

Batch-to-batch learning for
model-based control of
process systems with
application to cooling
crystallization

Marco Forgione

**BATCH-TO-BATCH LEARNING FOR
MODEL-BASED CONTROL OF
PROCESS SYSTEMS WITH
APPLICATION TO COOLING
CRYSTALLIZATION**

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op

1 juli 2014

door

Marco FORGIONE

Master of Science in Computer Engineering
geboren te Varese, Italië

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. P.M.J. Van den Hof

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. P.M.J. Van den Hof,	Technische Universiteit Delft, promotor
Dr. ir. X. Bombois,	Technische Universiteit Delft, copromotor
Prof. dr. R. Babuska,	Technische Universiteit Delft
Prof. dr. B. Wahlberg ,	KTH - The Royal Institute of Technology
Prof. dr. ir. A.C.P.M. Backx,	Technische Universiteit Eindhoven
Dr. ir. H.J.M. Kramer,	Technische Universiteit Delft
Dr. ir. P.J. Daudey,	Albemarle Catalysis

disc

This dissertation has been completed in partial fulfillment of the requirements of the dutch institute of systems and control (disc) for graduate study.



Sustainability by Innovation in Processing

ISPT

The work presented in this thesis has been supported by the Institute for Sustainable Process Technology (ISPT) under the project "PH-00-04: Intelligent Observer and Control for Pharmaceutical Batch Crystallization"

ISBN: 978-94-6203-605-5

Copyright © 2014 by Marco Forgione.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the copyright owner.

Printed in The Netherlands.

Author's email: marco.forgione1986@gmail.com

Acknowledgments

This thesis is the result of four years of research carried out at the Delft Center of Systems and Control (DCSC) of the Delft University of Technology. Looking back to these years, I feel deeply indebted to several people who helped me in different ways to make all this happen.

First of all, I would like to express sincere gratitude to my supervisors Paul Van den Hof and Xavier Bombois. Thank you Paul for giving me the opportunity to be a PhD candidate in DCSC. It has been one of the greatest challenges in my life so far and now I am looking forward to tackle new ones using the competences and skills that I have developed here. Thank you for directing my research without imposing your point of view and for reminding me to maintain a broad perspective while focusing on specific problems. Xavier, I would have never managed to finalize this PhD without your invaluable scientific and human support. Thank you for your supervision, friendship, and care during the good and the difficult times of these years. Thank you for helping me organizing my thoughts when I was confused and for cooking delicious Italian food when you invited me for lunch at your place. You are for me an outstanding professional and moral example to follow.

I am grateful to all the colleagues in DCSC. Working in close contact with you has been at the same time stimulating from an intellectual perspective, and a lot of fun. I would like to mention Ali and Anna for being super office mates, Andrea for showing me around when I had just arrived, Max for being an valid collaborator and for the help with the Dutch translation of this thesis, Aleksandar for the evenings out in Rotterdam, Amol for his wise life and research tips, Dario for the long march in Montréal and the road trip to Hana, Pawel for the (failed) road trip to Stuttgart, Roland for useful scientific discussions, Yashar for inviting me to his place several times, Ilya for his special sense of humor and for always calling for lunch, Mohammed, Renshi, Sadegh, and Pato for the exciting after-lunch foosball matches. Furthermore, I want to thank the secretaries of the department for being very professional and efficient and for always replying to my questions with a kind smile.

I am also grateful to the members of the ISPT project PH-00-04, in particular to Jochem and Somnath for helping me to get on track during my the first year and to Peter, Huub, Pieter, Juan, Jan, and Georgios for their excellent support during the experimental campaign at DSM.

A special thanks goes to the several generations of housemates in Vrouwenrecht 48, and also to Ruud *de buurman*. Thank you Luca, Markus, Alessandro, Marc, Shalini, Donato, Paola, Philip, Marie, Larissa, Meropi, Tiziano, Vicky, Euphy, Soto, Leondardo, and Dhariyash for always paying the rent right on time and for being my “temporary family” in the last four years. Besides a house and a garden, we shared a wide range of feelings, emotions, and life experiences that I cannot describe in the few lines of these acknowledgments: it would just take another book! Philip, I want to believe that you will be following my defence and making a lot of fun of me from somewhere above, and that one day we will toast together with another glass of *volle melk*.

Ringrazio i tanti italiani conosciuti in questi anni in Olanda per aver condiviso ed arricchito una simile esperienza di vita. Grazie in particolare a Samy per i grandiosi weekend di svago in giro per l’Europa.

