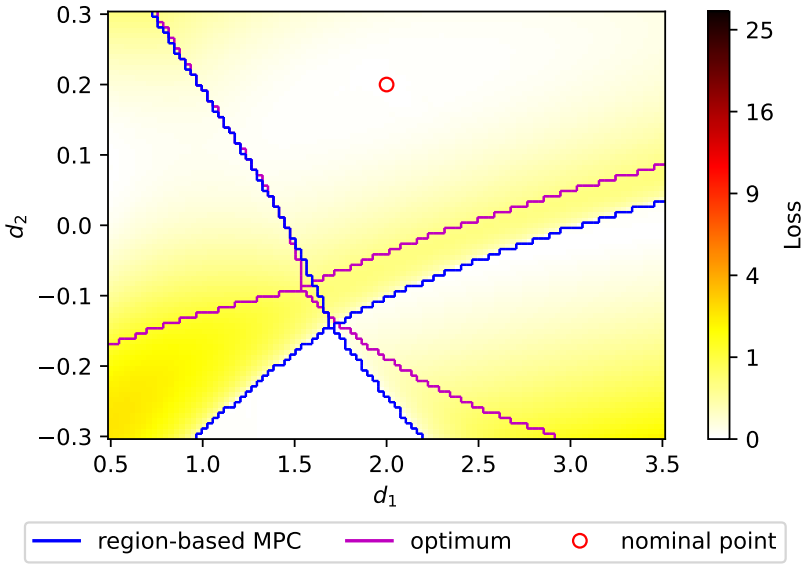


Figure 3.13: Steady-state economic loss for region-based MPC on case study 2 - linearized at $d^\star = [2.0, +0.2]$

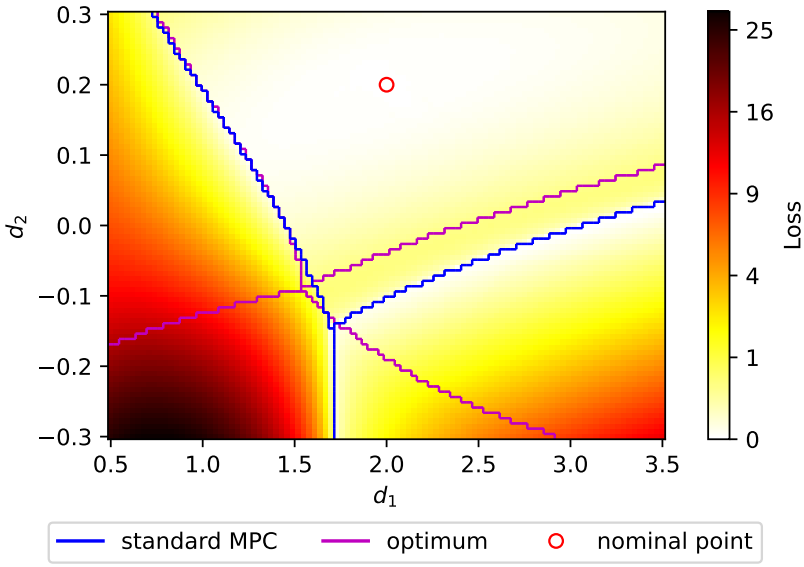


Figure 3.14: Steady-state economic loss for standard MPC on case study 2 - linearized at $d^\star = [2.0, +0.2]$

3.4.3 Case study 3 - Williams-Otto reactor revisited

We now revisit the previous case study, to consider a case with three rather than two constraints and only one disturbance. This is a case where standard decentralized selector-based region-based control would not work because $n_g > n_u$ Bernardino and Skogestad (2024c). We consider the following optimization problem:

$$\begin{aligned} \min_{u} \quad & J = p_A F_A + p_B F_B - (F_A + F_B)(p_P x_P + p_E x_E) \\ \text{s.t.} \quad & F_B \leq 4.0 \\ & T_r \leq 355.0 \\ & x_G \leq 0.105 \end{aligned} \tag{3.27}$$

Here, the operational constraints are related to maximum allowed values for F_B, T_r, and x_G. The MVs are the same, $u = \begin{bmatrix} F_B & T_r \end{bmatrix}^T$, but we only consider one disturbance, $d = F_A$, which is in the range $0.5 \leq d \leq 3.5$. We again normalize the constraints. The normalized problem is given by:

$$\begin{aligned} \min_{u} \quad & J = p_A F_A + p_B F_B - (F_A + F_B)(p_P x_P + p_E x_E) \\ \text{s.t.} \quad & g_1 = \frac{F_B - 4.0}{2.68018} \leq 0 \\ & g_2 = \frac{T_r - 355.0}{9.55095} \leq 0 \\ & g_3 = \frac{x_G - 0.105}{0.00411912} \leq 0 \end{aligned} \tag{3.28}$$

Figure 3.15 present the active constraint regions as a function of the disturbance F_A. It can be seen that all possible feasible combinations of active constraints appear in the considered disturbance range.

This problem has three constraints and two MVs, and therefore a fixed pairing between constraints and MVs would not account for all possible active constraint regions. For instance, if u_1 is paired to g_1, u_2 is paired to g_2, and g_3 may be active at the same time as the other constraints, control of g_3 must have some sort of adaptive pairing, if decentralized control is to be achieved (see Bernardino et al. (2022b) for an example).

In Figure 3.16, we present results for the dynamic simulation of the system. It should be noted that the tuning used for the standard MPC is done such that it can operate acceptably even when constraints become active, which hinders the overall attainable performance. Because the region-based MPC can be tuned inde-

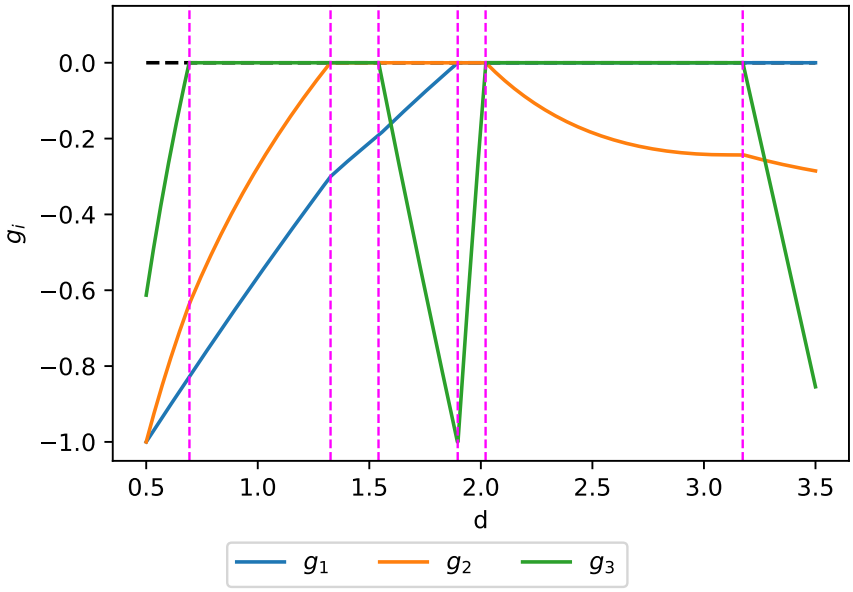


Figure 3.15: Active constraint regions for case study 3 in terms of optimal constraint values as a function of d (magenta dashed lines represent region switches)

pendently for every active constraint region, dynamic performance can be expected to be better.

In Figures 3.17 to 3.19 we compare the steady-state behavior of the region-based MPC and the standard MPC, respectively, in terms of the constraints' values and economic losses. We can see that the linearization strategy is such that the operation is exactly optimal at $d = d^\star = 1.0$ for the region-based MPC, but the same cannot be said for the standard MPC. This is because the system was linearized at a partly constrained region, and while the region-based MPC is able to use the correction $J_u^\star$ in Equation (3.13), the same correction applied to the standard MPC does not lead to optimal operation. In addition, standard MPC performs poorly at driving the system to the correct constraints to be controlled, which leads to huge discrepancies with relation to optimality for $d > 1.9$ and large economic losses.

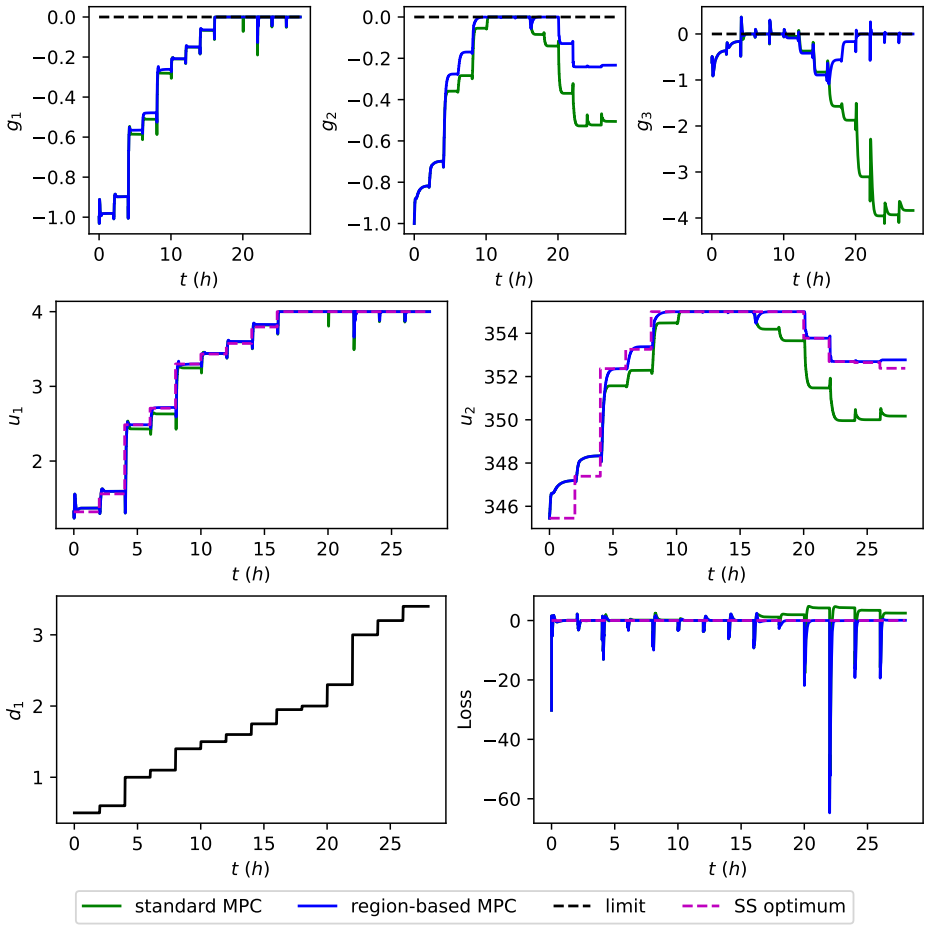


Figure 3.16: Dynamic simulation results for case study 3 - comparison between standard MPC (green) and the proposed region-based MPC (blue)

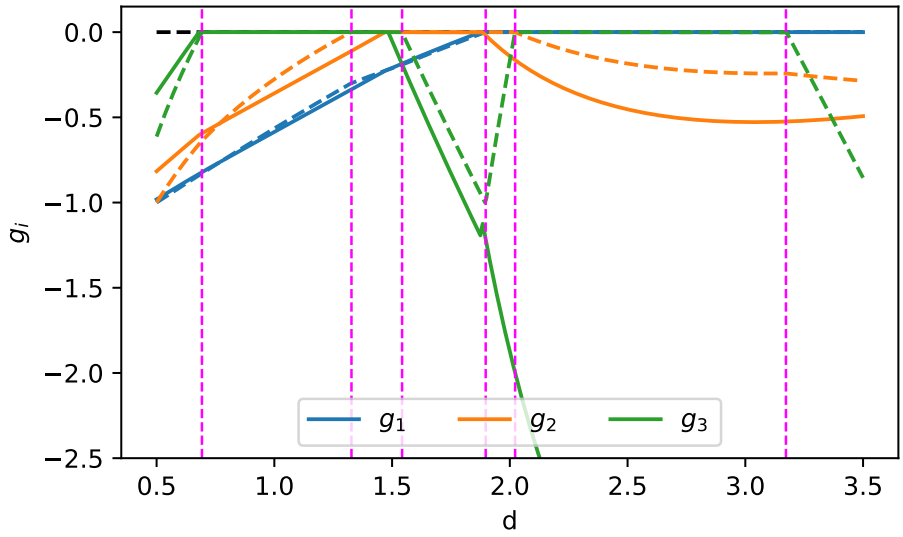


Figure 3.17: Steady-state constraint values for standard MPC on case study 3 (optimal values as dashed lines)

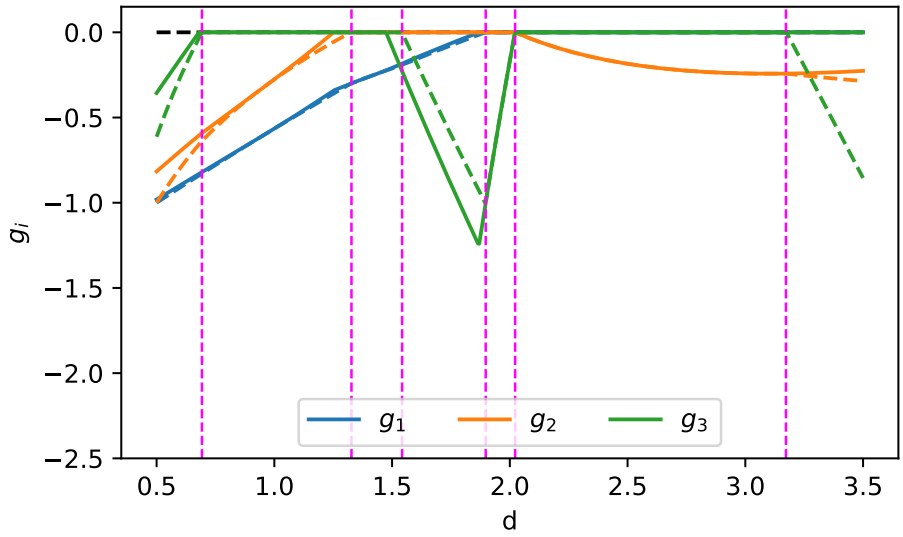


Figure 3.18: Steady-state constraint values for region-based MPC on case study 3 (optimal values as dashed lines)

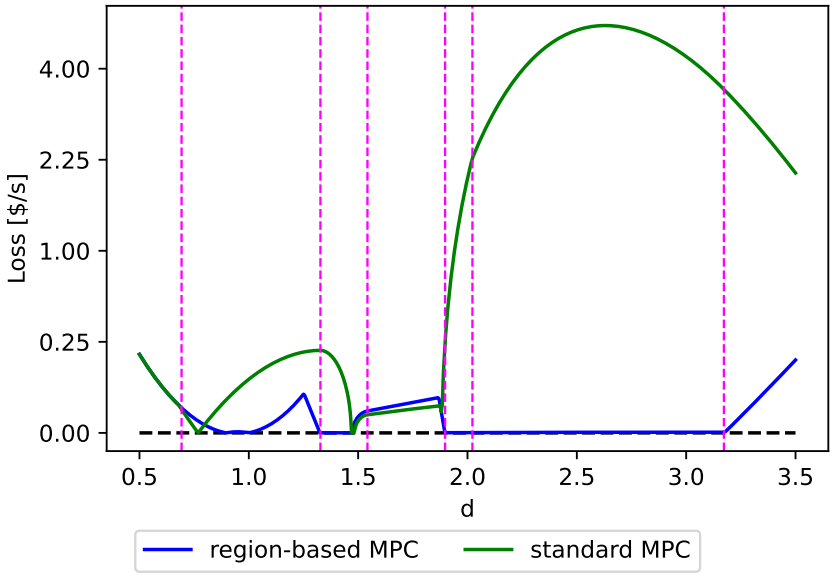


Figure 3.19: Closed-loop steady-state economic loss for controllers on case study 3 (y-axis is scaled quadratically for better visualization)

3.5 Discussion

3.5.1 Exact local method for gradient estimation

The gradient estimation in this work, first shown in Jäschke and Skogestad (2011), is based on the nullspace method of self-optimizing control. This estimated gradient is used to determine the unconstrained CVs and to detect switches in active constraints, and we use the nullspace method due to the simplicity of the resulting gradient estimate. This method, however, disregards measurement error, and is limited in applicability for cases with few measurements. In another paper (Bernardino and Skogestad 2024a), we propose a gradient estimation method that accounts for static measurement error (n^y), based on the exact local method, which results in an optimal linear combination of any number (n_y) of measurements, preserving the simplicity of the method.

3.5.2 Optimal operation under changing active constraints

The region-based MPC proposed in this work depends on a logic element that detects the current active constraint set so that the corresponding self-optimizing CVs, defined by $CV_{\mathcal{A}}$, are selected and controlled. The use of simple logic elements for changing control structures is very common among practitioners, but it is not generally clear how to use these elements optimally (Skogestad 2023). This issue has received attention in recent developments, especially when a low number of switching variables is involved (Reyes-Lúa and Skogestad 2020, Krishnamoorthy and Skogestad 2019). In general, it is necessary that one has analyzed the range of disturbances to be handled by the control structure, in order to propose a switching strategy, along with a pairing between MVs and CVs, that accommodates all control objectives. That procedure is, however, dependent on the case study and the engineering insight, and one may find cases where a decentralized strategy would be impossible or too complex to be considered in practice (Bernardino et al. 2022b).

In addition to this issue, even if the disturbance range is such that constraints paired to the same MV are never active at the same time, the whole control structure should in principle be changed according to which set of constraints is active because the optimal CVs related to the unconstrained degrees of freedom will change. Therefore, in terms of self-optimizing control of such systems, we can say that the general case of a switching logic between CVs must be in some sense centralized, as the complexity of the decision process becomes combinatorial. Because of these intrinsic limitations of decentralized SOC structures, centralized approaches for SOC become vital for guaranteeing optimal operation of systems with several changing constraints.

3.5.3 Estimation of active constraints

To determine the active set during operation, we use the method by Woodward et al. (2010), which is proven optimal for measured gradients. In this work, the cost gradient is estimated through a linear combination of the measurements, which is consistent with the CVs being used.

Another approach for detecting changes in the active constraint region is to track the values of the CVs in the neighboring regions. (Jäschke and Skogestad 2012, Manum and Skogestad 2012). The CVs determined for each region must be consistent to result in a unique solution to the switching problem. If this is not the case, one may encounter multiple steady-state solutions or lack of convergence where the control structures switch indefinitely. This was observed when applying this approach to the case studies. On the other hand, the solution presented in this work relies on a single model realization, and all CVs obtained from it are consistent.

3.5.4 Use of direct measurements

To simplify the active constraint detection block, one may consider using a gradient estimate based directly on the current measurements, instead of the expected steady-state measurements described in Algorithm 1. The use of direct measurements is illustrated in Figure 3.20, compared to the approach used in this work, with all other parameters being the same across the simulations.

We can see that the use of direct measurements on the active set detection algorithm gives a worse overall closed-loop behavior in this case. For the first step change on the disturbance ($t = 1.5$ h), even though g_3 is dynamically violated, the cost gradient estimated directly from the measurements dictates that it will be given up at steady state, and the constraint is therefore not deemed active. When the system is close to steady state, the cost gradient estimate becomes more accurate, and the constraint is then considered active. A similar behavior happens for the second step change on the disturbance ($t = 3$ h). For the last disturbance value (from $t = 6$ h), the closed-loop system faces problems, as the estimated active set does not match the actual system operation (g_3 is active despite the wrong estimation), and the state estimator is in conflict with the dynamic constraint. Because of this, we see oscillation in the result, and the system does not seem to converge.

3.5.5 Region-based MPC tuning

The region-based MPC can be seen as a set of multivariable feedback controllers coordinated by a logic element. This logic element introduces an additional information loop, besides the feedback controller itself, which may cause stability problems. Rapid changes in which of the controllers is active may occur from the

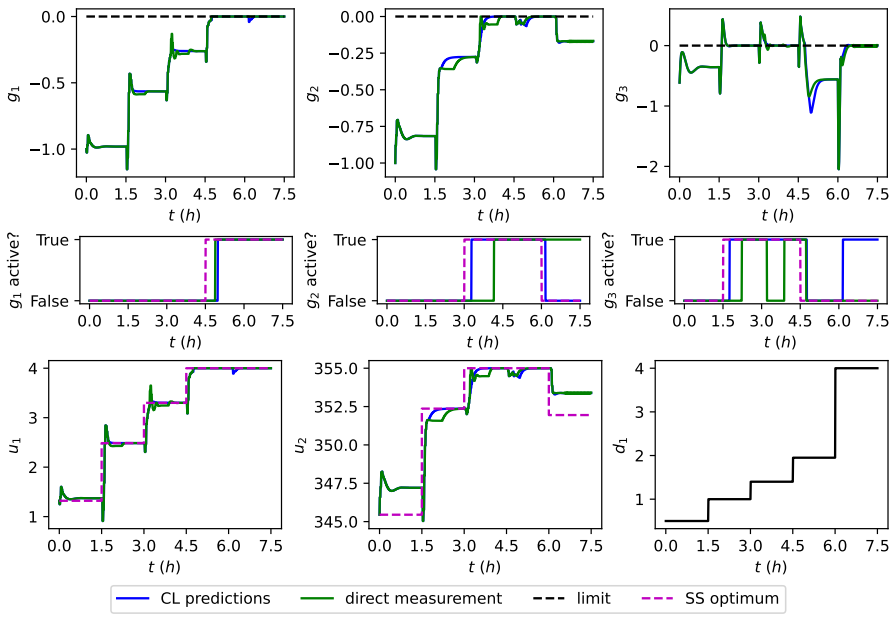


Figure 3.20: Comparison between region-based MPC with active set detection from closed-loop predictions (blue) and from direct measurements (green)

interaction between the switching element and the closed-loop dynamics, generating high-frequency, self-sustained switching. This is a known issue in closed-loop systems with selectors or other logical elements, and it may be counteracted by restricting how fast the logical element may change, leading to overall system stability (Lin and Antsaklis 2009). In this work, this is attained by the tuning parameter N_{sw}.

Additionally, the cost functions (3.8) for the region-based MPC must be independently tuned. This is necessary because different CVs usually have different dynamic behaviors. Careful evaluation of MPC tunings for different regions is therefore advised, so that good dynamic performance is attained in all relevant operating conditions.

3.5.6 Comparison between region-based MPC and other MPC approaches

The simulations verify that standard setpoint-tracking MPC is unable to deal with changing steady-state constraints. To satisfy a steady-state constraint which is not in the nominal region, the standard MPC gives up on tight control of its CVs, usually all at the same time. In our simulations, the prioritization of CVs is auto-

matically done by tuning the MPC control weights, and therefore the steady-state offset of the CVs will be indirectly determined by such tuning. This aspect is not usually prioritized in the design of MPC controllers, and therefore we have no quantitative control over how much offset we tolerate for each CV. Another possibility, which is used in industrial implementations of MPC, is solving a sequence of steady-state calculations, assessing constraint satisfaction, before solving the MPC problem itself (Strand and Sagli 2004). This allows for adapting the MPC problem, changing control specifications so that constraints are considered when necessary, and avoiding the use of dynamic constraints that may cause stability issues. The problem here becomes determining the control specifications, which is often based on process experience.

The present work has not focused on integrating the proposed tool with RTO, as other works have covered (Delou et al. 2021). Instead, the region-based MPC was formulated to be independent of the RTO layer, such that it operates near optimally without its updates. Naturally, the proposed tool can be integrated with RTO, by updating the gain matrices and reference values in Figure 3.2. Because these updates are associated with the steady-state conditions of optimality, the economic performance of the region-based MPC will be as good as the quality of these updates.

Some MPC frameworks (for example, that of Rawlings (2000)) include a target calculation block, which will define to what steady state the MPC will converge. In these frameworks, it seems possible that the approach presented in this work can be used at the target calculation block only, and the MPC problem remains unchanged for every active constraint region. The main benefit of this would be that the stability properties of the MPC problem would remain the same regardless of the detected active set. This does not completely solve the stability issue, as the estimator and the target calculator blocks must converge, but it would still be an appealing approach.

We must also note that the proposal of this work is fundamentally different from that of centralized approaches such as economic model predictive control (EMPC). In these approaches, the dynamic and economic problems are solved together, which requires a high level of detail in the available dynamic model (Ellis et al. 2014). In the proposed region-based MPC, we only require a reasonable dynamic model to ensure closed-loop stability for the tracking of CVs and an accurate economic steady-state problem that will define these CVs.

3.6 Conclusion

A framework for self-optimizing control under changing active constraints was presented, see Figure 3.2. Its main elements are (1) the active set detection block (see Algorithm 1), and (2) the design of self-optimizing CVs in each active constraint region for the unconstrained degrees of freedom, see $c_{\mathcal{A}} = N_{\mathcal{A}}^T H_0 y$ in Equation (3.14) and the respective setpoint $c_{\mathcal{A}}^{sp}$ in Equation (3.15). In this paper, we estimated the cost gradient J_u using the nullspace method from self-optimizing control, by obtaining a measurement combination matrix H_0. More generally, with measurement bias and any number of measurements y, it is recommended to obtain H_0 for estimating the cost gradient J_u using the exact local method (Bernardino and Skogestad 2024a). The setpoints $c_{\mathcal{A}}^{sp}$ are calculated based on the nominal operating point and were not updated during the simulations for the three case studies, to show the self-optimizing nature of the chosen CVs. We highlight that the switching of control objectives is done without the need for pairing MVs and CVs and without the need for RTO updates, making it applicable to a wide class of problems.

Funding

This work was funded by the Research Council of Norway through the IKTPLUSS programme (project number 299585).

Chapter 4

Optimal measurement-based cost gradient estimate for real-time optimization

This chapter has been submitted as a full paper:

> L. F. Bernardino and S. Skogestad. Optimal measurement-based cost gradient estimate for real-time optimization. *Submitted to Computers & Chemical Engineering*, 2024a

4.1 Introduction

When the aim is to implement a control strategy to achieve optimal steady-state operation, one usually thinks of adding a real-time optimization (RTO) layer which adjusts the setpoints to the control layer. However, conventional RTO requires online optimization of a detailed nonlinear model which is usually expensive to obtain and maintain. In addition, the success of RTO relies on estimating the disturbances, which is usually slow (Krishnamoorthy and Skogestad 2022). This limitation can be circumvented by proper selection of the controlled variables (CVs) so that the RTO layer may be eliminated or at least less frequent RTO updates are needed. This is the idea of self-optimizing control (Skogestad 2000). According to the first-order optimality conditions, the ideal self-optimizing CV would be the gradient of the economic cost function (reduced gradient for the constrained case), which when driven to zero achieves optimal operation without the need for an RTO layer. In this paper, the cost gradient is denoted J_u, but it is also sometimes

denoted $\nabla_u J$ in other works. The goal of this paper is then to derive a simple measurement-based estimate the gradient J_u.

