technische universität dortmund

Process Dynamics and Operations

TU Dortmund University
Department of Biochemical and Chemical Engineering
Process Dynamics and Operations Group
Prof. Dr.-Ing. Sebastian Engell

Master Thesis

An Improved Iterative Real-time Optimization Scheme for Slow Processes

by

José Luis Cadavid

Supervision:

Senior researcher:	M.Sc. Reinaldo Hernández
Examiner:	Prof. Dr.-Ing. Sebastian Engell
Co-Examiner:	Prof. Dr. David W. Agar

Dortmund, November 16, 2016

DECLARATION

I hereby declare and confirm that the master thesis

An Improved Iterative Real-time Optimization Scheme for Slow Processes

is entirely the result of my own work except where otherwise indicated.

On the next page of this document, I have acknowledged the supervision, guidance and help I received from Prof. Dr.-Ing. Sebastian Engell and M.Sc. Reinaldo Hernández

Dortmund, November 16, 2016

José Luis Cadavid C.

ACKNOWLEDGMENTS

This work was done in the Process Dynamics and Operations Group of the Department of Biochemical and Chemical Engineering at TU Dortmund. I would like to thank M.Sc. Reinaldo Hernández for giving me the freedom to explore the different topics in this thesis, and for always allowing me to make this work my own, while guiding me and helping me shape it; both this work and myself have benefited tremendously from his assistance. I would also like to thank Prof. Dr.-Ing. Sebastian Engell for allowing me to work on this field, and for his insightful comments. My gratitude also goes to Prof. Dr. David Agar for agreeing to co-supervise this work.

I would like to thank all of my friends and colleagues here in Dortmund for making these two years a great experience, and helping me grow as a person and an engineer; fortunately, there are too many of them as to list everyone individually.

Last but not least, special thanks go to my parents and brother for supporting me during my time in Germany, regardless of the distance. This work is dedicated to you.

Dortmund, November 16, 2016

José Luis Cadavid

ABSTRACT

Iterative Real-Time Optimization (RTO) has gained increasing attention in the context of model-based optimization of the operating points of chemical plants in the presence of plant-model mismatch. In all these schemes, it is necessary to wait for the plant having reached a steady-state to obtain the required information on plant performance and constraint satisfaction, which leads to slow convergence in the case of processes with slow dynamics. This works addresses this issue by considering both parametric, and structural plant-model mismatch. First, a simple approach to determine the type of plant-model mismatch with the use of transient data is discussed. An approach for dealing with parametric mismatch based on a sensitivity analysis of the nominal dynamic model is presented, and its performance is evaluated with the case-study of a Continuously Stirred Tank Reactor (CSTR), where fast convergence to the optimum can be obtained, even with noisy measurements.

For the case of structural mismatch, nonlinear system identification is integrated with iterative RTO. The identified models are used to predict the steady-state of the system, thus reducing the total optimization time. The performance of the strategy is illustrated by simulation studies of a CSTR and a hydroformylation process. It is shown that a mixed scheme, where both a linear and nonlinear model are used for steady-state prediction, results in fast convergence to a neighborhood of the true optimum, even in the presence of measurement noise. The use of taylored nonlinear models for dynamic system identification is shown to be a promising approach for reducing the time necessary to reach the optimum of a process.

CONTENTS

List of Tables **III**

List of Figures **V**

Nomenclature **VII**

1. Introduction **1**

2. Theoretical Background **5**
2.1. Static optimization problem 5
2.2. Real-time optimization 7
2.2.1. Classification of real-time optimization schemes 8
2.2.2. Modifier adaptation 9
2.3. Estimation of the plant gradients 11
2.3.1. Gradients from steady-state information 11
2.3.2. Modifier adaptation with quadratic approximation 13
2.3.3. Gradients from transient information 16

3. Determining the type of plant-model mismatch **21**
3.1. Rejecting parametric mismatch 21
3.2. Case study 1 24
3.2.1. Reactor model 24
3.2.2. Evaluation of plant-model mismatch 25

4. An improved RTO scheme for slow processes with parametric model uncertainties **27**
4.1. Parameter estimation from linearized dynamic model 28
4.1.1. Toy example 30
4.2. RTO with parameters from LDM 33

4.3. Case study 1 34
4.3.1. Optimization problem 34
4.3.2. Simulations with no measurement noise 35
4.3.3. Simulations with measurement noise 36
4.4. Additional remarks on the method for parameter estimation . . 39

5. Dynamic model identification for RTO **41**
5.1. Nonlinear system identification 41
5.1.1. The NARMAX model 42
5.1.2. Model structure selection 44
5.1.3. Parameter estimation 48
5.2. Case study 1 51
5.2.1. Dynamic model selection 51
5.2.2. Evaluation of the selected nonlinear model 53
5.2.3. Simulations with no measurement noise 56
5.2.4. Simulations with measurement noise 57
5.3. Case study 2 60
5.3.1. Process description 60
5.3.2. Optimization problem 61
5.3.3. Data reconciliation 62
5.3.4. Dynamic model selection 62
5.3.5. Simulations with no measurement noise 67
5.4. Additional remarks on the total optimization time 75

6. Conclusions **77**

Bibliography **i**

A. Model of the TMS process **vii**

B. Eidesstattliche Versicherung **ix**

LIST OF TABLES

3.1. Nominal model parameters for the CSTR case-study 25
3.2. Determination of P-Invalidity for the CSTR case-study 26

4.1. Parameters of the optimization problem for the CSTR case-study 35
4.2. Solutions to the model and plant optimization problems for the CSTR case-study 35
4.3. Optimization of the CSTR case-study with the LDM approach . 38

5.1. Input-output models for the description of the CSTR case-study 54
5.2. Performance of the dynamic models for the CSTR case-study . . 55
5.3. Optimization of the CSTR case-study with measurement noise . 58
5.4. Operating conditions of the TMS system 62
5.5. Effect of the regularization parameter α on the prediction of the steady-state for the TMS process 71
5.6. Optimization of the TMS process with dynamic system identification 73

A.1. Symbols and parameters of the TMS model viii

LIST OF FIGURES

2.1. Hierarchical control structure with real-time optimization (RTO) 7
2.2. Integration of system identification with RTO 17

3.1. Statistical test for rejecting the hypothesis of parametric mismatch 24
3.2. Diagram of the CSTR for case-study 1 24

4.1. Temporal derivatives predicted by the model and observed in the plant 29
4.2. Estimated reaction constant for the toy example with the linearized dynamic model 31
4.3. Estimated reaction constant with noisy measurements for the toy example with the linearized dynamic model 32
4.4. RTO scheme with parameters from the linearized dynamic model 33
4.5. RTO of the CSTR case-study using linearized dynamic model . 37

5.1. Representation of MIMO models with MISO models 44
5.2. Parameter estimation with the instrumental variables method . 50
5.3. Procedure for forming bootstrap data 52
5.4. Flowchart for the moving window algorithm for parameter estimation 55
5.5. Objective function values for the set-points applied with the nonlinear model, the linear model, and a mixed approach for case-study 1 56
5.6. Optimization trajectory of the nonlinear model and the linear model for case-study 1 58
5.7. Thermomorphic Multicomponent Solvent (TMS) system 60
5.8. Forward selection with pruning algorithm for model structure selection 64
5.9. Results of the dynamic model selection for case-study 2 66

5.10. Effect of the regularization penalty α and the minimum window length L_0 on the optimization trajectory for the TMS process . 69
5.11. Effect of the regularization penalty α and the minimum window length L_0 on the error on the predicted steady-state for the TMS process 69
5.12. Effect of the regularization penalty α and the minimum window length L_0 on the time required before a steady-state prediction is obtained 70
5.13. Optimization of the TMS process with static data, predictions from a linear model, and predictions from a mixed approach . . 74

NOMENCLATURE

Symbol	Meaning
$\mathcal{B}$	Search space
$\mathbf{f}$	Dynamic (continuous time) model
F	Dynamic (discrete time) model
G	Scalar constraint
$\mathbf{G}$	Constraint vector
$\mathbf{H}$	Static (vectorial) input-output mapping
J	Scalar objective function
L_0	Minimum window size
$\mathcal{M}$	Regressor set
$\mathbf{S}$	Set-point matrix
t_{RTO}	Re-optimization period
t_s	Sampling time
$\mathbf{u}$	Set-point (input vector)
$\underline{\mathbf{u}}$	Lower bound on inputs
$\overline{\mathbf{u}}$	Upper bound on inputs
$\mathcal{U}$	Regression set
x	General variable
$\hat{x}$	Measured, estimated, or reconciled quantity
$\bar{x}$	Steady-state variable, average value
y	Scalar output
$\mathbf{y}$	Multiple outputs vector
$\mathbf{Y}$	Stacked outputs

Greek letters	Meaning
α	Significance level, regularization constant
Δ	Difference
ϵ	Zeroth order modifiers, tolerance

Continued on next page

– continued from previous page

Greek letters	**Meaning**
γ	Scaling factor
λ	First order modifiers
ϕ	Regressor
$\boldsymbol{\Phi}$	Stacked regressor matrix
ρ	Prediction quality indicator
$\boldsymbol{\Sigma}$	Covariance matrix, summation
σ_n	Standard deviation of noise
$\boldsymbol{\theta}$	Parameter vector
$\boldsymbol{\varphi}$	Regressor vector
$\boldsymbol{\xi}$	Stacked error vector
χ^2	Chi-squared distribution

Indices	**Meaning**
$*$	Optimal
$+$	Moore-Penrose pseudoinverse
ad	Adapted model
e	Error, noise
i, j	General subindex for numbering
k	Iteration index, sampling point
m	Model
p	Plant
u	Inputs
ϕ	Quadratic approximation
θ	Parameter
T	Transposed
y	Outputs

Abbreviation	**Meaning**
ARX	Autoregressive with Exogenous Inputs
ARMAX	Autoregressive Moving Average with Exogenous Inputs
BS	Bootstrap
CSTR	Continuously Stirred Tank Reactor
DFO	Derivative-free optimization
DMF	Dimethylformamide
ELS	Extended Least Squares
ERR	Error Reduction Ratio
ESC	Extremum Seeking Control
FROE	Forward Regression Orthogonal Estimator

Continued on next page

– continued from previous page

Abbreviation	**Meaning**
I/O	Input-Output
IGMO	Iterative Gradient-Modification Optimization
ISOPE	Integrated System Optimization and Parameter Estimation
IV	Instrumental Variables
KKT	Karush-Kuhn-Tucker
LASSO	Least Absolute Shrinkage and Selection Operator
LDM	Linearized Dynamic Model
LS	Least Squares
MA	Modifier-Adaptation
MAWQA	Modifier Adaptation with Quadratic Approximation
MIMO	Multiple-Input Multiple-Output
MISO	Multiple-Input Single-Output
MPC	Model Predictive Control
NARMAX	Nonlinear Autoregressive Moving Average with Exogenous Inputs
NARX	Nonlinear Autoregressive with Exogenous Inputs
NCO	Necessary Conditions of Optimality
NE	Neighboring Extremals
NLS	Nonlinear Least Squares
NOE	Nonlinear Output Error
ODE	Ordinary Differential Equation
OLS	Ordinary (linear) Least Squares
PDF	Probability Density Function
PL	Profit Loss
PRBS	Pseudo-Random Binary Sequence
QA	Quadratic Approximation
RaMSS	Randomized Model Structure Selection
RIP	Regressor Inclusion Probability
RTO	Real-time Optimization
SISO	Single-Input Single-Output
SOC	Self-Optimizing Control
SRR	Simulation Error Reduction Ratio
TMS	Thermomorphic Multicomponent System

CHAPTER

ONE

INTRODUCTION

With increasing global competition, companies in the process industry face the challenge of constantly improving production efficiency, while meeting high quality standards, and satisfying safety and environmental regulations. Model-based real-time optimization (RTO) has emerged as an attractive approach to tackling this issue (Darby et al., 2011). The key idea in RTO schemes is to make use of a stationary nonlinear, usually first principle based mathematical model of the plant, and to use the available plant measurements to update some model parameters, with the goal of operating the process as closely as possible to its optimum.

Despite the increasing acceptance of RTO by the industry, there are important limitations of the approach. One of them is the fact that in the presence of model inaccuracies, convergence to the true plant optimum is not ensured; even worse, the optimization might lead to constraint violations. Over the last decades, research has mainly focused on this issue by developing algorithms that take into account measurements of the plant to compensate for plant-model mismatch; thus allowing convergence to the real plant optimum, even in the presence of structural mismatch of the model (Gao and Engell, 2005). Nonetheless, one of the main difficulties associated with such schemes is the experimental determination of plant gradients with noisy measurement data.

Furthermore, the problem of having to wait for steady-state after each new set-point has been applied to the plant is detrimental to the efficiency of the

scheme, especially for slow processes. This can result in significant profit losses from operating the plant at a sub-optimal set-point for an extended period of time, or can lead to wrong optima if slow disturbances affect the plant significantly during the timescale of the optimization procedure. For this reason, the use of measurements collected during transient operation has been proposed to accelerate the optimization scheme; be it by estimating the plant gradients, or by predicting the steady-state of the plant (Bamberger and Isermann, 1978; Zhang and Roberts, 1990; Gao and Engell, 2016).

In the case of parametric plant-model mismatch, different techniques based on the nominal structure of the model have been proposed to make use of transient data. For the more general case of structural mismatch, dynamic model identification has been successfully used, but the structure of the black-box model employed must be set a priori, and usually corresponds to linear models or generalized Hammerstein structures that do not necessarily capture the dynamics of the system properly (Mansour and Ellis, 2003).

The goal of this work is to use transient measurements for static RTO by systematically making use of available data to determine: the type of plant-model mismatch at hand, and the appropriate black-box model for use in the context of dynamic system identification. This work is intended to be used as a general toolbox, where the application of case-specific strategies is expected to improve the performance of the RTO system when transient measurements are used. The rest of this work is structured as follows: the fundamentals of different RTO schemes, with particular emphasis on Modifier Adaptation With Quadratic Approximation (MAWQA), are presented in Chapter 2, along with a summary of different techniques for experimental gradient calculation, and a brief review of the use of transient data in the context of RTO. A simple approach based on goodness-of-fit measures to determine the type of plant-model mismatch at hand is presented and evaluated in Chapter 3. Chapter 4 presents a modified scheme for handling parametric mismatch, based on an online parameter estimation method that uses the nominal (mismatched) dynamic model. The scheme is evaluated through simulation studies of a continuously stirred tank reactor (CSTR).

Chapter 5 is devoted to dealing with the case where plant-model mismatch is structural, by using dynamic model identification. A short introduction to

nonlinear system identification is first given. The case-study of Chapter 4 is again evaluated with numerous simulation studies: a nonlinear model structure is selected to describe its dynamics, and later used for RTO of the system. A second case-study consisting of a homogeneously catalyzed hydroformylation process is also evaluated and optimized in this chapter. Finally, some concluding remarks and possible research directions are outlined in Chapter 6.

CHAPTER

TWO

THEORETICAL BACKGROUND

This chapter discusses the different theoretical aspects required to formulate and understand the RTO problem, as well as the use of transient data in this context. First, the steady-state optimization problem is stated, followed by a brief overview of the general classes of RTO schemes. Special attention is given to Modifier-Adaptation (MA) schemes, where some of the general techniques for experimental gradient determination are discussed, and the use of Quadratic Approximation (QA) is treated in more depth. The state of the art on the use of transient measurements for static RTO is then presented.

2.1. Static optimization problem

Optimization for continuous processes operating at steady-state consists in determining the operating conditions that result in optimal process performance, usually in terms of maximizing profit or minimizing costs, while different constraints (e.g. requirements of product quality, safety constraints) are satisfied. This problem is formulated mathematically as:

$$\begin{aligned} \min_{\mathbf{u}} \quad & J_p(\mathbf{u}, \bar{\mathbf{y}}_p) \\ \text{s.t.} \quad & \mathbf{G}_p(\mathbf{u}, \bar{\mathbf{y}}_p) \leq 0 \\ & \underline{\mathbf{u}} \leq \mathbf{u} \leq \overline{\mathbf{u}} \end{aligned} \tag{2.1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ are the inputs to the plant, bounded by $(\underline{\mathbf{u}}, \overline{\mathbf{u}})$, $\bar{\mathbf{y}}_p \in \mathbb{R}^{n_y}$ are the measured (or estimated) plant outputs[1], $J_p \colon \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is the scalar objective function, and $\mathbf{G}_p \colon \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_g}$ is a vector of constraints. The structure of the objective function and the constraints is always known, since it is user-defined. However, plant-model mismatch results from the fact that the true mapping between the inputs and the outputs of the plant is not always perfectly known, and is approximated with a simplified model:

$$\bar{\mathbf{y}}_m = \mathbf{H}_m(\mathbf{u}, \boldsymbol{\theta}_m) \tag{2.2}$$

where $\mathbf{H}_m \colon \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_y}$ is the static input-output (I/O) mapping described by the model, and $\boldsymbol{\theta}_m \in \mathbb{R}^{n_\theta}$ is a vector containing its parameters. With this, problem 2.1 can be reformulated as:

$$\begin{aligned} \min_{\mathbf{u}} \quad & J_m(\mathbf{u}, \mathbf{H}_m(\mathbf{u}, \boldsymbol{\theta}_m)) \\ \text{s.t.} \quad & \mathbf{G}_m(\mathbf{u}, \mathbf{H}_m(\mathbf{u}, \boldsymbol{\theta}_m)) \leq 0 \\ & \underline{\mathbf{u}} \leq \mathbf{u} \leq \overline{\mathbf{u}} \end{aligned} \tag{2.3}$$

A distinction is made between the objective functions J_m and J_p given by the model and the plant, respectively, because they can be explicitly dependent on the mismatched parameters[2]. Throughout this work, the sub-index m refers to quantities estimated from a nominal model, while the sub-index p refers to the true quantities of the plant. Under certain regularity conditions, an solution to the optimization problem exists (Chachuat et al., 2009). At the optimal solution of problems 2.1 and 2.3, the first order necessary conditions of optimality (NCO) - also known as the Karush-Kuhn-Tucker (KKT) conditions - must hold (Bazaraa et al., 1992):

$$\begin{aligned} G_i(\mathbf{u}^*, \boldsymbol{\theta}) \leq 0, \nu_i^* \geq 0, i = 1, \ldots, n_g \\ \frac{\partial J(\mathbf{u}^*, \boldsymbol{\theta})}{\partial \mathbf{u}} + \sum_{i=1}^{n_g} \nu_i^* \frac{\partial G_i(\mathbf{u}^*, \boldsymbol{\theta})}{\partial \mathbf{u}} = 0 \\ \nu_i^* G_i(\mathbf{u}^*, \boldsymbol{\theta}) = 0, i = 1, \ldots, n_g \end{aligned} \tag{2.4}$$

where ν^* is the vector of KKT multipliers.

[1] The notation $\bar{x}$ is used in this work to refer to the steady-state value of x

[2] The same observations might apply to the constraints

2.2. Real-time optimization

The goal of RTO systems is to guarantee that the process operates as close as possible to its optimum, while satisfying different constraints, based on a timescale hierarchical decomposition of the problem. The position of an RTO system in the overall control hierarchy is illustrated in Figure 2.1: RTO acts in an upper control layer, where a model-based optimization routine is run iteratively in closed loop, and provides set-points to the controllers in the regulatory control layer; this scheme results in a clear separation between the tasks of optimization and control (Darby et al., 2011).

In general, the planning and scheduling layer sets goals and constraints in production in a timescale from days to weeks. This is translated into objectives for the RTO system, which computes the optimal set-points and operates on a medium timescale from hours to days. The regulatory control systems provide set-point tracking and disturbance rejection on a short timescale from seconds to minutes (Engell, 2007). The regulatory controllers can be any type of system, from simple linear controllers, to nonlinear model predictive controllers (Darby et al., 2011).

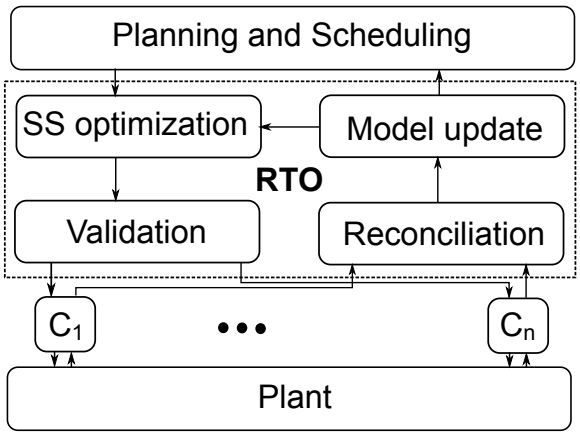


Figure 2.1.: Hierarchical control structure with real-time optimization (RTO), $C_1 \cdots C_n$ denote local regulatory controllers. Adapted from Engell (2007)

In the case of continuous processes, RTO is formulated as a static optimization problem as the one described in Section 2.1. The problem of RTO is akin to the optimization of rigorous detailed models with surrogate models, where a direct search, or optimization of the rigorous model - the plant in RTO - is

too costly; therefore, optimization is performed with simpler models. As such, convergence to the optimum of the plant is only possible if the reduced model and the detailed model have matching KKT conditions (Biegler et al., 2014).

Real-time optimization systems have been successfully implemented in different industrial processes; particularly in oil refineries, where the use of this technology results in a significant increase in profits (Shokri et al., 2009).

2.2.1. Classification of real-time optimization schemes

In broad terms, RTO strategies can be classified depending on how they use plant measurements to drive the process iteratively to its optimum, while still guaranteeing feasible operation. Chachuat et al. (2009) proposed the following categories, which are only briefly described here:

Model-parameter adaptation: This corresponds to the classical two-step approach, where the parameters of a rigorous steady-state model are updated once the real measurements are available. This updated model is then re-optimized, and the procedure is repeated until convergence is achieved. This method works well in the case of parametric plant-model mismatch, but for structural mismatch it is not possible to guarantee convergence to the optimum. Furthermore, deciding which, and how many parameters to update is also complicated: adapting too few parameters might lead to bad fitting, while adapting too many parameters requires a higher degree of plant excitation for proper identification (Chachuat et al., 2009).

Modifier-adaptation: The idea behind MA is to modify problem 2.3 by adding correction terms to the objective function and the constraints, in the form of zeroth- and first-order bias modifiers. In this way, the modified (or adapted) problem can match the KKT conditions of the plant after convergence. This allows convergence to the true optimum, even in the presence of structural plant-model mismatch. This approach is discussed in further detail in Section 2.2.2.

Direct input adaptation: The methods in this class differ from the previously described ones, in the sense that repeated optimization is transformed into a feedback control problem. This class of methods is also called implicit real-time optimization; François et al. (2012) present a comparison of six implicit control schemes related to RTO. Due to plant-model mismatch, a challenge in this approach is to determine the variables that should be kept constant in order to achieve optimal operation by manipulating the inputs. Another challenge in implicit RTO is the determination of the active-set of constraints, which can change over time. Possible solutions based on barrier functions have been proposed to tackle this problem (Srinivasan et al., 2008).

Skogestad (2000) proposed the idea of Self-Optimizing Control (SOC), where the sensitivity of the optimal inputs to parametric uncertainty is used to determine a set of variables, also called self-optimizing variables, to be controlled. They correspond to a linear combination of the output variables; the tracking of which results in almost-optimal performance. This method is based on the assumption that the plant-model mismatch is parametric, and works only for a given set of nominal disturbances, i.e. if a disturbance was not considered in the sensitivity analysis, optimality cannot be ensured.

Another approach, which does not make use of a model, is to keep track of the NCO of the plant. The two most popular schemes in this class are Extremum-Seeking Control (ESC) (Krstić and Wang, 2000) and NCO tracking (François et al., 2005). In the former, the plant is perturbed with a dither signal (usually a low-frequency sinusoidal input), and the gradients are estimated via correlation analysis. The latter seeks to track the NCO of the plant based exclusively on measurements.

