Johannes Jäschke

Invariants for Optimal Operation of Process Systems

Doctoral thesis
for the degree of philosophiae doctor

Trondheim, June 2011

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

NTNU
Innovation and Creativity

NTNU

Norwegian University of Science and Technology

Doctoral thesis
for the degree of philosophiae doctor

The Faculty of Natural Sciences and Technology
Department of Chemical Engineering

ISBN 978-82-471-2909-8 (printed version)
ISBN 978-82-471-2910-4 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2011:180

Printed by NTNU-trykk

To
Benedikt,
your footprints are in my heart.
And to
Samuel & Susanne

Abstract

State of the art strategies to achieve optimal process operation typically employ a hierarchical control structure, where different tasks are designated to different control layers. In the simplest case there is an optimization and a control layer. The optimization layer computes the optimal setpoints for the controlled variables, which are then implemented by the control layer. While the control layer is designed to keep the controlled variables at given setpoints, the optimization layer changes these setpoints to adapt operation optimally to varying conditions. For simple implementation, we want to change the setpoints only occasionally while still obtaining acceptable performance under varying disturbances.

The focus of this thesis is to study how to find good controlled variables, whose optimal value is invariant or near invariant to disturbances. These invariants are called *self-optimizing* variables, and keeping them constant will result in an acceptable, or in the ideal case, zero loss from optimality.

In the first part of this thesis, we consider controlled variables, which are linear combinations of measurements. The loss is used as the criterion for selecting the best set of controlled variables. Applying the inverse Choleski factor of the Hessian with respect to the inputs as a weighting factor, we derive a first order accurate expression of the loss in terms of the weighted square norm of the gradient of the optimization problem.

Next, we present a method for finding controlled variables by analyzing past optimal measurement data. Selecting combinations of measurements which correspond to directions of small singular values in the data, leads to controlled variables which mimic the original disturbance rejection.

Furthermore, the relationship between self-optimizing control and necessary conditions of optimality (NCO) tracking[1] is studied. We find the methods to be complementary, and propose to apply NCO tracking in the optimization layer, and self-optimizing control in the control layer. This will reject expected disturbances by self-optimizing control on a fast time scale, while unexpected disturbances are rejected by the setpoint updates from NCO tracking.

In the second part of the thesis, we extend the concept of self-optimizing control to polynomial systems with constraints. By virtue of the sparse resultant, we use the model equations to eliminate the unknown variables from the optimality conditions. This yields invariants which are polynomials in the measurements; controlling these invariants is equivalent to controlling the optimality conditions.

This procedure is not limited to steady state optimization, and therefore, we demonstrate that it can be used for finding invariants for polynomial input affine optimal control problems. Manipulating the inputs to control the invariant to zero gives optimal operation.

[1] François, G., Srinivasan, B., Bonvin, D. 2005. "Use of measurements for enforcing the necessary conditions of optimality in presence of constraints and uncertainty". Journal of Process Control 15 (6). 701-712

Acknowledgements

> Independence? That's middle class blasphemy. We are all dependent on one another, every soul of us on earth.
>
> G. B. Shaw

As most human endeavors, this work could not have been done without the help and support of many others. Therefore I would like to seize the opportunity to extend my gratitude for being part of such a wonderful human network.

First of all, I am grateful to Professor Sigurd Skogestad for giving me the opportunity to carry out my PhD studies under his supervision, and for sharing his engineering and control insight with me. This work could not have been done without his confidence in me. Moreover, I would like to thank him for many opportunities to meet leading scientists on interesting conferences.

I would also like to thank all my colleagues from the Process Systems Engineering group at NTNU. It has been inspiring to work with you. Special thanks go to Sridhar Narasimhan who introduced me to the world of polynomial systems, and to Håkon Dahl-Olsen, with whom I shared an office for several years.

Life has not always been easy during the last four years, and I am deeply indebted to my friends who supported me in tough times. Among them I would especially like to thank all members of our small group from Frikirke for being a family when we needed it most. I would like to mention Håvard Haugen and Troy Hawkins, who have been (and still are) role models for me. Further thanks go to Amanda Bjørge and Magnus Jacobsen for proof-reading some of this thesis.

My wonderful parents, siblings, and in-laws deserve special thanks for their continuous support and encouragement throughout my life. I owe very much of what I am today to you.

Finally, I would like to thank my son Samuel for his heart-warming smile, and my most lovely wife Susanne for all her support, love and patience. Sharing life with you is the greatest privilege I have.

J.E.P. J.

Contents

Chapter 1

Introduction

Und jedem Anfang wohnt ein Zauber inne,
Der uns beschützt und der uns hilft zu leben.

H. Hesse

1.1 Motivation

Rising competition in a global market makes it increasingly necessary to operate chemical plants at a greater profit. At the same time, environmental and safety regulations become stricter. To remain competitive in the face of these challenges requires improvements in the process design and strategies which help to maximize the profit for existing processes.

This work is concerned with optimal operation of process systems, where the design is assumed to be given. Although the profits from meliorating process operation might seem small compared to the total income, the accumulated effect over time is often significant for success on the market.

A common strategy for obtaining higher profits is to raise the degree of automation in a plant by using automatic controllers. These controllers are used to control process variables to their setpoints, which are given by operators or have been obtained by some optimization routine. Depending on the operating conditions and the control structure, these setpoints may have to be adjusted to the current operating conditions, such that the income generated by the plant, is maximized.

In practice, however, one wants to avoid changing the setpoints for the controlled variables frequently, to keep operation simple while still achieving optimal or near optimal operation.

A successful method for achieving near-optimal operation is self-optimizing control [Skogestad, 2000]. This term refers to finding controlled variables which remain constant at optimal operation, and which can be easily controlled using a simple feedback structure. An example for self-optimizing control is controlling the temperature in a kitchen oven at a given temperature. Baking your favourite cake in the oven at the given temperature gives good results, even for different ovens and if the kitchen temperature varies because a window has been opened. A different (non self-optimizing) strategy would be to directly control the heat input to the oven. Due to changing heat loss, the optimal value of this variable will not be constant with changing kitchen temperatures, or for different ovens. In addition, the second strategy is more difficult to implement, and requires knowledge about the kitchen temperature, the heat loss from the oven, and the heat capacity of the cake.

In other cases, the best strategy is to control some combination of measurements to a given constant setpoint, for example controlling the flow ratio of two streams entering a reactor, or the air/fuel ratio into a combustion engine.

Controlled measurement combinations, whose optimal values are invariant to disturbances, are the leitmotif in this thesis. We show how this concept can be derived, how data can be used for finding controlled variables, how using invariants and other methods interact, and we explore new ways for finding these invariants.

In the last decade, there has been much work on finding self-optimizing controlled variables, which are linear combinations of measurements. However, restricting controlled variables to be linear combinations of measurements can give unacceptable performance, if the process behaviour is nonlinear. In these cases it is desirable to allow nonlinear combinations of measurements, too. Thus, the main contribution of this thesis is the extension of the null-space method [Alstad and Skogestad, 2007], which gives a linear combination of measurements for an unconstrained problem to systems which are constrained and which can be described by polynomial or rational equations (Chapter 7). This extension to polynomial systems makes it possible to apply it to certain dynamic systems (Chapter 8).

1.2 Thesis overview

After an introduction to different methods for achieving optimal operation, the first part of this thesis (Chapters 2 – 5) deals with controlled variables, which are linear combinations of measurements. The second part (Chapters 6 – 8) describes how polynomial invariants can be found for systems which are modelled by rational or polynomial equations. Except for Chapter 8, all this thesis is concerned with finding controlled variables which optimize plant performance at steady state.

Part I Preliminaries and linear invariants for optimal operation

- Chapter 2 gives a short introduction to optimal operation strategies for chemical plants. Some important strategies are briefly mentioned, and the thesis is placed in a wider context.

- Chapter 3 points out the connection between the loss from optimality used in linear self-optimizing control, and the gradient of the underlying optimization problem. In particular, the loss is shown to be a weighted norm of the gradient. We show that weighting is important when comparing different control structures, because using the unweighted gradient norm as a criterion may be misleading.

- Chapter 4 presents a method for finding linear invariant variable combinations in the case where a model is not available. Optimal operation data is analyzed for finding invariant variable combinations which can be used for control.

- Chapter 5 shows how self-optimizing control and NCO[1] tracking [François et al., 2005] can be placed in the general concept of an hierarchical real-time optimization control structure. We show that the approaches can be considered complementary, with NCO tracking updating the setpoints for the self-optimizing controlled variables.

Part II Polynomial invariants for optimal operation

- Chapter 6 presents some very basic theory polynomial systems. We introduce some concepts to prepare the reader for the following chapters.

- Chapter 7 extends the theory of the null-space method [Alstad and Skogestad, 2007] to systems of polynomial equations with constraints. The idea is to formulate the optimality conditions, and then eliminate all unknown (unmeasured) variables from this expression.

- Chapter 8 demonstrates that the results from the previous chapter can be used to find invariants for a dynamic optimization.

- Chapter 9 gives our conclusions and suggestions for further work.

[1] NCO refers to “necessary conditions of optimality”

Appendix

- Appendix A contains a case study of a waste incineration plant. It is included here, because it nicely illustrates the concept of self-optimizing control in an industrial case study, and it shows how optimal operation can be implemented in a simple manner.

In the author's opinion, the Chapters 5, 7 and 8 are the most important contributions of this PhD work.

1.3 Publications

During my PhD work I have generated the following publications:

1.3.1 Publications contained in this thesis

- Chapter 3
  - Johannes Jäschke and Sigurd Skogestad “Optimal Operation by Controlling the Gradient to Zero” Accepted for publication at the 18th World Congress of the International Federation of Automatic Control (IFAC), 2011, Milano.
- Chapter 4
  - Johannes Jäschke and Sigurd Skogestad “Controlled Variables from Optimal Operation Data” Accepted for publication at the 21st European Symposium on Computer-Aided Process Engineering, ESCAPE 2011, Porto Carras.
- Chapter 5
  - Johannes Jäschke and Sigurd Skogestad “Self-optimizing Control and NCO tracking in the Context of Real-time Optimization” Submitted to Journal of Process Control.
- Chapter 7
  - Johannes Jäschke and Sigurd Skogestad “Self-Optimizing Control using Nonlinear Variable Combinations as Controlled Variables” Submitted to Journal of Process control.
- Chapter 8
  - Johannes Jäschke, Miroslav Fikar and Sigurd Skogestad “Self-optimizing Invariants in Dynamic Optimization” Submitted to 50th IEEE Conference on Decision and Control and European Control Conference, CDC-ECC 2011, Orlando.
- Appendix A
  - Johannes Jäschke, Helge Smedsrud, Sigurd Skogestad, Henrik Manum, “Optimal Operation of a Waste incineration Plant for District Heating” Proc. American Control Conference, St. Louis, USA, June 2009, 665-670.

1.3.2 Other publications as first author

2011

- Johannes Jäschke, Sigurd Skogestad "Measurement Polynomials as Controlled Variables" Book Chapter in: M. Huba, S. Skogestad, M. Fikar, M. Hovd, T. A. Johansen, B. Rohal'-Ilkiv (Editors) "Selected Topics on Constrained and Nonlinear Control", Textbook. ISBN: 978-80-968627-4-0.

- Johannes Jäschke, Sigurd Skogestad "Measurement Polynomials as Controlled Variables -Excercises" Book Chapter in: M. Huba, S. Skogestad, M. Fikar, M. Hovd, T. A. Johansen, B. Rohal'-Ilkiv (Editors) "Selected Topics on Constrained and Nonlinear Control", Workbook. ISBN: 978-80-968627-3-3.

2010

- Johannes Jäschke, Sigurd Skogestad, "Self-optimizing control and NCO tracking in the Context of Real-time optimization" Keynote lecture, 8th International Symposium on Dynamics and Control of Process Systems, Proc. DYCOPS 9 Leuven, Belgium, July 2010.

- Johannes Jäschke, Sigurd Skogestad "The Null Space Method for Finding Patterns from Optimal Data" 21st Norwegian Symposium on Chemometrics, March 2010 Sundvolden.

2009

- Johannes Jäschke, Sigurd Skogestad "Optimally Invariant Variable Combinations for Nonlinear Systems" Proc. ADCHEM, Istambul, Turkey, July 2009, 551-556.

- Johannes Jäschke Sigurd Skogestad, "Nonlinear Measurement Combinations for Optimal Operation" Nordic Process Control Workshop 2009 Porsgrunn, Norway.

2008

- Johannes Jäschke, Sridhar Narasimhan, Sigurd Skogestad, "Explicit real-time optimization", AIChE Annual Meeting, paper 471a, Philadelphia, USA, Nov. 2008.

1.3.3 Co-authored publications

- Sigurd Skogestad, Ramprasad Yelchuru, Johannes Jäschke “Optimal use of Measurements for Control, Optimization and Estimation using the Loss Method: Summary of Existing Results and Some New” Book Chapter in: M. Huba, S. Skogestad, M. Fikar, M. Hovd, T. A. Johansen, B. Rohal’-Ilkiv (Editors) “Selected Topics on Constrained and Nonlinear Control”, Workbook. ISBN: 978-80-968627-3-3.

- Henrik Manum, Sigurd Skogestad, Johannes Jäschke “Convex initialization of the H2 -optimal static output feedback problem” Proc. American Control Conference, St. Louis, USA, June 2009, 1724-1729.

References

Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. Industrial & Engineering Chemistry Research 46, 846–853.

François, G., Srinivasan, B., Bonvin, D., 2005. 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.

Skogestad, S., 2000. Plantwide control: The search for the self-optimizing control structure. Journal of Process Control 10, 487–507.

Part I

Preliminaries and linear invariants for optimal operation

Chapter 2

Optimal operation of chemical plants

> The fashion of the world is to avoid cost...
>
> W. Shakespeare

This chapter provides a short overview of different approaches to obtain optimal process operation. Some concepts will be revisited later in this thesis, however, it was decided to include them here to give an overview of how optimal operation can be implemented in chemical plants, and to present the larger setting in which this thesis is placed.

2.1 Optimal operation of process systems

Control means influencing an object to behave in a desired way [Findeisen et al., 1980], and in this thesis we desire to influence process plants such that the plant performance is optimized. Optimal plant performance here means to manipulate the available plant inputs in such a way that a scalar cost function is minimized (or equivalently, that the profit is maximized). This cost $J_{dynamic}$ is expressed as [Findeisen et al., 1980]

$$J_{dynamic} = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} j(t) \mathrm{d}t, \tag{2.1}$$

Here t_0 and t_f denote the starting and the final time of the operation period of interest, and $j(t)$ denotes the time varying integral cost. During operation, certain safety and environmental constraints have to be satisfied:

$$h(t) \leq 0. \tag{2.2}$$

An example for $J_{dynamic}$ could be the accumulated expenses for fuel or emissions over a year; and examples for operational constraints include the maximum temperature in a reactor or the minimum and maximum allowable level in a storage tank.

In practice, many plants are operating close to a steady state operating point for most of the time. In these cases, (2.1) can be approximated sufficiently well by introducing a quasi steady state assumption and using the inputs to optimize the steady state cost (rate),

$$J = j. \tag{2.3}$$

The type of plants, which are optimized in industry varies. Starting from small specialty chemical reactors, where one e. g. wants to avoid expensive off-spec product, the applications range to large-scale applications as optimizing the performance of complete large-scale plants, such as refineries. Thus, the number of measurements and inputs can vary significantly. A complete chemical plant may have hundreds to thousands of controlled variables [Harris et al., 1999; Trenchard and Boder, 2005], which have to be controlled by equally many inputs. In addition, there will be many variables which are not controlled, but which are measured and monitored for safety and environmental reasons.

The plant profit J is to be optimized not only at some design conditions, but also during different kinds of external disturbances which affect the plant. Generally, some disturbances will change slowly, or between long time intervals. Applying a quasi steady state assumption, these disturbances can be considered as constant. Examples for this kind of disturbances may include changes in the production rate, or the temperature of sea water which is used for cooling in a process.

Other disturbances will be changing at a higher frequency, and a steady state assumption would not be valid for these disturbances. To reject these disturbances truly optimally, one would need to solve a dynamic optimization problem. However, in many practical applications the disturbances can be rejected well with PI feedback control to keep the important process variables constant in spite of these disturbances. An example is the temperature of the cooling water for a cooling reactor. In this case, the coolant flow rate can be adjusted such that the reactor temperature stays at the steady state optimal value. However, in general, the question of whether a disturbance can be considered as constant (steady state) is dependent on the particular plant, and on the engineer's requirement to optimality.

In order to minimize the operating cost (2.1) or (2.3) for a complex system like a chemical plant in spite of disturbances, a common approach is to decompose

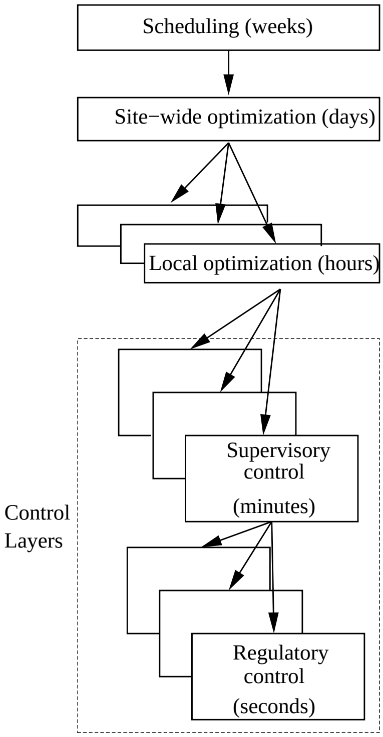


Figure 2.1: Hierarchical control structure [Skogestad, 2000a]

the system into subsystems, which are easier to manage, and can be considered individually. In terms of achieving optimal operation for a chemical plant, a hierarchical decomposition of the control structure is suggested, Figure 2.1 [Findeisen et al., 1980; Skogestad, 2000a], where one control layer receives control signals from the layer above, and passes new control signals to the layer below. The scheduling layer operates in the time scale of several weeks, and is based on the economic strategy of the company. Usually, this layer is not automated. The resulting strategy and operational targets are passed to the site-wide optimization layer, which may be automated, often using steady state models and numerical optimization. The control signals from the site-wide optimization layer are sent to the local optimization layer, which may also be automated. This layer operates in a timescale of hours.

Finally, the setpoints from the local optimization layer are passed to the control layer, which can be further subdivided vertically into the supervisory layer and the regulatory layer. The supervisory layer performs on a slower time scale, while the regulatory layer rejects the influence of disturbances on a fast time scale.

As with any abstraction, the concept of hierarchical layers does not reflect all issues which are present in a chemical plant. For example the time scale separation will not always be sharply defined. Depending on the particular plant in consideration and the degree of optimization of the process, not all layers are present in the control structure. However, elements of the layer concept will be found in most chemical plants, and for practical purposes, the layer model helps to organize the complex problem of operating the plant optimally.

For obtaining optimal operation in terms of the cost (2.1), obviously all layers have to act in concert. E.g. it does not help to have a perfectly working regulatory layer when the scheduling layer has forgotten to order the raw products. To achieve optimal plant operation, several approaches have been proposed in literature. In the next section we present some important concepts and relate them to the control layers in Figure 2.1.

2.2 Model based approaches

To optimize the plant performance, the real-world problem of optimizing (2.1) or (2.3) is typically translated into a mathematical optimization problem. Depending on whether a dynamic problem or a steady state problem is solved, we have two cases.

2.2.1 Dynamic real-time optimization (Dynamic RTO)

The in some sense simplest approach is to lump all the layers into a big, combined layer and simultaneously optimize this layer using a dynamic model to find the optimal input trajectories. The optimization problem is typically in a form similar to this,

$$\begin{aligned} \min_{\mathbf{u}(t)} J := \frac{1}{t_f - t_0} \int_{t_0}^{t_f} & j(\mathbf{u}(t), \mathbf{x}(t), \mathbf{d}(t), \mathbf{p}(t), t)\mathrm{d}t \\ \text{s.t.} & \\ \dot{x}(t) &= g\left(\mathbf{u}(t), \mathbf{x}(t), \mathbf{d}(t), t\right) \\ h(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) &\leq 0 \\ T\left(\mathbf{x}(t_f)\right) &\leq 0 \\ \mathbf{x}(0) &= \mathbf{x}_0, \end{aligned} \tag{2.4}$$

where $\mathbf{u}(t)$ is the $n_{\mathbf{u}}$-dimensional vector of inputs, $\mathbf{x}(t)$ the $n_{\mathbf{x}}$-dimensional state vector with given initial value $\mathbf{x}_0$, $\mathbf{d}(t)$ the $n_{\mathbf{d}}$-dimensional disturbance vector, and $\mathbf{p}(t)$ denotes known parameters concerning the prices of the products.

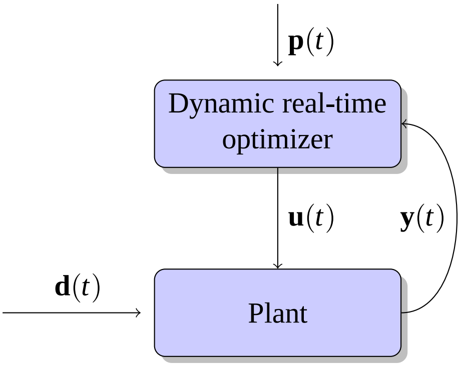


Figure 2.2: Dynamic RTO scheme. $\mathbf{d}(t)$: external disturbance, $\mathbf{p}(t)$: prices, $\mathbf{u}(t)$: input, $\mathbf{y}(t)$: measurements.

The function j is the integral cost term of the scalar cost function J, g is the right-hand side of the differential equation system which defines the state evolution, h denotes the operating constraints, and T the terminal constraints on the states.

Because the model generally does not describe the plant perfectly, and because the future evolution of the disturbance is seldom known, the optimization problem (2.4) is usually solved repeatedly at given sample times, and the input trajectory is updated accordingly, Figure 2.2. The plant measurements $\mathbf{y}$ are primarily used to update the model and model parameters. This monolithic approach is referred to as dynamic real-time optimization [Allgöwer and Zheng, 2000; Diehl et al., 2002; Grötschel et al., 2001; Engell, 2007], and it requires an accurate model, which is capable of describing every layer of Figure 2.1 in detail.

In dynamic real-time optimization there is no distinction between optimization and control. Thus, the time scale separation from Figure 2.1 is effectively undone. If the model was perfect, this approach would lead to the true optimum. Because of this potential, dynamic real-time optimization has gained much interest in academia, but it is not used so frequently in practice, because of several reasons. First, obtaining an accurate dynamic model for a complete chemical plant is often prohibitively expensive.

Second, solving the dynamic optimization problem arising from a complete plant in real-time is still challenging with today's computing power.

Third, since optimization and control are not distinguishable, a failure in the optimization routine will result in arbitrary inputs to the plant. Vice versa, a small failure, such as unmodelled stiction in a valve can lead to unforeseen upsetting of otherwise unrelated parts of the plant. Moreover, noise and plant-model mismatch may make operation infeasible.

Fourth, it is very difficult to include unforeseen operator interaction in the routine. If e.g. an operator sets a valve in manual mode, it is no longer available for the optimizer. It is almost impossible to include all possibilities when designing the dynamic real-time optimizer for an industrial plant.

Fifth, the optimality and feasibility of dynamic real-time optimization in practical applications depends critically on obtaining good state and parameter estimates for the model. As the estimation problem is equally complex as the dynamic optimization problem, this concept is not widely used in industry.

However, if the model is accurate and the optimization problem can be solved in real-time, the concept of optimizing the whole plant simultaneously will be optimal. Despite the challenges listed above, dynamic real-time optimization has been reported on industrial cases [Qin and Badgwell, 2000; Foss and Schei, 2007]. Mostly it is used for optimizing processes of inherently dynamic nature, such as batch processes and grade transitions. With ever growing computing power and modelling knowhow it will certainly continue to spread in industry.

2.2.2 Conventional real-time optimization (RTO)

Since many continuous processes make most profit at steady state, the largest savings can be made by optimizing the steady state performance of the plant. Optimizing only the steady state of the process, instead of including all dynamics into the optimization problem, allows us to consider the optimization problem and the control problem separately. This is done in conventional or steady state real-time optimization (RTO), where the steady state optimization problem is solved on a time scale which is larger than the plant settling time.

Of course, a chemical plant is never truly at steady state, but in many cases, disturbances can be rejected such that important variables remain almost constant at their optimal values. This is the ultimate reason for the existence of process control.

Assuming quasi steady state in all variables, the optimization problem typically looks like

$$\begin{aligned} \min_{\mathbf{c}_s} J(\mathbf{u},\mathbf{x},\mathbf{d},\mathbf{p}) & \\ \text{subject to} & \\ g(\mathbf{u},\mathbf{x},\mathbf{d}) = 0 & \\ \mathbf{u} = f(\mathbf{c}_s,\mathbf{x},\mathbf{d}) & \\ h(\mathbf{u},\mathbf{x},\mathbf{d}) \leq 0, & \end{aligned} \tag{2.5}$$

where $\mathbf{c}_s$ is the setpoint for the controlled variables, and the variables $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$, $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$, $\mathbf{d} \in \mathbb{R}^{n_\mathbf{d}}$, $\mathbf{p} \in \mathbb{R}^{n_\mathbf{p}}$ are the quasi steady state input, state, disturbance and

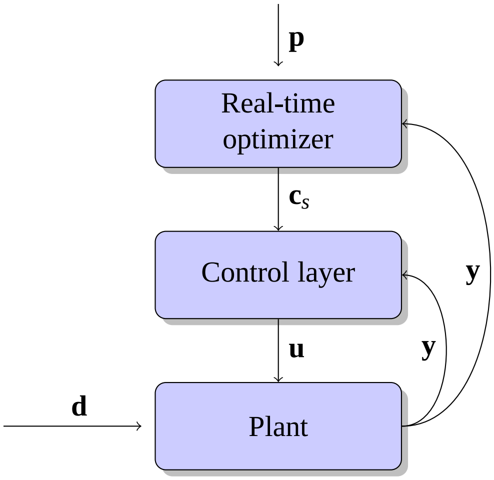


Figure 2.3: Conventional RTO scheme. **d**: external disturbance, **p**: prices, **u**: input, **y**: measurements.

price vectors, respectively. J denotes the scalar cost function to be minimized, g the model equations, and h the operational constraints. The function f describes the closed loop steady state relationship between the controlled variables and the inputs.

The conventional RTO scheme is illustrated in Figure 2.3. Using the price parameters (from the scheduling layer above), the new setpoints for the controlled layer are calculated once or twice a day and passed on to the control layer below which controls the controlled variables to the given setpoints.

Note that in practice all variables in Figure 2.3 are time varying. However, the real-time optimizer computes new setpoints only at discrete sample times, and optimal operation will only be achieved when the plant has settled down close to steady state, and the quasi steady state assumption is valid.

Implementing conventional RTO is easier to realize in industrial practice, because it is generally easier to obtain a good steady state model than building an exact dynamic model, which describes the process sufficiently well at all frequencies. In addition, it is numerically easier to solve large steady state optimization problems than large dynamic optimization problems. However, it should be noted that solving large nonlinear steady state optimization problems is still far from easy, and can be prohibitive for implementing RTO in some processes.

As in dynamic real-time optimization, when using conventional real-time optimization, a major challenge is to obtain good parameter and state estimates for the model. Since a steady state model is used for optimization, it is necessary to assure that the plant has settled down sufficiently before updating the model parameters and calculating the new inputs. The task of identifying steady state and reconciling the data with the model is not easy, and still a subject of ongoing research [Lid and Skogestad, 2008; Martinez et al., 2010].

When using RTO, some loss for the non steady state periods is accepted, but it is traded for the additional freedom of being able to design the control layer independently. Hence, we can for example select pairings which give good dynamic performance, and we can assign different controllers for different tasks. Thus, it is possible to design a simple and robust control structure in the control layer, while still achieving optimal operation at steady state. The additional savings which could be obtained using dynamic real-time optimization often do not justify the increased effort of installing and maintaining the dynamic real-time system.

2.3 Control structure design – self-optimizing control

A question which remains unsolved in this layer system (Figures 2.1 and 2.3) is: Which variables should be passed on from one layer to the next? Since e.g. the optimization layer operates on a slow time scale where the setpoints are changed only once or twice a day, disturbances which occur between the updates are not rejected optimally before the next update. Moreover, disturbances may change their values more frequently than the optimization layer updates the setpoints. Depending on the choice of the controlled variables, this may have a severe impact on the overall profitability of the plant. The question of which variables to control has been raised by Foss [1973], in his “Critique of chemical process control theory”:

> Which variables should be measured, which inputs should be manipulated, and what links should be made between these two sets? This problem is considered by many to be the most important problem encountered by designers of chemical process control systems. [Foss, 1973]

Since then many authors have been working on finding the best control structure [Morari et al., 1980; Morari, 1982; Skogestad, 2000a; Narraway and Perkins, 1994; Halvorsen et al., 2003; Skogestad, 2004; Cao, 2005; Kariwala and Cao, 2009], and also this thesis is a contribution to this research field.

To keep the time scale separation, it is desirable to find controlled variables which remain at a constant value whenever the system is operated optimally. These variables are called self-optimizing variables. Skogestad [2000a] writes:

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

The idea is to keep the overall optimization problem in mind when designing the control structure in each layer, such that information about optimal operation is contained in the controlled variables **c**. By controlling **c** to its optimal setpoint, the loss which comes from the time scale separation is minimized, because small disturbances are rejected locally on the fast time scale by the control layer. It is no longer necessary to wait for the RTO layer to compute new optimal setpoints, because the optimal setpoints do not change with varying disturbances. Therefore the self-optimizing variables have been referred to as the "missing link between steady state optimization and control" Skogestad [2000b].

For many processes, operation can be improved significantly by automatically rejecting the disturbances in the control layer, and not having to wait for the next RTO setpoint update.

In Chapter 3 we show that self-optimizing control variables can be considered as local approximations of the gradient of the economic optimization problem. If they are controlled to zero, the system is operated optimally. Hence, a good choice of controlled variables can disburden the real-time optimizer in the sense that the optimization problem has to be solved less frequently, while still giving good performance. In some cases, the real-time optimizer may even become superfluous. We discuss the combination of self-optimizing control and an optimization layer in Chapter 5.

2.3.1 Previous work on self-optimizing control

In this section we give a brief overview of previous work in self-optimizing control. Since this thesis is a collection of papers, we believe, that it will make it easier for the reader to see the novel developments in the subsequent chapters.

For a more detailed treatment and a complete derivation of the methods below, we refer to Halvorsen et al. [2003]; Alstad and Skogestad [2007]; Alstad et al. [2009]. The goal is to find controlled variables, whose steady-state optimal value is constant in spite of disturbances. We assume that quasi steady state optimal operation corresponds to the solution of

$$\begin{aligned} &\min_{\mathbf{u}} J(\mathbf{u},\mathbf{x},\mathbf{d},\mathbf{p}) \\ &\qquad \text{subject to} \\ &\qquad g(\mathbf{u},\mathbf{x},\mathbf{d}) = 0 \\ &\qquad h(\mathbf{u}.\mathbf{x},\mathbf{d}) \leq 0, \end{aligned} \tag{2.6}$$

where the variables are defined as in the conventional RTO problem (2.5). After satisfying all the active constraints, the remaining unconstrained problem can be approximated around the nominal optimal point as

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) = \left[\begin{array}{cc} \mathbf{u}^{\mathbf{T}} & \mathbf{d}^{\mathbf{T}} \end{array} \right] \left[\begin{array}{cc} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{array} \right] \left[\begin{array}{c} \mathbf{u} \\ \mathbf{d} \end{array} \right]. \tag{2.7}$$

Here $\mathbf{u} \in \mathbb{R}^{n_{\mathbf{u}}}$ and $\mathbf{d} \in \mathbb{R}^{n_{\mathbf{d}}}$ are the unconstrained inputs and the disturbance vector, respectively, and $\mathbf{J_{uu}}$, $\mathbf{J_{ud}}$, $\mathbf{J_{du}}$, $\mathbf{J_{dd}}$ are matrices of appropriate dimensions. Furthermore we require that $\mathbf{J_{uu}} > 0$. We assume that we have a linear plant model

$$\mathbf{y} = \mathbf{G}^{\mathbf{y}}\mathbf{u} + \mathbf{G}_{\mathbf{d}}^{\mathbf{y}}\mathbf{d}, \tag{2.8}$$

where $\mathbf{y} \in \mathbb{R}^{n_{\mathbf{y}}}$ are the measurements, and $\mathbf{G}^{\mathbf{y}}$ and $\mathbf{G}_{\mathbf{d}}^{\mathbf{y}}$ are matrices of appropriate sizes, which denote the steady state gain from the input and the disturbances, respectively, to the measurements.

In addition, we assume additive noise on the measurements which is given by the $n_{\mathbf{y}}$-dimensional vector $\mathbf{n}^{\mathbf{y}}$. Thus, the measured value of the plant output is

$$\mathbf{y}_m = \mathbf{y} + \mathbf{n}^{\mathbf{y}}. \tag{2.9}$$

It is assumed, that all uncertainty is captured in the vectors $\mathbf{d}$ and $\mathbf{n}^{\mathbf{y}}$.

Using the measurements, we want to select controlled variables of the form

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m \tag{2.10}$$

which give good steady state performance. In the case of no measurement noise, $\mathbf{n}^{\mathbf{y}} = 0$, we simply write the controlled variable as

$$\mathbf{c} = \mathbf{H}\mathbf{y}. \tag{2.11}$$

The criterion for evaluating the controlled variables is the economic loss in terms of the cost function.

$$L = J(\mathbf{u},\mathbf{d}) - J^{opt}(\mathbf{d}) \tag{2.12}$$

We do not write the loss as a function of the control structure, but as a function of the actual steady state inputs, which may be generated by either a control structure or an open loop policy.

There are two subproblems, which can be addressed in the framework of self-optimizing control. Assuming a set $\mathscr{D}$ of possible disturbances $\mathbf{d}$, we can:

1. Determine a $\mathbf{H}$ which minimizes the worst case (or average) loss L for all $\mathbf{d} \in \mathscr{D}$.

2. Given different control structures (given by $\mathbf{H}_1, \mathbf{H}_2, \ldots$) select the $\mathbf{H}_i$ which for all $\mathbf{d} \in \mathscr{D}$ gives the best performance in terms of the loss L.

Minimum singular value rule

The minimum singular value rule is a criterion for the selection of controlled variables, which is based on the scaled steady state input-output gain. We just give the result here, for details and the derivation, we refer to Halvorsen et al. [2003]; Skogestad and Postlethwaite [1996]. Assuming that we have different candidates (that is given $\mathbf{H}_i$) for controlled variables

$$\begin{aligned}\mathbf{c}_i &= \mathbf{H}_i\mathbf{y}_m \\ &= \mathbf{H}_i(\mathbf{y}+\mathbf{n}^{\mathbf{y}}) \\ &= \underbrace{\mathbf{H}_i\mathbf{G}^{\mathbf{y}}}_{=\mathbf{G}_i}\mathbf{u}+\underbrace{\mathbf{H}_i\mathbf{G}^{\mathbf{y}}_{\mathbf{d}}}_{=\mathbf{G}_{\mathbf{d},i}}\mathbf{d}+\mathbf{H}\mathbf{n}^{\mathbf{y}} \\ &= \mathbf{G}_i\mathbf{u}+\mathbf{G}_{\mathbf{d},i}\mathbf{d}+\mathbf{H}\mathbf{n}^{\mathbf{y}}.\end{aligned} \tag{2.13}$$

- Each candidate controlled variable c_i is scaled such that the sum of the optimal range and the implementation error is unity. Alternatively, we could scale the candidate variables $\mathbf{c}_i$ such that

$$\left|\left|\mathbf{c}_i(\mathbf{d})' - \mathbf{c}_i^{opt\,'}\right|\right|_2 \leq 1 \text{ for } \mathbf{d} \in \mathscr{D}. \tag{2.14}$$

- Each input vector element u_j is scaled such that a unit change in each input has the same effect on the cost function J.

The scaled candidate then is

$$\mathbf{c}_i' = \mathbf{G}_i'\mathbf{u}' + \mathbf{G}_{\mathbf{d},i}'\mathbf{d} + e_n, \tag{2.15}$$

where e_n is the term caused by the noise $\mathbf{n}^{\mathbf{y}}$. The minimum singular value rule suggests to select the controlled variable $\mathbf{c}_i$ which maximizes the minimum singular value of $\mathbf{G}_i'$,

$$\max \underline{\sigma}(\mathbf{G}_i'). \tag{2.16}$$

For most practical applications, this rule gives good results. However, since it is based on the assumption that $\mathbf{J}_{\mathbf{uu}}$ is orthogonal, it can yield poor results in ill-conditioned cases, see Halvorsen et al. [2003]. The minimum singular value method has been applied to many case studies, e. g. Govatsmark [2003], Skogestad [2000a], and references therein.

Exact local method

The exact local method [Halvorsen et al., 2003; Alstad et al., 2009] is a method for evaluating the loss caused by a given control policy. It is based on a Taylor approximation of the cost function. First, the measurement noise and the disturbances are scaled by diagonal scaling matrices $\mathbf{W_d}$ and $\mathbf{W_n}$ of appropriate sizes such that

$$\begin{aligned} \mathbf{d} &= \mathbf{W_d d'} \qquad (2.17)\\ \mathbf{n^y} &= \mathbf{W_{n^y} n^{y\prime}}, \qquad (2.18) \end{aligned}$$

and

$$\left|\left|\begin{bmatrix} \mathbf{d'} \\ \mathbf{n^{y\prime}} \end{bmatrix}\right|\right|_2 \leq 1. \qquad (2.19)$$

The loss L associated with a given control structure $\mathbf{H}$ can be written as

$$L = \frac{1}{2}\,||\mathbf{z}||_2^2 \qquad (2.20)$$

where

$$\mathbf{z} = \mathbf{M_d d'} + \mathbf{M_{n^y} n^{y\prime}} \qquad (2.21)$$

and

$$\begin{aligned} \mathbf{M_d} &= -\mathbf{J_{uu}}^{1/2}(\mathbf{HG^y})^{-1}\mathbf{HFW_d} \qquad (2.22)\\ \mathbf{M_{n^y}} &= -\mathbf{J_{uu}}^{-1/2}(\mathbf{HG^y})^{-1}\mathbf{HW_{n^y}}. \qquad (2.23) \end{aligned}$$

Here $\mathbf{F}$ is the optimal sensitivity matrix, which is defined as

$$\mathbf{F} = \frac{\partial \mathbf{y}^{\text{opt}}}{\partial \mathbf{d}}. \qquad (2.24)$$

By reopimizing a process model for different disturbance values and using finite differences $\mathbf{F}$ can be estimated. Alternatively, it can be calculated from the linearized model using

$$\mathbf{F} = \mathbf{G^y_d} - \mathbf{G^y J_{uu}}^{-1}\mathbf{J_{ud}}. \qquad (2.25)$$

Introducing

$$\mathbf{M} = [\mathbf{M_d} \quad \mathbf{M_{n^y}}], \qquad (2.26)$$

we can write

$$\mathbf{z} = \mathbf{M}\begin{bmatrix} \mathbf{d'} \\ \mathbf{n^{y\prime}} \end{bmatrix}, \qquad (2.27)$$

and the worst case loss is given by

$$\begin{aligned} L_{\text{wc}} &= \frac{1}{2} \max_{\left[\begin{smallmatrix} \mathbf{d'} \\ \mathbf{n^{y\prime}} \end{smallmatrix}\right] \leq 1} ||\mathbf{z}||_2^2 \qquad (2.28)\\ &= \frac{1}{2}\bar{\sigma}(\mathbf{M})^2. \end{aligned}$$

This loss can be used to rank different candidates for controlled variables. Alternatively, as we show in the next section, the loss expression may be used to determine a combination of measurements which minimizes the worst case loss L_{wc}.

Minimum loss method

The optimal linear combination of measurements $\mathbf{c} = \mathbf{H}\mathbf{y}_m$ can be found by minimizing the worst case loss (2.28):

$$\mathbf{H} = \arg\min_{\mathbf{H}} \bar{\sigma}(\mathbf{M}) \tag{2.29}$$

However, this is not an easy problem, because (2.29) is a nonconvex optimization problem. Introducing

$$\tilde{\mathbf{F}} = [\mathbf{F}\mathbf{W}_{\mathbf{d}} \quad \mathbf{W}_{\mathbf{n}^\mathbf{y}}]\,, \tag{2.30}$$

Alstad et al. [2009] present a method, which solves (2.29) by transforming it into a convex optimization problem. In particular, they show that

$$\mathbf{H} = \arg\left(\min_{\mathbf{H}} \bar{\sigma}(\mathbf{H}\tilde{\mathbf{F}}) \text{ s.t. } \mathbf{H}\mathbf{G}^\mathbf{y} = \mathbf{Q}\right), \tag{2.31}$$

where $\mathbf{Q} \in \mathbb{R}^{n_\mathbf{u} \times n_\mathbf{u}}$ is any nonsingular matrix, solves (2.29). Kariwala et al. [2008] have shown that

$$\begin{aligned} \min_{\mathbf{H}} \left|\left|\mathbf{H}\tilde{\mathbf{F}}\right|\right|_\mathrm{F} \\ \text{s.t.} \\ \mathbf{H}\mathbf{G}^\mathbf{y} = \mathbf{Q} \end{aligned} \tag{2.32}$$

gives solution, which minimizes both, the average and the worst case loss. Here $||\cdot||_\mathrm{F}$ denotes the Frobenius norm. We refer to Kariwala et al. [2008] and Alstad et al. [2009] for a deeper discussion on the choice of the norm and more details.

Null space method

The null-space method is a special case of the minimum loss method, where there is no noise in the measurements, $\mathbf{n}^\mathbf{y} = 0$. Then (2.26) simplifies to

$$\mathbf{M} = \mathbf{M}_\mathbf{d}. \tag{2.33}$$

In addition, we must have

$$n_\mathbf{y} \geq n_\mathbf{d} + n_\mathbf{u}. \tag{2.34}$$

That is, we have at least one independent measurement for each unknown variable[1] In this case we have

$$\tilde{\mathbf{F}} = \mathbf{F}\mathbf{W}_\mathbf{d}, \tag{2.35}$$

[1] As a starting point, **u** is considered an unknown variable. If **u**, or elements in **u** are measured (known), they are included in the measurement vector **y**. Thus they appear on both sides of the inequality, and (2.34) can be restated to read that we need at least as many measurements as unknown variables.

and it is always possible to find a nontrivial $\mathbf{H}$ such that $\mathbf{HF} = 0$. We simply select $\mathbf{H}$ in the left null-space of $\mathbf{F}$. Then we see from (2.22) that

$$\mathbf{M_d} = 0, \tag{2.36}$$

and the loss $\bar{\sigma}(\mathbf{M})$ is zero. As we have assumed zero noise, $\mathbf{n^y} = 0$, the null-space method rejects disturbances perfectly, but does not take into account the effect of measurement noise $\mathbf{n^y}$.

Some personal comments on the previous work

The starting point in the research of self-optimizing control is the desire to find a control structure which minimizes the cost (or equivalently the loss L from optimality). It was soon discovered that minimizing the norm of the gradient was not equal to minimizing the loss, therefore the gradient norm was not considered a good criterion and the research focus moved away from the gradient as a criterion. However, the author suspected that there was some connection between minimizing the gradient and the loss. In Chapter 3 this connection is explored, and it is shown that the loss can be expressed as the norm of the weighted gradient, where $\mathbf{J_{uu}}^{-1/2}$ is the weighting factor. Naturally, when the loss is zero, the gradient must be zero, too. Therefore, the controlled variable obtained from the null-space method can be considered as the gradient, where we have eliminated all unmeasured variables by using the measurement model. Obviously, for this elimination, we need at least as many measurement equations as we have unknowns. This is reflected in the condition for the number of measurements $n_\mathbf{y} \geq n_\mathbf{d} + n_\mathbf{u}$.

Using tools from polynomial elimination theory, this concept could be extended systematically to polynomial systems in Chapter 7, and to dynamic optimization problems which are described by polynomial equations (Chapter 8).

Originally, the optimal sensitivity matrix $\mathbf{F}$ had to be estimated using a process model and re-optimization, or by evaluating (2.25). Since this requires a good process model, this is often difficult. In Chapter 4 we show how plant data can be used to obtain an estimate of $\mathbf{F}$.

2.4 Uncertainty in model based approaches

Whenever a model is used for describing and optimizing a process, there will be some uncertainty in the model which leads to a discrepancy between the model and the reality. The uncertainty can be classified into different types, which usually all are present to some extent. This uncertainty will cause plant-model mismatch.

The first type is parametric uncertainty, which occurs when the model is correct, but some parameters are not known. Consider for example the heat $\dot{Q}$ which is removed from a stream,

$$\dot{Q} = \dot{m} c_p \Delta T, \tag{2.37}$$

where $\dot{m}$, c_p, ΔT denotes the mass flow, the specific heat capacity, and the temperature difference between inlet and outlet, respectively. An example for parametric uncertainty (a disturbance) is a varying c_p value. Different c_p values will lead to different amounts of removed heat $\dot{Q}$.

Another example for a parametric uncertainty could be an unmeasured state. Even though the model describes the state correctly, its value is not known, and any expression containing it cannot be evaluated.

Parametric uncertainty can often be handled by using other measurements to infer the unmeasured quantity. This thesis contributes to the problem of handling parametric uncertainty, by showing how uncertain variables can be eliminated using a process model.

A second type of uncertainty is structural model mismatch. This means that the model equations do not describe the process correctly. For example, a reaction rate r may have been modelled to follow a first order law $r = kc$, where k is a constant and c is a concentration. In the real process, however, the reaction may follow a second order law, $r = kc^2$, or some more complicated kinetics. This kind of uncertainty tends to have a more severe impact on nonlinear models.

Many different strategies have been developed to handle plant-model mismatch. Maybe the simplest way to handle this kind of uncertainty is to simply add a bias term to the model, and to adapt the term such that the model describes the reality better. For a more advanced treatment of this subject we refer to Marchetti et al. [2010] and references herein. Other references on this subject are Forbes et al. [1994]; Forbes and Marlin [1996]; Zhang and Forbes [2000]; Chachuat et al. [2009].

In many cases the effect of the uncertainty on the plant profit is significant; in other cases the effect is negligible. This depends very much on the combination of plant, model, the particular uncertainty and the how the model is used.

2.5 Other related concepts

We briefly discuss some other optimization concepts in this section, and set them into the context to this work.

2.5.1 Model predictive control

In model predictive control [García et al., 1989; Mayne et al., 2000; Maciejowski, 2002; Rawlings and Mayne, 2009], the controlled variables are controlled to their setpoints by using an input trajectory which has been obtained by solving an optimization problem online.

The term “model predictive controller” is generally used for controllers, where the input trajectories have been obtained by solving a quadratic program (QP); that is, the objective function is quadratic and the process model is linear. If a nonlinear model is used, it is referred to as “nonlinear model predictive control” or NMPC. The only difference between NMPC and dynamic real-time optimization (Section 2.2.1) is that in the former the objective is to control a set of variables to their setpoints, while in the latter the objective is to minimize a cost function of economic nature.

The concept of using a model for calculating the optimal input trajectories was first reported by Richalet et al. [1978], and has gained much attention in the control community since then. A nice survey about industrial use of model predictive control is given in Qin and Badgwell [2003].

Due to the task of tracking setpoints, model predictive controllers are used in the control layer of Figure 2.1. The question about which variables to use as controlled variables, still remains open and needs to be answered.

2.5.2 NCO-tracking

Necessary conditions of optimality (NCO) tracking is the idea to control the necessary conditions of optimality [Srinivasan et al., 2003; Kadam et al., 2007; François et al., 2005]. Although the NCO could be controlled by some continuous controller, such as a PI controller if they were available as online measurements, the publications concerning steady state optimization (e.g. [François et al., 2005; Gros et al., 2009]) have been applying iterative input updates, which are shown to converge to the optimal steady state operating point.

As in numerical optimization, also in real plants, obtaining the gradients efficiently is a challenge, because the gradients (and hence the NCO) cannot usually be measured directly. Therefore they must be estimated or approximated somehow. A comparison of methods for obtaining the gradients is given in Mansour and Ellis [2003]. More on NCO tracking and how it can be used together with self-optimizing control is found in Chapter 5.

NCO tracking is very close to self-optimizing control, since the necessary conditions of optimality for steady state optimization have a constant optimal setpoint. In fact, the NCO are the ideal self-optimizing variables; unfortunately they are often difficult to measure.

2.5.3 Experimental methods

Box [1957] proposed a procedure which is based on carefully designed experiments, and measurements of the objective function. The experiments are designed such that gradient information of the profit is estimated. Using this gradient information, new experiments are performed. This is continued until the optimum has been reached (the gradient is zero). However, this approach requires the objective function to be measurable, and if the disturbances change, the whole procedure has to be repeated. If the disturbance changes with a higher frequency than the experimental procedure, the gradient information is confounded with the disturbance, and does not give correct information about the profit surface. However, for systems with (constant) parametric uncertainties, this method may be applied successfully. Since the number of degrees of freedom (the variables which can be manipulated to optimize the process) has to be small for the problem to be tractable, this method can only be applied for small systems. More on this kind of methods can be found in e.g. Hunter [1960]; Carpenter and Sweeny [1965]; Box and Draper [1987], and Box and Draper [1998].

2.5.4 Extremum seeking control

An approach which automatically drives the process to the optimum is "Extremum seeking control", where the inputs are excited to obtain gradient information and this information is used to move the system to the optimum. This approach is based on measurement of the objective function, or on knowledge about how the objective function depends on the states and the disturbances [Guay and Zhang, 2003]. A nice introduction to this method is given in Ariyur and Krstic [2003].

This method could be used in the optimization layer, however, due to the required excitation signal, the process is constantly disturbed. This is generally in conflict with the desire to keep the process at the steady state optimal operating point, and it might cause problems in other down stream processes.

2.6 Conclusion

We have briefly presented some concepts for achieving optimal operation in a chemical plant, and have outlined some strategies for achieving optimal operation in continuous plants.

One concept is to use dynamic real time optimization, and to solve a dynamic optimization problem for the optimal inputs. This approach is the simplest in concept, however in practice it is the most difficult to realize due to the complexity of real industrial plants. In particular, problems arise because of modelling, numerical, and practical issues. Therefore it is not very commonly used in industry.

The second approach is to decompose the control system vertically based on the different timescales the layers operate in. This decomposition approach makes it possible to consider one layer at a time and to engineer and optimize the layers separately. It facilitates handling the complexity, because instead of considering the whole control structure, we consider one layer at a time. Thus it adds clarity about what is happening in the process, and the reduced complexity makes it easier to include possible faults and eventualities in the layer design.

The layers communicate downwards via the setpoints of the controlled variables, and upwards via the measurements. By selecting self-optimizing controlled variables, we guarantee that the control layers act in accord to minimize the operational cost. This will improve plant performance, while at the same time keeping the control structure simple and manageable. In some cases, when good self-optimizing controlled variables are used, an RTO layer may even become unnecessary.

References

Allgöwer, F., Zheng, A. (Eds.), 2000. Nonlinear Model Predictive control. Birkhäuser Verlag Basel.

Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. Industrial & Engineering Chemistry Research 46, 846–853.

Alstad, V., Skogestad, S., Hori, E. S., 2009. Optimal measurement combinations as controlled variables. Journal of Process Control 19 (1), 138–148.