The most common approach is to combine the available measurements with a plant model and from this derive an estimate of the states (including disturbances) and subsequently the gradient. One approach is described in Krishnamoorthy et al. (2018), where a Kalman filter is used for dynamic state estimation, and the model is then linearized around the estimated operating point to give an estimate for J_u.

An alternative model-free approach is to directly estimate the cost gradient J_u from plant data by input excitation, which is done in extremum-seeking control strategies (Tan et al. 2010). However, this assumes that a cost measurement J is available and that the plant dynamics are fast, neither of which are usually satisfied in process control applications.

Both these existing model- and data-based estimation approaches have in common that they are rather complex and that the gradient estimation and its use for control are divided into separate tasks. However, this separation between estimation and control is not generally optimal. In other words, since it is not clearly defined what the gradient J_u will be used for in the estimation step, we cannot expect that the resulting estimate will be optimal in terms of minimizing the cost J.

A third approach for estimating the gradient, the focus of this paper, is to make use of self-optimizing control methods. These methods aim to design controlled variables (CVs) that directly minimize the cost, that is, there is no intermediate step to estimate disturbances or gradients. A further advantage of this approach is simplicity. The resulting CVs are static linear combinations of the available measurements, which greatly simplifies implementation. It has been known that self-optimizing CVs are linked to the cost gradient for the simple case with a sufficient number of noise-free measurements Jäschke and Skogestad (2011). The main contribution of this work is to extend this link and derive a simple static linear measurement-based expression for the gradient J_u for any number of noisy measurements. The basis is the exact local method Alstad et al. (2009) of self-optimizing control. We also show in this paper how this gradient estimate is useful when dealing with constraints, both to set the unconstrained degrees of freedom and to identify constraint switching.

The paper is organized as follows. Section 4.2 presents the mathematical problem considered in this work. Section 4.3 describes how this problem is related to self-optimizing control. In Section 4.4 we present the main result of this work based on the analysis of the unconstrained problem, which is complemented by the analysis of the constrained problem in Section 4.5. An example of the application of these

results to a decentralized control framework is shown in Section 4.6 with a numerical example, showing its use with changing active constraints. Some remarks about the presented results are made in Section 4.7, and the paper is concluded in Section 4.8.

4.2 Problem statement

The steady-state optimization problem considered in this work is of the form:

$$\begin{aligned} \min_{u} \quad & J(u,d) \\ \text{s.t.} \quad & g(u,d) \leq 0 \end{aligned} \tag{4.1}$$

Here, J: $\mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}$ denotes the objective function, g: $\mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_g}$ the inequality constraints, $u \in \mathbb{R}^{n_u}$ the decision variables (inputs; manipulated variables for steady-state control), and $d \in \mathbb{R}^{n_d}$ the disturbance variables (including model parameters) which are assumed varying and generally unknown in this paper. The available online information about the system is assumed to be the measured variables $y \in \mathbb{R}^{n_y}$ (which usually include u and may include measured disturbances). Any internal states have been formally eliminated from the mathematical formulation in (4.1).

The optimal input, which is the solution to the problem in Equation (4.1), is in the paper denoted $u^{opt}(d)$. It satisfies the following first-order KKT conditions:

$$J_u(u^{opt},d) + g_u(u^{opt},d)^T \lambda^{opt} = 0 \tag{4.2a}$$

$$g(u^{opt},d) \leq 0 \tag{4.2b}$$

$$\lambda^{opt} \geq 0 \tag{4.2c}$$

$$g(u^{opt},d)^T \lambda^{opt} = 0 \tag{4.2d}$$

Here, $J_u(u,d) \in \mathbb{R}^{n_u}$ denotes the gradient of J with respect to u, $g_u(u,d) \in \mathbb{R}^{n_g \times n_u}$ denotes the gradient of g with respect to u, and $\lambda^{opt} \in \mathbb{R}^{n_g}$ denotes the Lagrange multipliers at the optimum. Note that it is the *unconstrained* cost gradient J_u that enters into the first-order optimality conditions.

The cost $J(u,d)$ and the constraints $g(u,d)$ in Equation (4.1) can be approximated locally by the following Taylor expansions centered at the nominal point (u^*, d^*):

$$\begin{aligned} J(u,d) = J^* + \begin{bmatrix} J_u^{*T} & J_d^{*T} \end{bmatrix} \begin{bmatrix} (u-u^*) \\ (d-d^*) \end{bmatrix} \\ + \frac{1}{2} \begin{bmatrix} (u-u^*)^T & (d-d^*)^T \end{bmatrix} \underbrace{\begin{bmatrix} J_{uu} & J_{ud} \\ J_{ud}^T & J_{dd} \end{bmatrix}}_{\mathcal{H}} \begin{bmatrix} (u-u^*) \\ (d-d^*) \end{bmatrix} \end{aligned} \tag{4.3}$$

$$g(u,d) = g^* + \begin{bmatrix} g_u^* & g_d^* \end{bmatrix} \begin{bmatrix} (u-u^*) \\ (d-d^*) \end{bmatrix} \tag{4.4}$$

where $(u-u^*)$ and $(d-d^*)$ denote, respectively, the inputs and disturbances as their deviation from the nominal point.

The cost expression in Equation (4.3) is exact for quadratic problems where the Hessian $\mathcal{H}$ (including J_{uu}) is independent of the operating point. In general, there will be an approximation error if the actual operation moves away from the nominal point. Strictly speaking, the elements in the Hessian matrix $\mathcal{H}$ should have a superscript $*$ (e.g. J_{uu}^*), but this is omitted to simplify notation, and also because it assumed that they remain approximately constant.

The objective of this paper is to find from the available measurements y (which are subject to noise n^y) an optimal estimate of the gradient J_u (which will vary as a function of u and d) for use in real-time optimization. The expected magnitudes of the disturbances and measurement errors are quantified by diagonal weight matrices W_d and W_{n^y}. That is, we assume that:

$$\begin{aligned} (d-d^*) &= W_d d' \\ n^y &= W_{n^y} n^{y\prime} \end{aligned} \tag{4.5}$$

where the combined generating set of possible d' and $n^{y\prime}$ is unit two-norm bounded, i.e.:

$$\left\| \begin{bmatrix} d' \\ n^{y\prime} \end{bmatrix} \right\|_2 \leq 1 \tag{4.6}$$

Note that we are considering steady-state operation, so n^y represents the static measurement error, that is, the measurement bias. Often, n^y is called measurement noise, but this may be a bit misleading because the average (steady-state) value is not zero, as is usually assumed in stochastic optimal control. For example, $n^y = 0.15$ means that if the actual value is $y = 2.7$, then the measured value is $y_m = y + n^y = 2.85$. Finally, note that the objective of this paper is not to find the "optimal" gradient J_u in itself, but the optimal estimate $\hat{J}_u$ to be used in the first-order optimality condition (4.2a) to solve the problem in (4.1).

4.3 Optimal operation for the unconstrained case: Self-optimizing control

In the following consider the case with no constraints g and assume that the nominal operating point is optimal, that is,

$$u^* = u^{opt}(d^*)$$

It then follows from the first-order KKT condition (4.2a) that:

$$J_u^* = 0$$

This assumption is made to simplify the expressions for the loss, and the controlled variables derived here do not depend on this assumption (see chapter 6 in Alstad (2005)).

Following Halvorsen et al. (2003), we can derive from Equation (4.3) the economic loss encountered by applying an input u, compared to using the optimal input $u^{opt}(d)$:

$$L = J(u,d) - J^{opt}(d) = \frac{1}{2}(u-u^{opt})^T J_{uu}(u-u^{opt}) = \frac{1}{2}||z||_2^2 \tag{4.7}$$

where $J^{opt}(d) = J(u^{opt}(d),d)$ is the optimal cost for a given d and the loss variable z is defined as:

$$z \triangleq J_{uu}^{1/2}(u-u^{opt}) \tag{4.8}$$

The idea of self-optimizing control is to achieve optimal operation using feedback control. In this paper, the controlled variables (CVs) c are assumed to be linear combinations of the measured variables, $c = Hy$, and we use a linear steady-state measurement model:

$$y = G^y u + G_d^y d \tag{4.9}$$

Note that the actual measured value is $y_m = y + n^y$. The setpoints c_s are assumed to be constant; see Figure 4.1. To be nominally optimal (with no disturbances or measurement noise), we must choose $c_s = c^* = Hy^*$ where $y^* = y^{opt}(d^*)$. The controller K has integral action, which means that at steady state the control error

$$(c_m - c^*) = H(y_m - y^*)$$

is controlled to a constant value of zero. The controlled variables c should use up all the available degrees of freedom, and therefore $n_c = n_u$. In this paper, H is allowed to be a full matrix, that is, there are no structural limitations on H.

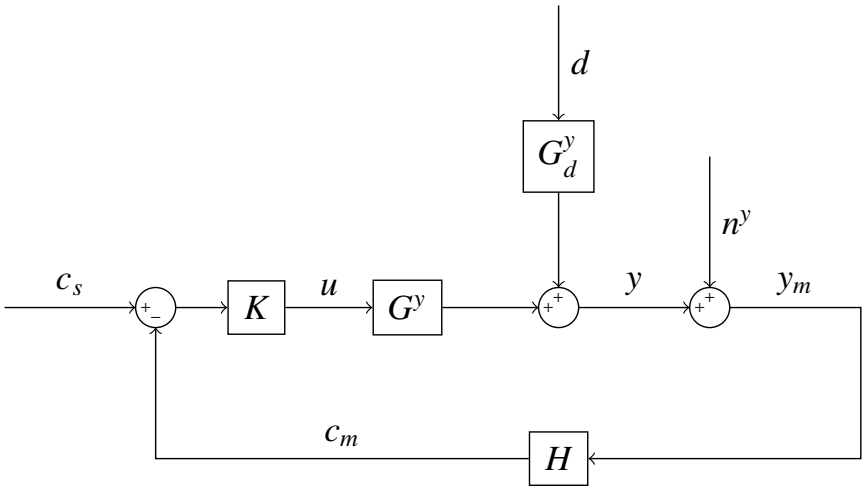


Figure 4.1: Block diagram of closed-loop system. When H is selected as proposed in this paper, the input to the controller K is the negative cost gradient, that is, $c_s - Hy_m = -\hat{J}_u$ see eq. (4.21). This achieves optimal steady-state operation if in addition any active constraints are controlled.

For the expected disturbances and noise in Equation (4.6), Alstad et al. (2009) derived the following analytical expression for the optimal H, known as the "exact local method", which minimizes both the worst-case and average loss L in Equation (4.7):

$$H = M_n^{-1} J_{uu}^{1/2} \left[G^{yT} \left(\tilde{F}\tilde{F}^T \right)^{-1} G^y \right]^{-1} G^{yT} \left(\tilde{F}\tilde{F}^T \right)^{-1} \tag{4.10}$$

where

$$\tilde{F} = [FW_d \quad W_{n^y}]$$

$$F = \frac{dy^{opt}}{dd} = G_d^y - G^y J_{uu}^{-1} J_{ud} \tag{4.11}$$

The solution for H is not unique as the matrix $M_n = J_{uu}^{1/2}(HG^y)^{-1}$ can be freely chosen. The non-uniqueness comes because if $c - c^* = 0$ then so is $D(c - c^*) = 0$ for any non-singular D. In the solution derived in Alstad et al. (2009), the choice is $M_n = I$. The simplest expression for the optimal H results if we select M_n such that $H = G^{yT} \left(\tilde{F}\tilde{F}^T \right)^{-1}$ (Yelchuru and Skogestad 2012).

However, in the next section, we want to find an estimate for J_u (also when $J_u \neq 0$), and in this case directions matter. For this reason, we will choose:

$$M_n = J_{uu}^{-1/2} \tag{4.12}$$

and we show below that the optimal estimate for the gradient J_u is then equal to $H^J(y - y^*)$, where according to the exact local method:

$$H^J = J_{uu}\left[G^{yT}\left(\tilde{F}\tilde{F}^T\right)^{-1}G^y\right]^{-1}G^{yT}\left(\tilde{F}\tilde{F}^T\right)^{-1} \tag{4.13}$$

With different assumptions, other expressions for H may be derived. For the case with a sufficient number of independent measurements ($n_y \geq n_u + n_d$) it is possible to achieve zero disturbance loss for the case with no measurement noise by choosing H such that $HF = 0$ (nullspace method). For the case $n_y = n_u + n_d$, we have the following explicit expression for the nullspace method:

$$H = M_n^{-1}\tilde{J}(\tilde{G}^y)^{-1} \tag{4.14}$$

where $\tilde{G}^y = [G^y \quad G_d^y]$ and $\tilde{J} = J_{uu}^{1/2}[I \quad J_{uu}^{-1}J_{ud}]$. The generalization to use all measurements ($n_y \geq n_u + n_d$) in a way that also minimizes the effect of measurement noise is known as the extended nullspace method Alstad et al. (2009) for which we have:

$$H = M_n^{-1}\tilde{J}(W_{n^y}^{-1}\tilde{G}^y)^{\dagger}W_{n^y}^{-1} \tag{4.15}$$

All these expressions for H can be used for gradient estimation, provided that we choose $M_n = J_{uu}^{-1/2}$, or equivalently $HG^y = J_{uu}$.

4.4 Optimal gradient estimate for the unconstrained case

We will now use the results from self-optimizing control to derive the optimal gradient estimate, where by "optimal" we mean that controlling the gradient estimate to zero achieves optimal steady-state operation, that is, it minimizes the loss L in Equation (4.7) (worst-case or average value) for the expected disturbances and noise as in Equation (4.6).

To do this, we want to express the loss variable z from (4.8) in terms of the gradient J_u. First, note that (Figure 4.1):

$$(c - c^{opt}(d)) = HG^y(u - u^{opt})$$

Second, a first-order Taylor expansion of the gradient around the optimal operating point gives:

$$J_u(u,d) = \underbrace{J_u(u^{opt},d)}_{J_u^{opt}(d)} + J_{uu}(u - u^{opt}(d))$$

Inserting the above two expressions into the definition of the loss variable z in (4.8) gives:

$$z \triangleq J_{uu}^{1/2}(u - u^{opt}) = \underbrace{J_{uu}^{1/2}(HG^y)^{-1}}_{M_n}(c - c^{opt}(d)) = J_{uu}^{-1/2}(J_u - J_u^{opt}(d)) \quad (4.16)$$

For the unconstrained case, we have $J_u^{opt}(d) = 0$, and this is assumed in the following. We then get $z = J_{uu}^{-1/2} J_u$ and to minimize the norm of z, and thereby the loss in (4.7), we conclude that we ideally want $J_u = 0$ at steady state. However, as we will see, it is not possible to achieve $J_u = 0$ in practice because of measurement error.

For the choice $M_n = J_{uu}^{-1/2}$ (which we will use in the following), we derive from (4.16) the following expression for the gradient:

$$J_u = c - c^{opt}(d) = Hy - Hy^{opt}(d)$$

which may be rewritten as:

$$J_u = H(y_m - y^*) - H\underbrace{(y_m - y)}_{n^y} - H(y^{opt}(d) - y^*) \quad (4.17)$$

where we choose $y^* = y^{opt}(d^*)$ because the nominal point is assumed optimal. Note from (4.11) that $(y^{opt}(d) - y^*) = F(d - d^*)$ for the unconstrained case. We then have:

$$J_u = H(y_m - y^*) - Hn^y - HF(d - d^*) \quad (4.18)$$

Note that with a fixed matrix H, the last two terms are unaffected by the input u, that is, unaffected by control.

With no measurement error ($n^y = 0$), the second term in Equation (4.18) is zero. If we use the nullspace method to choose H, then $HF = 0$, and also the third term is zero. The optimal control policy, according to self-optimizing control, is then to adjust u such that the first term is zero, for example, to use feedback control to keep the measurement combinations keep $c_m = Hy_m$ at a constant setpoint $c^* = Hy^*$. This gives $J_u = 0$ and the loss is zero.

More generally, with measurement noise and disturbances, we can use the exact local method to choose the H that minimizes the combined effect of the second and third terms in (4.18). The optimal control policy, similarly to the case without noise, is then to adjust u such that the first term in Equation (4.18) is zero. This minimizes the expected norm of z as in (4.16), and consequently the economic loss L in (4.7). More importantly, and this is the main result of the paper, the

optimal gradient estimate for unconstrained operation, which should be kept at zero at steady state, is simply the first term in (4.18), that is:

$$\hat{J}_u = H(y_m - y^*) \tag{4.19}$$

where y_m is the measurement vector, $y^* = y^{opt}(d^*)$ is the nominal optimal value of the measurement y, and H is given by H^J in Equation (4.13) (exact local method). This follows from self-optimizing control theory, because choosing $H = H^J$ minimizes the effect of the second and third terms in Equation (4.18) (it minimizes both the expected and worst-case loss when d and n^y vary as given in (4.6)).

Interestingly, since the second and third terms in (4.18) are generally nonzero (due to measurement noise and disturbances), it follows that optimal operation (in terms of minimizing the economic loss) does not give $J_u = 0$ at steady state. This may seem surprising, but it is expected because one cannot achieve truly optimal steady-state operation (with $J_u = 0$ and zero loss) with unknown disturbances and static measurement bias (nonzero n^y).

In summary, the steady-state loss L in Equation (4.7) is minimized when we keep $\hat{J}_u = H^J(y_m - y^*) = 0$, and we have proven the following theorem:

Theorem 4.1. Optimal unconstrained gradient estimate. *Consider the static optimization problem in* (4.1) *with no active constraints, where the quadratic approximation* (4.3) *holds. The available measurements are* $y_m = G^y u + G^y_d d + n^y$ *(linear approximation) where the unknown disturbances* d *and static measurement errors* n^y *are bounded as given in* (4.5) *and* (4.6)*. Consider further that the point* (u^*, d^*) *is an optimal unconstrained point, such that* $J_u(u^*, d^*) = 0$*,* $u^* = u^{opt}(d^*)$ *and* $y^* = y^{opt}(d^*)$*. The cost gradient* J_u *is then given in* (4.18) *and the estimate* $\hat{J}_u = H^J(y_m - y^*)$ *with* H^J *in* (4.13) *is an optimal estimate in the sense that adjusting the inputs* u *to make* $\hat{J}_u = 0$ *(e.g., by feedback control, see Figure 4.1) minimizes both the average and the worst-case value of the economic loss* (4.7)*.*

If there is no measurement error ($n^y = 0$, that is,$W_{n^y} = 0$) and we have a sufficient number of measurement ($n_y = n_u + n_d$) then instead of using $H = H^J$ from the exact local method, we may use H from the nullspace method (equation (4.14) with $M_n = J_{uu}^{-1/2}$). This gives H in the nullspace of F ($HF = 0$) and achieves zero loss for disturbances (with no measurement error), that is, the last term in (4.18) is zero. If we have additional measurements ($n_y > n_u + n_d$) then we may use H from the "extended nullpace method" (equation (4.15) with $M_n = J_{uu}^{-1/2}$) which uses the extra measurements to minimize also the second term in (4.18). However, in general we recommend using $H = H^J$ from the exact local method.

It gives the optimal balance between disturbances and measurement error (as it minimizes both the average and worst-case sum of last two terms in (4.18)) and importantly applies also to the case with fewer measurements ($n_y < n_u + n_d$).

4.5 Optimal gradient estimate for the constrained case

Now, we focus on the use of this result in the operation with changing active constraints. For that, we state the following:

Theorem 4.2. Optimal gradient estimate in constrained case. *The optimal unconstrained gradient estimate $\hat{J}_u = H^J(y_m - y^*)$ (Theorem 4.1) is optimal also in the constrained case when used in the first-order KKT conditions* (4.2)*. This also means that the optimal estimate of the reduced gradient (which should be zero at the optimal point) is ${N_\mathcal{A}}^T \hat{J}_u = {N_\mathcal{A}}^T H^J(y_m - y^*)$ where $N_\mathcal{A}$ is a basis for the nullspace of $g_{u,\mathcal{A}}$, that is, $g_{u,\mathcal{A}} N_\mathcal{A} = 0$, and $\mathcal{A}$ represents the set of active constraints.*

The theorem may seem straightforward and require no further proof since J_u in (4.2a) is the unconstrained gradient, and the gradient estimate $\hat{J}_u$ in (4.19) is the one that minimizes the loss in the unconstrained case for a given measurement set y. Nevertheless, in Appendix 4.A, we provide a detailed proof that controlling the reduced gradient estimate ${N_\mathcal{A}}^T \hat{J}_u$ minimizes the loss for the constrained case.

It is important to note that Equation (4.19) is valid when the nominal point (u^*, d^*) is an *optimal unconstrained* reference point. If the reference point has a non-zero gradient, the optimal gradient estimate takes the form (the reader is referred to Appendix 4.B for a derivation of this expression):

$$\hat{J}_u = H(y_m - y^*) + J_u^* \tag{4.20}$$

where $J_u^* = J_u(u^*, d^*)$ (obtained from the nonlinear model). Note here that both (4.19) and (4.20) can be written in the form:

$$\hat{J}_u = H y_m - c_s \tag{4.21}$$

where c_s is a constant (see Figure 4.1).

The simple gradient estimate in (4.19) and (4.20) avoids implementing a model-based estimator, for example, a dynamic Kalman filter, and thus greatly simplifies the practical use of feedback-based real-time optimization, which is based on the first-order KKT condition (4.2a).

The gradient estimate can be used in a wide array of RTO control applications. In particular, it may be used in the following approaches for optimal steady-state operation with changes in active constraints:

1. *Primal-dual approaches* (Krishnamoorthy 2021) based directly on the optimality condition (4.2a) with a (slow) update of the Lagrange multiplier λ. This may be done using a slow controller K_{dual} which controls the measured constraints by manipulating the dual variables (λ) and with max-selectors for switching active constraints, see Figure 4.2 (Dirza et al. 2021, Dirza and Skogestad 2024).

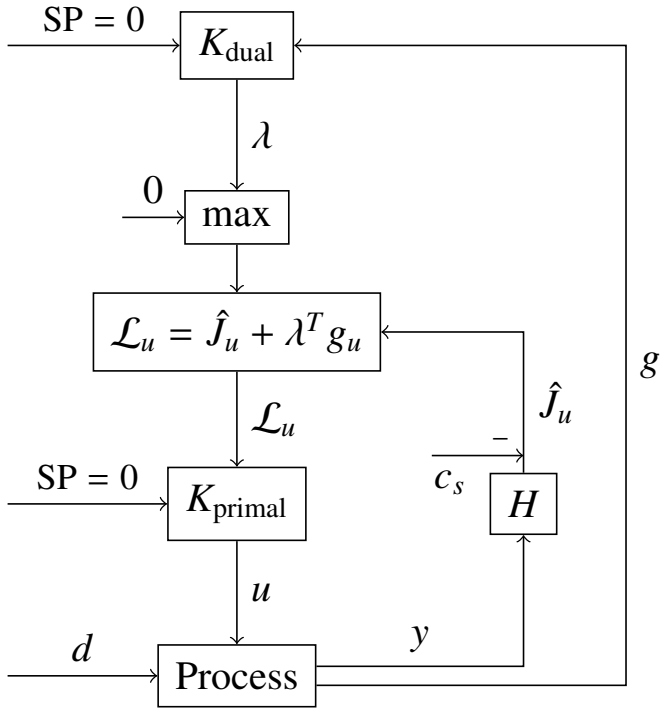


Figure 4.2: Primal-dual optimizing control structure using the proposed gradient estimate. The controller K_{dual} is always diagonal (decentralized), whereas the controller K_{primal} may be multivariable or diagonal.

2. *Region-based control* (Jäschke and Skogestad 2012, Krishnamoorthy and Skogestad 2022) where we in each region i control the active constraints and the associated reduced gradient $N_{\mathcal{A},i}^T \hat{J}_u$ to zero, see Figure 4.3.

2A. Region-based control may be applied to multivariable control, for example, model predictive control, by changing the cost function for designing the controller for each region, according to Bernardino and Skogestad (2024b). There, the gradient estimate is also used for constraint switching.

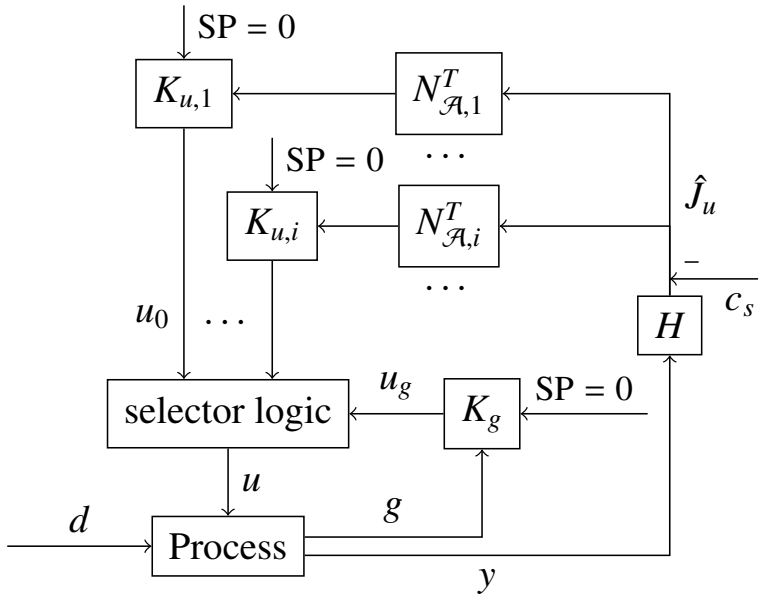


Figure 4.3: Region-based optimizing control structure using the proposed gradient estimate. In this scheme, each projection matrix $N_{\mathcal{A},i}$ is linked to a different set of active constraints $\mathcal{A}_i$, and the resulting gradient projection $N^T_{\mathcal{A},i}\hat{J}_u$ is controlled by a different controller $K_{u,i}$ (which in general is multivariable). If $n_u \geq n_g$, a fixed projection matrix can be used for all $\mathcal{A}_i$, and simple max/min-selectors can be used (see Figure 4.4).

2B. Decentralized region-based control with constraint switching using selectors (Bernardino et al. 2022a, Bernardino and Skogestad 2024c) (Figure 4.4). This approach requires at least as many inputs (degrees of freedom) as constraints, that is, $n_u \geq n_g$. An example of its application is given next.