2.2.2. Modifier adaptation

The idea of MA can be traced back to the work of Roberts (1979), who proposed the Integrated System Optimization and Parameter Estimation (ISOPE) scheme. In ISOPE, the objective function given by the model is augmented with a first-order modifier, equivalent to the difference between the plant and model gradients. This allows convergence to the true plant optimum, even in the presence of structural plant-model mismatch; further research on the con-

vergence properties of ISOPE was done by Brydyś et al. (1987). A drawback of this scheme, is that the model parameters must still be estimated at each iteration.

Tatjewski (2002) reformulated ISOPE by showing that the parameter estimation must not be performed at every iteration, and could even be completely ommited; the name "modified gradient optimizing set-point control" is more appropriate to describe this scheme. Gao and Engell (2005) extended this idea to deal with process-dependent constraints, by also applying modifiers on them. This Iterative Gradient Modification and Optimization (IGMO) scheme is based on the following iterative problem formulation:

$$\begin{aligned} \mathbf{u}^{(k+1)} = \arg\min_{\mathbf{u}} \quad & J_m(\mathbf{u}, \boldsymbol{\theta}_m) + \epsilon_J^{(k)} + \boldsymbol{\lambda}_J^{(k)^{\mathrm{T}}}(\mathbf{u} - \mathbf{u}^{(k)}) \\ \text{s.t.} \quad & \mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta}_m) + \boldsymbol{\epsilon}_G^{(k)} + \boldsymbol{\lambda}_G^{(k)^{\mathrm{T}}}(\mathbf{u} - \mathbf{u}^{(k)}) \leq 0 \\ & \underline{\mathbf{u}} \leq \mathbf{u} \leq \overline{\mathbf{u}} \end{aligned} \tag{2.5}$$

where the zeroth order modifiers $\epsilon_J^{(k)}$ and $\boldsymbol{\epsilon}_G^{(k)}$ correct the bias between the values of the plant and the model, and the first order modifiers $\boldsymbol{\lambda}_J^{(k)^{\mathrm{T}}}$ and $\boldsymbol{\lambda}_G^{(k)^{\mathrm{T}}}$ correct the bias between the *gradients* of the plant and the model. At each new iteration, the optimal set-points $\mathbf{u}^{(k+1)}$ are applied to the plant. When steady-state is reached, the modifiers are calculated and the procedure is repeated:

$$\begin{aligned} \epsilon_J^{(k)} &= J_p(\mathbf{u}^{(k)}) - J_m(\mathbf{u}^{(k)}, \boldsymbol{\theta}_m) \\ \boldsymbol{\epsilon}_G^{(k)} &= \mathbf{G}_p(\mathbf{u}^{(k)}) - \mathbf{G}_m(\mathbf{u}^{(k)}, \boldsymbol{\theta}_m) \\ \boldsymbol{\lambda}_J^{(k)^{\mathrm{T}}} &= \frac{\partial J_p(\mathbf{u}^{(k)})}{\partial \mathbf{u}} - \frac{\partial J_m(\mathbf{u}^{(k)}, \boldsymbol{\theta}_m)}{\partial \mathbf{u}} \\ \boldsymbol{\lambda}_J^{(k)^{\mathrm{T}}} &= \frac{\partial \mathbf{G}_p(\mathbf{u}^{(k)})}{\partial \mathbf{u}} - \frac{\partial \mathbf{G}_m(\mathbf{u}^{(k)}, \boldsymbol{\theta}_m)}{\partial \mathbf{u}} \end{aligned} \tag{2.6}$$

The whole set of modifiers can be grouped as $\boldsymbol{\Lambda}^{(k)^{\mathrm{T}}} = \left(\epsilon_J^{(k)}, \boldsymbol{\epsilon}_G^{(k)}, \boldsymbol{\lambda}_J^{(k)^{\mathrm{T}}}, \boldsymbol{\lambda}_G^{(k)^{\mathrm{T}}}\right)$. The zeroth order modifier in the adapted objective function is often dropped, since it does not affect the position of the optimum. If the scheme converges, it will do so to a KKT point of the plant; the model should be adequate in order to ensure convergence (Forbes et al., 1994). The calculated modifiers are

often not applied directly, but are filtered to reduce the influence of noise and avoid over-correction (Marchetti et al., 2009):

$$\mathbf{\Lambda}^{(k+1)} = (\mathbf{I} - \mathbf{K})\mathbf{\Lambda}^{(k)} + \mathbf{K}\mathbf{\Lambda}^{(k+1)^*} \tag{2.7}$$

where $\mathbf{K}$ is a gain matrix, usually selected as a block-diagonal matrix to decouple the different modifiers, and $\mathbf{\Lambda}^{(k+1)^*}$ refers to the unfiltered modifier set calculated from eqs. 2.6. Another alternative is to filter the newly calculated set-point $\mathbf{u}^{(k+1)^*}$ in order to avoid large movements:

$$\mathbf{u}^{(k+1)} = (\mathbf{I} - \mathbf{K})\mathbf{u}^{(k)} + \mathbf{K}\mathbf{u}^{(k+1)^*} \tag{2.8}$$

where again $\mathbf{K}$ is a gain matrix, usually diagonal, $\mathbf{u}^{(k+1)}$ is the set-point applied to the plant, and $\mathbf{u}^{(k+1)^*}$ is the optimal set-point calculated from problem 2.5. Recent work has explored how the MA scheme can be formulated in a globally-convergent trust-region framework for the unconstrained case. In such a formulation, the necessity of filtering is eliminated because the search space is limited to an adaptive region where the adapted model can make accurate predictions (Biegler et al., 2014; Bunin, 2014).

2.3. Estimation of the plant gradients

A crucial part of the MA schemes is the estimation of the experimental plant and constraints gradients; this section briefly discusses the most common methods used for this task. The techniques based on static measurements are introduced first, followed by an overview of the techniques that use transient information. For a more comprehensive description of these methods, and a comparison between them, refer to (Mansour and Ellis, 2003; François et al., 2012).

2.3.1. Gradients from steady-state information

The most common ways for calculating plant gradients require steady-state to be reached. These schemes can also be used to calculate the gradients of the

constraints, but for simplicity sake all of the equations refer to the objective function J_p.

Gradients from finite differences: The simplest way of approximating the experimental gradients of the plant is via finite differences, e.g. with forward differences:

$$\frac{\partial J_p}{\partial u_i} \approx \frac{J_p(\mathbf{u} + \Delta h_i) - J_p(\mathbf{u})}{\Delta h_i} \tag{2.9}$$

where Δh_i is a perturbation applied to the i-th input. The use of finite differences is straightforward, but it requires n_u ($2n_u$ if central differences are used) disturbances at each set-point to estimate all the components of the gradient. This feature makes finite differences prohibitive for slow plants, or plants with many manipulated inputs. The selection of the perturbation size Δh_i is critical, since a too large step makes the approximation worse due to truncation error, but a too small step results in wrong estimations when the measurements are noisy.

Gradients from past set-points: This approach is also based on finite differences, but it uses the information collected from past set-points instead of disturbing the plant at each new set-point, which can reduce the total number of plant movements. The idea is to use the gradient of the hyperplane that contains the information of the most recent n_u evaluated set-points:

$$\begin{aligned} \left.\frac{\partial J_p}{\partial \mathbf{u}}\right|_{\mathbf{u}^{(k)}} &\approx \left(\mathbf{S}^{(k)}\right)^{-1} \cdot \begin{bmatrix} J_p^{(k)} - J_p^{(k-1)} \\ \vdots \\ J_p^{(k)} - J_p^{(k-n_u)} \end{bmatrix} = \left(\mathbf{S}^{(k)}\right)^{-1} \cdot \Delta \mathbf{J}_p^{(k)} \\ \mathbf{S}^{(k)} &= \left[\mathbf{u}^{(k)} - \mathbf{u}^{(k-1)}, \cdots, \mathbf{u}^{(k)} - \mathbf{u}^{(k-n_u)}\right]^{\mathrm{T}}, \end{aligned} \tag{2.10}$$

However, a problem arises if the matrix $\mathbf{S}^{(k)}$ becomes ill-conditioned. To overcome this problem, Brydyś and Tatjewski (1994) proposed to use this technique in a dual-control fashion, where the new set-point generated after optimization has to satisfy a constraint on the condition number of the $\mathbf{S}^{(k+1)}$

matrix. By limiting the set-point change with this constraint, the approach tackles the conflict between quickly advancing to the optimum, and calculating accurate gradients.

Gao and Engell (2005) also used eq. 2.10 to estimate the plant gradients, but instead of placing the condition number as a constraint in the optimization problem, they used the reciprocal condition number of $\mathbf{S}^{(k)}$ to decide whether an additional set-point was needed to keep the matrix well conditioned. In this way, additional perturbations are only applied when strictly necessary. In order to guarantee accurate gradients, Marchetti et al. (2010) introduced an upper bound on the error of the gradient due to truncation and noise, and used it to restrict the next optimization move by forcing this upper bound to be below a specified threshold.

Broyden update: Another way of using past data is the so-called Broyden method. This is based on the following formula (Broyden, 1965):

$$\begin{aligned} \left.\frac{\partial J_p}{\partial \mathbf{u}}\right|_{\mathbf{u}^{(k+1)}} &= \left.\frac{\partial J_p}{\partial \mathbf{u}}\right|_{\mathbf{u}^{(k)}} + \frac{\left(\Delta J_p^{(k)} - \left.\frac{\partial J}{\partial \mathbf{u}}\right|_{\mathbf{u}^{(k)}} \Delta\mathbf{u}^{(k)}\right)(\Delta\mathbf{u}^{(k)})^{\mathrm{T}}}{(\Delta\mathbf{u}^{(k)})^{\mathrm{T}}\Delta\mathbf{u}^{(k)} + \epsilon} \\ \Delta J_p^{(k)} &= J_p^{(k+1)} - J_p^{(k)} \\ \Delta\mathbf{u}^{(k)} &= \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}, \end{aligned} \tag{2.11}$$

where the small number ϵ is used to avoid division by zero. This formula becomes ill-conditioned when the set-point changes are small (i.e. $\Delta\mathbf{u} \to 0$). Furthermore, the gradient estimate needs to be properly initialized (Mansour and Ellis, 2003).

2.3.2. Modifier adaptation with quadratic approximation

Gao et al. (2016) recently proposed a robust RTO algorithm based on MA and inspired by Derivative-Free Optimization (DFO) methods. This concept, named "Modifier Adaptation with Quadratic Approximation" (MAWQA), is based on constructing quadratic approximations (QA) of the objective function

and constraints by selecting a suitable set of past set-points. The QA for the cost function is defined by:

$$J_\phi(\mathbf{u}, \boldsymbol{\theta}_\phi) = \sum_{i=1}^{n_u} \sum_{j=1}^{i} a_{i,j} u_i u_j + \sum_{i=1}^{n_u} b_i u_i + c \tag{2.12}$$

with the parameter set $\boldsymbol{\theta}_\phi = \{a_{1,1}, \cdots, a_{n_u,n_u}, b_1, \cdots, b_{n_u}, c\}$ obtained by solving the least squares (LS) problem

$$\min_{\boldsymbol{\theta}_\phi} \sum_{i=1}^{n_r} \left(J_p\left(\mathbf{u}^{(r_i)}\right) - J_\phi\left(\mathbf{u}^{(r_i)}, \boldsymbol{\theta}_\phi\right)\right)^2 \tag{2.13}$$

where $\mathbf{u}^{(r_i)}$ is part of the regression set $\mathcal{U}^{(k)}$ composed of past set-points. In order to guarantee well-poisedness of the regression problem, the regression set is determined with a screening algorithm. The new set-point obtained from the iteration on problem 2.5 is additionally restricted by an elliptical trust region defined by the covariance of the regression set. The scheme also includes a switching mechanism that verifies the quality of the approximation done by both the adapted problem 2.5, and the QA, and selects the best model for optimization on each iteration. The steps involved in the algorithm are directly presented from the original paper (Gao et al., 2016):

1. Choose an initial set-point $\mathbf{u}^{(0)}$ and probe the plant at $\mathbf{u}^{(0)}$ and $\mathbf{u}^{(0)} + h\mathbf{e}_i$, where h is a suitable step size and $\mathbf{e}_i \in \mathbb{R}^{n_u} (i = 1, \cdots, n_u)$ are mutually orthogonal unit vectors. Calculate the gradients at $\mathbf{u}^{(0)}$ with finite differences, and run the IGMO algorithm of Gao and Engell (2005) until $(n_u+1)(n_u+2)/2$ set-points have been generated. Run the screening algorithm to acquire the regression set $\mathcal{U}^{(k)}$. Initialize $\rho_m^{(k)} = 0$ and $\rho_\phi^{(k)} = 0$.
2. Calculate the quadratic functions $J_\phi^{(k)}$ and $\mathbf{G}_\phi^{(k)}$ by solving problem 2.13. Define the search space $\mathcal{B}^{(k)}$ from the regression set $\mathcal{U}^{(k)}$ as:

$$\mathcal{B}^{(k)} \colon (\mathbf{u} - \mathbf{u}^{(k)})^{\mathrm{T}} \mathbf{M}^{-1} (\mathbf{u} - \mathbf{u}^{(k)}) \leq \gamma^2 \tag{2.14}$$

where $\mathbf{M} = \mathrm{cov}\left(\mathcal{U}^{(k)}\right)$ is the covariance matrix of the regression set, and γ is a scaling parameter related to the size of the hyper-ellipsoid defined by $\mathcal{B}^{(k)}$.

3. Extract the gradients from the quadratic functions. Determine a new candidate set-point $\hat{\mathbf{u}}^{(k)}$ as follows:

   (a) If $\rho_m^{(k)} \leq \rho_\phi^{(k)} = 0$, run the adapted model-based optimization in problem 2.5 with the additional constraint $\mathbf{u} \in \mathcal{B}^{(k)}$.

   (b) Else perform an optimization based on the quadratic approximation:

$$\begin{aligned} \min_{\mathbf{u}} \quad & J_\phi^{(k)}(\mathbf{u}) \\ \text{s.t.} \quad & \mathbf{G}_\phi^{(k)}(\mathbf{u}) \leq 0 \\ & \mathbf{u} \in \mathcal{B}^{(k)} \\ & \underline{\mathbf{u}} \leq \mathbf{u} \leq \overline{\mathbf{u}} \end{aligned} \tag{2.15}$$

4. If $\|\hat{\mathbf{u}}^{(k)} - \mathbf{u}^{(k)}\| < \Delta\mathbf{u}$, where $\Delta\mathbf{u}$ is the parameter used by the screening algorithm to define the regression set, and there exists at least one point $\mathbf{u}^{(j)} \in \mathcal{U}^{(k)}$ such that $\|\hat{\mathbf{u}}^{(k)} - \mathbf{u}^{(j)}\| > 2\Delta\mathbf{u}$, set $\hat{\mathbf{u}}^{(k)} = (\hat{\mathbf{u}}^{(k)} + \mathbf{u}^{(j)})/2$. This step improves the QA, since the new point $\hat{\mathbf{u}}^{(k)}$ will replace the more distant point $\mathbf{u}^{(j)}$ in the following regression set.

5. Evaluate the process at $\hat{\mathbf{u}}^{(k)}$ to acquire $J_p(\hat{\mathbf{u}}^{(k)})$ and $\mathbf{G}_p(\hat{\mathbf{u}}^{(k)})$. Define the next iterate by

   (a) Successful iteration. If $J_p(\hat{\mathbf{u}}^{(k)}) < J_p(\mathbf{u}^{(k)})$, define $\mathbf{u}^{(k+1)} = \hat{\mathbf{u}}^{(k)}$ and run the screening algorithm to define the regression set $\mathcal{U}^{(k+1)}$. Calculate the quality indices as:

$$\rho_m^{(k)} = \max\left\{ \left|1 - \frac{J_{ad}^{(k)} - J_{ad}^{(k-1)}}{J_p^{(k)} - J_p^{(k-1)}}\right|, \left|1 - \frac{G_{ad,1}^{(k)} - G_{ad,1}^{(k-1)}}{G_{p,1}^{(k)} - G_{p,1}^{(k-1)}}\right|, \cdots, \left|1 - \frac{G_{ad,n_g}^{(k)} - G_{ad,n_g}^{(k-1)}}{G_{p,n_g}^{(k)} - G_{p,n_g}^{(k-1)}}\right| \right\} \tag{2.16}$$

$$\rho_\phi^{(k)} = \max\left\{ \left|1 - \frac{J_{\phi}^{(k)} - J_{\phi}^{(k-1)}}{J_p^{(k)} - J_p^{(k-1)}}\right|, \left|1 - \frac{G_{\phi,1}^{(k)} - G_{\phi,1}^{(k-1)}}{G_{p,1}^{(k)} - G_{p,1}^{(k-1)}}\right|, \cdots, \left|1 - \frac{G_{\phi,n_g}^{(k)} - G_{\phi,n_g}^{(k-1)}}{G_{p,n_g}^{(k)} - G_{p,n_g}^{(k-1)}}\right| \right\} \tag{2.17}$$

where J_{ad} and $\mathbf{G}_{ad}$ refer to the adapted objective function and constraints, as shown in problem 2.5. Increase k by one and go to Step 2.

(b) Unsuccessful iteration. If $J_p(\hat{\mathbf{u}}^{(k)}) \geq J_p(\mathbf{u}^{(k)})$, run the screening algorithm to update the regression set for $\mathbf{u}^{(k)}$ with the new probe $\hat{\mathbf{u}}^{(k)}$ in the collected data. Due to the screening algorithm, $\hat{\mathbf{u}}^{(k)}$ will be included in the regression set to improve the QA around $\mathbf{u}^{(k)}$. Go to Step 2.

2.3.3. Gradients from transient information

Regardless of the method used for estimating the plant gradients, the major bottleneck in static RTO is waiting for steady-state before collecting the required measurements; especially in slow processes. If measurements obtained during transient operation are available, they can be used to accelerate the optimization scheme by reducing the required waiting time. Another less evident advantage of using dynamic information, is that the violation of constraints can be monitored, and corrective actions can be taken in real time. This section summarizes the main contributions in the field of using transient data in RTO; special attention is given to the use of dynamic model identification, and the use of Neighboring-Extremals (NE).

Dynamic model identification: The idea of using model identification for static optimization goes back to the work of Bamberger and Isermann (1978). They proposed using the transient information of the process to estimate the steady-state gradients, by using an adaptive Hammerstein model to describe the dynamic behavior of the system; resulting in a significant reduction in the required time to achieve optimality. Zhang and Roberts (1990) extended this idea by using linear dynamic models, and placing the scheme inside the ISOPE framework, but did not consider process dependent constraints. Figure 2.2 illustrates how model identification is integrated in RTO.

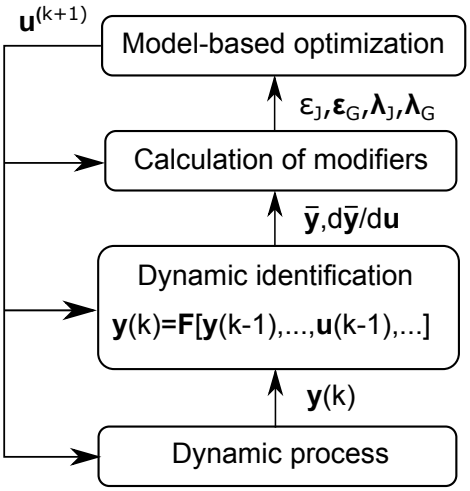


Figure 2.2.: Integration of system identification with RTO. Adapted from Zhang and Roberts (1990)

Consider, for simplicity, the use of a linear I/O dynamic model to illustrate this idea:

$$\mathbf{y}(k) = \mathbf{A}(q^{-1})\mathbf{y}(k) + q^{-d}\mathbf{B}(q^{-1})\mathbf{u}(k) + \mathbf{D} \tag{2.18}$$

where $\mathbf{A}(q^{-1})$ and $\mathbf{B}(q^{-1})$ are matrix polynomials in the backward-shift operator q^{-1}, d is the minimum delay (in sampling periods) between the inputs $\mathbf{u}(k)$ and the outputs $\mathbf{y}(k)$[3] at sample point k, and $\mathbf{D}$ is a vector of constant terms. This model is suitable for describing the dynamics of different well-behaved processes when linearization is valid, i.e. close to steady-state. If the system is stable, its final value can be calculated by setting $q = 1$; from the estimated steady-state, the gradients can be directly obtained:

$$\begin{aligned} \bar{\mathbf{y}} &= [\mathbf{I} - \mathbf{A}(1)]^{-1} [\mathbf{B}(1)\mathbf{u}_s + \mathbf{D}] \\ \frac{\partial \bar{\mathbf{y}}}{\partial \mathbf{u}} &= [\mathbf{I} - \mathbf{A}(1)]^{-1} \mathbf{B}(1) \end{aligned} \tag{2.19}$$

The idea can be extended to use nonlinear models such as a generalized Hammerstein model, which includes nonlinear terms in the inputs. A drawback of this method is that the dynamic order of the models has to be pre-specified, and some experimentation might be required for the proper selection of the

[3]The notation $\mathbf{y}(k)$ and $\mathbf{y}_k$ are both used in this work to denote data sampled at point k.

dynamic structure of the model. Furthermore, test signals such as a Pseudo-Random Binary Sequence (PRBS) must usually be superimposed on the inputs so that enough excitation is present for parameter estimation (persistency of excitation); especially for nonlinear processes (Nelles, 2001). This continuous perturbation of the plant, although useful in simulation studies, is undesirable in practice; especially when the plant is operating close to its optimum.

Lin et al. (1989) used stochastic-systems theory to avoid applying additional perturbations to the plant, by realizing that disturbances on the process affect the inputs to the system via feedback control, which can provide enough excitation for model identification. However, the assumption that there is a feedback loop between the measured outputs and the inputs is not always valid.

Instead of estimating the gradients from transient data, Gao and Engell (2016) proposed using a linear Autoregressive with Exogenous Inputs (ARX) model to predict the steady-state values of the system, and using these predictions in MAWQA. The main advantage of this approach, is that test signals do not have to be superimposed on the inputs, therefore avoiding dynamic plant disturbances. The idea is based on the fact that for predicting the steady-state, the parameters in $\mathbf{B}$ and $\mathbf{D}$ in eq. 2.19 must not be determined individually. This means that during a step-response corresponding to a set-point change, where the inputs stay constant, one could lump part of eq. 2.18 as $\mathbf{D}' = q^{-d}\mathbf{B}(q^{-1})\mathbf{u}(k) + \mathbf{D}$. This newly defined vector can be identified online after each set-point change to take into account the implicit dependency on the inputs $\mathbf{u}^{(k)}$. The plant gradients would then be determined by the QA. A drawback of this approach is that the dynamics of the plant must be approximately linear for the identified model to be valid, which can sometimes require waiting for the plant to be close to the steady-state.

Neighboring extremals: Other approaches for the use of transient information are inspired by gradient control schemes, and include the techniques classified as direct input adaptation in Section 2.2.1. François et al. (2012) present these schemes in more detail, and compare them in the framework of RTO.

François and Bonvin (2014) used the gradients estimated from NE in the framework of MA, and showed that a remarkable improvement in convergence

time could be achieved, even reaching the optimum in one settling time of the analyzed plant. The idea behind NE is that under the assumption of parametric uncertainty, the deviations between the gradients of the model and the plant can be attributed to differences in the parameters of the model, which must be compensated by changing the plant inputs. Originally, NE proposes to perform a variational analysis around the nominal operating point given by solving the original unconstrained optimization problem with the nominal parameter values $\boldsymbol{\theta}_m$, in which case $\partial J(\mathbf{u}_m^*, \boldsymbol{\theta}_m)/\partial \mathbf{u} = 0$ (Gros et al., 2009).