Grazie anche gli amici in Italia e in particolare Massimo e Jacopo per avermi sempre cercato, anche quando lunghi periodi ero praticamente irreperibile. Ringrazio pure Attilio anche se non è mai passato a trovarmi e Bamba anche se si fa sentire con il contagocce.

Ringrazio infine i miei genitori per avere accettato pur soffrendo la mia scelta di vivere tanti anni lontano da casa e mi scuso di tornare a visitarli soltanto di rado. Li ringrazio inoltre per avermi donato i miei fratelli Andrea e Paola, i tesori più preziosi della mia vita.

Pijnacker, May 2014
Marco Forgione

Contents

Acknowledgments	vii
1 Introduction	1
1.1 Process engineering and process control	1
1.2 Continuous and batch processes	3
1.3 Control-relevant aspects of batch processes	5
1.4 Control of batch processes	6
1.5 Identification of batch processes	9
1.6 Batch cooling crystallization	11
1.7 Control of batch cooling crystallization	12
1.8 Problem statement	13
1.9 Approach	14
1.9.1 Development of a parametric model update strategy for batch cooling crystallization	14
1.9.2 Development of a non-parametric model update approach .	15
1.9.3 Validate the batch-to-batch control algorithms	16
1.9.4 Investigate the role of excitation in an iterative identifica- tion/controller design scheme for linear dynamical systems	17
1.9.5 Develop experiment design tools for nonlinear systems . . .	18
1.10 Organization of this thesis	20
1.11 About the project	21
2 The batch cooling crystallization process	23
2.1 Introduction	23
2.2 Principles of Crystallization	24
2.2.1 Chemical solution	24
2.2.2 Objective of the crystallization process	24
2.2.3 Phase diagram	25
2.2.4 Solubility	25
2.2.5 Metastable region and Metastable Zone Width	26
2.2.6 Unstable region and nucleation	27
2.2.7 Unseeded batch and crystal growth	28
2.2.8 Seeded batch	28

2.2.9	Dissolution	29
2.3	Seeded batch cooling crystallization	30
2.3.1	Operation	30
2.3.2	Population balance	32
2.3.3	Mass balance	34
2.3.4	Energy balance	36
2.3.5	Kinetic Relations	36
2.3.6	Reference PBE model	37
2.3.7	Moment model reduction	38
2.3.8	Reference moment model	38
2.3.9	Measured output and control output	39
2.3.10	Model coefficients and scaling	40
2.3.11	Simulations	41
2.3.12	Nonlinearity study	43
2.4	Control of batch cooling crystallization processes	45
2.5	Summary	47
2.6	Symbols and units	47
3	Batch-to-batch control for cooling crystallization	49
3.1	Introduction	49
3.2	Model of the batch cooling crystallization process	53
3.3	Batch-to-batch supersaturation control	54
3.3.1	B2B+PI Configuration	55
3.3.2	Design of the PI controller	55
3.3.3	Nominal cooling trajectory	56
3.3.4	Batch-to-Batch Control	57
3.3.5	Iterative Identification Control	57
3.3.6	Iterative Learning Control	60
3.4	Simulation Results	61
3.4.1	Case 1	63
3.4.2	Case 2	65
3.4.3	Overall Results and Discussion	68
3.5	Iterative control and dual control	69
3.6	Conclusion	71
4	Iterative model improvement for model-based control	73
4.1	Introduction	73
4.2	The Framework	76
4.2.1	Iterative identification	77
4.2.2	Controller design	79
4.2.3	Total cost, modeling error cost, excitation cost	80
4.3	Experiment Design	83
4.4	Receding horizon	85
4.5	Simulation Study	86
4.6	Conclusions	90
5	Experiment design for parameter estimation in nonlinear systems based on multilevel excitation	93
5.1	Introduction	94
5.2	The Framework	97
5.2.1	Data-generating system and model structure	97
5.2.2	Input signal	98