In summary, the cost gradient estimate presented in Equation (4.20) can be used in a wide array of control applications focused on optimal operation, eliminating the need for a dynamic state estimator and thus greatly simplifying implementation.

4.6 Example: Decentralized region-based control

Here, we consider a system with more inputs than constraints ($n_u \geq n_g$) and design a region-based decentralized control structure with simple min/max-selectors (Figure 4.4) that minimizes the loss in all active constraint regions (Bernardino and Skogestad 2024c). In order to use simple switching, the nullspace associated with the unconstrained gradients (Theorem 4.2) needs to be selected in accordance with the constraint directions. This is done using the following steps (Bernardino and Skogestad 2024c):

- Define N^0 as an orthonormal basis for the nullspace of g_u, such that $g_u N^0 = 0$;
- Find $W = \begin{bmatrix} g_u \\ {N^0}^T \end{bmatrix}^{-1}$, and define the vectors N_i, $i = 1, \cdots, n_g$ as the first n_g normalized columns of W.

Then, controlling the active constraints g_i, for $i \in \mathcal{A}$ and the remaining unconstrained degrees of freedom $N_i^T J_u$, for $i \notin \mathcal{A}$, and ${N^0}^T J_u$ will lead to optimal operation (Bernardino and Skogestad 2024c). The final simple decentralized control system with min or max selectors can be implemented as shown in Figure 4.4 where all controllers (K) are single-input single-output (SISO), for example, PID controllers. The controllers linked to selectors must have anti-windup action, to cancel the integral action when the controllers are inactive.

As a case study, we consider a linear dynamic system with a quadratic cost function given by:

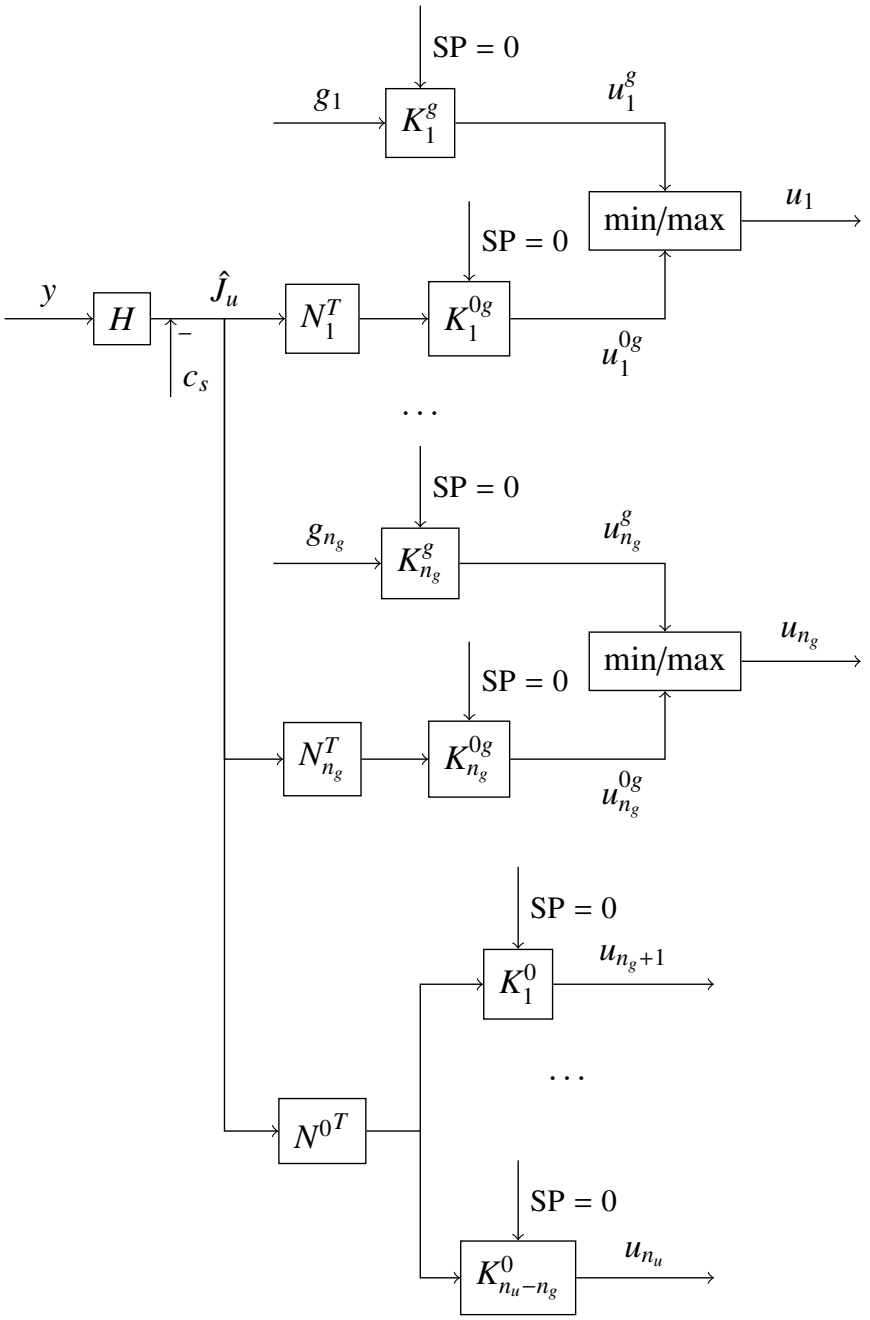


Figure 4.4: Decentralized region-based optimizing control structure using SISO controllers and selectors.

$$\begin{aligned} \min_{u} \quad & \frac{1}{2}x^T \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} x + \frac{1}{2}u^T \begin{bmatrix} 1 & -0.1 & -0.2 \\ -0.1 & 0.8 & -0.1 \\ -0.2 & -0.1 & 0.3 \end{bmatrix} u \\ \text{s.t.} \quad & \begin{cases} g_1 = x_1 - 0.8x_2 \leq 0 \\ g_2 = u_1 + u_2 + u_3 \leq 0 \end{cases} \end{aligned} \tag{4.22}$$

$$\dot{x} = \begin{bmatrix} -\frac{1}{\tau_1} & 0 \\ 0 & -\frac{1}{\tau_2} \end{bmatrix} x + \begin{bmatrix} \frac{0.2}{\tau_1} & 0 & 0 \\ 0 & \frac{0.2}{\tau_2} & 0 \end{bmatrix} u + \begin{bmatrix} \frac{1}{\tau_1} & 0 \\ 0 & \frac{1}{\tau_2} \end{bmatrix} d \tag{4.23}$$

with $\tau_1 = 1$ and $\tau_2 = 2$. The set of optimal active constraint regions can be visualized as a function of the two disturbances as shown in Figure 4.5. Here, the upper left green region is unconstrained and the lower middle grey region is with all constraints being active (and one unconstrained degree of freedom).

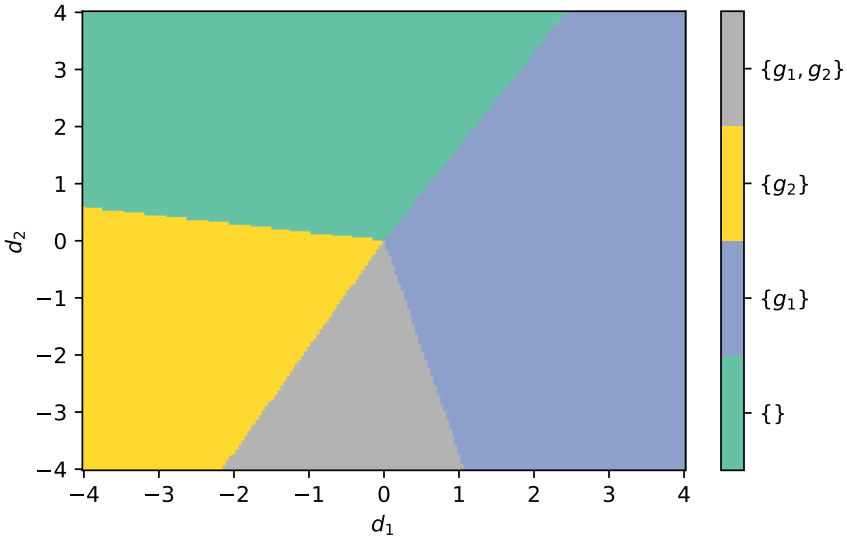


Figure 4.5: Active constraint regions as a function of disturbances for case study 1

For estimating the cost gradient, the following measurements are available:

$$y = \begin{bmatrix} g_1 \\ g_2 \\ x_1 \\ x_2 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 & -0.8 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} u \tag{4.24}$$

Note that both constraints and both states are measured. In addition, we choose to include two of the three inputs. The expected static disturbance and noise magnitudes are $W_d = \text{diag}([4,\ 4])$ and $W_{n^y} = \text{diag}([0,\ 0,\ 1,\ 2,\ 1.5,\ 5])$. The two first zeros in W_{n^y} imply that the constraints have no static measurement error, that is, the constraints can be perfectly controlled. In general, static measurement error for a constraint may be counteracted by using back-off for its setpoint, but this issue is not explored in the case study.

To find the optimal cost gradient estimate using the formulation proposed in this work, we first use (4.23) with $\dot{x} = 0$ to derive the steady-state relationship:

$$x = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \end{bmatrix} u + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} d \tag{4.25}$$

This is used to eliminate the states x from the problem (4.22), resulting in the following steady-state optimization problem:

$$\begin{aligned} \min_{u} \quad & J = \frac{1}{2}\, u^T \underbrace{\begin{bmatrix} 1.04 & -0.1 & -0.2 \\ -0.1 & 1.2 & -0.1 \\ -0.2 & -0.1 & 0.3 \end{bmatrix}}_{J_{uu}} u + u^T \underbrace{\begin{bmatrix} 0.2 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix}}_{J_{ud}} d \\ \text{s.t.} \quad & g = \underbrace{\begin{bmatrix} 0.2 & -0.16 & 0 \\ 1 & 1 & 1 \end{bmatrix}}_{g_u} u + \begin{bmatrix} 1 & -0.8 \\ 0 & 0 \end{bmatrix} d \leq 0 \end{aligned} \tag{4.26}$$

From the matrix g_u, we can find the projections N_i and N^0 to be multiplied with the unconstrained gradient J_u. N^0 is the nullspace of g_u given by:

$$N^0 = \begin{bmatrix} -0.36214 & -0.45268 & 0.81482 \end{bmatrix}^T \tag{4.27}$$

The vectors N_i are the first n_g normalized columns of $W = \begin{bmatrix} g_u \\ N^{0T} \end{bmatrix}^{-1}$, calculated as:

$$W = \begin{bmatrix} 2.8689 & 0.29508 & -0.36214 \\ -2.6639 & 0.36885 & -0.45267 \\ -0.20491 & 0.33607 & 0.81482 \end{bmatrix} \tag{4.28}$$

$$N_1 = \begin{bmatrix} 0.73179 & -0.67952 & -0.052271 \end{bmatrix}^T \tag{4.29}$$

$$N_2 = \begin{bmatrix} 0.50902 & 0.63627 & 0.57971 \end{bmatrix}^T \tag{4.30}$$

To estimate the gradient from the measurements, we also need their corresponding steady-state model. Plugging the steady-state expression for the states into (4.24) leads to:

$$y = \underbrace{\begin{bmatrix} 0.2 & -0.16 & 0 \\ 1 & 1 & 1 \\ 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{G^y} u + \underbrace{\begin{bmatrix} 1 & -0.8 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{G_d^y} d \tag{4.31}$$

The optimal sensitivity is then:

$$F = \frac{dy^{opt}}{dd} = G_d^y - G^y J_{uu}^{-1} J_{ud} = \begin{bmatrix} 0.9599 & -0.5830 \\ -0.4207 & -2.8867 \\ -0.0065 & 0.6479 \\ -0.0324 & -1.7605 \\ -0.1618 & -0.8026 \\ 0.9547 & -0.0647 \end{bmatrix} \tag{4.32}$$

With this information and the matrices from Equation (4.26), we can calculate the measurement combinations H^J from Equation (4.13), which gives:

$$H^J = \begin{bmatrix} 0.2741 & 0.9842 & 0.1560 & -1.0715 & -1.1842 & 0.0050 \\ -0.1897 & -0.0735 & 1.7813 & 0.8869 & -0.0265 & 0.0570 \\ -0.0180 & -0.1964 & -0.0091 & 0.0953 & 0.4964 & -0.0003 \end{bmatrix} \tag{4.33}$$

and the estimated gradient is $\hat{J}_u = H^J(y - y^*) = H^J - c_s$. Here, we note that the approximations in (4.3) and (4.4) are exact for this example, and therefore H^J does not depend on the nominal point to be considered. However, we still need a reference point to calculate the constant $c_s = H^J y^*$, and for that, we choose an optimal point with $d^* = [0,\ 0]^T$. This gives $c_s = [0,\ 0,\ 0]^T$.

Dynamic simulation results for the closed-loop system with the proposed control structure in Figure 4.4 with $H = H^J$ are shown in Figure 4.6. The PI controllers tuning are given in Table 4.1. The simulated disturbances cover all four active constraint regions but we did not include measurement noise. The responses are fairly smooth (see the three input profiles) and there are as expected three changes in active constraints. The gradient estimate with $H = H^J$ is optimal in terms of

minimizing the average loss with the expected (assumed) disturbances and noise. However, this means that the gradient estimates (and resulting CVs) are not designed to reject the disturbances completely, as they simultaneously try to reduce the effect of measurement noise. This is the reason why the resulting steady-state inputs u_i (blue lines) do not match exactly the corresponding optimal values (magenta dashed lines). At steady state, the economic loss L resulting from this input mismatch is, however, very small.

Controller	Parameter	Value
K_1^g	K_c	50
	K_I	50
K_2^g	K_I	100
K_1^{0g}	K_I	-1.191
K_2^{0g}	K_I	1.528
K^0	K_I	2.761

Table 4.1: Proportional and integral gains of controllers for the example. All controllers have anti-windup with tracking time $\tau_T = 0.01$.

In Figure 4.7, we present the steady-state loss obtained in closed loop both without and with static measurement noise (bias). The loss is shown as a heatmap for each disturbance combination. The much larger loss (note the difference in scale) with measurement noise in Figure 4.7(b) is for the worst-case measurement error satisfying $n^y = W_{n^y} n^{y\prime}$ with $||n^{y\prime}||_2 \leq 1$. The optimal active constraint regions (same as Figure 4.5) are shown by black lines whereas the actual operating regions resulting from using the control structure are shown by blue lines. Note that the constraint switching is moved away from the optimal, which is not surprising (see discussion).

Figure 4.7(a) shows that the measurement combination $H = H^J$ (which is based on the exact local method of self-optimizing control) does not perfectly reject disturbances, even without measurement error. To achieve zero loss for disturbances, H must be in the nullspace of F. For instance, if we apply the extended nullspace method (4.15) to this problem (with $M_n = J_{uu}^{-1/2}$), we get:

$$H = \begin{bmatrix} 0.195 & 1 & 0.156 & -1.1 & -1.2 & 0.005 \\ -0.0624 & -0.1 & 1.95 & 0.9 & 0 & 0.0624 \\ 0 & -0.2 & 0 & 0.1 & 0.5 & 0 \end{bmatrix} \tag{4.34}$$

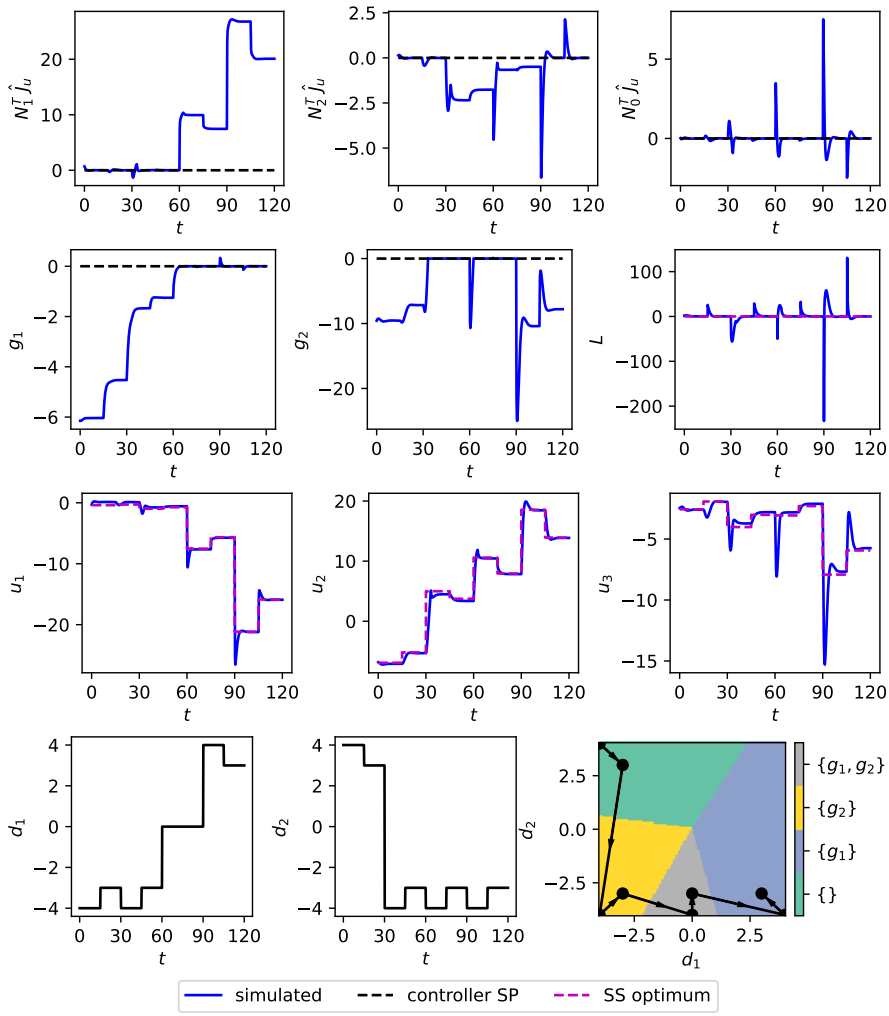
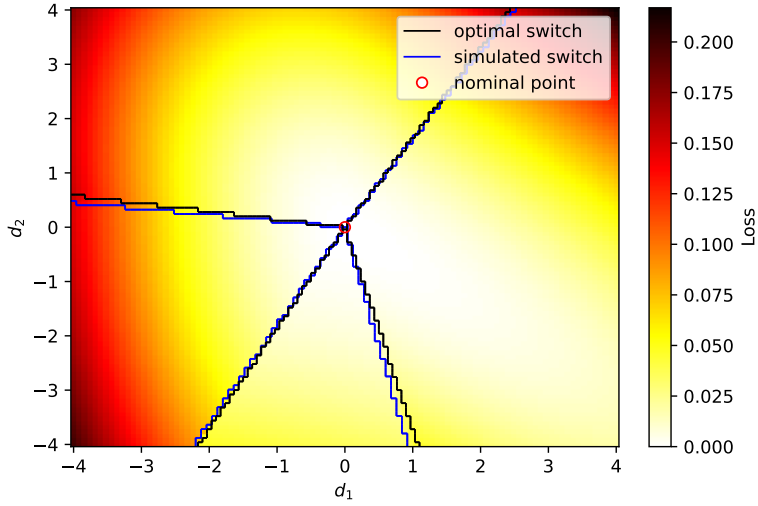
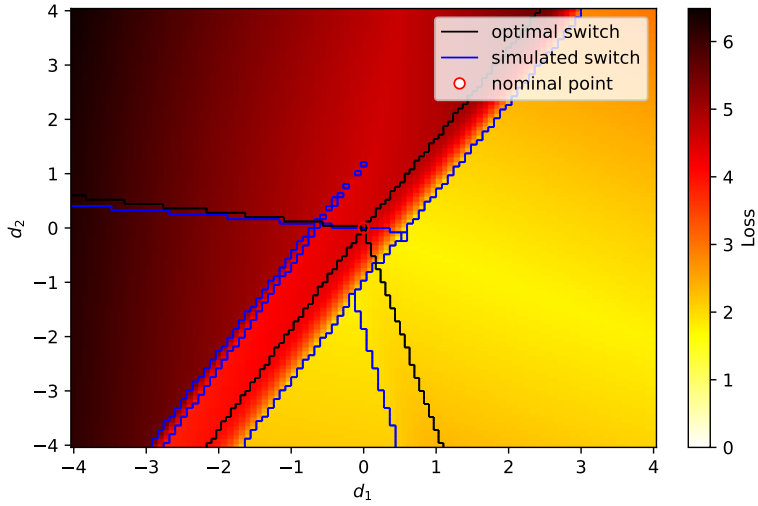


Figure 4.6: Dynamic simulation over all active constraint regions using the proposed control structure with $H = H^J$ (exact local method).



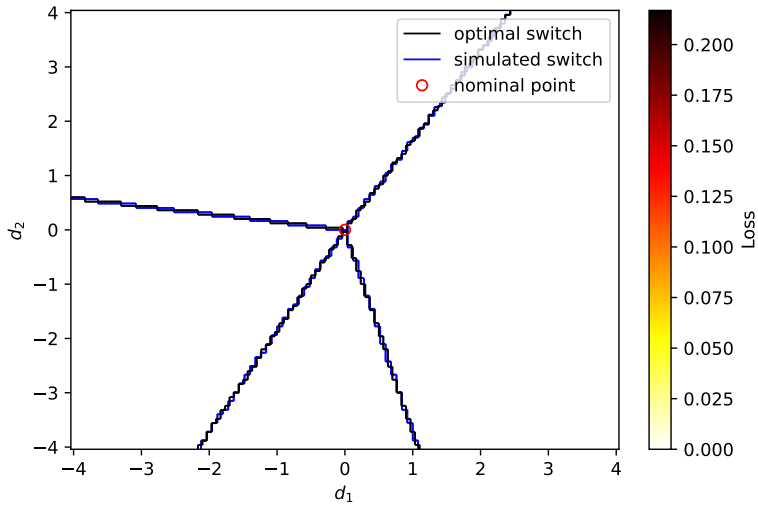
(a) Loss without measurement error ($n^y = 0$).



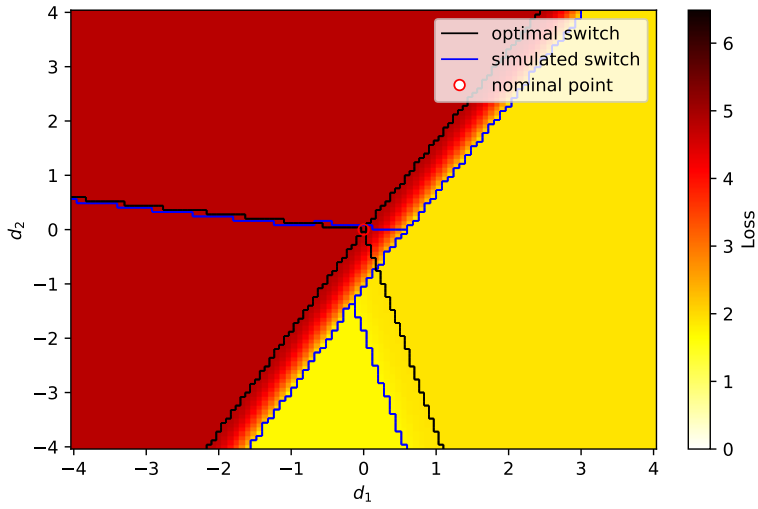
(b) Worst-case loss with measurement error. The new narrow operating region (which starts from point $d = [-2.8,\ -4]^T$) has both constraints g_1 and g_2 active.

Figure 4.7: Steady-state loss for closed-loop operation with $H = H^J$ from the exact local method.

With the resulting gradient estimate (and set of CVs), the steady-state closed loop loss for the extended nullspace method (without noise and with the worst-case noise) are presented in Figure 4.8. We see that in the case without noise (Figure 4.8(a)), the economic loss is exactly zero in all constraint regions. This is expected since the original problem is linear with a quadratic cost. However, we see that the exact local method (H^J) is better at locally rejecting noise (note that the worst-case loss in Figure 4.7(b) is smaller around the nominal point), but the extended nullspace method (Figure 4.8(b)) handles large disturbances better, as expected.



(a) Zero loss without measurement error ($n^y = 0$).



(b) Worst-case loss with measurement error.

Figure 4.8: Steady-state loss for closed-loop operation with H from the extended null-space method

4.7 Discussion

4.7.1 Local gradient estimation (block-diagonal H)

The matrix H^J in (4.14) for the optimal gradient estimate is a full matrix. This means that control systems in Figures 4.2 and 4.4 may not be decentralized, even if the controllers K themselves are decentralized. To obtain a decentralized control system, the relationship from y (measurements) to u (inputs) needs to be decoupled. For example, if we have a complex process with many units, then decentralized control implies that only measurements from unit k should be used by the control system to compute the inputs for unit k. To accomplish this, the matrix H needs to be block-diagonal. There exists no analytical solution in this case so the optimal block-diagonal H must be obtained numerically. Depending on the case study, there may be a small or large performance loss compared to using a full H. This problem has been studied in detail by Yelchuru and Skogestad (2012) using mixed-integer quadratic programming (MIQP). However, their objective was to find self-optimizing controlled variables $c = Hy_m + c_s$, so their results need to be modified to estimate instead the gradient, $\hat{J}_u = Hy_m + c_s$.

Finally, note that for the primal-dual optimizing control structure in Figure 4.2, the Lagrange multiplier (λ) may introduce a coupling from the measured constraint (g) to the inputs (u) even for cases where the gradient controller (K_{primal}) is diagonal and the gradient estimator (matrix H) is block-diagonal.

4.7.2 Addition of RTO layer

The optimality of the static gradient estimate is based on a quadratic approximation (4.3) of the cost, and a linear approximation of the constraints (4.4) and of the measurement model (4.9). In general, these assumptions are not satisfied, and in this case, a static real-time optimization layer may be used to provide updates of the constants presented in this work, namely the controller setpoints c_s, the measurement combinations H, and the projection matrices N_i and N^0 (or $N_{\mathcal{A}}$ when generalizing to centralized approaches).

Using the RTO layer to update the setpoints c_s is the simplest and most important, being sufficient to drive the system to optimality in a new operating condition. That is, c_s is optimally updated, while the matrix H and the projection matrices constant can be kept constant. The use of constant matrices implies the self-optimizing properties (related to optimality on a shorter time scale) may degrade somewhat in a new operating point. On the other hand, changing these matrices will affect the control problem and, consequently, the controllers' tuning that should be used. Thus, updating only c_s is recommended in most practical applications.