Given a static I/O mapping of the process $\bar{\mathbf{y}} = \mathbf{H}(\mathbf{u}, \boldsymbol{\theta})$, the variation of the steady-state outputs with respect to the nominal operating point can be expressed as:

$$\Delta\bar{\mathbf{y}} = \frac{\partial \mathbf{H}}{\partial \mathbf{u}}\Delta\mathbf{u} + \frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}}\Delta\boldsymbol{\theta} \tag{2.20}$$

with $\Delta\bar{\mathbf{y}} = \bar{\mathbf{y}}_p - \bar{\mathbf{y}}_m^*$, $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}_m^*$ and $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_p - \boldsymbol{\theta}_m$, where $\bar{\mathbf{y}}_m^* = \mathbf{H}(\mathbf{u}_m^*, \boldsymbol{\theta}_p)$. If the number of measurements is at least equal to the number of uncertain parameters ($n_y \leq n_\theta$), eq. 2.20 forms a system of (over)determined linear equations that can be solved for the parametric uncertainty $\Delta\boldsymbol{\theta}$ as:

$$\Delta\boldsymbol{\theta} = \left(\frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}}\right)^{+} \left(\Delta\bar{\mathbf{y}} - \frac{\partial \mathbf{H}}{\partial \mathbf{u}}\Delta\mathbf{u}\right) \tag{2.21}$$

where $()^+$ denotes the Moore-Penrose pseudo-inverse of $()$, which solves the equation in the LS sense. The gradient of the plant can be similarly obtained as:

$$\frac{\partial J_p(\mathbf{u}, \boldsymbol{\theta})}{\partial \mathbf{u}} = \frac{\partial^2 J_p}{\partial \mathbf{u}^2}\Delta\mathbf{u} + \frac{\partial^2 J_p}{\partial \mathbf{u}\partial \boldsymbol{\theta}}\Delta\boldsymbol{\theta} \tag{2.22}$$

which, by substituting the parametric uncertainty $\Delta\boldsymbol{\theta}$ obtained from eq. 2.21, gives an estimate of the steady-state gradient based on plant measurements. In the constrained case, the gradient of the objective function at the optimum is not necessarily equal to zero, and therefore eq. 2.22 becomes:

$$\frac{\partial J_p(\mathbf{u}, \boldsymbol{\theta})}{\partial \mathbf{u}} = \frac{\partial J_p(\mathbf{u}_m^*, \boldsymbol{\theta}_m)}{\partial \mathbf{u}} + \frac{\partial^2 J_p}{\partial \mathbf{u}^2}\Delta\mathbf{u} + \frac{\partial^2 J_p}{\partial \mathbf{u}\partial \boldsymbol{\theta}}\Delta\boldsymbol{\theta} \tag{2.23}$$

Equation 2.23 can also be used to estimate the gradients of the constraints by replacing J_p with $\mathbf{G}_p$.

In order to use the NE approach during transient operation, François and Bonvin (2014) proposed substituting the static differences in eq. 2.20 with transient differences $\Delta\mathbf{y}(t) = \mathbf{y}_p(t) - \bar{\mathbf{y}}_m^*$ and $\Delta\mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_m^*$. Note that when steady-state is reached, this approach reduces to the static NE formulation. A strong disadvantage of the NE approach in general, is that its accuracy is limited by the linearization of the mapping function $\bar{\mathbf{y}} = \mathbf{H}(\mathbf{u}, \boldsymbol{\theta})$. This can be improved by re-linearizing $\bar{\mathbf{y}}$ around $\mathbf{u}^{(k)}$, which implies updating the matrices in equations 2.21 and 2.22 (François and Bonvin, 2014)

CHAPTER

THREE

DETERMINING THE TYPE OF PLANT-MODEL MISMATCH

As discussed in Chapter 2, the strategies for using transient information in the context of RTO can also be classified depending on the type of plant-model mismatch assumed (i.e. structural or parametric). This chapter presents a simple approach to discard parametric mismatch: first, the concept of P-invalidity and its relation with the χ^2 distribution is introduced; this is used to analyze whether the plant-model mismatch can be considered parametric, depending on a simple goodness-of-fit test. The concept is illustrated with the case-study of a CSTR.

3.1. Rejecting parametric mismatch

Parametric plant-model mismatch results from using a model that has a correct mathematical structure, but contains the wrong value in one, or more parameters. Correcting such a model involves parameter estimation, and requires knowing the identity of the mismatched parameters. Determining whether a proposed dynamic model is structurally correct in the presence of measurement noise is of high use, since simple techniques based on the assumption of parametric uncertainty exist to exploit transient information in the context of RTO.

The general framework for parameter estimation is the maximum likelihood formulation, which can be written as follows if the measurement errors are assumed to be independent from each other:

$$\begin{aligned} \max_{\hat{\boldsymbol{\theta}},\hat{\mathbf{e}},\hat{\mathbf{y}}} \quad & \prod_{i=1}^{n} p(\hat{e}_i) \\ \text{s.t.} \quad & \hat{e}_i = y_i - \hat{y}_i \\ & \hat{y}_i = f(\hat{\boldsymbol{\theta}}) \end{aligned} \tag{3.1}$$

where $\hat{\boldsymbol{\theta}}$ are the parameters to be estimated, $\hat{\mathbf{e}}$ is the error vector between the measured data $\mathbf{y}$ and the predicted data $\hat{\mathbf{y}}$ from the model $f(\hat{\boldsymbol{\theta}})$, and p is the probability density function (PDF) of the error, which is assumed to be known. Problem 3.1 is usually transformed by instead minimizing the logarithm of the likelihood, which in the case of Gaussian noise leads to the widely known LS problem.

Testing whether a dynamic model is structurally correct can be rephrased as testing how well a best-fit version of that model can represent real data: a poor fit means that the mathematical formulation of the model is, even in the case of optimal parameters, unable to properly represent the underlying phenomena that generate the measured data. Unless the real model is known, or the whole input space is explored to compare the model and the data, it is not possible to say with absolute certainty that a proposed dynamic model is structurally correct. However, it is possible to work from the other end and reject the assumption of parametric mismatch with a certain probability. This is a hypothesis test, where the null hypothesis is that the model is structurally correct, and the alternate hypothesis is that it is not.

Bunin et al. (2013) proposed the concept of P-invalidity in the context of determining the validity of a proposed regularizing structure for the estimation of bounds on experimental gradients. The idea behind P-invalidity testing, is that *if the resulting residuals after a best-fit procedure of data to a specified model are very unlikely, then it is probable that the model is not structurally correct.* Quantifying whether a model structure is P-invalid can be done by

defining a likelihood value L_p that serves as a lower bound, with probability P, on the likelihood of the estimated errors:

$$\text{prob}\left(\prod_{i=1}^{n} p(\hat{e}_i) \geq L_p\right) = P \tag{3.2}$$

A model is said to be P-invalid if there is no solution for problem 3.1 that results in a likelihood higher than L_p. Of course, if the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ is P-invalid, every other solution will also be invalid. Bunin et al. (2013) proposed using Monte Carlo sampling to estimate L_p, since calculating it analytically implies integrating the likelihood function, which is not possible in the general case. For the particular case of Gaussian error with variance deviation σ_n^2, however, one can reformulate the P-invalidity criterion using the fact that maximizing the likelihood function corresponds to minimizing the sum of squared-errors:

$$\text{prob}\left(\sum_{i=1}^{n} \hat{e}_i^2 \leq E_p\right) = P \tag{3.3}$$

where the relation between L_p and E_p is not trivial, but is unnecessary, since this is now a different criterion for P-invalidity. In simple terms, this equation means that if the sum of the squared residuals after finding the best fit is very large, then it is unlikely that the model is structurally correct.

The advantage of this formulation, is that the sum of squared errors of a Gaussian distribution follows a Chi-squared (χ^2) distribution. The null hypothesis that the model is structurally correct is rejected if $T > \chi^2_{1-\alpha,k}$, with the test statistic $T = \sum_{i=1}^{n} \hat{e}_i^2/\sigma_n^2$, and the critical value $\chi^2_{1-\alpha,k}$ calculated from the χ^2 distribution with significance α, and $k = n - n_\theta$ degrees of freedom. The degrees of freedom are reduced to take into account the fitting of n_θ parameters; in the case of dynamic system identification, an additional degree of freedom should be subtracted for every initial condition that is taken as fixed. A possible shortcoming of this approach, is that the variance of the noise σ_n^2 must be known, which is not always the case. Figure 3.1 illustrates this procedure; although not shown, a data collection step is of course necessary.

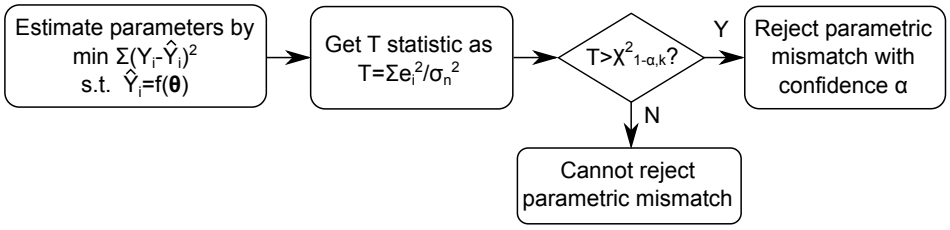


Figure 3.1.: Statistical test for rejecting the hypothesis of parametric mismatch

3.2. Case study 1

This section presents one of the two case-studies used in this work. The model describing the system is first presented, and then used to illustrate the usefulness of the statistical test introduced in Section 3.1.

3.2.1. Reactor model

The case-study is based on an isothermal CSTR, where the reactions A+B $\xrightarrow{k_1}$ C and 2B $\xrightarrow{k_2}$ D occur; with C being the desired product (see Figure 3.2). The manipulated variables are the feed rates of A and B (u_A and u_B respectively). The model consists of mass and energy balances (François and Bonvin, 2014):

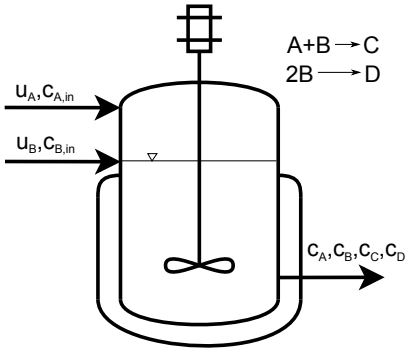


Figure 3.2.: Diagram of the CSTR for case-study 1

$$
\begin{aligned}
\dot{c}_A(t) &= -k_1 c_A(t) c_B(t) + \frac{u_A(t)}{V} c_{A,\text{in}} - \left(\frac{u_A(t) + u_B(t)}{V} \right) c_A(t) \\
\dot{c}_B(t) &= -k_1 c_A(t) c_B(t) - 2k_2 c_B^2(t) + \frac{u_B(t)}{V} c_{B,\text{in}} - \left(\frac{u_A(t) + u_B(t)}{V} \right) c_B(t) \\
\dot{c}_C(t) &= k_1 c_A(t) c_B(t) - \left(\frac{u_A(t) + u_B(t)}{V} \right) c_C(t) \\
\dot{c}_D(t) &= k_2 c_B^2(t) - \left(\frac{u_A(t) + u_B(t)}{V} \right) c_D(t) \\
Q(t) &= V k_1 c_A(t) c_B(t) (-\Delta H_{r,1}) + V k_2 c_B^2(t) (-\Delta H_{r,2})
\end{aligned}
\tag{3.4}
$$

where c_i denotes the concentration of species i, V is the constant reactor volume, Q is the generated heat, $\Delta H_{r,1}$ and $\Delta H_{r,2}$ are the reaction enthalpies of the two reactions, $c_{A,\text{in}}$ and $c_{B,\text{in}}$ are the inlet concentrations, and k_1 and k_2 are the rate constants of the reactions. The only available measurements are the concentrations at the output of the reactor, which are assumed to be obtained with a sampling time $t_s = 1$ min. Table 3.1 shows the numerical values of these parameters as used in the nominal model.

Table 3.1.: Nominal model parameters for the CSTR case-study

k_1	0.75	L/(mol min)
k_2	1.5	L/(mol min)
$c_{A,\text{in}}$	2	mol/L
$c_{B,\text{in}}$	1.5	mol/L
V	500	L
$(-\Delta H_{r,1})$	3.5	kcal/mol
$(-\Delta H_{r,2})$	1.5	kcal/mol

3.2.2. Evaluation of plant-model mismatch

The χ^2 test discussed in Section 3.1 was used to check whether the plant-model mismatch in the case-study is parametric or structural. The data used for parameter estimation was the one corresponding to the start-up of the reactor: the initial concentrations c_A, c_B, c_C, and c_D are assumed to be zero, and the flow rates are set to their nominal optimal value (c.f. Table 4.2). The

parameters to be estimated correspond to the two reaction constants and the inlet concentration $c_{A,\text{in}}$. The data was corrupted with zero-mean Gaussian noise with three different standard deviations; all of the four measurements were corrupted with the same level of noise. The test was repeated one hundred times, each experiment with a different noise realization, in order to verify the number of times that the test rejected parametric mismatch.

Table 3.2.: Determination of P-Invalidity for the CSTR case-study

σ_n	$1{\cdot}10^{-3}$	$5{\cdot}10^{-3}$	$1{\cdot}10^{-2}$
n_{invalid}	4	3	7

Table 3.2 summarizes the results of this study, where it is shown that the number of false-negatives per noise level, i.e. the number of times that the test discarded parametric mismatch, is consistent with the significance level used ($\alpha = 0.05$). Although not shown here, the test was also repeated without indicating the whole set of mismatched parameters; in this case, the test always rejected parametric mismatch. This is one of the main difficulties when dealing with the assumption of parametric mismatch: if the model is indeed structurally correct, failure to identify the set of mismatched parameters will result in a deficient model fit, which can be interpreted as a possible structural mismatch. Although the test presented in this section can be useful for determining structural mismatch, Type I and Type II errors cannot be avoided, and its results should be interpreted with caution.

In Chapter 4, parametric plant-model mismatch will be addressed in the context of RTO, where a simple, yet efficient, approach for fast parameter estimation based on linearization will be presented. Chapter 5 deals with the more general (and common) case of structural mismatch, where the use of black-box dynamic models in RTO is discussed.

CHAPTER

FOUR

AN IMPROVED RTO SCHEME FOR SLOW PROCESSES WITH PARAMETRIC MODEL UNCERTAINTIES

If parametric mismatch cannot be rejected, several RTO techniques inspired in implicit control can be used. As mentioned in Chapter 2, a recently proposed approach is the extension of the NE scheme to use transient information (François and Bonvin, 2014). However, the use of transient measurements in this context implies that the estimated parametric mismatch $\Delta\boldsymbol{\theta}$ depends on the trajectory of the outputs $\mathbf{y}_p(t)$. Therefore, different initial conditions will lead to different gradient estimates, i.e. the plant must approach steady-state so that the estimated gradients are close to their static counterparts. The original NE scheme does not make full use of the dynamic information present, since it only uses the available dynamic model in its static form.

This chapter builds on the work of François and Bonvin (2014) on NE update, by applying linearization to the *dynamic* process model, instead of the *static* one. First, the method of parameter estimation based on the linearized dynamic model (LDM) is discussed and evaluated with a toy example. This technique is integrated with MA, and is applied to the optimization of a CSTR.

4.1. Parameter estimation from linearized dynamic model

For the derivation of this scheme, the dynamic behavior of the plant is assumed to be described by the following state-space representation:

$$\begin{aligned} \dot{\mathbf{x}}_p(t) &= \mathbf{f}_p(\mathbf{x}_p(t), \mathbf{u}_p(t), \boldsymbol{\theta}_p) \\ \mathbf{y}_p(t) &= \mathbf{C}(\mathbf{x}_p(t)) \end{aligned} \tag{4.1}$$

where $\mathbf{x}_p(t)$ is a vector of time-varying plant states, $\mathbf{u}_p(t)$ is a vector of plant inputs, $\boldsymbol{\theta}_p$ is a vector of the true plant parameters, $\mathbf{f}_p$ is the dynamic mapping of the plant, $\mathbf{C}$ is a known measurement function, and $\mathbf{y}_p(t)$ is a vector of measured plant outputs. The nominal (mismatched) model is assumed to be represented in the same way, with the subindex m instead of p. If full-state measurement is assumed, then $\mathbf{y}_p(t) = \mathbf{x}_p(t)$. For brevity, the temporal dependency of the different quantities in equation 4.1 will not be explicitly written.

If plant-model mismatch is parametric, the mathematical form of the functions $\mathbf{f}_p$ and $\mathbf{f}_m$ is the same, and their predictions differ only due to a difference in the parameters $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_m$; which means that $\mathbf{f}_p = \mathbf{f}(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}_p)$ and $\mathbf{f}_m = \mathbf{f}(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}_m)$. If the dynamic mapping is assumed to be differentiable with respect to $\mathbf{y}$, $\boldsymbol{\theta}$, and $\mathbf{u}$, the degree to which the dynamic responses predicted by the model differ from the ones observed in the plant can be quantified with a sensitivity analysis:

$$\mathbf{f}(\mathbf{y}_p, \boldsymbol{\theta}_p) = \mathbf{f}(\mathbf{y}_m, \boldsymbol{\theta}_m) + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} + \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \Delta\mathbf{y} \tag{4.2}$$

where $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_p - \boldsymbol{\theta}_m$ and $\Delta\mathbf{y} = \mathbf{y}_p - \mathbf{y}_m$. The dependency on the inputs $\mathbf{u}$ is dropped, since the linearization is done assuming the same inputs for the plant and for the model, i.e. $\mathbf{u}_p = \mathbf{u}_m = \mathbf{u}$.

In order to isolate the effect of the parametric mismatch on the predicted dynamic response of the model, one can force $\mathbf{y}_m$ to be equal to $\mathbf{y}_p$ at every sampling point. This idea is illustrated in Figure 4.1, where $\dot{\mathbf{y}}_p = \mathbf{f}(\mathbf{y}_p, \mathbf{u}, \boldsymbol{\theta}_p)$ is the *true* dynamic response of the plant, and corresponds to the slope of the red lines, whereas the dynamic response $\dot{\mathbf{y}}_m$ *predicted* by the model $\mathbf{f}(\mathbf{y}_p, \mathbf{u}, \boldsymbol{\theta}_m)$

based on the nominal parameters and the *true* plant outputs $\mathbf{y}_p$ corresponds to the slope of the blue lines.

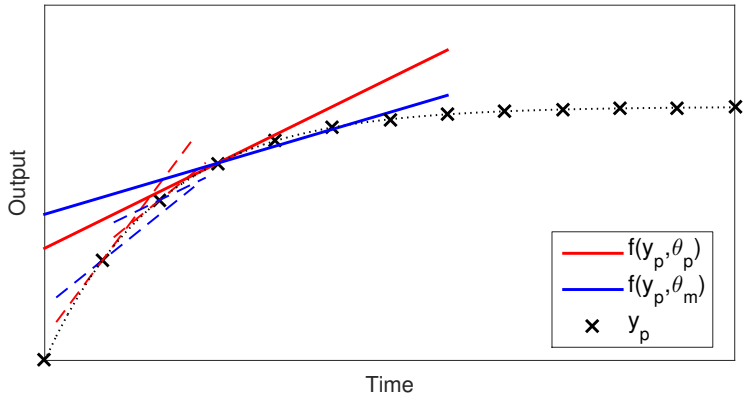


Figure 4.1.: Temporal derivatives predicted by the model (blue) and observed in the plant (red). The black icons represent the sampled data from the plant

With this in mind, the difference between these two slopes is exclusively due to parametric mismatch. Assuming that the number of measurements is larger or equal to the number of uncertain parameters ($n_\theta \leq n_y$), eq. 4.2 can be reorganized to obtain an estimate of the parametric mismatch $\Delta\boldsymbol{\theta}$ as:

$$\Delta\boldsymbol{\theta} = \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}\right)^{+} \left[\dot{\mathbf{y}}_p - \dot{\mathbf{y}}_m\right] \tag{4.3}$$

In general, the estimated parametric mismatch will change at each sampling point due to the accuracy of linearization; the exception being the case where $\mathbf{f}$ is linear with respect to the mismatched parameters, where eq. 4.3 is exact. Therefore, it is more appropriate to write eq. 4.3 for every sampling point k as:

$$\Delta\boldsymbol{\theta}_k = \left(\left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}\right|_{\mathbf{y}_{p,k},\mathbf{u},\boldsymbol{\theta}_m}\right)^{+} \left[\dot{\mathbf{y}}_{p,k} - \dot{\mathbf{y}}_{m,k}\right] \tag{4.4}$$

Equation 4.4 makes full use of the transient data of the process. However, this improvement comes with a higher computational load, since the sensitivity $\partial\mathbf{f}/\partial\boldsymbol{\theta}$ must be calculated at each sampling point. This scheme can even work if the plant settles to steady-state (i.e. $\dot{\mathbf{y}}_p = 0$), because as long as there is a parametric mismatch, the model will not have the same steady-state as the plant.

The main difficulty in this approach is to estimate the temporal derivative of the plant $\dot{\mathbf{y}}_p = \mathbf{f}(\mathbf{y}_p, \boldsymbol{\theta}_p)$, especially when there is measurement noise. In this work, a Savitzky-Golay (SG) filter of order 3 and window size of 5 points was used to estimate these temporal derivatives (Savitzky and Golay, 1964). This type of filter is widely used for smoothing data and estimating its derivatives. An SG filter works by fitting a polynomial of degree l to m data points; from the fitted polynomial, the derivatives can be easily extracted. A smoothed version of the data is also obtained:

$$\begin{aligned} \mathbf{y}_k &\approx \frac{1}{35}\left(-3\mathbf{y}_{k-2} + 12\mathbf{y}_{k-1} + 17\mathbf{y}_k + 12\mathbf{y}_{k+1} - 3\mathbf{y}_{k+2}\right) \\ \dot{\mathbf{y}}_k &\approx \frac{1}{12t_s}\left(\mathbf{y}_{k-2} - 8\mathbf{y}_{k-1} + 8\mathbf{y}_{k+1} - \mathbf{y}_{k+2}\right) \end{aligned} \tag{4.5}$$

Using a larger window size results in more smoothing. However, if the window is too large, the bias error will increase, since the chosen polynomial might only approximate the data locally. Since the filter uses points $k+1$ and $k+2$ for the estimation of the derivatives at point k, the estimation is always delayed by two sampling times. If the mismatched parameters are assumed to not change significantly during each transient period, the results of eq. 4.4 can also be filtered to improve the accuracy of estimation:

$$\Delta\hat{\boldsymbol{\theta}}_k = \frac{(k-1)\Delta\hat{\boldsymbol{\theta}}_{k-1} + \Delta\boldsymbol{\theta}_k}{k} \tag{4.6}$$

with $\Delta\hat{\boldsymbol{\theta}}_1 = \Delta\boldsymbol{\theta}_1$, where $\Delta\hat{\boldsymbol{\theta}}_k$ is the filtered running estimate of the parametric mismatch at sampling point k, and $\Delta\boldsymbol{\theta}_k$ is the parametric mismatch at sampling point k from eq. 4.4. The filter can be reset after each new set-point is applied, in order to take into account the possible effect of different operating conditions on the parameters, and to avoid using old information if slow drifts are expected (e.g. catalyst deactivation, heat exchanger fouling).