5.2.3	Subsequence patterns	98
5.2.4	Information Matrix	99
5.3	Experiment Design	100
5.3.1	Relative frequencies as design variables	100
5.3.2	Experiment Design Problem	102
5.3.3	Chicken and the egg issue	103
5.3.4	Input Realization	103
5.3.5	Parameter Estimation	103
5.4	Numerical Example	104
5.5	Conclusions	108
6	Experimental Results	111
6.1	Introduction	111
6.2	The experimental setup	112
6.2.1	The instruments	115
6.2.2	Software architecture	116
6.3	Design of the temperature controller	118
6.4	Crystallization experiments using the pure water-succinic system	120
6.4.1	Preliminary data and process design	120
6.4.2	Calibration of the K-Patents	120
6.4.3	Determination of the solubility	123
6.4.4	Unseeded batch cooling experiment	126
6.4.5	Seeded batch experiment	127
6.4.6	Conclusions on the preliminary experiments	127
6.5	Crystallization experiments in the presence of Fumaric acid as impurity	132
6.5.1	Determination of the solubility line in the presence of Fumaric Acid	132
6.5.2	Unseeded batch experiments	133
6.5.3	Seed preparation	133
6.5.4	Nominal seeded batch experiment	134
6.5.5	Conclusions on the experiments in the presence of the impurity	135
6.6	Batch to batch control experiments	139
6.6.1	Control architecture	139
6.6.2	Experimental Procedure	139
6.6.3	Batch to batch supersaturation control experiments	140
6.6.4	Conclusions for the B2B experiments	147
6.7	Conclusions	147
6.8	Symbols and units	148
7	Conclusions	151
7.1	Contribution of this thesis	151
7.2	Recommendations for future research	153
A	Applied Dynamic Optimization	157
A.1	Pontryagin Maximum Principle	158
A.2	Numerical solution using the simultaneous optimization approach	161
A.2.1	Matlab Implementation	162
A.2.2	GAMS Implementation	164
A.3	Numerical solution using the sequential optimization approach	166
A.3.1	Matlab Implementation	166

B Numerical solution of Partial Differential Equations	169
B.1 Problem Formulation	169
B.2 Finite volume discretization scheme	170
B.3 First-order upwind approximation scheme	171
B.4 Second-order approximation schemes	171
B.4.1 Flux limiter function	172
Bibliography	173
List of Abbreviations	183
Summary	185
Samenvatting	187
Curriculum Vitae	189

Introduction

In this chapter, we give a general introduction to the field of process control and we describe the specific challenges and opportunities encountered in the case of batch processes. Following, we present the state-of-the-art control strategies for batch processes. Finally, we introduce the process of batch cooling crystallization, that is the main case study considered in this thesis. This material builds up the settings required to motivate and formulate the research objective dealt with in this thesis.

1.1 Process engineering and process control

From an engineering perspective, the term *process* refers to a conversion of *raw materials* into intermediate or final *products* using chemical, physical, or biological operations (Seborg et al., 2010). In an industrial settings, a process typically consists of several *process steps* such as reactions, distillations, and crystallizations. Each of these process steps takes place in a confined *process unit*. There exist specific process units for all the standard process steps. For instance, the three steps mentioned above are performed in units called reactors, distillation columns, and crystallizers, respectively.

In an industrial process, a number of *process variables* such as temperatures, pressures, and concentrations have to be maintained close to suitable values in order to guarantee safety, maintain product quality, and maximize economic criteria. Control techniques have been applied for a long time in order to achieve these goals. The application of control techniques to process engineering is known as *process control*. Process control techniques are applied at the levels of the single units, the interconnection of the several units composing a process, and the full production plant. The latter case is referred to as *plant-wide process control* (Luyben et al., 1998).

In recent years, the performance requirements in the process industry have become increasingly difficult to satisfy. Strong competition, strict safety regulations, and rapidly changing economic conditions have tightened the product quality specifications and reduced the margins for profits. This has been a drive for the