Ariyur, K. B., Krstic, M., 2003. Real-Time Optimization by Extremum-Seeking Control. Wiley-Interscience.

Box, G. E. P., 1957. Evolutionary operation: A method for increasing industrial productivity. Journal of the Royal Statistical Society. Series C (Applied Statistics) 6 (2), 81–101.

Box, G. E. P., Draper, N. R., 1987. Empirical Model-building and Response Surfaces. John Wiley, New York.

Box, G. E. P., Draper, N. R., 1998. Evolutionary Operation. A Statistical Method for Process Improvement. John Wiley & Sons.

Cao, Y., 2005. Direct and indirect gradient control for static optimisation. International Journal of Automation and Computing 2, 60–66.

Carpenter, B., Sweeny, H., July 5 1965. Process improvement with ‘simplex’ self-directing evolutionary operation. Chemical Engineering, 117–126Fra Thz.

Chachuat, B., Srinivasan, B., Bonvin, D., 2009. Adaptation strategies for real-time optimization. Computers & Chemical Engineering 33 (10), 1557 – 1567.

Diehl, M., Bock, H. G., Schlöder, J. P., Findeisen, R., Nagy, Z., Allgöwer, F., 2002. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. Journal of Process Control 12 (4), 577 – 585.

Engell, S., 2007. Feedback control for optimal process operation. Journal of Process Control 17 (3), 203 – 219.

Findeisen, W., Nailey, F., Brdys, M., Malinowski, K., Tatjewski, P., Wozniak, A., 1980. Control and Coordination in Hierachical Systems. John Wiley & Sons.

Forbes, J., Marlin, T., MacGregor, J., 1994. Model adequacy requirements for optimizing plant operations. Computers & Chemical Engineering 18 (6), 497 – 510, an International Journal of Computer Applications in Chemical Engineering.

Forbes, J. F., Marlin, T. E., 1996. Design cost: a systematic approach to technology selection for model-based real-time optimization systems. Computers & Chemical Engineering 20 (6-7), 717 – 734, fifth International Symposium on Process Systems Engineering.

Foss, A. S., 1973. Critique of chemical process control theory. AIChE Journal 19 (2), 209–214.

Foss, B. A., Schei, T. S., 2007. Putting nonlinear model predictive control into use. In: Findeisen, R., Allgöwer, F., Biegler, L. (Eds.), Assessment and Future Directions of Nonlinear Model Predictive Control. Vol. 358 of Lecture Notes in Control and Information Sciences. Springer Berlin / Heidelberg, pp. 407–417.

François, G., Srinivasan, B., Bonvin, D., 2005. 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.

García, C. E., Prett, D. M., Morari, M., 1989. Model predictive control: Theory and practice–a survey. Automatica 25 (3), 335 – 348.

Govatsmark, M., 2003. Integrated optimization and control. Ph.D. thesis, Norwegian University of Science and Technology, NTNU.

Gros, S., Srinivasan, B., Bonvin, D., 2009. Optimizing control based on output feedback. Computers & Chemical Engineering 33 (1), 191 – 198.

Grötschel, M., Krumke, S. O., Rambau, J. (Eds.), Sep. 2001. Online Optimization of Large Scale Systems. Springer.

Guay, M., Zhang, T., 2003. Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties. Automatica 39 (7), 1283 – 1293.

Halvorsen, I. J., Skogestad, S., Morud, J. C., Alstad, V., 2003. Optimal selection of controlled variables. Industrial & Engineering Chemistry Research 42 (14), 3273–3284.

Harris, T. J., Seppala, C. T., Desborough, L. D., 1999. A review of performance monitoring and assessment techniques for univariate and multivariate control systems. Journal of Process Control 9 (1), 1 – 17.

Hunter, J. S., September 19 1960. Optimize your chemical process with evolutionary operations. Chemical Engineering, 193–202Fra Thz.

Kadam, J. V., Marquardt, W., Srinivasan, B., Bonvin, D., 2007. Optimal grade transition in industrial polymerization processes via nco tracking. AIChE Journal 53 (3), 627–639.

Kariwala, V., Cao, Y., 2009. Bidirectional branch and bound for controlled variable selection. part ii: Exact local method for self-optimizing control. Computers & Chemical Engineering 33 (8), 1402 – 1412.

Kariwala, V., Cao, Y., Janardhanan, S., 2008. Local self-optimizing control with average loss minimization. Industrial & Engineering Chemistry Research 47, 1150–1158.

Lid, T., Skogestad, S., 2008. Data reconciliation and optimal operation of a catalytic naphtha reformer. Journal of Process Control 18 (3-4), 320 – 331, festschrift honouring Professor Dale Seborg.

Maciejowski, J. M., 2002. Predictive Control: with constraints. Prentice Hall.

Mansour, M., Ellis, J. E., 2003. Comparison of methods for estimating real process derivatives in on-line optimization. Applied Mathematical Modelling 27 (4), 275 – 291.

Marchetti, A., Chachuat, B., Bonvin, D., 2010. A dual modifier-adaptation approach for real-time optimization. Journal of Process Control 20 (9), 1027 – 1037.

Martinez, D., Schwaab, M., Lima, E. L., Pinto, J. C., 2010. Simultaneous robust data reconciliation and gross error detection through particle swarm optimization for an industrial polypropylene reactor. Chemical Engineering Science 65 (17), 4943 – 4954.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., Scokaert, P. O. M., 2000. Constrained model predictive control: Stability and optimality. Automatica 36 (6), 789 – 814.

Morari, M., 1982. Integrated plant control: A solution at hand or a research topic for the next decade? In: Proceedings of of Second International Conference on Chemical Process Control. pp. 467–495.

Morari, M., Stephanopoulos, G., Arkun, Y., 1980. Studies in the synthesis of control structures for chemical processes. Part I: Formulation of the problem. Process decomposition and the classification of the control task. Analsysis of the optimizing control structures. AIChE Journal 26 (2), 220–232.

Narraway, L., Perkins, J., 1994. Selection of process control structure based on economics. Computers and Chemical Engineering 18 (SUPPL), S511–S515.

Qin, S. J., Badgwell, T. A., 2000. An overview of nonlinear model predictive control applications. In: Frank Allgöwer, A. Z. (Ed.), Nonlinear Model Predictive Control. Progress in Systems and Control Theory. Birkhäuser, pp. 369–392.

Qin, S. J., Badgwell, T. A., 2003. A survey of industrial model predictive control technology. Control Engineering Pactice 11.

Rawlings, J. B., Mayne, D. Q., 2009. Model Predictive Control. Nob Hill Publishing.

Richalet, J., Rault, A. J., Testud, J., Papon, J., 1978. Model predictive heuristic control: Applications to industrial processes. Automatica 14 (5), 413 – 428.

Skogestad, S., 2000a. Plantwide control: The search for the self-optimizing control structure. Journal of Process Control 10, 487–507.

Skogestad, S., 2000b. Self-optimizing control: the missing link between steady-state optimization and control. Computers & Chemical Engineering 24 (2-7), 569 – 575.

Skogestad, S., 2004. Control structure design for complete chemical plants. Computers & Chemical Engineering 28.

Skogestad, S., Postlethwaite, I., 1996. Multivariable Feedback Control. John Wiley & Sons.

Srinivasan, B., Palanki, S., Bonvin, D., 2003. Dynamic optimization of batch processes: I. characterization of the nominal solution. Computers & Chemical Engineering 27 (1), 1 – 26.

Trenchard, A., Boder, H., aug.-sept. 2005. How do you know which control loops are the most important? Computing Control Engineering Journal 16 (4), 24 – 29.

Zhang, Y., Forbes, J. F., 2000. Extended design cost: a performance criterion for real-time optimization systems. Computers & Chemical Engineering 24 (8), 1829 – 1841.

Chapter 3

Optimal operation by controlling the gradient to zero

If you cry ’forward’, you must without fail make plain in what direction to go.

A. Chekhov

From an optimization point of view, the gradient is the key variable which gives information about the optimality of a process. In this paper we present how the gradient is related to the loss from optimality, and show how determining a good set of controlled variables can be considered as *weighted* approximation of the gradient. We show that even if there are setpoint changes for the controlled variables, this can still be considered as approximating the gradient.

Based on the paper accepted for publication at the IFAC World Congress 2011, Milano.

3.1 Introduction

The overall objective of process operation is to minimize the cost J (or equivalently to maximize the profit $P = -J$) subject to given constraints. However, when using control, the objective is to keep selected controlled variables $\mathbf{c}$ at their optimal setpoints,

$$\mathbf{c}(\mathbf{y}) = \mathbf{c}_s. \tag{3.1}$$

With respect to these two goals, Morari et al. [1980] stated

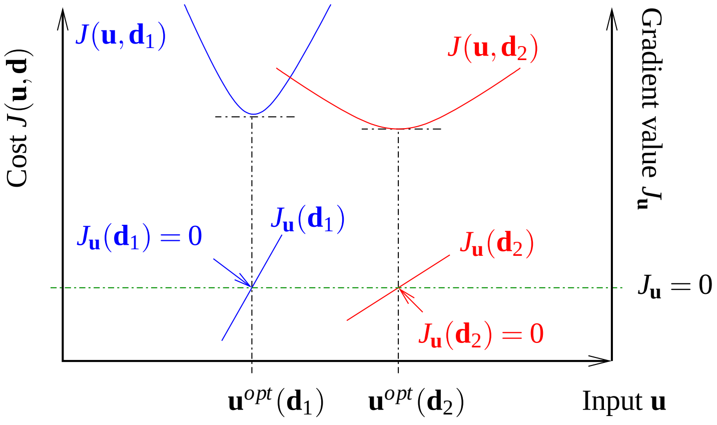

Cost $J(\mathbf{u},\mathbf{d})$
$J(\mathbf{u},\mathbf{d}_1)$
$J(\mathbf{u},\mathbf{d}_2)$
Gradient value $J_\mathbf{u}$
$J_\mathbf{u}(\mathbf{d}_1) = 0$
$J_\mathbf{u}(\mathbf{d}_1)$
$J_\mathbf{u}(\mathbf{d}_2)$
$J_\mathbf{u} = 0$
$J_\mathbf{u}(\mathbf{d}_2) = 0$
$\mathbf{u}^{opt}(\mathbf{d}_1)$
$\mathbf{u}^{opt}(\mathbf{d}_2)$
Input $\mathbf{u}$

Figure 3.1: Cost and gradients for different disturbances $\mathbf{d}$

> "…our main objective is to translate the economic objectives into process control objectives. In other words we want to find a function $\mathbf{c}$ of the process variables […] which when held constant leads automatically to the optimal adjustment of the manipulated variables, and with it, the optimal operating conditions."

However, they do not give a systematic method for finding the controlled variables, nor do they mention that for the unconstrained case, the obvious approach to get consistency between economic and process control objectives is to select the gradient as the controlled variable. That is, to select

$$\mathbf{c} = J_\mathbf{u}(\mathbf{u},\mathbf{d}), \tag{3.2}$$

and keep the setpoint constant at zero, $\mathbf{c}_s = 0$. Here $\mathbf{u}$ are the unconstrained degrees of freedom, $\mathbf{d}$ are unmeasured disturbances, and $J_\mathbf{u}(\mathbf{u},\mathbf{d}) = \partial J(\mathbf{u},\mathbf{d})/\partial \mathbf{u}$ is the gradient. Irrespective of the disturbance, the optimal value of $J_\mathbf{u}$ is zero, (Figure 3.1). This was proposed by Halvorsen and Skogestad [1997a], who write that the ideal controlled variable would be

$$\mathbf{c} = \mathbf{c}_1 J_\mathbf{u} + \mathbf{c}_0, \tag{3.3}$$

where $\mathbf{c}_0$ and $\mathbf{c}_1$ are constants. The idea has also been proposed by Halvorsen and Skogestad [1997b, 1999]; Bonvin et al. [2001]; Cao [2003, 2005]; Srinivasan et al. [2008], and intuitively it seems to be an excellent idea. The elements of the gradient change sign when moving from one side of the optimum to another side (Figure 3.2), thus, it is well suited for feedback control.

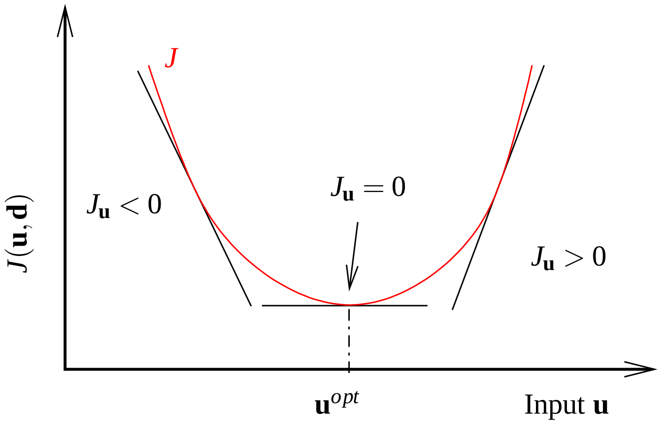


Figure 3.2: Cost and gradient values

However, in practice, we rarely have a measurement of the gradient and it is often not clearly defined what it means to control the gradient to zero. The gradient is a vector, and in many practical cases is not possible to control all elements exactly to zero. What should we do in these cases? A first attempt to answer this question will be to find a control structure, which minimizes the norm of the gradient. This is a good start; however, it is important to keep in mind that our ultimate goal is to minimize the cost J, so this original criterion has to be applied to evaluate the possible control structures.

The starting point is to write the controlled variables as a function of measurements $\mathbf{y}_m$,

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m, \tag{3.4}$$

which is controlled to zero, $\mathbf{c} = \mathbf{H}\mathbf{y}_m = 0$. Here, $\mathbf{c} \in \mathbb{R}^{n_\mathbf{c}}$, and $\mathbf{H} \in \mathbb{R}^{n_\mathbf{c} \times n_\mathbf{y}}$. For the system to be fully specified, we need as many controlled variables as we have inputs $\mathbf{u}$, that is $n_\mathbf{c} = n_\mathbf{u}$.

Since the gradient is optimally at zero, we can consider $\mathbf{c} = \mathbf{H}\mathbf{y}_m$ as an approximation of the gradient. If the approximation is exact, $\mathbf{H}\mathbf{y}_m = J_\mathbf{u}$ then we will have optimal operation whenever $\mathbf{c} = 0$, provided convexity. If it is not possible to control the gradient (because of e.g. unmeasured disturbances, noise and missing measurements), there will be some loss associated to the chosen control structure. To evaluate the performance of the chosen control policy, we use the original cost function and define the loss from optimality

$$L = J(\mathbf{u},\mathbf{d}) - J(\mathbf{u}^{opt}(\mathbf{d}),\mathbf{d}). \tag{3.5}$$

Note that the loss is deliberately not expressed as a function of the control structure. This is because we want to be able to use the loss to evaluate open-loop strategies, too. If we want to evaluate the loss for a particular control structure, we insert the **u** which corresponds to the applied control structure. For more details on the loss, and its calculation, we refer to Halvorsen et al. [2003].

Considering the problem of selecting the best control structures, there are two important questions, which we would like to address in this paper:

Q1. Does a **H** which minimizes $||J_{\mathbf{u}}(\mathbf{H}\mathbf{y}_m = 0)||_2$ also minimize $L(\mathbf{H}\mathbf{y}_m = 0)$?

Q2. If not, is the difference significant?

In terms of **Q1.** we show in Theorem 1 that minimizing the norm of the gradient is not quite the same as minimizing the loss L.

In terms of **Q2.** we show that it is important in the case, when we have structural constraints on **H**. That is, we have control structures involving different measurements.

Another contribution of this paper is an extremely simple derivation of the null space method [Alstad and Skogestad, 2007].

Furthermore, we show we show how setpoint changes of the controlled variables can be seen in the context of minimizing the loss or approximating the gradient.

This paper is structured such that the next section presents our main result, a derivation of the expression for the economic loss based on the gradient. In Section 3.3 we describe how this interpretation is connected to existing methods, and its importance. Section 3.4 discusses how the case of varying setpoints can be treated in this framework. After presenting a distillation case study in Section 3.5, we close the paper with a discussion and conclusions.

3.2 Derivation of the loss expression using the gradient

3.2.1 Preliminaries

Consider the feedback system in Figure 3.3, where the variables $\mathbf{c}$ and $\mathbf{c}_s$ denote the n_c-dimensional vector valued controlled variable and its setpoint, respectively, and where the variables $\mathbf{n}^{\mathbf{y}} \in \mathbb{R}^{n_{\mathbf{y}}}, \mathbf{n}^{\mathbf{c}} \in \mathbb{R}^{n_{\mathbf{c}}}$ denote the noise and the steady state control error, respectively. The noisy measurements are denoted $\mathbf{y}_m \in \mathbb{R}^{n^{\mathbf{y}}}$, and we assume that the controllers have integral action so that there is no steady state error, $\mathbf{n}^{\mathbf{c}} = 0$; then at steady state $\mathbf{c}_s = \mathbf{c}$.

After all active constraints are satisfied (controlled), the remaining unconstrained problem can be approximated by a quadratic problem in the neighborhood

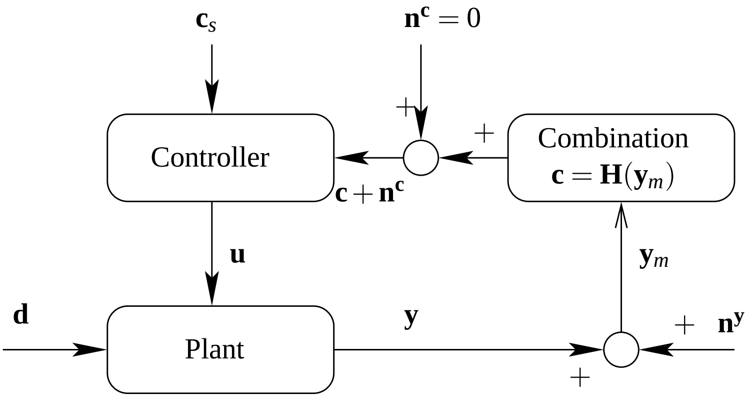


Figure 3.3: Control structure (with integral action, $\mathbf{c}_s = \mathbf{c}$ at steady state, $\mathbf{n^c} = 0$), (Adapted from [Alstad et al., 2009])

of the optimal point,

$$\min_{\mathbf{u}} \frac{1}{2} \begin{bmatrix} \mathbf{u}^{\mathrm{T}} & \mathbf{d}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}. \tag{3.6}$$

Here $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$, $\mathbf{d} \in \mathbb{R}^{n_\mathbf{d}}$, and $\mathbf{J_{uu}}$, $\mathbf{J_{ud}}$, $\mathbf{J_{du}}$, $\mathbf{J_{dd}}$ are matrices of appropriate sizes. In addition, we assume that $\mathbf{J_{uu}} > 0$. For small deviations around the nominal optimum, the plant can be described by the linear model

$$\begin{aligned} \mathbf{y} &= \mathbf{G^y u} + \mathbf{G^y_d d} \\ &= \tilde{\mathbf{G}}^\mathbf{y} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}, \end{aligned} \tag{3.7}$$

where $\mathbf{G^y} \in \mathbb{R}^{n_\mathbf{y} \times n_\mathbf{u}}$ and $\mathbf{G^y_d} \in \mathbb{R}^{n_\mathbf{y} \times n_\mathbf{d}}$ are the steady state gain matrices from $\mathbf{u}$ and $\mathbf{d}$ to the outputs $\mathbf{y}$. Our goal is to find controlled variables of the form

$$\begin{aligned} \mathbf{c} &= \mathbf{H y}_m \\ &= \mathbf{H}(\mathbf{y} + \mathbf{n^y}) \end{aligned} \tag{3.8}$$

where $\mathbf{H} \in \mathbb{R}^{n_\mathbf{c} \times n_\mathbf{y}}$, which, when controlled to zero, yield optimal or near optimal operation.

3.2.2 Approximating the gradient

The gradient of the approximated problem (3.6) is

$$J_\mathbf{u}(\mathbf{u}, \mathbf{d}) = \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix} = \mathbf{J_{uu} u} + \mathbf{J_{ud} d}. \tag{3.9}$$

Assuming that (3.6) matches the real plant, the necessary condition for optimality is

$$J_{\mathbf{u}}(\mathbf{u},\mathbf{d}) = \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix} = 0. \tag{3.10}$$

As mentioned above, the ideal controlled variable is the gradient, $\mathbf{c} = J_{\mathbf{u}}(\mathbf{u},\mathbf{d})$. When it is known exactly, using it as a controlled variable is the best choice and works fine. In practice, however, the gradient must be estimated somehow using measurement information. Then the controlled variable becomes

$$\mathbf{c} = \hat{J_{\mathbf{u}}}. \tag{3.11}$$

Obtaining the gradient estimate $\hat{J_{\mathbf{u}}}$ can be done in several ways, such as e.g. black box modelling or estimating the gradient using statistical methods. In the case of zero-mean noise, the effects may cancel out, but if there is a constant non-zero offset, the noise can deteriorate performance severely, thus we have to include the noise in the analysis, too.

A first approach would be to find a controlled variable $\mathbf{c} = \mathbf{H}\mathbf{y}_m$ which minimizes the worst case gradient norm, e.g. to select $\mathbf{H}$ as

$$\mathbf{H} = \arg\left(\min_{\mathbf{H}}\max_{\mathbf{d}} ||J_{\mathbf{u}} - \mathbf{H}\mathbf{y}_m||_2\right). \tag{3.12}$$

In the non-ideal case, when $\mathbf{H}\mathbf{y}_m \neq J_{\mathbf{u}}$, controlling $\mathbf{H}\mathbf{y}_m$ to zero will result in a gradient which has nonzero elements, and therefore has nonzero norm,

$$||J_{\mathbf{u}}(\mathbf{H}\mathbf{y}_m = 0)||_2 \neq 0. \tag{3.13}$$

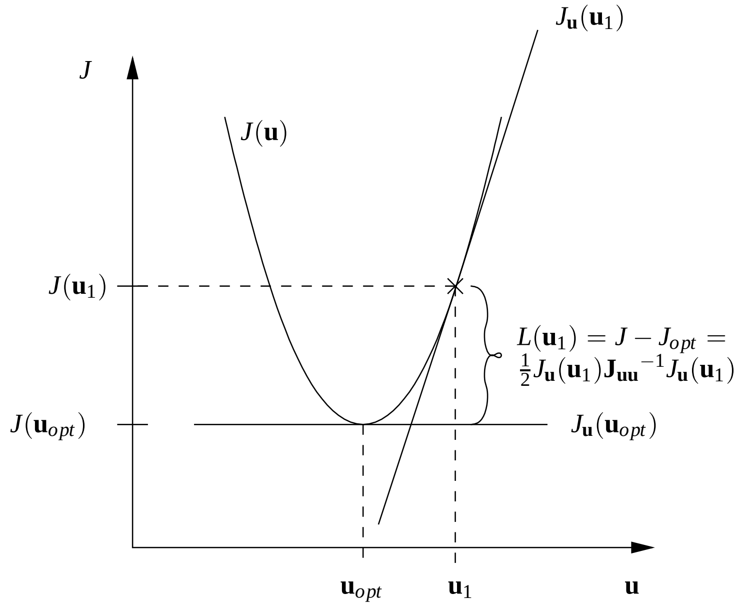
The norm of the gradient may seem a good criterion to evaluate suboptimality; however it does not truly reflect the performance in terms of the original cost function. To quantify the suboptimality, we consider the loss L, which is defined as the difference between the actual cost and the optimal cost for a given disturbance $\mathbf{d}$,

$$L = J(\mathbf{u},\mathbf{d}) - J(\mathbf{u}^{opt}(\mathbf{d}),\mathbf{d}). \tag{3.14}$$

Note that we are considering the loss with respect to the truly optimal instead of the cost. The loss has the properties of a weighted norm.

Theorem 1. *The local economic loss can be expressed to first order in terms of the current gradient value as*

$$L = \frac{1}{2}\left|\left|\mathbf{J_{uu}}^{-1/2} J_{\mathbf{u}}\right|\right|_2^2. \tag{3.15}$$



$J_{\mathbf{u}}(\mathbf{u}_1)$
J
$J(\mathbf{u})$
$J(\mathbf{u}_1)$
$L(\mathbf{u}_1) = J - J_{opt} = \frac{1}{2}J_{\mathbf{u}}(\mathbf{u}_1)\mathbf{J_{uu}}^{-1}J_{\mathbf{u}}(\mathbf{u}_1)$
$J(\mathbf{u}_{opt})$
$J_{\mathbf{u}}(\mathbf{u}_{opt})$
$\mathbf{u}_{opt}$
$\mathbf{u}_1$
$\mathbf{u}$

Figure 3.4: Loss L imposed by non-optimal operation

Proof. From Halvorsen et al. [2003] it is known that the loss can be written as

$$L = \frac{1}{2}(\mathbf{u} - \mathbf{u}^{opt}(\mathbf{d}))^{\mathrm{T}}\mathbf{J_{uu}}(\mathbf{u} - \mathbf{u}^{opt}(\mathbf{d})). \tag{3.16}$$

Solving $J_{\mathbf{u}} = 0$ (3.10) for $\mathbf{u}^{opt}(\mathbf{d}) = -\mathbf{J_{uu}}^{-1}\mathbf{J_{ud}}\mathbf{d}$, and inserting into (3.16) yields (note that $\mathbf{J_{uu}}$ is symmetric):

$$\begin{aligned} L &= \frac{1}{2}(\mathbf{u} + \mathbf{J_{uu}}^{-1}\mathbf{J_{ud}}\mathbf{d})^{\mathrm{T}}\mathbf{J_{uu}}(\mathbf{u} + \mathbf{J_{uu}}^{-1}\mathbf{J_{ud}}\mathbf{d}) \\ &= \frac{1}{2}(\mathbf{u}^{\mathrm{T}} + \mathbf{d}^{\mathrm{T}}\mathbf{J_{ud}}^{\mathrm{T}}\mathbf{J_{uu}}^{-T})\mathbf{J_{uu}}(\mathbf{u} + \mathbf{J_{uu}}^{-1}\mathbf{J_{ud}}\mathbf{d}) \\ &= \frac{1}{2}(\mathbf{u}^{\mathrm{T}}\mathbf{J_{uu}}^{\mathrm{T}} + \mathbf{d}^{\mathrm{T}}\mathbf{J_{ud}}^{\mathrm{T}})\mathbf{J_{uu}}^{-1}(\mathbf{J_{uu}}\mathbf{u} + \mathbf{J_{ud}}\mathbf{d}) \\ &= \frac{1}{2}J_{\mathbf{u}}^{\mathrm{T}}\mathbf{J_{uu}}^{-1}J_{\mathbf{u}} \\ &= \frac{1}{2}\left|\left|\mathbf{J_{uu}}^{-1/2}J_{\mathbf{u}}\right|\right|_2^2. \end{aligned} \tag{3.17}$$

□

At the optimum, $\mathbf{u}^{opt}$, the gradient $J_{\mathbf{u}} = 0$, and the loss $L = 0$. Around the optimum $J_{\mathbf{u}} \neq 0$, the loss L is equal to the norm the weighted gradient, where the weight factor is $\mathbf{J_{uu}}^{-1/2}$, Figure 3.4.

Remark 1 (Effect of constraints). *The above analysis is locally valid for a system where all active constraints are known and have been satisfied, $g(\mathbf{u},\mathbf{d}) = 0$. If an active constraint is not satisfied exactly, $g(\mathbf{u},\mathbf{d}) = \varepsilon$, then the effect on the objective function will be given by the corresponding Lagrangian multiplier [Nocedal and Wright, 2006]*

$$\lambda = \partial J/\partial\varepsilon. \tag{3.18}$$

A perturbation of the constraints ε has therefore a first order effect on the cost function, while from (3.15)*, a small change in $J_{\mathbf{u}}$ has a second order effect on the cost. From an economic point of view, tight control of the active constraints will generally be more important than tight control of the unconstrained variable* **c***.*

3.3 Minimizing the gradient vs. minimizing the loss

Theorem 1 shows that a controlled variable which minimizes $||J_{\mathbf{u}}||_2$, does not necessarily minimize the loss L. One case, where $\mathbf{J_{uu}}^{-1/2}$ has no effect is, when it is orthogonal, $\mathbf{J_{uu}}^{-1/2} = \mathbf{J_{uu}}^{-1/2^{\mathrm{T}}}$, or scalar. In the next sections, we examine in which further cases an $\mathbf{H}$ which minimizes $||J_{\mathbf{u}}||_2$ is the same that minimizes the loss L.

3.3.1 Enough measurements, no noise, full H: same H

If it is possible to have zero loss (no noise and sufficient measurements), optimal operation corresponds to $J_{\mathbf{u}} = 0$. Then, $\mathbf{J_{uu}}^{-1/2}$ has no effect. Assume that $\mathbf{y}$ contains all available information, then we require that

$$J_{\mathbf{u}} = \mathbf{H}\mathbf{y}. \tag{3.19}$$

Theorem 2 (Null space method, no noise). *Given a linear model as in* (3.7)*, with a sufficient number of independent measurements ($n_{\mathbf{y}} \geq n_{\mathbf{u}} + n_{\mathbf{d}}$) and no noise ($\mathbf{n^y} = 0$), selecting*

$$\mathbf{H} = \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{bmatrix} \left[\tilde{\mathbf{G}}^{\mathbf{y}}\right]^{-1}, \tag{3.20}$$

and controlling $\mathbf{Hy} = 0$ *gives zero loss from optimal operation. Here,* $\tilde{\mathbf{G}}^{\mathbf{y}}$ *is the gain matrix of any subset of $n_u + n_d$ measurements.*

Proof. The gradient from (3.9) is

$$J_{\mathbf{u}} = \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}. \tag{3.21}$$

We want to eliminate the variables $[\mathbf{u}, \mathbf{d}]^{\mathrm{T}}$ using the available measurements,

$$\mathbf{y} = \tilde{\mathbf{G}}^{\mathbf{y}} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}. \tag{3.22}$$

Solving for $[\mathbf{u}^{\mathrm{T}}, \mathbf{d}^{\mathrm{T}}]^{\mathrm{T}}$,

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix} = \left[\tilde{\mathbf{G}}^{\mathbf{y}}\right]^{-1} \mathbf{y}, \tag{3.23}$$

and inserting into (3.21) gives:

$$\begin{aligned} J_{\mathbf{u}} &= \begin{bmatrix} \mathbf{J}_{\mathbf{uu}} & \mathbf{J}_{\mathbf{ud}} \end{bmatrix} \left[\tilde{\mathbf{G}}^{\mathbf{y}}\right]^{-1} \mathbf{y} \\ &= \mathbf{H}\mathbf{y}. \end{aligned} \tag{3.24}$$

Controlling $\mathbf{c} = \mathbf{H}\mathbf{y} = 0$ results in zero loss. □

- This is a new derivation of the null space method reported in [Alstad and Skogestad, 2007]. It shows that the optimal controlled variable found by self-optimizing control is identical to the gradient,

$$\mathbf{c} = J_{\mathbf{u}} = \mathbf{H}\mathbf{y}. \tag{3.25}$$

3.3.2 Enough measurements, noise, full H: same H

The case of finding a controlled variable combination, which minimizes the loss in presence of sufficient (noisy) measurements and a full **H** matrix is addressed in the "exact local method" Alstad et al. [2009]. First, we scale the disturbances and the noise, such that

$$\mathbf{d} = \mathbf{W}_{\mathbf{d}} \mathbf{d}', \tag{3.26}$$

and

$$\mathbf{n}^{\mathbf{y}} = \mathbf{W}_{\mathbf{n}^{\mathbf{y}}} \mathbf{n}^{\mathbf{y}\prime}, \tag{3.27}$$

where

$$\left\| \begin{bmatrix} \mathbf{d}' \\ \mathbf{n}^{\mathbf{y}\prime} \end{bmatrix} \right\|_2 \leq 1 \tag{3.28}$$

and $\mathbf{W}_{\mathbf{d}}$ and $\mathbf{W}_{\mathbf{n}^{\mathbf{y}}}$ are diagonal scaling matrices of appropriate sizes. Then we:

1. Express L as a function of $\mathbf{H}$, $\mathbf{d}$ and $\mathbf{n}^{\mathbf{y}}$ (assuming $\mathbf{c} = \mathbf{H}(\mathbf{y} + \mathbf{n}^{\mathbf{y}}) = 0$).

2. Then find an expression for the worst-case loss $L(\mathbf{H})$ (worst-case w.r.t. $\mathbf{d}$ and $\mathbf{n}^{\mathbf{y}}$); which is the maximum singular value $\bar{\sigma}(\mathbf{M})$. Here

$$\mathbf{M} = \mathbf{J}_{\mathbf{uu}}{}^{1/2} (\mathbf{H}\mathbf{G}^{\mathbf{y}})^{-1} \mathbf{H}\tilde{\mathbf{F}} \tag{3.29}$$

and

$$\tilde{\mathbf{F}} = [\mathbf{F}\mathbf{W_d} \quad \mathbf{W_{n^y}}], \tag{3.30}$$

where

$$\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}} \tag{3.31}$$

is the optimal measurement sensitivity matrix, see Halvorsen et al. [2003]. (Kariwala et al. [2008] have shown that the average loss is given by $||\mathbf{M}||_F$, where $||\cdot||_F$ denotes the Frobenius norm).

3. Find a convex problem formulation for finding **H** (see Alstad et al. [2009]).

The convex problem for finding an **H** which minimizes the average and worst case loss for a given set of disturbances is [Alstad et al., 2009]

$$\begin{aligned} &\min_{\mathbf{H}} ||\mathbf{H}[\mathbf{F}\mathbf{W_d} \quad \mathbf{W_{n^y}}]||_F \\ &\text{subject to } \mathbf{H}\mathbf{G}^{\mathbf{y}} = \mathbf{Q}, \end{aligned} \tag{3.32}$$

where **Q** is any non-singular $n_{\mathbf{u}} \times n_{\mathbf{u}}$ matrix, and **F** as defined in (3.31). Here, too, $\mathbf{J_{uu}}$ is not needed for fining the best measurement combination. However, if we want to know the actual worst case or average loss, we need $\mathbf{J_{uu}}^{1/2}$ in (3.29).

- In the case of no structural constraints on **H**, it is found that $\mathbf{J_{uu}}$ is not needed for finding the best measurement combination **H**. That is, a controlled variable which minimizes $||J_{\mathbf{u}}||_2$ minimizes also the loss $L = \frac{1}{2}\left|\left|\mathbf{J_{uu}}^{-1/2} J_{\mathbf{u}}\right|\right|_2^2$.

3.3.3 Structural constraints on H: not the same H

In the above cases, we used all measurements **y** to generate the controlled variables as linear combinations of all measurements. In practice however, there are often structural constraints on the controlled variables. Examples for structural constraints include controlling single measurements, or using only two measurements from the rectifier section and two measurements from the stripping section of a distillation column. When we have to decide between two or more controlled structures, the norm of the gradient (if it is nonzero) does no longer give accurate information about what controlled variable is best. To be able to make a good decision in these cases, we need to consider norm of the weighted gradient $L = \frac{1}{2}\left|\left|\mathbf{J_{uu}}^{-1/2} J_{\mathbf{u}}\right|\right|_2^2$.

As an example, consider a process with

$$\mathbf{J_{uu}} = \begin{bmatrix} 244 & 222 \\ 222 & 202 \end{bmatrix} \text{ and } \mathbf{J_{ud}} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \tag{3.33}$$

and assume that

$$\tilde{\mathbf{G}}\mathbf{y} = \begin{bmatrix} 1 & 0.275 & 10 \\ 2.78 & 2 & 10 \\ 5 & -4.913 & 10 \\ 2.1826 & 2 & 0 \end{bmatrix}. \tag{3.34}$$

Assume further that we have the choice between using

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.35}$$

and

$$\mathbf{H}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.36}$$

This gives the controlled variables

$$\mathbf{c}_1 = \begin{bmatrix} 1 & 0.275 \\ 2.78 & 2 \end{bmatrix} \mathbf{u} + \begin{bmatrix} 10 \\ 10 \end{bmatrix} d \tag{3.37}$$

and

$$\mathbf{c}_2 = \begin{bmatrix} 5 & -4.9130 \\ 2.1826 & 2 \end{bmatrix} \mathbf{u} + \begin{bmatrix} 10 \\ 0 \end{bmatrix} d. \tag{3.38}$$

For a disturbance $d = 1$ the resulting inputs are found by setting $\mathbf{c}_1 = 0$ and $\mathbf{c}_2 = 0$, and solving for $\mathbf{u}$. Inserting them into the gradient expression

$$J_{\mathbf{u}} = \begin{bmatrix} \mathbf{J}_{\mathbf{uu}} & \mathbf{J}_{\mathbf{ud}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ d \end{bmatrix}, \tag{3.39}$$

gives

$$J_{\mathbf{u}}(\mathbf{H}_1\mathbf{y} = 0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{3.40}$$

and

$$J_{\mathbf{u}}(\mathbf{H}_2\mathbf{y} = 0) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}. \tag{3.41}$$

The norm of the gradient is in both cases

$$||J_{\mathbf{u}}(\mathbf{c}_1 = 0)||_2 = ||J_{\mathbf{u}}(\mathbf{c}_2 = 0)||_2 = \sqrt{2}, \tag{3.42}$$

which indicates that the two controlled variables give equivalent performance. However, if we consider the loss imposed by the two different control structures, we have that

$$L(\mathbf{c}_1) = \frac{1}{2}\left|\left|\mathbf{J}_{\mathbf{uu}}{}^{-1/2} J_{\mathbf{u}}(\mathbf{c}_1)\right|\right|_2^2 = 0.25 \tag{3.43}$$

and

$$L(\mathbf{c}_2) = \frac{1}{2}\left|\left|\mathbf{J}_{\mathbf{uu}}{}^{-1/2} J_{\mathbf{u}}(\mathbf{c}_2)\right|\right|_2^2 = 111.25. \tag{3.44}$$

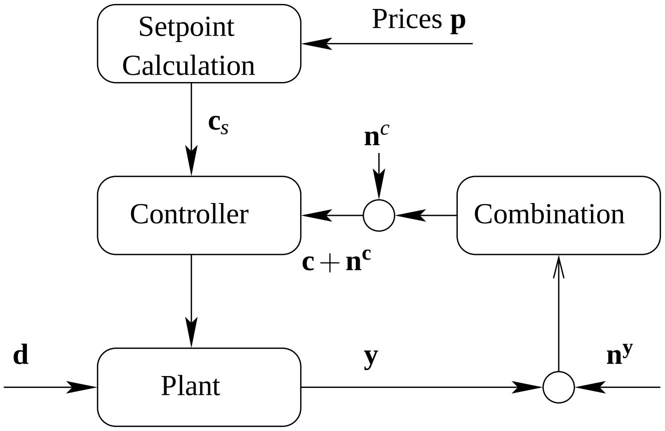


Figure 3.5: Feedback control structure with setpoint calculation (with integral action, $\mathbf{n}^{\mathbf{c}} = 0$)

- When we want to compare sets of controlled variables with each other, we need to examine the loss, as the gradient does not give sufficient information.

- When searching for the best linear combination for a given set of measurements, it is sufficient to consider the gradient.

3.4 Varying setpoints for the controlled variables

Many processes are operated such that the setpoints of the controlled variables are changed, when for example product prices **p** and specifications change, Figure 3.5. To handle this in the framework above, we consider the reason for the setpoint change as a *measured disturbance*. The relationship between the measurements $\mathbf{y}_m$ and the controlled variables is

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m, \tag{3.45}$$

and the relationship between the setpoint change and the prices **p** is

$$\mathbf{c}_s = \mathbf{H}_s\mathbf{p}. \tag{3.46}$$

We define $J_{\mathbf{u}} = \mathbf{c} - \mathbf{c}_s$,

$$\begin{aligned} J_{\mathbf{u}} &= \mathbf{H}\mathbf{y}_m - \mathbf{H}_s\mathbf{p} = \begin{bmatrix} \mathbf{H} & -\mathbf{H}_s \end{bmatrix} \begin{bmatrix} \mathbf{y}_m \\ \mathbf{p} \end{bmatrix} \\ &= \mathbf{H}_{aug}\mathbf{y}_{aug}. \end{aligned} \tag{3.47}$$

The gain matrices are augmented according to

$$\mathbf{G}^{\mathbf{y}}_{aug} = \left[\begin{array}{c} \mathbf{G}^{\mathbf{y}}_{n_y \times n_u} \\ 0_{n_p \times n_u}, \end{array}\right], \quad \mathbf{G}^{\mathbf{y}}_{aug,\mathbf{d}} = \left[\begin{array}{cc} \mathbf{G}^{\mathbf{y}} & 0_{n_p \times n_p} \\ 0_{n_p \times n_d} & \mathbf{I}_{n_p \times n_p} \end{array}\right], \tag{3.48}$$

and the scaling matrices according to

$$\mathbf{W}_{\mathbf{d},aug} = \left[\begin{array}{cc} \mathbf{W}_{\mathbf{d}} & 0_{n_d \times n_p} \\ 0_{n_p \times n_d} & \mathbf{W}_{\mathbf{p}_{n_p \times n_p}} \end{array}\right], \tag{3.49}$$

$$\mathbf{W}_{\mathbf{n},aug} = \left[\begin{array}{cc} \mathbf{W}_{\mathbf{n}} & 0_{n_y \times n_p} \\ 0_{n_p \times n_y} & \mathbf{W}_{\mathbf{np}_{n_p \times n_p}} \end{array}\right]. \tag{3.50}$$

Here, $\mathbf{W}_{\mathbf{p}}$ and $\mathbf{W}_{\mathbf{np}}$ are diagonal matrices with the expected price variations and uncertainties, respectively. If the prices are known exactly, $\mathbf{W}_{\mathbf{np},aug} = 0$. The sensitivity matrix $\mathbf{F}_{aug}$ may be found by re-optimization or by evaluating

$$\mathbf{F}_{aug} = \mathbf{G}^{\mathbf{y}}_{aug,d} - \mathbf{G}^{\mathbf{y}}_{aug}\mathbf{J}_{\mathbf{uu}}{}^{-1}\mathbf{J}_{\mathbf{ud},aug}, \tag{3.51}$$

[Alstad and Skogestad, 2007], where $\mathbf{J}_{\mathbf{ud}aug}$ is calculated by including the price $\mathbf{p}$ as an additional disturbance. After the problem has been formulated, the optimal $\mathbf{H}_{aug}$ which minimizes the loss, is found by solving

$$\begin{aligned} &\min_{\mathbf{H}_{aug}} ||\mathbf{H}_{aug}[\mathbf{F}_{aug}\mathbf{W}_{\mathbf{d},aug} \quad \mathbf{W}_{\mathbf{n},aug}]||_F \\ &\text{subject to } \mathbf{H}_{aug}\mathbf{G}^{\mathbf{y}} = \mathbf{Q}. \end{aligned} \tag{3.52}$$

De-partitioning $\mathbf{H}_{aug} = [\mathbf{H} \quad -\mathbf{H}_s]$, the controlled variables and the setpoint updates are

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m \tag{3.53}$$

and

$$\mathbf{c}_s = \mathbf{H}_s\mathbf{p}. \tag{3.54}$$

3.5 Distillation case study

3.5.1 Problem description and setup

A binary distillation column is used to demonstrate the results. The column model is taken from Skogestad [1997]. It is controlled in the LV configuration and has 41 stages, Figure 3.6. We assume that the temperatures on stage 9, 16, 24, and 33 are measured and that they can be used for control, i.e. $\mathbf{y} = [T_9, T_{16}, T_{24}, T_{33}]^{\mathrm{T}}$. The

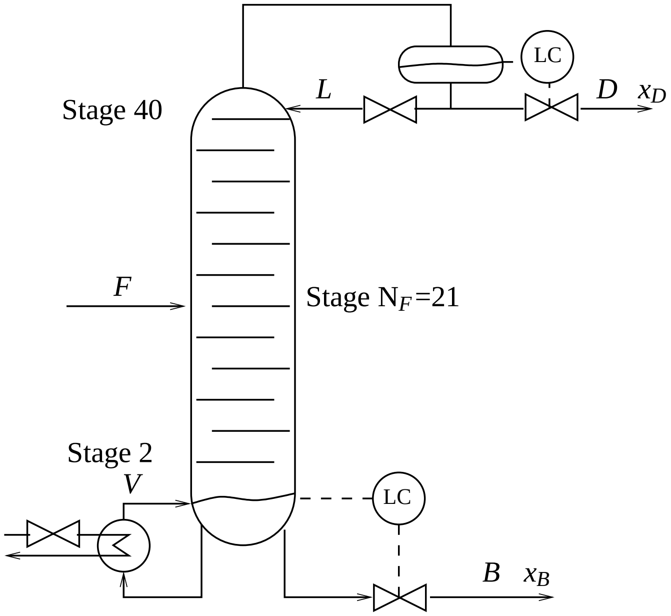


Figure 3.6: Distillation column

temperatures are calculated as a linear function of the liquid composition for the respective stages *i*,

$$T_i = 10(1-x_i). \tag{3.55}$$

This corresponds to a pure product boiling point difference of 10°C. In order to be able to sell the top product, a purity of 99% is required for the distillate *D*. This is considered an active constraint, and is controlled to its setpoint using the liquid reflux *L*. The remaining degree of freedom **u** (the boil-up *V*) can be used to maximize the profit which is the same as minimizing the difference between the costs for the feed and evaporation, and the profit from selling the purified products:

$$J = -(p_D D + p_B B - p_V V - p_F F). \tag{3.56}$$

Assuming the price for the feed is equal to the price of the bottom product, $p_F = p_B$, and introducing the overall mass balance, the cost function can be simplified to

$$J = p_V \left(\frac{p_F - p_D}{p_V} D + V \right) = p_V(p' D + V). \tag{3.57}$$

The only parameter which affects the location of the minimum is the relative price difference of the feed and the distillate,

$$p' = \frac{(p_F - p_D)}{p_V}, \tag{3.58}$$

and we assume $p' = -64$ currency units.

As disturbances, we consider the flow rate F, composition z and liquid fraction q of the feed. These disturbances are detectable through the measurement model in deviation variables:

$$\mathbf{y} = \mathbf{G}^{\mathbf{y}}\mathbf{u} + \mathbf{G}^{\mathbf{y}}_{\mathbf{d}}\mathbf{d}. \tag{3.59}$$

In addition, we assume that the product prices change and are known. We use the prices to update the setpoint of **c**. The self-optimizing controlled variable is selected as a linear combination of the four tray measurements. The augmented gain matrices and the augmented optimal sensitivity matrix are

$$\mathbf{G}^{\mathbf{y}}_{aug} = \begin{bmatrix} 1.71 \\ 3.22 \\ 1.36 \\ 0.20 \\ 0.00 \end{bmatrix}, \quad \mathbf{G}^{\mathbf{y}}_{\mathbf{d},aug} = \begin{bmatrix} -8.14 & -3.67 & -1.50 & 0 \\ -12.28 & -11.09 & -2.75 & 0 \\ -4.36 & -9.81 & -1.44 & 0 \\ -0.65 & -1.45 & -0.21 & 0 \\ 0.00 & 0.00 & 0.00 & 1 \end{bmatrix}, \tag{3.60}$$

$$\mathbf{F}_{aug} = \begin{bmatrix} -2.63 & -2.37 & -0.49 & -0.0055 \\ -1.90 & -8.65 & -0.84 & -0.0103 \\ 0.02 & -8.78 & -0.64 & -0.0044 \\ 0.01 & -1.30 & -0.09 & -0.0007 \\ 0.00 & 0.00 & 0.00 & 1.0000 \end{bmatrix}. \tag{3.61}$$

The weighting matrix $\mathbf{W}_{\mathbf{n},aug}$ is chosen such that all temperature measurements have an uncertainty of 0.5°C, and the price uncertainty is zero. The expected variation in the disturbances is captured in

$$\mathbf{W}_{\mathbf{d},aug} = \mathrm{diag}([0.1, 0.1, 0.1, 6.4]), \tag{3.62}$$

which corresponds to 10% variation in every disturbance variable. The corresponding second derivatives are

$$\begin{aligned} \mathbf{J}_{\mathbf{uu}} &= 4.85 \\ \mathbf{J}_{\mathbf{ud}aug} &= [-15.64, -3.68, -2.87, 0.02]. \end{aligned} \tag{3.63}$$

This gives a controlled variable combination $\mathbf{c} = \mathbf{H}\mathbf{y}$ with

$$\mathbf{H} = [0.23, 0.69, -0.28, -0.04], \tag{3.64}$$

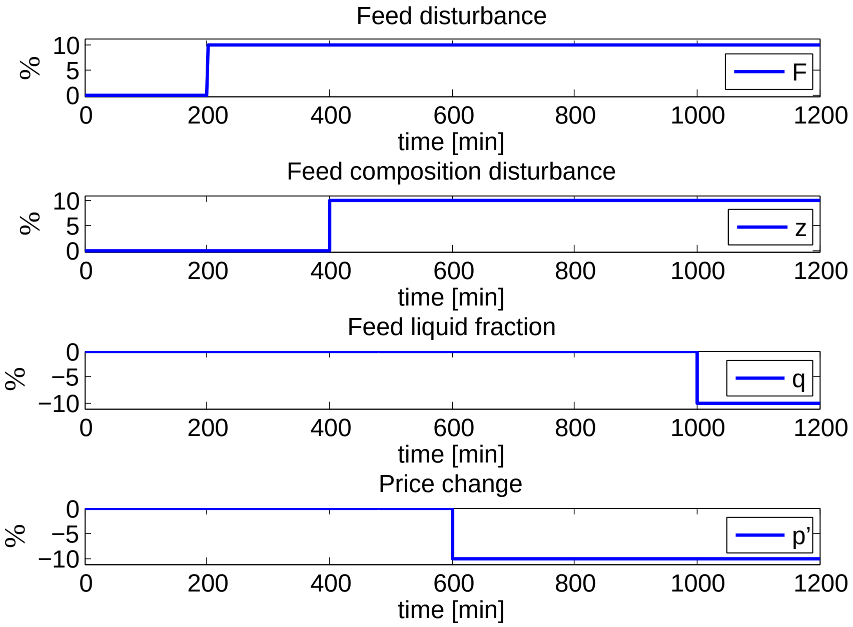


Figure 3.7: Disturbance trajectories

and the setpoint is updated using $\mathbf{c}_s = \mathbf{H}_s p'$ with

$$\mathbf{H}_s = 0.0071. \tag{3.65}$$

The first order loss from optimality estimate is calculated according to $L = \left|\left|\mathbf{J_{uu}}^{-1/2} J_{\mathbf{u}}\right|\right|_2^2$, or alternatively according to Halvorsen et al. [2003] as

$$\left|\left|\mathbf{J_{uu}}^{1/2}(\mathbf{H}_{aug}\mathbf{G}^{\mathbf{y}}_{aug})^{-1}\mathbf{H}_{aug}\left[\mathbf{F}_{aug}\mathbf{W}_{\mathbf{d},aug}\,\mathbf{W}_{\mathbf{n},aug}\right]\right|\right|_2^2, \tag{3.66}$$

and equals 1.4869 currency units.

3.5.2 Simulations

We consider disturbances in the flow rate, $\Delta F = 10\%$, the feed concentration, $\Delta z = 10\%$, the feed liquid fraction, $\Delta q = -10\%$, and the price, $\Delta p' = 10\%$. The disturbance scenario is given in Figure 3.7, and the resulting profit is plotted together with the inputs in Figure 3.8.