As an alternative to model-based RTO, data-based methods based on perturbing the process, for example, extremum-seeking control may be used to update c_s. However, data-based methods are not realistic for most process control applications because the convergence of these methods is too slow to track changing disturbances. Regardless, these methods are complementary to the method discussed in this work, as they are applied on an upper layer.

4.7.3 Required model information

The methods for self-optimizing control usually only need model information in the form of the matrices F and G^y, which can be estimated from plant data with relative ease. For estimating the gradient J_u, we additionally require knowledge of the Hessian matrix J_{uu} so that the directions of the unconstrained gradient are retrieved. The Hessian is harder to estimate from measurement information, as it requires more data. In addition, the constraint gradient g_u is needed to find the nullspace matrix for the reduced gradient $N_{\mathcal{A}}^T J_u$, but g_u is easy to estimate from data. However, if a steady-state model is available for control structure design, all of these matrices can easily be obtained.

4.7.4 Discussion of case study

In this work, we illustrate the method with a case study where the formulation is exact, that is, Equations (4.3), (4.4) and (4.9) hold. It was shown that the exact local method (4.13) is not designed to perfectly reject disturbances, that is $\Delta c_{opt} = HF\Delta d \neq 0$, which results in non-zero loss as shown in Figure 4.7(a). Therefore, if a new estimate of the disturbances is available, an update of c_s will lead to improved performance around the new operating point, even if the optimal H is unchanged. This is not the case for the extended nullspace method, where we see in Figure 4.8(a) that the obtained loss is zero for all disturbance values, which means that the optimal setpoint value is constant, i.e. $\Delta c_{opt} = HF\Delta d = 0$.

We see from Figure 4.7 that measurement bias has a comparatively bigger effect on the economic loss than the disturbances in this numerical example, which is worsened the further the disturbances are from their design value. We also see that the measurement bias may trigger control of constraints that are not optimally active, which could be a problem if there were no constraint controllers. This is the reason why the pattern of the operating regions is so different from the optimal in Figure 4.7(b). Overall, we see that for the nominal case (Figure 4.7(a)), optimal behavior is well captured, with the closed-loop operating regions closely resembling the optimal active constraint regions. For the worst-case loss (Figure 4.7(b)), the resulting economic loss is still small when compared to the values attained dynamically in Figure 4.6.

4.8 Conclusion

The optimal local gradient estimate for use in steady-state real-time optimization is simply $\hat{J}_u = H^J(y_m - y^*) + J_u^*$ with H^J as in Equation (4.13) (Theorem 4.1). This gradient estimate is optimal also in the constrained case when used with the KKT optimality conditions (4.2) (Theorem 4.2). The gradient estimate $\hat{J}_u$ may be used in a multitude of control applications (Figures 4.1 to 4.4) where it is desired to include the optimality conditions (4.2) directly into the feedback control layer.

Funding

This work was funded by the Research Council of Norway through the IKTPLUSS programme (project number 299585).

4.A Proof: Optimal gradient estimate for the constrained case

We begin by describing the loss function for the constrained optimization problem, resulting in a simple form. Then, we show that the ideal variables for a given set of active constraints are the projection of the unconstrained gradient estimate onto the nullspace of the gradient of the active constraints, in the sense that they minimize the expected loss.

4.A.1 Loss for constrained optimization problem

From Equation (4.3), we have:

$$\begin{aligned} L = J(u,d) - J^{opt}(d) = J_u^{*T}(u - u^{opt}) \\ + \frac{1}{2}(u-u^*)^T J_{uu}(u-u^*) + (d-d^*)^T J_{ud}^T(u-u^{opt}) \\ - \frac{1}{2}(u^{opt}-u^*)^T J_{uu}(u^{opt}-u^*) \end{aligned}$$

$$\begin{aligned} L = \left(J_u^* + J_{ud}(d-d^*)\right)^T (u-u^{opt}) + \frac{1}{2}(u-u^*)^T J_{uu}(u-u^*) \\ - \frac{1}{2}(u^{opt}-u^*)^T J_{uu}(u^{opt}-u^*) \end{aligned} \tag{4.35}$$

The optimality conditions state that:

$$\begin{aligned}
&\mathcal{L}_u(u^{opt}, d, \lambda^{opt}) = J_u(u^{opt}, d) + {g_u^*}^T \lambda^{opt} = 0 \\
&\implies J_u^* + J_{uu}(u^{opt} - u^*) + J_{ud}(d - d^*) + {g_u^*}^T \lambda^{opt} = 0 \\
&\implies J_u^* + J_{ud}(d - d^*) = -(J_{uu}(u^{opt} - u^*) + {g_u^*}^T \lambda^{opt})
\end{aligned} \tag{4.36}$$

We can therefore rewrite Equation (4.35) as:

$$\begin{aligned}
L =& -\left(J_{uu}(u^{opt} - u^*) + {g_u^*}^T \lambda^{opt}\right)^T (u - u^{opt}) \\
&+ \frac{1}{2}(u - u^*)^T J_{uu}(u - u^*) - \frac{1}{2}(u^{opt} - u^*)^T J_{uu}(u^{opt} - u^*) \\
=& -{\lambda^{opt}}^T g_u^*(u - u^{opt}) + \frac{1}{2}(u - u^*)^T J_{uu}(u - u^*) \\
&- \frac{1}{2}(u^{opt} - u^*)^T J_{uu}(u^{opt} - u^*) - (u^{opt} - u^*)^T J_{uu}(u - u^{opt}) \\
=& -{\lambda^{opt}}^T g_u^*(u - u^{opt}) + \frac{1}{2}(u - u^*)^T J_{uu}(u - u^*) \\
&- \frac{1}{2}(u^{opt} - u^*)^T J_{uu}(u^{opt} - u^*) - (u^{opt} - u^*)^T J_{uu}(u - u^*) \\
&+ (u^{opt} - u^*)^T J_{uu}(u^{opt} - u^*) \\
=& -{\lambda^{opt}}^T g_u^*(u - u^{opt}) + \frac{1}{2}(u - u^*)^T J_{uu}(u - u^*) \\
&- (u^{opt} - u^*)^T J_{uu}(u - u^*) + \frac{1}{2}(u^{opt} - u^*)^T J_{uu}(u^{opt} - u^*)
\end{aligned}$$

From this, we conclude that:

$$L = \frac{1}{2}(u - u^{opt})^T J_{uu}(u - u^{opt}) - {\lambda^{opt}}^T g_u^*(u - u^{opt}) \tag{4.37}$$

This expression is very similar to Equation (4.7), the difference being the linear term ${\lambda^{opt}}^T g_u^*(u - u^{opt})$, which is related to constraint control. Because the optimal Lagrange multipliers for the inactive constraints are zero, we have that ${\lambda^{opt}}^T g_u^*(u - u^{opt}) = {\lambda_{\mathcal{A}}^{opt}}^T g_{u,\mathcal{A}}(u - u^{opt})$, with $g_{u,\mathcal{A}}$ defined as the gradient of the active constraints with respect to the inputs. If the optimal active constraint set $\mathcal{A}$ is perfectly controlled, we have:

$$\begin{cases} g_{\mathcal{A}}(u^{opt}, d) = g^*_{\mathcal{A}} + g_{u,\mathcal{A}}(u^{opt} - u^*) + g_{d,\mathcal{A}}(d - d^*) = 0 \\ g_{\mathcal{A}}(u, d) = g^*_{\mathcal{A}} + g_{u,\mathcal{A}}(u - u^*) + g_{d,\mathcal{A}}(d - d^*) = 0 \end{cases} \\ \implies g_{u,\mathcal{A}}(u - u^{opt}) = 0 \tag{4.38}$$

This means that only the quadratic term on Equation (4.37) is relevant when the correct constraints are controlled, with the additional restriction on the allowed directions of $(u - u^{opt})$, which are in the nullspace of $g_{u,\mathcal{A}}$. Define $N_{\mathcal{A}}$ as a basis for the nullspace of $g_{u,\mathcal{A}}$. This means that the loss from Equation (4.37) is further simplified when the correct constraints are controlled to give:

$$L = \frac{1}{2}(u - u^{opt})^T J_{uu}(u - u^{opt}) = \frac{1}{2} w^T {N_{\mathcal{A}}}^T J_{uu} N_{\mathcal{A}} w \tag{4.39}$$

Here, w is an appropriately sized vector that represents the unconstrained degrees of freedom.

4.A.2 Connection with the unconstrained problem

We now show that the ideal controlled variables for this problem are directly linked to the ones from the unconstrained problem. First, note that the matrix $J_{ww} = {N_{\mathcal{A}}}^T J_{uu} N_{\mathcal{A}}$ is invertible by definition, and therefore we can write:

$$L = \frac{1}{2} w^T J_{ww} J_{ww}^{-1} J_{ww} w \tag{4.40}$$

From this, we can see that the loss variable z_w for this problem can be represented by:

$$z_w = J_{ww}^{-1/2} {N_{\mathcal{A}}}^T J_{uu} N_{\mathcal{A}} w = J_{ww}^{-1/2} {N_{\mathcal{A}}}^T J_{uu}(u - u^{opt}) \tag{4.41}$$

Similarly to Equation (4.8), we can write z_w in terms of the unconstrained CVs c as:

$$\begin{aligned} z_w &= J_{ww}^{-1/2} {N_{\mathcal{A}}}^T J_{uu} (H G^y)^{-1} (c - c^{opt}) \\ &= J_{ww}^{-1/2} {N_{\mathcal{A}}}^T J_{uu}^{1/2} M_n (c - c^{opt}) \end{aligned}$$

We can similarly write z_w in terms of the unconstrained gradient:

$$J_u = J_u(u^{opt}, d) + J_{uu}(u - u^{opt})$$
$$\implies z_w = J_{ww}^{-1/2} {N_{\mathcal{A}}}^T (J_u - J_u(u^{opt}, d))$$

Note that, because of the optimality conditions, we have that:

$$J_u(u^{opt}, d) + {g_u^*}^T \lambda^{opt} = 0 \implies {N_{\mathcal{A}}}^T J_u(u^{opt}, d) = 0$$

and with the choice of $M_n = J_{uu}^{-1/2}$, we compare both expressions for z_w and we see that:

$$\begin{aligned} &{N_{\mathcal{A}}}^T J_u = {N_{\mathcal{A}}}^T (c - c^{opt}) \\ &= {N_{\mathcal{A}}}^T \left(H(y_m - y^*) - Hn^y - H(y^{opt}(d) - y^*) \right) \end{aligned} \tag{4.42}$$

This formulation is similar to that of Equation (4.18), with the exception that now $u^{opt}(d)$ and $y^{opt}(d)$ represent a constrained optimal point, and therefore are a different function of the disturbances, $(y^{opt}(d) - y^*) = F_{\mathcal{A}}(d - d^*)$. We can determine $F_{\mathcal{A}}$ from the constrained optimization problem as follows:

$$\begin{bmatrix} J_{uu} & g_{u,\mathcal{A}}^T \\ g_{u,\mathcal{A}} & 0 \end{bmatrix} \begin{bmatrix} \Delta u^{opt} \\ \Delta \lambda_{\mathcal{A}}^{opt} \end{bmatrix} = \begin{bmatrix} -J_{ud} \\ -g_{d,\mathcal{A}} \end{bmatrix} \Delta d \tag{4.43}$$

First we eliminate Δu^{opt} by premultiplying both sides by $\begin{bmatrix} g_{u,\mathcal{A}} J_{uu}^{-1} & -I \end{bmatrix}$, leading to the solution $\Delta \lambda_{\mathcal{A}}^{opt} = W_{\mathcal{A}} \Delta d$, where

$$W_{\mathcal{A}} = \left(g_{u,\mathcal{A}} J_{uu}^{-1} g_{u,\mathcal{A}}^T \right)^{-1} \left(g_{d,\mathcal{A}} - g_{u,\mathcal{A}} J_{uu}^{-1} J_{ud} \right)$$

The solution for the new optimal inputs follows as:

$$\Delta u^{opt} = -J_{uu}^{-1} \left((g_{u,\mathcal{A}})^T W_{\mathcal{A}} + J_{ud} \right) \Delta d$$

and the optimal sensitivity matrix $F_{\mathcal{A}}$ can be obtained as:

$$F_{\mathcal{A}} = F - G^y J_{uu}^{-1} g_{u,\mathcal{A}}^T W_{\mathcal{A}} \tag{4.44}$$

with F being the unconstrained optimal sensitivity matrix. The second term of $F_{\mathcal{A}}$ is related to constraint control, and we can see that, with $M_n = J_{uu}^{-1/2}$:

$$
\begin{aligned}
{N_{\mathcal{A}}}^T H F_{\mathcal{A}} &= {N_{\mathcal{A}}}^T H F - {N_{\mathcal{A}}}^T \overbrace{H G^y J_{uu}^{-1}}^{=I} g_{u,\mathcal{A}}^T W_{\mathcal{A}} \\
&= {N_{\mathcal{A}}}^T H F - \overbrace{{N_{\mathcal{A}}}^T g_{u,\mathcal{A}}^T}^{=0} W_{\mathcal{A}} = {N_{\mathcal{A}}}^T H F
\end{aligned}
$$

This means that the last two terms in Equation (4.42) are minimized by the unconstrained self-optimizing control solution for $H = H^J$ (4.13), and therefore the reduced gradient estimate

$$
{N_{\mathcal{A}}}^T \hat{J}_u = {N_{\mathcal{A}}}^T H^J (y_m - y^*) \tag{4.45}
$$

is the unconstrained CV that should be kept at zero to minimize the expected norm of z_w.

4.B Effect of nominal setpoint

Here, we evaluate the effect of having a non-optimal reference point. From Equation (4.16) and choosing $M_n = J_{uu}^{-1/2}$, we have:

$$
c(u,d) - c(u^{opt}(d),d) = J_u(u,d) - J_u(u^{opt}(d),d)
$$

The same expression is valid for the nominal point, according to:

$$
c(u^*,d^*) - c(u^{opt}(d^*),d^*) = J_u(u^*,d^*) - J_u(u^{opt}(d^*),d^*)
$$

Here, we assume that $u^* \neq u^{opt}(d^*)$, that is, the nominal point is not optimal. For the unconstrained problem, $J_u(u^{opt}(d),d) = J_u(u^{opt}(d^*),d^* = 0$, and we subtract the two equations to give:

$$
\begin{aligned}
J_u(u,d) = J_u(u^*,d^*) &+ c(u,d) - c(u^*,d^*) \\
&- \big(c(u^{opt}(d),d) - c(u^{opt}(d^*),d^*)\big)
\end{aligned}
$$

or

$$
J_u(u,d) = J_u(u^*,d^*) + H(y^m - y^*) - H \underbrace{(y_m - y)}_{n^y} - HF(d - d^*) \tag{4.46}
$$

Choosing the exact local method solution for H from (4.13), we minimize the last two terms from the previous equation, and the optimal gradient estimate to be controlled is given by:

$$\hat{J}_u(u,d) = H^J(y^m - y^*) + J_u(u^*, d^*)$$

as stated in Equation (4.20). As previously shown, this gradient estimate is also valid for the constrained region, with the corresponding reduced gradient estimate being the optimal variable to be controlled.

Chapter 5

Optimal operation of heat exchanger networks with changing active constraint regions

This chapter has been published:

> L. F. Bernardino, D. Krishnamoorthy, and S. Skogestad. Optimal operation of heat exchanger networks with changing active constraint regions. In *Computer Aided Chemical Engineering*, volume 49, pages 421–426. Elsevier, 2022b

5.1 Introduction

In the context of optimal operation of process systems, the choice of controlled variables plays a vital role, as it will dictate how efficiently a process can operate without interference of higher layers (Skogestad 2000). The ideal design of a supervisory control layer would result in a structure that is able to operate optimally under constant setpoints. This concept is known as self-optimizing control, and recent developments aim for systematic choice of control objectives (Krishnamoorthy and Skogestad 2019). A known challenge in supervisory layer design is the change in optimally active constraints during operation, which can be caused by changes in disturbances that affect process objectives. When that happens, reconfiguration of the controlled structure is usually desired to minimize the oper-

ational losses. If that does not happen, interactions with the higher optimization layer become stronger, as the sensitivity of the optimal setpoint values with relation to the changing disturbances is high when there are no changes in the control structure. Krishnamoorthy and Skogestad (2019) discusses the handling of changes in active constraints through feedback control, without the solution of online optimization problems, by selector-based control structures. This approach is to be evaluated in this work, compared to the solution of real-time optimization (RTO) problems, which can be problematic in the presence of model-plant mismatch.

5.2 Case study modeling

The case study considered in this work consists of three heat exchangers in parallel, see Figure 5.1. Each exchanger has its own source of hot fluid, such that the cold fluid is split and sent to the exchangers, and the operational goal is to maximize the outlet temperature of the cold fluid, subject to constraints related to the maximum temperature in the individual exchangers.

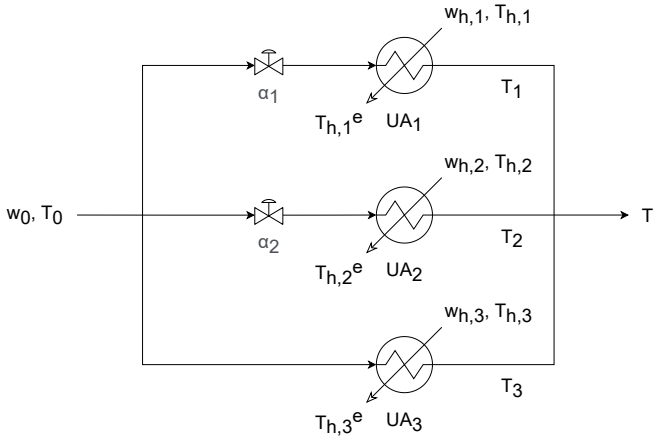


Figure 5.1: Heat exchanger network scheme

In addition to the mass and energy balances, an additional relation is necessary for calculating the total exchanged heat in each equipment, Q_i. The analytic solution, assuming constant heat capacities and countercurrent flow, is given by Eq.(5.1).

$$Q_i = UA_i \, \Delta T_{LM,i} \tag{5.1}$$

In this equation, $\Delta T_{LM,i}$ represents the logarithmic mean of temperature differences inside the heat exchanger. Although exact, this model presents some numerical challenges, especially when the heat capacities are too close, or when the

temperature differences assume opposite signs during iteration. A simplified linear version of this model makes use of the arithmetic mean of temperature differences, $\Delta T_{AM,i}$, and for this model, simple analytic expressions for the gradient can be derived (Jäschke and Skogestad 2014).

The steady-state optimization problem considered for the optimal operation of this system can therefore be written as:

$$\begin{aligned} \min_{\alpha} \quad & J = -T \\ \text{s.t.} \quad & g_i = T_i - T_{max} \leq 0, \quad i = 1, 2, 3 \end{aligned} \tag{5.2}$$

5.3 Proposed control structure

The optimal operation of heat exchanger networks has been extensively studied by Jäschke and Skogestad (2014) for the unconstrained case. In this case, the gradient J_u to be driven to zero can be approximately written in terms of the Jäschke temperatures. For the constrained case, however, the set of controlled variables need to change so that optimal operation is achieved. Given that the active constraints g_A are effectively controlled, there are still unconstrained degrees of freedom that need to be used for optimal operation. As proven by Krishnamoorthy and Skogestad (2019), we can find the additional controlled variables as a linear combination of the gradient such that the necessary conditions of optimality are satisfied. These correspond to $c = N^T J_u$, where N is the nullspace of the gradient of the active constraints with relation to the inputs, $\nabla_u g_A$, at the optimal point. This procedure results in a set of controlled variables per region, defined by the respective set of active constraints.

For this case study, there are 7 feasible operating regions, one of which is fully unconstrained, 3 being partially constrained (one active constraint per region), and the remaining being fully constrained (two active constraints per region). The case with all 3 constraints being active is infeasible with the available degrees of freedom, and will therefore not be considered. The fully unconstrained region can be optimally operated by controlling the plant gradient to zero, and the fully constrained regions are optimally operated through active constraint control. For the optimal operation in the partially constrained regions, the combinations of the gradient to be controlled in addition to the active constraints are given in Table 5.1.

The next step for the design of a simple control structure is defining the pairing between manipulated and controlled variables, and the switching between active controllers. In the current case study, there are 2 manipulated variables and 3 constraints, which means that the constraints cannot be assigned to one specific

Active constraint	N^T
g_1	$\begin{bmatrix} 0 & 1 \end{bmatrix}$
g_2	$\begin{bmatrix} 1 & 0 \end{bmatrix}$
g_3	$\begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

Table 5.1: Linear combinations of gradient per active constraint

input if optimal operation over all regions is desired. Therefore, at least one of the constraints needs to be controlled by multiple inputs.

Based on this reasoning, this work proposes an adaptive control structure to deal with all possible active constraint regions. The full control structure, showing the logic blocks and controllers, is presented in Figure 5.2. and the pairing between manipulated and controlled variables is summarized in Table 5.2. All presented controllers have integral action, so that steady-state offset is eliminated.

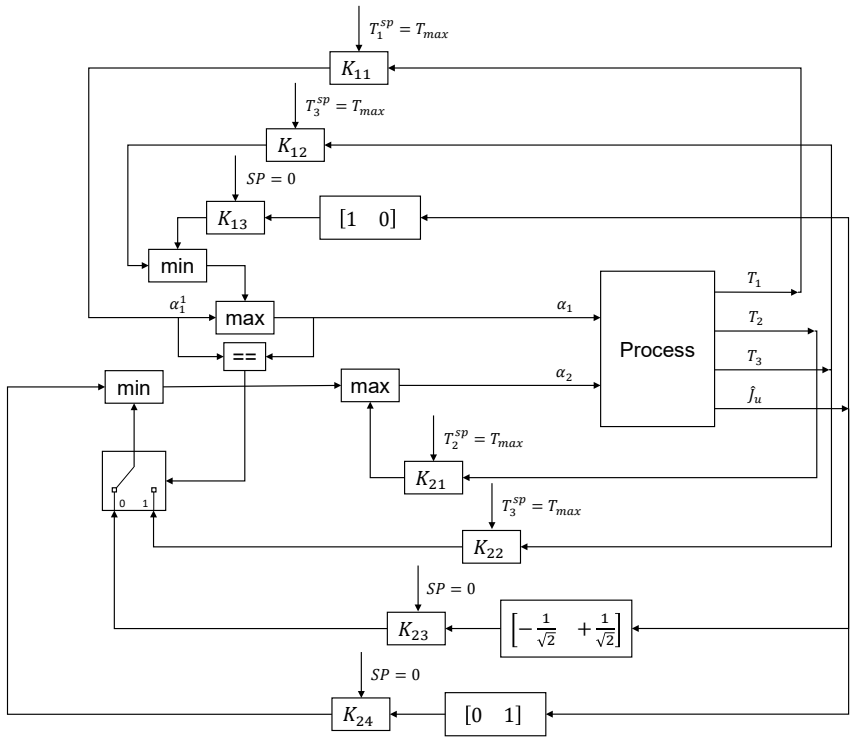


Figure 5.2: Proposed adaptive control structure

α_1	α_2 (T_1 inactive)	α_2 (T_1 active)
T_1	T_2	T_2
$\begin{bmatrix}1 & 0\end{bmatrix} J_u$	$\begin{bmatrix}0 & 1\end{bmatrix} J_u$	$\begin{bmatrix}0 & 1\end{bmatrix} J_u$
T_3	$\begin{bmatrix}-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix} J_u$	T_3

Table 5.2: Proposed adaptive pairing for all operating regions

5.4 Simulation results and discussion

The control structure previously presented is now evaluated in closed-loop simulation face to changing disturbances. Figure 5.3 shows the simulation results, where all 7 possible regions are explored. As the process itself is considered to be at steady state at all times, the dynamics of the system is fully attributed to the tuning of the controllers. Operation in the fully constrained regions is optimal at steady state, whereas there is some deviation from the optimal conditions in the partially constrained and unconstrained regions. This is due to the estimation of gradients by Jäschke temperatures, which does not fully represent the plant model, but gives a reasonable estimate for control, so that low operational loss is achieved.

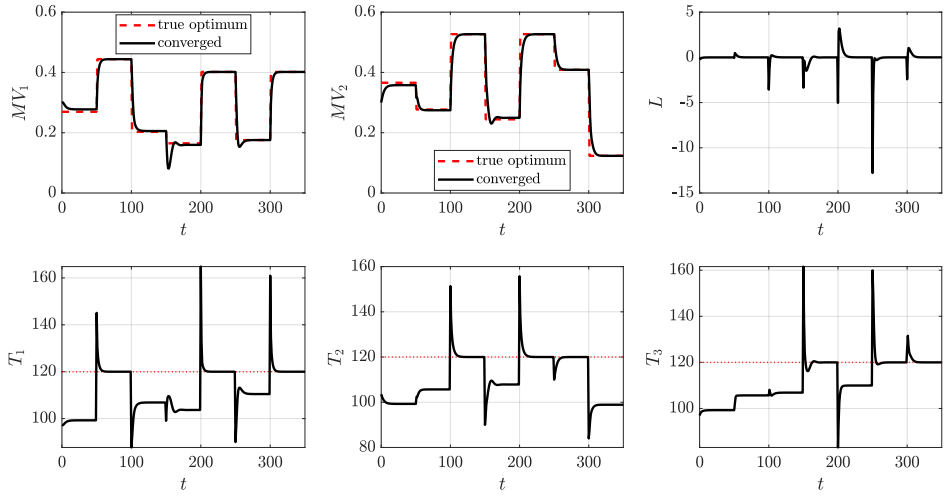


Figure 5.3: Simulation of region-based control structure using Jäschke temperatures

These results are compared with a traditional RTO implementation, see Figure 5.4. This implementation consists of a two-step approach, with disturbance estimation followed by model-based constrained optimization. The system converges in few iterations, with similar steady-state behavior to the region-based control structure. The unconstrained and partially constrained regions suffer from deviations from

the true optima, due to model-plant mismatch, and the converged state is quite similar to that of the region-based control structure. This is to be expected, as Jäschke temperatures represent the gradient information extracted from the model used in the RTO framework.