4.1.1. Toy example

Consider the following toy example used to compare the proposed parameter estimation approach, with the parameters estimated from NE. This example is a simplified version of the case-study presented in Chapter 2, and consists of a

CSTR where one reaction takes place, namely the decomposition of substance A. The reactor operates isothermally with a constant volume, and can be modeled by the following ODE:

$$\dot{c}_A(t) = -kc_A^2(t) + \frac{u_A(t)}{V}\left(c_{A,\text{in}} - c_A(t)\right) \tag{4.7}$$

where c_A is the concentration of species A in the reactor, k is the reaction constant, u_A is the feed rate of component A, V=500 L is the volume of the reactor, and $c_{A,\text{in}}$=2.5 mol/L. The reaction constant k is the mismatched parameter, with a nominal value $k_{\text{nom}} = 0.05$ L/(mol·min), and a true value $k_{\text{true}} = 0.1$ L/(mol· min).

In order to verify how well the NE approach estimates the mismatched parameter k, different step-response tests were done starting from the steady-state corresponding to a nominal input $u_A^* = 1$ L/min. The input was varied in increments of 0.1 from 1.1 to 3 L/min. Three different approaches were used to estimate the mismatched parameter: NE with linearization around the nominal point, NE with linearization around the operating point, and LDM around the trajectory of the real data (eq. 4.4).

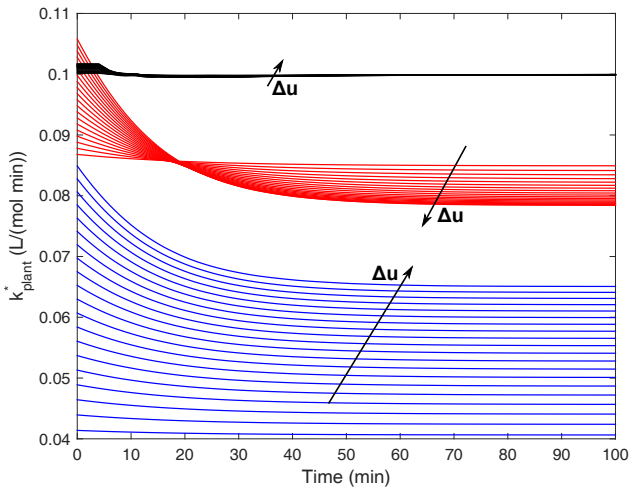
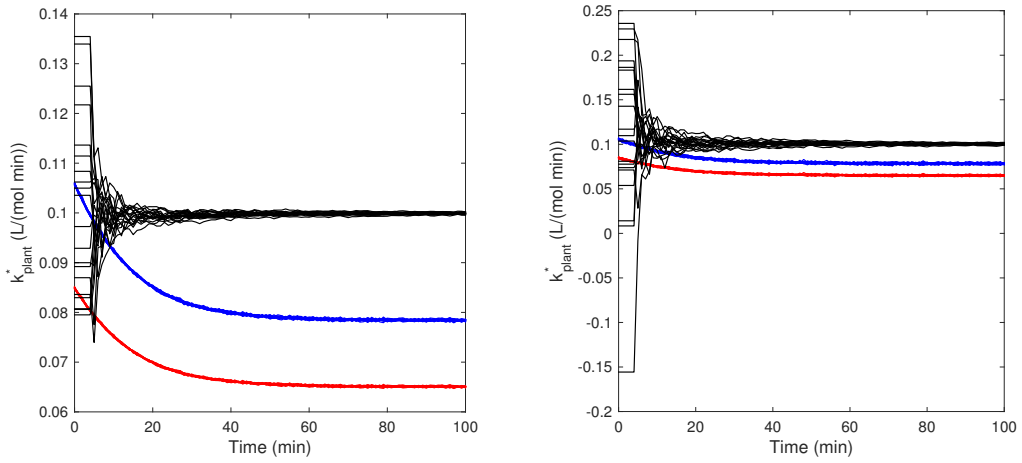


Figure 4.2.: Estimated reaction constant for the toy example with the linearized dynamic model (black), NE (blue), and NE with updated matrices (red). The arrow indicates the order of the curves according to the step-size Δu

Figure 4.2 illustrates the results of the test, where it is evident that the estimated parameters based on NE follow the dynamics of the plant. The estimates based on NE with linearization around the current operating point (updated matrices) are closer to the true value k_{true}, than the estimates based on NE linearized around the nominal point. This is expected, since updating the model matrices reduces the error of linearizing in the **u** direction. However, the estimates based on LDM vary little with the size of the step change, and converge to a value very close to the true one. Even though the ODE is linear on the mismatched parameter, exact estimation is not achieved because the exact temporal derivatives are not known, but approximated with the SG filter.

In order to verify how these three approaches perform with noisy measurements, a step-response from the steady-state of the nominal operating condition, up to $u = 3$ L/min was performed 20 times for two different noise variances σ_n^2. Figure 4.3 illustrates the results of this test.



(a) $\sigma_n^2 = 10^{-6}$ **(b)** $\sigma_n^2 = 2.5 \cdot 10^{-5}$

Figure 4.3.: Estimated reaction constant with noisy measurements for the toy example with the linearized dynamic model (black), NE (blue), and NE with updated matrices (red).

As expected, LDM is more sensitive to measurement noise than the approaches based on NE, since it requires numerical differentiation of the data. However, even with a high level of measurement error the estimate from LDM converges to a neighborhood of the true parameter, showing how the simple filter of eq 4.6 is effective enough to compensate for noise propagation.

4.2. RTO with parameters from LDM

From the assumption of parametric uncertainty, it is clear that the main advantage of the NE scheme is that the parameters of the plant are easily estimated via linearization. However, using linearization for the calculation of the plant gradients is not necessarily advantageous, since the same assumption also means that $\mathbf{H}_p(\mathbf{u}, \boldsymbol{\theta}_p) \approx \mathbf{H}_m(\mathbf{u}, \boldsymbol{\theta}_m + \Delta\boldsymbol{\theta})$ and therefore $\frac{\partial J_p(\mathbf{u},\boldsymbol{\theta}_p)}{\partial \mathbf{u}} \approx \frac{\partial J_m(\mathbf{u},\boldsymbol{\theta}_m+\Delta\boldsymbol{\theta})}{\partial \mathbf{u}}$; the same logic applies to the constraints and their gradients. Figure 4.4 illustrates an RTO scheme based on parameter estimation with LDM.

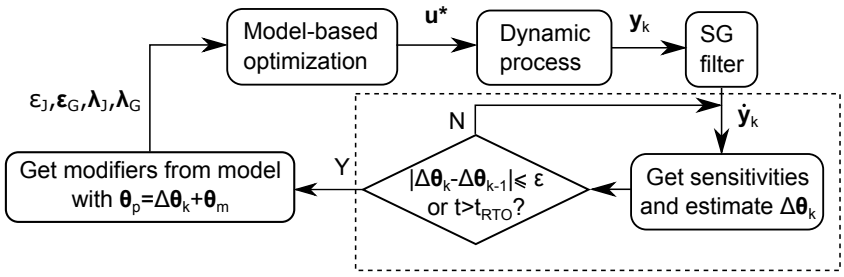


Figure 4.4.: RTO scheme with parameters from the linearized dynamic model

This corresponds to a temporal decomposition between parameter estimation (in the dotted box), and optimization. The RTO algorithm with parameters from LDM can be summarized in the following steps:

1. Collect the plant measurements $\mathbf{y}_{p,k}$ at sampling point k, and filter them with the SG filter to get an estimate of the *true* temporal derivative of the plant $\dot{\mathbf{y}}_{p,k}$.
2. With the filtered $\mathbf{y}_{p,k}$, obtain the temporal derivative *predicted* by the nominal model $\dot{\mathbf{y}}_{m,k} = \mathbf{f}(\mathbf{y}_{p,k}, \mathbf{u}, \boldsymbol{\theta}_m)$.
3. Get the sensitivity matrix $\partial\mathbf{f}/\partial\boldsymbol{\theta}$ evaluated at $\mathbf{y}_{p,k}$, $\mathbf{u}$, and $\boldsymbol{\theta}_m$. Estimate the parametric mismatch $\Delta\boldsymbol{\theta}_k$ with eq. 4.4.
4. Filter $\Delta\boldsymbol{\theta}_k$ with eq. 4.6. If the re-optimization criteria are not met, go to Step 1 to collect another measurement; else, continue.
5. Get the modifiers by calculating the gradients as $\frac{\partial J_p(\mathbf{u},\boldsymbol{\theta}_p)}{\partial \mathbf{u}} = \frac{\partial J_m(\mathbf{u},\boldsymbol{\theta}_m+\Delta\boldsymbol{\theta})^*}{\partial \mathbf{u}}$, where $\Delta\boldsymbol{\theta}^*$ is the obtained parametric mismatch from the inner estimation loop.

6. Run the adapted optimization problem of IGMO with the calculated modifiers. Apply the calculated new set-point to the plant, and go to Step 1.

There are two criteria for re-optimization, namely parameter convergence (a tolerance of 0.1% was used in this work) and a maximum optimization interval t_{RTO}. The choice of t_{RTO} is critical under noisy conditions, since it determines the amount of filtering the estimated parameters go through before re-optimization, as will be shown in the following section.

4.3. Case study 1

In this section, the proposed RTO scheme is evaluated with the optimization of the CSTR presented in Chapter 3. The optimization problem is initially stated, and followed by simulation studies.

4.3.1. Optimization problem

The optimization problem aims at maximizing the performance of the reactor at steady-state, while satisfying process dependent constraints:

$$\begin{aligned} \max_{u_A,u_B} \quad & J := \frac{c_C^2(u_A+u_B)^2}{u_A c_{A,\mathrm{in}}} - w(u_A^2+u_B^2) \\ \text{s.t.} \quad & G_1 := \frac{Q}{Q_{\max}} - 1 \leq 0 \\ & G_2 := \frac{D}{D_{\max}} - 1 \leq 0 \\ & 0 \leq u_A \leq u_{\max} \\ & 0 \leq u_B \leq u_{\max} \end{aligned} \tag{4.8}$$

where G_1 constrains the amount of heat generated, while G_2 imposes a limit on the amount of the by-product D, where $D = c_D/(c_A + c_B + c_C + c_D)$ is the molar fraction of this component in the product stream; the inputs are also bounded. The objective function is based on the yield of product C with respect to reactant A; w is a parameter that penalizes the control action. The parameters for the optimization problem are given in Table 4.1

Table 4.1.: Parameters of the optimization problem for the CSTR case-study

w	0.004	mol min/L^2
$Q_{\max}$	110	kcal
$D_{\max}$	0.1	-
$u_{\max}$	50	L/min

To comply with the general formulation of the static optimization problem, the maximization in problem 4.8 is transformed to $\min -J$. Plant-model mismatch is introduced by using different parameters in the "true" plant, namely: $c_{A,\text{in}} = 2.5$ mol/L, $k_1 = 1.4$ L/(mol·min), and $k_2 = 0.4$ L/(mol·min) (see Table 3.1 for the nominal parameters). Note that the mismatched parameters are also part of the objective function and the first constraint. This means that parameter estimation is required, since the nominal objective function and constraints are also mismatched and not directly measured.

The solutions to the optimization problem for the model and the plant are summarized in Table 4.2. Note that the set of active constraints is different in both cases, with G_2 being active in the nominal (model) solution, and G_1 being active in the true solution. As in Chapter 3, the sampling time for the measurements was taken as $t_s = 1$ min.

Table 4.2.: Solutions to the model and plant optimization problems for the CSTR case-study

model optimal solution			plant optimal solution		
$u^*_{A,m}$	14.52	L/min	$u^*_{A,p}$	17.2	L/min
$u^*_{B,m}$	14.9	L/min	$u^*_{B,p}$	30.3	L/min
$J(\mathbf{u}^*)$	4.51	mol/min	$J(\mathbf{u}^*)$	15.42	mol/min
$G_1(\mathbf{u}^*)$	-0.48	-	$G_1(\mathbf{u}^*)$	0	-
$G_2(\mathbf{u}^*)$	0	-	$G_2(\mathbf{u}^*)$	-0.19	-

4.3.2. Simulations with no measurement noise

Instead of filtering the calculated modifiers as in (Marchetti et al., 2009), the new set-point $\mathbf{u}^{(k+1)^*}$ given by the optimizer was filtered before applying it to the plant, such that $\mathbf{u}^{(k+1)} = (\mathbf{I} - \mathbf{K})\mathbf{u}^{(k)} + \mathbf{K}\mathbf{u}^{(k+1)^*}$; the damping factor was

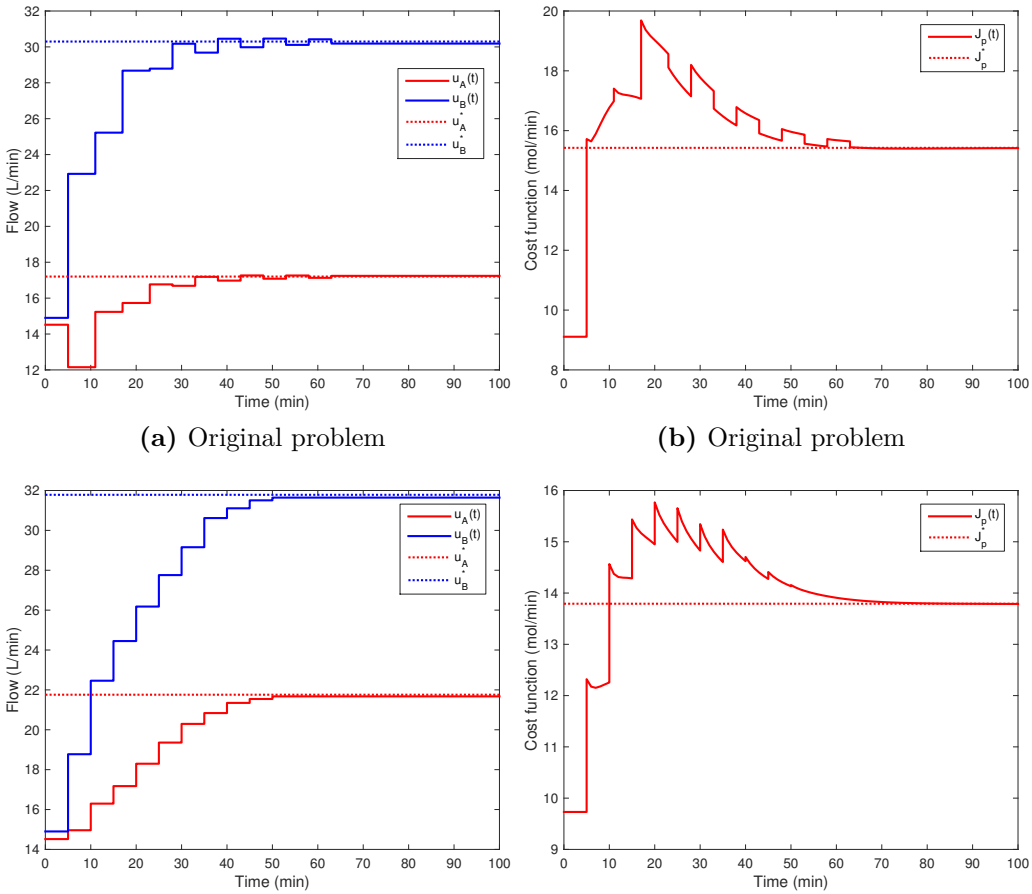
chosen as $\mathbf{K} = 0.5\mathbf{I}$. The estimated mismatch was reset at each new set-point. However, another possibility is to verify how the estimated parameters change when the set-point changes: if they do not vary significantly, then it might be faster to optimize the nominal model with the updated parameters, instead of iteratively using MA.

Figures 4.5a and 4.5b show the results for the optimization of the CSTR problem with LDM, with a maximum optimization period $t_{\text{RTO}} = 5$ min and noise-free measurements. The left figure displays the set-points applied to the plant, where the dotted lines represent the true optimal values, and the right figure displays the transient objective function, with the dotted line representing the true optimal cost function of the plant. The inputs are within $\pm 10\%$ of their optimal values in 30 minutes, and converge to the optimum in around 60 minutes; roughly the setting time of the plant.

Optimization was also performed for the case in which the plant parameter $c_{A,\text{in}} = 2$ mol/L, to verify the effect of varying the degree of plant-model mismatch. Figures 4.5c and 4.5d show the results for this case, where it can be seen that convergence to the true optimum of the plant is also achieved in a period of under 60 minutes.

4.3.3. Simulations with measurement noise

The effect of measurement noise on the optimization procedure was evaluated by using the combinations of three levels of measurement noise σ_n, and three levels of maximum re-optimization intervals t_{RTO}. For each combination, one hundred optimization runs were simulated, each with a maximum of 50 iterations. The results of this study are shown in Table 4.3, and are summarized in terms of: the average Profit Loss (PL) incurred throughout the optimization run, the average time $\bar{t}_{1\%,\text{first}}$ required for the plant to first reach an objective function within $\pm 1\%$ of the true optimum, the average time $\bar{t}_{1\%,\text{last}}$ required for the plant to *stay* within of $\pm 1\%$ of the true optimum, the number of runs n_{first} where $t_{1\%,\text{first}} > 1.5\bar{t}_{1\%,\text{first}}$, and the number of runs n_{last} where $t_{1\%,\text{last}} > 1.5\bar{t}_{1\%,\text{last}}$. The quantities are expressed as their mean value $\pm$ standard deviation.



(a) Original problem **(b)** Original problem

(c) Reduced mismatch **(d)** Reduced mismatch

Figure 4.5.: RTO of the CSTR case-study using linearized dynamic model. Figures (a) and (c) represent the flow rates, while figures (b) and (d) represent the objective function

The profit loss was calculated as the sum of the difference between the optimal cost function J_p^* and the static objective function at set-point $\mathbf{u}(t)$, for the whole optimization run:

$$\mathrm{PL} = \int_0^{t_{\mathrm{sim}}} \left[J_p^* - J_p\left(\mathbf{u}(t)\right) \right] \mathrm{d}t \tag{4.9}$$

From the results, it is evident that there is a trade-off between the speed at which the plant reaches the neighborhood of the optimum, and its robustness to noise: with a short optimization interval of 5 minutes, the plant enters the $\pm 1\%$ region around the optimum in an average time close to, or less than,

Table 4.3.: Optimization of the CSTR case-study with the LDM approach

t_{RTO} (min)	σ_n (mol/L)	PL (mol)	$\bar{t}_{1\%,\text{first}}$ (min)	$\bar{t}_{1\%,\text{last}}$ (min)	n_{first}	n_{last}
5	$1{\cdot}10^{-3}$	68.8 ±23.9	30.3 ± 2.7	36.5 ± 34.3	0	4
	$5{\cdot}10^{-3}$	163.0 ± 94.8	43.1 ± 18.4	319.1 ± 57.6	12	0
	$1{\cdot}10^{-2}$	604.7 ± 810.5	79.2 ± 68.7	355.9 ± 52.7	15	0
10	$1{\cdot}10^{-3}$	123.9 ± 59.2	55.9 ± 0.3	76.1 ± 77.9	0	7
	$5{\cdot}10^{-3}$	125.2 ± 42.4	56.4 ± 6.6	95.2 ± 96.9	0	15
	$1{\cdot}10^{-2}$	172.3 ± 89.8	63.8 ± 15.3	418.0 ± 160.2	6	16
20	$1{\cdot}10^{-3}$	228.3 ± 91.4	105.1 ± 2.1	141.7 ± 149.0	0	6
	$5{\cdot}10^{-3}$	218.9 ± 41.5	105.4 ± 3.6	115.2 ± 81.9	0	2
	$1{\cdot}10^{-2}$	227.5 ± 34.8	104.3 ± 11.4	132.0 ± 113.7	0	7

the setting time of the plant. However, the scheme is strongly affected by an increase in the noise level, as shown by the high standard deviations in convergence time. Note the extremely high standard deviation of the profit loss when the noise level is high; after inspection, it was observed that for some noise realizations the profit loss of the scheme was above 5000 mol. For this particular case-study, the best results are obtained with a t_{RTO} of 10 min, which results in a robust performance in terms of the profit loss.

When using an optimization interval of 10 minutes, the scheme becomes more robust to noise, and reaches the optimal objective function in a time close to the setting time of the plant; even out-performing the cases with $t_{\text{RTO}} = 5$ min. Using a larger optimization interval allows the estimated mismatch to be filtered more, which naturally makes the estimation more robust to noise. The same conclusion can be drawn for the case where the optimization interval is set to 20 minutes, where the plant enters the neighborhood of its optimal objective function at an almost-constant time of around 105 minutes.

However, the scheme can take a longer time to ensure that the objective function actually stays in the neighborhood of the optimum, as shown by the values of $\bar{t}_{1\%,\text{last}}$. The cause for this, is that after initial convergence to the optimum the set-point changes are small, and the dynamics of the plant become obscured by the measurement noise. This leads to wrong parameter estimates if not enough filtering is applied (see the results when using an $t_{\text{RTO}} =$ of 5 min).

Using a larger optimization interval can help with this problem. However, if the interval becomes too large, the dynamics of the plant can also fade *during* the interval, as shown by the results when using a t_{RTO} of 20 minutes with a low level of noise. This will also result in a poor estimation of the parameters.

The results in the last two columns of Table 4.3 explain the high standard deviations obtained: for some noise realizations, the optimizer made drastic moves after initial convergence due to poor parameter estimation, which leads to a longer convergence time, and higher PL. The scheme then converges again, since this large set-point change results in evident dynamics, from which good parameter estimates are obtained. A possibility to improve the robustness and performance of the algorithm, could be to use transient measurements with the LDM in the first optimization iterations to advance quickly towards the optimum, and then switching to a static MA approach. Rodríguez-Blanco et al. (2016) recently proposed such a mixed approach, and showed that it allows for quick convergence, even with structural plant-model mismatch, since transient information is only used to advance towards the optimum, but then a static MA scheme is used to finally drive the plant to optimality.

Although not considered in this work, online parameter estimation can also be performed with a state-estimator, such as an Extended Kalman Filter. However, the advantage of using LDM for parameter estimation is that it does not require additional tuning, as is the case for other state-estimators. Further research is required to compare both approaches, especially under light structural plant-model mismatch, and for the case when the ODE system is not linear in the mismatched parameters, where the use of linearization in LDM might lead to wrong estimates.

4.4. Additional remarks on the method for parameter estimation

The method of parameter estimation based on LDM can be viewed as the first iteration of a Gauss-Newton algorithm, where the initial value corresponds to

the nominal model parameters $\boldsymbol{\theta}_m$. Equation 4.3 is then part of the solution to the following LS problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & (\dot{\mathbf{y}}_p - \dot{\mathbf{y}}_m)^{\mathrm{T}}(\dot{\mathbf{y}}_p - \dot{\mathbf{y}}_m) \\ \text{s.t.} \quad & \dot{\mathbf{y}}_m = \mathbf{f}(\mathbf{y}_p, \mathbf{u}_p, \boldsymbol{\theta}) \end{aligned} \tag{4.10}$$

The Gauss-Newton method solves this problem iteratively as (Nelles, 2001):

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \left(\mathbf{J}_{\mathbf{f}}^{\mathrm{T}}\mathbf{J}_{\mathbf{f}}\right)^{-1}\mathbf{J}_{\mathbf{f}}^{\mathrm{T}}\left(\dot{\mathbf{y}}_p - \mathbf{f}(\mathbf{y}_p, \mathbf{u}_p, \boldsymbol{\theta}^{(k)})\right) \tag{4.11}$$

where the Jacobian $\mathbf{J}_{\mathbf{f}} = \partial\mathbf{f}/\partial\boldsymbol{\theta}$ is evaluated at the current iterate, and $\boldsymbol{\theta}^{(0)} = \boldsymbol{\theta}_m$. If there are more measurements than uncertain parameters ($n_\theta \leq n_y$), the term $\left(\mathbf{J}_{\mathbf{f}}^{\mathrm{T}}\mathbf{J}_{\mathbf{f}}\right)^{-1}\mathbf{J}_{\mathbf{f}}^{\mathrm{T}} = (\mathbf{J}_{\mathbf{f}})^{+}$, which leads to eq. 4.3.