In Figure 3.9, the controlled variables are given together with their setpoints. The self-optimizing controlled variable is nicely controlled back to the setpoint after a disturbance enters the process. As long as the prices are constant, the setpoint

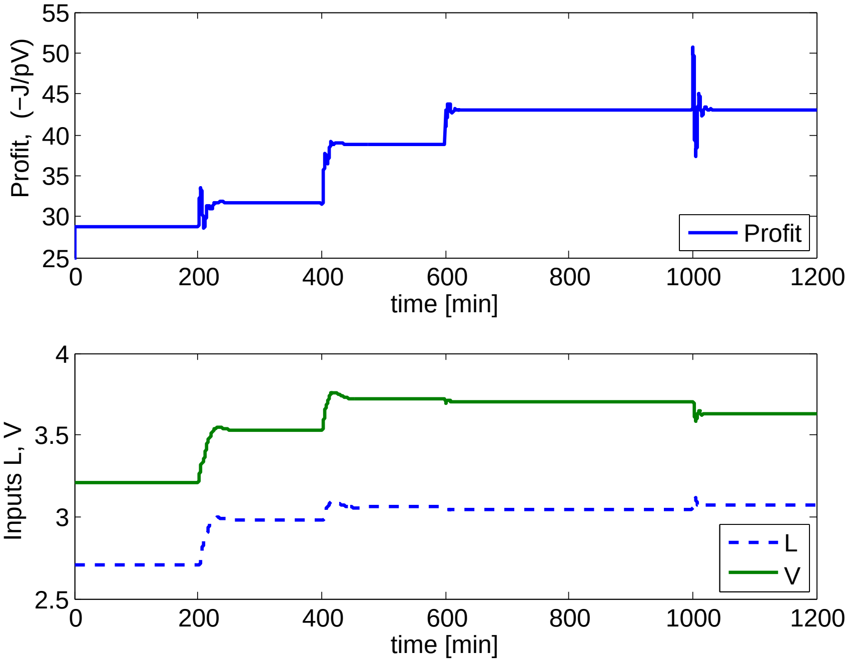


Figure 3.8: Profit and inputs for the distillation column

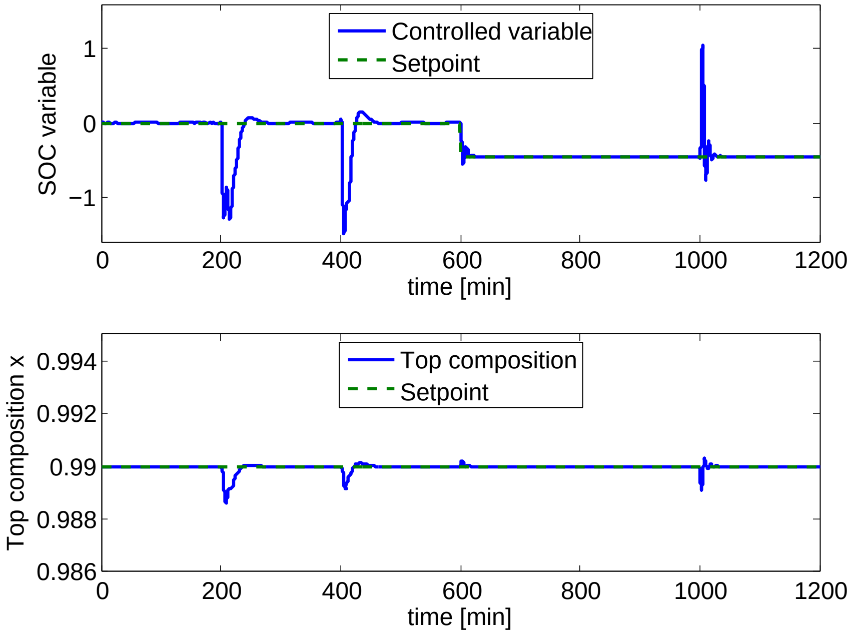


Figure 3.9: Controlled variables: Self-optimizing controlled variable and top composition

is zero. When the price ratio p' changes, the setpoint is adapted to a new value. The top composition is controlled well at its specification, as can be noted from the plotting scale.

3.6 Discussion and conclusions

We have given a first order approximation of the loss as the weighted norm of the gradient, and we have shown that if all measurements are used, the weighting is not required to find the best $\mathbf{H}$. However, when selecting between different sets of controlled variables, we need to consider the weighted gradient, because neglecting the weighting can be seriously misleading. The previously published "exact local method" indicates how close the norm of the weighted gradient is to zero when a particular set of controlled variables $\mathbf{c} = \mathbf{H}\mathbf{y}_m$ is used.

The key points are to weight the gradient when approximating it and include noise in the analysis. Otherwise we may approximate the gradient well while still suffering from unnecessary economic loss. The controlled variables obtained by this method have robustness against measurement noise. However, the underlying linear model and the cost function parameters are assumed to be locally exact, that is all the uncertainty is assumed to be taken care of in the measurement noise and disturbances.

Our analysis is based on the assumption that the active constraints do not to change. If the active constraints change, it is necessary to adjust the control structure to satisfy the new active set. However, if there are unconstrained degrees of freedom in the new active set, the above analysis can be reapplied.

The second part of this paper dealt with disturbances which do not enter through the model. By considering them as additional measurements, this can be formulated in terms of minimizing the weighted gradient, and the techniques from self-optimizing control can be used to update the setpoints to ensure optimal operation for all considered process and price disturbances.

References

Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. Industrial & Engineering Chemistry Research 46, 846–853.

Alstad, V., Skogestad, S., Hori, E. S., 2009. Optimal measurement combinations as controlled variables. Journal of Process Control 19 (1), 138–148.

Bonvin, D., Srinivasan, B., Ruppen, D., 2001. Dynamic Optimization in the Batch Chemical Industry. in: Chemical Process Control-VI.

Cao, Y., 2003. Self-optimizing control structure selection via differentiation. In: Proceedings of the European Control Conference. pp. 445–453.

Cao, Y., 2005. Direct and indirect gradient control for static optimisation. International Journal of Automation and Computing 2, 60–66.

Halvorsen, I. J., Skogestad, S., 1997a. Indirect on-line optimization through setpoint control. AIChE 1997 Annual Meeting, Los Angeles; paper 194h.

Halvorsen, I. J., Skogestad, S., 1997b. Optimizing control of petlyuk distillation: Understanding the steady-state behavior. Computers & Chemical Engineering 21 (Supplement 1), S249 – S254.

Halvorsen, I. J., Skogestad, S., 1999. Optimal operation of petlyuk distillation: steady-state behavior. Journal of Process Control 9 (5), 407 – 424.

Halvorsen, I. J., Skogestad, S., Morud, J. C., Alstad, V., 2003. Optimal selection of controlled variables. Industrial & Engineering Chemistry Research 42 (14), 3273–3284.

Kariwala, V., Cao, Y., Janardhanan, S., 2008. Local self-optimizing control with average loss minimization. Industrial & Engineering Chemistry Research 47, 1150–1158.

Morari, M., Stephanopoulos, G., Arkun, Y., 1980. Studies in the synthesis of control structures for chemical processes. Part I: Formulation of the problem. Process decomposition and the classification of the control task. Analsysis of the optimizing control structures. AIChE Journal 26 (2), 220–232.

Nocedal, J., Wright, S., 2006. Numerical Optimization. Springer.

Skogestad, S., 1997. Dynamics and control of distillation columns: A tutorial introduction. Chemical Engineering Research and Design 75 (6), 539 – 562.

Srinivasan, B., Biegler, L. T., Bonvin, D., 2008. Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function. Computers & Chemical Engineering 32, 572–279.

Chapter 4

Controlled variables from optimal operation data

Causa latet, vis est notissima

Ovid

In this paper we show how optimal operation data and concepts from self-optimizing control can be used for finding controlled variables which give optimal operation for the disturbances included in the data set. The method extracts the operation strategy which is hidden in the optimal data and may help analyze and improve operation in the common case where it is difficult or very expensive to obtain a good model.

Based on the published paper from the ESCAPE-21 Proceedings, Porto Carras, 2011

4.1 Introduction

For many processes, obtaining a good mathematical process model is important for successful operation. However, obtaining a good model is often inhibited by several factors, such as a tight budget and limited knowledge or time. Thus, obtaining a good process model and keeping the model up to date is one of the major bottlenecks for the application of advanced process control in industrial applications [Dochain et al., 2008]. It is therefore desirable to minimize the modelling effort, while still achieving good process performance. In this work we present a method for finding controlled variables, which is based on readily available logged process data. This data is used to find self-optimizing controlled variables whose optimal setpoint does not change with varying disturbances [Skogestad, 2000].

4.2 Motivation and problem formulation

An example for a system which is hard to model is a marathon runner. However, it is easy to collect data from runners, such as e.g. heart rate, stride frequency, temperature, blood oxygen content and breathing frequency. The data from the best runs of the runners subject to expected disturbances such as hilly terrain and wind is collected in an optimal data matrix Y. This data is used to determine a linear combination of measurements, which is (almost) constant for all the best runs. By running such as to keep this linear combination of variables at their optimal values, an optimal running strategy can be implemented. A single variable, which may give quite good performance, would be the heart rate. By running in such a way that the heart rate is constant at some optimal value, the running speed will automatically adjust to the disturbances (e.g. wind or changing slopes).

Similarly, in a process plant, some operators may be able to operate the process more profitably than others. Analyzing the "optimal operation data" of these operators can reveal linear combinations of variables, which other operators can use as guidance when operating the plant. Alternatively, these variables can be used for feedback control.

We assume that optimal operation corresponds to minimizing a cost J, and that the optimization problem can be approximated in deviation variables around the optimal point as

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) = \begin{bmatrix} \mathbf{u}^T & \mathbf{d}^T \end{bmatrix} \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}, \tag{4.1}$$

where $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$ and, $\mathbf{d} \in \mathbb{R}^{n_\mathbf{d}}$ are the inputs and the disturbances, respectively. In addition we require that $\mathbf{J_{uu}}$ is positive definite. For each degree of freedom $\mathbf{u}$ we search for a controlled variable $\mathbf{c}$ which is a linear combination of measurements $\mathbf{y}_m$,

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m. \tag{4.2}$$

Here $\mathbf{y}_m$ is defined as the sum of the “actual” measurement values $\mathbf{y}$ and the measurement noise $\mathbf{n}^\mathbf{y}$,

$$\mathbf{y}_m = \mathbf{y} + \mathbf{n}^\mathbf{y}. \tag{4.3}$$

If the controlled variables $\mathbf{c}$ give acceptable performance when controlled at constant setpoints, they are called self-optimizing. This term was coined by Skogestad [2000], who writes:

> *Self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables (without the need to reoptimize when disturbances occur).*

The loss is defined as $L = J(\mathbf{u}, \mathbf{d}) - J(\mathbf{u}^{opt}, \mathbf{d})$, where $\mathbf{u}$ is the input generated by the current operating policy, for example adjusting $\mathbf{u}$ such that $\mathbf{c} = \mathbf{H}\mathbf{y}_m$ is kept constant.

4.3 Data method

The new method for finding these measurement combinations is directly inspired by the null-space method [Alstad and Skogestad, 2007] which we present in the following.

4.3.1 Null space method

This method is based on the quadratic approximation of the cost function (1). In addition it is assumed that a linear noise free measurement model ($\mathbf{n}^{\mathbf{y}} = 0$) is available, so $\mathbf{y}_m = \mathbf{y} = \mathbf{G}^{\mathbf{y}}\mathbf{u} + \mathbf{G}^{\mathbf{y}}_{\mathbf{d}}\mathbf{d}$. Here, $\mathbf{y} \in \mathbb{R}^{n_{\mathbf{y}}}$ is the vector of linear independent measurements and $\mathbf{G}^{\mathbf{y}}, \mathbf{G}^{\mathbf{y}}_{\mathbf{d}}$ are the gain matrices of the system.

Theorem 3 (Null space method)**.** *Given a sufficient number of noise-free linear independent measurements, $n_{\mathbf{y}} \geq n_{\mathbf{u}} + n_{\mathbf{d}}$, select $\mathbf{H}$ such that $\mathbf{HF} = 0$, where*

$$\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$$

is the optimal sensitivity matrix. Then controlling $\mathbf{c} = \mathbf{H}\mathbf{y}$ to zero gives optimal operation with zero loss.

Proof. Close to $\mathbf{d}^{nom}$, by definition of $\mathbf{F}$ we have $\mathbf{y}^{opt}(\mathbf{d}) - \mathbf{y}^{opt}(\mathbf{d}^{nom}) = F(\mathbf{d} - \mathbf{d}^{nom})$. The optimal change in the controlled variables is: $\mathbf{c}^{opt}(\mathbf{d}) - \mathbf{c}^{opt}(\mathbf{d}^{nom}) = \mathbf{HF}(\mathbf{d} - \mathbf{d}^{nom})$. Since $\mathbf{H}$ is selected such that $\mathbf{HF} = 0$ optimal variation $\mathbf{c}^{opt} - \mathbf{c}^{opt}_{nom}$ is zero , too. Hence, controlling $\mathbf{c} = \mathbf{H}\mathbf{y}$ to zero leads to optimal operation. □

The optimal sensitivity matrix $\mathbf{F}$ is usually obtained numerically, by optimizing a model or by linearizing at the nominal point, and evaluating $\mathbf{F} = \mathbf{G}^{\mathbf{y}}_{\mathbf{d}} - \mathbf{G}^{\mathbf{y}}\mathbf{J}_{\mathbf{uu}}{}^{-1}\mathbf{J}_{\mathbf{ud}}$ [Alstad and Skogestad, 2007].

4.3.2 Using optimal operation data

In the case where we do not have an explicit model, we will not know the optimal sensitivity matrix $\mathbf{F} = \partial \mathbf{y}^{opt}/\partial \mathbf{d}$. Now let us assume that we have "optimal" data for $\mathbf{y}$ for various disturbances collected in the data matrix

$$\mathbf{Y} = \left[\mathbf{y}_1(\mathbf{d}_1), \ldots, \mathbf{y}_i(\mathbf{d}_i), \ldots, \mathbf{y}_{n_{sample}}(\mathbf{d}_{n_{sample}})\right], \tag{4.4}$$

where we have collected all measurements vectors $\mathbf{y}_i$ for different samples i, such that $n_\mathbf{u} + n_\mathbf{d} \leq n_{sample}$, and we assume that all variables are deviation variables from the nominal point. Note that the disturbances $\mathbf{d}_i$ for the different sample times are not known. In this case we can use following result:

Theorem 4 (Optimal data method - No noise)**.** *Given a sufficient number of independent measurements, $n_\mathbf{y} \geq n_\mathbf{u} + n_\mathbf{d}$, and optimal measurement data* $\mathbf{Y}$*, where each independent disturbance* $\mathbf{d}$ *is rejected at least once in the data* $\mathbf{Y}$*. Then the optimal measurement combination can be determined by selecting* $\mathbf{H}$ *such that* $\mathbf{HY} = 0$.

Proof. Using the definition of the optimal measurement matrix $\mathbf{F}$, we can write the optimal variation in the measurements as

$$\mathbf{y}^{opt}(\mathbf{d}) = \mathbf{Fd}. \tag{4.5}$$

Since every data point $\mathbf{y}_i$ in $\mathbf{Y}$ corresponds to a linear combination of "basis" disturbances,

$$\mathbf{d}_i = \alpha_1 \begin{bmatrix} d_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ d_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \ldots + \ldots + \alpha_{n_\mathbf{d}} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ d_{n_\mathbf{d}} \end{bmatrix}, \tag{4.6}$$

and we know that the data in $\mathbf{Y}$ is optimal, each column in $\mathbf{Y}$ can be considered as a linear combination of columns in $\mathbf{F}$. Then we have

$$\mathbf{y}_i^{opt}(\mathbf{d}_i) = \mathbf{Fd}_i = \mathbf{F}\left(\alpha_1 \begin{bmatrix} d_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ d_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \ldots + \alpha_{n_\mathbf{d}} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ d_{n_\mathbf{d}} \end{bmatrix} \right). \tag{4.7}$$

Therefore, we have

$$\mathbf{HF} = 0 \Leftrightarrow \mathbf{HY} = 0. \tag{4.8}$$

□

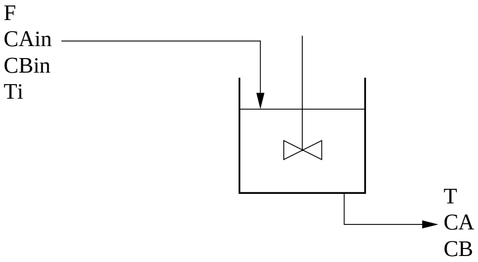


Figure 4.1: CSTR

In practice, the data matrix $\mathbf{Y}$ will not be consistent such that a null space $\mathbf{HY} = 0$ exists, either because of too many disturbances, or more likely because of measurement noise. One approach to handle this is to do a singular value decomposition $\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^{\mathbf{T}}$, and select the transpose of the $n_{\mathbf{u}}$ columns in $\mathbf{U}$ which correspond to the smallest singular values in Σ. This is equivalent to approximating $\mathbf{Y}$ by the closest matrix with rank $n_{\mathbf{u}}$.

More generally, the minimum loss method (exact local method) of Alstad et al. [2009] may be used, to handle cases with measurement noise, but this requires that we also have some "non-optimal" data:

Theorem 5 (Optimal data method with noise)**.** *Given noisy optimal measurement data* $\mathbf{Y}$ *and given "nonoptimal" data for the effect of the inputs (degrees of freedom)* $\mathbf{u}$ *on the measurements* $\mathbf{Y}$*, so* $\Delta\mathbf{y} = \mathbf{G}^{\mathbf{y}}\Delta\mathbf{u}$*, the optimal measurement combination can be determined by finding the* $\mathbf{H}$ *which minimizes* $||(\mathbf{HG}^{\mathbf{y}})^{-1}\mathbf{HY}||_F$.

Proof. Given in Alstad et al. [2009] □

Note that we want $\mathbf{HG}^{\mathbf{y}}$ to be large, that is we want to use "sensitive" measurements. With large sensitivities and little measurement noise, the contribution from the term $\mathbf{HG}^{\mathbf{y}}$ is small, and then Theorem 4 is sufficient.

The obtained $\mathbf{H}$ matrix may also give valuable insight into the operation policy. After scaling and centering of the data, the elements in the left singular vector of $\mathbf{Y}$ can be used to analyze the operation strategy. We will demonstrate this in an example from economics below.

4.4 Case studies

4.4.1 Optimal operation of a chemical reactor (use of Theorem 4)

We consider a CSTR with a reaction $A \rightleftharpoons B$, Figure 4.1 [Alstad, 2005]. The feed contains mainly component A, and the objective is to maximize the profit, which is calculated as the difference between the income from selling the product B and the cost for heating the feed: $P = p_B C_B - p_{cool} T_i^2$. Here, T_i feed temperature which can be manipulated to optimize the performance. All model parameters and equations can be found in Alstad [2005].

The feed concentrations are the main disturbances; and the concentrations and the reactor temperature are measured, so $\mathbf{y} = [C_A, C_B, T]$. The optimal operation data is obtained by applying the NCO tracking procedure as described in François et al. [2005] in combination with finite difference gradient estimates, where the input is perturbed to obtain a gradient estimate, and based on this estimate, it is adjusted to iteratively force the gradient to zero. The optimal data is gathered in sample times of 10 minutes, and collected into the data matrix $\mathbf{Y}$. A singular value decomposition $\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^\mathbf{T}$ gives $(\sigma_1, \sigma_2, \sigma_3) = (86.5, 4.8, 0.28)$. Since there is one input, T_i, we select the column in $\mathbf{U}$ corresponding to $\sigma_3 = 0.28$. This gives a controlled variable

$$\mathbf{c} = \mathbf{H}\mathbf{y} \tag{4.9}$$

with

$$\mathbf{H} = [-0.77 \ 0.63 \ 0.005]. \tag{4.10}$$

In Figure 4.2 the simulated disturbance scenario is given and Figure 4.3 shows the input usage when applying NCO tracking (to generate the optimal data) and when using a PI controller to control

$$\mathbf{c} = \mathbf{H}\mathbf{y} = -0.77 C_A + 0.63 C_B + 0.005 T \tag{4.11}$$

to zero. Due to the continuous feedback control, controlling $\mathbf{c} = \mathbf{H}\mathbf{y}$ gives much smoother input action than we have in the “optimal” data. Comparing the final profit in Figure 4.4, shows that controlling the invariant gives practically the same performance as re-optimization the system using input updates from NCO tracking.

4.4.2 Economy example (use of Theorem 4)

We consider economic indicators from 1991 to 2006 for France, Germany, Italy, Norway, UK, and USA. The data is taken from the websites[1]: `www.unece.org, www.imf.org, www.oecd.org` The “measurements” $\mathbf{y} = [y_1, \ldots, y_6]^\mathbf{T}$ for each country are interest rate (y_1), unemployment (y_2), the industrial production index (IPI, y_3), the consumer price index (CPI, y_4), tax revenue (% of GDP, y_5) and exchange rate to SDR (special drawing rights, a “lumped” currency derived from the Yen, US Dollars, British Pounds and Euros, y_6). The GDP growth, Figure 4.5, is the criterion for optimality. The measurements of year prior to the three years

[1] Accessed in March 2010

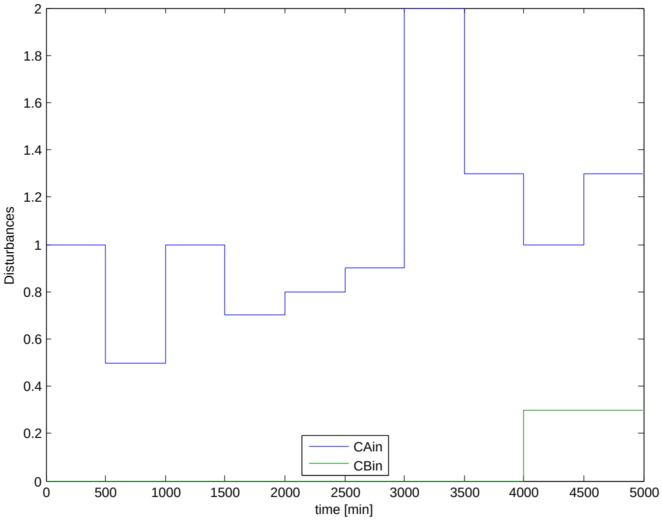


Figure 4.2: Disturbances $C_{A,in}$, $C_{B,in}$

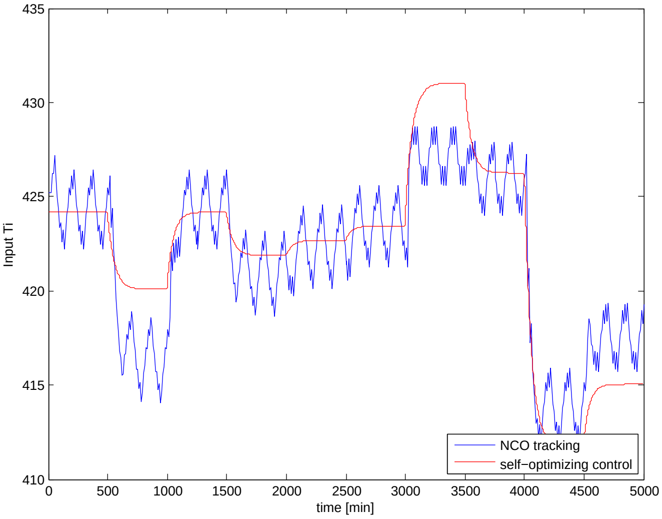


Figure 4.3: Inputs SOC and NCO tracking

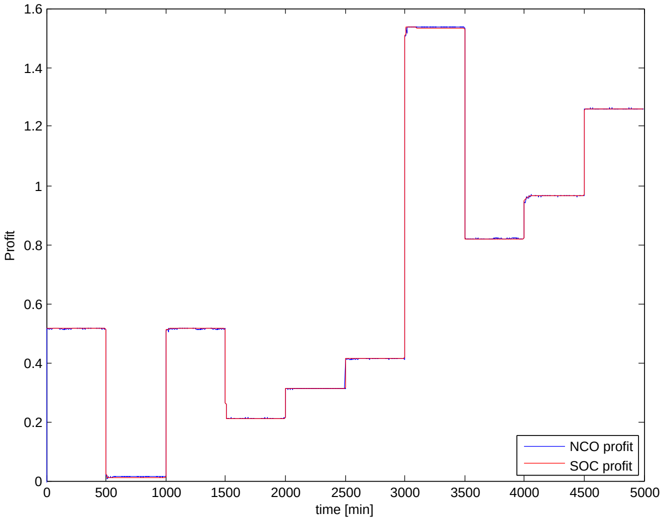


Figure 4.4: Profit Comparison

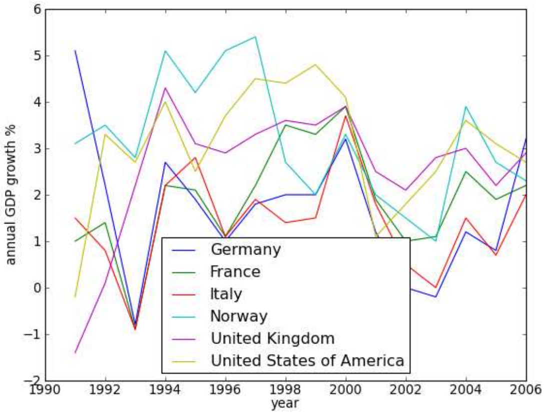


Figure 4.5: Annual GDP growth

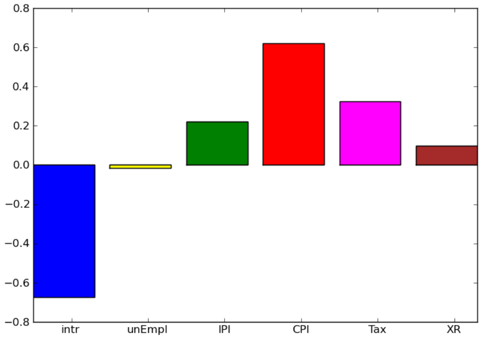


Figure 4.6: Magnitude of elements in *H*

with highest GDP growth are used for **Y**. This results in

$$\mathbf{H} = [\,\text{-}\,0.67\ \ \text{-}\,0.02\ 0.22\ 0.62\ 0.32\ 0.10]. \tag{4.12}$$

The magnitude of the elements in **H** is visualized in Figure 4.6. The most influential factors are the interest rate (-0.67) and the inflation rate (0.62). This is not unexpected, because the interest rate is used as a handle to control inflation. Of course the economics of countries is too complex to be described accurately by our selected variables, but we have shown that applying our method to economic data can reveal some of the operation strategy behind the data.

4.5 Discussion and conclusion

The proposed "null space data method" picks out the weak directions in the data **Y**, whereas other "chemometric" regression methods concentrate on the strong directions in the data [Martens and Naes, 1992]. An important reason for this is that we assume that the data is optimal, and we look for hidden combinations in this data that characterize the optimum. On the other hand, in regression methods one looks for relationships between variables X and Y. To show that the methods are different, assume our data contains two data sets, $\mathbf{Y} = [Y_1\ ,\ X]^T$ and we want to find how the relationship between Y_1 and X. We assume that $\dim(Y_1) = \dim(\mathbf{u}) = n_{\mathbf{u}}$. From our method, the problem becomes $\min_{\mathbf{H}} \| [\mathbf{H}[Y_1\ X]^{\mathrm{T}}] \|_F$.

Here, **H** is not unique, so we have that if **H** is an optimal solution, so is **DH**, where **D** is an invertible matrix [Alstad et al., 2009]. This degree of freedom may be used to set $\mathbf{H} = [\mathbf{I}\ \mathbf{H}_x]$, where **I** denotes the identity matrix. Then we optimize the problem $\min_{\mathbf{H}_x} ||\mathbf{Y}+\mathbf{H}_x X||_F$, and obtain the least squares solution $\mathbf{H}_x = -\mathbf{Y}X^{\dagger}$. Thus our method is equivalent to the normal regression methods for problems where the norm of $||\mathbf{HY}||_F$ is small, such that the contribution from the term $(\mathbf{HG}^{\mathbf{y}})^{-1}$ can be neglected, that is, for the noise free case.

However, a significant difference to standard regression methods in the case when we simply minimize $||\mathbf{HY}||_F$, is that we do not distinguish between Y_1 and X data and try to find a relationship between these, but instead focus on finding invariant variable combinations $\mathbf{c} = \mathbf{Hy} = \mathbf{H}_{\mathbf{y}} y_1 + \mathbf{H}_x x = 0$.

Our method has the advantage that it only uses data and does not rely on a model. Thus it is applicable to systems where it is very expensive or impossible to obtain an accurate model. Not even the cost function has to be known as long as the data is optimal. However, it is important that the data is consistent in the sense that the data gives the correct optimal sensitivity $\mathbf{F} = \partial \mathbf{y}^{opt}/\partial \mathbf{d}$ and contains little measurement noise.

The main drawback is that we rely on optimal data, and performance cannot be improved beyond the learning data. However, one could obtain the optimal data using some expensive method, and then analyze it to find a cheap method which gives similar performance, as is done in the CSTR example above. Other applications could be to find the “secret” of good operators or the “control strategy” of a marathon runner or of some economy.

References

Alstad, V., 2005. Studies on selection of controlled variables. Ph.D. thesis, Norwegian University of Science and Technology, Department of Chemical Engineering.

Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. Industrial & Engineering Chemistry Research 46, 846–853.

Alstad, V., Skogestad, S., Hori, E. S., 2009. Optimal measurement combinations as controlled variables. Journal of Process Control 19 (1), 138–148.

Dochain, D., Marquardt, W., Won, S. C., Malik, O., Kinnaert, M., Lunze, J., 2008. Monitoring and control of process and power systems adapting to environmental challenges, increasing competitivity and changing customer and consumer demands. status report prepared by the ifac coordinating committee on process and power systems. In: Proceedings of the 17th IFAC World Congress Seoul, Korea, July 6-11. pp. 7160–7171.

François, G., Srinivasan, B., Bonvin, D., 2005. 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.

Martens, H., Naes, T., 1992. Multivariate calibration. Wiley.

Skogestad, S., 2000. Plantwide control: The search for the self-optimizing control structure. Journal of Process Control 10, 487–507.

Chapter 5

NCO tracking and self-optimizing control in the context of real-time optimization

> “Pooh always liked a little something at eleven o’clock in the morning, and he was very glad to see Rabbit getting out the plates and mugs; and when Rabbit said, ”Honey or condensed milk with your bread? “ he was so excited that he said, ”Both, “ and then, so as not to seem greedy, he added, ”But don’t bother about the bread, please. “
>
> A. A. Milne

This paper reviews the role of self-optimizing control (SOC) and necessary conditions of optimality tracking (NCO tracking) as presented by François et al. [2005]. We show that self-optimizing control is not an alternative to NCO tracking for steady state optimization, but is to be seen as complementary. In self-optimizing control, offline calculations are used to determine controlled variables (CVs), which by use of a lower layer feedback controller, indirectly keep the process close to the optimum when a disturbance enters the process. Preferably, the setpoints are kept constant, but they may be adjusted by some optimization layer. Good CVs reduce the need for frequent setpoint changes. When selecting self-optimizing CVs, a set of disturbances must be assumed, as unexpected disturbances are not rejected in SOC.

On the other hand, the presented NCO tracking procedure adapts the inputs at given sample times without a model or any assumptions on the set of disturbances. However, disturbances with high frequencies or which do not lead to a steady state are not rejected. By using NCO tracking in the optimization layer and SOC in the lower control layer, we demonstrate that the methods complement each other, with SOC giving fast optimal correction for expected disturbances, while other disturbances are compensated by the model free NCO tracking procedure on a slower time scale.

Submitted to Journal of Process control. Based on the Keynote lecture at DYCOPS 2010, Leuven

5.1 Introduction

Most processes in industrial practice are operated in such a way that the operators set the setpoints for PID controllers to keep the controlled variables (CVs) at the desired setpoint. Which measurements are chosen as CVs is mostly decided based on process knowledge and best practices. However, due to stronger competition and environmental regulations, it has become increasingly important to operate the processes close to optimality. In many cases, steady state operation accounts for the largest part of the operating cost, and significant economical improvements can be achieved by operating the plant optimally at steady state.

Depending on how this is realized, the methods for achieving optimal process operation generally may be categorized into one of the following three categories:

- Model used online (e.g. Real-time optimization (RTO))
- Model used offline (e.g. self-optimizing control (SOC))
- Explicit Model not used (e.g. NCO tracking)

In all cases, measurements are collected online, with the aim of driving the process towards optimality. In the first approach, online optimization, measurements from the process are used together with a mathematical model to determine the optimal setpoints by solving an optimization problem online [Marlin and Hrymak, 1997].

In the offline approach, expensive online computations are avoided, and optimal operation is achieved by designing a “smart” control structure. This controlled variable (CV) selection procedures objective is to transform the economic objectives into control objectives [Morari et al., 1980]. A process model is used to support decision making in control structure design, but it will not be used online. Self-optimizing control [Skogestad, 2000] belongs into this category.

A third strategy avoids using an explicit process model, but uses measurements in order to obtain gradient information about the process. This information is used to update the inputs in order to obtain optimal operation. Necessary conditions of optimality tracking (NCO tracking) [François et al., 2005] and extremum seeking control [Krstic and Wang, 2000] represent this category. This idea is relatively old [Draper and Li, 1951], but has recently gained increased attention.

These approaches to achieve steady state optimal operation have been developed by research groups with different backgrounds for different kind of problems. The authors feel that there has been some confusion about the use, interplay, applicability and practicability of some of the concepts.

Our paper is structured as follows: The next two sections briefly describe the ideas from self-optimizing control and NCO tracking. In particular, this work focuses on the null space method as described in Alstad and Skogestad [2007], which uses a model offline, and the model free NCO tracking procedure for steady state optimization as described by François et al. [2005].

In Section 5.4 we describe the framework in which we place the two methods and consider the properties of the two approaches. Based on this discussion, we consider the methods as *complementary* and propose to use them together. The ideas are illustrated by simulation results for a dynamic CSTR in Section 5.5, followed by a discussion in Section 5.6, and conclusions, Section 5.7.

5.2 Self-optimizing control

In virtually all practical cases, plant operation is subject to operational and safety constraints, and the problem of achieving steady state optimal operation can be formulated as

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{x},\mathbf{d}) \quad \text{s.t.} \quad \begin{cases} \text{plant: } g(\mathbf{u},\mathbf{x},\mathbf{d}) = 0 \\ \text{constraints: } h(\mathbf{u},\mathbf{x},\mathbf{d}) \leq 0, \end{cases} \tag{5.1}$$

where $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$ denotes the vector of adjustable input variables (e.g. a valve opening or a pump speed), $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$ denotes the states, and $\mathbf{d} \in \mathbb{R}^{n_\mathbf{d}}$ denotes the vector of unknown disturbances and parameters. The vector valued functions g and h denote the model equations and the operational constraints, respectively.

In practice, not all constraints are active during optimal operation of the plant and some constraints will remain inactive. In terms of plant safety and economy it is often significantly more important to satisfy the active constraints than to handle the unconstrained degrees of freedom optimally. Therefore, the first step when designing the control structure is to determine the active constraints, and to control

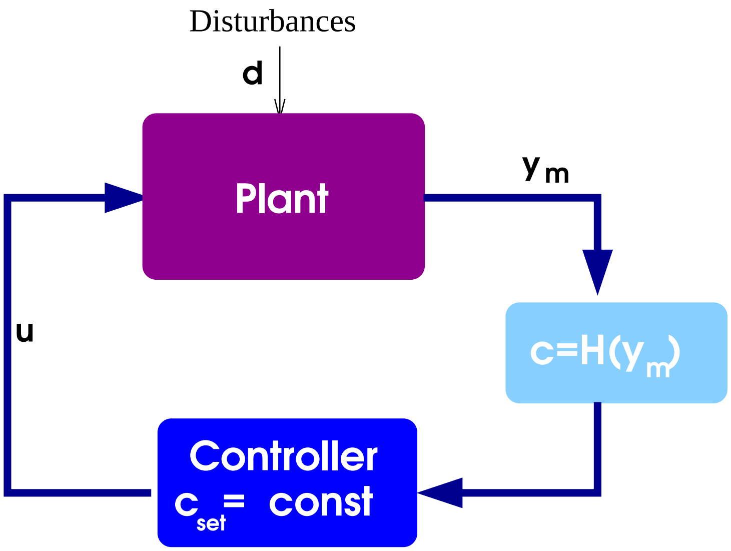


Figure 5.1: Block diagram SOC

them using some kind of (feedback) controller. After all active constraints have been implemented, problem (5.1) can be re-written as an unconstrained optimization problem,

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}), \tag{5.2}$$

where, by abuse of notation, **u** now denotes the remaining unconstrained degrees of freedom.

The term self-optimizing control refers to the procedure of selecting the controlled variables, which are controlled by some feedback controller (Figure 5.1). The focus is set on selecting the best controlled variables $\mathbf{c} = \mathbf{H}\mathbf{y}_m$ such that the operating cost $J(\mathbf{u},\mathbf{d})$ is minimized. Here $\mathbf{y}_m$ denotes the vector of measurements, and **H** is a selection or combination matrix. The criterion for evaluating different candidates for controlled variables is the loss from optimality

$$L = J(\mathbf{u},\mathbf{d}) - J(\mathbf{u}^{opt}(\mathbf{d}),\mathbf{d}), \tag{5.3}$$

which is imposed by the disturbance **d**, and the selected control structure which determines how **u** is adjusted. Using the loss L, Skogestad [2000] defined:

> Self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables (without the need to re-optimize when disturbances occur).

The ideal self-optimizing variable candidate for this kind of controlled variable would be the gradient $\mathbf{c} = J_{\mathbf{u}}(\mathbf{u}, \mathbf{d}) = \frac{\partial J}{\partial \mathbf{u}}$, which should be zero for optimal operation under all disturbances.

This was already formulated in Halvorsen and Skogestad [1997], where it is written:

> ...Thus the search is now reduced to find some measurement function $h(\mathbf{u}, \mathbf{d})$ with these required properties. An example of this kind of ideal measurement function is in fact the gradient of the criterion function.

This idea has been also mentioned in Halvorsen and Skogestad [1999], where the authors write of the gradient as a controlled variable. It satisfies the conditions of not being at a constraint, while the optimal value does not vary with changing disturbances. Controlling invariants, in particular the gradient of a process, has been proposed by other authors, too, see e.g. Bonvin et al. [2001] and Cao [2005].

However, in most cases, the gradient cannot be measured, for example, because it is a function of the unknown disturbances $\mathbf{d}$. The definition of self-optimizing control [Skogestad, 2000] includes the special case of gradient control, while leaving room for "suboptimal cases" in which the gradient cannot be determined exactly from measurements. In some cases it might be desirable to control only single measurements, or to exclude a set of measurements. Then the gradient will not be zero and the loss L provides an objective selection criterion. In other words, a self-optimizing control structure may be considered the best possible (in terms of the loss L) approximation to the unmeasured gradient $J_{\mathbf{u}}$ using the available measurements.

Several methods for finding self-optimizing variables have been reported in the literature [Halvorsen et al., 2003; Alstad and Skogestad, 2007; Kariwala et al., 2008; Alstad et al., 2009]. All these methods are based on a approximating the optimization problem (5.2) by a quadratic optimization problem

$$\min_{\mathbf{u}} [\mathbf{u}^{\mathrm{T}} \quad \mathbf{d}^{\mathrm{T}}] \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}, \tag{5.4}$$

and by using a linear measurement model

$$\mathbf{y} = \mathbf{G^y u} + \mathbf{G^y_d d}. \tag{5.5}$$

The goal is to find a matrix $\mathbf{H}$ such that the controlled variable $\mathbf{c}$ is

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m, \tag{5.6}$$

where the measured quantities are defined as

$$\mathbf{y}_m = \mathbf{y} + \mathbf{n^y} \tag{5.7}$$

with the measurement noise $\mathbf{n^y}$. Without measurement noise, $\mathbf{n^y} = 0$, we have that $\mathbf{y}_m = \mathbf{y}$; thus we write the controlled variable

$$\mathbf{c} = \mathbf{Hy}. \tag{5.8}$$

It is assumed that the inputs $\mathbf{u}$ are adjusted by a feedback controller to keep $\mathbf{c}$ at its setpoint $\mathbf{c}_s$. If the controller has integral action, then $\mathbf{c} = \mathbf{c}_s$ at steady state. In the case of single measurements, each row of $\mathbf{H}$ contains only one entry, whereas if combinations of measurements are allowed, $\mathbf{H}$ will be a full matrix.

5.2.1 Self-optimizing control using the null space method

In the following we describe the null space method [Alstad and Skogestad, 2007] for determining a controlled variable, which is a combination of measurements. We present a reformulation of the null space theorem [Alstad et al., 2009].

Theorem 6. *Given a sufficient number of measurements ($n_\mathbf{y} \geq n_\mathbf{u} + n_\mathbf{d}$) and no measurement noise $\mathbf{n^y} = 0$, select $\mathbf{H}$ in the null space of the optimal sensitivity matrix $\mathbf{F}$,*

$$\mathbf{HF} = 0, \tag{5.9}$$

where

$$\mathbf{F} = \frac{\partial \mathbf{y}^{\text{opt}}}{\partial \mathbf{d}}. \tag{5.10}$$

Controlling $\mathbf{c} = \mathbf{Hy}$ to zero yields locally zero loss from optimal operation.

To obtain the optimal sensitivity matrix $\mathbf{F}$ there are several possibilities. It can be obtained numerically by re-optimization of a process model, or calculated using

$$\mathbf{F} = -\mathbf{G^y}\mathbf{J_{uu}}^{-1}\mathbf{J_{ud}} + \mathbf{G^y_d}, \tag{5.11}$$

where $\mathbf{J_{ud}} = \partial^2 J/(\partial \mathbf{u} \partial \mathbf{d})$ and $\mathbf{J_{uu}} = \partial^2 J/\partial \mathbf{u}^2$, and we use the linearized process model (5.5). Alternatively, one could run experiments, or use optimal measurement data as shown in Chapter 4.

We sketch a proof: In the neighborhood of the nominal point $\mathbf{d}^{\text{nom}}$ the optimal change in the measurements can be expressed using (5.10) as

$$\mathbf{y}^{\text{opt}}(\mathbf{d}) - \mathbf{y}^{\text{opt}}(\mathbf{d}^{\text{nom}}) = \mathbf{F}(\mathbf{d} - \mathbf{d}^{\text{nom}}). \tag{5.12}$$

The optimal variation in the controlled variables $\mathbf{c}$ then becomes

$$\mathbf{c}^{\text{opt}}(\mathbf{d}) - \mathbf{c}^{\text{opt}}(\mathbf{d}^{\text{nom}}) = \mathbf{HF}(\mathbf{d} - \mathbf{d}^{\text{nom}}), \tag{5.13}$$

and since $\mathbf{H}$ is chosen in left null space of $\mathbf{F}$, we have $\mathbf{c}^{\text{opt}}(\mathbf{d}) = \mathbf{c}^{\text{opt}}(\mathbf{d}^{\text{nom}})$ for any disturbance $\mathbf{d}$, and thus we do not need to change the setpoint for $\mathbf{c} = \mathbf{Hy}$.

In Appendix 5.A we show that choosing $\mathbf{H}$ in the left null space of $\mathbf{F}$ is in indeed identical to selecting $\mathbf{c} = J_{\mathbf{u}}$, where $J_{\mathbf{u}} = \partial J/\partial\mathbf{u}$ is the gradient of (5.4).

However, when the measurements are corrupted by biased noise $\mathbf{n}^{\mathbf{y}}$, the null space method will not give the best possible solution. To find the best controlled variable with biased process noise on the measurements, we refer to Alstad et al. [2009].

5.3 NCO tracking

Necessary conditions of optimality (NCO) tracking is a general framework that turns a (dynamic or static) optimization problem into a control problem. It uses the fact that at the optimal operating point, the first order necessary optimality conditions must hold. Basically, the necessary conditions of optimality are the controlled variables. This general concept has been applied both to dynamic optimization problems (e.g. Srinivasan et al. [2003a,b],Bonvin et al. [2005]; Kadam et al. [2007]), and static optimization (e.g. François et al. [2005]; Srinivasan et al. [2008]; Gros et al. [2009]).

For steady state optimization, the Karush, Kuhn Tucker conditions [Bazaraa et al., 2006] represent the optimality conditions. If the sensitivities are available as online measurements (or estimates), they may be controlled by using a continuous feedback controller, such as a PI controller. Alternatively, the inputs may be updated iteratively until the NCO are satisfied. To the authors' knowledge, all publications on static optimization using NCO tracking have been applying discrete input updates for iteratively approaching the steady state optimal input value which satisfies the NCO (at least for the unconstrained part of the NCO).

5.3.1 NCO tracking procedure as described by François et al. [2005]

In this paper we will be referring to NCO tracking as described in François et al. [2005]. This is a truly measurement based optimization method, which does not rely on any process model. Instead of controlling "normal" measurements $\mathbf{y}$, the gradient is measured (or estimated), and used as a controlled variable. When a disturbance enters the process, the NCO tracking control scheme adapts the inputs iteratively such that the NCO are satisfied after some iterations. The block diagram is given in Figure 5.2.

We do not present the general NCO tracking procedure (with constraints) here, but we rather give a derivation of the special case without constraints, i.e. only the sensitivity seeking directions, as applied in e. g. Srinivasan et al. [2008]. Then the optimization problem in consideration is

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}), \tag{5.14}$$

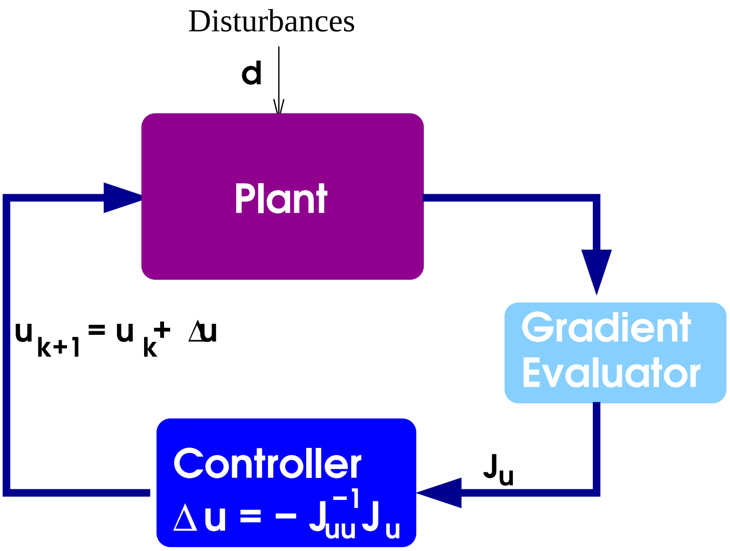


Figure 5.2: Block diagram NCO tracking

where **u** and **d** are defined as above. Omitting to write the explicit dependence on **d**, the first order necessary condition for optimality is:

$$J_{\mathbf{u}}(\mathbf{u}) = \frac{\partial J(\mathbf{u})}{\partial \mathbf{u}} = 0. \tag{5.15}$$

To achieve optimal operation, we update the input **u** at each sample time k using the update equation

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta\mathbf{u}, \tag{5.16}$$

until (5.15) is satisfied. To obtain the update term $\Delta\mathbf{u}$, we linearize (5.15) around the current operating point $\mathbf{u}_k$,

$$J_{\mathbf{u}}(\mathbf{u}_k + \Delta\mathbf{u}) = J_{\mathbf{u}}(\mathbf{u}_k) + \mathbf{J}_{\mathbf{uu}}(\mathbf{u}_k)\Delta\mathbf{u}. \tag{5.17}$$

Since we want the update $\Delta\mathbf{u}$ to force the sensitivity to zero, we set the left hand side of (5.17) to zero and solve for $\Delta\mathbf{u}$ [François et al., 2005].

$$\Delta\mathbf{u} = -\mathbf{J}_{\mathbf{uu}}{}^{-1}(\mathbf{u}_k)J_{\mathbf{u}}(\mathbf{u}_k) \tag{5.18}$$

This Newton update step is exact for a quadratic approximation of the system (5.14), in the sense that the NCO (5.15) are satisfied after one iteration. In practice we do not apply the full update step $\Delta\mathbf{u}$, because this may lead to feasibility and convergence problems as the process can move outside the region where the quadratic approximation is valid. To avoid this, the update term $\Delta\mathbf{u}$ is multiplied by some tuning parameter $\beta \in [0\,1]$, such that $\mathbf{u}_{k+1} = \mathbf{u}_k + \beta\Delta\mathbf{u}$.

To evaluate (5.18) we need the derivative $J_{\mathbf{u}}(\mathbf{u}_k)$ for a given input $\mathbf{u}_k$. In this work it is chosen to make a small perturbation in the input and to run the process for a given time to estimate the gradient by finite differences. The magnitude of the perturbation is desired to be small in order not to upset the process excessively. At the same time it has to be larger than the process noise to yield sufficient information about the descent direction.

Since the Hessian $\mathbf{J_{uu}}(\mathbf{u}_k)$ is difficult to obtain, it is often determined once at the nominal operating point. Alternatively, as we choose to do in this work, an approximation of the inverse of the Hessian can be obtained by a BFGS update scheme. The NCO tracking algorithm is summarized in Figure 5.3. This procedure is analog to a Newton(like) method in optimization. In the analogy, the steady state operating periods correspond to function evaluations in the newton procedure, and the solution is found when the NCO hold.

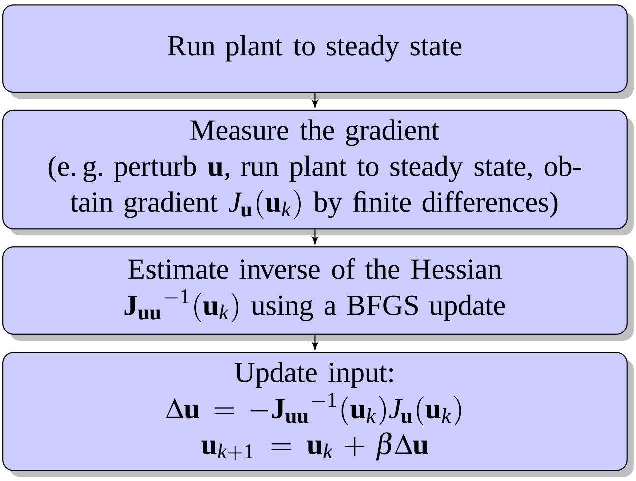

Run plant to steady state
Measure the gradient
(e. g. perturb $\mathbf{u}$, run plant to steady state, obtain gradient $J_{\mathbf{u}}(\mathbf{u}_k)$ by finite differences)
Estimate inverse of the Hessian
$\mathbf{J_{uu}}^{-1}(\mathbf{u}_k)$ using a BFGS update
Update input:
$\Delta\mathbf{u} = -\mathbf{J_{uu}}^{-1}(\mathbf{u}_k)J_{\mathbf{u}}(\mathbf{u}_k)$
$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta\Delta\mathbf{u}$

Figure 5.3: Simple NCO tracking procedure

Just like any (quasi) Newton method, NCO tracking depends crucially on the availability of good gradient estimates. Beside estimating the gradients using input perturbations and finite differences, there exist other methods, which do not require frequent perturbations. In Roberts [2000], past inputs are used in Broyden's formula to obtain the gradients. Other methods which do not rely on input perturbations are described in Brdyś and Tajewski [1994] and Gao and Engell [2005]. However, in this work, the authors choose to use finite differences because of its simplicity. Avoiding input perturbations for gradient estimation will result in less nervous process operation, however, inputs will still be updated iteratively, but only at given sample times.