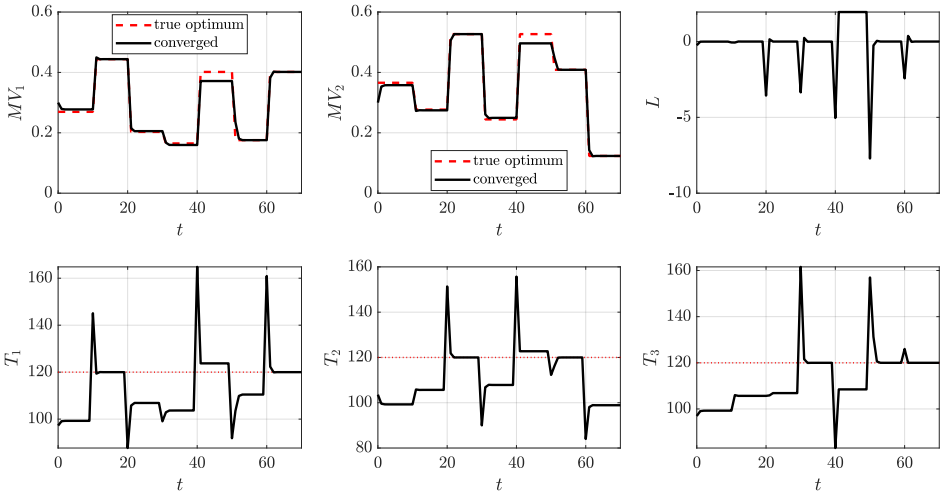


Figure 5.4: Simulation of steady-state RTO with model-plant mismatch

In the RTO simulation, a curious undesired behavior is observed. From $t = 40$, in the fifth simulated region, the system converges to an infeasible point. This happens because the disturbance estimation step returns parameter values that make the optimization problem infeasible, meaning that there are no inputs that satisfy all constraints on the model with the given parameters, even if the estimation step returns parameters that agree with the plant measurements. Some workarounds are therefore deemed necessary for the effective implementation of the RTO strategy, such as the adaptation of the optimization problem itself, based on the estimation of gradients from the true plant (Marchetti et al. 2009).

5.5 Conclusion

In this work, we extended previous work on the optimal operation of heat exchanger networks to the constrained case, where the ideal self-optimizing variables known as Jäschke temperatures cannot be applied to every operating condition. Instead, control of the active constraints becomes necessary for optimal operation, and the challenge lies in deciding automatically what are the best controlled variables during operation. This has been achieved with the use of selectors, with steady-state performance comparable to a traditional model-based RTO implementation. With the proposed control implementation, one avoids the solution

of online optimization problems, which can be problematic, as highlighted by the presented results. However, the simultaneous use of the presented tools is encouraged, so that near-optimal operation is achieved in the faster timescales, and optimization tools can correct for mismatches under more careful evaluation of the results.

Chapter 6

Comparison of simple feedback control structures for constrained optimal operation

This chapter has been published:

> L. F. Bernardino, D. Krishnamoorthy, and S. Skogestad. Comparison of simple feedback control structures for constrained optimal operation. *IFAC-PapersOnLine*, 55(7):883–888, 2022a

6.1 Introduction

Optimal operation is one of the main objectives in process operation, as it is always desired that losses are minimized when possible. Optimal operation requires optimization of economic objectives, i.e. maximization of profit or minimization of costs, subject to constraints related to intrinsic or external conditions, such as operational capacity, product specification, or emission limit values. This is often formulated as a steady-state optimization problem, which can be solved through a plethora of methods (Nocedal and Wright 2006), given that a full model for the system is known. While all constraints are satisfied in the solution of such problems, some constraints influence the location of the solution, but others do not. The former type of constraints is typically referred to as active constraints.

The main challenge related to implementation of real-time optimization (RTO) strategies lies on the lack of knowledge about the system. This can be detected

through the available system measurements, and how far these measurements are from the model predictions gives a metric of how inaccurate the model is. If the model is parametrized by disturbances which cannot be known or measured, online parameter estimation can be used to fit the model to the measurements and thus allowing for better predictions (Roberts and Williams 1981). This translates into a two-step approach for RTO implementation, consisting of parameter estimation and reoptimization, which is very simple, but depends heavily on the structural similarity between model and plant. If this condition is not met, operation might converge to a suboptimal or even infeasible point (Marchetti et al. 2009).

Therefore, the use of model-based RTO approaches has fundamental limitations, as model-plant mismatch is always present in some degree, and may not be completely removed even in the presence of measurements. In this context, an interesting area of research is attempting to satisfy optimality conditions without solving the model-based optimization problem, and at the same time requiring the least amount of knowledge of the system. One particularly useful concept is self-optimizing control (Skogestad 2000), which is based on the translation of the optimization problem into a feedback control problem, and the focus becomes the selection of variables that, when kept controlled under a fixed setpoint, allow for optimal operation. The resulting control structure minimizes the effect of disturbances by design, based on the available model, and any model-plant mismatch is dealt by the upper control layers through setpoint changes. With this, the magnitude of setpoint changes that the upper layers must perform is minimized, guaranteeing that near-optimal operation is attained even in the faster timescales, when the necessary update is not yet available.

This class of strategies, however, has limitations regarding the treatment of constraints, see Gros et al. (2009), François et al. (2005). If the set of active constraints is fixed through operation, a single set of variables can be controlled for optimal operation. In particular, if there are no active constraints, the ideal self-optimizing variables are the gradient of the cost with relation to the inputs (Jäschke and Skogestad 2011), and in presence of active constraints, the ideal self-optimizing variables become the active constraints themselves and the reduced cost gradient projected in the unconstrained directions (Krishnamoorthy and Skogestad 2019). However, if the set of active constraints change during operation, these control objectives no longer apply, and restructuring of the control system is required.

Another setback in the implementation of RTO strategies is related to the computational effort necessary for implementation. As the solution of optimization problems is computationally expensive, RTO is often performed in a slow timescale, and operation must always be performed with the aid of fast controllers that stabilize the process and control key variables for operation. This means that rapid

changes in active constraints may not be counteracted efficiently if changes in active constraints are only dealt by the RTO layer, even if these RTO strategies have the capability of completely eliminating the offset in steady state (Marchetti et al. 2020). There is therefore great interest in the implementation of fast and feedback-based approaches to optimal operation of processes with changes in active constraints.

For this end, a classic approach is analyzing the possible active constraint regions, and designing a control structure that is able to switch between the controlled variables (CVs) (Krishnamoorthy and Skogestad 2019, Reyes-Lúa et al. 2018). A limitation of this approach is the necessity of pairing, which becomes problematic when constraints are independent and may activate at the same time. In such cases, the pairing needs to be adaptive, but proposing adaptive structures may be cumbersome or even infeasible. In this work, such types of case study are explored, and we aim to evaluate the viability of region-based control structures and the use of more general feedback control structures in the optimal operation of these systems. Specifically, we propose the use of a primal-dual feedback optimizing control structure, based on the work presented in Krishnamoorthy (2021), which can be applied to solve most steady-state optimal operation problems of interest. In this structure, the Lagrange multipliers are introduced as extra degrees of freedom that can be used for constraint control, and therefore the resulting approach presents both primal and dual decision variables as manipulated variables, which enables for tracking of all necessary conditions of optimality.

6.2 Control structures for optimal operation

In this section we present the control structures considered in the present work, which aim to solve a steady-state optimization problem through feedback. This generic optimization problem can be defined as:

$$\begin{aligned} \min_{u} \quad & J(u,d) \\ \text{s.t.} \quad & g(u,d) \leq 0 \end{aligned} \tag{6.1}$$

In this definition, $u \in \mathbb{R}^{n_u}$ represents the manipulated variables (MVs), $d \in \mathbb{R}^{n_d}$ represents process disturbances, J: $\mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}$ represents the objective function, and g: $\mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_g}$ represents all process inequality constraints. For this problem, by introducing $\lambda \in \mathbb{R}^{n_g}$ as the Lagrange multipliers associated to the inequality constraints, the Lagrangian function is written as:

$$\mathcal{L}(u,d,\lambda) = J(u,d) + g(u,d)^T \lambda \tag{6.2}$$

The necessary Karush-Kuhn-Tucker (KKT) conditions for the optimization problem state that the optimal pair (u^*, λ^*) satisfies:

$$\nabla_u \mathcal{L}(u^*, d, \lambda^*) = \nabla_u J(u^*, d) + \nabla_u g(u^*, d)^T \lambda^* = 0 \tag{6.3a}$$

$$g(u^*, d) \leq 0 \tag{6.3b}$$

$$\lambda^* \geq 0 \tag{6.3c}$$

$$g(u^*, d)^T \lambda^* = 0 \tag{6.3d}$$

The main challenge in solving this type of problem is related to the lack of knowledge about the set of active constraints g_A, which is here written as the vector composed of the elements of g such that $g_A(u^*, d) = 0$. If this set is known beforehand, the problem is simplified to an equality-constrained optimization problem, which is written as:

$$\nabla_u \mathcal{L}(u^*, d, \lambda^*) = \nabla_u J(u^*, d) + \nabla_u g_A(u^*, d)^T \lambda_A^* = 0 \tag{6.4a}$$

$$g_A(u^*, d) = 0 \tag{6.4b}$$

6.2.1 Active constraint region-based control using selectors

This strategy can be regarded as a generalization of the classic approach to designing an advanced supervisory control layer for optimal operation. In this strategy, each possible set of active constraints define a set of control objectives for optimal operation (Krishnamoorthy and Skogestad 2019). For a given region defined by the active set A, controlling $g_A(u, d) = 0$ is the first straightforward choice of CVs, which fills n_a degrees of freedom. The remaining $n_u - n_a$ unconstrained degrees of freedom are filled with a projection of the cost gradient $N^T \nabla_u J(u, d)$ such that N is the nullspace of $\nabla_u g_A(u, d)$. This leads to optimal operation because it automatically satisfies the KKT condition given in Equation (6.4a), since $N^T \nabla_u J(u, d) = -N^T \nabla_u g_A(u, d)^T \lambda_A = 0$ at the stationary point. Therefore, the control objectives in this strategy are calculated from the plant gradients, and fully independent of the optimal Lagrange multipliers.

A scheme of the general control strategy is presented in Figure 6.1, where the gradient $\nabla_u J$ is considered to be obtained with the aid of the model and measurements. The control structure is designed taking into account the constraints g and all possible gradient projections $N_i \nabla_u J$, each of them being paired to specific plant inputs, and logic must be applied to select between the corresponding control actions u_g and u_0. This logic serves therefore as a detection mechanism

of the active constraints, which can rapidly change during operation. In this work, we attempt to implement the switching logic through the use of selectors, as it is usually done in practice. However, this strategy alone is not effective when it is necessary to switch pairings for different regions. This shall be discussed within the first case study considered in this work.

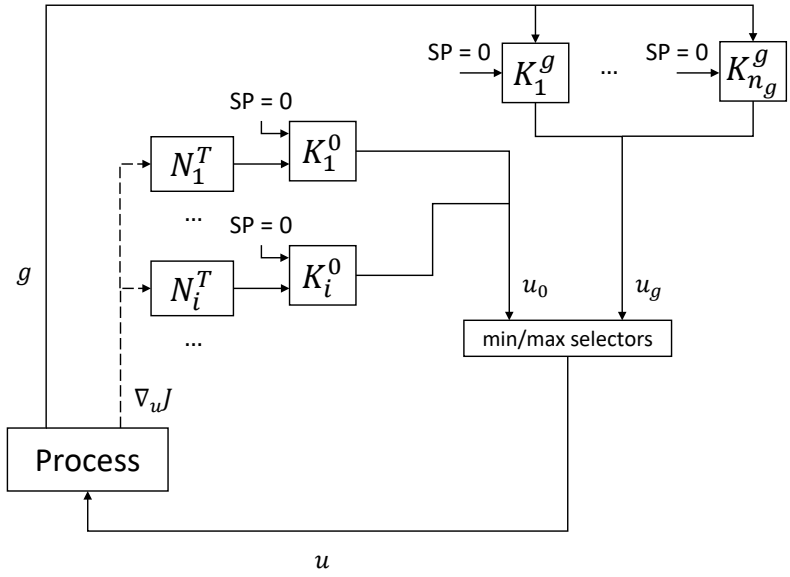


Figure 6.1: Region-based control strategy using selectors

6.2.2 Primal-dual feedback optimizing control

In this strategy, a single control structure is used for all regions, with controllers arranged in a cascade layout, according to Figure 6.2. The components of the Lagrangian gradient $\nabla_u \mathcal{L}$ are paired to the respective process inputs u with simple controllers, and each constraint g is controlled in an outer loop by manipulating the estimate of the respective Lagrange multipliers λ, entailing the use of $n_u + n_g$ controllers, labelled as K^u and K^λ in Figure 6.2. The constraint controllers must become inactive when the constraints are not violated ($g(u, d) < 0$), and a switching logic to enforce $\lambda \geq 0$ is thus introduced, guaranteeing steady-state primal feasibility (6.3b), dual feasibility (6.3c), and complementarity conditions (6.3d). This control structure is based on the works of Krishnamoorthy (2021), Dirza et al. (2021), which were written under a distributed optimization perspective, but it can also be applied to a generic constrained optimal operation problem (Krishnamoorthy and Skogestad 2022).

This structure gives up on tight control of constraints in fast timescales, as their

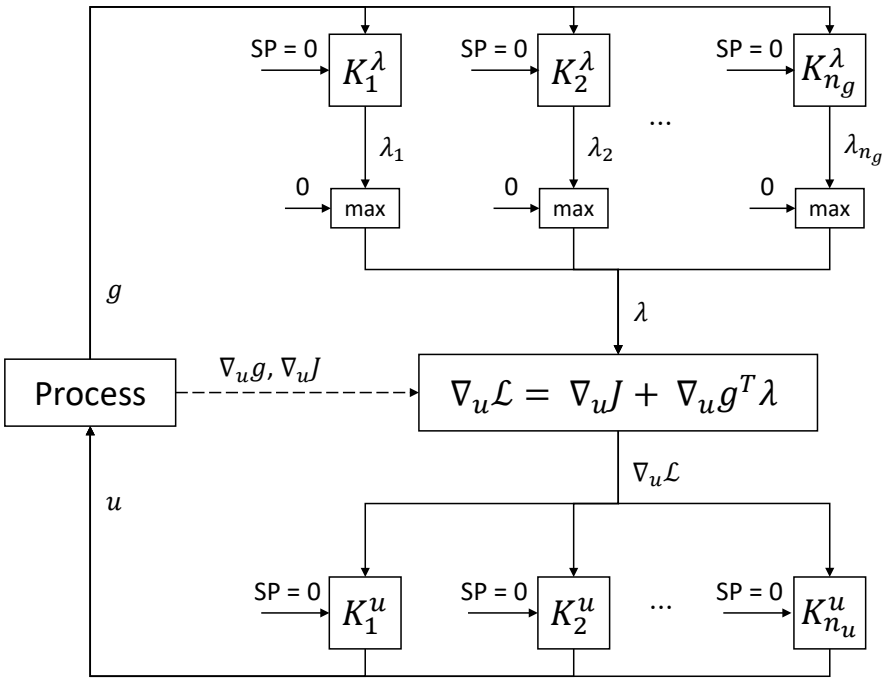


Figure 6.2: Primal-dual feedback optimizing control framework, based on the DFRTO framework (Krishnamoorthy 2021).

control is placed in an internal loop mediated by the Lagrange multiplier estimates. However, given that the gradient calculations are accurate, and that integral action is present in all controllers, optimality is attained at steady state. The switching logic between active constraints is mediated by the max blocks, as the calculated λ shifts the relevant directions for control in the actuator layer when the constraints are violated, and zero is selected when the calculated λ value becomes negative, which happens when the constraint is no longer being violated. One advantage of this strategy is that pairing between constraints and MVs is not required, since this association is done through the Lagrangian gradient calculation.

6.3 Case study 1: heat exchanger network

The first system considered in this work, based on the work of Jäschke and Skogestad (2014), consists of three heat exchangers in parallel. The network is fed with a cold stream, which is split to be heated into each line by different hot streams. The goal of the process is therefore to maximize the final temperature of the heated stream, T, but subject to constraints of maximum allowed temperature in each branch, $T_i \leq T_{max}$. For this system, the available manipulated variables are the splits for each line, $u = \alpha$, and the possible disturbances are $d = [T_0, w_0, T_{h,i}, w_{h,i}, UA_i]$, namely the cold stream inlet temperature and flow, hot streams inlet temperatures and flows, and heat transfer coefficients for the exchangers. A representation of the system is shown in Figure 6.3, and the respective optimization problem is written as:

$$\begin{aligned} \min_{u} \quad & J = -T \\ \text{s.t.} \quad & g_i = T_i - T_{max} \leq 0, \quad i = 1, 2, 3 \end{aligned} \tag{6.5}$$

6.3.1 Active constraint region-based control

The case study has a total of $2^3 = 8$ possible regions, but only 7 are feasible regions, as it is not possible to satisfy all 3 constraints with 2 manipulated variables. In the case that all constraints are violated, it is acceptable that one constraint is given up for the operation, but this is beyond the scope of this analysis. For each region, the control objectives that allow for optimal operation are given in Table 6.1. In this case study, due to the nature of the constraints, the derivation of linear combinations of the gradient, N^T, for the different regions results in constant coefficients inside the region.

Upon inspection of the control objectives, it is clear that a single pairing strategy cannot account for all regions, especially due to the regions with 2 active constraints. When g_1 is paired to u_1 and g_2 is paired to u_2, which is a natural pairing

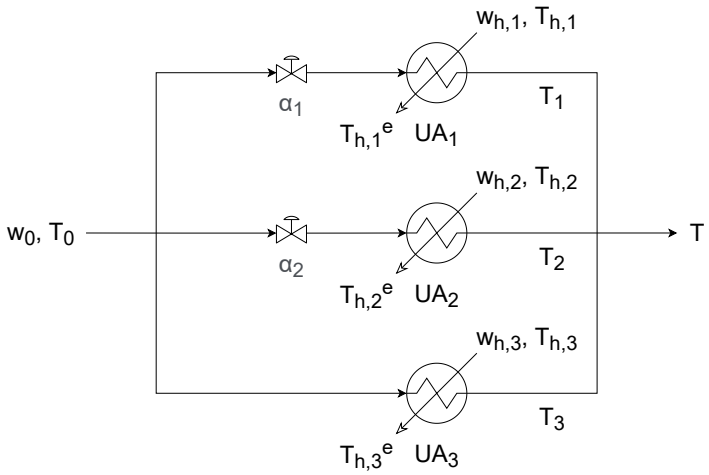


Figure 6.3: Heat exchanger network scheme.

Table 6.1: Control objectives per region for case study 1

Active constraints	Control objectives
-	$\nabla_u J$
g_1	g_1, $[0 \quad 1]\nabla_u J$
g_2	g_2, $[1 \quad 0]\nabla_u J$
g_3	g_3, $[-1 \quad 1]\nabla_u J$
g_1, g_2	g_1, g_2
g_2, g_3	g_2, g_3
g_1, g_3	g_1, g_3

choice, g_3 cannot be attributed to a single MV if control over all regions is desired. If g_3 is paired to a single MV in this case, a selector strategy can account for 6 regions at most. One pairing example is given at Table 6.2. In this case, following a classic approach of pairing and implementing a switching logic, the region where g_1 and g_3 are simultaneously active cannot be optimally controlled.

Table 6.2: Example of classic pairing for region-based control of case study 1

u_1		u_2	
	g_1		g_2
$[1$	$0]\nabla_u J$	$[0$	$1]\nabla_u J$
	g_3	$[-1$	$1]\nabla_u J$

Figure 6.4 shows the performance of this control structure over a disturbance sequence that activates all possible operation modes. In spite of it being able to handle most regions correctly, there is steady-state constraint violation for g_3 in the region that cannot be handled, from $t = 200$ to $t = 300$.

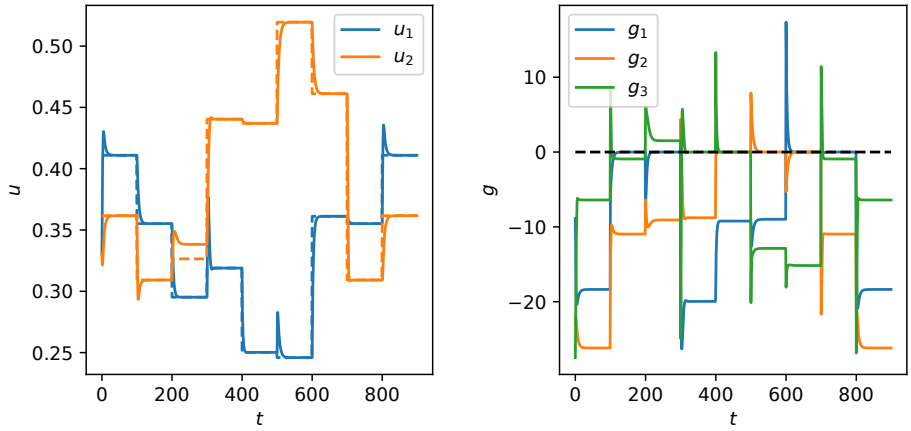


Figure 6.4: Operation of case study 1 with region-based control using classic pairing, along with optimal inputs (dashed).

From these results, it becomes clear that a control structure that handles all regions must be more flexible. Specifically, g_3 must also be controlled using u_2 in some cases. This introduces external conditions on the controllers activation, since this possibility should only be accessed when g_1 is being controlled by u_1. Implementing this logical statement should guarantee optimal operation over all possible regions. This adaptive pairing is presented in Table 6.3.

Table 6.3: Adaptive pairing for region-based control of case study 1

u_1	u_2 (g_1 inactive)	u_2 (g_1 active)
g_1	g_2	g_2
$[1 \quad 0]\nabla_u J$	$[0 \quad 1]\nabla_u J$	$[0 \quad 1]\nabla_u J$
g_3	$[-1 \quad 1]\nabla_u J$	g_3

The performance of the adaptive region-based control structure over the same disturbance realization is presented in Figure 6.5. In this case, no steady-state constraint violation is obtained, and the expected peaks that happen when disturbances change are quickly corrected. At $t = 200$, quick oscillations in u_2 can be noticed, due to the changes in the active control structure.

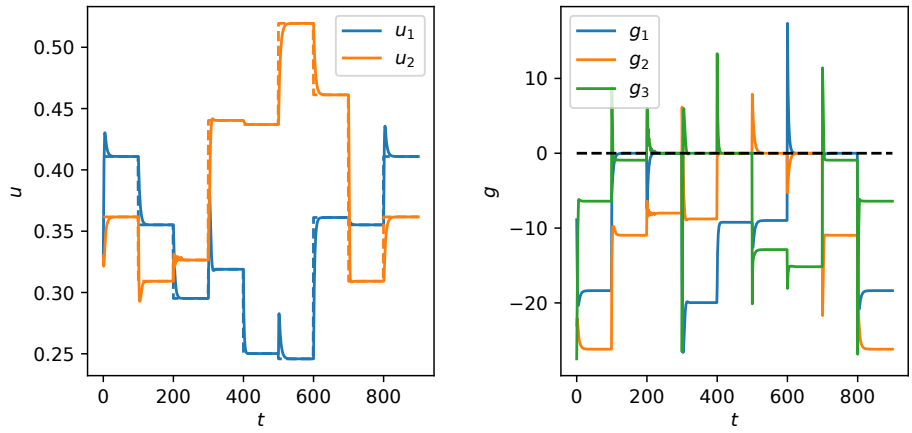


Figure 6.5: Operation of case study 1 with region-based control using adaptive pairing, along with optimal inputs (dashed).

6.3.2 Primal-dual feedback optimizing control

The performance of the primal-dual feedback optimizing control structure is presented in Figure 6.6. The steady-state performance of this structure is the same when compared to what is attained by selectors, and constraint violation is efficiently corrected even though it is regulated by an extra control layer.

In addition, Figure 6.7 shows the dual variables of the primal-dual control structure. The combination of layers leads to slower responses in some disturbance changes, especially when there are big changes in the values of Lagrange multipliers. Nevertheless, the estimated Lagrange multipliers smoothly converge to the optimal values in each region, and spikes due to instantaneous constraint violation

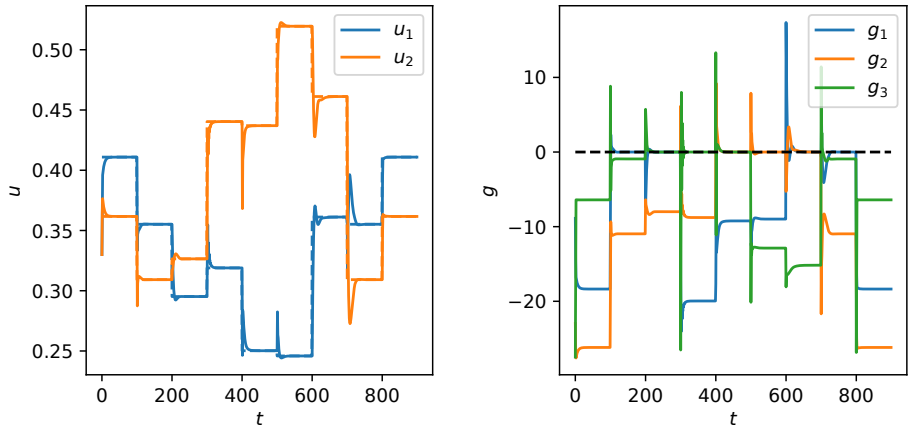


Figure 6.6: Test of primal-dual control framework over case study 1, along with optimal inputs (dashed).

are corrected at steady state.

6.4 Case study 2: two distillation columns in sequence

Another system studied in this work has been described by Jacobsen and Skogestad (2012), and it consists of two distillation columns in series, as presented in Figure 6.8. The inlet stream, composed of three components A, B, and C, is fed into the first column, with the goal of separating the most volatile component with minimal purity specification x_A. The bottom product is then fed into the second column, which generates distillate and bottom products with minimal purity specifications x_B and x_C, respectively. In addition, there are constraints related to maximum boilup of the columns, V_1 and V_2. The operational goal is to optimize plant economics, with costs related to feed and vapor consumption, and profit from selling the products. The steady-state optimization problem can then be written as:

$$\begin{aligned}
\min_{u} \quad & J = p_F F + p_V(V_1 + V_2) - p_A D_1 - p_B D_2 - p_C B_2 \\
\text{s.t.} \quad & g_1 = x_{A,min} - x_A \leq 0 \\
& g_2 = x_{B,min} - x_B \leq 0 \\
& g_3 = x_{C,min} - x_C \leq 0 \\
& g_4 = V_1 - V_{1,max} \leq 0 \\
& g_5 = V_2 - V_{2,max} \leq 0
\end{aligned} \tag{6.6}$$

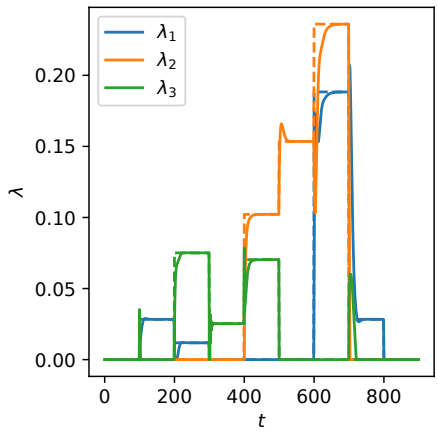


Figure 6.7: Lagrange multiplier estimates from the primal-dual control framework in case study 1, along with optimal multiplier values (dashed).