This suggests that an improvement on the method presented in Section 4.1 can be done, based on the assumption that the parameters of the plant only change slowly in time. Instead of performing the iterative Gauss-Newton algorithm on the data $\dot{\mathbf{y}}_{p,k}$ obtained at sampling point k, the iterations can be performed *online* as new data arrives, resulting in a dynamic difference equation:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \left(\mathbf{J}_{\mathbf{f},k}^{\mathrm{T}}\mathbf{J}_{\mathbf{f},k}\right)^{-1}\mathbf{J}_{\mathbf{f},k}^{\mathrm{T}}\left[\dot{\mathbf{y}}_{p,k} - \mathbf{f}(\mathbf{y}_{p,k}, \mathbf{u}_{p,k}, \boldsymbol{\theta}_k)\right] \tag{4.12}$$

If accurate values of $\dot{\mathbf{y}}_{p,k}$ and $\mathbf{y}_{p,k}$ are available, eq. 4.13 can converge to the parameters of the plant, even if the function $\mathbf{f}$ is not linear in the mismatched parameters. The scheme can also be used for parameter tracking if slow drifts occur. Equation 4.13 can be regularized with an adaptive parameter α, resulting in an online Levenberg-Marquardt method:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \left[\mathbf{J}_{\mathbf{f},k}^{\mathrm{T}}\mathbf{J}_{\mathbf{f},k} + \alpha\mathrm{diag}\left(\mathbf{J}_{\mathbf{f},k}^{\mathrm{T}}\mathbf{J}_{\mathbf{f},k}\right)\right]^{-1}\mathbf{J}_{\mathbf{f},k}^{\mathrm{T}}\left(\dot{\mathbf{y}}_{p,k} - \mathbf{f}(\mathbf{y}_{p,k}, \mathbf{u}_{p,k}, \boldsymbol{\theta}_k)\right) \tag{4.13}$$

Future research will focus on evaluating this extended scheme for parameter estimation, especially when there is measurement noise. An option for making the algorithm more robust to noise, is to use the measurements of more than one sampling point at a time, which is done by stacking several $\dot{\mathbf{y}}_{p,k}$ and $\mathbf{J}_{\mathbf{f},k}$.

CHAPTER

FIVE

DYNAMIC MODEL IDENTIFICATION FOR RTO

If a test, such as the one presented in Chapter 3, determines that the dynamic model at hand is structurally mismatched, or if such a model is not available, e.g. in early development stages of a process, a black-box dynamic model can be used for optimization. This chapter presents how such a model can be chosen, and integrated with MAWQA. The procedure is illustrated with two case-studies, namely the CSTR presented in Chapter 3, and the optimization of a homogeneously catalyzed hydroformylation process.

5.1. Nonlinear system identification

System identification consists in selecting an appropriate model structure to describe the dynamics of a process, and estimating the parameters of that selected model to best fit the observed data (Ljung, 1999). In this field, the identification of black-box discrete-time models has been the subject of extensive research; therefore, this type of model will be used in this work. This section presents some of the fundamentals of the system identification problem: a general representation of nonlinear discrete-time models is first introduced, followed by the description of the more specific polynomial models. Some fundamentals of parameter estimation are then presented. Lastly, the problem of model structure selection is discussed. For a more comprehensive treatment on the subject, refer to (Ljung, 1999; Nelles, 2001).

5.1.1. The NARMAX model

The Nonlinear Autoregressive Moving Average with Exogenous Inputs (NARMAX) representation (Leontaritis and Billings, 1985a,b) has become popular for describing the I/O behavior of an arbitrary process. The current output is obtained via a nonlinear functional expansion of lagged input, output, and noise terms:

$$\begin{aligned} y(k) = F[&y(k-1), \ldots, y(k-n_y), u(k-d), \ldots, u(k-d-n_u), \\ &e(k-1), \ldots, e(k-n_e)] + e(k) \end{aligned} \tag{5.1}$$

where y represents the output of the system, u is the exogenous input, e is a noise element, d is the minimum delay (dead time) of the inputs to the outputs, and n_y, n_u and n_e are the maximum lags of the outputs, inputs, and noise, respectively. This class of model encompasses particular structures such as: polynomial expansions, rational functions, and neural networks. The linear Autoregressive Moving Average with Exogenous Inputs (ARMAX) model is a specific case of this general formulation.

Polynomial expansions in particular have been widely used due to their global approximation capability, and have the advantage of being linearly parametrized (Chen and Billings, 1989). This makes parameter estimation straightforward because methods from the Ordinary Least Squares (OLS) family can be used directly. A polynomial single-input single-output (SISO) NARMAX model of order l has the form:

$$\begin{aligned} y(k) =& \theta_0 + \sum_{i_1=1}^{n} \theta_{i_1} x_{i_1}(k) + \sum_{i_1=1}^{n} \sum_{i_2=i_1}^{n} \theta_{i_1 i_2} x_{i_1}(k) x_{i_2}(k) + \cdots \\ &+ \sum_{i_1=1}^{n} \cdots \sum_{i_l=i_{l-1}}^{n} \theta_{i_1 i_2 \cdots i_l} x_{i_1}(k) x_{i_2}(k) \cdots x_{i_l}(k) + e(k) \end{aligned} \tag{5.2}$$

with $x(k) = \{y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u), e(k-1), \ldots, e(k-n_e)\}$. The different products of the elements in $x(k)$ result in noncommutative monomials of order up to l. A particular case of the full expression is when lagged noise terms are not used in the model, which results in the polynomial Nonlinear Autoregressive with Exogenous Inputs (NARX) model. The

polynomial NARMAX model can be separated into three parts: the process model, where no noise terms are included, a pure noise model that consists of a pure noise expansion, and a mixed noise-inputs-outputs expansion (Piroddi and Spinelli, 2003).

Building on the idea of Gao et al. (2016), system identification is used in this work to predict the steady-state of the process. This is calculated by solving eq. 5.2 when setting $y(k) = y(k-1) = ... = y(k-n_y)$ and $u(k) = u(k-1) = ... = u(k-n_y)$ as long as the identified model is stable[1]; this can be viewed as a long-term simulation of the model until the outputs no longer change significantly. For this purpose, only the process model of the NARMAX model is necessary, since no noise or measurements are used for this calculation. Indeed, the purpose of using a noise model is to obtain consistent parameter estimates by taking additional information from the error terms into account (Billings, 2013).

Given that the noise model can easily have more parameters than the process model, NARX structures are usually assumed. However, the noise assumption of the NARX model, namely disturbances affecting the process, is seldom satisfied in practice (Nelles, 2001). A more realistic assumption is the one of the (nonlinear) Output Error model (NOE), where the noise is assumed to affect the measurements, and not the process:

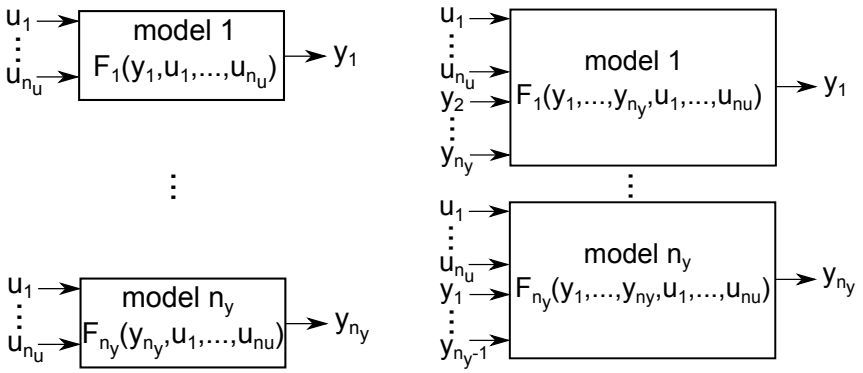
$$\begin{aligned} y(k) &= F[y(k-1), \ldots, y(k-n_y), u(k-d), \ldots, u(k-d-n_u)] \\ \hat{y}(k) &= y(k) + e(k) \end{aligned} \tag{5.3}$$

where $\hat{y}(k)$ denotes the measured output that is corrupted by noise. In this work, the nonlinear models considered are polynomial NOE models. Since only the prediction of the steady-state is of interest, the process inputs $\mathbf{u}$ are not explicitly included, but are implicitly represented in the parameters $\boldsymbol{\theta}$ of the model.

Extending the NARMAX formulation to multiple-input multiple-output (MIMO) systems is typically done by describing each output with a func-

[1] A "steady-state" can also be calculated if the system is unstable, but it does not have physical meaning

tion such as the one in eq. 5.1, which results in a system of (possibly coupled) multiple-input single-output (MISO) models. A way of decoupling the different models is to use higher dynamic orders in each output (Nelles, 2001); however, this is not a good option if the different outputs significantly affect each other, e.g. in a chemical reaction. Figure 5.1 depicts the two representations of MIMO systems.



(a) Decoupled **(b)** Coupled

Figure 5.1.: Representation of MIMO models with MISO models. Adapted from Nelles (2001)

5.1.2. Model structure selection

One of the main drawbacks of using polynomial expansions to represent dynamic systems, is that they suffer heavily from the so called "curse of dimensionality", which means that the number of terms in eq. 5.2 grows quickly as the dynamic orders and polynomial degree increase. Certainly, determining all of the parameters of a full polynomial expansion will result in an ill-conditioned problem that requires a large amount of data due to the many degrees of freedom of the model. A large number of model parameters will reduce the bias of the prediction, but will increase its variance error (overfitting), which results in models that are highly sensitive to noise. It is intuitive to think that not all of the candidate regressors are important, or necessary, to obtain a good representation of the dynamics of the plant.

Model structure selection techniques aim at finding a parsimonious representation of the dynamics of a process, by selecting a subset of the full set

of candidate regressors. In its essence, the idea of the problem is to find the optimum model complexity in terms of the bias/variance trade-off. Methods for model structure selection can be loosely classified as explicit or implicit methods (Nelles, 2001), some of which are briefly discussed in what follows.

Explicit structure optimization: A direct alternative for structure selection would be to evaluate all of the possible models formed from the full regressor set. However, this is only feasible for small problems: for a polynomial NARX model of degree l, and a total number of model inputs $n = n_y + n_u$, the number of candidate regressors is $n_\phi = (n+l)!/(n!l!)$. Clearly, this alternative would yield the best model, but it is computationally prohibitive for most practical applications.

One of the most popular alternatives to tackle the structure selection problem, is the Forward Regression Orthogonal Estimator (FROE) (Korenberg et al., 1988). This method is an incremental model building procedure, where at each iteration a new candidate regressor is added to the model, depending on an importance index related to the improvement obtained in terms of prediction (or explained data variance) when adding a new term. This scheme is very efficient for linearly parametrized models and also exploits orthogonal least squares to make the estimation of different parameters independent. However, since terms are added sequentially, the method is said to follow a "greedy" scheme, which can result in sub-optimal models being selected.

Piroddi and Spinelli (2003) argue that using the Error Reduction Ratio (ERR) of Korenberg et al. (1988) can result in the identification of wrong models, since this index does not capture the long term predictions of the model. This is motivated by the fact that some models might yield very good one-step ahead predictions, but perform very poorly, or even be unstable, when used for long-term predictions (simulation). Consequently, they proposed using an index based on the mean squared simulation error called the Simulation Error Reduction Ratio (SRR). They also proposed extending the FROE algorithm to include a pruning procedure: at each iteration, the regressors present in the model are examined in order to remove the least significant one; thus taking the possible synergy between regressors into account.

One of the drawbacks of this last approach, is that it still used OLS for parameter estimation due to the computational cost of using a simulation-based parameter estimation procedure. Farina and Piroddi (2012) proposed using a full simulation-based model selection algorithm by exploiting a faster parameter estimation method based on multi-step prediction (Farina and Piroddi, 2010): instead of directly minimizing an LS type function, the parameters are identified by sequentially optimizing a $1, 2, ..., k$-step ahead prediction error.

More recently, Falsone et al. (2015) proposed a Randomized Model Structure Selection (RaMSS) algorithm, which selects a model structure by assigning a Regressor Inclusion Probability (RIP) to each candidate regressor. In each iteration, a large number of models are generated by randomly drawing regressors from the candidate set, depending on their RIP. These models are identified via OLS, and a t-test is performed on the obtained parameters to eliminate the ones that are not significantly different from zero. The reduced (non-redundant) models are identified again and evaluated in terms of the mean squared simulation error. The RIP of every regressor is updated depending on how, on average, the performance of the model is affected when the regressor is included. They show that the algorithm, although more computationally intensive than the FROE, can retrieve correct structures in the case of polynomial NARX models. This approach is similar to genetic algorithms, and can thus overcome the problem of local minima.

Another approach based on statistical tests of the estimated parameters, is the Bootstrapping (BS) procedure proposed by Kukreja et al. (1999). This procedure follows a backward selection scheme, which starts from a full model and iteratively removes non-significant regressors by estimating their confidence intervals with BS. This scheme is further discussed in Section 5.2.1.

So far, only model structure selection methods for SISO systems have been discussed. For the case of MIMO systems, the FROE algorithm was extended in (Billings et al., 1989). The problem of structure selection for MIMO systems is more complicated, because if there is significant coupling between MISO models, the number of possible regressors increases rapidly with the number of outputs. The nature of the problem also becomes more complex, since not only is the order in which terms are added or removed to each model important,

but the sequence in which each MISO model is altered can have a significant effect on the final model.

Implicit structure optimization Unlike explicit structure optimization, implicit methods do not directly add or remove regressors from the model, but they act on the bias/variance trade-off by penalizing model complexity. Two of such methods are briefly described here. The first method is the L_2 regularization, also known as ridge regression, where the objective LS function is penalized with the squared norm of the parameters:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \alpha \sum_{i=1}^{n_\theta} \theta_i^2 \tag{5.4}$$

The parameter α controls the degree of regularization. By using this type of penalty, parameters whose value does not significantly improve the fit of the model are decreased more severely towards zero. The solution for a regularized one-step ahead prediction problem is[2]:

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \alpha\mathbf{I}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \cdot \hat{\mathbf{Y}} \tag{5.5}$$

Setting $\alpha = 0$ results in the OLS, and when $\alpha \to \infty$, all of the parameters are set to zero. By increasing the bias of the estimation (parameters closer to zero), the variance is reduced. This approach is simple, but might not be desirable, since it still requires estimation of the full, possibly overparametrized model. However, using a small value of α might be helpful, even when an appropriate model structure is selected, since is improves the condition number of $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}$. This is especially useful when there is insufficient excitation in the system, e.g. close to steady state.

Another technique is based on the L_1 norm, and is called the Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1996). The LASSO objective function takes the form

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \alpha \sum_{i=1}^{n_\theta} |\theta_i| \tag{5.6}$$

[2]See Section 5.1.3 for notation

By using the sum of absolute values as a penalty, some of the parameters can actually be set to zero, so the LASSO can also be used for subset selection. However, due to the presence of the absolute value function, the objective function is no longer differentiable, and its solution is more computationally demanding than for ridge regression. Pillonetto et al. (2014) further reviews the use of regularization (Kernel) methods for system identification, with special emphasis on linear systems.

5.1.3. Parameter estimation

Parameter estimation consists in finding the values of the parameters in the selected model, such that it best represents the observed data; this is also known as training. A common framework for parameter estimation is the LS approach, which is derived from maximum likelihood estimation when the disturbances in the measurements are assumed to be Gaussian. This can be written as:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \\ \text{s.t.} \quad & \hat{y} = F(\boldsymbol{\theta}) \end{aligned} \tag{5.7}$$

where y is the calculated output from the selected model structure F, and $\hat{y}_i$ is the measured value of the output. The model structure can be a function of y or other exogenous variables as well, as is the case in recursive I/O models; but for simplicity, it is denoted as only a function of the model parameters $\boldsymbol{\theta}$.

Ordinary least squares: As seen in the previous section, selecting F as a polynomial expansion results in a model where the one-step ahead prediction is linear in the parameters. This can be expressed as:

$$\hat{y}(k) = \boldsymbol{\varphi}(k) \cdot \boldsymbol{\theta} + e(k) \tag{5.8}$$

where $\boldsymbol{\varphi}(k)$ is a row vector containing the different monomials (or regressors) that are used to predict output $y(k)$, e.g. $y(k-1)$, $y(k-1)u(k-1)$, $y(k-2)^2$. The output is a linear combination of the different regressors, weighted with

the parameters in the vector $\boldsymbol{\theta}$. If N observations of the output are available, one can stack multiple predictions as described in eq. 5.8 as:

$$\hat{\mathbf{Y}} = \boldsymbol{\Phi} \cdot \boldsymbol{\theta} + \boldsymbol{\xi} \tag{5.9}$$

where $\mathbf{Y} = [\hat{y}(1), \hat{y}(2), \cdots, \hat{y}(N)]^{\mathrm{T}}$, $\boldsymbol{\Phi} = \left[\boldsymbol{\varphi}(1)^{\mathrm{T}}, \boldsymbol{\varphi}(2)^{\mathrm{T}}, \cdots, \boldsymbol{\varphi}(N)^{\mathrm{T}}\right]^{\mathrm{T}}$, and $\boldsymbol{\xi} = [e(1), e(2), \cdots, e(N)]^{\mathrm{T}}$. In this case, the LS criterion in eq. 5.7 can be written as:

$$\min_{\boldsymbol{\theta}} \ \left(\hat{\mathbf{Y}} - \boldsymbol{\Phi} \cdot \boldsymbol{\theta}\right)^{\mathrm{T}} \cdot \left(\hat{\mathbf{Y}} - \boldsymbol{\Phi} \cdot \boldsymbol{\theta}\right) \tag{5.10}$$

If the regressors in eq. 5.8 do not include any lagged disturbance terms, i.e. a NARX model, eq. 5.10 has an analytic solution:

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \cdot \hat{\mathbf{Y}} \tag{5.11}$$

where $\hat{\boldsymbol{\theta}}$ is a point estimate of the true parameters. For this solution, it is required that the number of observations be larger or equal to the number of determined parameters ($N \geq n_\theta$), and also that the matrix $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}$ be not singular, which is achieved if the system is sufficiently excited (Ljung, 1999).

However, if the regressors in eq. 5.8 include lagged disturbance terms, i.e. a NARMAX model, the analytical solution in eq. 5.11 no longer applies directly because the disturbance terms are not known a priori. A solution for this is the Extended Least Squares (ELS). At each iteration of the ELS, eq. 5.11 is used to find the parameters of the model, which are then used to estimate the noise, or residuals, as $\boldsymbol{\xi} = \hat{\mathbf{Y}} - \boldsymbol{\Phi}\hat{\boldsymbol{\theta}}$. The matrix $\boldsymbol{\Phi}$ is updated, and the procedure is repeated until the parameters converge.

Method of instrumental variables: A desirable property of any estimator is unbiasedness, which is expressed as $E(\hat{\boldsymbol{\theta}}) = \boldsymbol{\theta}_{\text{true}}$, where $E()$ denotes mathematical expectation. This means that the estimate $\hat{\boldsymbol{\theta}}$ should not systematically deviate from its true value. In the case of an NOE model as the one in eq. 5.3, using eq. 5.11 to estimate its parameters results in biased and non-consistent

parameters[3], even if the model is linearly parametrized, e.g. a polynomial NOE model. This is due to the fact that the entries in matrix $\mathbf{\Phi}$ are correlated with noise.

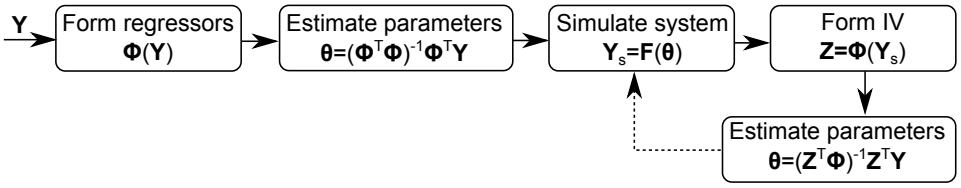


Figure 5.2.: Parameter estimation with the instrumental variables method

One possible solution for this issue is the method of instrumental variables (IV). The idea is to replace the matrix $\mathbf{\Phi}$ in some places of eq. 5.11 with a matrix of variables (the instrumental variables) $\mathbf{Z}$ that are uncorrelated with the noise (Ljung, 1999):

$$\hat{\boldsymbol{\theta}} = \left(\mathbf{Z}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{Z}^{\mathrm{T}} \cdot \hat{\mathbf{Y}} \tag{5.12}$$

One way of calculating such variables is by bootstrapping, where on a first iteration eq. 5.11 is used to obtain a first estimate of the parameters. From this estimation, the model is simulated to obtain a better approximation of the noise-free outputs $\mathbf{Y}_s$. Then, the instrumental variable matrix $\mathbf{Z}$ is calculated the same way that $\mathbf{\Phi}$ is, except that $\hat{\mathbf{Y}}$ is replaced by $\mathbf{Y}_s$. The procedure can be repeated until convergence is achieved (see Figure 5.2). Even though the IV method is consistent, its performance in small data samples might be sub-optimal.

Nonlinear least squares: The other possibility is to solve the LS problem 5.7 not in terms of one-step ahead prediction, but based on the simulation error with a Nonlinear Least Squares (NLS) algorithm such as the Levenberg-Marquardt method. Given that parameter estimation is done with a simulation-based error, i.e. by recursively calculating the simulated data, the problem is always nonlinear in the parameters, regardless of whether the one-step ahead prediction is linearly parametrized. At each iteration, the model parameters

[3]Non-consistency means that the bias in the estimated parameters does not tend to zero, even if the number of observations $N \to \infty$

are used to simulate a dynamic trajectory without the use of the measured data. The difference between these results and the measured outputs is minimized in the LS sense. The use of simulation error instead of prediction error is the natural framework for OE models, since their optimal one-step ahead prediction is a simulator, independent of the data (Nelles, 2001). Although good parameter estimates can be obtained via NLS, the computational cost of the method and the risk of convergence to local minima are its main drawbacks.

The methods discussed in this section can also be used for estimating the parameters of a MIMO system: when using OLS, the models are decoupled, since only measured information is used. In the case of simulation error minimization, however, the coupled MIMO system must be simulated together. Due to the higher number of parameters, and the possible interaction between the different MISO systems, the estimation is more computationally intensive.

5.2. Case study 1

This section considers the optimization of the CSTR case-study introduced in Chapter 3, for which the optimization problem was stated in Chapter 4. Recall that the plant-model mismatch introduced in this case-study is parametric, and that the mismatched parameters appear on the constraints and objective function of the problem. This means that an RTO scheme based on parameter estimation would be required to drive the plant to its optimum. For the sake of illustrating the use of dynamic model identification with this case-study, these mismatched parameters will only be considered to affect the dynamics of the plant, i.e. the objective function and constraints are calculated with the true parameters.

5.2.1. Dynamic model selection

The data used for model selection was assumed to correspond to the start-up of the reactor, with $c_A(0) = c_B(0) = c_C(0) = c_D(0) = 0$, and the input flows set to their nominal optimal value. The measurements were corrupted with zero-mean Gaussian noise with $\sigma_n = 10^{-3}$ in order to estimate parameter significance. For this particular step-response, the plant takes around 140

minutes to reach steady-state. Model selection was done with the data from the first 50 sampling points, where most of the dynamics are evident.