Remark 2 (Recent advances in NCO tracking)**.** *The gradient does not have to be obtained using finite differences. Any other method to measure or estimate the gradient is possible. The gradient may for example be estimated by a model, as is done in the null space method. Recently, the idea of NCO tracking has been extended to the case where the gradient estimate is based on output feedback. Gros and coworkers, Gros et al. [2009] use a linear measurement model to eliminate the disturbance from the gradient expression. Their results are based on the same idea as the null space method, which uses a measurement model to eliminate unknown disturbances and internal states from the gradient expression. However, Gros et al. [2009] goes one step further and determines the actual change in the inputs. This step is omitted in the self-optimizing control structure design context, because the focus is set on finding good controlled variables, and the generation of the corresponding inputs,* $\mathbf{u}$*, is left to the feedback controller.*

The authors of Gros et al. [2009] consider zero mean noise, and show that if the model is invertible and the number of unknowns ($n_{\mathbf{d}}$*) is lesser than or equal to the number of measurements (*$n_{\mathbf{y}}$*), the inputs can be updated to converge to the optimum.*

In the case of biased noise, neither the null space method nor the NCO tracking modifications introduced by Gros et al. [2009] will give the best achievable operation. In this case it is necessary to use other methods which find a trade-off between the loss caused by the disturbance and the loss caused by the measurement offset. A method applicable in this case is the "minimum loss method" [Alstad et al., 2009].

5.4 Self-optimizing control and NCO tracking in relation to each other

In order to suggest how to combine self-optimizing control as described in Alstad and Skogestad [2007] and NCO tracking as in [François et al., 2005], we first consider how a chemical plant is usually operated today.

5.4.1 Time scale separation of the overall control system

The control structure of a complete chemical plant can be decomposed vertically into different layers, which operate on different time scales, [Findeisen et al., 1980; Skogestad, 2004]. Each control layer implements the setpoints which are given from the layer above, Figure 5.4.

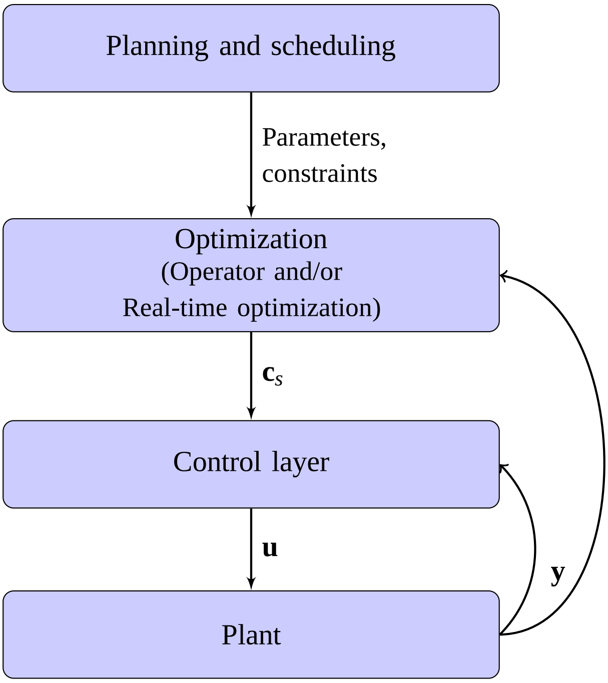


Figure 5.4: Vertical decomposition into control layers

The top layer consists of planning and scheduling. This includes management decisions on e.g. the product specifications, and on profit and safety parameters, such as utility prices and constraints. Usually this layer has a time scale of weeks or days, strongly depending on the type of process and the production scale.

The optimization layer is located below the planning and scheduling layer and implements the goals given from the planning and scheduling layer. In most plants this is done by operators, but in recent years, online optimization (RTO) has been increasingly used to find good setpoints for the controlled variables of the lower layers. However, this can become complicated as it involves several difficult steps such as steady state detection, data estimation and reconciliation and solving a large nonlinear optimization problem. Once the optimization problem has been solved successfully or the operators have decided to change the setpoints, the new setpoints are passed on to the control layer and implemented. It is typical for this layer that the setpoints are updated at discrete time instances with update intervals in the time scale of several hours.

The control layer below the optimization layer generally consists of PID controllers or model predictive controllers (MPC), which act directly on the plant inputs $\mathbf{u}$. This layer has a time scale ranging from fractions of seconds up to minutes and to a few hours. Finally, the plant layer contains the actual plant, but usually with some stabilizing (regulatory) control loops.

When a disturbance enters the system, the control layer will try to keep the setpoints of the controlled variables to their original setpoints. After the plant has settled down and (suboptimal) steady state has been reached (and detected), the operator may adjust the setpoints based on experience, or the real-time optimizer may re-calculate the setpoints. Then the setpoints of the controlled variables are ramped to their new values, and the plant has to settle down again. The long time delay between start of the disturbance and reaching the final optimized operation point is one of the challenges for the optimization layer. In particular if RTO is used, it is not possible to counteract disturbances which occur on a fast time scale [Engell, 2007]. This limits successful RTO applications for cases with sporadic disturbances, which, after a short transition period, lead to a new steady state, e.g. step changes in the plant throughput or the like. Disturbances occurring on a faster time scale cannot be detected and rejected in RTO implemented as described above.

Using a dynamic model in the real-time optimization with an economic objective function would allow setpoint changes without having to wait for steady state. However, practical obstacles have prevented the dynamic RTO (DRTO) from becoming a standard tool in process industries. The main problems arise from the reliability of the information used in the DRTO, because good models are difficult to obtain and maintain with justifiable efforts. In addition, the state estimation causes additional challenges. Even if a good model and states are available in the DRTO, the dynamic optimization problem itself is difficult to solve.

5.4.2 Properties of self-optimizing control and NCO tracking

Both methods, NCO tracking and self-optimizing control pursue the same goal, namely minimization of the operating cost. The main differences between Alstad and Skogestad [2007] and unconstrained NCO tracking as in François et al. [2005] are that in François et al. [2005] the gradient is "measured", and the inputs are updated iteratively, while in Alstad et al. [2009], a model is used to predict the gradient from outputs, and a continuous PI controller is used to manipulate the corresponding inputs.

In self-optimizing control, we focus on finding controlled variables which do not need frequent updates, such as the gradient, for example. However, since the gradient is usually not available as a measurement, self-optimizing control does not in general aim for controlling the gradient of the plant to zero, but to find controlled variables, which give acceptable operation. The self-optimizing variables are kept at their setpoints by feedback controllers, so there is no need for solving for the optimal inputs explicitly.

In summary, we may say that the NCO tracking procedure by François et al. [2005] measures the gradient, and works as a controller which calculates the required (steady state) input change $\Delta\mathbf{u}$ directly. In self-optimizing control [Alstad and Skogestad, 2007], we are not interested in the inputs, as they are taken care of by the controllers. We are rather interested in finding the right controlled variable $\mathbf{c} = \mathbf{H}\mathbf{y}$, which when kept constant, leads to the correct input action $\mathbf{u}$.

In Table 5.1 we have listed the main differences between the null space method and NCO tracking.

5.4.3 Using self-optimizing control and NCO tracking together

The previous observations lead us to consider self-optimizing control and NCO tracking as described by François et al. [2005] (or RTO) as *complementary*, and use them together. The NCO tracking fits better into the optimization layer and is thus an alternative to model based real-time optimization (RTO), while self-optimizing control should be used in the lower layer and follow the setpoints coming from the NCO tracking layer, as shown in Figure 5.5.

It may be argued that if NCO tracking or an RTO system is installed, there is no need to select a self-optimizing control structure because the setpoints are updated by the optimization layer. However, this combination of an RTO layer (or NCO tracking) and self-optimizing control avoids the shortcomings of conventional RTO:

Table 5.1: Summary of properties

Null space method [Alstad and Skogestad, 2007]	NCO tracking [François et al., 2005]
Procedure for finding $\mathbf{c} = \mathbf{H}\mathbf{y}$	Controlled variable: $\mathbf{c} = J_{\mathbf{u}}$
$J_{\mathbf{u}}$ not measured	$J_{\mathbf{u}}$ measured
Set of important $\mathbf{d}$ assumed a priori	No assumptions on disturbances
$\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$ obtained from model	No model needed
Near-optimal for expected disturbances	Optimal for unexpected disturbance
Linearized at nom. point	Linearization point moves
Fast (feedback)	Slow (acts only at sample times)

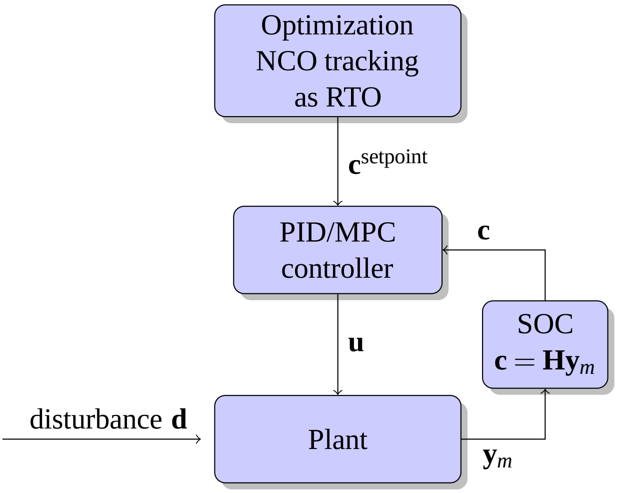


Figure 5.5: Relation between NCO tracking and self-optimizing control

1. The use of self-optimizing controlled variables enables a faster optimal reaction to expected (main) disturbances, not only at sample times.

2. The RTO has to change the setpoints less frequently.

Infrequent RTO updates result in fewer complex operations such as steady state detection, data reconciliation, and solving the resulting nonlinear optimization problems. At the same time, the self-optimizing control structure can benefit significantly from an RTO system or NCO tracking controller on top of it. One reason is that the null space method is based on a model and expected disturbances, whereas NCO tracking can also handle unknown disturbances. Another reason is that if a disturbance moves the process far from the linearization point, the local model approximation in the null space method may be poor. Therefore, the self-optimizing control structure cannot reject unexpected disturbances or disturbances which move the process far away from the linearization point. They have to be counteracted by re-optimization of the system.

In summary, it is recommended to always use a self-optimizing control layer underneath the optimization layer instead of directly computing the plant input **u**. This rejects expected disturbances on a fast timescale, while the unexpected disturbances are rejected by the NCO tracking/RTO layer updates. Applying self-optimizing control is thus an intelligent way to implement the control layer below the RTO layer.

Remark 3. *Even though we here have used the flavor of NCO tracking by François et al. [2005], our conclusion may be generalized further to combine a model based method in a lower control layer, which can act swiftly to compensate for expected disturbances; and to use a model free method on top reduce the effect of model mismatch and to reject unexpected disturbances by adjusting the setpoints of the lower layer on a slower time scale. Thus, also methods such as e.g. extremum seeking [Ariyur and Krstic, 2003] could be used in the upper layer.*

5.5 Case study

5.5.1 Model

To illustrate the ideas above, we present simulation results for a dynamic CSTR with a feed stream F containing mainly the component A, and a reversible chemical reaction $A \rightleftharpoons B$, see Figure 5.6. The process model is taken from Economou and Morari [1986], and the dynamics of the system are described by following set of

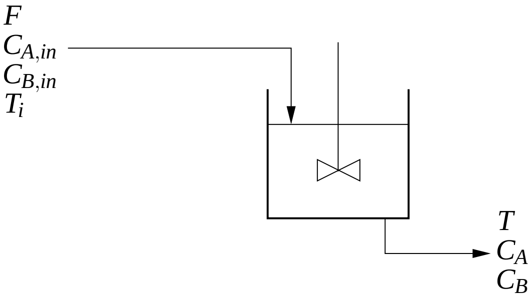


Figure 5.6: Schematic diagram of a CSTR

equations,

$$\frac{dC_A}{dt} = \frac{1}{\tau}(C_{A,in} - C_A) - r, \tag{5.19}$$

$$\frac{dC_B}{dt} = \frac{1}{\tau}(C_{B,in} - C_B) + r, \tag{5.20}$$

$$\frac{dT}{dt} = \frac{1}{\tau}(T_i - T) + \frac{-\Delta H_{rx}}{\rho c_p} r, \tag{5.21}$$

where C_A, C_B, T, and T_i denote the concentrations of components A and B, the reactor temperature and the feed inlet temperature, respectively. Further, τ is the residence time, ρ is the density, c_p is the heat capacity, and $-\Delta H_{rx}$ is the reaction enthalpy. The reaction rate r is defined by

$$r = k_1 C_A - k_2 C_B \tag{5.22}$$

where

$$k_1 = A_1 e^{\frac{-E_1}{RT}} \quad \text{and} \quad k_2 = A_2 e^{\frac{-E_2}{RT}}, \tag{5.23}$$

and A_1 and A_2 are the Arrhenius factors for the reaction constants k_1 and k_2.

This process has one manipulated input (u), the inlet temperature T_i. The expected disturbances d_1 and d_2 enter the process as variations in the feed concentrations $C_{A,in}$ and $C_{B,in}$, and the measured variables are

$$\begin{aligned} y_1 &= C_A, \\ y_2 &= C_B, \\ y_3 &= T. \end{aligned} \tag{5.24}$$

The objective is to maximize the profit function, which is the difference between the income from selling the product B and the cost for heating the feed (Alstad [2005]):

$$P = [p_{C_B} C_B - (p_{T_i} T_i)^2], \tag{5.25}$$

Table 5.2: Objective function parameters

Parameter	Value
p_{C_B}	2.009
p_{T_i}	$1.657 \cdot 10^{-3}$

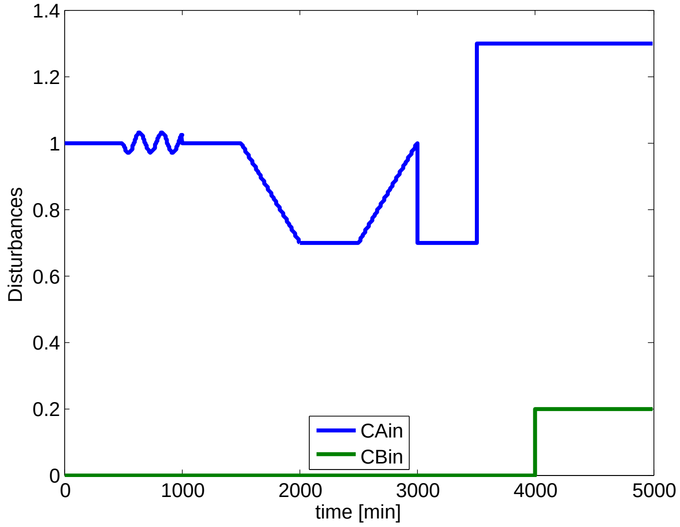


Figure 5.7: Disturbance trajectories $C_{A,in}, C_{b,in}$

Here p_{C_B} is the price of the desired product B and p_{T_i} is the cost for heating. The parameter values are given in Table 5.2, and the nominal operation values for all variables are listed in Table 5.3.

5.5.2 Simulations

First, we control the process for the expected disturbances using direct NCO tracking [François et al., 2005]. Next, we use a self-optimizing controlled variable $\mathbf{c} = \mathbf{H}\mathbf{y}_m$, obtained the null space method and compare the results with direct NCO tracking. After comparing both control structures for an unexpected disturbance, we finally combine the methods as shown in Figure 5.5.

The expected disturbance scenario is given in Figure 5.7. After 500 minutes at the nominal value, the concentration $C_{A,in}$ (d_1) varies sinusoidal before returning to its nominal value. Then ramp disturbances in $C_{A,in}$ are introduced, followed by

Table 5.3: Nominal values for the CSTR model

Variable	Value	Unit	Description
T_i	424.20	K	Feed temperature (input u)
C_A	0.4978	mol/l	Concentration A in product (y_1)
C_B	0.5022	mol/l	Concentration B in product (y_2)
T	426.71	K	Reactor temperature (y_3)
$C_{A,in}$	1.000	mol/l	Concentration A in feed, (d_1)
$C_{B,in}$	0.000	mol/l	Concentration B in feed, (d_2)
F	1.000	holdup min^{-1}	Flow rate
A_1	5000	s^{-1}	Arrhenius factor 1
A_2	$1 \cdot 10^6$	s^{-1}	Arrhenius factor 2
c_p	1000	cal $kg^{-1}\mathrm{K}^{-1}$	Heat capacity
E_1	10000	cal mol^{-1}	Activation energy 1
E_2	15000	cal mol^{-1}	Activation energy 2 (d_3)
R	1.987	cal $\mathrm{mol}^{-1}\mathrm{K}^{-1}$	Ideal gas constant
$-\Delta H_{rx}$	5000	cal mol^{-1}	Heat of reaction
ρ	1.000	kg/l	Density
τ	1.000	min	Residence time

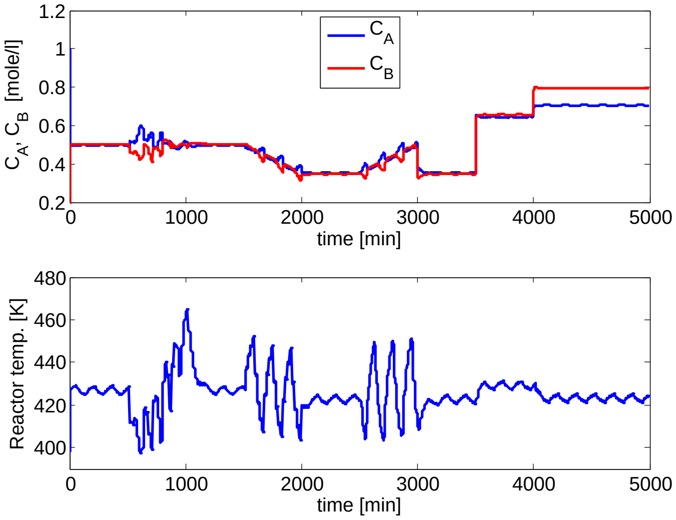


Figure 5.8: NCO tracking, concentrations and temperature

large step disturbances. At 4000 minutes, the concentration $C_{B,in}$ (d_2) makes a step change of 0.2 mol/l. The non-steady state periods (sinusoid and ramp) are included to test how the controller behaves in these cases. Note that strictly speaking, the gradient is not defined when the process is not at steady state.

Direct NCO tracking

To obtain the gradient information, the input T_i is perturbed with a step of 1 K. Starting with a positive value, the sign is altered every fourth NCO iteration. Changing the sign of the perturbation was found to give better overall performance of the NCO procedure. No steady state detection is implemented in the NCO tracking procedure. Instead, a step test is used to determine the approximate time for the system to settle down to a new steady state. At the nominal point, the system has a time constant of less than two minutes for an input step of $\Delta T_i = 5$ K. To let the system settle down far from the nominal point, where the system dynamics are different, a sample time of 10 minutes is chosen for the direct NCO tracking procedure. The step size parameter β is set to 0.4.

Figure 5.8 shows the concentration and temperature trajectories for the NCO tracking procedure. The control strategy enables acceptable control. It is furthermore found that the step disturbances are very well handled. Since the method assumes steady state after 10 minutes, and uses the results at each sample time for calculating the input update, it has difficulties handling sinusoidal and ramp disturbances which do not lead to a steady state. However, the controller manages to keep the system stable during these periods. The performance of the NCO tracking algorithm is very sensitive to the tuning parameter β, the sample time, and timing and kind of disturbance, and of course the perturbation for estimating the gradient.

Self-optimizing control using the null space method

Next, the process is controlled using the null space method from Section 5.2. Since we have one input and 2 disturbances to compensate for, we need three measurements for the invariant variable combination, so $\mathbf{y} = [\ C_A \quad C_B \quad T\ ]^{\mathrm{T}}$. We optimize the steady state system at the nominal operating point and then introduce small perturbations in the disturbance variables $\mathbf{d} = [C_{A,in} C_{B,in}]^{\mathrm{T}}$. After re-optimizing we calculate

$$\mathbf{F} = \frac{\partial \mathbf{y}^{\mathrm{opt}}}{\partial \mathbf{d}} = \begin{bmatrix} -0.4862 & -0.3223 \\ -0.5138 & -0.6777 \\ -9.9043 & 40.5807 \end{bmatrix}. \tag{5.26}$$

Then with $\mathbf{H} = [\ -0.7688 \quad 0.6394 \quad 0.0046\ ]$ we have that $\mathbf{HF} = 0$. Using a PI controller, the self-optimizing variable $\mathbf{c} = \mathbf{Hy}_m = -0.7688C_A + 0.6394C_B + 0.0046T$ is controlled at a constant setpoint (zero if we use the deviation from nominal steady state). The concentration and temperature trajectories with self-optimizing control are plotted in Figure 5.9. Compared with the concentrations and temperature using NCO tracking, the trajectories are much smoother.

Comparing inputs and profit for NCO tracking and self-optimizing control

As may be seen from the trajectories for NCO tracking and self-optimizing control, the input usage for the two cases is quite different, Figure 5.10. While the NCO tracking procedure needs large input variations to estimate the gradient and to iteratively update the input, the input usage of the self-optimizing control structure is very moderate and smooth. Especially during the non-steady state disturbances, the NCO tracking changes the input excessively.

Comparing the profits, Figure 5.11 shows that both systems are very similar in the steady state periods, but for disturbances where no steady state is reached within one sample time, NCO tracking is not performing as well as the self-optimizing control policy using the null space method.

Using NCO tracking as RTO and self-optimizing control in the lower layer

If it can be guaranteed that the disturbances in the feed concentration are the only ones entering the process, then using only self-optimizing control is sufficient, and an RTO layer is not necessary. However, the situation changes for disturbances

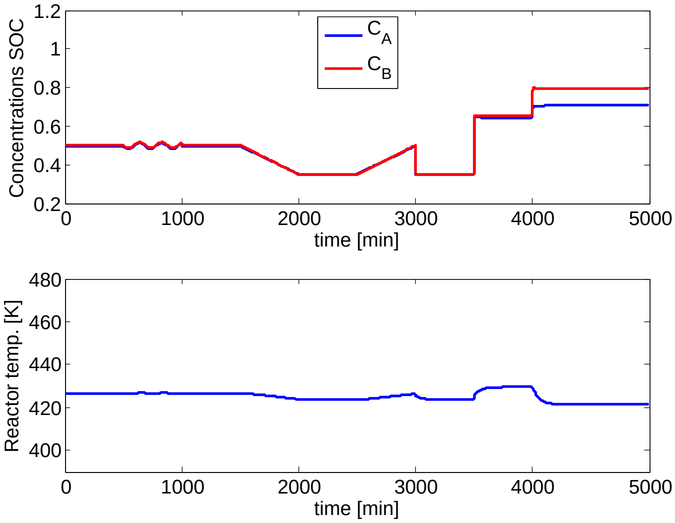


Figure 5.9: SOC, concentrations and temperature

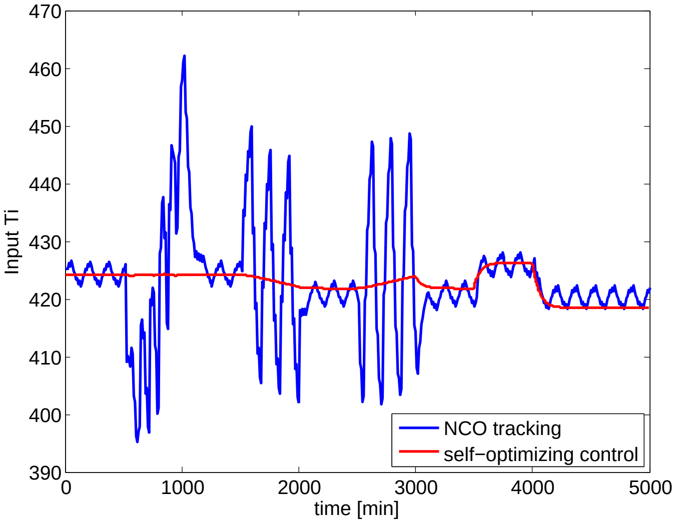


Figure 5.10: Input usage for SOC and NCO tracking

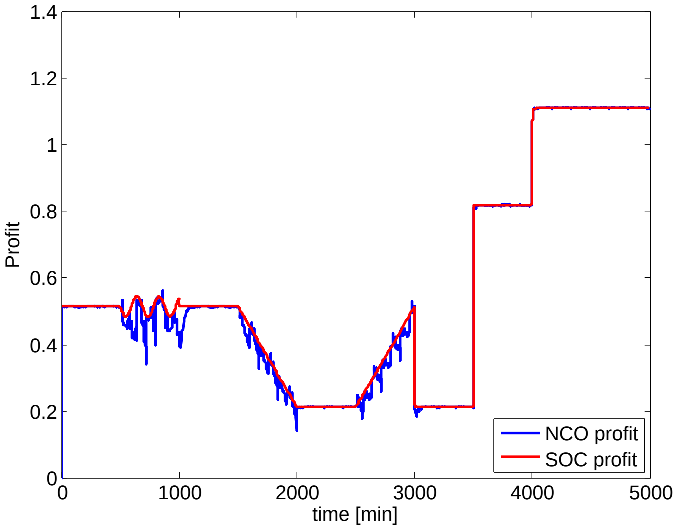


Figure 5.11: Profit for SOC and NCO tracking

not anticipated in the control structure design. Consider a positive step change in the activation energy E_2 (d_3) of 3% at time 3100 min. This disturbance reduces the reaction rate for the reverse reaction, especially at higher temperatures. Comparing the profits using the two control structures, Figure 5.12, shows that the self-optimizing control system cannot make use of the improved conditions caused by the unexpected disturbance.

Adapting the self-optimizing control setpoints using RTO or NCO tracking can solve this problem, and at the same time reduce RTO or NCO tracking sample time. In Figure 5.13 the instantaneous profit for direct NCO tracking [François et al., 2005] (sample time: 10 min) and the combined system with a sample time of 25 min is shown. The combined system operates smoother than the pure NCO tracking system while giving similar performance in terms of the profit. However, considering the input usage, Figure 5.14, we find that the combination of self-optimizing control and NCO tracking gives a substantially smoother input action than direct NCO tracking. Using online RTO, the performance could be improved even further because the setpoints would move directly to the optimal values instead of iteratively approaching them. However, unmodelled (unexpected) disturbances are not rejected in online RTO either.

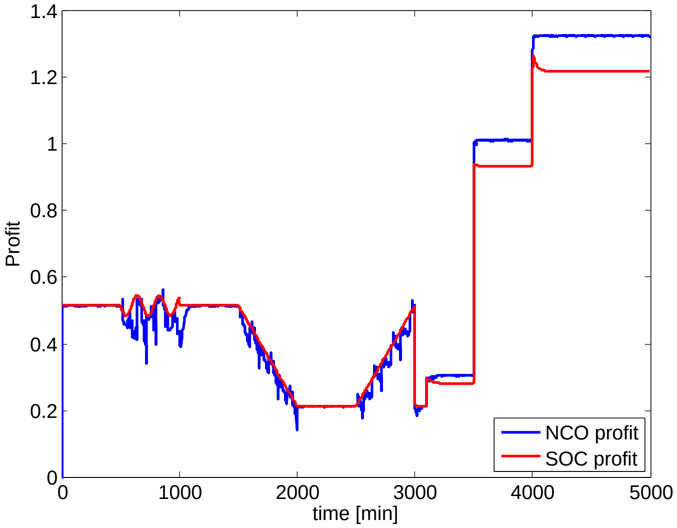


Figure 5.12: Profit, for NCO tracking and for SOC with unexpected disturbance (d_3) at 3100 min

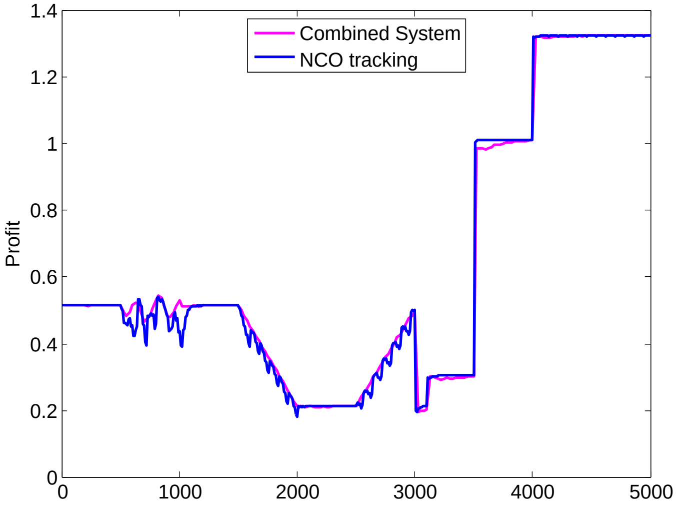


Figure 5.13: Profit for combined SOC/NCO tracking (25 min sample time) and direct NCO tracking (10 min sample time)

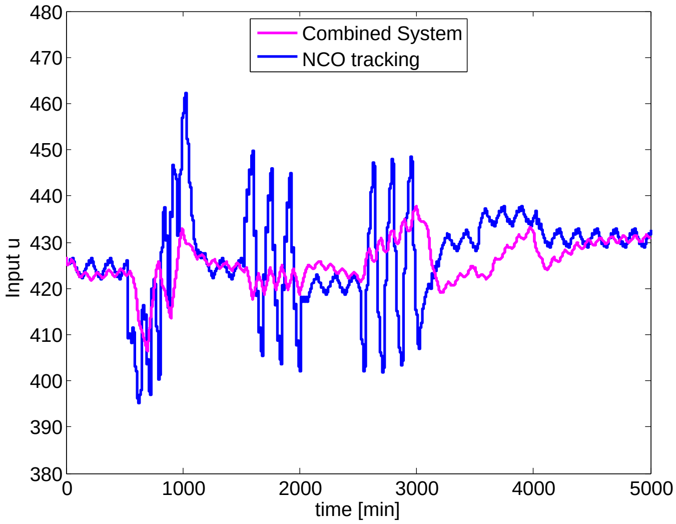


Figure 5.14: Input, combined NCO/SOC and direct NCO tracking

5.6 Discussion

There has been some confusion about the relationship between the “self-optimizing” control approach of Skogestad and coworkers and the NCO tracking approach of Bonvin and coworkers. The reason for the confusion is that both approaches seek to optimize operation and make the gradient zero ($J_{\mathbf{u}} = 0$), but there are significant differences:

In self-optimizing control, offline calculations are used to obtain good controlled variables, typically as linear combination of the measurements, $\mathbf{c} = \mathbf{H}\mathbf{y}_m$, where $\mathbf{c}$ may be considered an approximation of the weighted gradient. It is critical to have a model of the expected disturbances when obtaining $\mathbf{c}$. One does not compute the optimal inputs explicitly; they are generated by a feedback controller to make $\mathbf{c} = \mathbf{H}\mathbf{y}_m = \mathbf{c}_s$ (constant).

In the NCO tracking procedure by François et al. [2005] one aims at obtaining the optimal inputs $\mathbf{u}$ that drive the measured or estimated gradient to zero. It is not necessary to know the disturbances in advance, and no model is needed.

As shown in this paper, the two methods may be successfully combined by controlling the self-optimizing variables $\mathbf{c}$ in the lower layer, and letting NCO tracking adjust the setpoints, $\mathbf{c} = \mathbf{c}_{opt}$ based on online estimates of the gradient.

It was not easy to make the NCO tracking work in spite of the fact that we assumed no measurement noise. This could be partly attributed to the fact that we used a simple finite difference procedure to obtain the gradients, without, for example, steady state detection. The NCO tracking parameters (perturbation magnitude, step size β, sample time) which converge to the optimum, were found by trial and error. Parameters which perform well for one disturbance may give poor performance for a different disturbance. Non-steady state periods make it especially difficult to find parameters which optimize the cost with acceptable input usage.

The NCO tracking procedure hinges on good gradient estimates, and using finite differences for estimating the gradient gives poor NCO tracking updates, even with the assumptions of perfect measurements without noise, and perfect knowledge of the profit value. More advanced gradient estimation techniques and input adaptation methods may give better overall performance, especially in terms of input usage, because poor update steps caused by wrong gradient estimates would be avoided. In this work, we chose to apply the simple finite difference method, because our purpose is to demonstrate that the basic concepts of measurement based optimization techniques, such as NCO tracking, and the model based self-optimizing control concepts are complementary. Whatever technique for calculating the gradient and the NCO tracking updates is used, combining the two methods helps to overcome their limitations. An interesting task for future research might be to study the combination of self-optimizing control with a more advanced update/gradient estimation method and a more realistic case with non-zero mean random measurement noise.

5.7 Conclusion

The different characteristics of the two methods studied in this paper suggest to consider them as complementary, not competing. NCO tracking is most suitable for use in the optimization layer, as an alternative to online RTO, while self-optimizing control is used for selecting CVs in the control layer.

Since almost every RTO system has a dynamic control system in the layer below, using a self-optimizing control structure in the lower layer improves performance and can significantly reduce need for RTO updates. For NCO tracking as implemented in this paper, this means fewer perturbations for gradient estimation. For an online RTO, this means more time for complex, time intensive, computations, with few compromises on performance.

The matlab simulation files are available on the home page of S. Skogestad, `http://www.nt.ntnu.no/users/skoge`, or as supplementary material from the journal.

5.A Relationship between the gradient and the null space method

Consider the unconstrained optimization problem

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) = \min_{\mathbf{u}} [\mathbf{u}^T\ \mathbf{d}^T] \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}. \tag{5.27}$$

Differentiating the cost J with respect to $\mathbf{u}$ gives

$$J_{\mathbf{u}} = \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}. \tag{5.28}$$

The linear model (5.5) can be rewritten as

$$\mathbf{y} = \tilde{\mathbf{G}}^{\mathbf{y}} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}. \tag{5.29}$$

If we assume that we have a sufficient number of measurements, $n_{\mathbf{y}} = n_{\mathbf{u}} + n_{\mathbf{d}}$, then the model may be inverted, and substitution into (5.28) gives

$$J_{\mathbf{u}} = \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{bmatrix} [\tilde{\mathbf{G}}^{\mathbf{y}}]^{-1}\mathbf{y}. \tag{5.30}$$

At the optimum, we have $J_{\mathbf{u}} = 0$, or equivalently $\mathbf{c} = \mathbf{H}\mathbf{y} = 0$, where

$$\mathbf{H} = [\mathbf{J_{uu}}\ \mathbf{J_{ud}}][\tilde{\mathbf{G}}^{\mathbf{y}}]^{-1}. \tag{5.31}$$

This is the same expression for $\mathbf{H}$ as derived in Alstad et al. [2009]. And indeed, if we evaluate $\mathbf{HF}$ using $\mathbf{F}$ in (5.11), we get $\mathbf{HF} = 0$. This follows since $\mathbf{F}$ in (5.11) may be rewritten as

$$\mathbf{F} = \tilde{\mathbf{G}} \begin{bmatrix} -\mathbf{J_{uu}}^{-1}\mathbf{J_{ud}} \\ \mathbf{I} \end{bmatrix}. \tag{5.32}$$

Also note that the loss L and gradient are related by

$$L = \frac{1}{2} J_{\mathbf{u}} \mathbf{J_{uu}}^{-1} J_{\mathbf{u}}, \tag{5.33}$$

so $J_{\mathbf{u}} = 0$ is equivalent to $L = 0$. In summary, we see that the null space method is identical to controlling the gradient, $J_{\mathbf{u}} = 0$.

References

Alstad, V., 2005. Studies on selection of controlled variables. Ph.D. thesis, Norwegian University of Science and Technology, Department of Chemical Engineering.

Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. Industrial & Engineering Chemistry Research 46, 846–853.

Alstad, V., Skogestad, S., Hori, E. S., 2009. Optimal measurement combinations as controlled variables. Journal of Process Control 19 (1), 138–148.

Ariyur, K. B., Krstic, M., 2003. Real-Time Optimization by Extremum-Seeking Control. Wiley-Interscience.

Bazaraa, M. S., Sherali, H. D., Shetty, C. M., 2006. Nonlinear Programming: Theory and Algorithms. John Wiley & Sons.

Bonvin, D., Bodizs, L., Srinivasan, B., 2005. Optimal grade transition for polyethylene reactors via nco tracking. Chemical Engineering Research and Design 83 (6), 692 – 697.

Bonvin, D., Srinivasan, B., Ruppen, D., 2001. Dynamic Optimization in the Batch Chemical Industry. in: Chemical Process Control-VI.

Brdyś, M., Tajewski, P., 1994. An algorithm for steady-state optimising dual control of uncertain plants. In: Proceedings of the first IFAC workshop on new trends in design of control systems Smolenice, Slovakia. pp. 249–254.

Cao, Y., 2005. Direct and indirect gradient control for static optimisation. International Journal of Automation and Computing 2, 60–66.

Draper, C. S., Li, Y. T., 1951. Principles of optimizing control systems and an application to the internal combustion engine. ASME publications 160, 1–16.

Economou, C. G., Morari, M., 1986. Internal model control. 5. extension to nonlinear systems. Industrial & Engineering Chemistry Process Design and Development 25, 403–411.

Engell, S., 2007. Feedback control for optimal process operation. Journal of Process Control 17 (3), 203 – 219.

Findeisen, W., Nailey, F., Brdys, M., Malinowski, K., Tatjewski, P., Wozniak, A., 1980. Control and Coordination in Hierachical Systems. John Wiley & Sons.

François, G., Srinivasan, B., Bonvin, D., 2005. 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.

Gao, W., Engell, S., 2005. Iterative set-point optimization of batch chromatography. Computers & Chemical Engineering 29 (6), 1401 – 1409.

Gros, S., Srinivasan, B., Bonvin, D., 2009. Optimizing control based on output feedback. Computers & Chemical Engineering 33 (1), 191 – 198.

Halvorsen, I. J., Skogestad, S., 1997. Indirect on-line optimization through setpoint control. AIChE 1997 Annual Meeting, Los Angeles; paper 194h.

Halvorsen, I. J., Skogestad, S., 1999. Optimal operation of petlyuk distillation: steady-state behavior. Journal of Process Control 9 (5), 407 – 424.

Halvorsen, I. J., Skogestad, S., Morud, J. C., Alstad, V., 2003. Optimal selection of controlled variables. Industrial & Engineering Chemistry Research 42 (14), 3273–3284.

Kadam, J. V., Marquardt, W., Srinivasan, B., Bonvin, D., 2007. Optimal grade transition in industrial polymerization processes via nco tracking. AIChE Journal 53 (3), 627–639.

Kariwala, V., Cao, Y., Janardhanan, S., 2008. Local self-optimizing control with average loss minimization. Industrial & Engineering Chemistry Research 47, 1150–1158.

Krstic, M., Wang, H.-H., 2000. Stability of extremum seeking feedback for general nonlinear dynamic systems. Automatica 36 (4), 595 – 601.

Marlin, T. E., Hrymak, A., 1997. Real-time operations optimization of continuous processes. In: Proceedings of CPC V, AIChE Symposium Series vol. 93. pp. 156–164.

Morari, M., Stephanopoulos, G., Arkun, Y., 1980. Studies in the synthesis of control structures for chemical processes. Part I: Formulation of the problem. Process decomposition and the classification of the control task. Analsysis of the optimizing control structures. AIChE Journal 26 (2), 220–232.

Roberts, P., 2000. Broyden derivative approximation in isope optmizing and control algorithms. In: Proceedings of the 11th IFAC workshop on control applications of optimization. Vol. 1. pp. 283–288.

Skogestad, S., 2000. Plantwide control: The search for the self-optimizing control structure. Journal of Process Control 10, 487–507.

Skogestad, S., 2004. Control structure design for complete chemical plants. Computers & Chemical Engineering 28.

Srinivasan, B., Biegler, L. T., Bonvin, D., 2008. Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function. Computers & Chemical Engineering 32, 572–279.

Srinivasan, B., Bonvin, D., Visser, E., Palanki, S., 2003a. Dynamic optimization of batch processes: Ii. role of measurements in handling uncertainty. Computers & Chemical Engineering 27 (1), 27 – 44.

Srinivasan, B., Palanki, S., Bonvin, D., 2003b. Dynamic optimization of batch processes: I. characterization of the nominal solution. Computers & Chemical Engineering 27 (1), 1 – 26.

Part II

Polynomial invariants for optimal operation

Chapter 6

Some background on polynomials

> Happiness is ideal, it is the work of the imagination
>
> A. Lord Tennyson

This chapter gives some background on polynomials, algebraic geometry and elimination theory. The purpose of this chapter is to introduce the reader to some important concepts, which are used in later chapters.

6.1 Polynomial methods in control

Polynomials have always been an integral part of control theory and are closely linked to linear algebra (e.g. the characteristic polynomial of a matrix). Because polynomials have many nice properties (e.g. a polynomial multiplied by a polynomial is a polynomial), there exists a strong theoretical basis for working with polynomials. With the availability of computers, more complex symbolic operations became possible, and in the last 5-6 decades many constructive methods for handling and manipulating systems of polynomial equations have been developed. The best-known example maybe is the Gröbner basis [Buchberger, 1965], which can be thought of as a generalization of the Gaussian elimination procedure for polynomial equations.

Some important newer concepts include several improvements on the Gröbner basis calculations and the development of border bases [Kehrein and Kreuzer, 2006]. Border bases are a generalization of Gröbner bases which are numerically more stable and robust to numerical calculations. Another new development is the theory of sparse resultants, which generalize the classical resultants for homogeneous systems of polynomials to inhomogeneous systems [Pedersen and Sturmfels, 1993; Gelfand et al., 1994; Canny, 1990].

With increasing computational power, the algebraic techniques for manipulating polynomial systems have become interesting for other fields of research than pure algebraic geometry. Some examples are robot motion planning [Canny, 1988], automatic theorem proving [Chou, 1988] and cryptography [Ars et al., 2004]. Algebraic methods have also been used in the control community, e.g. for proving stability [Forsman, 1991b,a], or for calculating switching surfaces [Walther et al., 2001; Bistak, 2010]. Other applications have been to solve the parametric optimization problem which arises in nonlinear model predictive control [Fotiou et al., 2006].

A slightly different direction of research has been concerned with positive polynomials, in particular systems of polynomials, which can be written as sum of squares [Henrion and Garulli, 2005]. It has been found many problems involving sum of squares polynomials can be solved using linear matrix inequalities [Parrilo, 2000; Parrilo and Lall, 2003]. Handling sum of squares polynomials in the linear matrix inequality framework is computationally easier than other exact symbolic calculations. Polynomials, which are not sum of squares, may be relaxed, so that they can be written as sum of squares polynomials, [Parrilo, 2000]. This is especially useful for finding bounds on difficult optimization problems. Henrion and Garulli [2005] provide a nice overview of the applications of positive polynomials in a control context.

6.2 Some basic concepts from algebraic geometry

In this section we briefly present some basic details about polynomials and algebraic geometry. It is not meant to be an exhaustive treatment of the topic. For this, we refer to Cox et al. [1992, 2005]. The reader familiar with basic algebraic geometry may skip this chapter and continue directly with the application in Chapter 7.

If not marked otherwise, the presentations and examples in this chapter are taken or adapted from Cox et al. [1992]. Let us define the vector **x** as

$$\mathbf{x} = [x_1, \dots, x_n]^{\mathbf{T}}. \tag{6.1}$$

Definition 1. *A* **monomial** *in $x_1, \dots, x_n$ is a product of the form*

$$x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n}, \tag{6.2}$$

where all exponents $\alpha_1, \dots, \alpha_n$ are non-negative integers.

We use the short hand notation

$$\mathbf{x}^{\alpha} = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n}. \tag{6.3}$$

Let k be a field, for example $k = \mathbb{R}$, or $k = \mathbb{C}$.

Definition 2. *A* **polynomial** *f in $x_1, \dots, x_n$ with coefficient in k is a finite linear combination (with coefficients in k) of monomials. We write*

$$f = \sum_{\alpha} a_{\alpha} \mathbf{x}^{\alpha}, \quad a_{\alpha} \in k, \tag{6.4}$$

where the sum is over a finite number of n-tuples $\alpha = (\alpha_1, \dots, \alpha_n)$.

The set of all polynomials with variables $\mathbf{x}$ and coefficients in k is denoted $k[\mathbf{x}]$. We call $k[\mathbf{x}]$ a polynomial ring, and a_{α} is called the coefficient of the monomial $\mathbf{x}^{\alpha}$. There are many other fields than $k = \mathbb{C}$ or $k = \mathbb{R}$, but we will consider only cases where the coefficients are in $\mathbb{R}$, and solutions can be in $\mathbb{C}$. That is the $k = \mathbb{C}$ covers all our requirements.

Definition 3. *Let k be a field (for example $\mathbb{R}$ or $\mathbb{C}$), and let $f_1, \dots, f_s$ be polynomials in $k[\mathbf{x}] = k[x_1, \dots, x_n]$. Then we set*

$$\mathbf{V}(f_1, \dots, f_s) = \{(a_1, \dots, a_n) \in k^n : f_i(a_1, \dots, a_n) = 0 \quad \forall 1 \leq i \leq s\}. \tag{6.5}$$

The set $\mathbf{V}(f_1, \dots, f_s)$ is called the **affine variety** *defined by $f_1, \dots, f_s$.*

Thus, the affine variety is the set of $\mathbf{x} \in k^n$ for which we have

$$\begin{aligned} f_1(\mathbf{x}) &= 0 \\ f_2(\mathbf{x}) &= 0 \\ &\vdots \\ f_s(\mathbf{x}) &= 0. \end{aligned} \tag{6.6}$$

E.g. the variety of the polynomial

$$f = x^2 + y^2 - 1 \tag{6.7}$$

is the unit circle in $\mathbb{R}^2$, and is an example for a one-dimensional variety (because it is a one-dimensional curve).

Instead of working with the set of polynomials directly, it is often useful to work with the ideal which is generated by the polynomials.

Definition 4. *A subset $I \subset k[\mathbf{x}]$ is an ideal if it satisfies* *i.* $0 \in I$.

ii. *If* $f, g \in I$*, then* $f + g \in I$.

iii. *If* $f \in I$ *and* $h \in k[\mathbf{x}]$*, then* $hf \in I$.

Theorem 7. *Let* $f_1, \ldots, f_s$ *be polynomials in* $k[\mathbf{x}]$*. Then the set*

$$\langle f_1, \ldots, f_s \rangle = \left\{ \sum_{i=1}^{s} h_i f_i : \quad h_1, \ldots, h_s \in k[\mathbf{x}] \right\} \tag{6.8}$$

is an ideal. We call $\langle f_1, \ldots, f_s \rangle$ *the ideal generated by* $f_1, \ldots, f_s$.

Proof. See Cox et al. [1992] □

For example if we have two polynomials $f_1, f_2 \in \mathbb{C}[x_1, x_2]$, the ideal generated by these two polynomial is

$$\langle f_1, f_2 \rangle = \{h_1 f_1 + h_2 f_2\} \tag{6.9}$$

for any $h_1, h_2 \in \mathbb{C}[x_1, x_2]$. Now, if we consider the variety $\mathbf{V}(f_1, f_2)$, that is the set of all x_1, x_2, where we have

$$f_1(x_1, x_2) = f_2(x_1, x_2) = 0, \tag{6.10}$$

then we see that all polynomials in the ideal

$$\langle f_1, f_2 \rangle = \left\{ h_1(x_1, x_2) \underbrace{f_1(x_1, x_2)}_{=0} + h_2(x_1, x_2) \underbrace{f_2(x_1, x_2)}_{=0} \right\} \tag{6.11}$$

will vanish. Therefore, the ideal is also said to consist of all the "polynomial consequences" of the generating polynomials. Even though the ideal contains infinitely many polynomials, it can be shown [Cox et al., 1992], that all ideals can be generated by a finite number of polynomials. This is known as the Hilbert Basis Theorem. The set of generating polynomials of an ideal is not unique, that means that a certain affine variety can be represented by different sets of equations.

There is an analogy with linear algebra. The ideal has similar properties to a subspace, as both are closed under addition and multiplication. In linear algebra, we multiply with scalars, while the ideal is multiplied by polynomials. The analogy continues in that the subspace is spanned by a set of basis vectors, and the ideal is generated by a set of basis polynomials.

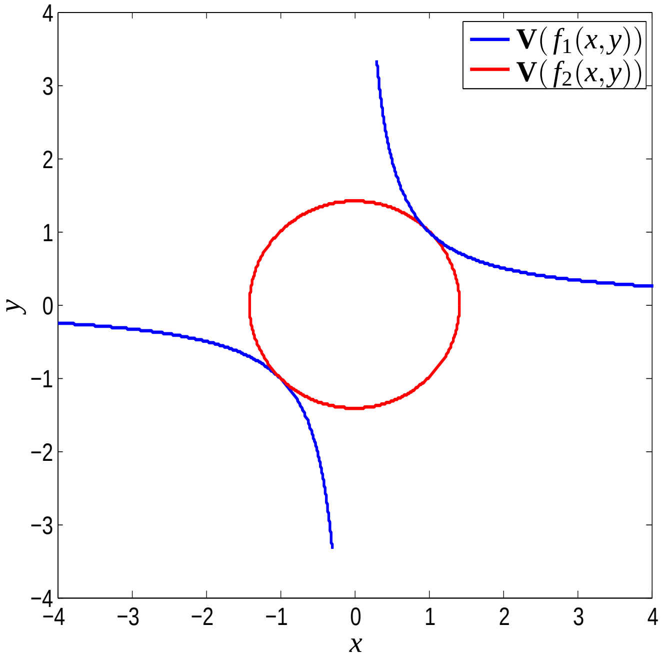


Figure 6.1: The varieties $\mathbf{V}(f_1)$ and $\mathbf{V}(f_2)$. The intersection points yield $\mathbf{V}(f_1, f_1)$

Example 1. *Consider the two polynomials*

$$\begin{aligned} f_1 &= xy - 1 \\ f_2 &= x^2 + y^2 - 2. \end{aligned} \tag{6.12}$$

The variety of the ideal generated by these two polynomials is the intersection of the individual varieties. This is illustrated in Figure 6.1. The variety of these polynomials is the same as the variety given by

$$\begin{aligned} \tilde{f}_1 &= y^4 - 2y^2 + 1 = (y^2 - 1)^2 \\ \tilde{f}_2 &= x + y^3 - 2y. \end{aligned} \tag{6.13}$$

These two sets of polynomials generate the same ideal, and the variety is $\mathbf{V}(f_1, f_2) = \mathbf{V}(\tilde{f}_1, \tilde{f}_2) = \{(1,1), (-1,-1)\}$.

Example 1 demonstrates nicely that the same variety, in this case the two points $(1,1)$ and $(-1,-1)$, can be represented by completely different equations. Note, that we have eliminated the variable x from the first equation. This property will be used in the next chapter, where we want to find a special representation, which does not contain a certain set of variables, while not changing the solution (variety) of the equations.

In linear algebra, the same subspace can be defined by many different basis vectors. Analogue to the linear algebra case, the same ideal (or variety) can be defined by many different basis polynomials.