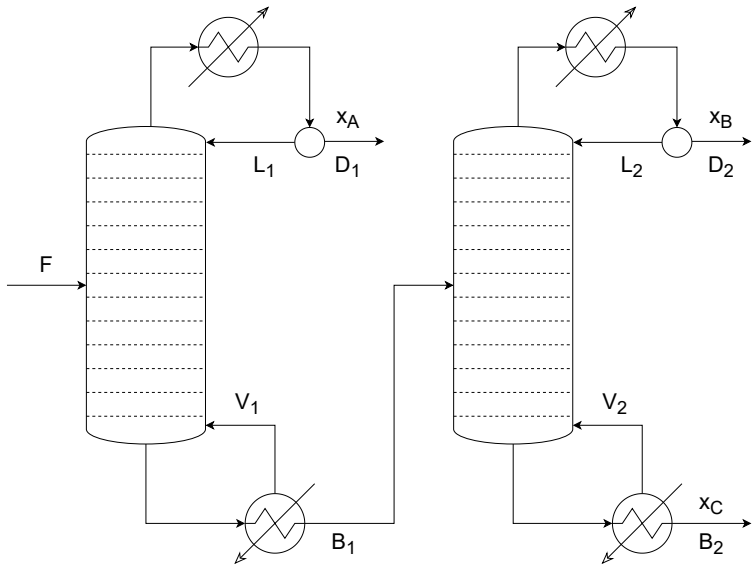


Figure 6.8: Scheme of two distillation columns in sequence, based on Jacobsen and Skogestad (2012)

The vector of manipulated variables for operation are the internal flows of the columns $u = [L_1, L_2, V_1, V_2]$, and the possible disturbances are $d = [F, p_V]$, namely the feed to the first column and the steam generation price. The system is considered to be at steady state at all times, meaning that all changes in the operating conditions are quickly accommodated by the system. This assumption imposes the limitation that this control structure operates in a slower timescale than that of the process, which is a common practice when dealing with supervisory control.

In the considered operating range, there are 8 possible active constraint regions. These regions are described in Table 6.4. Unlike case study 1, calculating gradient projections for each region becomes more complicated, as the system constraints have a nonlinear relationship with the inputs. Therefore, even though constraint control can be achieved by using a selector-based logic, control of the unconstrained degrees of freedom requires an elaborate and interconnected logic, and is therefore deemed outside of the scope of this paper.

Table 6.4: Active constraints per region for case study 2

Region number	Active constraints
I	x_B
II	x_B, x_A
III	x_B, V_1
IV	x_B, x_A, x_C
V	x_B, x_A, V_1
VI	x_B, V_1, V_2
VII	x_B, x_A, V_1, V_2
VIII	x_B, x_A, x_C, V_1

The performance of primal-dual optimizing control in this system is presented in Figure 6.9, and the corresponding Lagrange multiplier estimates are shown in Figure 6.10. As in the previous case study, the optimal values for the plant inputs and estimated Lagrange multipliers are attained at steady state, and any constraint violation is corrected.

6.5 Conclusion

The presented case studies illustrate the complexity of optimal operation problems when there is switching of active constraint regions. We explore the concept of ideal controlled variables for optimal operation, and the close link between steady-state plant gradients and optimal operation. It is assumed that these gradients are available for the control structure, which can be burdensome especially in cases

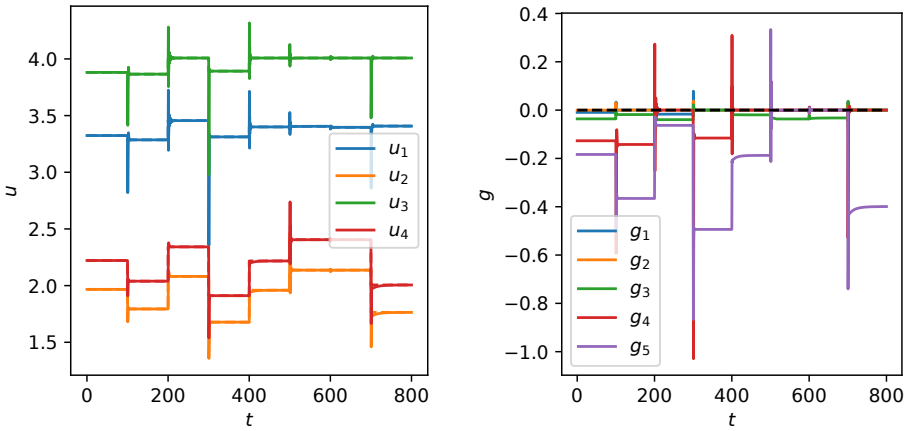


Figure 6.9: Test of primal-dual control framework over case study 2, along with optimal inputs (dashed).

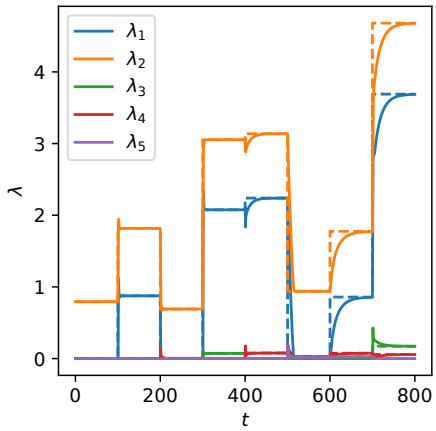


Figure 6.10: Lagrange multiplier estimates from the primal-dual control framework in case study 2, along with optimal multiplier values (dashed).

where these depend on the plant states. However, as this layer is meant to operate in the slower timescales, it can be assumed that the system is settled, and gradient estimation is possible. This can be done by using a model, with further correction by the plant data. In this case, the use of an incorrect model is not detrimental, as the inclusion of biases in the gradients do not impose hard constraints on the control problem. The performance of this control structure may be improved even further by making use of optimization results, as one can retrieve estimated gradients and Lagrange multipliers from these calculations.

The implementation of simple feedback structures that deal with constraints and allow for optimal operation was accomplished. The choice of which structure is favored lies in the nature of the system at hand. If there is a low number of constraints, and pairing can be done to contemplate all regions, a simple structure consisting of selectors is to be considered, as in this case constraints are kept more effectively under control. However, no big loss was noticed with the implementation of indirect constraint control mediated by the control of the KKT conditions. The latter is also not affected by the combinatorial nature of the number of active constraint regions, and by the concern of adequate pairing between variables. The use of primal-dual feedback optimizing control is therefore deemed promising in the handling of constrained systems, and further evaluation of this strategy is encouraged, especially in terms of controller tuning and accounting for process dynamics. The further study of control structures that deal with active constraint switching in fast timescales remains relevant, bearing in mind that the studied primal-dual framework requires a timescale separation between the primal and dual control layers.

Chapter 7

Bidirectional inventory control with optimal use of intermediate storage and minimum flow constraints

This chapter has been published:

> L. F. Bernardino and S. Skogestad. Bidirectional inventory control with optimal use of intermediate storage and minimum flow constraints. *IFAC-PapersOnLine*, 56(2):2665–2670, 2023a

7.1 Introduction

When operating a chemical plant, the periodic shutdown of units becomes necessary, either due to planned maintenance, or due to failures that must be corrected. To decouple the effect of these temporary shutdowns, buffer tanks are employed, such that production is continued using the accumulated inventory, or accumulated until unit reactivation, without compromising the overall processing rate. The same principle is applied to maintaining a constant production rate in periodic batch/semi-continuous processes (Karimi and Reklaitis 1983). Managing the inventory of these units during operation becomes then paramount for maximizing the processing rate of the plant during these events. The problem of inventory management face to disturbances is also relevant for supply chain management (Schwartz and Rivera 2010).

The installation of buffer tanks is especially important around critical units that operate normally at full capacity, since unnecessary shutdown of these units leads to irrecoverable losses that could otherwise be avoided with proper planning. The unit that limits the overall processing capacity of a plant is known as the process bottleneck. In terms of maximizing production, a good idea is therefore to set the production rate of the plant close to this bottleneck (Downs and Skogestad 2011). The valve that sets the overall production rate of a plant is known as the throughput manipulator (TPM), and the control of the inventories must be defined as a function of this TPM. For a consistent inventory control layer, the input-output pairs should radiate from the chosen TPM, being in the direction of the flow for downstream units, and opposite to the flow direction for upstream units (Price et al. 1994). This rule is sufficient for processes with units arranged in series, which are the focus of this work.

The management of buffer levels when shutdowns occur is often performed by the plant operators, which switch the affected inventory controllers to manual mode, until normal operation is restored. If the planned stop is long, this may be accompanied with some accumulation prior to the unit shutdown, such that production may be continued without problems. This strategy, although often optimal, needs human intervention, and therefore an automatic control framework that deals with this issue is desired. On the other hand, the control strategies often employed in the literature rely on dynamic models and optimization (Chong and Swartz 2013, Boucheikhchoukh et al. 2022), which is costly to implement, and does not reflect the simplicity of the policy that is implemented by experienced operators. The bidirectional control structure, presented by Shinskey (1981) and further discussed in Zotică et al. (2022), is able to solve this problem in a simple automatic control framework, comprised of PI controllers and selectors.

The main idea of the bidirectional control structure, shown in black in Figure 7.1, is to use the inventory of each unit for maximizing the time in which the process can run with maximum throughput. At steady state, the control structure treats the process bottleneck as the TPM, as it is saturated and therefore cannot be used for inventory control. As a consequence, due to the reconfiguring logic of the control loops, all downstream inventories will operate at the minimum level $M_i = L$ controlled by the respective outlets z_i, and all upstream inventories will operate at the maximum level $M_i = H$ controlled by the respective inlets z_{i-1}. If a new bottleneck is introduced anywhere before the current TPM, reducing the flow at that point, the downstream unit inventory is depleted until the minimum, becoming then controlled by its outlet, generating a cascade effect that ends with the previous TPM being used for inventory control of the unit before it. This results in an effective change in the TPM position in the process, since the introduced

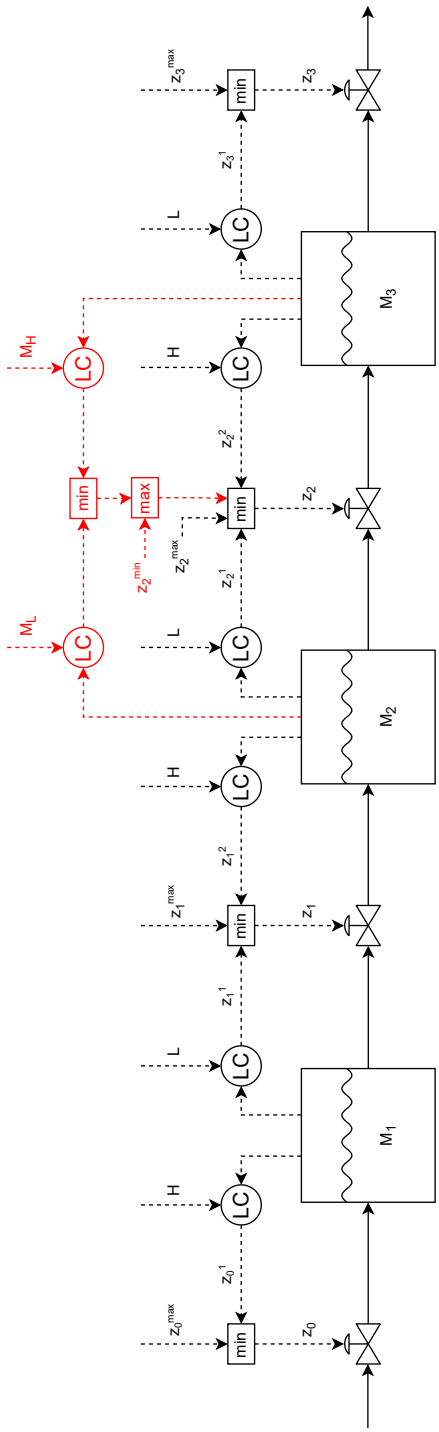


Figure 7.1: Proposed bidirectional inventory control structure with minimum flow constraint handling (black denotes original bidirectional structure, red denotes the addition proposed by this work)

bottleneck sets the production rate. A similar analysis can be made for new bottlenecks after the original TPM. The variation in the inventories affected by this chain of events serves as a buffer time in which the system can operate with the same overall production, and if the introduced bottleneck is active only during a small period, the system can revert to its original behavior without affecting production.

Although the solution given by bidirectional level control is valid for processes with varying bottlenecks, care must be taken when implementing this control structure in processes where a minimum flow must be guaranteed in certain sections of the process, a limitation that naturally appears for some types of equipment. In these cases, if the inventory before the constrained section is critically low, or if the inventory after it is critically high, there is no margin for satisfying this constraint dynamically. A reasonable strategy in these cases would be then to use an intermediary value for the inventory of the neighboring units, so that there is enough dynamic margin for satisfying these constraints, as well as temporary bottlenecks. Together with this, additional control logic must be implemented, in order to automatically reconfigure the control structure when these minimum flow constraints become relevant. The additional control logic for dealing with minimum flow constraints is the novelty presented in this paper.

Mathematically, the goal can be defined as the maximization of the overall production $\bar{F}$ over a sufficiently long time horizon T of a series of N buffer inventories subject to constraints in the manipulated variables, which are the valve positions z_i, $i = 0, \ldots, N$, and constraints in the inventory levels M_i, $i = 1, \ldots, N$, according to:

$$\begin{aligned}
\max_{z(t)} \bar{F} &= \frac{1}{T}\frac{1}{N+1}\sum_{i=0}^{N}\int_0^T F_i(t)\,dt \\
\text{s.t. } & M_i^{min} \leq M_i(t) \leq M_i^{max}, && i = 1, \ldots, N \\
& z_i^{min} \leq z_i(t) \leq z_i^{max}, && i = 0, \ldots, N \\
& \frac{dM_i}{dt} = \frac{1}{V_i^t}\left(F_{i-1} - F_i\right), && i = 1, \ldots, N \\
& F_i = C_i z_i, && i = 0, \ldots, N
\end{aligned} \tag{7.1}$$

Here, V_i^t represents the total capacity of the i-th inventory, and C_i represents the flow coefficient of the i-th valve. While bidirectional level control can be used to solve this problem when only the constraints $z_i \leq z_i^{max}$ and $M_i^{min} \leq M_i \leq M_i^{max}$ are relevant, through the previously described logic, this work considers the case where the constraints $z_i \geq z_i^{min}$ may also be activated during operation.

To solve this problem, we propose an extension to the bidirectional inventory control framework, shown in red in Figure 7.1. In the proposal, we account for the minimum flow constraints by using controllers with intermediary setpoints and additional selectors. This portion of the control logic will be active as long as it is feasible to satisfy the minimum flow constraints. The control framework will be now presented and exemplified in a case study.

7.2 Proposed control structure

In this work, we consider a system of three tanks in series, described in Zotică et al. (2022). The physical parameters necessary to simulate the system are reproduced in Table 7.1. For simplicity, we consider that only z_2 is subject to a minimum flow constraint, but the approach can naturally be extended to include minimum flow constraints on other sections of the process. For designing the inventory control layer, the desired closed-loop time constant is chosen as $\tau_c = 0.5$ min, and is used as the tuning parameter for tuning the PI controllers following the SIMC rules (Skogestad 2003). All controllers are implemented with antiwindup action, using the backcalculation strategy (Åström and Rundqwist 1989) with tracking time constant $\tau_T = \tau_I/4$, where τ_I is the controller integral time.

i	C_i [m^3/min]	V_i^t [m^3]
0	1	-
1	1.25	2.3
2	1.428	4.2
3	1.667	6.4

Table 7.1: Physical parameters for the three tank system

The proposed control structure is presented in Figure 7.1. In this structure, normal inventory control done by z_2 aims for intermediary setpoints M_L and M_H. This normal mode may be overriden by the minimum flow constraint $z_2 \geq z_2^{min}$, and in this case inventory control is given up until normal operation is reestablished, or until the inventory control that was given up reaches the associated critical value, L or H. In this moment, inventory control must be resumed, to avoid complete inventory depletion or overflow.

In terms of satisfying the minimum flow constraint on z_2, it is desired that a high value for M_L and a low value for M_H are selected, as this allows for continued transfer from M_2 to M_3 when temporary bottlenecks appear. However, this conflicts with the usage of the buffer inventories for maximizing production, as the

solution provided by the bidirectional inventory control dictates. This shall be evidenced by the simulations presented next.

7.3 Simulation results

We present simulation results for two control structures:

- Proposed bidirectional structure with minimum flow constrain on F_2 (Figure 7.1).
- Simple bidirectional structure (without the red parts in Figure 7.1).

The simulations represent some common disturbance scenarios, allowing us to highlight when the current proposal succeeds or fails. Under all simulations, a minimum flow at z_2, $z_2^{min} = 60\%$, is desired, and the high and low level setpoints are chosen as $H = 90\%$ and $L = 10\%$. In addition, as a compromise for maximizing immediate production and satisfying minimum flow constraints, the intermediate setpoints are initially chosen as $M_H = M_L = 50\%$. This choice will be further analyzed.

First, we show that the structures have similar behavior when the disturbances do not affect the minimum flow constraint, and when these disturbances are only present for a short period. In Figure 7.2 we present a simulation where the steady-state bottleneck is at the process outlet with $z_3 = 100\%$, and therefore all nominal inventories are at a high state. At $t = 10$ min, a temporary bottleneck is introduced at the process inlet, with $z_0 = 50\%$, and due to that, all levels are depleted, in order to keep production at the maximum. When the temporary bottleneck is removed at $t = 40$ min, all tanks are filled back, and operation is back to steady state, without affecting the production at the steady-state bottleneck. When the production rate is changed at the TPM at $t = 80$ min and $t = 100$ min, both control structures quickly respond accordingly, due to all inventory controllers being active. Instead, if the temporary bottleneck on z_0 is instead removed at $t = 90$ min, see Figure 7.3, both control structures must reduce the flow at the original bottleneck. It can be seen that the proposed control structure must reduce the flow at z_3 earlier than the simple bidirectional structure, which is expected since there is less inventory to be used for rejecting the disturbance, as $M_3 = M_H$ initially. Additionally, both approaches tend to satisfy steady-state mass balances, working as consistent inventory control structures.

We now consider cases where the minimum flow constraint may be violated, and the differences between the control structures are highlighted. In the simulation presented in Figure 7.4, the steady-state bottleneck is still at the process outlet,

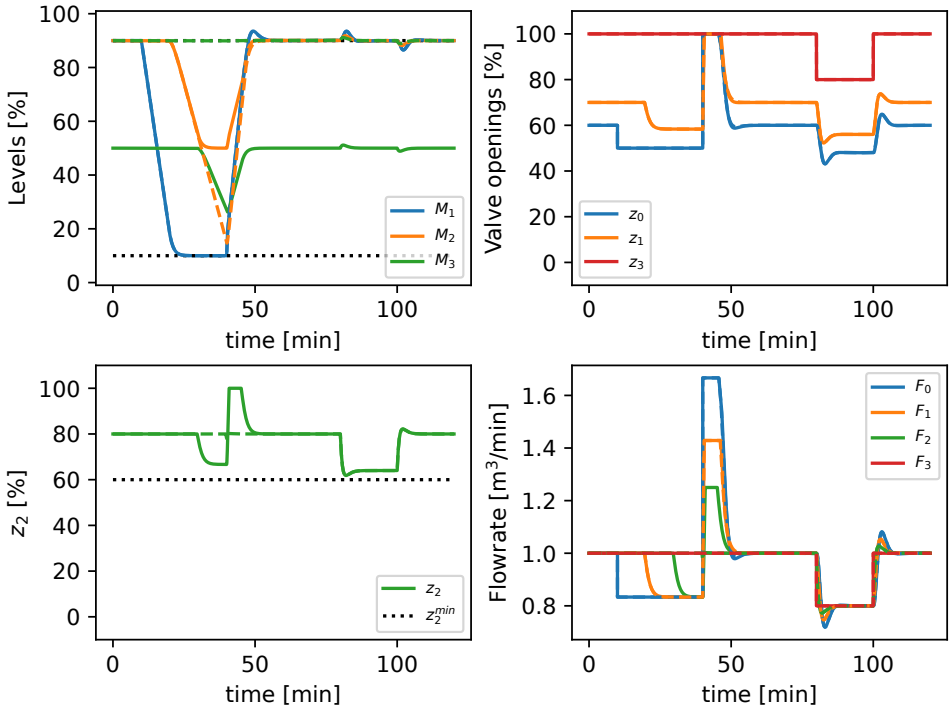


Figure 7.2: Both control structures are able to maximize production at the bottleneck under temporary disturbances (continuous lines represent the proposed structure, dashed lines represent simple bidirectional control) — simulation with TPM at z_3 with short flow reductions at z_0 and z_3

but greater temporary bottlenecks are introduced. As the first disturbance, z_1 is lowered to 40% at $t = 7$ min, which forces M_2 to be emptied. While the inventory is uncontrolled until $M_2 = L$ for the simple bidirectional structure, the proposed framework activates the minimum flow constraint when M_2 reaches M_L, since it cannot keep normal inventory control at $M_2 = M_L$ without violating z_2^{min}. That behavior can be kept until $M_2 = L$, but as the temporary bottleneck on z_1 is removed before the inventory reaches its critical value (at $t = 20$ min), feasibility is maintained. This behavior comes at the expense of slightly affecting the steady-state bottleneck, as M_3, which started from M_H, was depleted during the event. Afterwards, at $t = 40$ min, the original TPM (z_3) is further constrained to 50%, forcing all the flows in the bidirectional control structure to drop almost immediately to attain inventory control, which leads to infeasible operation. The proposed control framework is able to use the margin between $M_3 = M_H$ and $M_3 = H$ to satisfy the minimum flow constraint, and after M_3 reaches the upper limit, operation becomes infeasible until the bottleneck is removed at $t = 60$ min.

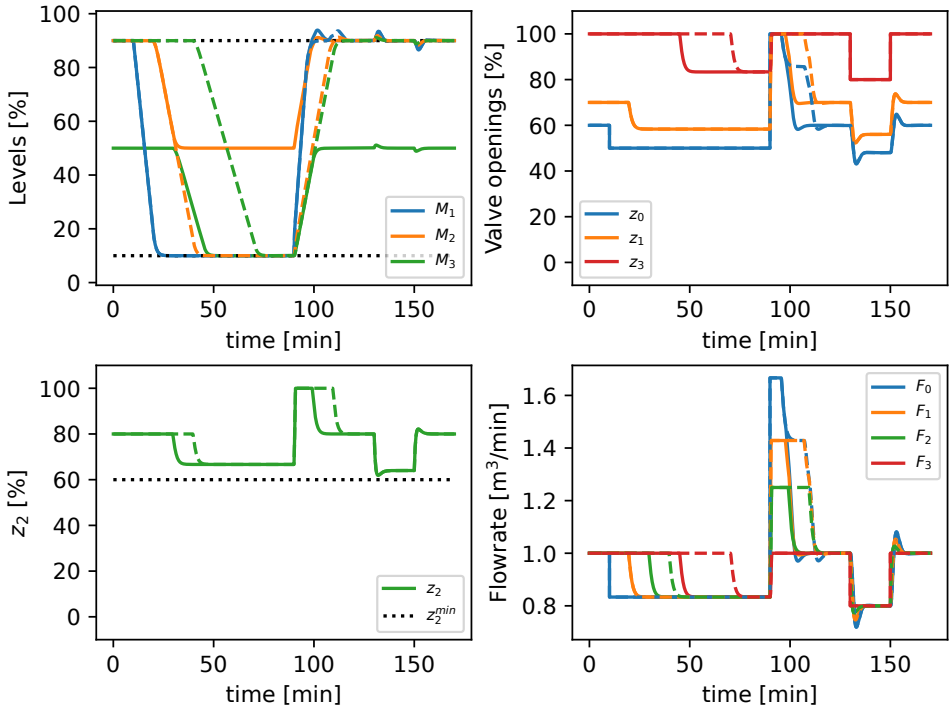


Figure 7.3: Long disturbances force reduction on production at steady-state bottleneck, with the proposed structure (continuous lines) being affected before simple bidirectional control (dashed) — simulation with TPM at z_3 with flow reductions at z_0 and z_3

After that, M_3 must be emptied out until M_H while satisfying $z_2 \geq z_2^{min}$ in the proposed framework, while all the flows go immediately up for the simple bidirectional scheme. In terms of mass balances, the proposed framework is forced to operate in imbalance for the longest possible time, due to the minimum flow constraint.

For the first disturbance in Figure 7.4, it is interesting to note that there is an inversion of behavior during operation. In a very short timescale, while M_2 is between H and M_L, the control structures behave equally. If the disturbance continues to be active, the original bidirectional control becomes best performing, as it still maximizes the flow through z_2, while the proposed control structure is more conservative. In the longer run, however, the original bidirectional structure loses feasibility first, and in that case the proposed control structure performs best.