The model selection algorithm used is based on BS, and is summarized in the following steps. The IV method was used for parameter estimation:

1. Compute an estimate of the unknown parameter vector $\boldsymbol{\theta}$ with the current model, and estimate the residuals $\boldsymbol{\xi}$.
2. Generate N bootstrap data sets $\mathbf{Y}_{BS}$ and compute the BS parameter replications $\boldsymbol{\theta}_{BS}$ (see Figure 5.3).
3. Rank the N estimates of each parameter in ascending order to form percentile intervals.
4. Estimate the upper and lower bounds of each parameter's confidence interval for a significance level α ($\overline{\boldsymbol{\theta}}_{i,\alpha}$ and $\underline{\boldsymbol{\theta}}_{i,\alpha}$, respectively).
5. List the parameters i for which zero lies in their confidence interval, in ascending order of the amplitude of the interval defined by $\overline{\boldsymbol{\theta}}_{i,\alpha} - \underline{\boldsymbol{\theta}}_{i,\alpha}$. Set $i_\theta = 1$.
6. Propose the regressor corresponding to parameter i_θ for elimination: fit the reduced model and evaluate its mean simulation error.
7. If the simulation error increases significantly, keep the regressor and increase i_θ by one, and go to Step 6. Else, eliminate the regressor and go to Step 1 until convergence.

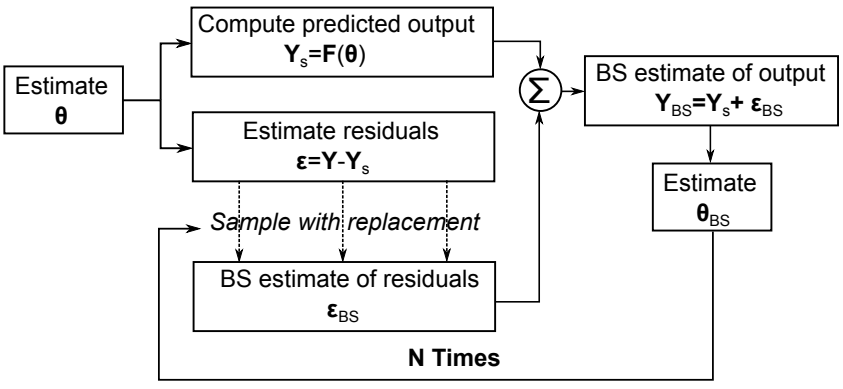


Figure 5.3.: Procedure for forming bootstrap data. Adapted from Kukreja et al. (1999)

The original BS algorithm by Kukreja et al. (1999) was used for model selection of a SISO NARMAX system, and instead of eliminating one regressor per iteration, they proposed eliminating *all* of the regressors whose confidence intervals include zero. In the proposed approach, only one parameter is eliminated at a time, depending on how large their confidence interval is: parameters with a very large confidence interval are not necessarily non-significant, but their large variance can be due to ill-conditioning of the initially large model.

Initially, a full quadratic model of 15 terms per model was proposed. The causal structure of the nominal ODE model was used to reduce the potential regressor set by taking into account that, e.g. c_C does no physically affect c_A; this corresponds to using physical insight of the process. The reduced quadratic model fitted the data well, and was therefore used as an initial model for the BS procedure. The causality of the ODE model was also used to perform regressor selection sequentially: initially, the models for c_A and c_B were determined together. The model for c_D was then obtained along with the converged models. Finally, the model for c_C was evaluated, and the final MIMO system was obtained.

Table 5.1 indicates which regressors are present in the linear model ($\Diamond$), the nonlinear model selected with BS ($\circ$), and a nonlinear model corresponding to an explicit Euler discretization of the ODE system ($\square$). This last model was only used as a reference, since the ODE system describing the CSTR is polynomial. The selected nonlinear model almost exactly matches the one generated by Euler discretization of (3.4), except for the term $c_B(k-1)^2$ in the model for c_B. This is due to the fact that the reaction constant k_2 is significantly less than the constant k_1 and, since c_A and c_B are highly correlated, the term $c_B(k-1)^2$ is not distinguished from the term $c_A(k-1)c_B(k-1)$ at the significance level and noise amplitude chosen.

5.2.2. Evaluation of the selected nonlinear model

The prediction of the steady-state in RTO operates in an independent layer, regardless of the optimization scheme used. This is because the optimizer has to wait for the steady-state measurements in order to calculate the new set-points in the iterative process, which means that it remains inactive during transient

Table 5.1.: Input-output models for the description of the CSTR case-study

Regressor	$c_A(k)$	$c_B(k)$	$c_C(k)$	$c_D(k)$
1	∘,◊,□	∘,◊,□	◊	◊
$c_A(k-1)$	∘,◊,□	◊	◊	
$c_B(k-1)$	◊	∘,◊,□	◊	◊
$c_C(k-1)$			∘,◊,□	
$c_D(k-1)$				∘,◊,□
$c_A(k-1)c_B(k-1)$	∘,□	∘,□	∘,□	
$c_B(k-1)^2$		□		∘,□

operation. Therefore, the difference between RTO schemes is essentially the trajectory they generate to reach the optimum.

In order to evaluate the quality of the selected nonlinear model, a random sequence of 200 set-point changes was generated from the bivariate normal distribution centered at $\mathbf{u}_c = 0.5(\mathbf{u}_m^* + \mathbf{u}_p^*)$ with covariance matrix $\mathbf{\Sigma}$ where $\text{diag}(\mathbf{\Sigma}) = 0.25\|\mathbf{u}_m^* - \mathbf{u}_p^*\|^2$; starting from the nominal optimum. Once a suitable steady-state prediction was obtained, the next set-point change was applied, resulting in a sort of "race" between the two competing models (linear and nonlinear). For this study, the measurements were noise-free; therefore, parameter estimation was performed with OLS in a moving-window scheme, as shown in Figure 5.4 and described in what follows:

The procedure is based on the displacement and growth of a data window of minimum size L_0. There are two main loops: one in which the window grows into the past to improve the regression, and another one where the window shrinks to L_0 and is displaced to the more-recent points to "forget" older information. In order to guarantee that the steady-state predicted is physically consistent, Gao and Engell (2016) proposed using a mass-balance criterion taken from the reaction network in the nominal ODE model:

$$\left| \frac{(\bar{c}_A + \bar{c}_B + 2\bar{c}_C + 2\bar{c}_D)(u_A + u_B)}{(c_{A,\text{in}} u_A + c_{B,\text{in}} u_B)} - 1 \right| \leq 0.002 \tag{5.13}$$

The predicted steady-state should also not change more than a tolerance (0.1%) between iterations.

Table 5.2 displays the average stopping time $\bar{t}_{\text{stop}}$ of the windowing algorithm at each point, and the average error of the predicted steady-state objective

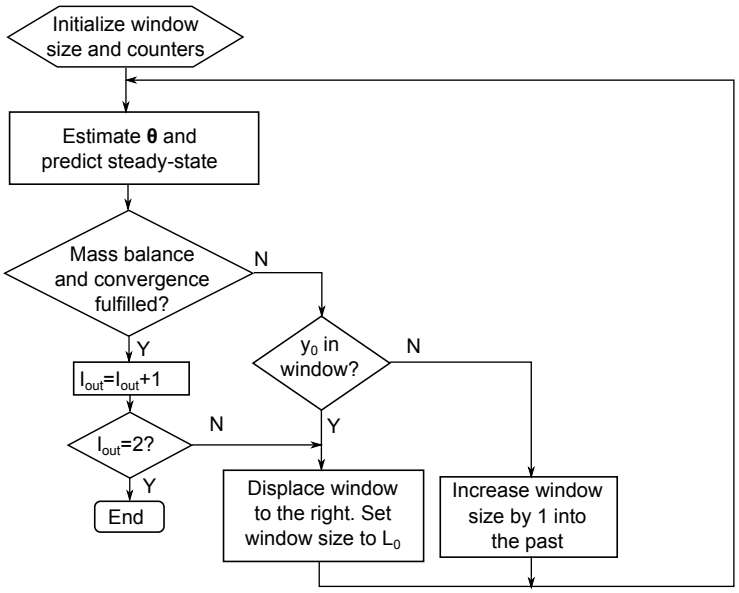


Figure 5.4.: Flowchart for the moving window algorithm for parameter estimation

Table 5.2.: Performance of the dynamic models for the CSTR case-study

	L_0	$\bar{t}_{\text{stop}}$ (min)	$\bar{E}_J$ (%)
Linear model	6	24.4±8.5	0.3±0.2
	10	25.2±8.1	0.3±0.2
	20	30.5±6.6	0.2±0.2
Nonlinear model	6	21.5±7.9	0.5±0.8
	10	23.3±7.2	0.5±0.6
	20	29.4±5.3	0.4±0.5

function $\bar{E}_J$; with standard deviations. Both models predict the steady-state with an accuracy of around 1%. Although the nonlinear model seems to perform faster than the linear one for the different window sizes used, the difference is not large; and due to the large standard deviations, it is not possible to say that the nonlinear model will, in general, perform better than the linear model. Clearly, the nonlinear model will perform better in set-point changes that result in a nonlinear trajectory of the outputs. However, even for highly nonlinear systems, such a trajectory might only result from a large set-point change, which might not happen frequently. As in most optimization schemes, there is a trade-off between convergence speed and robustness of the algorithm,

the latter being favored if the algorithm is applied online to a real plant, which is why large set-point changes are usually avoided.

5.2.3. Simulations with no measurement noise

For the optimization, the flow rates u_A and u_B are normalized in the range 0-50 (L/min). The starting set-point was taken as the nominal optimum $\mathbf{u}_m^*$, and MAWQA was initialized with a perturbation of size $\Delta h = 0.1$. The parameters $\mathbf{\Delta u} = 0.1$ and $\gamma = 3$ were chosen as in (Gao and Engell, 2016). Two cases were analyzed: optimization with the predictions from the nonlinear model, and optimization with the predictions from the linear model.

Figure 5.5 illustrates the real cost function corresponding to the different set-points applied during optimization with the nonlinear and linear model, respectively. The optimum is reached in both cases, with the nonlinear model resulting in a slightly faster convergence (352 min vs. 385 min for the linear model). Both models perform significantly better than when no transient measurements are used, which requires 13 iterations (about 780 minutes) for convergence. In both cases, the RTO iterations approach the optimum quickly but then the MAWQA algorithm introduces additional moves in order to improve the estimation of the gradients.

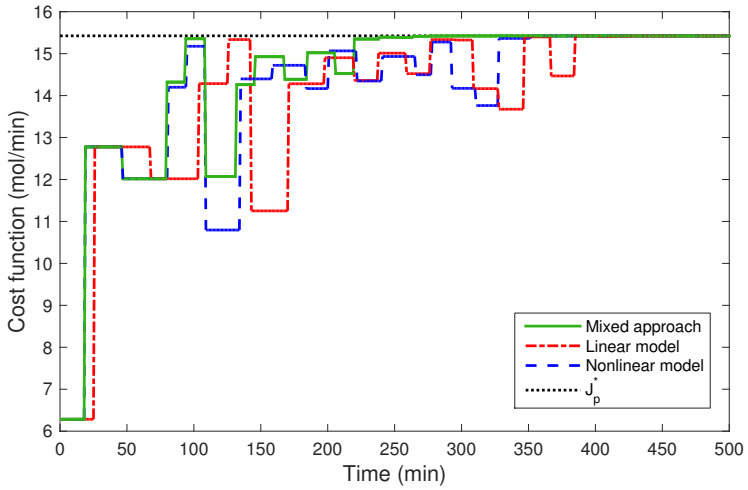


Figure 5.5.: Objective function values for the set-points applied with the nonlinear model (blue), the linear model (red), and a mixed approach (green) for case-study 1

The convergence using the nonlinear model is clearly faster during the first six iterations than with the linear model. In these iterations, finite differences are used to estimate the plant gradients; after the 6th iteration, the QA is employed. After six iterations, the trajectories of both models start deviating from each other, as shown in Figure 5.6. The reason for this is that the steady-states predicted by both dynamic models, while always within 1% of the true value, still are inaccurate. This difference affects the QA, especially in the first iterations where it is used, since there are not enough data points to make the approximation robust to these errors, resulting in biased gradient estimates. This is similar to overfitting a model with noisy measurements.

Therefore, a combination of the two approaches was tested (also shown in Figure 5.5). The idea was to identify simultaneously both models (linear and nonlinear) online and use the predicted steady-state of the model that first satisfied the convergence criteria. In this case, convergence is achieved after only 225 min. During the first RTO iterations (when the changes in the inputs are larger) the nonlinear model is used; then when the system approaches its optimum, the algorithm switches to the linear model.

There is a trade-off between getting a precise estimate of the steady-state, and the speed at which it is obtained: during the first iterations it makes sense to get quick, albeit inaccurate, predictions to start moving towards the optimum. However, when the set-point changes are small, which happens close to convergence, it might be more beneficial to wait for more accurate estimates. This is especially true when there is measurement noise, since performing parameter estimation under insufficient excitation can yield highly biased predictions that drive the plant away from the optimum.

5.2.4. Simulations with measurement noise

In order to evaluate the impact of measurement noise, fifty optimization runs were simulated for two minimum regression window sizes L_0, and a noise level $\sigma_n = 0.001$, which roughly corresponds to 5% of the weakest signal, i.e. $c_D(t)$. Parameter estimation was performed with the windowing algorithm of Figure 5.4 and the method of IV to reduce the bias in the estimated parameters.

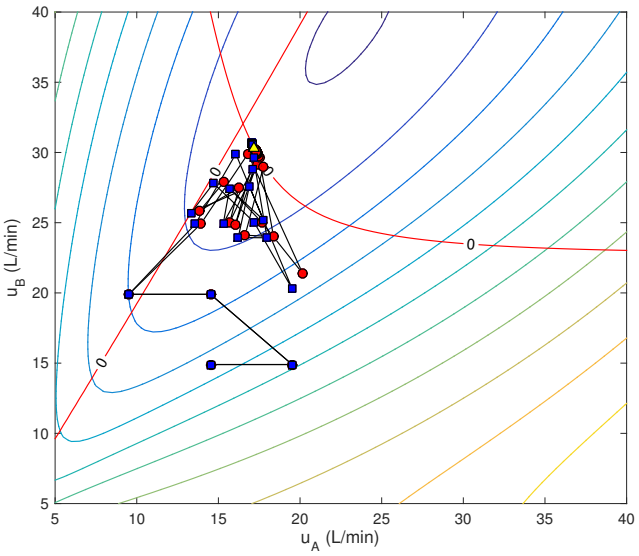


Figure 5.6.: Optimization trajectory of the nonlinear model (blue) and the linear model (red) for case-study 1. The yellow triangle represents the optimum of the plant.

The simulations were also performed for the case where no dynamic information is used, by assuming an average setting time of 60 minutes, and corrupting the steady-state data with the same noise level. In a real plant where measurement error is present, one would use a steady-state detection algorithm to determine when the plant settles to steady-state (Kelly and Hedengren, 2013); however, this approach was not pursued in this work.

Table 5.3.: Optimization of the CSTR case-study with measurement noise

L_0	Profit loss (mol)	$\bar{t}_{1\%,\text{last}}$ (min)	n_{nlin} (%)
20	2376.7±614.5	1084.2±364.1	18.6±3.4
30	2311.8±404.5	1101.1±334.8	22.1±3.4
Static	2106.1 ± 777.1	1996.8±777.1	

The results of this study are shown on Table 5.3, where the percentage of occasions n_{nlin} where the nonlinear model was used for prediction is also included. While the average profit loss is similar in the three scenarios, its standard deviations decreases with an increasing regression window size. This is due to the fact that using larger regression windows leads to better noise

filtering, which improves the robustness of the scheme to noise. Note that the nonlinear model is used in around 20% of the iterations during optimization, which indicates that using such an additional model can improve the time performance of the overall algorithm.

The time required for convergence is also lower for the cases in which transient data is used, which shows that the approach can be promising, even in the presence of measurement noise. However, convergence takes significantly longer in this case, than when no measurement noise is present. This happens because the windowing algorithm takes longer to find a steady-state prediction that satisfies the mass balance criterion; sometimes even reaching steady-state with no suitable prediction. Future research will need to focus on how to improve the windowing algorithm to avoid the double-loop structure, and only increase or shrink the size of the window when favorable: a larger window can even-out the effect of noise, but if the window is too large, the dynamic model might not approximate the data well enough and give biased predictions. A statistical test as the one presented in Chapter 3 could be used to determine how well the model fits the data in the window, and decide whether the window should be shrunk, expanded, or displaced.

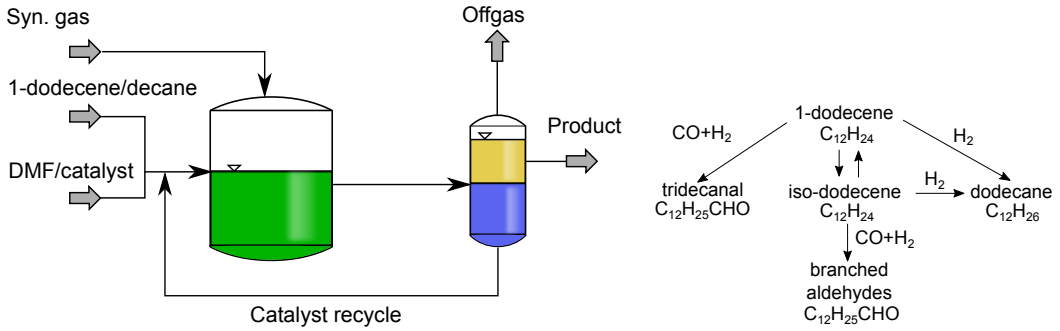
Although the method of IV is useful to get good parameter estimates when there is measurement noise, the best parameter estimates for a NOE model would still be obtained by using an NLS algorithm based on simulation error. The development of a fast parameter estimation algorithm, especially for coupled MIMO systems, would be of great benefit. A possible research direction is to adapt the parameter estimation procedure for differential equations developed by Ramsay et al. (2007) to NOE models. In this method, parameters are not directly estimated by minimizing the simulation error, but are obtained by relaxing the constraints represented by the model, which improves the convergence of the estimation in terms of computational time and conditioning.

5.3. Case study 2

The second case-study in this thesis is the homogeneously catalyzed hydroformylation of 1-dodecene with a Thermomorphic Multicomponent System (TMS) (Brunsch and Behr, 2013). This is a good example of a process where the presence of a recycle stream increases the time constant of the system, resulting in very slow dynamics (Lakshminarayanan and Takada, 2001).

5.3.1. Process description

In this process, the substrate 1-dodecene is fed to the reactor, along with the polar solvent dimethylformamide (DMF), the apolar solvent *n*-decane, the catalyst system [Rh(acac)$(CO)_2$]/Biphephos, and synthesis gas (CO/H_2). During reaction, the system is homogeneous and no mass transport limitation between liquid phases occurs. The temperature is then reduced in the decanter, which results in two liquid phases: an apolar phase with the desired products, and a polar phase that contains the catalyst, and is recycled to the reactor. This concept was proven in a continuously-operated miniplant at TU Dortmund (Zagajewski et al., 2014). Figure 5.7 shows a scheme of the process, as well as the different reactions that take place in the system, with tridecanal being the product of interest.



(a) Process schematic **(b)** Reaction network

Figure 5.7.: Thermomorphic Multicomponent Solvent (TMS) system. Adapted from Hernández and Engell (2016)

Hernández and Engell (2016) proposed and validated a model for the TMS based on material balances, and empirical correlations for the separation in the decanter. A description of the model is presented in Appendix A.

The measurements correspond to the input flow F_e of 1-dodecene/DMF, the input flow F_m of the DMF/catalyst makeup, and the mass percentages of the seven species present in the liquid product stream: DMF (1), n-decane (2), 1-dodecene (3), iso-dodecene (4), dodecane (5), tridecanal (6), and iso-aldehyde (7). The data is assumed to be collected with a sampling time $t_s = 20$ min.

5.3.2. Optimization problem

The optimization problem for this system consists on the minimization of the cost per unit tridecanal produced. In this work, only the costs associated with the raw materials are considered; a more complete version of the problem would include the cost of cooling and heating in the system, as well as the cost of compression for the gas. The objective function is:

$$\min_{\mathbf{u}} \quad \frac{Pr_{\text{1-dodecene}} F_{\text{1-dodecene}} + Pr_{\text{Rh}} F_{\text{Rh}}}{F_{\text{tridecanal}}} \tag{5.14}$$

where $Pr_{\text{1-dodecene}}$ and Pr_{Rh} are the prices of 1-dodecene and the Rhodium catalyst precursor, respectively; $F_{\text{1-dodecene}}$, and F_{Rh} are flow rates of 1-dodecene and Rhodium catalyst precursor in the inlets, respectively; and $F_{\text{tridecanal}}$ is the flow of the produced tridecanal. The inputs $\mathbf{u}$ comprise the reactor temperature, the catalyst dosage, the gas pressure, and the CO molar fraction in the gas. The inputs are additionally constrained to guarantee feasible operation.

For this problem, plant-model mismatch is introduced by decreasing the Henry coefficients $H_{j,0}$ by 50% and setting the parameter $K_{cat,2} = 0$ (See Appendix A). Table 5.4 summarizes the optimal operating conditions based on the nominal model, and the actual optimal operating conditions of the plant. The initial point refers to the operating conditions evaluated by Zagajewski et al. (2014), with maximum catalyst dosage.

Table 5.4.: Operating conditions of the TMS system

	Operating interval	Initial point	Model optimum	Plant optimum
Reactor temperature (°C)	85-105	90	85.1	88.69
Catalyst dosage (ppm)	0.25-2	2	0.301	0.51
Gas pressure (bar)	1-3	2	3	3
CO fraction	0.01-0.99	0.5	0.66	0.55
Cost (Euro/kmol)		937.84	818.88	761.34

5.3.3. Data reconciliation

Steady-state data reconciliation was used to estimate the unmeasured flow of product F_p. The data reconciliation problem was formulated as:

$$\begin{aligned}
\min_{\hat{F}_p, \hat{\mathbf{w}}_p} \quad & (\hat{\mathbf{w}}_p - \mathbf{w}_p)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\hat{\mathbf{w}}_p - \mathbf{w}_p) \\
\text{s.t.} \quad & F_m w_{m,\mathrm{DMF}} + F_e w_{f,\mathrm{DMF}} - \hat{F}_p \hat{w}_{p,\mathrm{DMF}} = 0 \\
& F_m w_{m,\mathrm{decane}} + F_e w_{f,\mathrm{decane}} - \hat{F}_p \hat{w}_{p,\mathrm{decane}} = 0 \\
& \frac{F_m w_{m,\mathrm{dodecene}}}{M_{\mathrm{dodecene}}} + \frac{F_e w_{f,\mathrm{dodecene}}}{M_{\mathrm{dodecene}}} - \hat{F}_p \sum_{i=3}^{7} \frac{\hat{w}_{p,i}}{M_i} = 0 \\
& 1 - \sum_{i=1}^{7} \hat{w}_{p,i} = 0
\end{aligned} \tag{5.15}$$

where $w_{i,j}$ represents the measured (or known) mass fraction of component j in stream i, and the $\hat{()}$ denotes a reconciled quantity. The covariance matrix $\boldsymbol{\Sigma}$ is used to reflect the confidence on the measurements, and to normalize them if different physical quantities are reconciled. The reconciled quantities must satisfy the mass balance of DMF (1), decane (2), C-12 chains (3), and a closure condition for the reconciled mass fractions (4).

5.3.4. Dynamic model selection

Model structure selection was done with what was assumed to be a step response of two hundred points from the initial operating point, to the nominal model-based optimum. The first one hundred data points were used for model training. In contrast to the first case-study, no information on causality from

the ODE model was used to reduce the subset of possible regressors, due to the complexity of the reaction network and the separation in the decanter. For this case-study, the measurements were assumed to be noise-free, since the effect of measurement noise was already discussed in the previous case-study.