The theory of algebraic geometry is much richer than we can present here. For more details, we refer the reader to Cox et al. [1992] and Cox et al. [2005]. We will now rather introduce a tool which will be used in the next chapters.

6.3 The sparse resultant

We will only present some basic concepts in this section. For a more detailed exposure, we refer to Cox et al. [2005].

> The resultant of an overconstrained polynomial system characterizes the existence of common roots as a condition on the input coefficients. [...] Since it eliminates the input variables, it is also known as the eliminant [Emiris and Mourrain, 1999].

In the rest of this section, we will restrict all the elements in $\mathbf{x} \in \mathbb{C}^n$ to be nonzero. We define

$$\mathbb{C}^* = \mathbb{C} \setminus 0, \tag{6.14}$$

and we consider only solutions of polynomials which are in the set $(\mathbb{C}^*)^n$. The vanishing of the sparse resultant is the necessary and sufficient condition for common roots $\mathbf{x} \in (\mathbb{C}^*)^{n_d}$ [Cox et al., 2005].

Remark 4. *The set $(\mathbb{C}^*)^n$ is an example of a toric variety [Cox, 2003]. The condition that all elements in* $\mathbf{x}$ *have to be nonzero, makes it possible to allow Laurent polynomials (that is we allow the polynomials to have negative integer exponents) in the ring $\mathbb{C}[\mathbf{x},\mathbf{x}^{-1}]$. In this notation $\mathbf{x}^{-1} = [x_1^{-1},\dots,x_n^{-1}]^{\mathbf{T}}$.*

Using the shorthand notation introduced above, we write a system of polynomials as

$$f_i = \sum_{j=1}^{m_i} a_{i,j}\mathbf{x}^{\alpha_{i,j}} \quad \text{for } i = 1,\dots,n. \tag{6.15}$$

Without loss of generality we assume $a_{i,j} \neq 0$. For a given polynomial $f_i = \sum_{j=1}^{m_i} a_{i,j}\mathbf{x}^{\alpha_{i,j}}$ we collect the exponent vectors to form the set

$$\mathscr{E}_i = \{\alpha_{i,1},\dots,\alpha_{i,m_i}\}. \tag{6.16}$$

This set $\mathscr{E}_i$ is called the *support* of the polynomial f_i. If the supports of all i are the same, the system is called unmixed. Otherwise the system is called a mixed system.

Definition 5. *The Newton polytope of f_i is the convex hull of $\mathscr{E}_i$. We write*

$$Q_i = \operatorname{conv}(\mathscr{E}_i) \subset \mathbb{R}^n. \tag{6.17}$$

Without loss of generality, we assume that the lattice spanned by the supports $\mathscr{E}_i$ is n-dimensional, and it can be identified with $\mathbb{Z}^n$ [Sturmfels, 1994]

Definition 6. *We define the Minkowski sum of to polytopes A and B*

$$A+B = \{\alpha+\beta \quad | \quad \alpha \in A, \beta \in B\}. \tag{6.18}$$

We denote the n-dimensional volume of a polytope Vol(A), and define the mixed volume of polytopes $Q_1 \dots Q_m$ as follows:

Definition 7. *The mixed volume of polytopes $Q_1,\dots,Q_m$, $\mathscr{M}(Q_1,\dots,Q_m)$ can be calculated as*

$$\mathscr{M}(Q_1,\dots,Q_m) = \frac{1}{m!}\sum_{k=1}^{m}(-1)^{m-k} \sum_{1\leq i_1 \leq \cdots \leq i_k \leq, m} \mathscr{M}(Q_{i_1}+\cdots+Q_{i_k}). \tag{6.19}$$

We refer to Cox et al. [2005]; Gelfand et al. [1994] for proofs and details for the calculation of the mixed volume. In the case of 2-dimensional polytopes, there is a simple formula to calculate the mixed volume:

$$\mathscr{M}(Q_1, Q_2) = area(Q_1 + Q_2) - area(Q_1) - area(Q_2). \tag{6.20}$$

Theorem 8 (Pedersen and Sturmfels [1993])**.** *The sparse resultant is well defined, separately homogeneous in its coefficients of each f_i and the degree in these coefficients equals the mixed volume of the other n Newton polytopes.*

The sparse resultant is calculated by formulating a specially designed coefficient matrix **M**, and calculating its determinant. Describing how to construct matrices which can be used to extract the sparse resultant is out of the scope of this work. For this, we refer to Canny [1990]; Gelfand et al. [1994]; Sturmfels [2002]; Emiris and Mourrain [1999] and Cox et al. [2005].

Most available methods construct matrices **M**, of which the determinants do not give the sparse resultant directly, but a polynomial multiple of the resultant. Then the sparse resultant can be calculated as the greatest common divisor of $\det(\mathbf{M})$ and all f_i, with $i = 1, 2, ..., n+1$. Alternatively, the determinant of **M** can be factorized, and the resultant can be found by examining the using the degree values of the coefficients from Theorem 8.

Example 2. *Consider the system of 3 polynomials in two unknowns x_1, x_2,*

$$\begin{aligned} f_1 &= a_{1,1} + a_{1,2}x_1 + a_{1,3}x_2 \\ f_2 &= a_{2,1} + a_{2,2}x_1 + a_{2,3}x_2^2 \\ f_3 &= a_{3,1}x_1 + a_{3,2}d_1^2 + a_{3,3}d_1x_2. \end{aligned} \tag{6.21}$$

This is a mixed system of polynomials with 3 equations in 2 variables. For generic coefficients $a_{i,j}$ the system is overdetermined and does not have a solution (in $(\mathbb{C}^*)^2$). The necessary and sufficient condition for (6.21) *to have a common solution is that the sparse resultant is zero*

$$\mathscr{R}(f_1, f_2, f_3) = 0.$$

The three polynomials have the supports

$$\mathscr{E}_1 = \{(0,0), (1,0), (0,1)\} \tag{6.22}$$

$$\mathscr{E}_2 = \{(0,0), (1,0), (0,2)\} \tag{6.23}$$

$$\mathscr{E}_3 = \{(1,0), (2,0), (1,1)\}. \tag{6.24}$$

The Newton polytopes Q_1, Q_2, Q_3 *are triangles with the volumes (areas)*

$$\begin{aligned} area(Q_1) &= 1/2, && (6.25)\\ area(Q_2) &= 1, && (6.26)\\ area(Q_3) &= 1/2. && (6.27) \end{aligned}$$

The mixed volume is calculated according to (6.20), *so using the volumes of the Minkowski sums*

$$\begin{aligned} area(Q_1+Q_2) &= 7/2, && (6.28)\\ area(Q_1+Q_3) &= 2, && (6.29)\\ area(Q_2+Q_3) &= 7/2, && (6.30) \end{aligned}$$

we obtain

$$\begin{aligned} \mathscr{M}(Q_1,Q_2) &= 2, && (6.31)\\ \mathscr{M}(Q_2,Q_3) &= 2, && (6.32)\\ \mathscr{M}(Q_1,Q_3) &= 1, && (6.33) \end{aligned}$$

for our example.

From Theorem 8 we know that the sparse resultant is of degree 2 in $a_{1,i}$ *and* $a_{3,i}$*, and it is of degree 1 in* $a_{2,i}$*. Using the software* `multires`[1]*[Busé and Mourrain, 2003] to construct the coefficient matrix* **M**, *and calculating its determinant in maple, yields*

$$\begin{aligned} \det(\mathbf{M}) = a_{3,1}(-a_{2,2}a_{3,1}+a_{3,2}a_{2,1})\,\{&a_{2,1}a_{1,3}^2a_{3,2}^2 - 2a_{2,1}a_{1,3}a_{3,3}a_{1,2}a_{3,2}\\ &+a_{2,1}a_{3,3}^2a_{1,2}^2+a_{2,3}a_{1,1}^2a_{3,2}^2+a_{3,2}a_{1,3}a_{1,1}a_{3,3}a_{2,2}-2a_{3,2}a_{2,3}a_{3,1}a_{1,1}a_{1,2}\\ &-a_{3,2}a_{1,3}^2a_{3,1}a_{2,2}-a_{3,3}^2a_{1,1}a_{1,2}a_{2,2}+a_{1,3}a_{3,1}a_{3,3}a_{1,2}a_{2,2}+a_{2,3}a_{3,1}^2a_{1,2}^2\}\,. \end{aligned} \quad (6.34)$$

Examining the degrees of the determinant reveals that the last term in curly brackets is the sparse resultant, so if and only if the coefficients $c_{i,j}$ *cause the last term to vanish, then the system* (6.21) *has a solution* (x_1,x_2) *in* $(\mathbb{C}^*)^2$.

6.4 Some practical considerations for working with polynomials

Even though the next chapters deal with “ideal” systems, that is systems without measurement noise or model mismatch, we would like mention that we must be careful when these assumptions are not satisfied. We present some practical problems which may occur when dealing with multivariate polynomials.

[1] Available on: http://www-sop.inria.fr/galaad/software/multires/

6.4.1 Ill conditioning

Polynomial equations can be very ill conditioned. By ill-conditioned, we mean that a small change in one of the coefficients can have a large effect on the number and the quality of the solutions. We demonstrate this by considering the famous “Wilkinson polynomial” Wilkinson [1984] Consider for example the polynomial

$$f = (x-1)(x-2)\dots(x-20). \tag{6.35}$$

This polynomial has 20 real roots, the natural numbers from 1 to 20. However, perturbing the coefficient of x^{19} (which is 210) by $2^{-23} \approx 10^{-7}$ results in a large change of the zeros. The perturbed polynomial has 5 complex zeros, and a real zero at ≈ 20.847., and the roots at 18 and 19 merge into a double root at ≈ 18.62.

It generally cannot be seen directly from the polynomials, how sensitive the roots are with respect to perturbations in the coefficients. This has to be kept in mind when working with polynomials.

To see another issue that can arise when working with polynomials consider

$$f(x,y) = 25x^3 - 20x + 1 - y. \tag{6.36}$$

A part of the variety $\mathbf{V}(f(x,y))$ of this polynomial is shown in Figure 6.2. When x is in the interval $[-1,1]$, the absolute value of y is lower than 10. However, when $x = 2$ we have $f(x=2) = 161$, and $f(x=3) = 616$. At values of $x \approx 10$, the y takes values about $2.5 \cdot 10^4$. This shows in a simple example that relatively small changes in one variable (x) can cause other variables to vary over several orders of magnitude.

6.4.2 Size of polynomials

A partially related issue which arises with symbolic computation with polynomials is that the expressions can become very large, and can include very high exponents (which in turn makes the problem more difficult to handle numerically). However, as we have seen in Example 1, the same solution set can be defined by completely different equations, and some representations will be smaller and better behaving than others.

6.4.3 Desired properties for Polynomials

From a practical perspective, we would like the polynomials to satisfy the following requirements.

1. The representation should be “simple”, in the sense that the polynomials have

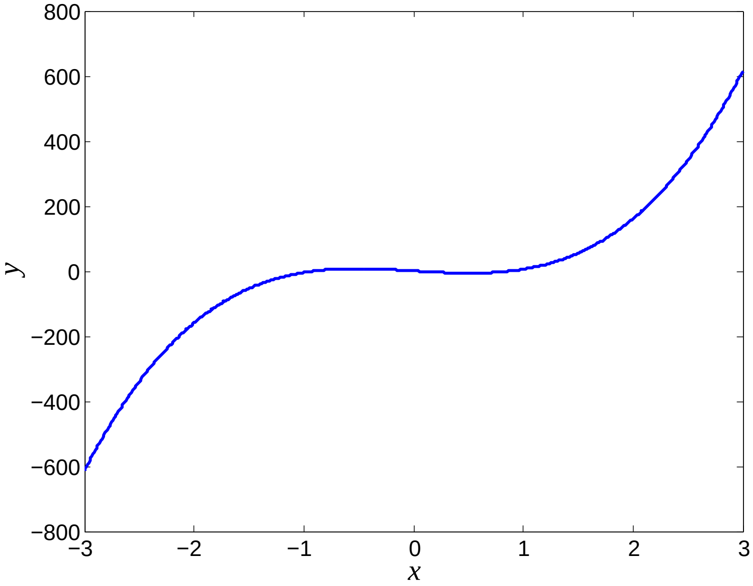


Figure 6.2: Variety of the polynomial $f(x,y) = 25x^3 - 20x + 1 - y$

(a) few terms and

(b) low degree in the variables.

2. The polynomial expression should be robust with respect to perturbations in the coefficients.

3. The all variables should have approximately the same variability in the region of interest.

These requirements are partly related to each other. Polynomials of high degree will generally behave more sensitive to changes in the variables than polynomials of low degree. Therefore it is advisable to find and use representations of low degree.

However, it lies in the nature of polynomials that they can be very difficult to handle in a numerical context. Some of the issues may be resolved by scaling the variables properly, or finding a polynomial representation which is robust to numerical errors. However, in general, one must be aware of these potential pitfalls when working with polynomials in practical applications.

References

Ars, G., Faugère, J.-C., Imai, H., Kawazoe, M., Sugita, M., 2004. Comparison between XL and Gröbner basis algorithms. In: Lee, P. (Ed.), Advances in Cryptology - ASIACRYPT 2004. Vol. 3329 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 157–167.

Bistak, P., 2010. Time optimal control using gröbner bases. In: In International Conference on Cybernetics and Informatics, Slovak Republic.

Buchberger, B., 1965. An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. Ph.D. thesis, University of Innsbruck.

Busé, L., Mourrain, B., 2003. Using the maple multires package.
URL http://www-sop.inria.fr/galaad/software/multires

Canny, J. F., 1988. The complexity of robot motion planning. Ph.D. thesis, Massachusetts Institute of Technology.

Canny, J. F., 1990. Generalised characteristic polynomials. Journal Symbolic Computation 9, 241–250.

Chou, S.-C., 1988. Mechanical geometry theorem proving. D. Reidel Publishing Company.

Cox, D., 2003. What is a toric variety. In: Goldman, R., Krasauskas, R. (Eds.), Topics in Algebraic Geometry and Geometric Modeling. American Mathematical Society, pp. 203–223.

Cox, D., Little, J., O'Shea, D., 1992. Ideals, Varieties, and Algorithms. Springer-Verlag.

Cox, D., Little, J., O'Shea, D., 2005. Using Algebraic Geometry. Springer.

Emiris, I. Z., Mourrain, B., 1999. Matrices in elimination theory,. Journal of Symbolic Computation 28 (1-2), 3–43.

Forsman, K., 1991a. Applications of gröbner bases to nonlinear systems. In: First European Control Conference. pp. 164–169.

Forsman, K., 1991b. Construction of lyapunov functions using grÃűbner bases. In: In Proceedings of the 30th Conference on Decision and Control. IEEE, pp. 798–799.

Fotiou, I. A., Rostalski, P., Parrilo, P. A., Morari, M., 2006. Parametric optimization and optimal control using algebraic geometry methods. International Journal of Control 79 (11), 1340 – 1358.

Gelfand, I. M., Kapranov, M. M., Zelevinsky, A. V., 1994. Discriminants, Resultants and Multidimensional Determinants. Birkhäuser, Boston, MA.

Henrion, D., Garulli, A. (Eds.), 2005. Positive Polynomials in Control. Springer Verlag.

Kehrein, A., Kreuzer, M., 2006. Computing border bases. Journal of Pure and Applied Algebra 205 (2), 279 – 295.

Parrilo, P. A., 2000. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. Ph.D. thesis, California Institute of Technology.

Parrilo, P. A., Lall, S., 2003. Semidefinite programming relaxations and algebraic optimization in control.

Pedersen, P., Sturmfels, B., 1993. Product formulas for chow forms. Mathematische Zeitschrift 214, 377–396.

Sturmfels, B., 1994. On the newton polytope of the resultant. Journal of Algebraic Combinatorics 3, 207–236.

Sturmfels, B., 2002. Solving systems of polynomial equations.

Walther, U., Georgiou, T. T., Tannenbaum, A., 2001. On the computation of switching surfaces in optimal control: A grobner basis approach. IEEE Transactions on Automatic Control 46 (4), 534 –540.

Wilkinson, J. H., 1984. The perfidious polynomial. In: G.H.Golub, Ed. Studies in mathematics. Mathematical Association of America. Vol. 24. pp. 1–28.

Chapter 7

Measurement polynomials as controlled variables

The purpose of computing is
insight, not numbers.

R.W. Hamming

We present a method for finding controlled variables, which are polynomial combinations of measurements. Controlling these measurement combinations gives optimal operation. Our work extends the concept of self-optimizing control to processes described by polynomial equations. Using the first-order necessary optimality conditions, invariant variable combinations are determined for the unconstrained degrees of freedom. Any unknown internal variables (states) and disturbances are eliminated to obtain new invariant variable combinations containing only known variables (measurements). If the disturbance causes the active constraints to change, the invariants may be used to identify, and switch to the right region. This makes the method applicable over a wide disturbance range with changing active sets. The procedure is applied to two case studies of continuous stirred tank reactors.

Submitted to Journal of Process Control

7.1 Introduction

For continuous processes, which are operated in steady state most of the time, an established method to achieve optimal operation in spite of varying disturbances is real-time optimization (RTO) [Marlin and Hrymak, 1997]. The real-time optimizer generally uses a nonlinear steady state model, which is updated at intervals based on measurements. This updated model is used to on-line recompute optimal setpoints for the controlled variables in the control layer below. This concept has gained acceptance in industry and is increasingly used for improving plant performance. However, installing an RTO system and maintaining it generally entails large costs.

A second approach to optimizing plant performance is to use a process model off-line to find a self-optimizing control structure. The basic concept of self-optimizing control was conceived by Morari et al. [1980], who write that we "want to find a function c of the process variables which when held constant leads automatically to the optimal adjustments of the manipulated variables", but they did not provide any method for identifying this function. The idea is to use this function as a controlled variable and keep it at a constant setpoint by simple control structures, e.g. PID controllers, or by more complex model predictive controllers (MPC). Using this kind of controlled variables disburdens the real-time optimizer, or may even make it unnecessary [Jäschke and Skogestad, 2010].

The term "self-optimizing control" was defined in the context of controlled variable selection with the purpose of describing the practical goal of finding "smart" controlled variables $\mathbf{c}$. Skogestad [2000] writes:

> Self-optimizing control is achieved if a constant setpoint policy results in an acceptable loss L (without the need to re-optimize when disturbances occur).

Many industrial processes are operated using self-optimizing control, although it is not always called that. For example, optimally active constraints may be viewed as self-optimizing variables, e.g., maximum cooling of an air stream before entering a compressor. However, the more difficult problem is to identify self-optimizing control variables associated with unconstrained degrees of freedom. In most cases, engineering insight and experience leads to the choice of self-optimizing controlled variables, and the optimization problem is not formulated explicitly. An example for the unconstrained case is controlling the air/fuel ratio entering a combustion engine at a constant value.

It has been noted previously [Halvorsen and Skogestad, 1997; Cao, 2003; Halvorsen et al., 2003; François et al., 2005; Chachuat et al., 2009], that the gradient of the cost function with respect to the degrees of freedom $\mathbf{u}$ would be the ideal controlled variable, $\mathbf{c} = J_{\mathbf{u}}$. However, the gradient $J_{\mathbf{u}}$ is usually not directly measurable, and analytical expressions for the gradient generally contain variables which are unmeasured (unknown disturbances). Therefore, the methods in self-optimizing control theory can be thought of as an approximation (in some "best" way) of the gradient using a measurement model.

Use of self-optimizing controlled variables enables us to separate the two problems of optimizing the system and designing the controller. Thus, in a first step the controlled variables are determined based on the steady state first order optimality conditions, and in a second step a suitable controller is designed. In most cases, a simple PI controller will be sufficient, but also more advanced controllers can be used to control the self-optimizing variable. The advantage of this separation is that it makes it possible to focus completely on steady state optimal behavior when designing the control structure, while all issues which arise when handling dynamic systems are considered when designing the actual controllers.

In the last decade, several contributions have been made on the systematic search of controlled variables which have self-optimizing properties [Halvorsen et al., 2003; Alstad and Skogestad, 2007; Kariwala et al., 2008; Alstad et al., 2009; Heldt, 2009], but to the author's knowledge, self-optimizing control has only been considered for cases with linear measurement models and a quadratic cost function. This results in linear measurement combinations $\mathbf{c} = \mathbf{H}\mathbf{y}$ as controlled variables. In cases where a strong curvature is present at the optimum, the loss imposed by using linear measurement combinations may not be acceptable, and the controlled variables are not self-optimizing.

The main contribution of this work is to extend the ideas of self-optimizing control, in particular the concept of the null-space method [Alstad and Skogestad, 2007], to constrained systems described by multivariable polynomials. This results in controlled variables which are polynomials in the measurements, $\mathbf{c} = \mathbf{c}(\mathbf{y})$.

We further show that, under some assumptions, the controlled variables can be used to determine when the steady state set of active constraints changes, and which set it changes to. Also for changing active constraints, the separation of the steady state optimization and the control problem applies. This means that all dynamic problems which come with changing control structures can be considered separately.

7.2 Overall procedure

The proposed procedure for achieving optimal operation is summarized in Figure 7.1. In steps 1 and 2 we formulate the optimization problem and determine regions of constant active constraints, also called critical regions. This is done by offline calculations, for example, by gridding the disturbance space with a sufficiently fine

1. Formulate optimization problem
2. For the expected set of disturbances, find all regions with different sets of active constraints $\mathscr{A}_i$
3. For each region of active constraints $\mathscr{A}_i$

   a Formulate optimality conditions

   b Eliminate Lagrangian multipliers λ from optimality conditions to obtain invariants $J_{\mathbf{z},red}$ (reduced gradient)

   c Obtain measurement invariants $\mathbf{c}(\mathbf{y})$ by eliminating unknowns, such that

$$\mathbf{c}(\mathbf{y}) = 0 \Longleftrightarrow J_{\mathbf{z},red} = 0$$

4. In each region $\mathscr{A}_i$, control the

   a Active constraints

   b Invariants $\mathbf{c}(\mathbf{y})$

Use controlled variables and measured constraints for changing regions

Figure 7.1: Procedure for finding nonlinear invariants as controlled variables

grid and optimizing the process for each grid point.

In step 3, for each critical region, (a) the optimality conditions are formulated, and (b) the Lagrangian multipliers are eliminated. Then (c) the unknown variables, i.e. the disturbances and the internal state variables are eliminated from the optimality conditions to obtain an invariant variable combination $\mathbf{c}(\mathbf{y})$ which contains only measured variables and known parameters. Optimal operation is achieved in each critical region by controlling the active constraints and the invariant measurement combinations.

Finally, in step 4 we monitor the active constraints and the invariants of the neighboring regions to determine when to switch to a new region.

7.3 Optimal operation using the optimality conditions

7.3.1 Problem formulation

Optimal operation is defined as minimizing a scalar cost index $J(\mathbf{u},\mathbf{x},\mathbf{d})$ subject to satisfying the model equations, $g=0$, and operational constraints, $h \leq 0$:

$$\min_{\mathbf{u},\mathbf{x}} J(\mathbf{u},\mathbf{x},\mathbf{d}) \quad \text{s.t} \quad \begin{cases} g(\mathbf{u},\mathbf{x},\mathbf{d}) = 0 \\ h(\mathbf{u},\mathbf{x},\mathbf{d}) \leq 0. \end{cases} \tag{7.1}$$

Here $\mathbf{u}$, $\mathbf{x}$, $\mathbf{d}$ denote the manipulated input variables, the internal state variables, and the unmeasured disturbance variables, respectively. We assume that, in addition, we have measurements $\mathbf{y} = \mathbf{y}(\mathbf{u},\mathbf{x},\mathbf{d})$, which provide information about internal states, inputs, and disturbances. To handle the measurements in a consistent way when dealing with polynomials, we will write the measurement relations implicitly, as

$$m(\mathbf{u},\mathbf{x},\mathbf{d},\mathbf{y}) = 0, \tag{7.2}$$

To simplify notation, we combine state and input variables in a vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{u} \\ \mathbf{x} \end{bmatrix}. \tag{7.3}$$

Problem (7.1) is the same problem as the one solved on-line at given sample times when using RTO. In this work, however, we do not wish to solve the optimization problem on-line; instead, we analyze the problem using offline calculations, in order to find good controlled variables which yield optimal operation when controlled at their setpoints.

Optimality conditions

Let $\mathbf{z}^*$ be a feasible point of the optimization problem (7.1), and assume that all gradient vectors $\nabla_{\mathbf{z}} g_i(\mathbf{z}^*,\mathbf{d})$ and $\nabla_{\mathbf{z}} h_i(\mathbf{z}^*,\mathbf{d})$ associated with $g_i(\mathbf{z}^*,\mathbf{d}) = 0$ and the active constraints, $h_i(\mathbf{z}^*,\mathbf{d}) = 0$, are linearly independent. If $\mathbf{z}^*$ is locally optimal, then there exist Lagrangian multiplier vectors λ and ν, such that the following conditions, known as the KKT conditions, are satisfied [Nocedal and Wright, 2006; Bazaraa et al., 2006][1]:

[1] We follow the notation of Bazaraa et al. [2006], where the gradient of a function $f : \mathbb{R}^{n_\mathbf{z}} \to \mathbb{R}$ is defined as $\nabla_z f = \left[\frac{\partial f}{\partial z_1}, \frac{\partial f}{\partial z_2}, \dots, \frac{\partial f}{\partial z_{n_z}}\right]^\mathbf{T}$. Given a differentiable vector function $g : \mathbb{R}^{n_\mathbf{z}} \to \mathbb{R}^{n_g}$, where $g(\mathbf{z}) = [g_1(\mathbf{z}), g_2(\mathbf{z}), \dots, g_{n_g}(\mathbf{z})]^\mathbf{T}$, the Jacobian in gradient notation $\nabla_\mathbf{z} g(\mathbf{z})$ is given by the $n_z \times n_g$ matrix $\nabla_\mathbf{z} g = \begin{bmatrix} [\nabla_z g_1]^\mathbf{T} \\ \vdots \\ [\nabla_z g_{n_g}]^\mathbf{T} \end{bmatrix}_{n_g \times n_\mathbf{z}}$.

$$\begin{aligned} \nabla_{\mathbf{z}} J(\mathbf{z}^*, \mathbf{d}) + [\nabla_{\mathbf{z}} g(\mathbf{z}^*, \mathbf{d})]^{\mathbf{T}} \lambda + [\nabla_{\mathbf{z}} h(\mathbf{z}^*, \mathbf{d})]^{\mathbf{T}} \nu &= 0 \\ g(\mathbf{z}^*, \mathbf{d}) &= 0 \\ h(\mathbf{z}^*, \mathbf{d}) &\leq 0 \\ [h(\mathbf{z}^*, \mathbf{d})]^{\mathbf{T}} \nu &= 0 \\ \nu &\geq 0. \end{aligned} \tag{7.4}$$

The condition that the Jacobian of the active constraints has independent rows (or full rank) is called the linear independence constraint qualification (LICQ) and guarantees that the Lagrangian multipliers λ and ν are uniquely defined at the optimum $\mathbf{z}^*$.

When optimizing nonlinear systems, such as polynomial systems, there are several complications which may arise. The optimality conditions (7.4) will in general not have a unique solution. There may be multiple maxima, minima and saddle points, so finding the global minimum is not an easy task in itself. When a solution to (7.4) is found, it has to be checked whether it indeed is the desired solution (minimum). In addition, there may be solutions which are not physical (complex). So before controlling $\mathbf{c}(\mathbf{y})$ to zero, it has to be assured that the process actually is at the desired optimum.

These and other issues from nonlinear and polynomial optimization are not addressed in this work. The focus of this paper is rather to present a method which gives a controlled variable $\mathbf{c}(\mathbf{y})$ which is zero at all points that satisfy the KKT conditions, and which is nonzero whenever the KKT conditions are not satisfied.

7.3.2 Partitioning into sets of active constraints

Generally, the set of inequality constraints $h_i(\mathbf{z}, \mathbf{d}) \leq 0$ that are active varies with the value of the elements in $\mathbf{d}$. The disturbance space can hence be partitioned into regions which are characterized by their individual set of active constraints. These regions will be called critical regions.

The concept of critical regions allows one to decompose the original optimization problem (7.1) into a sequence of equality constrained optimization problems, which are valid in the corresponding critical region. This idea is also applied in multi-parametric programming [Pistikopoulos et al., 2007]. However, we do not search for an explicit expression for the inputs $\mathbf{u}^*$, as in multi-parametric programming. We rather use each subproblem to find good controlled variables $\mathbf{c}$ for the corresponding critical region.

In order to obtain a fully specified system in each region,

1. the active constraints need to be controlled, and

2. a controlled variable has to be controlled for each unconstrained degree of freedom.

The number of unconstrained degrees of freedom, $n_{\mathbf{c}} = n_{DOF}$ is calculated according to

$$n_{DOF} = n_{\mathbf{z}} - n_g - n_{h,active}, \tag{7.5}$$

where $n_{\mathbf{z}}$, n_g, $n_{h,active}$ denote the number of variables $\mathbf{z}$, the number of model equations, g, and the number of constraints from h which are active ($h_i = 0$). This method will give the right amount of degrees of freedom for practically all systems encountered in reality.

Remark 5. *When the optimization problem* (7.1) *is composed of polynomial equations, the critical regions are defined by semialgebraic sets in* $\mathbb{R}^{n_u+n_x+n_d}$*. A semialgebraic set is defined as the finite union of sets defined by a finite number of polynomial equalities and inequalities,*

$$g(\mathbf{x},\mathbf{u},\mathbf{d}) = 0 \tag{7.6}$$

and

$$h(\mathbf{x},\mathbf{u},\mathbf{d}) \leq 0, \tag{7.7}$$

Where g and h are polynomials in the variables $\mathbf{x},\mathbf{u}$ *and* $\mathbf{d}$*, with coefficients in* $\mathbb{R}$*. Loosely speaking, a semialgebraic set can be thought of a set defined by a finite number of polynomial inequalities. The interior of an ellipsoid, or the set of points on a curve in the* $\mathbb{R}^n$ *are examples of semialgebraic sets.*

In the rest of the paper, by abuse of notation, all active constraints $h_i(\mathbf{z},\mathbf{d}) = 0$ are included in the equality constraint vector $g(\mathbf{z},\mathbf{d}) = 0$. Then in every critical region, the optimization problem (7.1) can be written as

$$\begin{gathered}\min_{\mathbf{z}} J(\mathbf{z},\mathbf{d}) \\ \text{s.t.} \\ g(\mathbf{z},\mathbf{d}) = 0.\end{gathered} \tag{7.8}$$

The KKT first-order optimality conditions (7.4) simplify for problem (7.8) in each critical region, to

$$\begin{aligned}\nabla_{\mathbf{z}} J(\mathbf{z},\mathbf{d}) + [\nabla_{\mathbf{z}} g(\mathbf{z},\mathbf{d})]^{\mathbf{T}} \lambda &= 0, \\ g(\mathbf{z},\mathbf{d}) &= 0.\end{aligned} \tag{7.9}$$

These expressions cannot be used for control, because they still contain unknown variables, $\mathbf{x}$ (in $\mathbf{z} = [\mathbf{u},\mathbf{x}]$), $\mathbf{d}$, and λ, which must be eliminated.

7.3.3 Eliminating the Lagrangian multipliers λ

In every critical region, a control structure that gives optimal operation has to satisfy (7.9). Recall that the LICQ is assumed to hold, i.e. at the optimum, $\nabla_{\mathbf{z}} g(\mathbf{z},\mathbf{d})$ has full row rank for every value of $\mathbf{d}$ within the critical region.

Proposition 1. *Let* $\mathbf{N}(\mathbf{z},\mathbf{d}) \in \mathbb{R}^{n_z \times (n_z - n_g)}$ *be a basis for the null space of* $\nabla_{\mathbf{z}} g(\mathbf{z},\mathbf{d})$. *Keeping the active constraints* $g(\mathbf{z},\mathbf{d}) = 0$, *and the variable combination* $J_{\mathbf{z},red} = [\mathbf{N}(\mathbf{z},\mathbf{d})]^{\mathbf{T}} \nabla_{\mathbf{z}} J(\mathbf{z},\mathbf{d}) = 0$ *then results in optimal steady state operation.*

Proof. Select $\mathbf{N}(\mathbf{z},\mathbf{d})$ such that $[\mathbf{N}(\mathbf{z},\mathbf{d})]^{\mathbf{T}}[\nabla_{\mathbf{z}} g(\mathbf{z},\mathbf{d})]^{\mathbf{T}} = 0$. Since the LICQ are satisfied, the constraint Jacobian $\nabla_{\mathbf{z}} g(\mathbf{z},\mathbf{d})$ has full row rank and $\mathbf{N}(\mathbf{z},\mathbf{d})$ is well defined and does not change dimension within the region. The first equation in (7.9) is premultiplied by $[\mathbf{N}(\mathbf{z},\mathbf{d})]^{\mathbf{T}}$ to get

$$\begin{aligned} [\mathbf{N}(\mathbf{z},\mathbf{d})]^{\mathbf{T}} \left(\nabla_{\mathbf{z}} J(\mathbf{z},\mathbf{d}) + [\nabla_{\mathbf{z}} g(\mathbf{z},\mathbf{d})]^{\mathbf{T}} \lambda\right) &= [\mathbf{N}(\mathbf{z},\mathbf{d})]^{\mathbf{T}} \nabla_{\mathbf{z}} J(\mathbf{z}.\mathbf{d}) + \underline{0}\lambda \\ &= [\mathbf{N}(\mathbf{z},\mathbf{d})]^{\mathbf{T}} \nabla_{\mathbf{z}} J(\mathbf{z},\mathbf{d}). \end{aligned} \tag{7.10}$$

Since $\mathbf{N}(\mathbf{z},\mathbf{d})$ has full rank, we have that (7.9) are satisfied whenever $g(\mathbf{z},\mathbf{d}) = 0$ and $[\mathbf{N}(\mathbf{z},\mathbf{d})]^{\mathbf{T}} \nabla_{\mathbf{z}} J(\mathbf{z},\mathbf{d}) = 0$. □

We call $J_{\mathbf{z},red} = [\mathbf{N}(\mathbf{z},\mathbf{d})]^{\mathbf{T}} \nabla_{\mathbf{z}} J(\mathbf{z},\mathbf{d})$ the reduced gradient. By construction, the reduced gradient has $n_{DOF} = n_z - n_g$ elements. Keeping

$$J_{\mathbf{z},red} = [\mathbf{N}(\mathbf{z},\mathbf{d})]^{\mathbf{T}} \nabla_{\mathbf{z}} J(\mathbf{z},\mathbf{d}) = 0 \tag{7.11}$$

together with the active constraints, $g(\mathbf{z},\mathbf{d}) = 0$, fully specifies the system at the optimum and is equivalent to controlling the first-order optimality conditions (7.9). However, $J_{\mathbf{z},red}$ cannot generally be used for control directly because it still depends on the variables $\mathbf{d}$ and $\mathbf{x}$ ($\mathbf{x}$ enters through $\mathbf{z} = [\mathbf{u}\,,\mathbf{x}]^{\mathbf{T}}$), which are usually unknown. Thus, we want to eliminate the unknown disturbances $\mathbf{d}$ and the internal states $\mathbf{x}$ from the expression (7.11).

The simplest approach is to solve the measurement equations $m(\mathbf{x},\mathbf{u},\mathbf{d},\mathbf{y}) = 0$ and the active constraints $g(\mathbf{z},\mathbf{d}) = 0$ for the unknowns $\mathbf{d}$ and $\mathbf{x}$, and substitute the solution into $J_{\mathbf{z},red}$. As we show, this is straightforward in case of linear equations, but it becomes significantly more complicated when working with polynomials of higher degree.

7.4 Elimination for linear quadratic systems

In this section we describe the basic concept of how the unknowns are eliminated form $J_{\mathbf{z},red}$. This will lead to the linear zero loss method, or null-space method

Alstad and Skogestad [2007]). Our procedure is demonstrated step by step for minimizing a quadratic cost function subject to linear constraints and a linear measurement model. Solving the model and measurement equations for the unknowns and substituting into $J_{\mathbf{z},red}$ is avoided, as this is difficult to extend to the polynomial case. Instead, we search for necessary and sufficient conditions which guarantee that the measurement model $m(\mathbf{x},\mathbf{u},\mathbf{d},\mathbf{y}) = 0$, the active constraints and the model $g(\mathbf{z},\mathbf{d}) = 0$, and the reduced gradient $J_{\mathbf{z},red} = 0$ are satisfied at the same time. We require that the necessary and sufficient condition is a function of measurements $\mathbf{y}$ and known parameters, only.

The optimization problem we consider is

$$\begin{gathered}\min_{\mathbf{z}} J(\mathbf{z},\mathbf{d}) = \begin{bmatrix} \mathbf{z}^{\mathbf{T}} & \mathbf{d}^{\mathbf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{J_{zz}} & \mathbf{J_{zd}} \\ \mathbf{J_{zd}}^{\mathbf{T}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{d} \end{bmatrix} \\ \text{s.t.} \\ \mathbf{Az} - \mathbf{b} = 0, \end{gathered} \tag{7.12}$$

and the linear measurement model is

$$\begin{aligned} m(\mathbf{z},\mathbf{d},\mathbf{y}) &= \mathbf{y} - [\mathbf{G}^{\mathbf{y}}\,\mathbf{G}^{\mathbf{y}}_{\mathbf{d}}] \begin{bmatrix} \mathbf{z} \\ \mathbf{d} \end{bmatrix} = 0 \\ &= \mathbf{y} - \tilde{\mathbf{G}}^{\mathbf{y}} \begin{bmatrix} \mathbf{z} \\ \mathbf{d} \end{bmatrix} = 0. \end{aligned} \tag{7.13}$$

We consider $[\mathbf{z}\,,\mathbf{d}]^{\mathbf{T}}$ as unknown and we assume that (7.12) has a solution, $\mathbf{J_{zz}} > 0$, and $\mathbf{A}$ has full rank. In addition, we assume that the measurements are linearly independent, and $\tilde{\mathbf{G}}^{\mathbf{y}} = [\mathbf{G}^{\mathbf{y}}\ \mathbf{G}^{\mathbf{y}}_{\mathbf{d}}]$ invertible.

The null space of the constraint gradient, $\mathbf{N}$, is a constant matrix which is independent of $\mathbf{z}$, such that $\mathbf{AN} = 0$. The first-order necessary optimality conditions require that at the optimum

$$J_{\mathbf{z},red} = \mathbf{N}^{\mathbf{T}}\nabla_{\mathbf{z}} J(\mathbf{z},\mathbf{d}) = \mathbf{N}^{\mathbf{T}} \begin{bmatrix} \mathbf{J_{zz}} & \mathbf{J_{zd}} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{d} \end{bmatrix} = 0. \tag{7.14}$$

If the number of independent measurements (n_y) is equal to the number of unknown variables ($n_z + n_d$), the measurement relations (7.13) can be solved for the unknowns and substituted into the gradient expression (7.14) to obtain

$$\mathbf{c}(\mathbf{y}) = J_{\mathbf{z},red} = \mathbf{N}^{\mathbf{T}} \begin{bmatrix} \mathbf{J_{zz}} & \mathbf{J_{zd}} \end{bmatrix} \left[\tilde{\mathbf{G}}^{\mathbf{y}}\right]^{-1} \mathbf{y}. \tag{7.15}$$

Controlling $\mathbf{c}(\mathbf{y}) =$ and the active constraints $\mathbf{Az} - \mathbf{b}$ to zero, then results in optimal operation.

However, in the case of polynomial equations of higher degrees, it is generally not possible to solve for the unknown variables. Therefore, we consider the problem from a slightly different perspective. Suppose $n_y = n_z + n_d$, then for any disturbance $\mathbf{d}$ and feasible $\mathbf{z}$, the output $\mathbf{y}$ is uniquely defined. Since the mapping from $[\mathbf{z}\,\mathbf{d}]$ to $\mathbf{y}$ is invertible, one can also say that for every output and disturbance pair $(\mathbf{y}, \mathbf{d})$ there exist a unique $\mathbf{z}$, which satisfies the measurement equations (7.13). However, an arbitrary feasible $\mathbf{z}$ with its corresponding pair ($\mathbf{y}$,$\mathbf{d}$) will fail to satisfy the first-order optimality condition (7.14). Moreover, since there is no requirement on the rank of $\mathbf{J_{zd}}$, the condition $\mathbf{J_{zz}} > 0$ guarantees that only one $\mathbf{z}$ (or equivalently one pair $(\mathbf{y}, \mathbf{d})$) satisfies the first-order optimality conditions and that this pair $(\mathbf{y}, \mathbf{d})$ corresponds to the unique minimizer $\mathbf{z}$. This can be used for a more subtle elimination method which does not require solving the measurement relation (7.13) for the unknown variables.

Consider the elements of the reduced gradient vector (7.14), one at a time, together with all the measurement equations (7.13). Let the superscript (i) denote the i-th row of a matrix or a vector. We write the reduced gradient (7.14) together with the measurement equations (7.13) as a sequence of square linear systems

$$\underbrace{\begin{bmatrix} [\mathbf{N^T J_{zz}}]^{(i)} & [\mathbf{N^T J_{zd}}]^{(i)} & 0 \\ \mathbf{G^y} & \mathbf{G^y_d} & \mathbf{y} \end{bmatrix}}_{\mathbf{M}^{(i)}} \begin{bmatrix} \mathbf{z} \\ \mathbf{d} \\ -1 \end{bmatrix} = 0. \tag{7.16}$$

Here $\mathbf{M}^{(i)}$, $i = 1 \ldots n_{DOF}$ are square matrices of size $(n_y + 1)$. We want to find a particular output combination which satisfies (7.16). A solution for $[\mathbf{z}, \mathbf{d}]^\mathbf{T}$ exists only if $\text{rank}(\mathbf{M^{(i)}}) = n_y = n_z + n_d$.

The submatrix $[\mathbf{G^y}\ \mathbf{G^y_d}\ \mathbf{y}]$ already has rank n_y, irrespective of the value of $\mathbf{y}$ (or the control policy that generates the input $\mathbf{u}$ which in turn generates $\mathbf{y}$). This follows because $[\mathbf{G^y}\ \mathbf{G^y_d}\ \mathbf{y}]$ has more columns than rows, and because $\text{rank}([\mathbf{G^y}\,\mathbf{G^y_d}]) = n_y$. Therefore, the condition for a common solution is:

$$\det(\mathbf{M}^{(i)}) = 0 \quad \text{for all } i = 1..n_{DOF}. \tag{7.17}$$

This condition guarantees that a common solution to (7.16) exists, so the elements of the controlled variable $\mathbf{c}$ are selected as $c_i = \det(\mathbf{M}^{(i)})$.

It remains to show that controlling the determinants $c_i = \det(\mathbf{M}^{(i)})$ gives the inputs which lead to the optimum. Since the system is linear and the rank of the measurement equations is n_y, there is a unique linear invertible mapping between the measurements $\mathbf{y}$ and the vector $[\mathbf{z}, \mathbf{d}]^\mathbf{T}$. Therefore every value of $\mathbf{y}$ corresponds uniquely to some value in $\mathbf{z}$.

In the case with more measurements, $n_y > n_z + n_d$, any subset of $n_z + n_d$ measurements may be chosen such that rank($[\mathbf{G^y}\,\mathbf{G^y_d}]$)=$n_z + n_d$.

Remark 6. *When there are no constraints, we have that* $\mathbf{z} = \mathbf{u}$*, and this method results in the null space method [Alstad and Skogestad, 2007]. In this case,* $\mathbf{N}$ *may be set to any nonsingular matrix, for example the identity matrix* $\mathbf{N} = \mathbf{I}$*. Then we have that*

$$\mathbf{c}_{\text{Nullspace}} = [\mathbf{J_{uu}}\ \mathbf{J_{ud}}][\tilde{\mathbf{G}^{\mathrm{y}}}]^{-1}\mathbf{y}, \tag{7.18}$$

as has been derived in Alstad et al. [2009].

The null space method was originally derived by Alstad and Skogestad [2007] using the optimal sensitivity matrix $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$*. However, controlling* $\mathbf{c} = \mathbf{Hy}$ *with* $\mathbf{H}$ *selected such that* $\mathbf{HF} = 0$*, is indeed the same as controlling the gradient to zero.*

Remark 7. *Actually, in the linear case presented above, the constraints*

$$\mathbf{Az} - \mathbf{b} = 0 \tag{7.19}$$

can always be used to eliminate the variables $\mathbf{x}$*. Then, for the remaining unconstrained problem the condition is* $\mathrm{rank}([\mathbf{G}^{\mathrm{y}}\quad \mathbf{G}^{\mathbf{y}}_{\mathbf{d}}]) = n_{\mathbf{u}} + n_{\mathbf{d}}$*, and we need* $n_{\mathbf{y}} = n_{\mathbf{u}} + n_{\mathbf{d}}$ *measurements*

The next example is included to demonstrate that our “determinant method” gives the same result as the previously published null-space method Alstad and Skogestad [2007].

Example 3 (Linear model and quadratic objective)**.** *Consider a system from Alstad [2005]. The cost to minimize is*

$$J = (u-d)^2, \tag{7.20}$$

and the measurement relations (model equations) are

$$\begin{aligned} y_1 &= G_1^y u + G_{d,1}^y d \\ y_2 &= G_2^y u + G_{d,2}^y d \end{aligned} \tag{7.21}$$

where the variables u, d, y *denote the input, the disturbance and the measurements, respectively. The values of the gains are given in Table 7.1. We are searching for a condition on* y_1 *and* y_2 *such that the optimality condition is satisfied. The gradient is* $\nabla_u J = 2(u-d)$ *and* $J_{uu} = 2$*,* $J_{ud} = -2$*. It is easily verified that measurements are linearly independent. This gives an equation system of 3 equations in 2 unknowns:*

$$\mathbf{M}\begin{bmatrix} u \\ d \\ -1 \end{bmatrix} = 0, \tag{7.22}$$

Table 7.1: Gain values for Example 3

Variable	Value
G_1^y	0.9
$G_{d,1}^y$	0.1
G_2^y	0.5
$G_{d,2}^y$	-1.0

where

$$\mathbf{M} = \begin{bmatrix} J_{uu} & J_{ud} & 0 \\ G_1^y & G_{d,1}^y & y_1 \\ G_2^y & G_{d,2}^y & y_2 \end{bmatrix}. \tag{7.23}$$

Equation (7.22) *has a solution* $\begin{bmatrix} u \\ d \\ -1 \end{bmatrix}$ *if and only if*

$$\det(\mathbf{M}) = 0. \tag{7.24}$$

Therefore the necessary and sufficient condition for the existence of a nontrivial solution is

$$\det\left(\begin{bmatrix} J_{uu} & J_{ud} & 0 \\ G_1^y & G_{d,1}^y & y_1 \\ G_2^y & G_{d,2}^y & y_2 \end{bmatrix}\right) = -y_1(J_{uu}G_{d,2}^y - G_2^y J_{ud}) + y_2(J_{uu}G_{d,1}^y - G_1^y J_{ud}) \\ = 0. \tag{7.25}$$

On inserting the parameter values from Table 7.1, we obtain

$$c = \det(\mathbf{M}) = y_1 + 2y_2. \tag{7.26}$$

Controlling $c = y_1 + 2y_2$ *to zero therefore yields optimal operation. This is the same variable combination as found by applying the null-space method in Alstad [2005].*

Even though obtaining the invariants via the determinant may seem cumbersome, it eliminates the necessity of inverting the measurements and solving for the unknowns. While this is of little advantage for systems of linear equations, the concept can be extended to systems of polynomial equations which cannot easily be solved for the right set of unknowns.

7.5 Elimination for systems of polynomial equations

Let $\hat{\mathbf{d}}$ now denote the vector of all unmeasured (unknown) variables,

$$\hat{\mathbf{d}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{d} \end{bmatrix}, \tag{7.27}$$

not only including disturbances $\mathbf{d}$, but also unknown states $\mathbf{x}$, and let $\mathbf{y}$ include all measurements and known inputs. Thus, every variable belongs either to $\hat{\mathbf{d}}$ or $\mathbf{y}$, and we write the optimality conditions as

$$\begin{aligned} J_{\mathbf{z},red}(\mathbf{y},\hat{\mathbf{d}}) &= 0 \\ g(\mathbf{y},\hat{\mathbf{d}}) &= 0, \end{aligned} \tag{7.28}$$

and the measurement relations as

$$m(\mathbf{y},\hat{\mathbf{d}}) = 0. \tag{7.29}$$

Remark 8. *Note that in the elimination step, we do not distinguish between internal states variables* $\mathbf{x}$ *and external disturbances* $\mathbf{d}$*. All variables which are not available as measurements (that is,* $\hat{\mathbf{d}} = [\mathbf{x},\mathbf{d}]^{\mathbf{T}}$*) have to be eliminated from the optimality conditions.*

For polynomial equations, eliminating the unknown variables from $J_{\mathbf{z},red}$ is not as straightforward as in the linear case, as we cannot just solve the measurement equations for the unknowns and insert them in to the expression of $J_{\mathbf{z},red}$. Even for the case of a univariable polynomial of degree 5 and higher, for example $d^5 - d + 1 = 0$, there exist no general analytic solution formulas, as was proven by Abel [1826]. Therefore we are interested in finding a way to eliminate the unknown variables $\hat{\mathbf{d}}$ from $J_{\mathbf{z},red}(\mathbf{y},\hat{\mathbf{d}}) = 0$ without solving g and m for them first. This is exactly what was done in Section 7.4, where we used the determinant of a carefully constructed coefficient matrix, which characterizes the existence of a common solution in $\mathbf{d}$, to replace $J_{\mathbf{z},red}$. The determinant is a function of the known variables only, that is, the measurements $\mathbf{y}$ and the parameters $\tilde{\mathbf{G}}^{\mathbf{y}}$, $\mathbf{J_{zz}}$ and $\mathbf{J_{zd}}$.

The generalization of the determinant to systems of polynomial equations is called resultant. According to Emiris and Mourrain [1999],

> “the resultant of an overconstrained polynomial system characterizes the existence of common roots as a condition on the input coefficients”.

To be more specific, we consider multivariate polynomials $f \in \mathbb{R}[\mathbf{y},\hat{\mathbf{d}}]$, that is, real polynomial functions with coefficients in $\mathbb{R}$, and variables $\mathbf{y} = [y_1, y_2, \ldots, y_{n_y}]$ and $\hat{\mathbf{d}} = [\mathbf{x},\mathbf{d}] = [\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_{n_{\hat{d}}}]$. Given an $n_{\hat{d}}$-tuple,

$$\alpha_{i,j} = \left(\alpha_{i,j}(1), \alpha_{i,j}(2), \ldots, \alpha_{i,j}(n_{\hat{d}})\right), \tag{7.30}$$

we use the shorthand notation

$$\hat{\mathbf{d}}^{\alpha_{i,j}} = \hat{d}_1^{\alpha_{i,j}(1)} \hat{d}_2^{\alpha_{i,j}(2)} \ldots \hat{d}_{n_{\hat{d}}}^{\alpha_{i,j}(n_{\hat{d}})}. \tag{7.31}$$

Then we can write a system of n polynomials in compact form

$$f_i(\mathbf{y}, \hat{\mathbf{d}}) = \sum_{j=0}^{k_i} a_{i,j}(\mathbf{y}) \hat{\mathbf{d}}^{\alpha_{i,j}}, \quad i = 1..n, \tag{7.32}$$

where the coefficients $a_{ij}(\mathbf{y}) \neq 0$ are polynomials in $\mathbb{R}[\mathbf{y}]$, that is, polynomials in $\mathbf{y}$ with coefficients in $\mathbb{R}$.