Figure 7.5 illustrates the case where the bottleneck is at the process inlet, with $z_0 = 50\%$, and therefore all inventories are initially at the lower state. The introduced

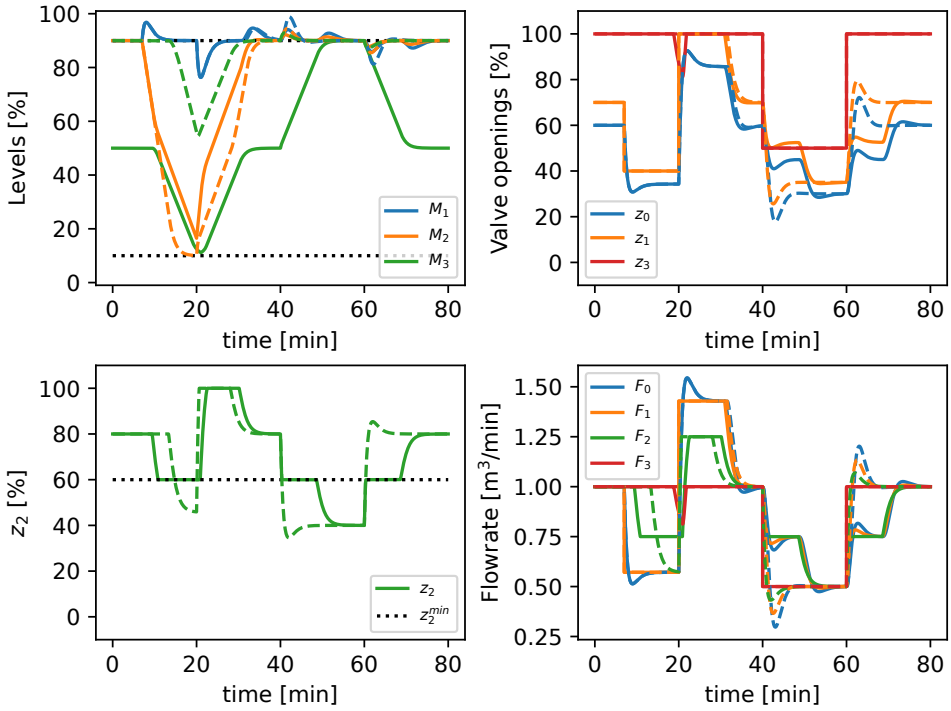


Figure 7.4: The proposed structure (continuous line) allows for feasible operation during longer periods than simple bidirectional control (dashed) — simulation with TPM at z_3 with larger flow reductions at z_1 and z_3

disturbances in the simulation are $z_0 = 40\%$, from $t = 5$ min to $t = 20$ min, and $z_3 = 60\%$, from $t = 45$ min to $t = 70$ min. Analogously to the previous simulation, the reduction on z_0, which was the original TPM, immediately makes traditional bidirectional control infeasible, whereas the proposed framework uses the buffer from $M_2 = M_L$ until $M_2 = L$ to keep feasibility. The temporary bottleneck on z_3 leaves M_3 uncontrolled in the simple bidirectional control until it reaches the limit H, whereas the minimum flow constraint becomes active in the proposed framework when M_3 passes through M_H. The proposed control structure is able to keep feasibility for both disturbances, while simple bidirectional control violates the constraint on both cases.

It must also be noted that changes in the intermediary setpoint values may improve response face to some disturbances. Figure 7.6 illustrates the effect of raising all intermediary inventory setpoints when the process bottleneck is originally at its outflow, z_3. In this case, the disturbances are $z_1 = 40\%$ from $t = 5$ min to $t = 20$ min, and $z_3 = 65\%$ from $t = 50$ min $t = 65$ min. A larger gap between M_L

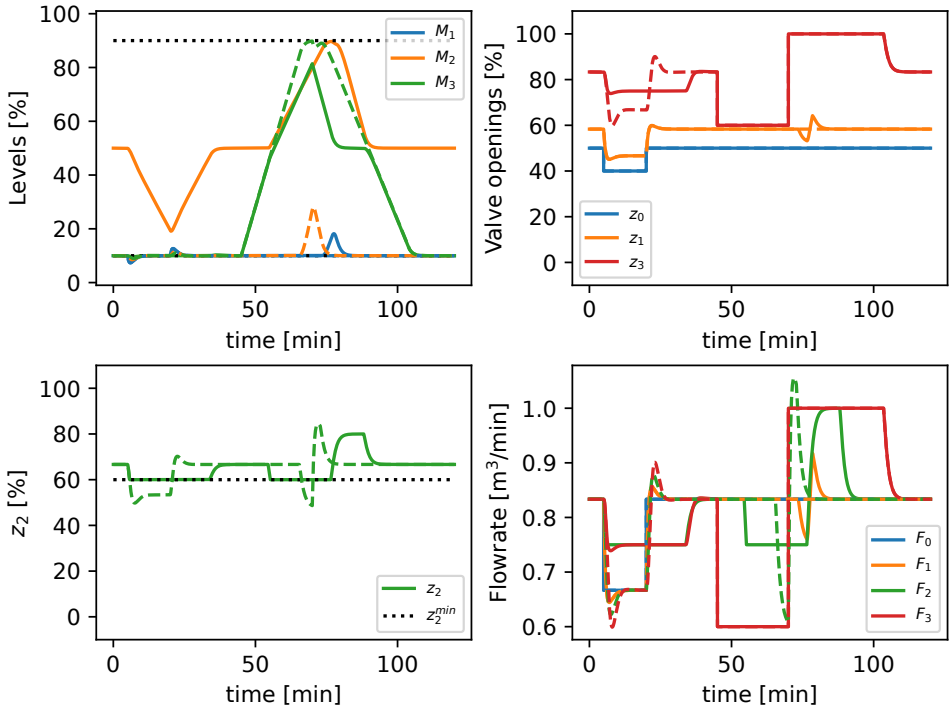


Figure 7.5: The proposed structure (continuous line) completely avoids violating the minimum flow constraint, as opposed to simple bidirectional control (dashed) — simulation with TPM at z_0 with flow reductions at z_0 and z_3

and L allows for improving operation when bottlenecks appear before z_2, in the sense that the use of a higher intermediate setpoint lets the system operate with feasibility for longer. In addition, the higher M_H is, the slower the inventory M_3 is consumed, which allows for keeping z_3 unaltered for longer. Conversely, when z_3 is forced to be lowered, the gap between M_H and H dictates how long the system can be kept feasible, and a low M_H would be desired.

Figure 7.7 illustrates the case where feasibility is prioritized in operation, with high value of M_L and low value of M_H. The original bottleneck is at $z_0 = 50\%$, and the tested disturbances are $z_0 = 40\%$ from $t = 5$ min to $t = 35$ min, and $z_3 = 65\%$ from $t = 75$ min to $t = 120$ min. While the use of normal intermediary setpoints fail to keep the process feasible face to these disturbances, the adjust of setpoints allow for that end, at the expense of reducing the flow at the steady-state bottleneck.

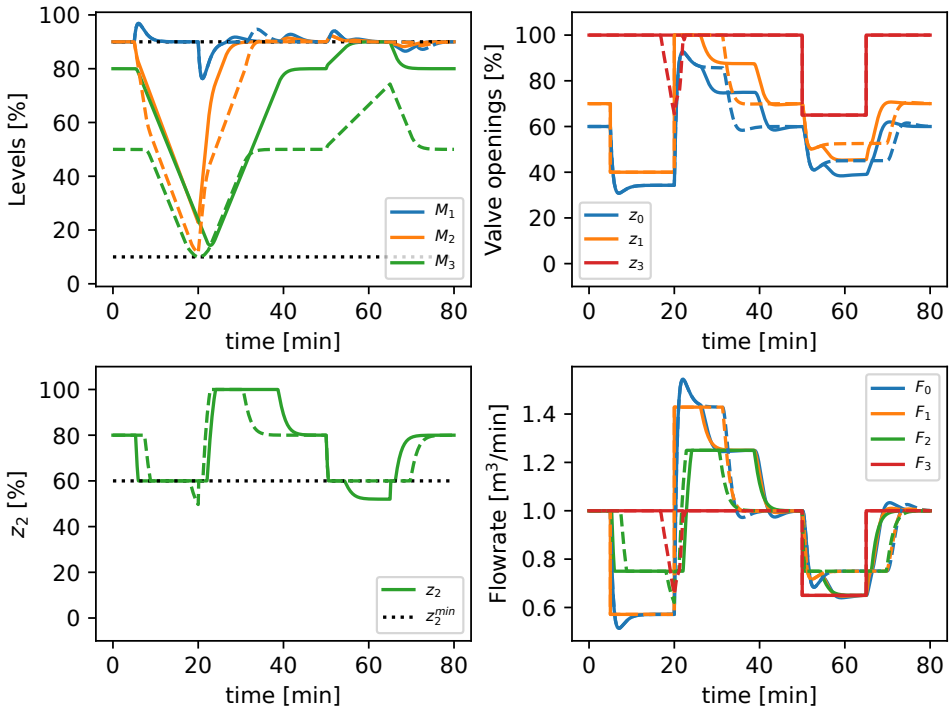


Figure 7.6: With TPM at z_3, higher intermediary setpoints (continuous line, $M_H = M_L = 80\%$) improve operation when inlet is disturbed, but worsen performance when outlet is disturbed (dashed lines represent $M_H = M_L = 50\%$)

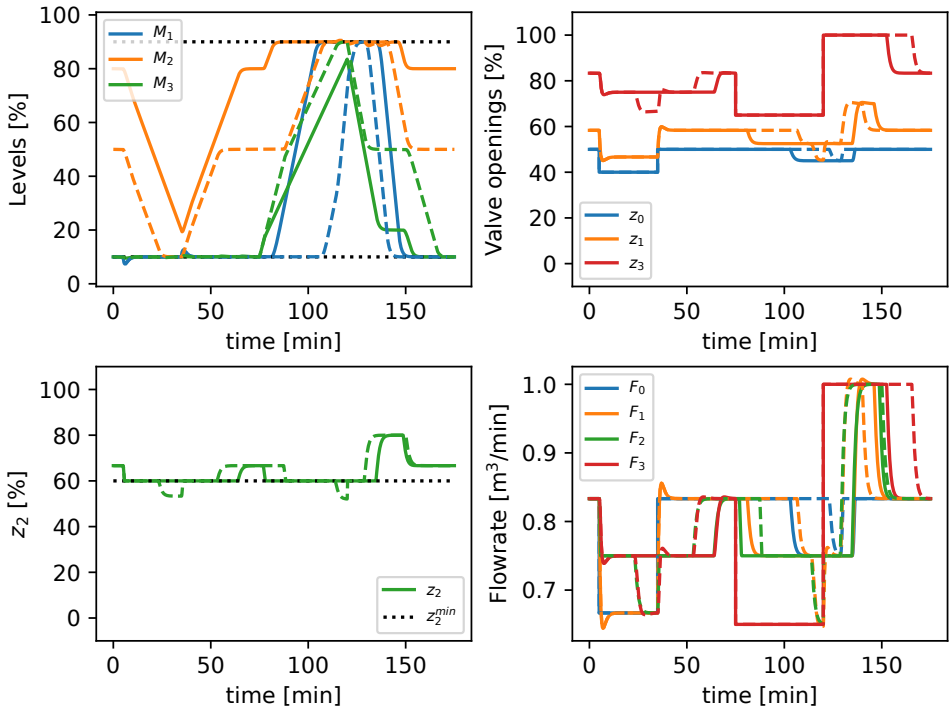


Figure 7.7: With intermediary setpoints farther from critical values (continuous line, $M_H = 20\%$, $M_L = 80\%$), the period of feasible operation is maximized for disturbances on z_0 and z_3 (dashed lines represent $M_H = M_L = 50\%$)

7.4 Discussion

The case study presented in this paper aims to reproduce a simplified version of unit operations in series. These unit operations are for simplicity represented as valves, and the buffer tanks represent the holdups between units. Therefore, for the simulations presented in this work, temporary constraints on maximum allowed valve opening represent temporary reductions on the operating capacity of the units. For example, the case presented in Figures 7.2 and 7.3 refers to a hypothetical process with a steady-state capacity bottleneck on the last unit ($z_3 = 100\%$), and all other units operate below their nominal capacity. The temporary bottleneck introduced by setting $z_0 = 50\%$ represents a temporary limit on the processing capacity of that unit. Finally, the unit represented by z_2 must always operate above a certain throughput ($z_2^{min} = 60\%$), so that abnormal operation is avoided.

As evidenced by the presented results, the margin between intermediary and extreme inventory levels is used for satisfying minimum flow constraints during transients. However, satisfying this constraint is not always feasible, since the mass balances are forcibly not satisfied to attain feasibility. This contrasts with the principle of consistent inventory control, which states that mass balances should be satisfied at steady state with the proposed control structure. Due to this, satisfaction of the minimum flow constraint must be given up, being overriden by consistent inventory control loops at critical inventory levels. It must be noted that the problem of maximizing the flow through a series of tanks is always feasible, and it is solved automatically by the bidirectional inventory control structure.

As can be noted from the bidirectional control structure, the use of min selectors automatically yields the maximum feasible inputs, since the inputs that were not selected would violate the objective corresponding to the input that was selected. Although this maximizes production, a max selector must be used to check for violation of the minimum flow constraint. Since satisfying the minimum flow constraint may not always be feasible, such constraint must be placed at a lower priority than level control at extreme conditions. Similarly to Krishnamoorthy and Skogestad (2020), where min and max selectors are combined for optimal switching between constraints at steady state, the order in the implementation of the selectors is related to the order in which the objectives must be given up. Therefore, if the minimum flow constraint is to be given up face to critical inventory levels, the max selector must be implemented before the min selector related to the extreme inventory control loops in Figure 7.1.

While the extreme inventory levels are defined by conditions such as drying out or overflow, there are several conflicting objectives when selecting the intermediary inventory setpoints, as shown by the simulations. In some cases, the override with

the minimum flow constraint can be regarded as too conservative, since for short enough disturbances this override may prove unnecessary. This requires some knowledge on the nature of the expected disturbances, so that a reasonable value for those setpoints is selected. For instance, if the minimum flow constraint was to be fully prioritized, the choice of intermediary setpoints as $M_H = L$ and $M_L = H$ would maximize the time for which the system can run feasibly, at the expense of bypassing the buffering ability of the inventories for dealing with temporary bottlenecks. On the other hand, choosing $M_H = H$ and $M_L = L$, which is the same as removing the red portion of Figure 7.1, maximizes production under bottlenecks, ignoring the minimum flow constraints. The selection of $M_L = M_H = 50\%$ is the more conservative approach, when information about the possible scenarios is not available. If the buffer tanks are large enough, this choice will be sufficient to reject all types of disturbances, whether they affect the minimum flow constraint or not.

It was shown in Figure 7.3 that implementing intermediate level setpoints leads to some loss in terms of rejecting temporary bottlenecks, since the period for which the system can run with maximum production is proportional to the gap between low and high inventory setpoints. The bidirectional inventory control structure implements the optimal policy of maximizing production at the bottleneck, constrained to the inventory bounds, and the control logic added on top of it makes a compromise between this objective and minimum flow constraints.

In industrial applications with minimum flow constraints, if such constraint is to be violated, the system must be shut down to prevent equipment damage. This is done until inventories are restored to operational levels, and operation can be then restarted. Instead, if a shutdown is not desired, the minimum flow constraints can be often dealt with by anti-surge systems, which recycle part of the outflow of the unit so as to guarantee minimum flow. However, this solution may be too expensive, since it generates a recycle flow, which is in turn tied to more pumping costs in the operation. The control structure proposed in this work may reduce these costs while normal operation is feasible, and can also be overriden by anti-surge control when the system reaches critical levels, which is a simple and effective way of solving the feasibility issues of the current proposal.

If the optimization problem from Equation (7.1) was to be solved through dynamic optimization, assumptions about the nature of the disturbances should be clearly made. For example, if economic MPC was to be implemented in the present case study, the optimal levels for operation would not matter, unless some disturbance is expected. Therefore, in order to determine the optimal operating inventory levels, and to make a compromise between the conflicting cases we presented in the simulations of this work, robust approaches such as multi-stage NMPC (Lucia et al.

2014) should be employed. This would come at the expense of implementation complexity, in terms of system and disturbances modeling, and the high computational cost inherent to the tool. The control structure presented in the current work aims to solve the operational problem using simple control structures, and the compromise between objectives is done by setting reasonable values to the setpoints of the inventory control loops. This can be done with offline analysis through simulation of the different disturbance scenarios, but it can also be easily done manually after implementation. Such flexibility is hardly obtained when using centralized optimization-based strategies.

7.5 Conclusion

The control structure proposed in this work was able to account for constraints related to minimum allowed flow, when satisfying these constraints is feasible dynamically. When compared to the original bidirectional control structure, similar behavior is observed when the system is far from the constraint, and a more conservative behavior is observed when the constraint becomes active, where feasible operation is favored over maximizing immediate production. The implementation of this control structure is therefore recommended when constraints related to maximum and minimum allowed flow must be considered simultaneously, together with other strategies that ensure feasibility, such as anti-surge loops.

The main limitation of the proposed control strategy is that, being built as an extension of the bidirectional inventory control, it only considers a linear arrangement of the inventories. As splits and recycles are very common in process systems, an interesting topic of research would be to propose extensions to this framework to more complex arranges of inventories, such that production maximization is achieved, respecting operational constraints, using simple feedback control elements.

Chapter 8

Self-optimizing control methods for ill-conditioned problems and optimal disturbance rejection

Here, a variation of existing methods for self-optimizing control is presented, focusing on addressing the limitations of their applicability. The existing methods are briefly presented in Section 8.1, the proposed reformulation is presented in Section 8.2, and a numerical example is presented in Section 8.3.

8.1 Problem formulation and existing methods

The formulation presented in this section is mostly based on the material and derivations from Alstad et al. (2009).

Consider that we wish to find the optimum of a system through feedback. The steady-state optimization problem for the system is:

$$\min_u \; J(u,d) = \; J^* + \begin{bmatrix} J_u^{*T} & J_d^{*T} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta u^T & \Delta d^T \end{bmatrix} \underbrace{\begin{bmatrix} J_{uu} & J_{ud} \\ J_{ud}^T & J_{dd} \end{bmatrix}}_{\mathcal{H}} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} \tag{8.1}$$

Define $\Delta u = u - u^*$ and $\Delta d = d - d^*$ as deviation variables from the nominal conditions (u^*, d^*), for which information is available. The system has measured variables y, which follow the linear model:

$$\Delta y = y - y^* = G^y \Delta u + G_d^y \Delta d \tag{8.2}$$

The system is subject to unmeasured disturbances d, and the measurements y_m are subject to error, such that $y_m = y + n^y$, where n^y represents the steady-state measurement error (bias). The goal is to find controlled variables as a linear combination of the measurements, $c = Hy$, such that the operational loss is minimized. We assume that the problem Hessian $\mathcal{H}$ is constant, and that J_{uu} is a positive definite matrix, which means that the unconstrained cost gradient $J_u(u, d)$ is a linear function of the inputs and disturbances:

$$J_u(u, d) = J_u^* + J_{uu}\Delta u + J_{ud}\Delta d \tag{8.3}$$

The loss encountered by applying an input u, compared to using the optimal input $u^{opt}(d)$, is derived from (8.1) as:

$$L = J(u, d) - J^{opt}(d) = \frac{1}{2}(u - u^{opt})^T J_{uu}(u - u^{opt}) = \frac{1}{2}||z||_2^2 \tag{8.4}$$

where $J^{opt}(d) = J(u^{opt}(d), d)$ is the optimal cost for a given d and the loss variable z is defined as:

$$z \triangleq J_{uu}^{1/2}(u - u^{opt}) \tag{8.5}$$

For simplicity, assume that the nominal conditions are optimal, meaning that $J_u^* = 0$. The optimal values $\Delta u^{opt} = u^{opt} - u^*$ and $\Delta y^{opt} = y^{opt} - y^*$ for the unconstrained problem are then obtained by setting $J_u(u^{opt}, d) = 0$ to give:

$$\Delta u^{opt} = -J_{uu}^{-1} J_{ud} \Delta d \tag{8.6}$$

$$\Delta y^{opt} = G^y \Delta u^{opt} + G_d^y \Delta d = F \Delta d \tag{8.7}$$

with $F = -G^y J_{uu}^{-1} J_{ud} + G_d^y$. We then rewrite the loss variable z in terms of the controlled variables c:

$$\begin{cases} \Delta c = HG^y \Delta u + HG_d^y \Delta d \\ \Delta c^{opt} = HG^y \Delta u^{opt} + HG_d^y \Delta d \end{cases}$$
$$\implies u - u^{opt} = \Delta u - \Delta u^{opt} = (HG^y)^{-1}(\Delta c - \Delta c^{opt})$$

$$z = J_{uu}^{1/2}(u - u^{opt}) = J_{uu}^{1/2}(HG^y)^{-1}(\Delta c - \Delta c^{opt}) \tag{8.8}$$

where Δc and Δc^{opt} are given by:

$$\Delta c = \Delta c_m - Hn^y \tag{8.9}$$

$$\Delta c^{opt} = H\Delta y^{opt} = HF\Delta d \tag{8.10}$$

It is desired that the measured value c_m is controlled to a constant setpoint, meaning $\Delta c_m = 0$.

Here, we define normalized vectors d' and $n^{y\prime}$, such that $\Delta d = W_d d'$ and $n^y = W_{n^y} n^{y\prime}$ and the vectors are norm bounded, that is:

$$\left|\left|\begin{bmatrix} d' \\ n^{y\prime} \end{bmatrix}\right|\right|_2 \leq 1 \tag{8.11}$$

The loss variable z is then written in terms of these normalized vectors as:

$$z = M_d d' + M_{n^y} n^{y\prime} \tag{8.12}$$

where:

$$M_d \triangleq -J_{uu}^{1/2}(HG^y)^{-1}HFW_d \tag{8.13}$$

$$M_{n^y} \triangleq -J_{uu}^{1/2}(HG^y)^{-1}HW_{n^y} \tag{8.14}$$

We then introduce M_n and M as follows:

$$M_n \triangleq J_{uu}^{1/2}(HG^y)^{-1} \tag{8.15}$$

$$M \triangleq \begin{bmatrix} M_d & M_{n^y} \end{bmatrix} = -M_n H \underbrace{\begin{bmatrix} FW_d & W_{n^y} \end{bmatrix}}_{\tilde{F}} \tag{8.16}$$

Given the expected disturbances and implementation error, the worst-case loss is then given by (Halvorsen et al. 2003):

$$L_{wc} = \max_{\left\| \begin{bmatrix} d' \\ n^{y\prime} \end{bmatrix} \right\|_2 \leq 1} L = \frac{1}{2} \left(\bar{\sigma}[M] \right)^2 \tag{8.17}$$

Similarly, the average loss for this problem is given by (Kariwala et al. 2008):

$$L_{av} = \frac{1}{6(n_y + n_d)} ||M||_F^2 \tag{8.18}$$

Kariwala et al. (2008) have shown that the matrix H that minimizes the Frobenius norm of M in this problem also minimizes its biggest singular value, although the reverse is not true. They have also shown that if the realizations of d' and $n^{y\prime}$ are infinity-norm bounded, minimizing the Frobenius norm of M places an upper bound on the worst-case loss.

Therefore, the optimal measurement combination H to minimize both the worst-case and the average loss, which is known as the **exact local method**, can be found by solving the optimization problem:

$$H = \arg\min_H ||M||_F^2 \tag{8.19}$$

This problem presents an infinite number of solutions, and Alstad et al. (2009) derived an analytical solution by introducing the constraint $HG^y = J_{uu}^{1/2}$, which results in:

$$H^T = \left(\tilde{F} \tilde{F}^T \right)^{-1} G^y \left[G^{yT} \left(\tilde{F} \tilde{F}^T \right)^{-1} G^y \right]^{-1} J_{uu}^{1/2} \tag{8.20}$$

This constraint can be added without any losses because if H is a solution to Equation (8.19), then so is $H_1 = DH$, where D is any invertible $n_u \times n_u$ matrix. This can be verified by checking that M_d and M_{n^y} are unaffected by the choice of D with the candidate solution H_1. Therefore, we can rewrite this solution using the definition of M_n in (8.15) as:

$$H = M_n^{-1} J_{uu}^{1/2} \left[G^{yT} \left(\tilde{F} \tilde{F}^T \right)^{-1} G^y \right]^{-1} G^{yT} \left(\tilde{F} \tilde{F}^T \right)^{-1} \tag{8.21}$$

Alternatively, one may want to find measurement combinations H that completely reject disturbances (i.e., by enforcing $M_d = 0$). To accomplish that, we must

find an H that is in the left nullspace of F as defined in Equation (8.7) (Alstad and Skogestad 2007). With that end, the **extended nullspace method** (Alstad et al. 2009) presents a solution that prioritizes the minimization of M_d, attaining $M_d = 0$ if there are enough degrees of freedom, and minimizes $||M_{n^y}||_F$ with the remaining degrees of freedom. It does this by defining the residual matrix E as:

$$E \triangleq M_n H \tilde{G}^y - \tilde{J} = \underbrace{M_n H W_{n^y}}_{-M_{n^y}} W_{n^y}^{-1} \tilde{G}^y - \tilde{J} \tag{8.22}$$

where $\tilde{G}^y = [G^y \quad G_d^y]$ and $\tilde{J} = J_{uu}^{1/2}[I \quad J_{uu}^{-1} J_{ud}]$. Here, the solution M_{n^y} that minimizes $||E||_F$ if the system is overdetermined (i.e. $E = 0$ cannot be satisfied) and minimizes $||M_{n^y}||_F$ if the system is underdetermined (i.e. there are too many degrees of freedom) is given by:

$$M_{n^y} = \tilde{J}(W_{n^y}^{-1} \tilde{G}^y)^\dagger$$

The optimal H in this case is given by:

$$H = M_n^{-1} \tilde{J}(W_{n^y}^{-1} \tilde{G}^y)^\dagger W_{n^y}^{-1} \tag{8.23}$$

This solution achieves an upper bound on the value of $||M_d||_F$, because E and M_d are related by:

$$M_d = -M_n H F W_d = -M_n H \tilde{G}^y \begin{bmatrix} -J_{uu}^{-1} J_{ud} \\ I \end{bmatrix} W_d$$

$$\implies M_d = (E + \tilde{J}) \begin{bmatrix} J_{uu}^{-1} J_{ud} \\ -I \end{bmatrix} W_d = E \begin{bmatrix} J_{uu}^{-1} J_{ud} \\ -I \end{bmatrix} W_d$$

$$\implies ||M_d||_F \leq ||E||_F \cdot \left\| \begin{bmatrix} J_{uu}^{-1} J_{ud} \\ -I \end{bmatrix} W_d \right\|_F$$

This solution, although simple and compact, requires W_{n^y} to be invertible, and it does not achieve the optimal $||M_d||_F$ nor it enforces $HG^y = M_n^{-1} J_{uu}^{1/2}$ for the case with few measurements ($n_y < n_u + n_d$).

In both methods, there is a limitation regarding the conditioning of the problem. The exact local method as posed requires that $\tilde{F}\tilde{F}^T$ is invertible, which is the case when all measurements have noise, but it cannot be directly applied to the

noiseless case (W_{n^y} not invertible). In the next section, we present an alternative formulation also valid for these limiting cases.