The selection procedure was split into two parts after noticing that the mass fractions of DMF and *n*-decane do not vary significantly during the step response; these two substances do not react, and have constant flow rates, so any dynamics observed result from changes in the decanter and the recycle stream. Therefore, it was assumed that they do not have a significant effect on the dynamics of the other species. A coupled system of five MISO systems (1-dodecene, iso-dodecene, dodecane, tridecanal, and iso-aldehyde) was initially selected. After this, the models for DMF and decane were included and selected.

The BS structure selection algorithm of Section 5.2.1 cannot be used in this case, since it is based on directly estimating parameter variance based on BS replicates of noise. Furthermore, given the size of the full MIMO system (seven MISO submodels) and the number of potential regressors, even for low polynomial degrees, the BS computations would be computationally demanding. For this reason, the following variation of the forward selection algorithm with pruning of Piroddi and Spinelli (2003) was used for structure selection (see Figure 5.8).

1. Define the sets $\mathcal{M}_j^{\text{out}}$ of candidate regressors for each submodel j.
2. Initialize each MISO submodel i with its corresponding autoregressive term and the exogenous term, i.e. $\varphi(k) = [y_i(k-1), 1]$. Set the iteration counter $k = 1$.
3. For every submodel j propose the inclusion of each candidate regressor i in the candidate subset $\mathcal{M}_j^{\text{out}}$. Obtain the model parameters with OLS and calculate the mean simulation error of the whole coupled system.
4. Include the regressor $\phi_{j,i}$ that results in the lowest simulation error in the iteration, and remove it from its corresponding candidate subset: $\mathcal{M}_j^{\text{out}} \leftarrow \mathcal{M}_j^{\text{out}} \setminus \phi_{j,i}$, $\mathcal{M}_j^{\text{in}} \leftarrow \mathcal{M}_j^{\text{in}} \cup \phi_{j,i}$.
5. For every submodel j propose removing each candidate regressor i in the model subset $\mathcal{M}_j^{\text{in}}$. Obtain the model parameters with OLS and calculate the mean simulation error of the whole coupled system.

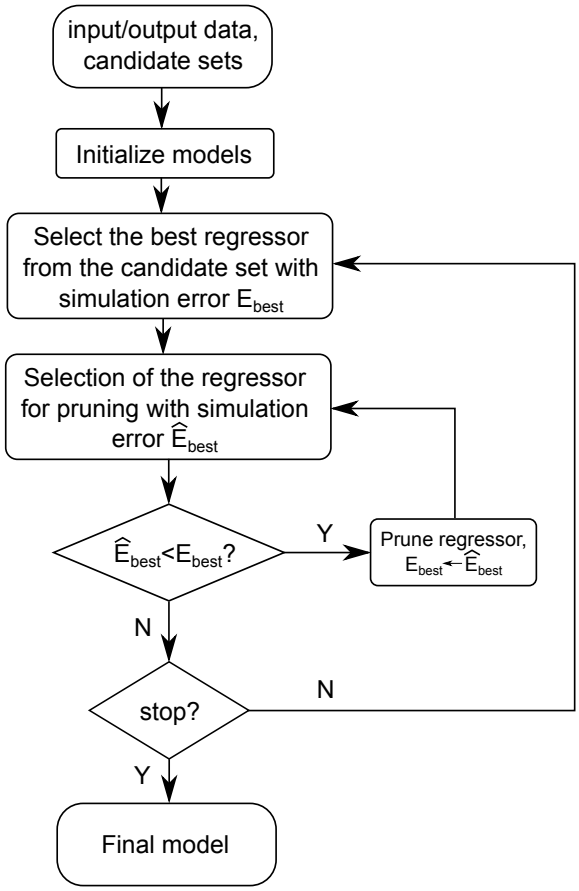


Figure 5.8.: Forward selection with pruning algorithm for model structure selection. Adapted from Piroddi and Spinelli (2003)

6. If the best model obtained after pruning regressor $\phi_{j,i}$ has a lower or equal simulation error than the model without pruning, remove the regressor from the model and go to Step 5: $\mathcal{M}_j^{\text{in}} \leftarrow \mathcal{M}_j^{\text{in}} \setminus \phi_{j,i}$, $\mathcal{M}_j^{\text{out}} \leftarrow \mathcal{M}_j^{\text{out}} \cup \phi_{j,i}$.
7. If all candidate regressor subsets are empty, or the improvement in simulation error is below a tolerance, end. Else, increase the iteration counter by one, and go to Step 3.

Two additional restrictions were imposed during the pruning procedure:

- The autoregressive term $y_j(k-1)$ cannot be eliminated from the model for $y_j(k)$. If the underlying system that generates the measured data is

continuous, the exact value of the sampled data can be obtained from the unknown true continuous mapping $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta})$[4] as:

$$\mathbf{y}(k+1) = \mathbf{y}(k) + \int_{kt_s}^{(k+1)t_s} \mathbf{f}(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}) \mathrm{d}t \tag{5.16}$$

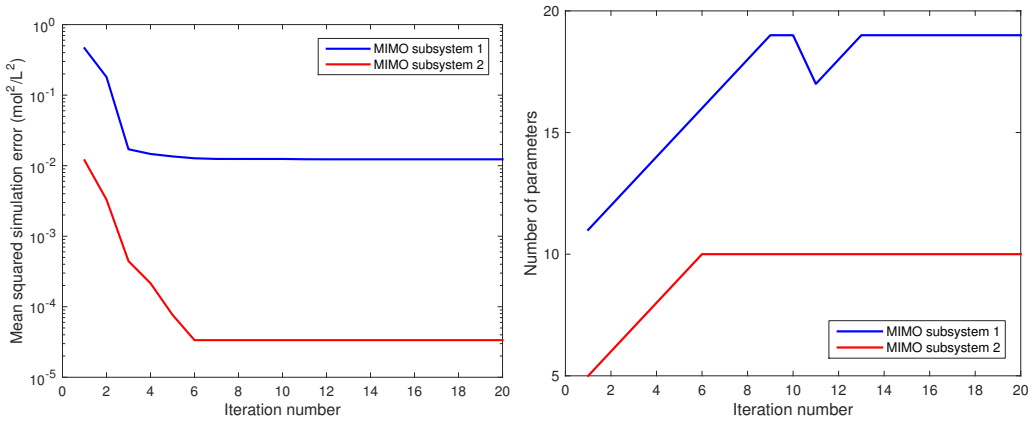
From which it is evident that the autoregressive term should be present in the model.

- A submodel can only be reduced if it has more than two regressors. Because one term in each submodel corresponds to its autoregressive term, there should be at least another regressor in the model: a purely autonomous system is either unstable, or has a steady-state equal to zero.

The results of the model selection procedure for the TMS process from a full quadratic model with maximum dynamic order $n_y = 1$ are represented in Figure 5.9. The blue lines represent the MIMO subsystem composed of the first five species (1-dodecene, iso-dodecene, dodecane, tridecanal, and iso-aldehyde), while the red lines represent the full MIMO model, where only the subsystem composed of DMF and *n*-decane was modified. The mean squared simulation error was calculated with all of the 200 measured points. Since the model was only trained with the initial 100 points, this validation was done in order to guarantee that the selected model does not predict highly deviated trajectories, even when it is not trained with the full data set.

Figure 5.9b shows that after reaching a MIMO system with 19 parameters, the size of the model does not increase. However, the simulation error does decrease, which indicates that the selection algorithm starts adding terms and pruning less significant regressors. Note that at iteration 10 the size of the model is reduced, and then increased again. Due to the pruning procedure, the algorithm can take into account the synergy between regressors, which can in fact reduce the size of the model if a better combination of regressors is found (Piroddi and Spinelli, 2003). The mean squared simulation error is monotonically decreasing, but it stops changing significantly after the third iteration (See Figure 5.9a). For this reason, the model obtained at the third iteration was selected for the MIMO subsystem of five species.

[4]See Chapter 4 for notation



(a) Mean squared simulation error **(b)** Number of parameters in the subsystem

Figure 5.9.: Results of the dynamic model selection for case-study 2. MIMO subsystem 1 (blue) and MIMO subsystem 2 (red)

For the selection of the models for DMF and *n*-decane, the model obtained at the third iteration was selected (notice the slope change at this iteration). Even though the simulation error decreases significantly until the sixth iteration, the improvement in the predicted steady-state is less than 0.1%, and does not justify the increase in the number of parameters. Note that for the two analyzed subsystems the error does not go to zero, which means that the selected quadratic model cannot perfectly predict the steady-state of the system with the given data.

Model selection was also performed without the pruning algorithm. It was observed that after some point, the simulation error became worse, only to become better after more regressors were added. These overparametrized models, although displaying an excellent fit, were usually unstable when fitted to different data, and were therefore not chosen. The final dynamic model

consists of seven MISO submodels (one for each mass fraction w_i) with a total of 20 regressors:

$$
\begin{aligned}
w_1(k) &= \begin{bmatrix} w_1(k-1) & w_3(k-1)w_7(k-1) & 1 \end{bmatrix} \boldsymbol{\theta}_1 \\
w_2(k) &= \begin{bmatrix} w_2(k-1) & w_3(k-1) & w_4(k-1)w_6(k-1) & 1 \end{bmatrix} \boldsymbol{\theta}_2 \\
w_3(k) &= \begin{bmatrix} w_3(k-1) & w_3(k-1)^2 & 1 \end{bmatrix} \boldsymbol{\theta}_3 \\
w_4(k) &= \begin{bmatrix} w_4(k-1) & 1 \end{bmatrix} \boldsymbol{\theta}_4 \\
w_5(k) &= \begin{bmatrix} w_5(k-1) & 1 \end{bmatrix} \boldsymbol{\theta}_5 \\
w_6(k) &= \begin{bmatrix} w_6(k-1) & w_3(k-1)w_5(k-1) & 1 \end{bmatrix} \boldsymbol{\theta}_6 \\
w_7(k) &= \begin{bmatrix} w_7(k-1) & w_5(k-1)w_7(k-1) & 1 \end{bmatrix} \boldsymbol{\theta}_7
\end{aligned}
\tag{5.17}
$$

5.3.5. Simulations with no measurement noise

The optimization of the TMS process was done with MAWQA using the predicted steady-states from the identified models, and with $\Delta\mathbf{u} = 0.05$, $\Delta h = 0.05$ and $\gamma = 2$. A maximum of 100 optimization iterations was allowed. The inputs were normalized in their respective operating ranges. Parameter estimation was done with the same moving window concept of Figure 5.4. For this case-study, the predicted mass fractions had to fulfill the following closure condition:

$$
\left| \sum_{i=1}^{7} w_i - 1 \right| \leq \epsilon \tag{5.18}
$$

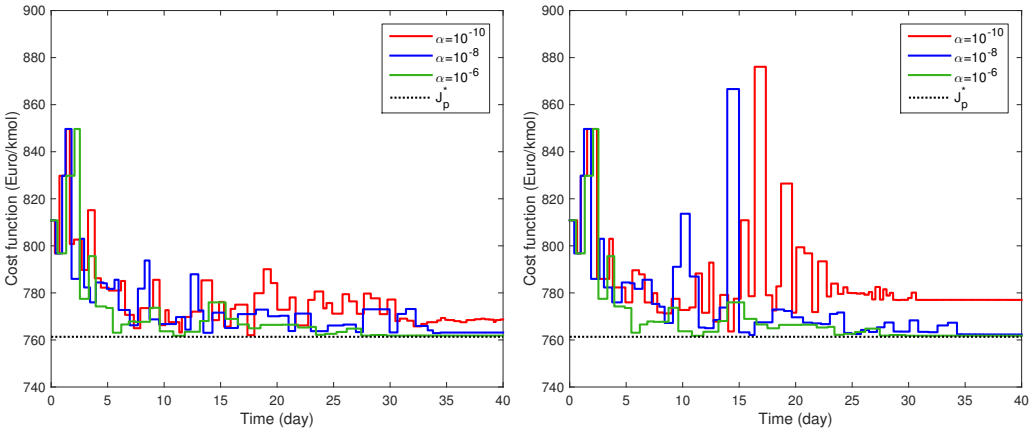
The tolerance ϵ is a tuning parameter that balances the accuracy of the predicted steady-state, with the time required for the prediction to be obtained. This tolerance was chosen as $\epsilon = 0.002$, as in the first case-study. The predicted steady-state was then used with the data-reconciliation algorithm to estimate the flow of product F_p and the objective function. Although not pursued in this work, the data reconciliation step could be performed with a weighting matrix $\boldsymbol{\Sigma}$ (possibly a covariance matrix) that reflects the confidence in the predicted steady-state value of each output.

Initially, a full linear model with a minimum window size $L_0 = 10$ was used to evaluate the potential improvement in optimization time when transient data is used:

$$\begin{bmatrix} w_1(k) \\ w_2(k) \\ w_3(k) \\ w_4(k) \\ w_5(k) \\ w_6(k) \\ w_7(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_1^{\mathrm{T}} \\ \boldsymbol{\theta}_2^{\mathrm{T}} \\ \boldsymbol{\theta}_3^{\mathrm{T}} \\ \boldsymbol{\theta}_4^{\mathrm{T}} \\ \boldsymbol{\theta}_5^{\mathrm{T}} \\ \boldsymbol{\theta}_6^{\mathrm{T}} \\ \boldsymbol{\theta}_7^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} w_1(k-1) \\ w_2(k-1) \\ w_3(k-1) \\ w_4(k-1) \\ w_5(k-1) \\ w_6(k-1) \\ w_7(k-1) \\ 1 \end{bmatrix} \tag{5.19}$$

where each $\boldsymbol{\theta}_i$ is a column vector containing eight parameters. It was observed that estimating the parameters directly with OLS as in eq. 5.11 resulted in bad steady-state predictions due to ill-conditioning of the regression matrices, especially when the process approached steady-state. This problem was addressed by regularizing the regression matrices with a small L_2 penalty (See eq. 5.5). Figure 5.10 shows the result of the optimization procedure for three different values of the penalty α, and two different minimum window sizes L_0; while Figures 5.11 and 5.12 show the accuracy of the predicted steady-state, and the time required before a prediction is made, respectively. The figures on the left correspond to $L_0 = 10$, while the ones on the right correspond to $L_0 = 20$. The penalty α was set to 10^{-10} (red), 10^{-8} (blue), and 10^{-6} (green); these values were chosen small to avoid introducing a large bias in the parameters.

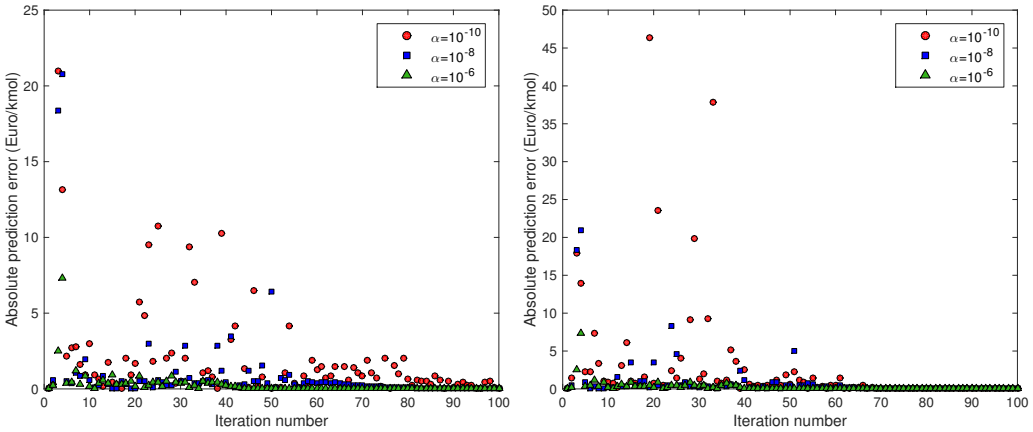
Figures 5.10a and 5.10b show the true objective function during optimization for the two window lengths considered. It can be seen that applying more regularization (higher α) results in faster convergence and less drastic moves (note that in the cases when $\alpha = 10^{-10}$, the algorithm does not converge to the optimum). This behavior is in part due to the accuracy with which the steady-state is predicted during transient operation: a highly biased prediction can be seen as a measurement with a high noise level, which will affect the accuracy of the gradients, and mislead the optimization trajectory.



(a) $L_0 = 10$ **(b)** $L_0 = 20$

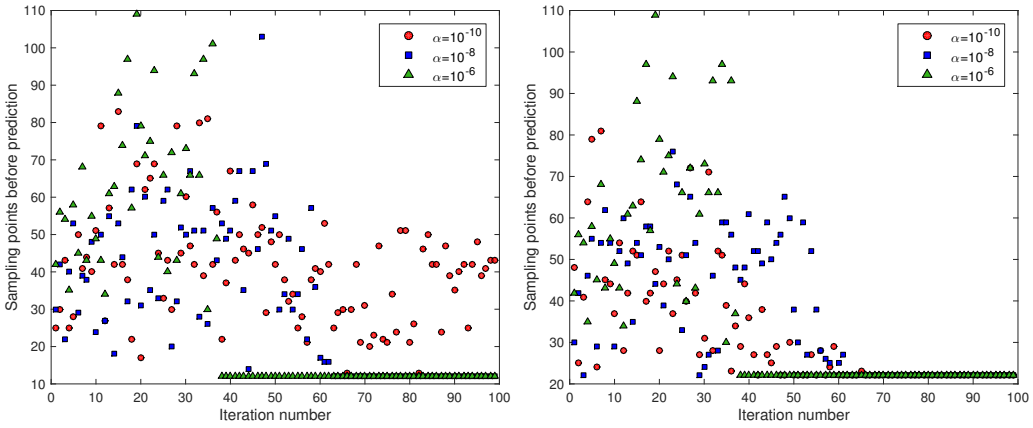
Figure 5.10.: Effect of the regularization penalty α and the minimum window length L_0 on the optimization trajectory for the TMS process

Figures 5.11a and 5.11b indicate that when using regularization, the error in the predicted objective function is indeed reduced, meaning that the optimizer receives more reliable information. This is due to the fact that it takes longer to obtain a satisfactory steady-state prediction, as shown in Figures 5.12a and 5.12b. However, it is still not clear why regularization leads to slower, and more accurate, predictions; this issue is a matter of future research.



(a) $L_0 = 10$ **(b)** $L_0 = 20$

Figure 5.11.: Effect of the regularization penalty α and the minimum window length L_0 on the error on the predicted steady-state for the TMS process



(a) $L_0 = 10$ **(b)** $L_0 = 20$

Figure 5.12.: Effect of the regularization penalty α and the minimum window length L_0 on the time required before a steady-state prediction is obtained

Since these experiments were performed without measurement noise, the error in the predicted steady-state is largely due to the fact that the used dynamic model is not able to properly describe the dynamics of the data in the regression window. Using a larger regression window usually results in better predictions due to noise filtering. However, if the dynamic model at hand is not able to describe the data, the bias error increases as the regression window grows, which also results in wrong predictions. Note how worse steady-state predictions are obtained when using a larger minimum window size.

The results of these experiments are summarized in Table 5.5, where the number of sample points $\bar{k}_{\text{prediction}}$ required for a satisfactory steady-state prediction, as well as the error $\bar{E}_j$ in the predicted steady-state, are expressed as a mean quantity with standard deviation. The results for the case where no regularization is used ($\alpha = 0$) are also shown. For the case of $L_0 = 20$ and $\alpha = 0$ one of the predicted values was extremely high ($J > 10^{10}$), and was not included in the calculation of the mean error. However, this high error resulted in convergence to an objective function $J^* = 772.9$ (Euro/kmol) that is significantly far away from the true optimum.

The error in the predicted steady-state can be also attributed to the convergence criteria used in the windowing algorithm: even though the convergence tolerances were set relatively tight (0.1% for the mass balance criterion and

Table 5.5.: Effect of the regularization parameter α on the prediction of the steady-state for the TMS process

L_0	α	$\bar{E}_j$ (%)	$\bar{k}_{\text{prediction}}$
10	0	0.36±0.52	57.6±16.8
	10^{-10}	0.23±0.4	41.1±15.3
	10^{-8}	0.17±0.43	43.1±17.5
	10^{-6}	0.08±0.14	62.2±22.1
20	0	0.35±0.66	56.8±18.1
	10^{-10}	0.53±1.1	38.1±15.2
	10^{-8}	0.18±0.45	45.4±14.1
	10^{-6}	0.24±0.55	61.9±21.4

0.2% for the convergence condition), satisfaction of the mass balance criterion is not a sufficient condition to guarantee that the predicted steady-state is correct. For this case-study, the mass balance criterion corresponds to a closure condition that depends on the sum of seven quantities; it is clear that many combinations of incorrect predictions can appear to satisfy the criterion. If the flow of the product stream F_p were known, the constraints in the reconciliation problem 5.15 could be used as more strict mass balance criteria that the predicted steady-state outputs must fulfill.

In order to asses the advantage of using transient data for the TMS process, three simulation cases were evaluated:

- Using the full linear model for steady-state prediction, with a regularization penalty $\alpha = 10^{-6}$, and a minimum window size $L_0 = 10$.
- Using a mixed approach, where both the linear and nonlinear model are used simultaneously, with a regularization penalty $\alpha = 10^{-6}$, and a minimum window size $L_0 = 10$. The predictions of the model that first (in terms of sampling times) returns a satisfactory steady-state are used.
- Not using a dynamic model, but defining steady-state at the point where the absolute change in *all* the outputs is less than $\epsilon = 0.001\%$. The measurements at that point were used to obtain the flow of product F_p by solving the data reconciliation problem.

Table 5.6 summarizes the performance of MAWQA for the TMS process when the three different described approaches are used. The absolute prediction error

$|E_J|$ corresponds to the difference between the true objective function, and its predicted or estimated value. This value was calculated for the iterations before convergence, and is presented as a mean value ± standard deviation. The three approaches have similar prediction errors, given their standard deviations; however, the comparison is not completely fair, since all three approaches result in different optimization trajectories. Note that even in the case where "static" measurements are used, the predicted steady-state is not perfect, which shows how sensitive the objective function is to small errors in the steady-state values. This is partially due to error propagation, since the objective function has a target concentration in the denominator. The calculated flow of product is also affected by the predicted steady-state, as previously discussed, which makes the objective function less robust to errors.

The table also reports the time $t_{5\%,\text{first}}$ required for the objective function to be in a range of $J_p^* \pm 0.05(J_m^* - J_p^*)$ around the optimal cost, and the time $t_{5\%,\text{last}}$ required for the plant to stay completely within this range. The amplitude of the range was not defined in terms of the absolute value of the cost function, but in terms of the improvement achievable by transitioning from the nominal optimum into the true optimum. The reason for this, is that the absolute value of the objective function is irrelevant for the optimization algorithm: one can always add a constant term to the objective function without affecting the optimization procedure. This observation also applies for defining the relative measurement noise level: the changes in the objective function, and not its absolute value as such, should be appreciable regardless of the noise level.

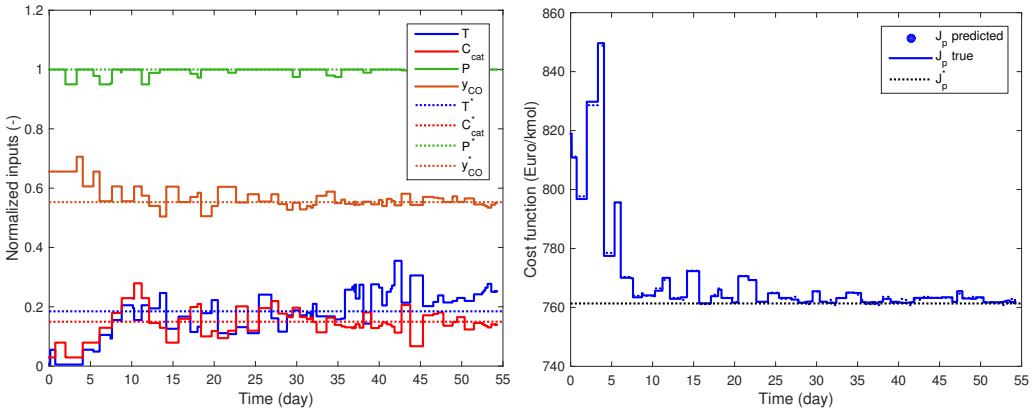
The results indicate that using system identification results in an improvement in terms of optimization time with respect to the case where no dynamic model is used. Although the difference is only significant in the $t_{5\%,\text{first}}$, this is still an advantage of using the dynamic model, since the plant starts operating sooner with a better performance. The results for the mixed approach are noteworthy, since it significantly reduces the time required to stay in a neighborhood of the optimum; doing it in almost half of what it takes when no system identification is used. In this mixed approach, the nonlinear model was used in 12 out of the 22 RTO iterations required for convergence. Note that this particular system is extremely slow, and in reality slow disturbances can alter the behavior of the plant before the optimum is reached.