We consider the functions $a_{i,j}(\mathbf{y})$ as polynomial coefficients, and $\hat{\mathbf{d}}$ as variables. For every polynomial f_i, we collect the exponent vectors in the set $\mathscr{E}_i = \{\alpha_{i,1}, \ldots, \alpha_{i,k_i}\}$. This set is called support of the polynomial f_i.

The support of the polynomial $f = d_1^2 + d_1 d_2 - 1$, for example, is found to be $\mathscr{E} = \{(2,0),(1,1),(0,0)\}$. We denote as Q_i the convex hull of the support of a polynomial, $Q_i = \mathrm{conv}(\mathscr{E}_i)$.

Further, we denote the set of complex numbers without zero as $\mathbb{C}^*$ ($\mathbb{C}^* = \mathbb{C} \setminus 0$). Next we present some basic concepts from algebraic geometry taken from Cox et al. [2005].

Definition 8 (Affine variety). *Consider $f_1, \ldots, f_n$ polynomials in $\mathbb{C}[\hat{d}_1, \ldots \hat{d}_{n_{\hat{d}}}]$. The affine variety defined by $f_1, \ldots, f_n$ is the set*

$$V(f_1, \cdots, f_n) = \left\{ (\hat{d}_1, \ldots, \hat{d}_{n_{\hat{d}}}) \in \mathbb{C}^{n_{\hat{d}}} : f_i(\hat{d}_1, \ldots \hat{d}_{n_{\hat{d}}}) = 0 \quad i = 1 \ldots n \right\} \tag{7.33}$$

Casually speaking, the variety is the set of all solutions in $\mathbb{C}^{n_{\hat{d}}}$.

Definition 9 (Zariski closure). *Given a subset $S \subset \mathbb{C}^m$, there is a smallest affine variety $\overline{S} \subset \mathbb{C}^m$ containing S. We call $\overline{S}$ the Zariski closure of S.*

Let $L(\mathscr{E}_i)$ be the set of all polynomials whose terms all have exponents in the support $\mathscr{E}_i$:

$$L(\mathscr{E}_i) = \left\{ a_{i,1} \hat{\mathbf{d}}^{\alpha_{i1}} + \cdots + a_{i,k_i} \hat{\mathbf{d}}^{\alpha_{ik_i}} : a_{i,j} \in \mathbb{C} \right\} \tag{7.34}$$

Then the coefficients $a_{i,j}$ of n polynomials define a point in $\mathbb{C}^{n \times k_i}$. Now let

$$Z(\mathscr{E}_1, \ldots \mathscr{E}_n) \subset L(\mathscr{E}_1) \times \cdots \times L(\mathscr{E}_n) \tag{7.35}$$

be the Zariski closure of the set of all $(f_1, \ldots f_n)$ for which (7.32) has a solution in $(\mathbb{C}^*)^{n_{\hat{d}}}$ (that is the Zariski closure of the points defined by all coefficients $a_{i,j} \in \mathbb{C}$ for which (7.32) has a solution). For an overdetermined system of $n_{\hat{d}} + 1$ polynomials in $n_{\hat{d}}$ variables we have following result:

Theorem 9 (Sparse resultant [Gelfand et al., 1994; Cox et al., 2005])**.** *Assume that $Q_i = \operatorname{conv}(\mathscr{E}_\mathrm{i})$ is an $n_{\hat{d}}$-dimensional polytope for $i = 1,\ldots,n_{\hat{d}}+1$. Then there is an irreducible polynomial $\mathscr{R}$ in the coefficients of the f_i such that*

$$(f_1,\ldots,f_{n_{\hat{d}}+1}) \in Z(\mathscr{E}_1,\ldots,\mathscr{E}_{n_{\hat{d}+1}}) \Longleftrightarrow \mathscr{R}(f_1,\ldots,f_{n_{\hat{d}}+1}) = 0. \tag{7.36}$$

In particular, if

$$f_1(d_1\ldots d_{n_{\hat{d}}}) = \cdots = f_{n_{\hat{d}}+1}(d_1\ldots d_{n_{\hat{d}}}) = 0 \tag{7.37}$$

has a solution $(\hat{d}_1,\ldots,\hat{d}_{n_{\hat{d}}})$ in $(\mathbb{C}^*)^{n_{\hat{d}}}$, then

$$\mathscr{R}(f_1,\ldots,f_{n_{\hat{d}}+1}) = 0. \tag{7.38}$$

Remark 9. *The requirement that Q_i has to be $n_{\hat{d}}$-dimensional is no restriction and can be relaxed, [Sturmfels, 1994]. However, for simplicity, we chose to present this result here.*

Depending on the allowed space for the roots, there are other resultant types (e.g. Bezout resultants and Dixon resultants for system of homogeneous polynomials), with different algorithms to generate them. Generally they will be conditions for roots in the projective space with homogeneous (or homogenized) polynomials. For more details on different resultants, we refer to Gelfand et al. [1994]; Sturmfels [1994]; Cox et al. [2005]. An overview of different matrix constructions in elimination theory is given in Emiris and Mourrain [1999].

We choose to use the sparse resultant, since most polynomial systems encountered in practice are sparse in the supports. That means, for example, a polynomial of degree 5 in two variables x,y will not contain all 21 possible combinations of monomials $x^5,y^5,x^4y,xy^4,\ldots,x^4,y^4,x^3y,\ldots,y,x,1$. Just as in linear algebra, this sparseness can be exploited for calculating the resultant. Another reason for using the sparse resultant is that it gives the necessary and sufficient conditions for toric roots, that is, roots in $(\mathbb{C}^*)^{n_{\hat{d}}}$, such that the input polynomials need not be homogeneous (or homogenized), as in other resultants.

Finally, the sparse resultant enables us to work with Laurent polynomials, that is, polynomials with positive and negative integer exponents.

Usually, resultant algorithms set up a matrix in the coefficients of the system. The determinant of this matrix is then the resultant or a multiple of it. Generating the coefficient matrices and their determinants efficiently is a subject to ongoing research, but there are some useful algorithms freely available. In this work, we use the maple software package `multires` [Busé and Mourrain, 2003], which can be downloaded from the internet[2].

[2] `http://www-sop.inria.fr/galaad/logiciels/multires`

For more details on the theory of sparse resultants, we refer to Gelfand et al. [1994]; Emiris and Mourrain [1999]; Sturmfels [2002]; Dickenstein and Emiris [2005].

7.5.1 Finding invariant controlled variables for polynomial systems

After introducing the concepts above, we are ready to apply them in the context of controlled variable selection and self-optimizing control. As in the linear case above, we assume that the active constraints and the model equations, $g(\mathbf{y},\hat{\mathbf{d}})=0$, and the measurement relations, $m(\mathbf{y},\hat{\mathbf{d}})=0$, are satisfied.

Let $J^{(i)}_{\mathbf{z},red}$ denote the i-th element in the reduced gradient expression. To obtain the $n_{\mathbf{c}}=n_{DOF}$ controlled variables needed for the unconstrained degrees of freedom we have:

Theorem 10 (Nonlinear measurement combinations as controlled variables)**.** *Given* $\hat{\mathbf{d}}\in(\mathbb{R}^*)^{n_{\hat{d}}}$*, and* $n_y+n_g=n_{\hat{d}}$*, independent relations* $g(\mathbf{y},\hat{\mathbf{d}})=m(\mathbf{y},\hat{\mathbf{d}})=0$ *such that the system*

$$\begin{aligned} g(\mathbf{y},\hat{\mathbf{d}})&=0\\ m(\mathbf{y},\hat{\mathbf{d}})&=0 \end{aligned} \tag{7.39}$$

has finitely many solutions for $\hat{\mathbf{d}}\in(\mathbb{C}^*)^{n_{\hat{d}}}$*. Let* $\mathscr{R}(J^{(i)}_{\mathbf{z},red},g,m),\quad i=1\dots n_{\mathbf{c}}$ *be the sparse resultants of the* $n_{\mathbf{c}}$ *polynomial systems composed of*

$$J^{(i)}_{\mathbf{z},red}(\mathbf{y},\hat{\mathbf{d}})=0,\quad g(\mathbf{y},\hat{\mathbf{d}})=0,\quad m(\mathbf{y},\hat{\mathbf{d}})=0\quad i=1\cdots n_{\mathbf{c}}, \tag{7.40}$$

then controlling the active constraints, $g(\mathbf{y},\hat{\mathbf{d}})=0$*, and* $c_i=\mathscr{R}(J^{(i)}_{\mathbf{z},red},g,m)\quad i=1,\dots,n_{\mathbf{c}}$*, yields optimal operation throughout the region.*

Proof. The active constraints are controlled, thus $g(\mathbf{y},\hat{\mathbf{d}})=0$ and $m(\mathbf{y},\hat{\mathbf{d}})=0$ are satisfied always, and there is no condition on the parameters for this part of the system.

The system $g(\mathbf{y},\hat{\mathbf{d}})=0, m(\mathbf{y},\hat{\mathbf{d}})=0$ has only finitely many solutions for $\hat{\mathbf{d}}$, so the set of possible $\hat{\mathbf{d}}$ is fixed. Moreover, we know that a real solution to the subsystem $g(\mathbf{y},\mathbf{d})=m(\mathbf{y},\mathbf{d})=0$ exists, since it is the given disturbance.

From Theorem 9, the sparse resultant gives the necessary and sufficient conditions for the existence of a solution for (7.40) in $\mathbf{d}\in(\mathbb{C}^*)^{n_d}$. Therefore, whenever $J^{(i)}_{z,red}=0$, the resultant is zero (necessary condition). On the other hand if $\mathscr{R}(J_{z,red},g,m)=0$ then the system (7.40) is satisfied (sufficient condition).

This holds for any solution $\hat{\mathbf{d}}\in(\mathbb{C}^*)^{n_{\hat{d}}}$, and in particular the "actual" values of $\hat{\mathbf{d}}$. Because there are as many resultants as unconstrained degrees of freedom, controlling $\mathscr{R}(J^{(i)}_{z,red},g,m)$ for $i=1,\dots,n_u$ satisfies the necessary conditions of optimality in the region. □

Remark 10. *In cases where the* $\hat{\mathbf{d}} \notin (\mathbb{C}^*)^{n_{\hat{d}}}$*, we may apply a variable transformation to formulate the problem such we get* $\hat{\mathbf{d}} \in (\mathbb{C}^*)^{n_{\hat{d}}}$*. For example a translation* $d = \tilde{d} - 1$.

Remark 11. *By partitioning the overall optimization problem into several regions of active constraints, we assume that we have obtained well behaving systems for each region. In particular it is assumed that there are no base points (values of* $a_{i,j}(\mathbf{y})$*, where a polynomial in g or m vanishes for all values of* $\hat{\mathbf{d}}$*).*

Remark 12. *In some cases, the matrix of coefficients may be singular, yielding an identically zero determinant. These cases can be handled by a perturbation of the system at that point. This is a standard method of handling degeneracies in resultants Canny [1990]; Rojas [1999].*

Example 4 (Elimination)**.** *Consider a system with one disturbance d, where we want to minimize a cost J subject to one constraint. The reduced gradient is* $J_{z,red} = \mathbf{N^T}\nabla_{\mathbf{z}} J(y,d) = a_{1,1}(y) + a_{1,2}(y)d$*, and the constraint is*

$$g(y,d) = a_{2,1}(y) + a_{2,2}(y)d + a_{2,3}(y)d^2 = 0. \tag{7.41}$$

All coefficients $a_{i,j}(y)$ *are known functions of the measurements. At the optimum we must have*

$$J_{z,red} = a_{1,1}(y) + a_{1,2}(y)d = 0. \tag{7.42}$$

For arbitrary coefficients $a_{1,1}, a_{1,2}, a_{2,1}, a_{2,2}, a_{2,3}$*, this system of univariate polynomials in d does not have a common solution. However if the sparse resultant is zero, then there exist a common solution* $d \neq 0$ *for* (7.41)-(7.42)*. In the case of univariate polynomials, the sparse resultant coincides with the classical resultant, which is the determinant of the Sylvester matrix [Cox et al., 1992], and the vanishing of the resultant is the necessary and sufficient condition for the existence of a common root. We construct the Sylvester matrix*

$$Syl = \begin{bmatrix} a_{1,2}(y) & a_{1,1}(y) & 0 \\ 0 & a_{1,2}(y) & a_{1,1}(y) \\ a_{2,3}(y) & a_{2,2}(y) & a_{2,1}(y) \end{bmatrix}, \tag{7.43}$$

and the resultant is (where we omit writing the dependence on y explicitly)

$$\mathscr{R}\left(J_{z,red}, g(y,d)\right) = \det(Syl) = a_{1,2}^2 a_{2,1} - a_{1,2}a_{1,1}a_{2,2} + a_{2,3}a_{1,1}^2. \tag{7.44}$$

For a common root d^* *to exist, the polynomial in the coefficients* $\mathscr{R}(J_{\mathbf{z},red}, g(y,d))$ *must vanish. Since the constraints are satisfied,* $g(y,d) = 0$ *for any disturbance* $d \in \mathbb{R}$*, controlling the resultant to zero is the condition for the reduced gradient* $J_{\mathbf{z},red}$ *to become zero. So for any real* $d \neq 0$*, the optimality conditions will be satisfied, and operation will be optimal whenever* $\mathscr{R}(J_{\mathbf{z},red}, g(y,d)) = 0$.

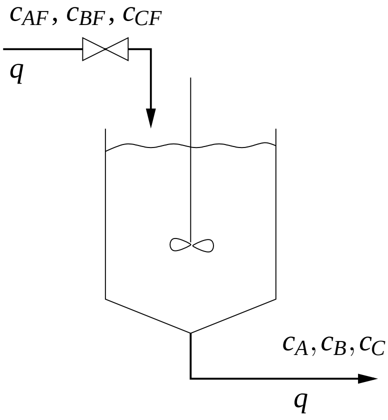


Figure 7.2: Isothermal CSTR

7.6 Case study I

Consider a CSTR as in Figure 7.2, with a feed stream containing mainly component A, and possibly also components B and C, and with two first order chemical reactions,

$$\begin{aligned} A &\longrightarrow B \quad r_1 = k_1 c_A \\ B &\longrightarrow C \quad r_2 = k_2 c_B. \end{aligned} \tag{7.45}$$

Component B is the desired product, while C is an undesired side product. We have one manipulated variable, the feed stream $u = q$, which can be adjusted to achieve profitable performance. We assume that product B can be sold very profitably, and can be separated easily from the other components. The operational objective is therefore to maximize the concentration of the desired product. Since the level in the reactor is controlled at a constant value, this corresponds to maximizing the total production of B.

It is assumed that the unmeasured disturbances are the rate constants k_1 and k_2. This could be due to imperfect temperature control in the reactor, catalyst decay or unknown reaction paths, which have been approximated by first order kinetics. In addition we assume that the concentration c_B is too difficult (expensive) to measure online.

Thus, we will have to eliminate two unknown disturbance variables k_1, k_2, and one unmeasured state variable c_B. This gives

$$\hat{\mathbf{d}} = \begin{bmatrix} c_B \\ k_1 \\ k_2 \end{bmatrix} \tag{7.46}$$

Table 7.2: Unmeasured variables $\hat{\mathbf{d}}$

Symbol	Description
k_1	Reaction constant for reaction 1
k_2	Reaction constant for reaction 2
c_B	Concentration of desired product

Table 7.3: Known variables: Inputs, measurements and parameters

Symbol	Description	Type	Value	Unit
q	Feed flow rate	Known input u	varying	m^3/min
c_A	Outlet concentration A	Measurement	”	kmol/m^3
c_C	Outlet concentration C	”	”	kmol/m^3
V	Tank volume	Known parameter	given	m^3
c_{AF}	Feed concentration A	”	”	kmol/m^3
c_{BF}	Feed concentration B	”	”	kmol/m^3
c_{CF}	Feed concentration C	”	”	kmol/m^3

The unmeasured variables are summarized in Table 7.2.

All measurements and known parameters are shown in Table 7.3. The task is to find a controlled variable which can be controlled using the total flow rate, and which maximizes the desired concentration.

Step 1: Formulate the optimization problem We collect the input $u = q$, and the states $\mathbf{x} = [c_A, c_B, c_C]$ into a vector

$$\mathbf{z} = [q, c_A, c_B, c_C]^{\mathbf{T}}. \tag{7.47}$$

Then the optimization problem is

$$\begin{aligned} \min_{\mathbf{z}} J &= -c_B \\ s.t.& \\ g(\mathbf{z}) &= 0, \end{aligned} \tag{7.48}$$

where the constraints $g(\mathbf{z}) = 0$ are the model equations which are derived from the mass balances:

$$\begin{aligned} g_1 &= qc_{AF} - qc_A - k_1 c_A V = 0 \\ g_2 &= qc_{BF} - qc_B + k_1 c_A V - k_2 c_B V = 0 \\ g_3 &= qc_{CF} - qc_C + k_2 c_B V = 0. \end{aligned} \tag{7.49}$$

Step 2: Find regions of active constraints In our example, there are no other constraints as the model equations. Therefore we have only one region of active constraints, which is defined by (7.49). Since we have four variables and three constraints, the number of degrees of freedom is

$$n_{DOF} = n_{\mathbf{z}} - n_g = 4 - 3 = 1, \tag{7.50}$$

and thus the number of controlled variables is $n_c = n_{DOF} = 1$.

Step 3: Formulate optimality conditions Using $\mathbf{z} = [q, c_A, c_B, c_C]^{\mathrm{T}}$, the first order optimality conditions are

$$\begin{aligned} \nabla_{\mathbf{z}} J(\mathbf{z}) + [\nabla_{\mathbf{z}} g(\mathbf{z})]^{\mathbf{T}} \lambda = 0, \\ g(\mathbf{z}) = 0. \end{aligned} \tag{7.51}$$

Step 3a: Eliminate Lagrangian multipliers We calculate the null-space of the constraint Jacobian $\mathbf{N} = [n_1, n_2, n_3, n_4]^{\mathbf{T}}$ with

$$n_1 = -q(q + k_2 V)(q + k_1 V) \tag{7.52}$$

$$n_2 = -(q + k_2 V) q (c_{AF} - c_A) \tag{7.53}$$

$$n_3 = q(-k_1 V c_{AF} + k_1 c_A V - qc_{BF} - c_{BF} k_1 V + qc_B + c_B k_1 V) \tag{7.54}$$

$$\begin{aligned} n_4 = {} & k_1[(-c_{BF} + c_B - c_{AF} + c_A - c_{CF} + c_C) V^2 k_2 + V(-qc_{CF} + qc_C)] \\ & + (qc_B - qc_{CF} + qc_C - qc_{BF}) V k_2 + q^2 (c_C - c_{CF}). \end{aligned} \tag{7.55}$$

Step 3b: Reduced gradient The reduced gradient for our system is defined as $J_{\mathbf{z},red} = [\mathbf{N}(\mathbf{z})]^{\mathbf{T}} \nabla_{\mathbf{z}} J(\mathbf{z}) = 0$. Using $\nabla_{\mathbf{z}} J(\mathbf{z}) = [0,\, 0,\, -1,\, 0]^{\mathbf{T}}$ we have that

$$\begin{aligned} J_{\mathbf{z},red} &= -n_3 \\ &= -q(-k_1 V c_{AF} + k_1 c_A V - qc_{BF} - c_{BF} k_1 V + qc_B + c_B k_1 V). \end{aligned} \tag{7.56}$$

Step 3c: Eliminating unknowns k_1, k_2 and c_B We have three model equations $g_1 = g_2 = g_3 = 0$, (7.49), and three unknowns $\hat{\mathbf{d}} = [c_B, k_1, k_2]^{\mathbf{T}}$. Before we can apply Theorem 10, we have to check the assumptions first:

1. Under normal operation (nonzero feed, etc.), when all other variables are given, $g_1 = g_2 = g_3 = 0$ has one solution for k_1, k_2, c_B (finite number of solutions).
2. Under normal operation we have that $k_1 \neq 0$, $k_2 \neq 0$ and $c_B \neq 0$. Therefore we have that $\mathbf{d} \in (\mathbb{C}^*)^3$.

Since all requirements are fulfilled, we can use the resultant $\mathscr{R}(J_{\mathbf{z},red}, g_1, g_2, g_3)$ as controlled variable. We use software `multires` Busé and Mourrain [2003] to calculate the sparse resultant and obtain for the controlled variable

$$c = \mathscr{R}(J_{\mathbf{z},red}, g_1, g_2, g_3) = c_{AF}c_A + c_{AF}c_{CF} - c_{AF}c_C - c_A^2. \tag{7.57}$$

Here, we chose to eliminate k_1, k_2 and c_B. However, depending on the choice of eliminated variables, there exist many other expressions for the invariant c.

Step 4: Control the invariant Controlling the invariant such that

$$c = 0 \tag{7.58}$$

yields optimal operation.

Remark 13. *Note that the self-optimizing invariant* (7.57) *has become simpler than the expression for the reduced gradient* (7.56). *Depending on the structure of the equations, the resultant may become surprisingly simple, as in this case. In other cases, it may become more complicated. In general it is difficult to make statements about the form of the invariant a-priori.*

7.7 Changes in the active constraints.

In this section, we present a pragmatic approach for detecting when to change the control structure, because of changes in the active set. This task is a research field in itself and has received some attention in literature. Baotić et al. [2008] e. g. has worked on linear systems with quadratic objectives, and Woodward et al. [2010] present an extremum seeking method, which can handle changing active constraints.

An exhaustive study on this topic is outside the scope of this paper. However, we would like to present a procedure, which may be used as starting point for a more thorough investigation of this problem in future work.

From a pure optimization perspective, there is no difference between a constraint and a controlled variable $\mathbf{c}(\mathbf{y})$, as the controlled variable may be simply seen as an active constraint, and, similarly, an active constraint may be considered a variable which is controlled at its constant setpoint. From this perspective, there is no difference between an active constraint and the model equations, either.

However, from an implementation point of view, there are differences between the model, the active constraints, and the controlled variables $\mathbf{c}(\mathbf{y})$. First of all, the active constraints and the controlled variables $\mathbf{c}(\mathbf{y}) = 0$ are not satisfied automatically, that is one must control them to their setpoints. Secondly, since their values are known (or calculated using known measurements) they may be used for detecting when to switch control structures. The basic idea is to monitor the controlled variables and the active constraints of all neighboring regions.

The main assumptions are that all the regions are adjacent, the disturbance moves the system continuously from one region to another, and the system cannot jump over regions. In addition, we assume that controlling $\mathbf{c}(\mathbf{y}) = 0$ is equivalent to controlling the reduced gradient to zero, as shown in the Section 7.5. Starting in the correct region, the control structure should be switched when:

1. (A new constraint becomes active) When a new constraint becomes active, change the control structure to the corresponding region

2. (A constraint becomes inactive) As soon as the controlled variable **c** in one of the neighboring regions becomes zero (reaches its optimal setpoint), change the control structure to the corresponding region.

Since our controlled variables are derived from the optimality conditions, this method will give optimal operation (and switching), as long as the same optimality conditions cannot be satisfied at two distinct **d**. This will hold if the optimization problem is convex in the disturbance space of interest.

In addition, we have to assume that the regions of active constraints are adjacent, and that a changing disturbance moves the system continuously from one region into another. Although this is the case for many systems in practice, it has to be confirmed that these requirements are satisfied for each case.

Similar to our approach, Woodward et al. [2010] present a method which detects active set changes based on the optimality conditions. Therefore, their approach will be applicable in the same cases as our approach.

However, there are significant differences in the approaches. We separate the steady state optimization problem and the dynamic control problem, by using self-optimizing controlled variables. Once the steady state optimal regions of active constraints are known, and control structures are set up for each region, we start with designing the dynamic controllers and an appropriate switching law, which can handle the dynamic system and avoids e.g. switching back and forth for high frequency disturbances.

In contrast, Woodward et al. [2010] attempt to design an optimizing controller which can handle all complications which come with dynamic systems.

However, the main focus of this work is to find steady state optimal controlled variables and different regions where they should be used. The actual dynamic implementation is a separate problem, which is not considered in depth here.

7.8 Case Study II

We consider an isothermal CSTR with two parallel reactions, as depicted in Figure 7.3, taken from Srinivasan et al. [2008]. The reactor is fed with two feed streams F_A and F_B which contain the reactants A and B in the concentrations c_A and c_B. In the main vessel, the two components react to the desired product C, and the undesired side product D. The reactants A and B are not consumed completely during the reaction, so the outflow contains all four products. The CSTR is operated isothermally, and we assume that perfect temperature control has been implemented.

The products C and D are formed by the reactions:

$$\begin{aligned} \mathrm{A} + \mathrm{B} &\xrightarrow{k_1} \mathrm{C} \\ 2\,\mathrm{B} &\xrightarrow{k_2} \mathrm{D}. \end{aligned} \tag{7.59}$$

We wish to maximize the amount of desired product $(F_A + F_B)c_C$, weighted by a yield factor $(F_A + F_B)c_C/(F_A c_{A,in})$ [Srinivasan et al., 2008]. The amount of heat to remove and the maximum flow rate are limited by the equipment, and we formulate the mathematical optimization problem as follows [Srinivasan et al., 2008]:

$$\max_{F_A,F_B} \frac{(F_A + F_B)c_C}{F_A c_{A_{in}}}(F_A + F_B)c_C \tag{7.60}$$

subject to

$$\begin{aligned} F_A c_{A_{in}} - (F_A + F_B)c_A - k_1 c_A c_B V &= 0 \\ F_B c_{B_{in}} - (F_A + F_B)c_B - k_1 c_A c_B V - 2k_2 c_B^2 V &= 0 \\ -(F_A + F_B)c_C + k_1 c_A c_B V &= 0 \\ F_A + F_B &\leq F_{max} \\ k_1 c_A c_B V(-\Delta H_1) + 2k_2 c_B^2 V(-\Delta H_2) &\leq q_{max}. \end{aligned} \tag{7.61}$$

Here, k_1 and k_2 are the rate constants for the two reactions, $(-\Delta H_1)$ and $(-\Delta H_2)$

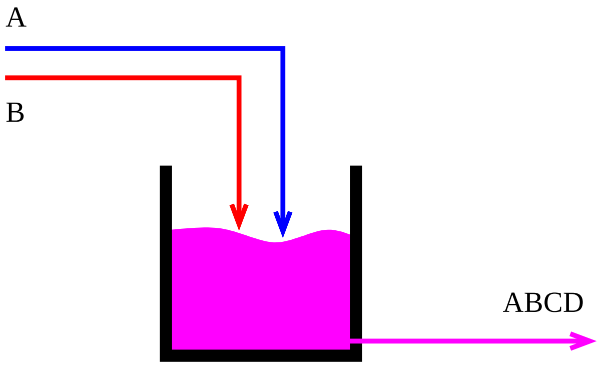


Figure 7.3: CSTR with two reactions

Table 7.4: Overview of variables

Symbol	Description	Comment
F_A	Inflow stream A	Measured input
F_B	Inflow stream B	"
F	total flow	Measured variable
q	Heat produced	"
c_B	Concentration of B	"
c_A	Concentration of A	Unmeasured state
c_C	Concentration of C	"
k_1	Rate constant reaction 1	Unmeasured disturbance

are the reaction enthalpies, q_{max} the maximum allowed heat, V the reactor volume, and F_{max} the maximum total flow rate. The measured variables (**y**), the manipulated variables (**u**), the disturbance variables (**d**), and the internal states (**x**) are given in Table 7.4, and the parameter values of the system are listed in Table 7.5.

We write the combined vector of states $\mathbf{x} = [c_A, c_B, c_C]$ and manipulated variables $\mathbf{u} = [F_A, F_B]$ as

$$\mathbf{z} = \begin{bmatrix} c_A, & c_B, & c_C, & F_A, & F_B \end{bmatrix}^{\mathbf{T}}. \tag{7.62}$$

7.8.1 Identifying operational regions

Next, the system is optimized off-line for the range of possible disturbances $d = k_1$. This shows that the system can be partitioned into three adjacent critical regions,

Table 7.5: Parameters

Symbol	Unit	Value
k_1	l/(mol h)	0.3 - 1.5
k_2	l/(mol h)	0.0014
$(-\Delta H_1)$	J/mol	7×10^4
$(-\Delta H_2)$	J/mol	5×10^4
$c_{A,in}$	mol/l	2
$c_{B,in}$	mol/l	1.5
V	l	500
$F_{\max}$	l/h	22
q_{max}	kJ/h	1000

defined by their active constraints.

The critical regions are visualized in Figure 7.4, where the normalized constraints are plotted over the disturbance range. In the first region, for disturbances below about $k_1 = 0.65\frac{\mathrm{l}}{\mathrm{mol\,h}}$, the flow constraint is the only active constraint. The second critical region for values between about $k_1 = 0.65\frac{\mathrm{l}}{\mathrm{mol\,h}}$ and $k_1 = 0.8\frac{\mathrm{l}}{\mathrm{mol\,h}}$ is characterized by two active constraints, i. e. both the flow constraint and the heat constraint are active. Finally, in the third region, above about $k_1 = 0.8\frac{\mathrm{l}}{\mathrm{mol\,h}}$ only the heat constraint remains.

7.8.2 Eliminating λ

In each critical region, the set of controlled variables contains the active constraints (we know that they should be controlled at the optimum). This leaves the unconstrained degrees of freedom, which is the difference between the number of manipulated variables and the active constraints, $n_{DOF} = n_{\mathbf{z}} - n_g$. For each of the unconstrained degrees of freedom one controlled variable is needed.

In the first critical region this gives $n_{DOF,1} = 5 - 4 = 1$ unconstrained degrees of freedom, so apart from the active constraint, which is the first controlled variable, we need to control one more variable (invariant).

To obtain the reduced gradient, we calculate the null space of Jacobian of the active set $\mathbf{N}_z^{\mathbf{T}}$ and multiply it with the gradient of the objective function $\nabla_{\mathbf{z}} J(\mathbf{z}, \mathbf{d})$ to obtain $J_{z,red,1} = \mathbf{N}_z^{\mathbf{T}} \nabla_{\mathbf{z}} J$. Depending on the algorithm to compute the null space, this may become a fractional expression, but since we want to control the process at the optimum, i. e. we control $J_{z,red,1}$ to zero, it is sufficient to consider only the numerator of $J_{z,red,1}$. This is possible because a fraction vanishes if the numerator is zero (provided the denominator is nonzero which is the case here because $\nabla_{\mathbf{z}} g$

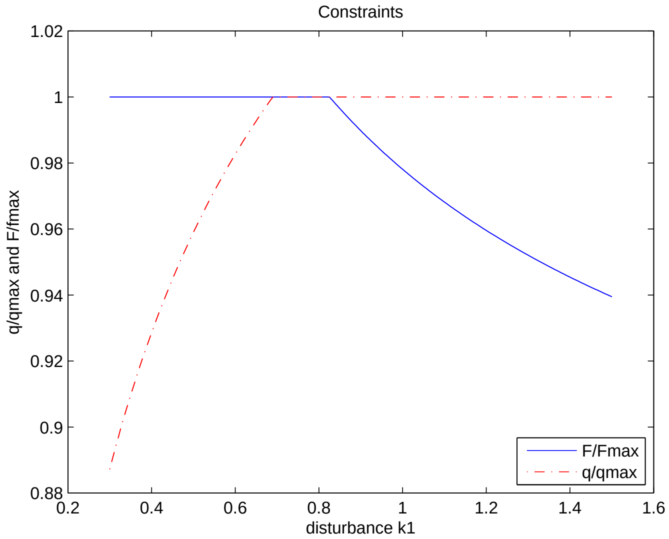


Figure 7.4: Optimal values of the constrained variables

has full rank). For the critical region 1, we obtain from (7.11) the reduced gradient

$$\begin{aligned} J_{z,red,1} = &-(F_A+F_B)^2 c_C \left[-3c_C F_B^2 F_A - 3c_C F_A^2 F_B \right.\\ &- 4c_C c_B F_A^2 k_2 V - 4c_C k_2 V^2 k_1 c_B^2 F_A - c_C F_A^3 \\ &- c_C F_B^3 - 4c_C k_2 V^2 k_1 c_B^2 F_B - c_C c_B F_A^2 k_1 V \\ &- 4c_C c_B F_B^2 k_2 V - c_C c_B F_B^2 k_1 V - c_C F_A^2 c_A k_1 V \\ &- c_C F_B^2 c_A k_1 V - 8c_C F_A c_B F_B k_2 V \\ &- 2c_C F_A c_B F_B k_1 V - 2c_C F_A F_B c_A k_1 V \\ &+ 8F_A k_1 V^2 c_{A,in} k_2 c_B^2 + 2F_A^2 k_1 V c_B c_{A,in} \\ &+ 2F_A k_1 V F_B c_B c_{A,in} - 2F_A^2 k_1 V c_{B,in} c_A \\ &\left. -2F_A k_1 V F_B c_{B,in} c_A \right], \end{aligned} \tag{7.63}$$

which should be controlled to zero. This expression may be simplified slightly, since it is known that $(F_A+F_B)^2 c_C \neq 0$. It is therefore sufficient to control the factor in square brackets in (7.63) to zero.

Similarly, in the second critical region $n_{DOF,2} = 5-5 = 0$, and here we simply control the active constraints, keeping q at $q_{\max}$ and F at $F_{\max}$.

In the third critical region $n_{DOF,3} = 5-4 = 1$, and we use one of the manipulated variables to control the active constraint ($q = q_{max}$) while the other one is used to control the invariant measurement combination $J_{z,red,3}$, which is an expression similar to (7.63).

7.8.3 Eliminating unknown variables

The reduced gradients for the first and the third critical region $J_{z,red,1}$ and $J_{z,red,3}$ still contain unknown variables, namely k_1, c_A and c_C, and cannot be used for feedback control directly.

To arrive at variable combinations which can be used for control, we include all known variables into $\mathbf{y}$, and all unknown variables into $\hat{\mathbf{d}}$, such that $\hat{\mathbf{d}} = [k_1, c_A, c_C]$. Then we write the necessary conditions for optimality for each region as

$$\begin{aligned} J_{z,red}(\mathbf{y}, \hat{\mathbf{d}}) &= 0 \\ g(\mathbf{y}, \hat{\mathbf{d}}) &= 0. \end{aligned} \tag{7.64}$$

Considering the known variables $\mathbf{y}$ as parameters of the system, we want to find conditions on these parameters such that (7.64) is satisfied. The system has $n_{\hat{d}} = 3$ unknown variables, k_1, c_A and c_C, of which we know that they are not zero. This corresponds to solutions $[k_1, c_A, c_C] \in (\mathbb{C}^*)^3$. According to section 7.5 we have that (7.64) is satisfied if and only if the sparse resultant is zero.

In critical region 1 and 3, the number of equations $n_{eq} = 5$ (model equations+active constrains+invariant), and the number of unknowns $n_{\hat{d}} = 3$. Hence we have more equations than necessary. Since we assume no measurement noise, all measurements are equally good, and we may select a subset of $n_{\hat{d}} + 1$ equations from (7.64) to compute the sparse resultant for the subset of equations. Obviously, the reduced gradient must be contained in this set of equations. Alternatively, as we do in the following, we can eliminate one more variable from the invariant.

For the first region, we use the sparse resultant of the system consisting of the invariant (7.63), the model equations (the first three equality constraints in (7.61)) and the first (active) inequality constraint in (7.61) to eliminate k_1, c_A, c_C and F_B and to calculate the controlled variable combination. The computations were performed using the `multires` software [Busé and Mourrain, 2003]. After division by nonzero factors, the controlled variable for region 1 becomes

$$\begin{aligned} c_1 = &-c_{b,in}^2 F_A^2 - F_A^2 c_{A,in} c_{b,in} + 6F_A c_{A,in} k_2 c_b^2 V + 2F_A c_{A,in} F_{max} c_b \\ &- F_A c_{A,in} F_{max} c_{b,in} + F_{max}^2 c_b^2 + c_{b,in}^2 F_{max}^2 + 4V^2 k_2^2 c_b^4 \\ &- 2c_{b,in} F_{max}^2 c_b - 4V k_2 c_b^2 c_{b,in} F_{max} + 4V k_2 c_b^3 F_{max}. \end{aligned} \tag{7.65}$$

Note that this invariant for control has become simpler than the expression for the reduced gradient (7.63).

In the second critical region, control is simple; the two manipulated variables are used to control the two active constraints $F = F_{max}$ and $q = q_{max}$.

The third critical region is controlled similar to the first one. One input variable is used to control the active constraint, and the second input is used to control the resultant. The model equations (the first three equations together with the energy constraint) in (7.61) and the reduced gradient are used to compute the resultant. Thus the unknown variables k_1, c_A, c_C, and F_B are eliminated from the reduced gradient. The controlled variable for region 3 is

$$\begin{aligned}
c_3 = {} & -4Vc_B^2k_2\Delta H_2F_Ac_{A,in}c_{B,in}q_{max}\Delta H_1 + F_Ac_{B,in}^2q_{max}^2\Delta H_1 \\
& + 4V^2c_B^4k_2^2\Delta H_2F_Ac_{A,in}c_{B,in}\Delta H_1^2 - 4V^2c_B^4k_2^2\Delta H_2^2F_Ac_{A,in}c_{B,in}\Delta H_1 \\
& - 2Vc_B^2k_2F_Ac_{A,in}c_{B,in}\Delta H_1^2q_{max} - 4Vc_B^2k_2\Delta H_2F_Ac_{B,in}^2\Delta H_1q_{max} \\
& - 2Vc_B^2k_2\Delta H_2F_A^2c_{A,in}c_{B,in}^2\Delta H_1^2 + 8Vc_B^3k_2\Delta H_2\Delta H_1F_Ac_{A,in}q_{max} \\
& - 8V^2c_B^4k_2^2\Delta H_2c_{B,in}\Delta H_1q_{max} - 12V^2c_B^4k_2^2F_A\Delta H_2^2c_{B,in}^2\Delta H_1 \\
& - 8V^2c_B^5k_2^2\Delta H_2F_Ac_{A,in}\Delta H_1^2 + 8V^2c_B^5k_2^2\Delta H_2^2\Delta H_1F_Ac_{A,in} \\
& + 8V^2c_B^5k_2^2F_A\Delta H_2^2c_{B,in}\Delta H_1 - q_{max}^3c_{B,in} + 2c_Bq_{max}^3 \\
& - \Delta H_1c_{B,in}F_Ac_{A,in}q_{max}^2 + 2c_BF_Ac_{A,in}q_{max}^2\Delta H_1 + F_A^2c_{A,in}c_{B,in}^2\Delta H_1^2q_{max} \\
& - 2c_BF_Ac_{B,in}q_{max}^2\Delta H_1 + 8Vc_B^3k_2\Delta H_2q_{max}^2 + 8V^2c_B^5k_2^2\Delta H_2^2q_{max} \\
& + 8V^3c_B^6k_2^3\Delta H_2^3c_{B,in} - 2c_BF_A^2c_{A,in}c_{B,in}\Delta H_1^2q_{max} \\
& - 2Vc_B^2k_2\Delta H_1q_{max}^2c_{B,in} - 2Vc_B^2k_2\Delta H_2q_{max}^2c_{B,in} \\
& + 4V^2c_B^4k_2^2\Delta H_2^2c_{B,in}q_{max} - 8V^3c_B^6k_2^3\Delta H_2^2c_{B,in}\Delta H_1.
\end{aligned} \tag{7.66}$$

Due to the structure of the polynomials in region 3, here the invariant has become more complicated after eliminating the unknown variables.

Although the expressions are quite complicated, they contain only known quantities, and can be simply evaluated and used for control. Before actually using the measurement combinations for control, they are scaled so that the order of magnitude is similar. That is, c_1 is scaled (divided) by F_{max}, and c_2 is scaled by $\Delta H_1^2\Delta H_2F_AF_B$.

7.8.4 Using measurement invariants for control and region identification

Having established the controlled variables for the three critical regions, it remains to determine when to switch between the regions. Starting in the first critical region, the flow rate is controlled such that $F_A + F_B = F_{max}$, and the first measurement combination c_1 is controlled to zero. As the value of the disturbance k_1 rises, the reaction rate increases as well as the required cooling to keep the system isothermal,

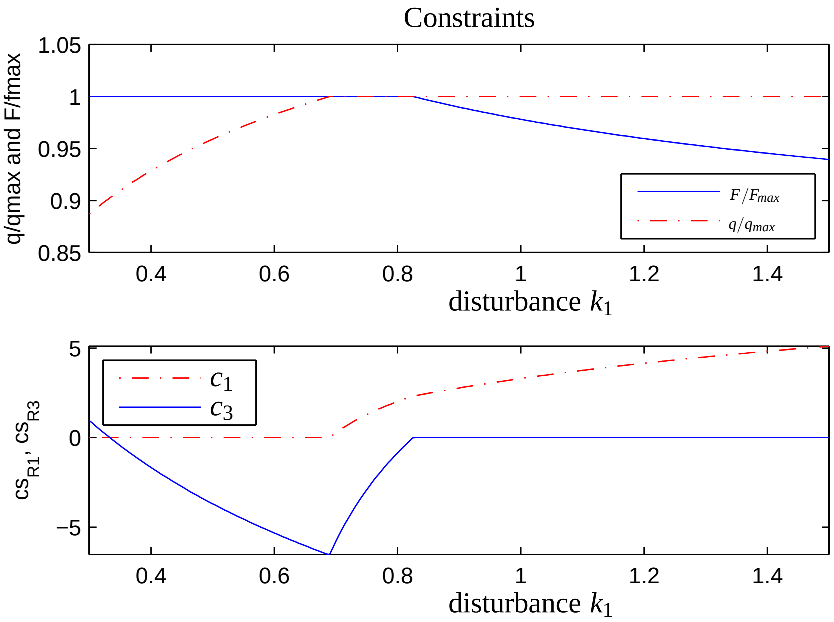


Figure 7.5: Optimal values of controlled variables

until maximum cooling is reached, Figure 7.5. When the constraint is reached, the control structure is switched to the next critical region, where the inputs are used to control $q = q_{max}$ and $F_A + F_B = F_{max}$. While operating in the second region, the controlled variables of the neighboring regions are monitored. As soon as one of the variables c_1 or c_3 reaches its optimal setpoint (i. e. 0) for its region the control structure is changed accordingly. Specifically, when k_1 is further increased, such that $c_3 = 0$ is reached, we must keep $F_A + F_B < F_{max}$ such to maintain the value $c_3 = 0$.

Table 7.6: Parameters for dynamic Simulation

Variable	Value	Unit	Description
c_p	4.0	kJ/kgK	Heat capacity
ρ	1.0	kg/l	Density
$T_{in,A}$	350	K	Temperature of feed A
$T_{in,B}$	350	K	Temperature of feed B

7.8.5 Implementation and dynamic simulations

We modify the model (7.61) to become a dynamic model,

$$\begin{aligned}
\dot{n}_A &= F_A c_{A_{in}} - F_{out} c_A - k_1 c_A c_B V \\
\dot{n}_B &= F_B c_{B_{in}} - F_{out} c_B - k_1 c_A c_B V - 2k_2 c_B^2 V \\
\dot{n}_C &= -F_{out} c_C + k_1 c_A c_B V \\
\dot{n}_D &= -F_{out} c_D + 2k_2 c_B^2 V \\
\dot{H} &= F_A c_p \rho (T_{in,A} - T) + F_B c_p \rho (T_{in,B} - T) + k_1 c_A c_B V(-\Delta H_1) + 2k_2 c_B^2 V(-\Delta H_2) - q \\
\dot{V} &= F_A + F_B - F_{out},
\end{aligned} \tag{7.67}$$

with n_i as the number of moles of component $i = A, B, C, D$, and the additional parameters in Table 7.6, and with the enthalpy H.

Implementation of optimal operation is straightforward, as simple PI controller can be used to control the invariants. In the steady state case, we have assumed that we have ideal temperature and level control under operation. In practice, this has to be achieved by control. Therefore we have included temperature and level controllers in our dynamic simulations. Further, it is assumed that the temperature should be 350 K under operation.

We present only simulations for regions 1 and 3, because since this involves the polynomial invariants. In region 2, only the active constraints are controlled.

Control structure in region 1 All variables are controlled using simple PI controllers. The control structure in region 1 is presented in Figure 7.6. The cooling duty q is used to control the temperature, and the feed flow F_B is used to control the invariant. Further, we use the outflow F_{out} to keep control the level, and the feed F_A to controlling the throughput to $F = F_{max}$. In order not to violate the heat constraint q_{max}, the controller output goes into saturation when the constraint is reached.

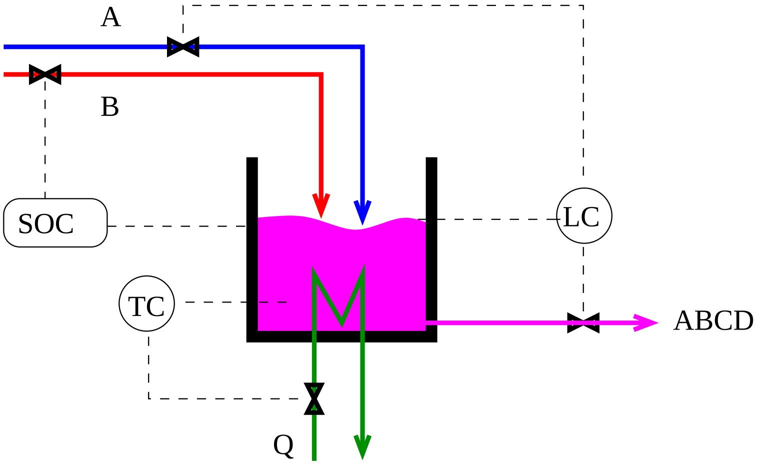


Figure 7.6: Variable pairings for Region 1

Control structure in region 3 All variables are controlled using PI controllers. The pairing was selected as in Figure 7.7. Since it is optimal to keep q at q_{max} in this region, it is set to this value in open loop. The feed flow F_A is used to control the invariant, and the feed flow F_B is used to control the temperature. As in region 1, the outflow F_{out} is used to stabilize the level of the reactor.

7.8.6 Simulation results

In Figure 7.8 we show the dynamic behavior of the system in regions 1 and 3. The left column shows the response to a disturbance in region one, and the right column shown the response to a disturbance in region 3. Note the differences in the magnitude of the disturbances in the first row. The controlled variables (constraints and self-optimizing invariants have been plotted in red, and they are nicely kept at their setpoints.

The simulations in both regions demonstrate nicely that it is possible to control the obtained invariants using simple PI controllers.

7.9 Discussion

The presented method is based on the same idea as NCO tracking [François et al., 2005]. However in contrast to François et al. [2005], where the optimality conditions are solved for the optimizing *inputs*, this work focuses on finding the right *outputs* which express the optimality conditions. The problem of generating the

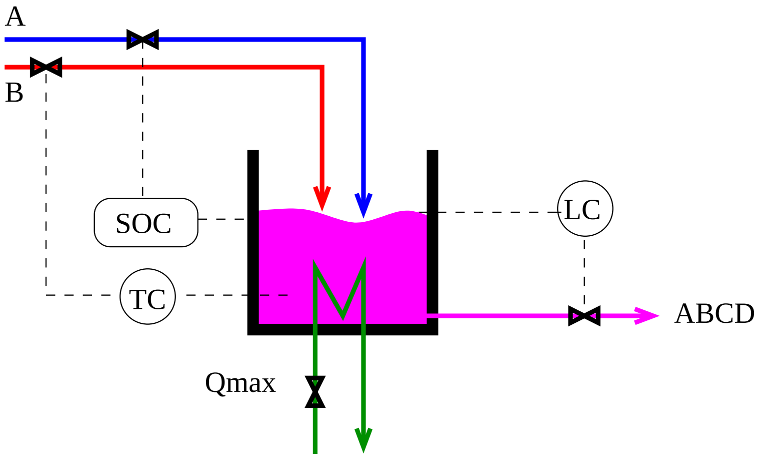


Figure 7.7: Variable Pairings for Region 3

inputs which control the outputs to zero is dealt with separately. In most cases, inputs are generated by feedback control, e.g. PI controllers.

The method was developed as an alternative derivation and a generalization of the existing null space method [Alstad and Skogestad, 2007] for linear systems.

In the linear case, eliminating the constraints is straightforward, while this is not trivial in the polynomial case. By premultiplying ∇J by the null space of the constraints $\mathbf{N}^{\mathbf{T}}$, we eliminate the Lagrangian multipliers from the equation set, and obtain the reduced gradient for the nonlinear case.

The elimination of the Lagrangian multipliers could also been done simultaneously with the other unknown variables, using the resultant. Under the strict complementarity condition (either $\lambda_i = 0$ *or* the constraint is active), the solutions for λ lie in the toric variety, and therefore the sparse resultant gives necessary and sufficient conditions on the known variables so that the KKT system has a solution. We chose to apply the two-step procedure in this work instead of eliminating everything by using the resultant because this results in lower computational load when computing the resultants.

As an alternative to calculating resultants, our initial approach was to attempt to compute the controlled variable combinations by Gröbner bases with an appropriate elimination ordering [Cox et al., 1992]. We tried to find an appropriate monomial ordering which eliminates the unknown variables, and then use a polynomial from the elimination ideal as controlled variable. However, with the Gröbner basis approach it is not straightforward to find an elimination order that eliminates the unknown variables from the equation system while not yielding the

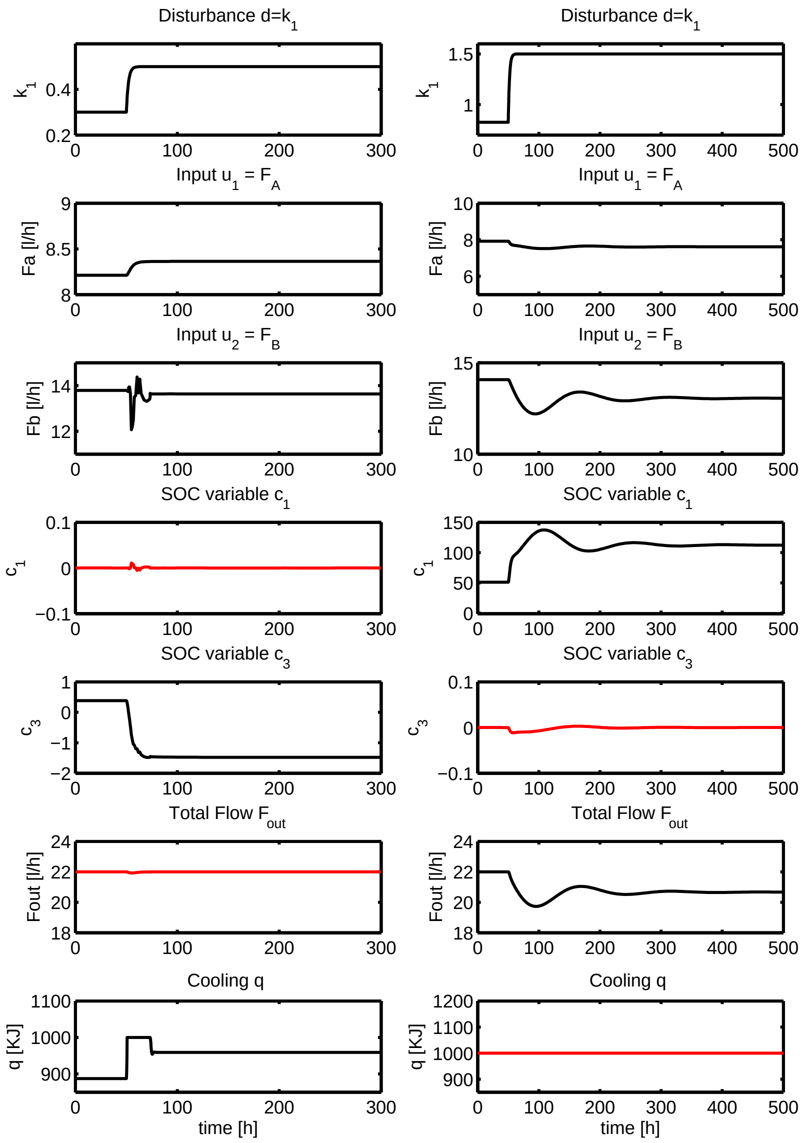


Figure 7.8: Simulation in region 1 (left) and region 3 (right). Controlled variables in each region (self-optimizing variable and active constraints) in red

"trivial solution" (i. e. the invariant is always zero when the constraints are satisfied). Another disadvantage with the Gröbner basis approach is that the selected invariant may give rise to additional "artificial solutions" which are not solutions of the original optimality conditions.