8.2 Proposed reformulation

The exact local method is formulated as the following optimization problem:

$$\begin{aligned} \min_{H} \quad & \frac{1}{2}||H\tilde{F}||_F^2 \\ \text{s.t.} \quad & HG^y = G \end{aligned} \tag{8.24}$$

Here, we use G to simplify notation, reminding that this definition is arbitrary for solving (8.19) (as long as G is invertible). Because $||H\tilde{F}||_F^2 = \mathbf{tr}(H\tilde{F}\tilde{F}^T H^T) = \sum_i h_i \tilde{F}\tilde{F}^T h_i^T$, each row of H behaves independently, according to the associated set of subproblems:

$$\begin{aligned} \min_{h_i} \quad & \frac{1}{2} h_i \tilde{F}\tilde{F}^T h_i^T \\ \text{s.t.} \quad & h_i G^y = G_i \end{aligned} \tag{8.25}$$

where h_i and G_i denote the i-th row of H and G respectively. We can see that the solution to the original problem can be written as the concatenation of the solutions to each subproblem, and all subproblems have the same formulation except for the right-hand side of the equality constraint. Therefore, the first-order KKT conditions for the full problem (8.24) can be written as:

$$\begin{bmatrix} \tilde{F}\tilde{F}^T & G^y \\ G^{yT} & 0_{n_u \times n_u} \end{bmatrix} \begin{bmatrix} H^T \\ \Lambda \end{bmatrix} = \begin{bmatrix} 0_{n_y \times n_u} \\ G^T \end{bmatrix} = \begin{bmatrix} 0_{n_y \times n_u} \\ I_{n_u} \end{bmatrix} G^T \tag{8.26}$$

Note that, by definition, $\tilde{F}\tilde{F}^T$ is positive semidefinite, and therefore the first-order conditions are necessary and sufficient optimality conditions. If $\tilde{F}\tilde{F}^T$ can be inverted, the solution is unique, and we arrive at the result presented in (8.20) by doing Gaussian elimination. Otherwise, we can find a minimum-norm solution through the pseudo-inverse of the full KKT matrix, leading to:

$$\begin{bmatrix} H^T \\ \Lambda \end{bmatrix} = \begin{bmatrix} \tilde{F}\tilde{F}^T & G^y \\ G^{yT} & 0_{n_u \times n_u} \end{bmatrix}^{\dagger} \begin{bmatrix} 0_{n_y \times n_u} \\ I_{n_u} \end{bmatrix} G^T \tag{8.27}$$

$$H^T = \begin{bmatrix} I_{n_y} & 0_{n_y \times n_u} \end{bmatrix} \begin{bmatrix} H^T \\ \Lambda \end{bmatrix} \Longrightarrow$$

$$H = G \begin{bmatrix} 0_{n_u \times n_y} & I_{n_u} \end{bmatrix} \begin{bmatrix} \tilde{F}\tilde{F}^T & G^y \\ G^{yT} & 0_{n_u \times n_u} \end{bmatrix}^{\dagger} \begin{bmatrix} I_{n_y} \\ 0_{n_u \times n_y} \end{bmatrix} \tag{8.28}$$

This solution is equal to the original exact local method in Equation (8.20) with $G = J_{uu}^{1/2}$ in the case where $\tilde{F}\tilde{F}^T$ is invertible and G^y is full column rank, and it can be directly used in the case where $\tilde{F}\tilde{F}^T$ is not invertible. In the latter case, the solution (8.27) minimizes the Frobenius norm of the solution matrix $\begin{bmatrix} H & \Lambda^T \end{bmatrix}^T$.

We can use the same idea to formulate an optimization-based extended nullspace method that directly minimizes $||M_d||_F$ (instead of $||E||_F$), and subsequently minimizes $||M_{n^y}||_F$ over all the minimizers of $||M_d||_F$. This problem is written as:

$$\begin{aligned} \min_{H} \quad & \frac{1}{2}||HW_{n^y}||_F^2 \\ \text{s.t.} \quad & H = \arg\min_{H} \frac{1}{2}||HFW_d||_F^2 \\ & \text{s.t.} \quad HG^y = G \end{aligned} \tag{8.29}$$

The KKT conditions for the internal problem are written as:

$$\begin{bmatrix} FW_d(FW_d)^T & G^y \\ G^{yT} & 0_{n_u \times n_u} \end{bmatrix} \begin{bmatrix} H^T \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} 0_{n_y \times n_u} \\ I_{n_u} \end{bmatrix} G^T \tag{8.30}$$

We express these as equality constraints to the external problem, leading to the corresponding KKT conditions:

$$\begin{aligned} \min_{H,\Lambda_i} \quad & \frac{1}{2}||HW_{n^y}||_F^2 \\ \text{s.t.} \quad & \begin{bmatrix} FW_d(FW_d)^T & G^y \\ G^{yT} & 0_{n_u \times n_u} \end{bmatrix} \begin{bmatrix} H^T \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} 0_{n_y \times n_u} \\ I_{n_u} \end{bmatrix} G^T \end{aligned}$$

$$\implies \Pi \begin{bmatrix} H^T \\ \Lambda_i \\ \Lambda_{e_1} \\ \Lambda_{e_2} \end{bmatrix} = \begin{bmatrix} 0_{n_y \times n_u} \\ 0_{n_u} \\ 0_{n_y \times n_u} \\ I_{n_u} \end{bmatrix} G^T \tag{8.31}$$

$$\Pi = \begin{bmatrix} W_{n^y} W_{n^y}^T & 0_{n_y \times n_u} & FW_d(FW_d)^T & G^y \\ 0_{n_u \times n_y} & 0_{n_u \times n_u} & G^{yT} & 0_{n_u \times n_u} \\ FW_d(FW_d)^T & G^y & 0_{n_y \times n_y} & 0_{n_y \times n_u} \\ G^{yT} & 0_{n_u \times n_u} & 0_{n_u \times n_y} & 0_{n_u \times n_u} \end{bmatrix} \tag{8.32}$$

The minimum-norm solution is given by:

$$H = G \begin{bmatrix} 0_{n_u \times (2n_y + n_u)} & I_{n_u} \end{bmatrix} \Pi^{\dagger} \begin{bmatrix} I_{n_y} \\ 0_{(2n_u + n_y) \times n_y} \end{bmatrix} \tag{8.33}$$

In problem (8.29), the internal problem is prioritized over the external. If the solution to the internal problem is unique, the external optimization problem only has one feasible point, regardless of the value of W_{n^y}. In the case where there are multiple solutions to the external problem, the solution given by applying the pseudo-inverse minimizes the norm of the full vector comprised of H and all Lagrange multipliers.

Different types of regularization can be used on this problem, depending on which solution for H is to be prioritized. To maximize the sparsity of H, ℓ_1-norm regularization (also known as lasso) can be used, while ℓ_2-norm regularization (also known as Tikhonov regularization or ridge regression) minimizes the Frobenius norm of H (as opposed to the norm of H together with the Lagrange multiplies) (Boyd and Vandenberghe 2004). A combination of both norms is used in the method known as the elastic net (Zou and Hastie 2005). These are alternatives to the use of the pseudo-inverses presented in Equations (8.28) and (8.33), but the parameters used in these methods are problem-dependent, and therefore the formulation with pseudo-inverses is deemed as the simplest generic approach.

The solution in Equation (8.29) is equivalent to Equation (8.23) for $n_y \geq n_u + n_d$, because $M_d = 0$ for both approaches. Furthermore, (8.29) minimizes the effect of disturbances on the loss, $||M_d||_F$, for $n_y < n_u + n_d$, as opposed to (8.23), which can only ensure an upper bound on $||M_d||_F$. It also follows that the nullspace method (without information of measurement bias) can be formulated by only considering the internal problem in Equation (8.29), or equivalently by using the proposed exact local method form in Equation (8.28) with $W_{n^y} = 0$. The advantages of using this form are that it is valid for $n_y < n_u + n_d$, in which case it minimizes $||M_d||_F$, and it finds the minimum-norm solution that drives M_d to zero for $n_y \geq n_u + n_d$, with the additional benefit that the resulting self-optimizing variables are normalized to have the steady-state input gain matrix equal to G.

We remark that Equation (8.28) is **not** equivalent to replacing $(\tilde{F}\tilde{F}^T)^{-1}$ by the cor-

responding pseudo-inverse in Equation (8.20). If one wants to use Equation (8.20) for an ill-conditioned problem, a regularization factor must be used. For example, $(\tilde{F}\tilde{F}^T+\delta I)^{-1}$ can be used as a replacement for $(\tilde{F}\tilde{F}^T)^{-1}$, where δ is a small number that guarantees good conditioning for the matrix inversion. This is equivalent to artificially using a small non-zero value for the measurement noise magnitude, W_{n^y}, or a ℓ_2-norm regularized solution for H.

8.3 Numerical example

Let us consider an example with the following matrices ($n_u = 4,\ n_d = 2,\ n_y = 5 < n_u + n_d$):

$$G^y = \begin{bmatrix} 1.03 & 0.89 & 1.44 & -0.10 \\ 0.73 & -1.15 & 0.33 & -0.24 \\ 0.49 & -0.79 & -2.94 & -1.71 \\ -0.30 & -1.07 & -0.75 & 0.32 \\ 0.29 & -0.81 & 1.37 & 0.31 \end{bmatrix},\ G_d^y = \begin{bmatrix} -0.03 & -0.86 \\ -0.16 & 0.08 \\ -0.86 & 1.11 \\ 0.63 & -1.21 \\ 1.09 & -1.11 \end{bmatrix},$$

$$J_{uu} = \begin{bmatrix} 3.84 & 1.08 & 0.66 & 0.79 \\ 1.08 & 2.00 & 1.17 & 0.14 \\ 0.66 & 1.17 & 1.95 & 1.87 \\ 0.79 & 0.14 & 1.87 & 3.10 \end{bmatrix},\ J_{ud} = \begin{bmatrix} -0.12 & 0.67 \\ 1.49 & -1.21 \\ 1.41 & 0.72 \\ 1.42 & 1.63 \end{bmatrix}$$

With this information, F is calculated as:

$$F = \begin{bmatrix} 1.8429 & -3.6811 \\ 3.9232 & -4.7642 \\ -1.8437 & 5.2544 \\ 0.6561 & -1.1400 \\ 5.3120 & -7.1543 \end{bmatrix}$$

Let us now apply the self-optimizing control methods, with the choice $HG^y = G = J_{uu}$, or equivalently $M_n = J_{uu}^{-1/2}$. With $W_d = \text{diag}([4,\ 4])$ and $W_{n^y} = \text{diag}([10^{-3},\ 10^{-3},\ 4,\ 10^{-3},\ 4])$, we get the following results for (8.28) and (8.21):

$$H = \begin{bmatrix} 4.9567 & 3.6539 & -1.4564 & 4.8593 & -6.0735 \\ 2.0198 & -0.7267 & -0.0081 & 1.0175 & -0.5543 \\ 1.7891 & 1.4224 & -1.5145 & 2.1563 & -2.8694 \\ 2.2643 & 3.3823 & -2.6184 & 4.0225 & -5.2468 \end{bmatrix}$$

For this solution, we get $||H\tilde{F}||_F = 53.1986$. Applying the original extended nullspace method (8.23) leads to:

$$H = \begin{bmatrix} 1.5179 & 1.2784 & -0.0141 & -0.0696 & -0.9260 \\ 1.0260 & -1.8992 & 0.6151 & -0.4482 & 1.4982 \\ -0.4031 & -0.3558 & -0.4830 & -1.0083 & 0.7187 \\ -1.1215 & 0.9656 & -1.1653 & -0.8372 & -0.0881 \end{bmatrix}$$

Even though we use $M_n = J_{uu}^{-1/2}$, this is not enough to guarantee $HG^y = J_{uu}$, because $W_{n^y}^{-1}\tilde{G}^y$ does not have independent rows (which comes from $n_y < n_u + n_d$). We must therefore rescale the result to compare the objective function value. This gives:

$$H = \begin{bmatrix} 22.3748 & -35.0732 & 12.7959 & 25.5267 & 26.8463 \\ 8.0654 & -14.1683 & 4.9387 & 8.1908 & 10.8716 \\ 13.4424 & -24.4872 & 8.0207 & 15.9834 & 19.1548 \\ 19.5764 & -35.1091 & 11.5471 & 24.5640 & 27.4725 \end{bmatrix}$$

with $||H\tilde{F}||_F = 543.3125$, which is an order of magnitude higher than the exact local method solution. On the other hand, applying (8.33) leads to:

$$H = \begin{bmatrix} 4.3009 & 5.1119 & -1.9929 & 4.0812 & -7.3128 \\ 1.9653 & -0.6054 & -0.0527 & 0.9528 & -0.6574 \\ 1.4443 & 2.1890 & -1.7966 & 1.7472 & -3.5211 \\ 1.6402 & 4.7699 & -3.1291 & 3.2820 & -6.4263 \end{bmatrix}$$

for which $||H\tilde{F}||_F = 56.4358$, a value closer to the overall optimum. We can measure the improved disturbance rejection by comparing the value of $||HF||_F$ for the solutions of (8.28) and (8.33), which are 9.6257 and 8.6293 respectively. The proposed method (8.33) works as expected, in the sense that it gives up overall performance to prioritize disturbance rejection.

Chapter 9

Conclusion

This thesis has investigated some aspects of dealing with changing active constraints for optimal operation. The approach followed in Chapters 2 to 4 attempted to be process-agnostic, in the sense that they can be applied to any process system, given that a steady-state model is available for design. Therefore, the author believes that these methods are useful for real applications and wishes that their limitations are addressed.

A clear limitation of Chapter 2 is the number of constraints that can be dealt with optimally. In Chapters 5 and 6 possible solutions are presented, but they do not seem to scale well with the number of extra constraints (in the case of region-based control), or there are fundamental performance limitations that need to be addressed (in the case of primal-dual optimizing control). It would be interesting to solve the case with more constraints in the general case, but this does not seem realistic under the presented framework.

With that in mind, Chapter 3 describes an MPC framework that is complimentary to the decentralized region-based approach, and can be used in scenarios with several constraints. The methodology does not require direct estimation of disturbances for optimization, as it relies on an offset-free formulation for the state estimator to give steady-state predictions for the measurements, which are in turn used to estimate the cost gradient. Although the MPC community recently favors the combination of optimization and control layers, this work is a step in the opposite direction of this trend. The optimization problem is considered in the design of the control layer but is considered as a different task. Aspects of stability and robustness were not addressed in the proposed framework, and therefore the presented ideas should be linked with other developments on MPC formulations

present in the literature.

In Chapter 4 the link between self-optimizing control and gradient estimation was further explored. This gives more applicability for feedback optimizing methods that require gradient information, given that a reasonable estimate can be obtained as a linear combination of the measurements. Furthermore, this increases the relevance of studying methods for designing self-optimizing CVs. In Chapter 8 some remarks about the self-optimizing control formulation are presented, with a simplified derivation and expanded results.

The system and control structure studied in Chapter 7 has a considerably different nature than what is studied in the rest of the thesis, but it highlights how selectors, which are used for steady-state active constraint switching in the other chapters, can also be used for dynamic constraint handling. In prior bidirectional control structures, the optimal steady-state solution can be implemented with optimal use of inventories to reject disturbances. The introduction of minimum flow constraints leads to a system that may have infeasible operation at steady state, but it is still possible to optimize the transients with additional control loops and then prioritize the constraints to be given up in the case of infeasibility. This case study also further illustrates how simple control solutions can be implemented to operate optimally for seemingly complex problems.

Bibliography

V. Alstad. *Studies on selection of controlled variables*. PhD thesis, Fakultet for naturvitenskap og teknologi, 2005.

V. Alstad and S. Skogestad. Null space method for selecting optimal measurement combinations as controlled variables. *Industrial & engineering chemistry research*, 46:846–853, 2007.

V. Alstad, S. Skogestad, and E. S. Hori. Optimal measurement combinations as controlled variables. *Journal of Process Control*, 19(1):138–148, 2009.

J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019. doi: 10.1007/s12532-018-0139-4.

K. J. Åström and L. Rundqwist. Integrator windup and how to avoid it. In *1989 American Control Conference*, pages 1693–1698. IEEE, 1989.

L. F. Bernardino and S. Skogestad. Bidirectional inventory control with optimal use of intermediate storage and minimum flow constraints. *IFAC-PapersOnLine*, 56(2):2665–2670, 2023a.

L. F. Bernardino and S. Skogestad. Decentralized control for optimal operation under changing active constraints. In *Computer Aided Chemical Engineering*, volume 52, pages 1699–1704. Elsevier, 2023b.

L. F. Bernardino and S. Skogestad. Real-time optimization with changing active constraints solved through decentralized feedback control. In *2023 AIChE Annual Meeting*. AIChE, 2023c.

L. F. Bernardino and S. Skogestad. Optimal measurement-based cost gradient estimate for real-time optimization. *Submitted to Computers & Chemical Engineering*, 2024a.

L. F. Bernardino and S. Skogestad. Optimal switching of MPC cost function for changing active constraints. *Submitted to Journal of Process Control*, 2024b.

L. F. Bernardino and S. Skogestad. Decentralized control using selectors for optimal steady-state operation with changing active constraints. *Journal of Process Control*, 137:103194, 2024c.

L. F. Bernardino, D. Krishnamoorthy, and S. Skogestad. Comparison of simple feedback control structures for constrained optimal operation. *IFAC-PapersOnLine*, 55(7):883–888, 2022a.

L. F. Bernardino, D. Krishnamoorthy, and S. Skogestad. Optimal operation of heat exchanger networks with changing active constraint regions. In *Computer Aided Chemical Engineering*, volume 49, pages 421–426. Elsevier, 2022b.

A. Boucheikhchoukh, V. Berger, C. L. Swartz, A. Deza, A. Nguyen, and S. Jaffer. Multiperiod refinery optimization for mitigating the impact of process unit shutdowns. *Computers & Chemical Engineering*, 164:107873, 2022.

S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Y. Cao. Constrained self-optimizing control via differentiation. *IFAC Proceedings Volumes*, 37(1):63–70, 2004.

Z. Chong and C. L. Swartz. Optimal operation of process plants under partial shutdown conditions. *AIChE Journal*, 59(11):4151–4168, 2013.

P. d. A. Delou, R. Curvelo, M. B. de Souza Jr, and A. R. Secchi. Steady-state real-time optimization using transient measurements in the absence of a dynamic mechanistic model: A framework of HRTO integrated with Adaptive Self-Optimizing IHMPC. *Journal of Process Control*, 106:1–19, 2021.

R. Dirza and S. Skogestad. Primal-dual feedback-optimizing control with override for real-time optimization. *Submitted to Computers & Chemical Engineering*, 2024.

R. Dirza, S. Skogestad, and D. Krishnamoorthy. Optimal resource allocation using distributed feedback-based real-time optimization. *IFAC-PapersOnLine*, 54(3): 706–711, 2021.

J. J. Downs and S. Skogestad. An industrial and academic perspective on plantwide control. *Annual Reviews in Control*, 35(1):99–110, 2011.

J. C. Doyle. Guaranteed margins for lqg regulators. *IEEE Transactions on automatic Control*, 23(4):756–757, 1978.

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.

G. François, B. Srinivasan, and D. Bonvin. Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *Journal of Process Control*, 15(6):701–712, 2005.

R. Freeman. Global internal stabilizability does not imply global external stabilizability for small sensor disturbances. *IEEE Transactions on Automatic Control*, 40(12):2119–2122, 1995.

J. E. A. Graciano, J. Jäschke, G. A. Le Roux, and L. T. Biegler. Integrating self-optimizing control and real-time optimization using zone control MPC. *Journal of Process Control*, 34:35–48, 2015.

S. Gros, B. Srinivasan, and D. Bonvin. Optimizing control based on output feedback. *Computers & Chemical Engineering*, 33(1):191–198, 2009.

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.

M. G. Jacobsen and S. Skogestad. Active constraint regions for optimal operation of distillation columns. *Industrial & engineering chemistry research*, 51(7): 2963–2973, 2012.

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.

J. Jäschke and S. Skogestad. Optimal controlled variables for polynomial systems. *Journal of Process Control*, 22:167–179, 2012.

J. Jäschke and S. Skogestad. Optimal operation of heat exchanger networks with stream split: Only temperature measurements are required. *Computers & chemical engineering*, 70:35–49, 2014.

J. Jäschke, Y. Cao, and V. Kariwala. Self-optimizing control–A survey. *Annual Reviews in Control*, 43:199–223, 2017.

I. Karimi and G. Reklaitis. Optimal selection of intermediate storage tank capacity in a periodic batch/semicontinuous process. *AIChE Journal*, 29(4):588–596, 1983.

V. Kariwala, Y. Cao, and S. Janardhanan. Local self-optimizing control with average loss minimization. *Industrial & Engineering Chemistry Research*, 47(4): 1150–1158, 2008.

D. Krishnamoorthy. A distributed feedback-based online process optimization framework for optimal resource sharing. *Journal of Process Control*, 97:72–83, 2021.

D. Krishnamoorthy and S. Skogestad. Online process optimization with active constraint set changes using simple control structures. *Industrial & Engineering Chemistry Research*, 58(30):13555–13567, 2019.

D. Krishnamoorthy and S. Skogestad. Systematic design of active constraint switching using selectors. *Computers & Chemical Engineering*, 143:107106, 2020.

D. Krishnamoorthy and S. Skogestad. Real-time optimization as a feedback control problem - a review. *Computers & Chemical Engineering*, page 107723, 2022.

D. Krishnamoorthy, E. Jahanshahi, and S. Skogestad. Feedback real-time optimization strategy using a novel steady-state gradient estimate and transient measurements. *Industrial & Engineering Chemistry Research*, 58(1):207–216, 2018.

J. Lee and T. F. Edgar. Conditions for decentralized integral controllability. *Journal of Process Control*, 12(7):797–805, 2002.

D. Liberzon and A. S. Morse. Basic problems in stability and design of switched systems. *IEEE control systems magazine*, 19(5):59–70, 1999.

H. Lin and P. J. Antsaklis. Stability and stabilizability of switched linear systems: A survey of recent results. *IEEE Transactions on Automatic control*, 54(2): 308–322, 2009.

S. Lucia, J. A. Andersson, H. Brandt, M. Diehl, and S. Engell. Handling uncertainty in economic nonlinear model predictive control: A comparative case study. *Journal of Process Control*, 24(8):1247–1259, 2014.

A. Maarleveld and J. Rijnsdorp. Constraint control on distillation columns. *Automatica*, 6(1):51–58, 1970.

U. Maeder, F. Borrelli, and M. Morari. Linear offset-free model predictive control. *Automatica*, 45(10):2214–2222, 2009.

H. Manum and S. Skogestad. Self-optimizing control with active set changes. *Journal of process control*, 22(5):873–883, 2012.

A. Marchetti, B. Chachuat, and D. Bonvin. Modifier-adaptation methodology for real-time optimization. *Industrial & engineering chemistry research*, 48(13): 6022–6033, 2009.

A. G. Marchetti, T. de Avila Ferreira, S. Costello, and D. Bonvin. Modifier adaptation as a feedback control scheme. *Industrial & Engineering Chemistry Research*, 59(6):2261–2274, 2020.

D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

M. Morari and U. Maeder. Nonlinear offset-free model predictive control. *Automatica*, 48(9):2059–2067, 2012.

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 tasks. Analysis of the optimizing control structures. *AIChE Journal*, 26(2):220–232, 1980.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer New York, NY, second edition, 2006.

G. Pannocchia, M. Gabiccini, and A. Artoni. Offset-free MPC explained: Novelties, subtleties, and applications. *IFAC-PapersOnLine*, 48(23):342–351, 2015.

R. M. Price, P. R. Lyman, and C. Georgakis. Throughput manipulation in plantwide control structures. *Industrial & engineering chemistry research*, 33 (5):1197–1207, 1994.

J. B. Rawlings. Tutorial overview of model predictive control. *IEEE control systems magazine*, 20(3):38–52, 2000.

A. Reyes-Lúa and S. Skogestad. Systematic design of active constraint switching using classical advanced control structures. *Industrial & Engineering Chemistry Research*, 59(6):2229–2241, 2019.

A. Reyes-Lúa and S. Skogestad. Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy. *Journal of Process Control*, 91:1–11, 2020.

A. Reyes-Lúa, C. Zotică, and S. Skogestad. Optimal operation with changing active constraint regions using classical advanced control. *IFAC-PapersOnLine*, 51(18):440–445, 2018.

A. Reyes-Lúa, C. Zotică, K. Forsman, and S. Skogestad. Systematic design of split range controllers. *IFAC-PapersOnLine*, 52(1):898–903, 2019.

P. D. Roberts and T. Williams. On an algorithm for combined system optimisation and parameter estimation. *Automatica*, 17(1):199–209, 1981.

J. D. Schwartz and D. E. Rivera. A process control approach to tactical inventory management in production-inventory systems. *International Journal of Production Economics*, 125(1):111–124, 2010.

F. G. Shinskey. *Controlling multivariable processes*. Instrument Society of America, 1981.

D. Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

S. Skogestad. Plantwide control: The search for the self-optimizing control structure. *Journal of process control*, 10(5):487–507, 2000.

S. Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of process control*, 13(4):291–309, 2003.

S. Skogestad. Control structure design for complete chemical plants. *Computers & Chemical Engineering*, 28(1-2):219–234, 2004.

S. Skogestad. Advanced control using decomposition and simple elements. *Annual Reviews in Control*, 56:100903, 2023.

S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. john Wiley & sons, 2005.

S. Strand and J. R. Sagli. MPC in Statoil–advantages with in-house technology. *IFAC Proceedings Volumes*, 37(1):97–103, 2004.

Y. Tan, W. H. Moase, C. Manzie, D. Nešić, and I. M. Mareels. Extremum seeking from 1922 to 2010. In *Proceedings of the 29th Chinese control conference*, pages 14–26. IEEE, 2010.

T. J. Williams and R. E. Otto. A generalized chemical processing model for the investigation of computer control. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 79(5):458–473, 1960.

L. Woodward, M. Perrier, and B. Srinivasan. Real-time optimization using a jamming-free switching logic for gradient projection on active constraints. *Computers & Chemical Engineering*, 34(11):1863–1872, 2010.

L. Ye, Y. Cao, and S. Skogestad. Global self-optimizing control for uncertain constrained process systems. *IFAC-PapersOnLine*, 50(1):4672–4677, 2017.

L. Ye, Y. Cao, Y. He, C. Zhou, H. Su, X. Tang, and S. Yang. Generalized global self-optimizing control for chemical processes part I. The existence of perfect controlled variables and numerical design methods. *Industrial & Engineering Chemistry Research*, 62(37):15051–15069, 2023.

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.

C. Zotică, K. Forsman, and S. Skogestad. Bidirectional inventory control with optimal use of intermediate storage. *Computers & Chemical Engineering*, 159: 107677, 2022.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67 (2):301–320, 2005.