Table 5.6.: Optimization of the TMS process with dynamic system identification

Model	Linear	Mixed	Static
Optimal cost (Euro/kmol)	761.84	762.33	761.67
Optimal temperature (°C)	86.71	85.90	90.04
Optimal catalyst dosage (ppm)	0.58	0.6	0.49
Optimal pressure (bar)	3	3	3
Optimal CO fraction ()	0.55	0.55	0.55
$\lvert E_J \rvert$ (Euro/kmol)	0.26 ± 0.79	0.49 ± 1.16	0.56 ± 0.35
$t_{5\%,\text{first}}$ (day)	5.6	5,1	7.6
$t_{5\%,\text{last}}$ (day)	22	12.7	22.6

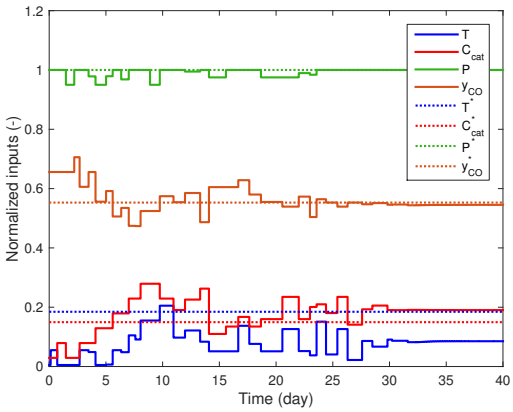
The three approaches converge to a value within 0.5% of the true optimum $J_p^* = 761.34$ (Euro/kmol), but do not exactly reach it due to the errors in the predicted steady-state. Note that all of the three approaches converge to the optimal value of the pressure and the CO fraction, but converge to a subobtimal temperature and catalyst dosage. However, this is not of importance since the objective function achieved is close to the optimal one, i.e. the objective function appears to be somewhat insensitive with respect to these inputs. The temporal evolution of the set-points and the objective function for the three approaches is illustrated in Figure 5.13.

The figures on the right indicate that even though the optimization with "static" measurements reaches a neighborhood of the optimum in a time similar to that of the linear model, it keeps changing and does not definitely settle. When using a linear model, the plant settles to a fixed point after 27 days, while for the mixed approach, the plant settles in around 14 days. Surprisingly, the faster convergence for the cases where system identification is used also seems to be enhanced by the error in the predicted steady-state objective function (blue dots). There is a close relation between the objective values used in MAWQA, and the trajectory it generates, as discussed in Section 5.2.3. This interplay can be either counter-productive, or beneficial. An option to reduce the impact of wrong predictions on the optimization algorithm is to introduce a restriction on the allowed "age" of the points that are used for regression, as proposed in (Wenzel et al., 2015).

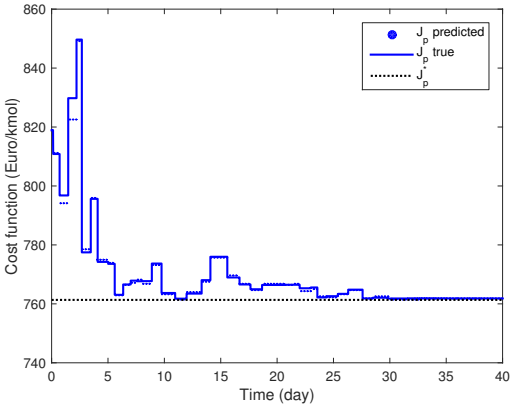


(a) Static optimization

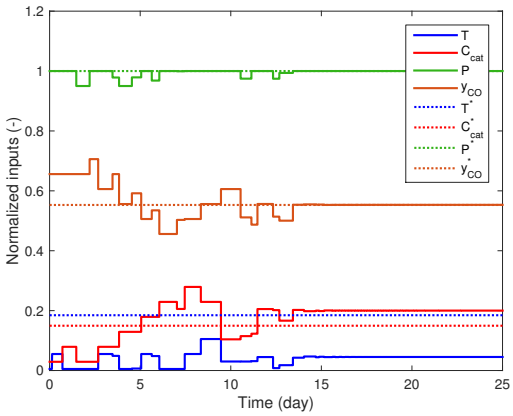
(b) Static optimization



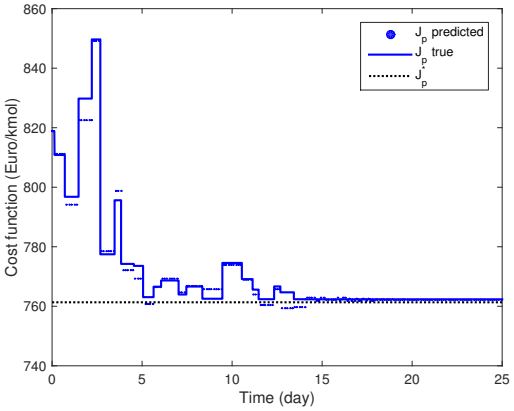
(c) Linear model



(d) Linear model



(e) Mixed approach



(f) Mixed approach

Figure 5.13.: Optimization of the TMS process with static data, predictions from a linear model, and predictions from a mixed approach. Figures on the left correspond to the temporal evolution of the set-points (normalized), while figures on the right correspond to the temporal evolution of the objective function of the plant

In general, the truly optimal trajectory is not known, which leads to the following important conclusion: even if using a dynamic model for steady-state prediction can reduce the time spent at each set-point, the error in the predicted values can lead to a trajectory that is, in the long run, slower than simply waiting for steady-state at each set-point. Conversely, it might also be possible that the wrong predictions lead to a faster trajectory. This uncertainty can only be addressed by using the right convergence criteria for the predicted steady-state.

The figures on the left illustrate what was mentioned on the discussion of Table 5.6, which is that in all of the three cases, the pressure and the CO fraction converge to a close neighborhood of their optimal values, while the catalyst dosage and the temperature do not. Because there is no explicit penalty for pressure, such as the cost of compression, in the objective function, the optimizer sets it as high as possible: a higher pressure results in higher concentrations of dissolved gas, and in higher tridecanal yields. The effect of the CO fraction on the objective function can also be explained, since too much CO will reduce the concentration of active catalyst, while too less CO will reduce the reaction rates.

5.4. Additional remarks on the total optimization time

A way of possibly reducing the optimization time further, would be to first optimize the system for certain privileged inputs, as proposed in (Costello et al., 2016). Since this reduced problem is a particular case of the full optimization problem, its optimal solution J_{sub} has to fulfill $J_m^* \geq J_{\text{sub}}^* \geq J_p^*$, indicating that a partial (and possibly faster) improvement can be achieved. The privileged inputs can be chosen based on a sensitivity analysis of the system: for the TMS case, the nominal optimum includes pressure at its upper bound, which means that one can assume that reducing the pressure will increase the cost function. Initially, one would solve an RTO problem where the pressure is kept at its nominal value; from this point, the full optimization problem can be continued. In general, reducing the number of inputs to optimize will reduce the number of

perturbations needed for gradient estimation. This is particularly advantageous in the context or MAWQA, where the number of parameters for the QA scales quadratically with the number of inputs.

Reducing the input space can be considered a passive strategy for accelerating the optimization procedure, since it only operates on the RTO layer, but does not deal with the slow dynamics of the plant. An active strategy for accelerating convergence, is to design input profiles that minimize the time required for the the plant to settle to steady-state, which can be done via Model Predictive Control (MPC). This approach would have the following advantages:

- Constraint violation can be monitored and handled during the transient phase.
- The input profile can be designed by explicitly taking into account economic considerations during the transient phase.
- The non-constant inputs during an RTO iteration can provide enough excitation to identify a complete NARX (or NARMAX) model, from which the steady-state gradients can be directly extracted.

However, such a scheme can be complicated to implement, since the MPC would use a model that is inherently mismatched. A possible solution would be to use the same black-box model that is used for steady-state prediction. Since the model is adapted online, the controller can make use full use of the measured data, even with structural mismatch of the nominal model.

These two approaches represent interesting research possibilities. Note that the active approach with MPC can also be more realistic, in the sense that the dynamics of the controllers are considered.

CHAPTER

SIX

CONCLUSIONS

In this work, two different approaches for using transient measurements in the context of RTO were presented, discussed, and evaluated. The first method is suitable for processes with parametric uncertainty, while the second one is suitable for processes with structural plant-model mismatch. Therefore, a simple method to evaluate the type of plant-model mismatch based on transient data was discussed. This technique is based on analyzing the resulting residuals after a parameter estimation procedure with a χ^2 test. It was shown that the test can be easily applied, and its potential drawbacks were also discussed. However, determining the type of mismatch of a model is still an open issue that requires further research; especially when limited data for model validation is available.

For the case of parametric plant-model mismatch, an RTO scheme that integrates online parameter estimation based on the sensitivities of the nominal dynamic model was introduced. The simulation studies performed on the case-study of a CSTR demonstrated the usefulness and efficiency of this approach, even in the presence of measurement noise. However, it was also shown that the scheme can lead to wrong parameter estimates when the process approaches steady-state. To solve this issue, a mixed approach where the optimizer switches to static optimization depending on the excitation of the system was proposed as a possible research topic. The online parameter estimation scheme was also shown to be a particular case of a Gauss-Newton algorithm for NLS problems,

and was reformulated as a dynamic difference equation; this extension can allow for accurate parameter estimation, even if the dynamic model is not linear in the mismatched parameters. The evaluation of this approach, and theoretical proofs of convergence and consistency for the obtained parameters are a topic of future research.

For the case of structural plant-model mismatch, the work of Gao and Engell (2016) was extended by incorporating model selection techniques from the field of nonlinear system identification. The proposed approach was evaluated with the case-study of a CSTR, where it was shown that using a taylored nonlinear model that better represents the dynamics of the system can result in faster predictions of the steady-state, than when only a linear model is used. Given that the nonlinear model is selected with little data, there is no guarantee that it will always outperform the linear model in terms of prediction time. For this reason, a mixed approach where both models are used simultaneously was proposed; in this way, the predictions from the first model to return a satisfactory steady-state are used, which can significantly reduce the optimization time. Extensive simulations were also carried out with added measurement noise, which demonstrated the robustness of the method.

The mixed identification approach was further tested on the homogeneously catalyzed hydroformylation of 1-dodecene, which displays a long settling time due to the presence of a recycle stream. It was shown that a considerable reduction in optimization time could be achieved by exploiting transient data with system identification. Although exact convergence to the optimum was not achieved, the scheme was able to drive the plant to a close neighborhood of the optimum in a relatively short period of time. The simulation results also suggested that there is a complex interplay between the steady-state prediction method, and the optimization algorithm (MAWQA in this work): there is an intrinsic trade-off between obtaining faster predictions, and obtaining accurate predictions. Possible solutions for this problem were proposed and discussed.

BIBLIOGRAPHY

Bamberger, W. and Isermann, R. (1978). Adaptive on-line steady-state optimization of slow dynamic processes. *Automatica*, 14(2):223–230.

Bazaraa, S., Sherali, H., and Shetty, C. (1992). *Nonlinear programming: theory and algorithms*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 2 edition.

Biegler, L. T., Lang, Y., and Lin, W. (2014). Multi-scale optimization for process systems engineering. *Computers and Chemical Engineering*, 60:17–30.

Billings, S. (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons.

Billings, S., Chen, S., and Korenberg, M. (1989). Identification of MIMO non-linear systems using a forward-regression orthogonal estimator. *International Journal of Control*, 49(6):2157–2189.

Broyden, C. (1965). A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593.

Brunsch, Y. and Behr, A. (2013). Temperature-controlled catalyst recycling in homogeneous transition-metal catalysis: Minimization of catalyst leaching. *Angewandte Chemie International Edition*, 52(5):1586–1589.

Brydyś, M., Ellis, J., and Roberts, P. (1987). Augmented integrated system optimisation and parameter estimation technique: derivation, optimality and convergence. *IEE Proceedings*, 134(3):201–209.

Brydyś, M. and Tatjewski, 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*, pages 249–254.

Bunin, G. A. (2014). On the equivalence between the modifier-adaptation and trust-region frameworks. *Computers and Chemical Engineering*, 71:154–157.

Bunin, G. A., François, G., and Bonvin, D. (2013). From discrete measurements to bounded gradient estimates: A look at some regularizing structures. *Industrial & Engineering Chemistry Research*, 52(35):12500–12513.

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

Chen, S. and Billings, S. (1989). Representations of non-linear systems: the NARMAX model. *International Journal of Control*, 49(3):1013–1032.

Costello, S., François, G., and Bonvin, D. (2016). A directional modifier-adaptation algorithm for real-time optimization. *Journal of Process Control*, 39:64–76.

Darby, M. L., Nikolaou, M., Jones, J., and Nicholson, D. (2011). Rto: An overview and assessment of current practice. *Journal of Process Control*, 21(6):874–884.

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

Falsone, A., Piroddi, L., and Prandini, M. (2015). A randomized algorithm for nonlinear model structure selection. *Automatica*, 60:227–238.

Farina, M. and Piroddi, L. (2010). An iterative algorithm for simulation error based identification of polynomial input-output models using multi-step prediction. *International Journal of Control*, 83(7):1442–1456.

Farina, M. and Piroddi, L. (2012). Identification of polynomial input/output recursive models with simulation error minimisation methods. *International Journal of Systems Science*, 43(2):319–333.

Forbes, J., Marlin, T., and MacGregor, J. (1994). Model adequacy requirements for optimizing plant operations. *Computers and Chemical Engineering*, 18(6):497–510.

François, G. and Bonvin, D. (2014). Use of transient measurements for the optimization of steady-state performance via modifier adaptation. *Industrial & Engineering Chemistry Research*, 53(13):5148–5159.

François, G., Srinivasan, B., and 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.

François, G., Srinivasan, B., and Bonvin, D. (2012). Comparison of six implicit real-time optimization schemes. *Journal européen des systémes automatisés*, 46:291–305.

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

Gao, W. and Engell, S. (2016). Using transient measurements in iterative steady-state optimizing control. In *Proceedings of the 26th European Symposium on Computer Aided Process Engineering*, pages 511–516.

Gao, W., Wenzel, S., and Engell, S. (2016). A reliable modifier-adaptation strategy for real-time optimization. *Computers and Chemical Engineering*, 91:318–328.

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

Hentschel, B., Kiedorf, G., Gerlach, M., Hamel, C., Seidel-Morgenstern, A., Freund, H., and Sundmacher, K. (2015). Model-based identification and experimental validation of the optimal reaction route for the hydroformylation of 1-dodecene. *Industrial & Engineering Chemistry Research*, 54(6):1755–1765.

Hernández, R. and Engell, S. (2016). Modelling and iterative real-time optimization of a homogeneously catalyzed hydroformylation process. In *Proceedings of the 26th European Symposium on Computer Aided Process Engineering*, pages 1–6.

Kelly, J. and Hedengren, J. (2013). A steady-state detection (SSD) algorithm to detect non-stationary drifts in processes. *Journal of Process Control*, 23(3):326–331.

Korenberg, M., Billings, S., Liu, Y., and McIlroy, P. (1988). Orthogonal parameter estimation algorithm for non-linear stochastic systems. *International Journal of Control*, 48(1):193–210.

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

Kukreja, S., Galiana, H., and Keaney, R. (1999). Structure detection of NARMAX models using bootstrap methods. In *Proceedings of the 38th Conference on Decision & Control*, pages 1071–1076.

Lakshminarayanan, S. and Takada, H. (2001). Empirical modelling and control of processes with recycle: some insights via case studies. *Chemical Engineering Science*, 56(11):3327–3340.

Leontaritis, I. and Billings, S. (1985a). Input-output parametric models for non-linear systems - part I: deterministic non-linear systems. *International Journal of Control*, 41(2):303–328.

Leontaritis, I. and Billings, S. (1985b). Input-output parametric models for non-linear systems - part II: sochastic non-linear systems. *International Journal of Control*, 41(2):329–344.

Lin, J., Han, C., Roberts, P., and Wan, B. (1989). New approach to stochastic optimizing control of steady-state systems using dynamic information. *International Journal of Control*, 50(6):2205–2235.

Ljung, L. (1999). *System Identification: Theory for the user*. Prentice Hall PTR, 2 edition.

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

Marchetti, A., Chachuat, B., and Bonvin, D. (2009). Modifier-adaptation methodology for real-time optimization. *Industrial & Engineering Chemistry Research*, 48(13):6022–6033.

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

Nelles, O. (2001). *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer Science & Business Media.

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 5(3):657–682.

Piroddi, L. and Spinelli, W. (2003). An identification algorithm for polynomial NARX models based on simulation error minimization. *International Journal of Control*, 76(17):1767–1781.

Ramsay, J., Hooker, G., Campbell, D., and Cao, J. (2007). Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796.

Roberts, P. (1979). An algorithm for steady-state system optimization and parameter estimation. *International Journal of Systems Science*, 10(7):719–734.

Rodríguez-Blanco, T., Sarabia, D., and De Prada, C. (2016). Modifier-adaptation approach to deal with structural and parametric uncertainty. *IFAC-PapersOnLine*, 49(7):851–856.

Savitzky, A. and Golay, M. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639.

Schäfer, E., Brunsch, Y., Sadowski, G., and Behr, A. (2012). Hydroformylation of 1-dodecene in the thermomorphic solvent system dimethylformamide/decane. Phase behavior-Reaction performance-Catalyst recycling. *Industrial & Engineering Chemistry Research*, 51(31):10296–10306.

Shokri, S., Hayati, R., Marvast, M., Ayazi, M., and Ganji, H. (2009). Real time optimization as a tool for increasing petroleum refineries profit. *Petroleum & Coal*, 51(2):110–114.

Skogestad, S. (2000). Plantwide control: The search for the self-optimizing control structure. *Journal of Process Control*, 10(5):487–507.

Srinivasan, B., Biegler, L., and Bonvin, D. (2008). Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function. *Computers and Chemical Engineering*, 32(3):572–579.

Tatjewski, P. (2002). Iterative optimizing set-point control - the basic principle redesigned. In *Proceedings of the 15th IFAC World Congress*, volume 35.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Wenzel, S., Gao, W., and Engell, S. (2015). Handling disturbances in modifier adaptation with quadratic approximation. *IFAC-PapersOnLine*, 48(25):132–137.

Zagajewski, M., Behr, A., Sasse, P., and Wittmann, J. (2014). Continuously operated miniplant for the rhodium catalyzed hydroformylation of 1-dodecene in a thermomorphic multicomponent solvent system (TMS). *Chemical Engineering Science*, 115:88–94.

Zhang, H. and Roberts, P. (1990). On-line steady-state optimisation of nonlinear constrained processes with slow dynamics. *Transactions of the Institute of Measurement and Control*, 12(5):251–261.

APPENDIX

ONE

MODEL OF THE TMS PROCESS

The model of the reactor in the TMS process consists of material balances for the different species in the liquid phase (both liquid, and dissolved gases), by taking into account gas solubility and reaction rates. All of the symbols used in the model are summarized and explained in Table A.1.

$$V_R \frac{\mathrm{d}C_i}{\mathrm{d}t} = \dot{V}_{\mathrm{in}} C_{i,\mathrm{in}} - \dot{V}_{\mathrm{out}} C_{i,\mathrm{out}} + V_R C_{cat} M_{cat} \sum_{l=1}^{n_{react}} \nu_{i,l} r_l \tag{A.1}$$

$$V_R \frac{\mathrm{d}C_j}{\mathrm{d}t} = -k_{\mathrm{eff}}(C_l - C_j^{eq}) + \dot{V}_{\mathrm{in}} C_{j,\mathrm{in}} - \dot{V}_{\mathrm{out}} C_{j,\mathrm{out}} + V_R C_{cat} M_{cat} \sum_{l=1}^{n_{react}} \nu_{j,l} r_l \tag{A.2}$$

The interested reader is referred to (Hentschel et al., 2015) for a comprehensive description of the reaction kinetics. The equilibrium concentration of the gases in the gas-liquid interface was calculated with a temperature-dependent Henry law:

$$C_j^{eq} = \frac{P_j}{H_{j,0} \exp\left(-E_j/RT\right)} \tag{A.3}$$

The concentration of the active catalyst depends on the concentration of catalyst precursor and dissolved gases via:

$$C_{cat} = \frac{C_{Rh,\mathrm{precursor}}}{1 + K_{cat,1} C_{CO} + K_{cat,2} C_{CO}/C_{H_2}} \tag{A.4}$$

In the decanter, the different species distribute in the two liquid phases according to:

Table A.1.: Symbols and parameters of the TMS model

Symbol	Description
C_i	Concentration of DMF (1), *n*-decane (2), 1-dodecene (3), iso-dodecene (4), dodecane (5), tridecanal (6), and iso-aldehyde (7)
C_j	Concentration of CO (1) and H_2 (2)
C_j^{eq}	Equilibrium concentration of CO and H_2 at the gas-liquid interface
V_R	Reactor volume
$\dot{V}_{\text{in}}$	Volumetric inflow rate
$\dot{V}_{\text{out}}$	Volumetric outflow rate
C_{cat}	Concentration of active catalyst
M_{cat}	Molar mass of the catalyst
$\nu_{i,l}$	Stoichiometric coefficient of species i in reaction l
r_l	Rate of reaction of reaction l
k_{eff}	Overall gas-liquid mass transfer coefficient
P_j	Partial pressure of species j
T	Temperature in the reactor
E_j	Activation energy of species j
R	Ideal gas constant
$H_{j,0}$	Henry coefficients
$C_{Rh,\text{precursor}}$	Concentration of the catalyst precursor
$K_{cat,1-2}$	Equilibrium constants for the catalyst
T_{decanter}	Temperature in the decanter
$A_{i,0-2}$	Coefficients for liquid-liquid equilibrium in the decanter
$n_{i,\text{product}}$	Molar flow of component i in the product stream
$n_{i,\text{recycle}}$	Molar flow of component i in the recycle stream
$n_{i,\text{decanter}}$	Molar flow of component i in the decanter inlet stream

$$n_{i,\text{product}} = \frac{K_i}{1+K_i} n_{i,\text{decanter}} \tag{A.5}$$

$$n_{i,\text{recycle}} = \frac{1}{1+K_i} n_{i,\text{decanter}} \tag{A.6}$$

Where the partition coefficients are calculated from the following thermodynamic correlation (Schäfer et al., 2012):

$$K_i = \exp\left(A_{i,0} + \frac{A_{i,1}}{T_{\text{decanter}}} + A_{i,2} T_{\text{decanter}}\right) \tag{A.7}$$

APPENDIX
TWO

EIDESSTATTLICHE VERSICHERUNG

Eidesstattliche Versicherung

_____ _____

Name, Vorname Matr.-Nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit/Masterarbeit* mit dem Titel

_____

_____

_____

selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

_____ _____

Ort, Datum Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -)

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird gfls. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

_____ _____

Ort, Datum Unterschrift