A similar approach is to calculate a Göbner basis for the ideal generated by the active constraints $g(\mathbf{y}, \hat{\mathbf{d}})$ and $m(\mathbf{y}, \hat{\mathbf{d}})$ using some elimination ordering, and to reduce $\mathbf{N}^{\mathbf{T}}\nabla_{\mathbf{z}}J$ modulo the ideal. This avoids the trivial solution, however, the problem of choosing a monomial ordering which eliminates all unknown variables, remains.

Generally the Gröbner basis approach tends to give even more complicated expressions than the sparse resultant approach presented here.

Since the sparse resultants can also give "large" expressions, our method is best suited for small systems, with few constraints and equations. This is further emphasized by the fact that calculating the analytical determinant for large matrices is computationally demanding and that the construction of the resultant matrices is based on the computation of the mixed volume, which is a hard enumerative problem [Cox et al., 2005]. However, large systems can often be decomposed into smaller subsystems which can be considered independently. Our method may be applicable for such a subsystem.

If there are more polynomial equations than unknowns, the engineer must choose which polynomials to use in the resultant calculations in addition to the reduced gradient. From a purely mathematical view, this does not make any difference, as long as the set of model equations has finitely many solutions for $\mathbf{d}$. However the controlled variables will look quite different for different choices. The best (in terms of simplicity) choice depends on the structure of the equations, and is thus specific to the problem. However, as a general guideline, it would be advisable to keep the degrees of the polynomials low in the unknown variables. This leads to simpler resultants.

Although we can specify which variables to eliminate from the reduced gradient, the variables which remain depend strongly on the structure of the model equations and the eliminated variables. In some cases all information about the optimum is contained in very few variables, in other cases many variables are needed to specify the optimum.

The resultant method, as presented in this paper, does not take into account measurement noise or model error. This is beyond the scope of this work. Our goal was to extend the idea of the null-space method [Alstad and Skogestad, 2007], and to demonstrate that the concept of finding variables which remain constant at optimal operation is possible also for polynomial systems.

Since our approach is to separate the controlled variables selection procedure and the controller design, it must be verified, that the measurement invariants are controllable using simple controllers. That is, we have to make sure that the controlled variables actually crosses zero [3]. Proving this mathematically is not trivial and a subject for further research. However, this problem is beyond the scope of this paper, since we are only interested in finding variables which contain only known quantities, and which characterize the optimum. In all our experience so far, the measurement invariants could be controlled by simple PI controllers.

Apart from handling noisy measurements, model mismatch and controllability issues, a subject for future research is to find methods which reproduce not all solutions of the optimality conditions, but only a certain set of interest. This could be all the real solutions or solutions which reside in some further specified semialgebraic set, so that stationary points which are saddle points or maxima are excluded.

7.10 Conclusions

In this paper we have presented an approach to obtain optimal steady state operation which does not require on-line calculations. We have shown that, after identifying the critical regions, there exist optimally invariant variable combinations for each region. If there are enough measurement/model relations ($n_g + n_m \geq n_{\hat{d}}$), the unknown variables can be eliminated by measurements and system equations, and the invariant combinations can be used for control using a decentralized self-optimizing control structure.

The dynamic simulations show that it is possible to control the invariants using simple PI controllers.

Depending on the problem structure, the invariants may become more complicated than the reduced gradient, as in region 3 of case study II, or much simpler, as in case study I and region 1 of case study II. Due to the symbolic nature of the calculations, this method is best suited for small systems. However, it is difficult to predict how the invariant will look like without analyzing the system more thoroughly (that is considering the mixed volumes of the corresponding Newton polytopes). Moreover, since the analysis is based on an exact model, it is difficult to make a-priori statements about the robustness to noise and model error. It is expected, that simple invariants of low degree will behave better than invariants with of high degrees in the variables. Thus for example the invariant of case study I can be expected to be relatively robust to measurement noise. However, this has to be evaluated individually for each case.

[3] For example, this would not happen when $c = x^2$ and we have an input with gain $x = Gu$, in this case the input has to move into different directions, depending on the value of x. This would be difficult to control in practice. However, this particular example cannot happen, because $c = x^2$ is not irreducible.

Further, we have shown that the measurement invariants can be used for detecting changes in the active set and for finding the right region to switch to. The active set changes are strictly speaking only valid for steady state operation. This assumes that the disturbance changes occur on a frequency which is low enough for the system to reach steady state. Faster disturbances have to be handled by the dynamic controllers.

Using methods from elimination theory, we have shown that in principle, measurement polynomials can be used as self-optimizing controlled variables to control the process optimally.

References

Abel, N. H., 1826. Beweis der Unmöglichkeit, algebraische Gleichungen von höheren Graden als dem vierten allgemein aufzulösen. Journal für die reine und angewandte Mathematik, 65–84.

Alstad, V., 2005. Studies on selection of controlled variables. Ph.D. thesis, Norwegian University of Science and Technology, Department of Chemical Engineering.

Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. Industrial & Engineering Chemistry Research 46, 846–853.

Alstad, V., Skogestad, S., Hori, E. S., 2009. Optimal measurement combinations as controlled variables. Journal of Process Control 19 (1), 138–148.

Baotić, M., Borrelli, F., Bemporad, A., Morari, M., 2008. Efficient on-line computation of constrained optimal control. SIAM Journal on Control and Optimization 47, 2470–2489.

Bazaraa, M. S., Sherali, H. D., Shetty, C. M., 2006. Nonlinear Programming: Theory and Algorithms. John Wiley & Sons.

Busé, L., Mourrain, B., 2003. Using the maple multires package.
URL http://www-sop.inria.fr/galaad/software/multires

Canny, J. F., 1990. Generalised characteristic polynomials. Journal Symbolic Computation 9, 241–250.

Cao, Y., 2003. Self-optimizing control structure selection via differentiation. In: Proceedings of the European Control Conference. pp. 445–453.

Chachuat, B., Srinivasan, B., Bonvin, D., 2009. Adaptation strategies for real-time optimization. Computers & Chemical Engineering 33 (10), 1557 – 1567.

Cox, D., Little, J., O'Shea, D., 1992. Ideals, Varieties, and Algorithms. Springer-Verlag.

Cox, D., Little, J., O'Shea, D., 2005. Using Algebraic Geometry. Springer.

Dickenstein, A., Emiris, I. Z., 2005. Solving Systems of Polynomial Equations. Springer Berlin Heidelberg.

Emiris, I. Z., Mourrain, B., 1999. Matrices in elimination theory,. Journal of Symbolic Computation 28 (1-2), 3–43.

François, G., Srinivasan, B., Bonvin, D., 2005. 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.

Gelfand, I. M., Kapranov, M. M., Zelevinsky, A. V., 1994. Discriminants, Resultants and Multidimensional Determinants. Birkhäuser, Boston, MA.

Halvorsen, I. J., Skogestad, S., 1997. Indirect on-line optimization through setpoint control. AIChE 1997 Annual Meeting, Los Angeles; paper 194h.

Halvorsen, I. J., Skogestad, S., Morud, J. C., Alstad, V., 2003. Optimal selection of controlled variables. Industrial & Engineering Chemistry Research 42 (14), 3273–3284.

Heldt, S., 2009. On a new approach for self-optimizing control structure design. In: ADCHEM 2009, July 12-15, Istanbul. pp. 807–811.

Jäschke, J., Skogestad, S., 2010. Self-optimizing control and NCO tracking in the context or real-time optimization. In: Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems, DYCOPS 2010, July 5-7, Leuven, Belgium. pp. 593–598.

Kariwala, V., Cao, Y., Janardhanan, S., 2008. Local self-optimizing control with average loss minimization. Industrial & Engineering Chemistry Research 47, 1150–1158.

Marlin, T. E., Hrymak, A., 1997. Real-time operations optimization of continuous processes. In: Proceedings of CPC V, AIChE Symposium Series vol. 93. pp. 156–164.

Morari, M., Stephanopoulos, G., Arkun, Y., 1980. Studies in the synthesis of control structures for chemical processes. Part I: Formulation of the problem. Process decomposition and the classification of the control task. Analsysis of the optimizing control structures. AIChE Journal 26 (2), 220–232.

Nocedal, J., Wright, S., 2006. Numerical Optimization. Springer.

Pistikopoulos, E., Georgiadis, M., Dua, V., 2007. Multiparametric programming. Wiley-VCH.

Rojas, J. M., 1999. Solving degenerate sparse polynomial systems faster. J. Symbolic Computation 28, 155–186.

Skogestad, S., 2000. Plantwide control: The search for the self-optimizing control structure. Journal of Process Control 10, 487–507.

Srinivasan, B., Biegler, L. T., Bonvin, D., 2008. Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function. Computers & Chemical Engineering 32, 572–279.

Sturmfels, B., 1994. On the newton polytope of the resultant. Journal of Algebraic Combinatorics 3, 207–236.

Sturmfels, B., 2002. Solving systems of polynomial equations.

Woodward, L., Perrier, M., Srinivasan, B., 2010. Real-time optimization using a jamming-free switching logic for gradient projection on active constraints. Computers & Chemical Engineering 34 (11), 1863 – 1872.

Chapter 8

Invariants for dynamic systems

O heaven! were man but constant,
he were perfect.

W. Shakespeare

In optimal control the optimal input trajectories are often solved numerically or analytically from the optimality conditions. This requires that all variables which enter the optimality conditions are known or measured. We use techniques from polynomial elimination theory to eliminate variables which are not known from the optimality conditions. The result is an expression of the optimality conditions in known variables only, which can easily be evaluated and controlled by feedback.

Submitted to the 50th IEEE Conference on Decision and Control and European Control Conference, 2011, Orlando

8.1 Introduction

Dynamic optimization problems are ubiquitous in science and engineering. In process control, they are found in the optimization of batch reactors or grade transitions in continuous processes. Although many problems can be approximated well by optimizing the behavior at steady state, other cases are inherently of dynamic nature, and must be approached by dynamic optimization.

Most approaches in literature deal with optimization based on a model. One of the oldest approaches is to find the optimal input using the Pontryagin minimum principle [Bryson and Ho, 1975]. This requires a simple model, and that all parameters and variables are known (measured).

A second approach, which is very popular today and can handle more complex models, is nonlinear predictive control, or dynamic real-time optimization. Here, we use measurements to update the process model parameters at given sample times, and solve the optimization problem iteratively using the updated model in order to obtain new input trajectories [Allgöwer and Zheng, 2000; Grötschel et al., 2001; Diehl et al., 2002; Schlegel et al., 2005].

A third approach, which may be placed in between the two previous, is to use the model off-line and exploit the solution structure to find variables, which give optimal or near optimal operation, when kept at constant setpoints using a feedback policy. This approach is followed in NCO tracking [Srinivasan et al., 2003] and self-optimizing control [Skogestad, 2000].

Whenever using a model to find an optimal operation strategy, handling uncertainty is a major challenge. Uncertainty may arise from different sources, such as incomplete information (unmeasured states), parametric disturbances and model structure error. To cope with uncertainties, there are several approaches:

1. Keep the unknown or varying parameters at some fixed value. In most cases this will give poor performance, or even feasibility problems.

2. Estimate the unknown variables using some filter or moving horizon estimation [Kühl et al., 2011]. This approach is used frequently, however, it can be difficult to obtain converging estimates for the unknowns within reasonable time.

3. Use a robust control approach [Terwiesch et al., 1994]. Here we attempt to find a control policy, which gives the best performance over a range of disturbances. Generally it has to compromise performance to gain robustness.

4. Neighbouring extremal control [Bryson and Ho, 1975], where the original optimization problem does not have to be re-solved completely. Instead, an easier, linear problem is solved to find the required corrections to the nominal input trajectory.

5. The approach presented in this paper, where we use model equations to eliminate the unknown or uncertain variables from the optimality conditions.

Our work contributes to handling parametric uncertainty for dynamic optimization problems, which are given as polynomial equations. The main contribution of this paper is to extend the ideas from steady state self-optimizing control Skogestad [2000] to dynamic optimization problems which are described by polynomial equations.

Our method is in the same spirit as Srinivasan et al. [2003] and Alstad and Skogestad [2007], where the controlled variables are the optimality conditions associated to the corresponding optimization problem. The idea is to formulate the optimality conditions ($H_u = 0$) which include unknown parameters, and then use tools from elimination theory [Emiris and Mourrain, 1999; Cox et al., 2005] for eliminating the unknown parameters to find invariants of optimal control systems, so that a feedback solution in known (measured) variables can be found.

Our approach consists of four steps: First, we formulate the optimality conditions. Second, we eliminate the adjoint variables from the optimality conditions, and third, we use model equations to eliminate unmeasured states and parameters from the optimality conditions to obtain optimally invariant variable combinations in known variables only. Finally, we use feedback to control these variable combinations.

This paper is structured as follows. In Section 8.2 we present the optimal control problem in consideration, and describe how to find combinations of variables which remain constant under optimal control. That is, how the adjoint variables can be eliminated from the optimality conditions. In Section 8.3, we introduce some basic concepts from toric elimination theory, which we will use for eliminating other unknown variables from the invariant. In Section 8.4, we show how to apply the results from elimination theory to eliminate unknown parameters from the optimality conditions. Section 8.5 gives a case study of a fed batch reactor, where we find controlled variables which do not contain unmeasured variables. Section 8.6 closes the paper with a short discussion and conclusion.

8.2 Optimal control

We assume that the dynamic optimization problem can be written in following form:

$$\begin{aligned}
\min_{\mathbf{u}(t)} \Phi(t_f) &= J(\mathbf{x}(t_f)) && (8.1)\\
\text{s.t.} \quad & \dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}(t)) + \mathbf{G}(\mathbf{x}(t))\mathbf{u}(t) && (8.2)\\
& \mathbf{x}(0) = \mathbf{x}_0 && (8.3)\\
& \mathbf{u}(t) \geq \mathbf{u}^L && (8.4)\\
& \mathbf{u}(t) \leq \mathbf{u}^U. && (8.5)
\end{aligned}$$

The scalar function J denotes the terminal cost, and the functions $\mathbf{u} : [0, t_f] \to \mathbb{R}^{n_\mathbf{u}}$ and $\mathbf{x} : [0, t_f] \to \mathbb{R}^{n_\mathbf{x}}$ denote the input and state functions, respectively. $\mathbf{F}(\mathbf{x})$ is a vector valued function of dimension $n_\mathbf{x}$, and $\mathbf{G}(\mathbf{x})$ is a matrix of dimension $n_\mathbf{x} \times n_\mathbf{u}$. The elements of $\mathbf{F}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ are polynomials in the ring $\mathbb{R}[\mathbf{x}]$, that is, every row in $\mathbf{F}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ contains polynomials in the variables $\mathbf{x}$ and coefficients in $\mathbb{R}$. The variables $\mathbf{u}^L$ and $\mathbf{u}^U$ denote the time invariant lower and upper bounds for the inputs $\mathbf{u}$. Note that the system is input affine and we have only input constraints. All functions are assumed to be sufficiently smooth and differentiable.

8.2.1 First order optimality conditions

Assumption 1. *The optimal control problem* (8.1) – (8.5) *is feasible and has a unique solution* $\mathbf{u}^*(t)$*.*

We define the Hamiltonian

$$\begin{aligned} H(\mathbf{x}(t),\mathbf{u}(t),\lambda(t),\mu^L(t),\mu^U(t)) = {} & \lambda^{\mathbf{T}}(\mathbf{F}(\mathbf{x}(t))+\mathbf{G}(\mathbf{x}(t))\mathbf{u}(t)) \\ & + \mu^{L^{\mathbf{T}}}(\mathbf{u}^L-\mathbf{u}(t))+\mu^{U^{\mathbf{T}}}(\mathbf{u}(t)-\mathbf{u}^U), \end{aligned} \tag{8.6}$$

where λ, μ^L and μ^U are the adjoint variables corresponding to the model, the lower and upper input constraints, respectively.

Theorem 11 (Pontryagin Minimum Principle [Geering, 2007; Bryson and Ho, 1975])**.** *If the control* $\mathbf{u}$ *is optimal, then there exist nontrivial vectors of adjoint variables* λ *and* μ*, such that the following conditions are satisfied (we omit to write the explicit time dependency):*

1.

$$\begin{aligned} \dot{\mathbf{x}} &= \frac{\partial H}{\partial \lambda} \\ \mathbf{x}(0) &= \mathbf{x}_0 \\ \dot{\lambda}^{\mathbf{T}} &= -\frac{\partial H}{\partial \mathbf{x}} \\ \lambda^{\mathbf{T}}(t_f) &= \frac{\partial J}{\mathbf{x}(t_f)} \\ \mu^{L^{\mathbf{T}}}(\mathbf{u}^L-\mathbf{u}) &= 0 \\ \mu^{U^{\mathbf{T}}}(\mathbf{u}-\mathbf{u}^U) &= 0 \end{aligned} \tag{8.7}$$

2. For all $t \in [t_0, t_f]$*, the Hamiltonian has a global minimum with respect to* $\mathbf{u}$*, i.e.*

$$H(\mathbf{x}^{opt},\mathbf{u}^{opt},\lambda^{opt},\mu^{L^{opt}},\mu^{U^{opt}}) \le H(\mathbf{x}^{opt},\mathbf{u},\lambda^{opt},\mu^{opt},\mu^{L^{opt}},\mu^{U^{opt}}) \tag{8.8}$$

for all $\mathbf{u}^L \le \mathbf{u} \le \mathbf{u}^U$ *and* $t \in [t_0, t_f]$*.*

3. If the final time is free, we have the transversality condition

$$H(\mathbf{x}(t_f),\mathbf{u}(t_f)^*,\lambda(t_f),\mu^L(t_f),\mu^U(t_f)) = 0. \tag{8.9}$$

8.2.2 Implementing the optimal solution

The solution of problem (8.1) – (8.5) generally consists of a sequence of arcs which are defined in certain intervals. We treat each arc of the optimal solution separately. The arcs are defined by the active constraints, and are continuous and differentiable within each interval [Bryson and Ho, 1975]. We can distinguish two types of arcs:

1. Constrained arcs (boundary arcs): One or more inputs are at a constraint.

2. Unconstrained arcs: The inputs are all unconstrained.

In the constrained arcs, implementation is easy; we simply keep the inputs at the active constraint. If there are unconstrained degrees of freedom left, the remaining problem can be reformulated as an unconstrained problem by redefining the input set. Therefore, in the following, we consider only the case where no constraint is active. To specify the optimal trajectory, we need one controlled variable for each unconstrained degree of freedom.

At the minimum of the Hamiltonian (8.8), we must have

$$H_{\mathbf{u}} = \frac{\partial H}{\partial \mathbf{u}} = 0. \tag{8.10}$$

We simplify notation by considering one input at a time,

$$H_{\mathbf{u}_i} = \frac{\partial H}{\partial \mathbf{u}_i} = 0, \tag{8.11}$$

for all inputs $i = 1, \ldots, n_{\mathbf{u}}$. Unfortunately, we cannot control $H_{\mathbf{u}_i}$ to zero, because it contains unknown adjoint variables λ. To eliminate the adjoint variables, we perform successive time differentiations.

Definition 10 (Lie bracket, [Marquez, 2003]). *Given two vector fields $f, g : \mathbb{R}^n \to \mathbb{R}^n$. The Lie bracket $[f, g]$ is the vector field defined by*

$$[f, g] = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g. \tag{8.12}$$

Recursive bracketing is defined as

$$ad_f^k g = [f, ad_f^{k-1} g], \tag{8.13}$$

with $ad_f^0 g = g$.

It can be shown [Srinivasan et al., 2003; Gros et al., 2009], that the k-th time derivative of $H_{\mathbf{u}_i}$ can be written as

$$H_{\mathbf{u}_i}^{(k)} = \frac{\mathrm{d}^{(k)} H_{\mathbf{u}_i}}{\mathrm{d}t^{(k)}} = \lambda^{\mathbf{T}} \left(ad^k_{\mathbf{F}(\mathbf{x})} \mathbf{G}_i(\mathbf{x}) \right) \\ = \lambda^{\mathbf{T}} A^i_k, \tag{8.14}$$

where $\mathbf{G}_i(\mathbf{x})$ denotes the i-th column in $\mathbf{G}(\mathbf{x})$. Since $H_{\mathbf{u}_i} = 0$, is zero at all times, its time derivatives must be zero at all times, too. Therefore, we can write the time derivatives up top the $n_{\mathbf{x}} - 1$-th derivative as

$$\lambda^{\mathbf{T}} \left[A^i_0, A^i_1, A^i_2, \ldots A^i_{n_{\mathbf{x}}-1} \right] = \lambda^{\mathbf{T}} \mathbf{A}^i = 0, \tag{8.15}$$

where all terms A^i_0, A^i_1, A^i_2,..., $A^i_{n_{\mathbf{x}}-1}$ are collected in the matrix $\mathbf{A}^i$. Equation 8.15 has a nontrivial solution for λ only if

$$\det(\mathbf{A}^i) = 0. \tag{8.16}$$

Therefore, controlling

$$c_i = \det(\mathbf{A}^i) \tag{8.17}$$

to zero gives optimal operation. If we have several inputs, we may collect all c_i into a vector $\mathbf{c} = [c_1, \ldots, c_i, \ldots, c_{n_u}]^{\mathbf{T}}$. The vector $\mathbf{c}$ will generally still contain unknown variables, such as unmeasured states or unmeasured disturbances $\mathbf{d}$. Therefore it cannot be evaluated online and used for control. However, after eliminating unknown variables from $\mathbf{c}$, it can be used for control and the correct inputs can be generated automatically by a P or PI feedback controller.

Since the optimal control system (8.1)–(8.5) is defined in polynomial equations, and all calculations above preserve the polynomial structure, we can use results from elimination theory to eliminate unknowns in each $c_i = \det(\mathbf{A}^i)$.

8.3 Toric elimination theory

We give a very short introduction to toric elimination theory, for more detailed information we refer to Cox et al. [1992]; Gelfand et al. [1994]; Sturmfels [2002]; Cox et al. [2005]; Emiris [2005]. More specifically, we present the sparse resultant from algebraic geometry Cox et al. [2005]; Emiris and Mourrain [1999] to eliminate the unknowns. Casually speaking, the resultant is a condition for an overdetermined system of polynomials to have a common root.

We consider a system of $n+1$ polynomials,

$$f_0 = \cdots = f_n = 0, \tag{8.18}$$

in n variables $\mathbf{x} = [x_1, \ldots x_n]^\mathbf{T}$, and let $\mathbb{C}^*$ denote the complex numbers without zero, $\mathbb{C}^* = \mathbb{C} \setminus 0$. Toric elimination theory considers solutions of the polynomials (8.18) in $(\mathbb{C}^*)^n$. Since none of the variables is allowed to be zero, the theory is valid for Laurent polynomials in $\mathbb{R}[\mathbf{x}, \mathbf{x}^{-1}, \mathbf{u}, \mathbf{u}^{-1}]$, that is, polynomials with positive and negative integer exponents.

Definition 11 (Monomial)**.** *We define a monomial* $\mathbf{x}^a$ *as the power product* $\mathbf{x}^a = x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$*, where* $(a_1, a_2, \ldots, a_n) \in \mathbb{Z}^n$*.*

Definition 12 (Support)**.** *Let the support* $\mathscr{E}_i = \{a_{i,1}, \ldots, a_{i,m_i}\}$ *denote the set of exponent vectors corresponding to monomials in*

$$f_i = \sum_{j=1}^{m_i} c_{i,j} \mathbf{x}^{a_{i,j}}, \quad c_{i,j} \neq 0. \tag{8.19}$$

We denote as $Q_i = conv(\mathscr{E}_i)$ the convex hull of the support of the polynomial f_i.

Definition 13 (Affine variety)**.** *Consider* $f_1, \ldots, f_m$ *polynomials in* $\mathbb{C}[x_1, \ldots x_n]$*. The affine variety* $V(f_1, .., f_m)$ *is defined by the set*

$$V(f_1, .., f_m) = \{(x_1, .., x_n) \in \mathbb{C}^s : f_i(x_1, .., x_n) = 0, i = 1 \ldots m\}. \tag{8.20}$$

Definition 14 (Zariski closure)**.** *Given a subset* $S \subset \mathbb{C}^m$*, the smallest affine variety containing* S *is called the Zariski closure of* S *and is denoted as* $\overline{S}$*.*

Let $L(\mathscr{E}_i)$ be the set of all polynomials that have exponents in the support $\mathscr{E}_i$

$$L(\mathscr{E}_i) = \left\{ c_{i,1} \mathbf{x}^{a_{i,1}} + \cdots + c_{i,m_i} \mathbf{x}^{a_{i,m_i}} : \quad c_{i,j} \in \mathbb{C}^* \right\}, \tag{8.21}$$

Then the coefficients of a polynomial define a point in $\mathbb{C}^{m_i}$. Now let

$$Z(\mathscr{E}_0, \ldots, \mathscr{E}_n) \subset L(\mathscr{E}_0) \times \cdots \times L(\mathscr{E}_n) \tag{8.22}$$

be the Zariski closure of the set of all $(f_0, \ldots f_n)$, for which (8.19) has a solution in $(\mathbb{C}^*)^n$. For an overdetermined system of polynomials we then have this result.

Theorem 12 (Sparse resultant,[Gelfand et al., 1994; Cox et al., 2005])**.** *Assume that* $Q_i = \mathrm{conv}(\mathscr{E}_i)$ *is a* n *dimensional polytope for* $i = 0, \ldots, n$*. Then there is an irreducible polynomial* $\mathscr{R}$ *in the coefficients of the* f_i *such that*

$$(f_0, .., f_n) \in Z(\mathscr{E}_0, \ldots, \mathscr{E}_n) \Leftrightarrow \mathscr{R}(f_0, .. f_n) = 0. \tag{8.23}$$

In particular, if the system

$$f_0 = f_1 = \cdots = f_n \tag{8.24}$$

has a solution $(x_1,\ldots,x_n) \in (\mathbb{C}^*)^n$*, then*

$$\mathscr{R}(f_1,\ldots,f_n) = 0. \tag{8.25}$$

We call $\mathscr{R}$ *the sparse resultant.*

Remark 14. *There exist more general versions of Theorem 12, which do not require the convex hull of the supports to be n-dimensional. [Sturmfels, 1994]. However, for simplicity we chose to present this simplified version here.*

Example 5 (One variable)**.** *Consider the system of two polynomials*

$$\begin{aligned} f_0 &= a_{11} + a_{12}x \\ f_1 &= a_{21} + a_{22}x + a_{2,3}x^2. \end{aligned} \tag{8.26}$$

The supports of this system are $\mathscr{E}_0 = \{(0),(1)\}$*, and* $\mathscr{E}_1 = \{(0),(1),(2)\}$*. Clearly, the convex hulls of the supports are the line segments* $[01]$ *and* $[0,2]$*, which have dimension* $n = 1$*. For generic coefficients* $a_{i,j}$ (8.26) *is inconsistent and does not have a solution in* $\mathbb{C}^*$*. The sparse resultant for this system is calculated as the determinant of the Sylvester matrix*

$$\begin{aligned} \mathscr{R}(f_0,f_1) &= \det\left(\begin{bmatrix} a_{12} & a_{11} & 0 \\ 0 & a_{12} & a_{11} \\ a_{23} & a_{22} & a_{21} \end{bmatrix}\right) \\ &= a_{12}^2 a_{21} - a_{12}a_{11}a_{22} + a_{23}a_{11}^2. \end{aligned} \tag{8.27}$$

Note that we have eliminated x from (8.26)*, and the statement* $\mathscr{R}(f_0,f_1) = 0$ *is identical to stating that there exist some x such that* $f_0 = f_1 = 0$*.*

The calculation of the sparse resultant for multivariate polynomials is more involved. An algorithm is given in Canny and Emiris [2000]. In this work, we use the free software `multires` [Busé and Mourrain, 2003] for the computations. We will use Theorem 12 to eliminate unknown variables from the control invariant from equation (8.16).

8.4 Using resultants in optimal control

After introducing the sparse resultant, we can apply it to our optimal control problem. Recall that generally the invariants $c_i = \det(\mathbf{A}^i)$ contain unknown variables. We collect all unknown (unmeasured) variables in a vector $\mathbf{d}$, so we have $c_i = c_i(\mathbf{d})$, and we write the model equations in the form

$$m(\mathbf{d}) = 0, \tag{8.28}$$

where we have omitted to explicitly state the dependency on the known variables.

Assumption 2. *The model equations are polynomials in the polynomial ring* $\mathbb{R}[\mathbf{d}]$

Assumption 3. *The variety* $V(\mathbf{m}(\mathbf{d}))$ *is zero dimensional, that implies that* $\mathbf{m}(\mathbf{d}) = 0$ *has a finite number of solutions for* $\mathbf{d}$*.*

Theorem 13 (Invariants for Control)**.** *If the number of unknown variables* n_d *is equal to the number of model equations* $n_{\mathbf{m}}$*, and Assumptions 2 and 3 hold, controlling*

$$\mathscr{R}(c_i, \mathbf{m}(\mathbf{d})) = 0 \tag{8.29}$$

is equivalent to controlling (8.16)*.*

Proof. By assumption, the model equations $\mathbf{m}(\mathbf{d}) = 0$ have a finite number of solutions. $c_i(\mathbf{d}) = \det(\mathbf{A}^i)$ is a polynomial in the variables $\mathbf{d}$ whose coefficients are functions of $\mathbf{u}$, and thus can be manipulated. Arbitrary input values $\mathbf{u}$ will cause that $c_i(\mathbf{d}) = 0$ does not have any solution. The sparse resultant $\mathscr{R}(c_i(\mathbf{d}), \mathbf{m}(\mathbf{d}))$ gives the necessary and sufficient condition for the combined system

$$\begin{aligned} \mathbf{m}(\mathbf{d}) &= 0 \\ c_i(\mathbf{d}) &= 0 \end{aligned} \tag{8.30}$$

to have a solution in $(\mathbb{C}^*)^{n_\mathbf{d}}$. By Theorem 12, we have

$$c_i = \det(\mathbf{A}^i) = 0 \Leftrightarrow \mathscr{R}(\mathbf{m}, c_i) = 0. \tag{8.31}$$

□

Thus, we may use model equations to eliminate unknown variables from $c_i = \det(\mathbf{A}^i)$, to obtain an expression which contains only known variables. More importantly, since the unknown variables are not contained in $\mathscr{R}(\mathbf{m}, c_i)$ we can control the resultant to zero by feedback control using online measurements.

Remark 15. *Note that it is not necessary to be able to solve the model equations* $\mathbf{m}(\mathbf{d}) = 0$ *uniquely for* $\mathbf{d}$*. The only condition is that the model equations have a finite number of solutions.*

Remark 16. *Since the unknown variables* $\mathbf{d}$ *assume real values in the process, the existence of complex solutions for* $\mathbf{m}(\mathbf{d}) = 0$ *does not matter, because the Theorem 12 states that* c_i *becomes zero whenever the resultant is zero.*

8.5 Case study: Fed batch reactor

The model and the optimization problem for this case study is taken from Gros et al. [2009]. Since controlling measured path constraints is straightforward, we present the unconstrained case where finding the controlled variable combination is more involved. Therefore we have selected the initial conditions as in Gros et al. [2009], such that we have only one unconstrained arc.

8.5.1 Model

We consider a fed batch reactor with two chemical reactions,

$$A + B \longrightarrow C \text{ and } 2B \longrightarrow D, \tag{8.32}$$

where C is the desired product and D is the undesired side product. The operational objective is to maximize difference between the amount of C and the amount of D at the final batch time t_f.

We use a simple dynamic model,

$$\begin{aligned} \dot{c}_A &= -k_1 c_A c_B - c_A u/V \\ \dot{c}_B &= -k_1 c_A c_B - 2k_2 c_B^2 - (c_B - c_B^{in})u/V \\ \dot{V} &= u, \end{aligned} \tag{8.33}$$

with the initial conditions: $c_A(0) = c_{A0}$, $c_B(0) = c_{B0}$, and $V(0) = V_0$. Initially the concentration of the products is zero, $c_{C0} = c_{D0} = 0$. All parameters and initial conditions are given in Table 8.1.

From the mass balance, we have ($c_{C0} = c_{D0} = 0$)

$$c_C(t) = \frac{1}{V}\left(c_{A0}V_0 - c_A(t)V\right) \tag{8.34}$$

and

$$c_D(t) = \frac{1}{2V}\left[\left(c_A + c_B^{in} - c_B\right)V - \left(c_{A0} + c_B^{in} - c_{B0}\right)V_0\right]. \tag{8.35}$$

8.5.2 Optimal control problem

The optimization problem is the formulated as

$$\min_{\mathbf{u}} J(t_f) \quad \text{s.t.} \quad \dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}, \tag{8.36}$$

where the objective is

$$J(t_f) = (c_D(t_f) - c_C(t_f))V(t_f). \tag{8.37}$$

Further, we have the state and input vectors $\mathbf{x} = [c_A, c_B, V]^{\mathbf{T}}$ and $\mathbf{u} = u$, and

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} -k_1 c_A c_B \\ -k_1 c_A c_B - 2k_2 c_B^2 \\ 0 \end{bmatrix}, \quad \mathbf{G}(\mathbf{x}) = \frac{1}{V}\begin{bmatrix} -c_A \\ c_B^{in} - c_B \\ V \end{bmatrix}. \tag{8.38}$$

The constraints for our system are

$$\begin{aligned} u &\leq u_{max} \\ u_{min} &\leq u. \end{aligned} \tag{8.39}$$

Table 8.1: Parameters and initial conditions

Symbol	Value	Unit	Description
k_1	0.053	l/mol/min	parameter
k_2	0.128	l/mol/min	”
c_B^{in}	5	mol/l	”
t_f	250	min	”
u_{min}	0	l/min	input constraint
u_{max}	0.001	l/min	”
c_{A0}	0.72	mol/l	initial condition
c_{B0}	0.0614	mol/l	”
c_{C0}	0.0	mol/l	”
c_{D0}	0.0	mol/l	”
V_0	1	l	”

8.5.3 Nominal optimal solution

Solving the system for the nominal conditions shows that none of the input constraints becomes active. The optimal trajectory is therefore a single interior arc. The Hamiltonian is

$$\begin{aligned} H = {} & \lambda_1 \left(-k_1 c_A c_B - \frac{c_A u}{V} \right) \\ & + \lambda_2 \left(-k_1 c_A c_B - 2k_2 c_B^2 + \frac{(c_B^{in} - c_B)u}{V} \right) \\ & + \lambda_3 u. \end{aligned} \tag{8.40}$$

Proceeding as in Section 8.2 we get $H_u = \lambda^{\mathbf{T}} A_0 = 0$ with

$$A_0 = \begin{bmatrix} \frac{-c_A}{V} & \frac{c_B^{in} - c_B}{V} & 1 \end{bmatrix}^{\mathbf{T}}. \tag{8.41}$$

We continue with the first and second time derivatives $\lambda^{\mathbf{T}} A_1 = 0$ and $\lambda^{\mathbf{T}} A_2 = 0$. Here, $A_1 = [a_{11}\, a_{12}, a_{13}]^{\mathbf{T}}$ with the elements

$$\begin{aligned} a_{11} &= \frac{1}{V}[-k_1 c_A (c_B - c_B^{in})] \\ a_{12} &= \frac{-1}{V}\left[k_1 c_A (c_B - c_B^{in}) + 2k_2 c_B (c_B - 2c_B^{in})\right] \\ a_{13} &= 0, \end{aligned} \tag{8.42}$$

Table 8.2: Simulation cases

	Case 1	Case 2
Unmeasured state:	c_A	c_B
Unmeasured disturbance:	none	k_1

and $A_2 = [a_{21}, a_{22}, a_{23}]^{\mathbf{T}}$, with

$$\begin{aligned} a_{21} &= \tfrac{1}{V^2}\left[c_B^{in} k_1 c_A V(k_1 c_A + 4k_2 c_B) + 2k_1 c_A \left(c_B - c_B^{in}\right) u\right] \\ a_{22} &= \frac{1}{V^2}\left[c_B^{in} V \left(4k_1 c_A k_2 c_B + 8k_2^2 c_B^2 + k_1^2 c_A^2\right)\right. \\ &\quad \left. +2\left(c_B - c_B^{in}\right)\left(k_1 c_A + 2k_2(c_B - c_B^{in})u\right)\right] \\ a_{23} &= 0. \end{aligned} \tag{8.43}$$

The determinant of $\mathbf{A} = [A_0, A_1, A_2]$ becomes zero when

$$c = 0, \tag{8.44}$$

with

$$c = 4k_2 c_B^2 c_B^{in} V + 2c_B c_B^{in} u - k_1 c_A c_B^2 V + 2k_1 c_A c_B^{in} V c_B - 2u(c_B^{in})^2. \tag{8.45}$$

In optimal control literature [Bryson and Ho, 1975], this expression is solved for u, and the input is implemented in the process. However, if there is unmodelled uncertainty or dead time in the process, this might lead to instability. Moreover, the coefficient of u might be very small (or large) resulting in numerically ill conditioned solution for u. In other cases, a relationship is found which does not contain u at all, and (8.45) cannot be solved for u. Therefore, we will not solve for the input, but rather use feedback-controller to control c to zero.

8.5.4 Eliminating unknown variables

We consider two cases, where different variables are considered unknown and have to be eliminated, Table 8.2.

Case 1: Unknown variables in algebraic equations

Let us first assume that the concentration of component A is very difficult or expensive to measure, so in this case we have one unmeasured state, namely c_A. All other variables in c from (8.45) are known. However, the unmeasured state is present in the algebraic relationship (8.34). From this we obtain the measurement polynomial

$$m_1 = Vc_C(t) - (c_{A0}V_0 - c_A(t)V) = 0, \tag{8.46}$$

and we calculate

$$\begin{aligned}\mathscr{R}(c,m_1) = &- Vc_B^2k_1c_C + 2Vc_Bk_1c_B^{in}c_C - 4Vc_B^2k2c_B^{in} - 2c_B^{in}uc_B \\ &+ 2{c_B^{in}}^2u - 2c_Bk_1c_B^{in}c_{A0}V_0 + c_B^2k_1c_{A0}V_0.\end{aligned} \tag{8.47}$$

$\mathscr{R}(c,m_1)$ does not contain the unmeasured state, and controlling it to zero will by Theorem 13 result in optimal operation.

Case 2: Unknown variables come from the differential equations

Now we assume that we have an unknown disturbance k_1, and that the concentration c_B is unmeasured. Since the reaction rate only enters through a differential equation, we need to eliminate k_1 from c (8.45) using a differential equation, and we need to use a change rate as a measured variable, too. However, if we have a good measurement of a process variable and a good clock, then estimating the time derivative is not difficult.

We assume that we can measure the concentration c_A together with an estimate of its time derivative, $\dot{c}_A$. If the measurement of c_A is good (little or no noise), then we may use its past values to estimate its time derivative. We assume that we measure $\dot{c}_A$ continuously using finite differences,

$$\dot{c}_A = \frac{c_A(t) - c_A(t - 1\text{min})}{1\text{min}}. \tag{8.48}$$

This does not give the exact derivative, but the approximation is considered good enough for our purposes.

To eliminate the unknowns c_B and k_1 we use an additional mass balance for component B,

$$m_2 = -c_BV + c_{B0}V_0 + c_B^{in}(V - V_0) - c_CV - 2c_DV = 0, \tag{8.49}$$

in combination with the implicit component balance for c_B from (8.33),

$$m_3 = -\dot{c}_BV - k_1c_Ac_BV - 2k_2c_B^2V - (c_B - c_B^{in})u = 0, \tag{8.50}$$

and we eliminate the unknowns by calculating the resultant with respect to the unknown variables k_1 and c_B:

$$
\begin{aligned}
&\mathscr{R}(c,m_2,m_3) = \\
&-16V^2c_Ak_2c_B^{in}c_D + V^2c_A\dot{c}_A + 4V^2k_2c_B^{in}{c_A}^2 + 8V^2c_Ak_2{c_B^{in}}^2 \\
&+Vu{c_A}^2 - 16Vc_Dk_2c_B^{in}V_0c_{B0} + 16Vc_Dk_2{c_B^{in}}^2V_0 \\
&+8VV_0c_{B0}k_2{c_B^{in}}^2 - 8Vc_Ak_2{c_B^{in}}^2V_0 + 16Vc_{A0}V_0k_2c_B^{in}c_D \\
&+8Vc_Ak_2c_B^{in}V_0c_{B0} - 8Vc_Ac_{A0}V_0k_2c_B^{in} - 8Vc_{A0}V_0k_2{c_B^{in}}^2 \\
&-8V_0^2c_{B0}k_2{c_B^{in}}^2 + 4V_0^2c_{B0}^2k_2c_B^{in} + 2V_0c_{B0}c_B^{in}u - c_AV_0c_B^{in}u \\
&+c_AV_0c_{B0}u + 16V^2c_D^2k_2c_B^{in} - 16V^2c_Dk_2{c_B^{in}}^2 - 2Vc_Ac_Du \\
&+Vc_Ac_B^{in}u - 4Vc_Dc_B^{in}u - 8V{c_B^{in}}^3k_2V_0 - VV_0c_B^{in}\dot{c}_A + VV_0c_{B0}\dot{c}_A \\
&-Vc_{A0}V_0\dot{c}_A - c_Ac_{A0}V_0u - 2c_{A0}V_0c_B^{in}u + 8c_{A0}V_0^2k_2{c_B^{in}}^2 \\
&+4c_{A0}^2V_0^2k_2c_B^{in} + 4V_0^2{c_B^{in}}^3k_2 - 2V_0{c_B^{in}}^2u + 4V^2{c_B^{in}}^3k_2 \\
&-V^2c_B^{in}\dot{c}_A - 2V^2c_D\dot{c}_A - 8c_{A0}V_0^2k_2c_B^{in}c_{B0}.
\end{aligned}
\tag{8.51}
$$

This expression does not contain any of the unknown variables, so it can be evaluated online and controlled to zero using a P or PI controller.

8.5.5 Simulation Results

Nominal operation

The state and input trajectories for nominal optimal operation are given in Figure 8.1. These trajectories are generated by applying the optimal input. The final optimal cost is value is $J = 0.2717$.

Controlling the invariant

Case 1.: Variable c_A unmeasured - all other variables known Here we cannot control $c = \det(A)$ to zero, because we cannot evaluate it since c_A is not known. Instead we control the resultant $\mathscr{R}(c,m_1)$ (8.47) to zero using a P controller. The trajectories appear identical to the optimal ones from the previous section, and the objective value is $J = 0.2717$. This is exactly as expected, because by Theorem 13, controlling c and $\mathscr{R}(c,m_1)$ are equivalent. The suboptimality, which is introduced by the added P controller, does not become visible when considering the first seven digits of the objective function. However, whereas we need to know the value of c_A to control c, this is not necessary for controlling $\mathscr{R}(c,m_1)$ to zero.

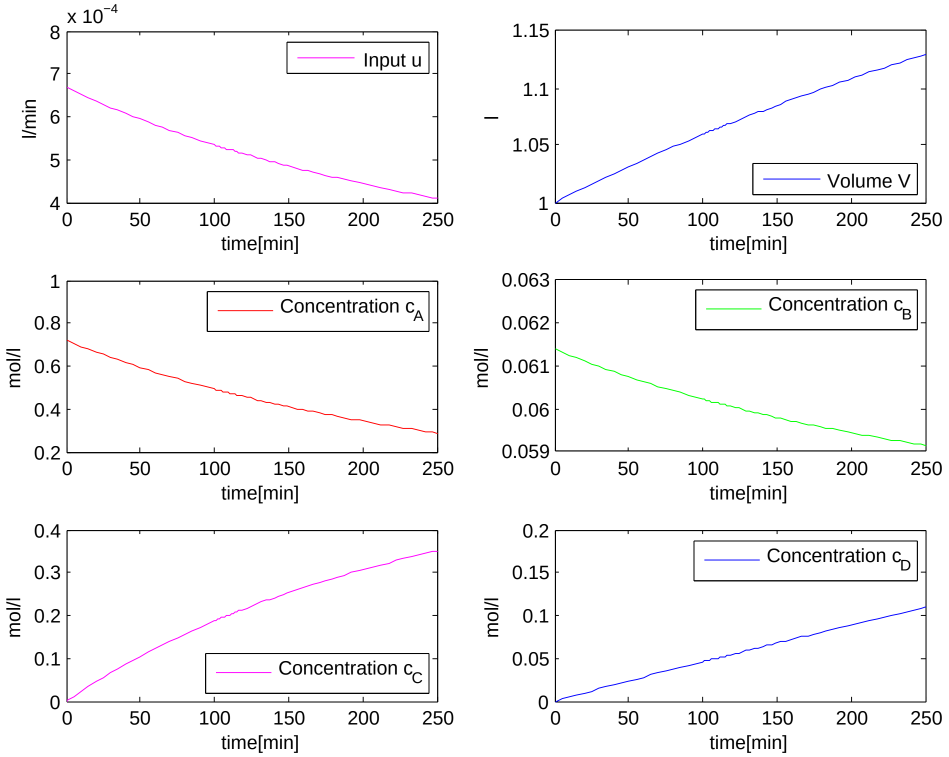


Figure 8.1: Nominal optimal input, volume, and concentration trajectories

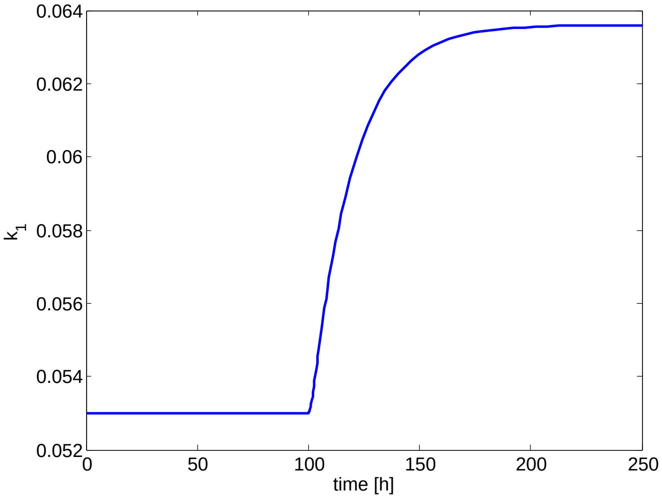


Figure 8.2: Disturbance k_1

Case 2.: Variables k_1, c_B unmeasured – $\dot{c}_A$ and c_A measured In this case, the state c_B is not known (measured) and the parameter k_1 is not known either. Therefore we cannot evaluate c and use it for control. Instead we control $\mathscr{R}(c, m_2, m_3)$, which contains neither k_1 nor c_B. This expression can be evaluated using the available measurements and controlled to zero. In the nominal case the trajectories look identically the same as in Figure 8.1.

Next, we consider a change in the reaction kinetics, where k_1 rises 20%, Figure 8.2. The input and the states are given in are given in Figure 8.3 The final profit when controlling $\mathscr{R}(c, m_2, m_3)$ to zero is $J = 0.2970$, while the profit using the optimal input is $J_{opt} = 0.2971$. This difference comes from the approximation of $\dot{c}_A$ in (8.48). Using the exact derivative, we obtain $J = 0.2971$, which is the same value as the optimal input gives.

If we had not eliminated k_1 in $\mathscr{R}(c, m_2, m_3)$, and just used the nominal value, the objective value would be lower, $J = 0.2873$.

8.6 Discussion and conclusion

We have shown that the concepts of finding invariant variable combinations can be extended to dynamic systems, which are described by polynomial or rational equations. By not explicitly solving for the input **u**, we do not have to be concerned whether the input appears in **c**, because we use a P or PI controller to generate the optimal inputs. This is a simple alternative to analytically finding the optimal input

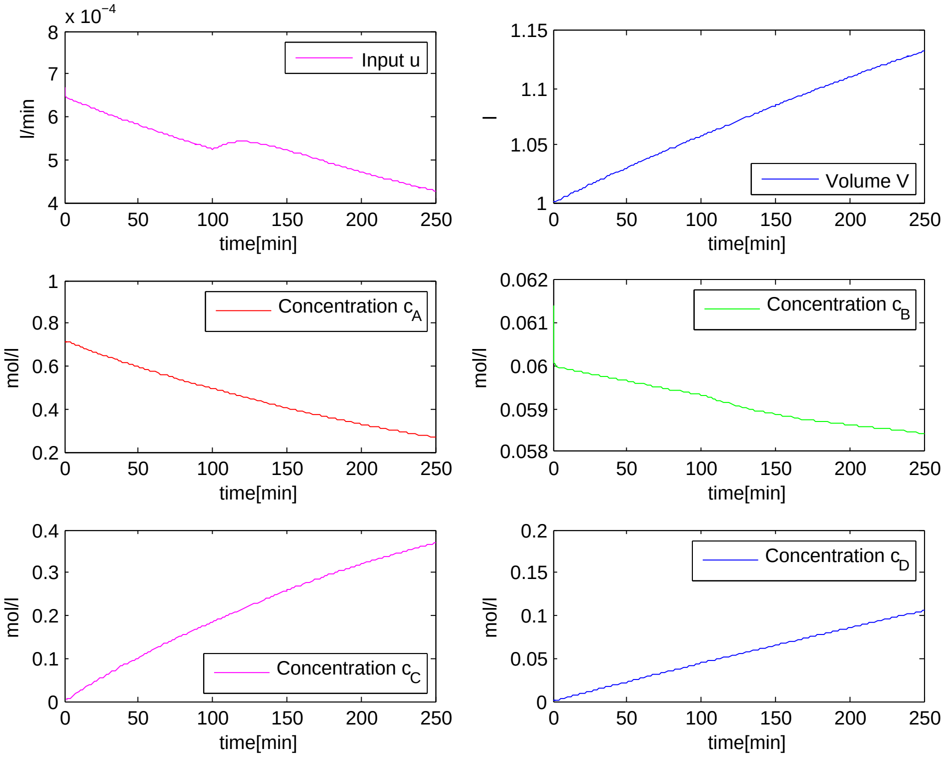


Figure 8.3: Inputs and concentrations for unmeasured change of k_1 at time 100

by further differentiations.

Adding a controller to control **c** will often come at a negligible loss. This is confirmed in our example, where controlling the invariants using only a P controller gives virtually the same performance as when analytically solving for the optimal input.

In this work, we considered only parametric uncertainties and unmeasured states. The equally important issues as model error and measurement noise are beyond the scope of this work and are have to be studied in future work.

We have assumed that the uncertainty does not change the active constraints. This is valid for sufficiently small disturbances. However, for larger disturbances, the new set of active constraints has to be determined.

Controlling the invariant can be used together with other NCO tracking methods to handle model mismatch (by adjusting model parameters) or terminal constraints on a run-to-run basis, similar to Srinivasan and Bonvin [2007]. Thus, our method can be considered as a part of a larger procedure for implementing optimal batch performance.

Beside “normal measurements” we have also allowed measurements of their time derivatives. They may be estimated by finite differences as above, or by using some filtering. If a measurement is assumed to be reliable, then its change over time should also be possible to estimate reasonably well. Introducing measurements of the time derivatives, makes it possible to eliminate variables, for which we do not have a purely algebraic expression, and which enter through the differential equations only.

In the procedure for eliminating the adjoint variables, we have presented the common case of input affine systems. If the model is not input affine, elimination of the adjoint variables comes at the cost of introducing time derivatives of the input, which have to be measured.

We used the resultant to eliminate the unknown variables. Other techniques, such as Gröbner bases [Cox et al., 1992], could also have been applied. However, it is not easy to find appropriate monomial orderings which eliminate the unknown variables, while avoiding the trivial solution (the invariant is always zero when the model equations are satisfied).

References

Allgöwer, F., Zheng, A. (Eds.), 2000. Nonlinear Model Predictive control. Birkhäuser Verlag Basel.

Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. Industrial & Engineering Chemistry Research 46, 846–853.

Bryson, A. E., Ho, Y.-C., 1975. Applied Optimal Control, revised edition. Taylor & Francis.

Busé, L., Mourrain, B., 2003. Using the maple multires package.
URL http://www-sop.inria.fr/galaad/software/multires

Canny, J. F., Emiris, I. Z., 2000. A subdivision-based algorithm for the sparse resultant. Journal of the ACM.

Cox, D., Little, J., O'Shea, D., 1992. Ideals, Varieties, and Algorithms. Springer-Verlag.

Cox, D., Little, J., O'Shea, D., 2005. Using Algebraic Geometry. Springer.

Diehl, M., Bock, H. G., Schlöder, J. P., Findeisen, R., Nagy, Z., Allgöwer, F., 2002. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. Journal of Process Control 12 (4), 577 – 585.

Emiris, I. Z., 2005. Toric resultants and applications to geometric modelling. In: Bronstein, M., Cohen, A. M., Cohen, H., Eisenbud, D., Sturmfels, B., Dickenstein, A., Emiris, I. Z. (Eds.), Solving Polynomial Equations. Vol. 14 of Algorithms and Computation in Mathematics. Springer Berlin Heidelberg, pp. 269–300.

Emiris, I. Z., Mourrain, B., 1999. Matrices in elimination theory,. Journal of Symbolic Computation 28 (1-2), 3–43.

Geering, H. P., 2007. Optimal Control with Engineering Applications. Springer.

Gelfand, I. M., Kapranov, M. M., Zelevinsky, A. V., 1994. Discriminants, Resultants and Multidimensional Determinants. Birkhäuser, Boston, MA.

Gros, S., Srinivasan, B., Chachuat, B., Bonvin, D., 2009. Neighbouring-extremal control for singular dynamic optimisation problems. part i: single-input systems. International Journal of Control 82 (6).

Grötschel, M., Krumke, S. O., Rambau, J. (Eds.), Sep. 2001. Online Optimization of Large Scale Systems. Springer.

Kühl, P., Diehl, M., Kraus, T., Schlöder, J. P., Bock, H. G., 2011. A real-time algorithm for moving horizon state and parameter estimation. Computers & Chemical Engineering 35 (1), 71 – 83.

Marquez, H. J., 2003. Nonlinear Control Systems. John Wiley & Sons.

Schlegel, M., Stockmann, K., Binder, T., Marquardt, W., 2005. Dynamic optimization using adaptive control vector parameterization. Computers & Chemical Engineering 29 (8), 1731 – 1751.

Skogestad, S., 2000. Plantwide control: The search for the self-optimizing control structure. Journal of Process Control 10, 487–507.

Srinivasan, B., Bonvin, D., 2007. Real-time optimization of batch processes by tracking the necessary conditions of optimality. Industrial & Engineering Chemistry Research 46 (2), 492–504.

Srinivasan, B., Palanki, S., Bonvin, D., 2003. Dynamic optimization of batch processes: I. characterization of the nominal solution. Computers & Chemical Engineering 27 (1), 1 – 26.

Sturmfels, B., 1994. On the newton polytope of the resultant. Journal of Algebraic Combinatorics 3, 207–236.

Sturmfels, B., 2002. Solving systems of polynomial equations.

Terwiesch, P., Agarwal, M., Rippin, D. W. T., 1994. Batch unit optimization with imperfect modelling: a survey. Journal of Process Control 4 (4), 238 – 258.

Chapter 9

Conclusions and future work

> Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.
>
> W. Churchill

9.1 Conclusions

In the first part of this thesis, existing methods from self-optimizing control have been considered and studied under an optimization point of view. A connection between the methods used in self-optimizing control and the gradient of the optimization problem was found. In this context, it was discovered, that self-optimizing control can be interpreted as finding the best possible approximation to the gradient of the process. In line with this interpretation, we found a simple derivation of the previously published null-space method [Alstad and Skogestad, 2007]. It was found that this derivation could be extended to systems of polynomial equations as was presented in this thesis.

For cases where no model is available, we have presented a method for finding invariant combinations which is based on using optimal data. The suggested approach can be used to find directions in the measurements, which indicate a good operation policy. This analysis tool may applied for finding the “secret” of good plant operators.

In Chapter 5, we studied the use of self-optimizing control and NCO tracking in the context of real-time optimization. The results from this work indicate that self-optimizing controlled variables should be used in the control layer, while the iterative NCO tracking procedure should be used in the optimization layer. Here, the self-optimizing variables are used to align the control objectives and the economic objectives in a hierarchical control structure which is based on time scale separation.

In the second part of this thesis, the foundations have been laid for extending the self-optimizing control concept for polynomial systems. The concept of null-space method has been extended to polynomial systems, and even to dynamic systems, which are described by polynomial equations. These results are very interesting from a theoretical point of view, because they demonstrate that the concept of finding optimal invariant variable combinations is not restricted to the linear-quadratic case. However, the results are "exact" in a rather mathematical sense, that is they may not work well in case of plant-model mismatch and measurement noise. But this will be very dependent on the structure of the model equations, and it may well be that there are cases, which give robust and simple invariants, which are suitable for use in industrial practice.

9.2 Further work

9.2.1 Robustify and simplify the polynomial approach

In this thesis the theoretical basis has been provided for extending the self-optimizing control concept for nonlinear (polynomial) systems. The results obtained are valid for a perfect model without measurement noise. It is difficult to make general statements about how this will impact the performance of the plant in case of model mismatch and measurement noise. Another partially related issue is that the polynomial invariants may become extremely long expressions of high degree variables. In practice, however, it is often more convenient to have simpler (fewer terms and low degree) robust expressions, because the model is not exactly known. Moreover, the computation of the resultant is computationally expensive, so the approach for finding polynomial invariants is in practice only applicable to small systems.

Based on the results of this work, future research could focus on systematically finding simpler, more robust invariants, which may generally introduce some loss in terms of the objective function, but which tolerate model mismatch and measurement noise better. A possible starting point could be to use optimization to find approximate invariants with a simple structure, which minimize the loss over the range of disturbances.

9.2.2 Further studying the polynomial approach

Since the results for polynomial systems are based on a perfect model without measurement noise, it does not matter which measurements are included in the invariant. In practice, there will be measurements, which are more reliable than others. So far, it has been left open which equations should be used for eliminating the unknowns.

The choice of the equations for eliminating the unknowns can have large impact on the size and the degree of the invariants. Therefore, future research may give indications on which equations to use for eliminating the unknowns.

9.2.3 Handling changing active constraints

Another field for future work is the consistent and tractable handling of changing active constraints. This is a very important topic, because the economic loss associated with changing active constraints is generally higher than the loss associated with the unconstrained degrees of freedom. In this work, we have used a pragmatic approach, where we monitored the controlled variables of the neighbouring regions. This is viable when the numbers of regions is small and the problems behave nicely. However, when there are many regions, it may become difficult to track all the controlled variables in the neighbouring regions. Some progress on this topic has been done in the explicit model predictive control community, but it seems that there are still many open issues when the optimization problem is not linear-quadratic. When considering model mismatch and measurement noise, handling changing active constraints becomes even more challenging, and a systematic, rigorous way of handling these issues has yet to be developed.

9.2.4 Further studies on the combination of real-time optimization and self-optimizing control

It has been shown that real-time optimization and self-optimizing control are complementary (Chapter 5). It would be worth studying the interaction of real-time optimization and self-optimizing control further. Topics of interest could be:

- Evaluating how the combination of real-time optimization and self-optimizing control compare to the truly optimal dynamic real-time optimization.
- Research possibilities for further improving the combination of self-optimizing control and real-time optimization.
- Which disturbances should be rejected in which layer.

A different direction of future work could be studying the role and effect of backoff and active constraint handling in the combined framework of self-optimizing control and real-time optimization. Deeper knowledge in these areas would allow us to further fine-tune the overall control structure for optimal interplay.

9.2.5 Disturbance discrimination

We have assumed that the important disturbances, which should be rejected optimally, are known a priori. This is, however, not generally the case in practice. It would be of great practical importance if we had theoretically sound methods for discriminating disturbances, such that we could focus on the important disturbances in modelling and control structure selection.

References

Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. Industrial & Engineering Chemistry Research 46, 846–853.

Appendix A

Optimal operation of a waste incineration plant for district heating

> In writing music, the structure of each piece is a very important factor.
>
> L. Ornstein

This paper demonstrates the concept of implementing a self-optimizing control structure on an industrial case study. Using off-line optimization, the structure of the optimization problem is exploited to find a set of variables, which give optimal operation when controlled at their setpoints. We first obtain a steady state plant model, which is optimized on grid points in the operating region in order to determine the set of active constraints and the optimally unconstrained variables together with their optimal values. The variables assuming a constant optimal value are candidates for self-optimizing variables. Four operational regions are found, and for each region a simple control structure is defined to 1.) satisfy constraints and 2.) to control the self-optimizing variables to their setpoints. To be able to change between different regions, switching rules are defined. Using these switching rules, the plant can be controlled close to optimality when a disturbance causes the system to change from one region to another. Finally some dynamic simulation results are presented to show the control performance within the regions and across region boundaries.

Based on the paper "Optimal Operation of a Waste Incineration Plant for District Heating" presented at the American Control Conference 2009, St. Louis

A.1 Introduction

Rising energy prices, increasing competition and environmental demands make it increasingly necessary to operate plants as close to optimality as possible. In order to remain close to optimality in spite of disturbances, there are two basic paradigms [Narasimhan and Skogestad, 2007]. The first paradigm is to obtain optimal operation via on-line optimization [Marlin and Hrymak, 1997]. This implies that the optimal setpoints of the controlled variables are computed on-line and are updated at certain time intervals based on the last available measurements. Setting up, solving and maintaining an such a real-time optimization (RTO) system can be a very time-consuming and complex task, as the uncertainty in the model and parameters can have a severe impact on the control performance, and the updated setpoints have to be available at the given sample times.

A second paradigm, which is very common in practice, is to identify appropriate “self-optimizing control” variables [Skogestad, 2000]. Controlling these variables at their set-points keeps the process at or close to the optimal operating point in presence of disturbances, without the need to re-optimize. Usually, such policies are obtained by intuition, experience, and technical insight. We present a case study of a waste incineration plant, where the control structure can be found by performing an off-line optimization and analysis.

In district heating networks, the network operators usually wish to obtain the lowest possible return temperature to the heat source, while the power plants are designed for providing a given amount of heat at a specified temperature range. In this case study, the power plant is not owned by the district heating provider, which can lead to conflicts as the district heating provider attempts to draw more energy than what is produced, thus cooling down the plant. We design a control structure which prevents the plant from being cooled down while minimizing the operating cost. The example illustrates nicely the principles and benefits of self-optimizing control.

The structure of the paper is as follows: First the fundamental ideas of self-optimizing control are presented, and then the waste incineration process is presented and explained together with the operating objectives. Next, the model is described and optimized. Based on inspection of the optimization results, a control structure is designed and tested on a dynamic model. After presenting and discussing representative results, the paper finishes with our conclusions.

A.2 Self-optimizing control

The idea behind self-optimizing control is defined in Skogestad [2000]:

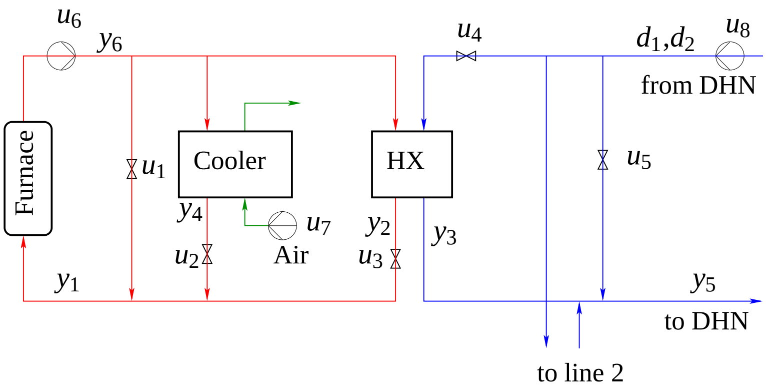


Figure A.1: Flowsheet of the incineration plant

> *Self-optimizing control is when we can achieve and acceptable loss with constant setpoint values for the controlled variables (without the need to reoptimize when disturbances occur)*

This means that for each region defined by the set of active constraints, we search for variables or variable combinations whose optimal values are constant in presence of disturbances. If they are controlled at their optimal values, which are the same for all disturbances within that region, we indirectly obtain optimal operation, without having to reoptimize for new setpoints.

A.3 The process

We consider a waste incineration plant with two production lines. The process flowsheet for one line is shown in Figure A.1. It is assumed that the lines are designed and operated symmetrically, such that it is sufficient to consider one line.

Cool water is flowing from the district heating network (DHN) and distributed equally onto the two production lines where is heated in the heat exchangers (HX) before it is returned to the network. Before the cold stream is split between the two plant lines, a bypass is installed to adjust the amount of water flowing through the heat exchangers.

In the two lines on the primary side, liquid water is heated in the furnaces to the desired temperature and transfers the heat to the secondary stream in the heat exchangers. The plant is equipped with an additional cooler, which is used when the DHN does not require all the produced heat. To prevent cooling down the plant, the heat exchangers can be bypassed on the primary side, too.

Table A.1: Measurements

Variable	Description
y_1	Return temperature to furnace
y_2	Primary side heat exchanger exit temperature
y_3	Secondary side heat exchanger exit temperature
y_4	Cooler exit temperature (liquid)
y_5	Secondary side return temperature (to DHN)
y_6	Primary side flow rate

Table A.2: Inputs

Variable	Description
u_1	Bypass valve opening
u_2	Cooler valve opening
u_3	Primary side heat exchanger valve opening
u_4	Secondary side heat echanger valve opening
u_5	Secondary side bypass valve opening
u_6	Primary side flow pump duty
u_7	Cooling fan duty
u_8	Secondary side flow pump duty

In this study, the plant operator is interested in operating the plant to provide 16 MW per line, while minimizing energy consumption for pumps and fans and still satisfying temperature and flow constraints. The available measurements and inputs are listed in Table A.1 and A.2.

The two lines are operated symmetrically and are subjected to operational constraints: The furnace entrance temperature is given as $y_1 = (126 \pm 1)^\circ$C, and should not be violated to avoid condensing of fume gasses and boiling in the pipes. The primary side flow rate is $y_6 = 250$ t/h, and the primary side heat exchanger exit temperature y_2 must not exceed 126°C. In addition, the return temperature to the district heating network y_5 must be in the interval from 90°C-150°C. The constraints are summarized in Table A.3.

Table A.3: Operational Constaints

Variable	Constraint	Description
y_1	$y_1 = 126 \pm 1°\text{C}$	Furnace return temperature
y_2	$y_2 \leq 126°\text{C}$	Heat exchanger exit temperature
y_5	$90 \leq y_5 \leq 150$	Return temperature to DHN
y_6	$y_6 = 250$ t/h	Primary side flow rate

A.4 Steady state plant model

The main modelling assumptions are: Symmetric lines, non-compressible fluids, no pressure drop in heat exchanger and pipes, no heat losses and ideal counter current heat exchangers. We give an overview over the most important aspects of the model. A more thorough description of the model and all its parameters and values is given in Smedsrud [2008].

A.4.1 Heat exchangers

The heat exchangers are modelled based on the logarithmic mean temperature difference. However, to make the numerical simulations more robust, the equations are transformed such that we obtain a linear relationship between the input and output temperatures [Skogestad, 2009]. The steady state transfer function from the inlet temperatures to the outlet temperatures is

$$\mathbf{T}_o = \mathbf{D}\mathbf{T}_i, \tag{A.1}$$

with

$$\mathbf{T}_i = \begin{bmatrix} T_{hot}^{in} \\ T_{cold}^{in} \end{bmatrix}, \quad \text{and} \quad \mathbf{T}_o = \begin{bmatrix} T_{hot}^{out} \\ T_{cold}^{out} \end{bmatrix} \tag{A.2}$$

as the vectors of inlet and outlet temperatures, respectively, and the dimensionless gain matrix

$$\mathbf{D} = \begin{bmatrix} \frac{\gamma(1-\beta)}{\gamma-\beta} & \frac{\beta(\gamma-1)}{\gamma-\beta} \\ \frac{\gamma-1}{\gamma-\beta} & \frac{1-\beta}{\gamma-\beta} \end{bmatrix}. \tag{A.3}$$

Using the mass flow rates w and the specific heat capacities c_p, the parameter β is defined as the ratio between cold and hot heat capacity flow rates,

$$\beta = w^c c_p^c / w^h c_p^h. \tag{A.4}$$

Further we define

$$\gamma = \exp(-\alpha), \tag{A.5}$$

Table A.4: Parameter values for heat transfer h

Unit	a_1(K^2/(W^{2}m^2)	a_2(K/(Wm^2)	a_3(/1(m^2)	a_4(W/(m^{2}K)
h_{HX}	$4 \cdot 10^{-4}$	-0.15	21.92	7615.8
h_{Cool}	0.42	-20.44	432.3	666.09

with α as

$$\alpha = UA\left[1/(w^h c_p^h) - 1/(w^c c_p^c)\right]. \tag{A.6}$$

Here, $U = h^h h^c/(h^h + h^c)$ is the overall heat transfer coefficient and A is the total heat transfer area. The variables h^c and h^h are assumed equal, and their flow dependency in heat exchanger and cooler is found from fitting the steady state model to the dynamic model used to test the results,

$$h_{unit} = a_1(w^c)^3 + a_2(w^c)^2 + a_3 w^c + a_4, \tag{A.7}$$

where the subscript *unit* stands for either the heat exchanger (HX) or the cooler (Cool). The parameter values are listed in Table A.4.

The matrix **D** becomes singular when $\beta = 1$ (parallel temperature profiles), which is the case for some operating conditions. In order be able to simulate also cases where $1 - \delta_{max} < \beta < 1 + \delta_{max}$ for small δ, we define the number of transfer units is η:

$$\eta = UA/(w^c c_p^c). \tag{A.8}$$

We expand the exponential term in (A.5) [Hertzberg, 2007] and define

$$S = \sum_{i=1}^{\infty} (-\eta\delta)^i/(i+1)!. \tag{A.9}$$

The series S is truncated after $i = 5$. Using (A.9), we write $\gamma = -\eta\delta(1-S)+1$, and for $1 - \delta_{max} < \beta < 1 + \delta_{max}$ we use

$$\mathbf{D} = \begin{bmatrix} \frac{\gamma}{1+\eta(1+S)} & \frac{\eta\beta(1+S)}{1+\eta(1+S)} \\ \frac{\eta(1+S}{1+\eta(1+S)} & \frac{1}{1+\eta(1+S)} \end{bmatrix} \tag{A.10}$$

instead of (A.3).

A.4.2 Pump, fan, valve and mixer modelling

The fan duty P is calculated by

$$P = \frac{1}{\eta}\left[\frac{w^3}{2\rho^2}\left(\frac{1}{A_2^2} - \frac{1}{A_1^2}\right) + \frac{w\Delta p}{\rho}\right], \tag{A.11}$$

where η, w ρ, Δp are the efficiency, flow rate, density and pressure difference, respectively, and A_1, A_2 denote cross sectional areas of the pipes before and after the fan, respectively.

Assuming equal pipe diameters before and after the pump and no elevation difference, the pump pressure outlet is calculated by

$$\Delta p = (\rho P \eta)/w. \tag{A.12}$$

The valves are modelled by

$$w = K_v \sqrt{(\rho_o/\rho)\Delta p}, \tag{A.13}$$

with K_v being constant on the primary side, and being a function of the valve opening in the secondary side, and ρ_0 is a reference density. The mixer equations are derived from the mass and energy balances:

$$w_{tot} = \sum_i w_i, \tag{A.14}$$

$$T_{out} = \sum_i (w_i/w_{tot}) T_i \tag{A.15}$$

A.5 Optimization

The optimization objective is to minimize the total work for the pumps and fans,

$$\min J = \sum w = u_6 + u_7 + u_8, \tag{A.16}$$

subject to the model equations and the operating constraints from Section A.3.

To obtain a map of the operating regions, the disturbance space is discretized in two disturbance variables, namely flow and temperature coming from the district heating network. The temperature grid ranges from 65°C to 90°C and has a resolution of 0.1°C. The flow disturbance from the district heating network ranges from 500 t/h to 900 t/h, and has a resolution of 1.6 t/h. The model is optimized for each of these grid points. Evaluating optimal operation for all grid points yields four regions, defined by active constraints. These regions are shown in Figure A.2. In Table A.5 all inputs are given with the optimal values for each region. The "*x*" in the table indicate that the corresponding variable does not assume constant value throughout the region. In most cases when an input assumes a constant value, it is at a constraint, i.e. 0% or 100%.

Table A.6 shows the optimal output values. The furnace return temperature, y_1, and the primary side flow rate, y_6, are always constant because they are operational constraints (see Section A.3). Otherwise, only the primary side heat exchanger and cooler temperatures y_2 and y_4 reach the constraint of 126 °C in Region δ. All other measurements are unconstrained and varying in the whole region.

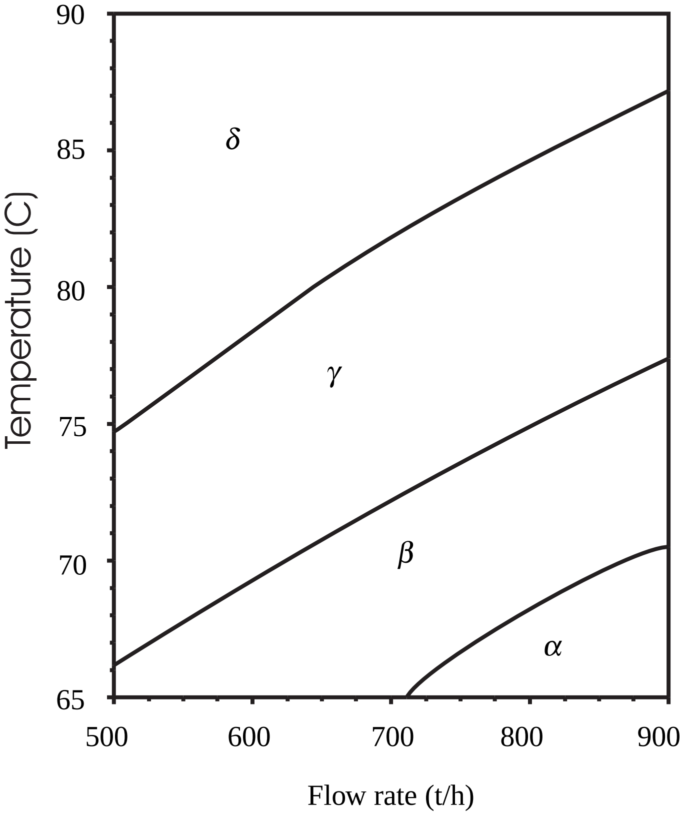


Figure A.2: Operating regions of the waste incineration plant

Table A.5: Optimal input values (active constraints in bold)

Region	u_1 [%]	u_2 [%]	u_3 [%]	u_4 [%]	u_5 [%]	u_6 [%]	u_7 [%]	u_8[a] [%]
α	92.6	**0**	**100**	*x*	**100**	6.78	**0**	*x*
β	92.6	**0**	**100**	**100**	*x*	6.78	**0**	*x*
γ	*x*	**0**	**100**	**100**	**0**	*x*	**0**	*x*
δ	**0**	*x*	**100**	**100**	**0**	*x*	*x*	*x*

[a] u_8 is not actually an input as it is used to set the disturbance flow rate

Table A.6: Optimal output values (active constraints in bold)

Region	y_1	y_2	y_3	y_4	y_5	y_6
	°C	°C	°C	°C	°C	t/h
α	**126**	x	x	x	x	**250**
β	**126**	x	x	x	x	**250**
γ	**126**	x	x	x	x	**250**
δ	**126**	**126**	x	**126**	x	**250**

A.6 Control structure design

In each region, the degrees of freedom (DOF) available for optimization N_{opt}^{free} are determined according to Skogestad and Postlethwaite [2005],

$$N_{opt}^{free} = N_m - N_0 - N_{active}\,, \tag{A.17}$$

where N_m is the number of control degrees of freedom, N_0 is the degrees of freedom without steady state effect, and N_{active} is the number of active constraints.

In our case, we have $N_m = 14$ control degrees of freedom, $N_0 = 0$ degrees of freedom without steady state effect, and the number of active constraints, N_{active}, varies in the different regions.

If the number of DOF is zero, all inputs are used to satisfy the constraints, and no self-optimizing variable is required, as the optimum is at a constraint. When the number of DOF is larger than zero we have a number of inputs which we do not need to satisfy a constraint, and we may use these inputs to minimize the operating cost. This is done by controlling a variable, which has an optimally invariant value, a self-optimizing control variable.

In Tables A.5 and A.6 the active input and output constraints are shown in bold face for each region. All inputs except u_5 are present in both lines, so when calculating the DOF free for optimization, this has to be taken into account when designing the control structure.

- In region α,where the bypass u_5 is fully open, we have

$$N_{opt}^{free,\alpha} = 14 \underbrace{-2}_{u_2} \underbrace{-2}_{u_3} \underbrace{-1}_{u_5} \underbrace{-2}_{u_7} \underbrace{-1}_{u_8} \underbrace{-2}_{y_1} \underbrace{-2}_{y_6} = 2.$$

- In region β,where u_4 is fully open, we have

$$N_{opt}^{free,\beta} = 14 \underbrace{-2}_{u_2} \underbrace{-2}_{u_3} \underbrace{-2}_{u_4} \underbrace{-2}_{u_7} \underbrace{-1}_{u_8} \underbrace{-2}_{y_1} \underbrace{-2}_{y_6} = 1.$$

- In region γ,where u_4 is fully open and the bypass u_5 is fully closed, we have

$$N_{opt}^{free,\gamma} = 14 \underbrace{-2}_{u_2} \underbrace{-2}_{u_3} \underbrace{-2}_{u_4} \underbrace{-1}_{u_5} \underbrace{-2}_{u_7} \underbrace{-1}_{u_8} \underbrace{-2}_{y_1} \underbrace{-2}_{y_6} = 0.$$

- In region δ where $u_1 = u_5 = 0$ and where u_3 and u_4 are fully open, we have

$$N_{opt}^{free,\delta} = 14 \underbrace{-2}_{u_1} \underbrace{-2}_{u_3} \underbrace{-2}_{u_4} \underbrace{-1}_{u_5} \underbrace{-1}_{u_8} \underbrace{-2}_{y_1} \underbrace{-2}_{y_4} \underbrace{-2}_{y_6} = 0.$$

Thus, in region α and β, we have unconstrained degrees of freedom for which we have to find self-optimizing control variables, while in regions γ and δ all degrees of freedom are used to satisfy the active constraints.

A.6.1 Region α and β

In region α, the inputs u_1, u_4 and u_6 are unconstrained, while we have two active constraints, $y_1 = 126°\text{C}$ and $y_6 = 250$ t/h. In region β, the inputs u_1, u_5 and u_6 are unconstrained, while we have two active constraints, $y_1 = 126°\text{C}$ and $y_6 = 250$ t/h.

In both, regions, the input u_1 is unconstrained. Since its optimal value is constant, we use it as a self-optimizing controlled variable. The remaining two DOFs have to be used to satisfy the constraints on the furnace return temperature y_1 and the flow rate y_6.

One possibility would be to control the furnace return temperature, y_1, using the secondary side heat exchanger valve u_4 in region α (and u_5 in region β), while keeping the bypass valve u_1 at a constant opening. However, this approach is not desirable from a dynamic point of view, because of the long time lag between the secondary side valves and the furnace inlet temperature y_1.

Therefore, in region α it is chosen to employ an input resetting structure, which utilizes the direct effect of bypass u_1 to control the furnace inlet temperature y_1, while the secondary side heat exchanger valve u_4 is used to reset the primary side bypass valve u_1 to the optimal value (Figure A.3).

In region β the bypass valve assumes the same constant value as in region α and is used as a self-optimizing variable as well. However, here the secondary side heat exchanger valve u_4 is in saturation, while u_5 may be use instead to reset the primary side bypass valve u_1 to its optimal value (Figure A.4).

A.6.2 Region γ

Region γ does not have an unconstrained degree of freedom for optimization. This means that the system is operated optimally when all the active constraints are satisfied. After using u_6 to control the flow rate in the primary side, the bypass valve u_1 is used to control the furnace return temperature y_1, Figure A.5.

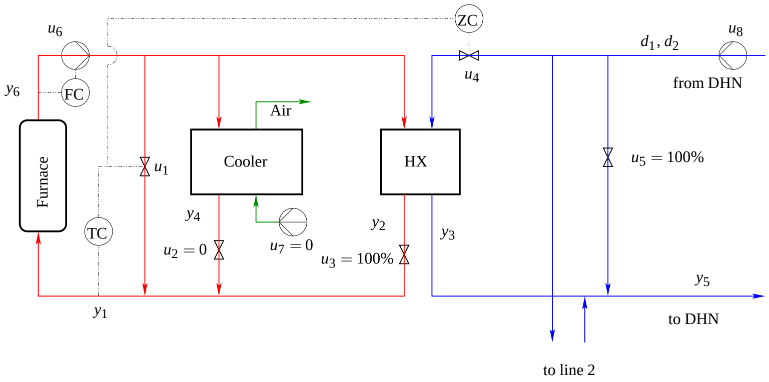


Figure A.3: Control structure for region α. (ZC: valve position controller for input u1)

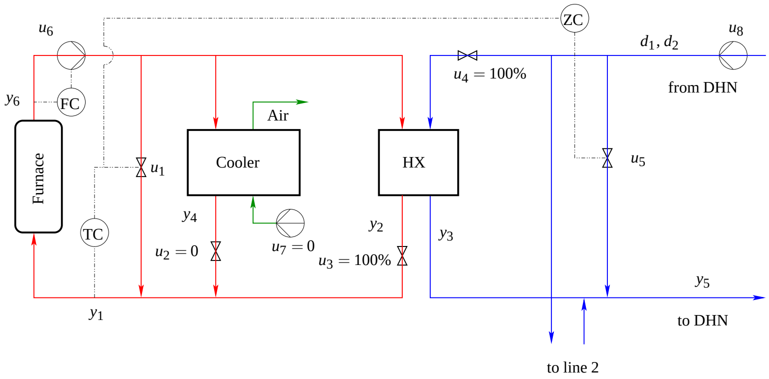


Figure A.4: Control structure for region β. (ZC: valve position controller for input u1)

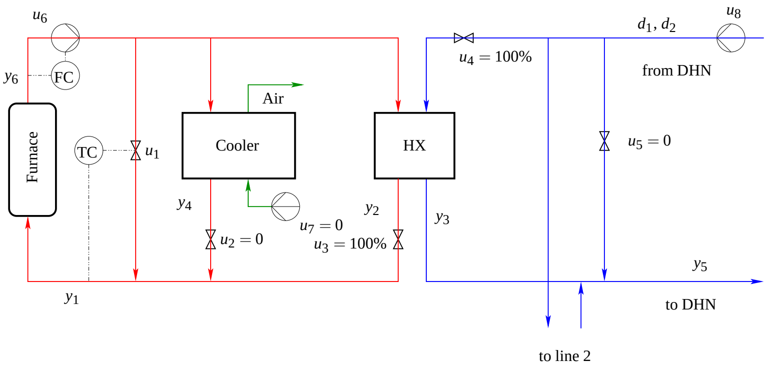


Figure A.5: Control structure for region γ

A.6.3 Region δ

In region δ, the bypasses u_1 and u_5 are closed, while the heat exchanger valves u_3 and u_4 are fully opened. After using u_6 to set the primary flow rate, the region has two unconstrained inputs, the cooler valve u_2 and the cooling fan duty u_7; they are needed to control the active constraints, the furnace return temperature y_1 and the heat exchanger exit temperature y_2 to their setpoints at 126°C. This means that all three temperatures become $y_1 = y_2 = y_4 = 126$°C. Because of the energy balance, the plant optimum is specified by controlling any two temperatures of this set to 126 °C.

The relatively large elements in the steady state relative gain array [Skogestad and Postlethwaite, 2005] from $\mathbf{u} = [u_2, u_7]^{\mathbf{T}}$ to $\mathbf{y} = [y_1, y_2, y_4]^{\mathbf{T}}$,

$$RGA = \begin{bmatrix} 1.1919 & -0.3854 \\ 0.1182 & 0.0758 \\ -0.3101 & 1.3096 \end{bmatrix}, \tag{A.18}$$

suggest pairing u_2 with y_1 and u_7 with y_4. However this leads to a very poor dynamic performance, because in this pairing u_2 has very little initial gain on y_1 due to the equality of the exit temperatures of heat exchanger and cooler.

From the energy balance it is clear that the heat has either to be removed in the heat exchanger or the cooler. Therefore opening the cooler valve u_2 alone will not have the desired effect on y_1. If the furnace return temperature y_1 becomes too hot and the cooler valve u_2 opens, it acts as a bypass and y_1 increases even further. However, as it closes, more water goes through the main heat exchanger, and the temperature y_1 increases as well. To effectively reduce the furnace return temperature y_1, it has to be controlled by the cooler duty u_7.

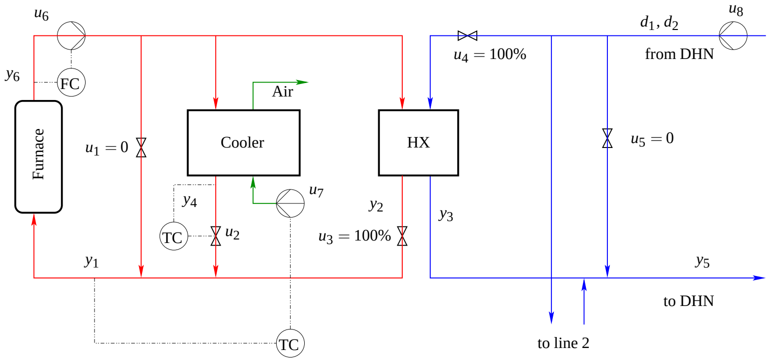


Figure A.6: Control structure for region δ

A set of pairings which gives good dynamic and steady state performance, is to use the cooling fan duty u_7 to control the furnace return temperature y_1 and to control the cooler exit temperature y_4 manipulating the cooler valve u_2 (Figure A.6).

This pairing ensures that the fan duty u_7 increases before the disturbance coming from the district heating network affects the cooler exit temperature y_4, and avoids the bypass effect when the cooler valve u_2 starts opening.

A.7 Dynamic model and simulations

To test the control structure and simulate the process, the model described in A.4 is extended to a dynamic model, where the heat exchangers are modelled as ideal tanks, Figure A.7. A first order transfer function with a time constant of $\tau = 1.5$s is used to add dynamics to the pumps, valves and fans. The mixers and splitters remain as modelled for the steady state case.

Each heat exchanger is modelled by 10 equal heat exchanger sections, with the governing equations [Mathisen, 1994]:

$$\frac{dT_h^i}{dt} = \left(T_h^{i+1} - T_h^i - \frac{h_h A}{w_h c_{p,h} N}\Delta T_h^i\right)\frac{w_h N}{\rho_h V_h} \tag{A.19}$$

$$\frac{T_w^i}{dt} = \left(h_h \Delta T_h^i - h_c \Delta T_c^i\right)\frac{A}{\rho_w c_{p,w} V_w} \tag{A.20}$$

$$\frac{dT_c^i}{dt} = \left(T_c^{i+1} - T_c^i - \frac{h_c A}{w_c c_{p,c} N}\Delta T_c^i\right)\frac{w_c N}{\rho_c V_c} \tag{A.21}$$

In the above equations T denotes the temperature, h the heat transfer coefficient,

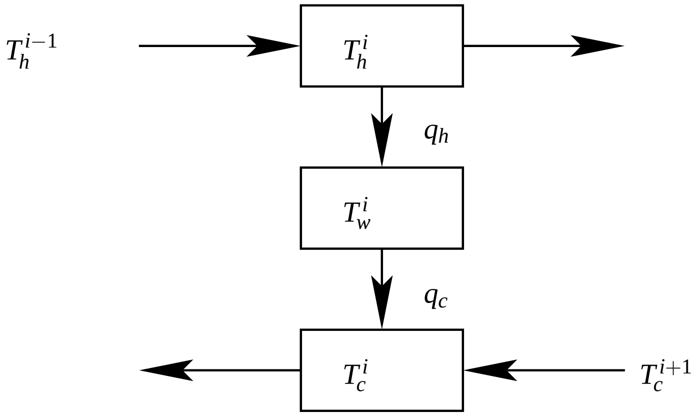


Figure A.7: Heat exchanger section

A the total heat transfer area, w the mass flow rate, c_p the heat capacity flow rate, N the number of sections, ρ the fluid density and V the volume. The superfix i denotes the compartment, while the suffices h, c, and w denote the hot side, the cold side, and the wall element, respectively. The terms ΔT_h^i and ΔT_c^i denote the signed difference between the wall and the hot and cold side of section i, respectively.

Modelling the heat exchangers discrete instead of continuous moves the regions in Figure A.2 slightly up, approximately 1 °C, but does not affect the structure of the optimal solution. Using the dynamic model and the control structures developed above, the process was simulated for various scenarios in the different regions and for disturbances across region boundaries.

A.7.1 Control within regions

As an example, the control performance in region α and δ is presented here. The performance in the remaining regions, β and γ, is similar.

Region α

In region α, the control structure is well capable of keeping the variation in the furnace return temperature close to its desired value, while the self-optimizing control variable u_1 returns to its optimal value (Figure A.8). This reflects the control priorities: first, the active constraints are satisfied (y_1 close to 126), and second the system readjusts to optimal operation.

Region α and β have in common, that they have one unconstrained degree of freedom. After all active constraints have been satisfied, this degree of freedom is

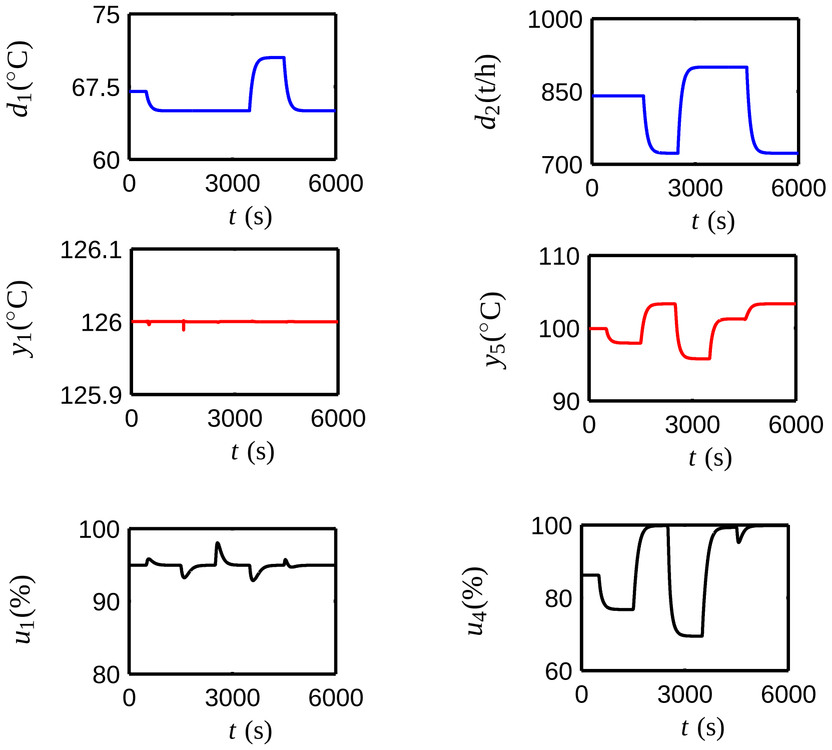


Figure A.8: Control performance in region α. d_1, d_2: disturbances, y_1: active constraint, u_1: self-optimizing variable, y_5: temperature to DHN, u_4: secondary HX valve

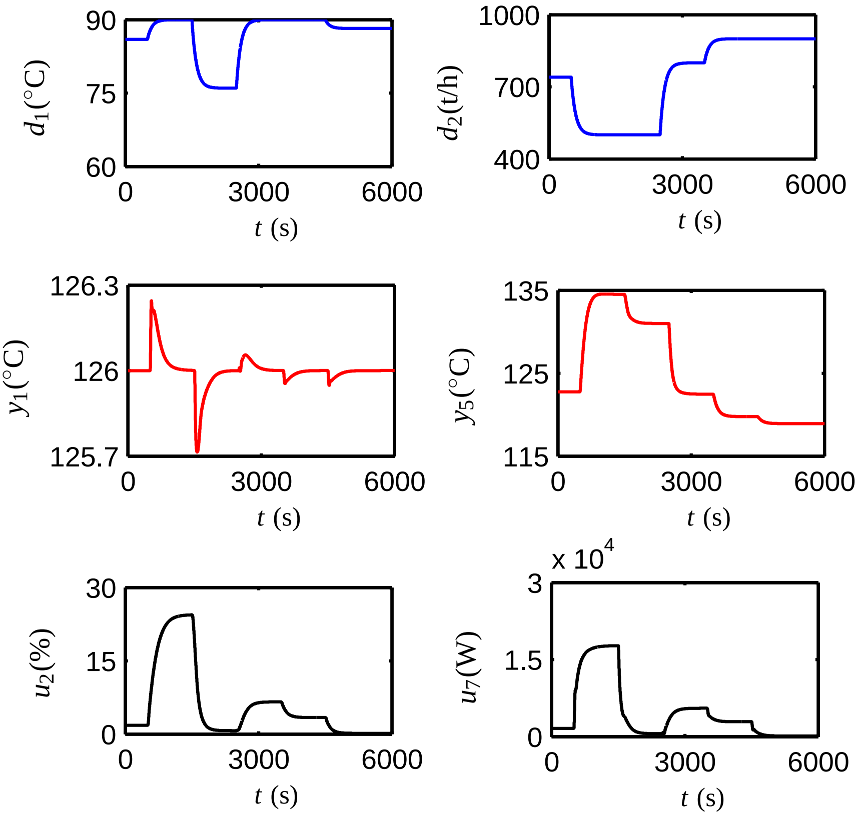


Figure A.9: Control performance in region δ. d_1, d_2: disturbances, y_1: active constraint, u_2: cooler valve, y_5: temperature to DHN, u_7: fan duty

used to minimize the pump work, which is minimized when u_1 is at 94.8%. The value differs a bit from the steady state value, because of the differences between the dynamic and the steady state value.

Region δ

In Figure A.9 the control performance for a disturbance in region δ is shown. It can be seen that the combination of cooling fan duty u_7 and cooler valve u_2 ensures deviation of less than 0.3°Cin y_1, even though the disturbances entering the plant are large.

In regions γ and δ, all inputs are needed to guarantee that the constraints are

Table A.7: Switching conditions

Transition	Condition 1	Condition 2	Condition 3
$\alpha \rightarrow \beta$	$u_4 = 100\%$	$t < t_s$	Region = α
$\beta \rightarrow \gamma$	$u_5 = 0.00\%$	$t < t_s$	Region = β
$\gamma \rightarrow \delta$	$u_1 = 0.00\%$	$t < t_s$	Region = γ
$\delta \rightarrow \gamma$	$u_7 = 0.00\%$	$t < t_s$	Region = δ
$\gamma \rightarrow \beta$	$u_1 \leq 94.8\%$	$t < t_s$	Region = γ
$\beta \rightarrow \alpha$	$u_5 = 100\%$	$t < t_s$	Region = β

satisfied. Therefore, therefore, optimal operation is completely specified by controlling the active constraints.

A.7.2 Switching between regions

When the change in the disturbances becomes so large that a region boundary is crossed, it is necessary to detect this event and to switch the control structure. For the switching strategy here it is assumed that the system cannot jump over a region, i. e. the disturbances move the system gradually into the new region. Then, the switching logic can be based on monitoring the unconstrained and self-optimizing variables of the current and neighbouring regions. For example, in region α variable u_4 is unconstrained. If a disturbance enters such that u_4 goes into saturation, the unconstrained variable of region β is released and used for control. The same strategy is used for switching in the other regions. Switching from γ into the unconstrained region β is done when the self-optimizing variable of region β, u_1, reaches its optimal set-point. This is possible since in region γ the valve u_1 assumes a strictly smaller value than in region β.

To avoid chattering, the regions are switched when the corresponding variable has been in saturation or crossed its value for more than $t_s = 2.5$ minutes. Using this strategy we ensure that the control structure of one region is active long enough to realize its effects before switching to the next control system. The conditions for switching are listed in the Table A.7.

As an example, we consider a temperature rise in the district heating network, moving the system from region β to γ, Figure A.10. The variable u_1 is constant until u_5 goes into saturation. Then u_1 leaves the optimal point of region α to control the furnace inlet temperature y_1.

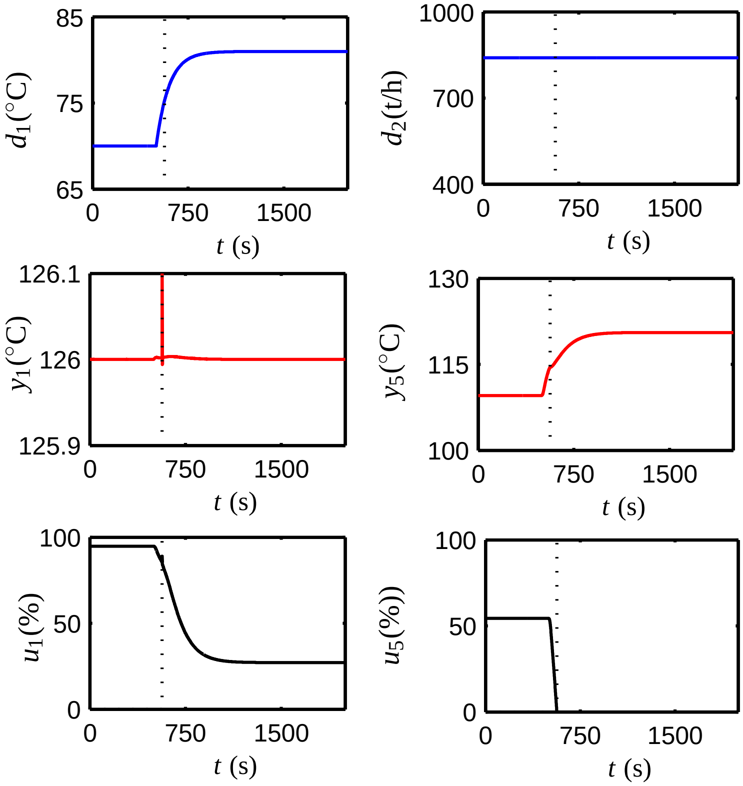


Figure A.10: Region switching from β to γ (dotted line shows switching instant)

A.8 Discussion

This case study demonstrates the nicely the concept and properties of self-optimizing control. First, the active constraints are controlled, and if there are remaining degrees of freedom, they are used to control self-optimizing variables. The advantage is that we have a very simple and easy to implement control structure in every region. Moreover, our procedure leads to a good understanding of the operating conditions and constraints. This understanding can be beneficial for operation of the plant. In addition, it is easy to communicate to operators, as the control structures for each region are simple and easy to understand and maintain.

In region β the valve u_5 was used to control the valve u_1. Since there are two lines (each having one valve u_1), we are lacking one degree of freedom. The above analysis was made based on the unrealistic assumption of two exactly symmetrical lines. This will not be the case in practice, therefore, a more practical solution could be to control $c = \frac{1}{2}(u_1^1 + u_1^2)$ to 94.8%, and letting u_1 control the furnace return temperature y_1. This would ensure the correct furnace return temperature, while in letting u_5 affect both valves. Alternatively, one could use the split between the lines as an additional degree of freedom to control one u_1 to its optimal setpoint.

The challenge in handling several control structures is clearly tracking the operating regions and switching correctly. In this case study it has been found that monitoring the controlled variables of the four regions yields good results. The self-optimizing approach has been found to be a simple alternative to methods like real-time optimization or model predictive control, where the constraints are handled implicitly, such that where the operating regions do not become visible. Considering the simplicity of the control structure and the excellent control performance, it seems that the effort of maintaining and installing an online optimizing control system may not be able to improve performance significantly. However this would have to be investigated in a separate study.

A.9 Conclusion

We have presented a case study of a waste incineration plant which is operated close to optimality using very simple control configurations and simple switching rules for changing between them. The applied procedure reveals the different operation regions obtained from steady state optimization, and yields an intuitive and understandable control structure, while at the same time giving optimal operation. The switching rules are based on monitoring the constrained and self-optimizing variables and information about the system dynamics. For this process, the self-optimizing control approach seems to be an attractive alternative to real-time optimization.

References

Hertzberg, T., 2007. Enkel varmeveksler, internal report, department of chemical engineering, ntnu, trondheim.

Marlin, T. E., Hrymak, A., 1997. Real-time operations optimization of continuous processes. In: Proceedings of CPC V, AIChE Symposium Series vol. 93. pp. 156–164.

Mathisen, K. W., 1994. Integrated design and control of heat exchanger networks. Ph.D. thesis, Norwegian University of Science and Technology, NTNU, Trondheim.

Narasimhan, S., Skogestad, S., 2007. Implementation of optimal operations using off-line computations. 8th international IFAC Symposium on Dynamics and Control of Process Systems (DYCOPS), Cancun, Mexico.

Skogestad, S., 2000. Plantwide control: The search for the self-optimizing control structure. Journal of Process Control 10, 487–507.

Skogestad, S., 2009. Chemical and Energy Process Engineering. CRC Press.

Skogestad, S., Postlethwaite, I., 2005. Multivariable Feedback Control, 2^{nd} Edition. John Wiley & Sons.

Smedsrud, H., 2008. Dynamic modeling and control of brobekk incineration plant. Master's thesis, Norwegian University of Science and Technology, NTNU.