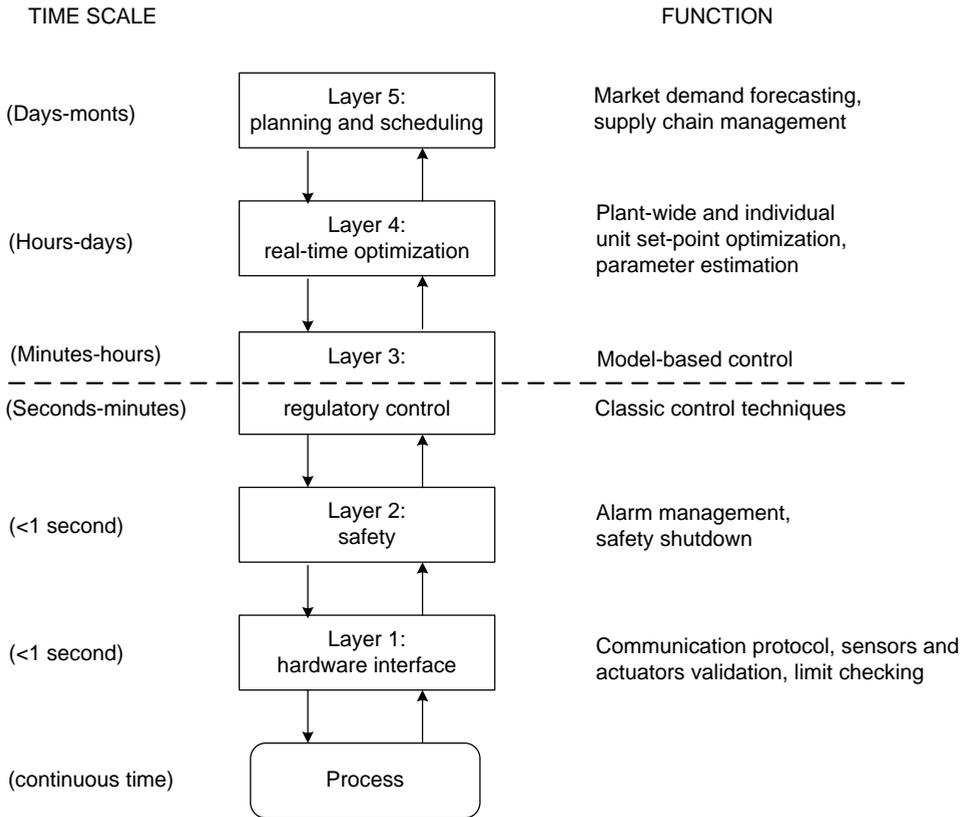


Figure 1.1: The five-layer architecture of modern process control software. Adapted from Nof (2009).

companies to optimize different aspects of their processes.

Process control has also been part of the plant-wide process optimization. Modern process control software architectures have been organized in a pyramidal structure consisting of five interdependent layers (Nof, 2009). The two lowest layers implement the interface with the hardware and safety functionalities. They consist for instance in the software routines used to communicate with sensors and actuators (layer 1), and emergency alarms or shutdown rules triggered by particularly dangerous events (layer 2). These layers should be designed to be as simple as possible in order to guarantee the maximum reliability. The two upper layers perform long-term optimizations based on economic criteria such as planning and scheduling based on market demand forecast (layer 5: planning and scheduling), and set-point optimizations based on steady-state plant models (layer 4: real-time optimization). The middle layer (layer 3: regulatory control) has the regulatory function and aims to track the set-point determined by the upper levels, while rejecting process disturbances. On top of the standard feedback controllers such as Proportional-Integral-Derivative (PID) (Astrom, 1995), more advanced control solutions such as Model Predictive Control (MPC) (Maciejowski

and Huzmezan, 1997) have been adopted in this layer for the loops having the highest impact on the overall process performance. These control strategies have been shown to be able to deliver significant economical improvements, sufficient for the companies to justify the investment costs in these new technologies.

Most of the advanced control strategies, including the celebrated MPC, are classified as *model-based*, in the sense that a (dynamic) model of the to-be-controlled process dynamics is explicitly required for the design of the controller. The accuracy of this model has actually a significant impact on the performance that the controller can deliver.

Unfortunately, the diverse and complex phenomena occurring in industrial processes are sometimes difficult to model with the accuracy that is required for high-performing control. Furthermore, the process dynamics are subject to slow time variations which may reduce over time the performance of a model-based controller, even when the performance was satisfactory with the original plant dynamics. For these reasons, the model development and maintenance have been recognized as the most challenging, time-consuming, and costly tasks required for the successful implementation of a model-based control system, and they are currently the bottleneck for an even more widespread penetration of this technology in the process industry (Van den Hof, 2014). With the recent improvements in the sensor technology and in the IT infrastructures installed in the plants, more and more measurements from the processes are collected and stored in a format that allows convenient elaboration. These large datasets contain important information about the processes and the performance of the operating model-based controllers. It is evident that using these data for a regular, quantitative, and automated performance monitoring and model update could push the capabilities of model-based control systems for process control applications far beyond the current state-of-the-art.

In this thesis, we work towards the development of strategies for the data-based model improvement for industrial processes, where the use of the model is the design of an high-performing model-based controller.

The specific challenges encountered in modeling and control of process systems are strictly related to the two main modes of operation of process engineering: *continuous* and *batch*. In this thesis, we give particular attention to batch operations. We will use as a test case the *batch cooling crystallization* process,¹ which is extensively utilized in the pharmaceutical, fine chemicals, semiconductors, and food industries.

1.2 Continuous and batch processes

Most of the process engineering steps can be performed either in continuous or in batch mode. There exist for instance continuous or batch reactors, distillation columns, and crystallizers.

¹In practice, batch crystallization is usually applied as a step in more complex process possibly consisting of multiple reactions, crystallizations, filtration, etc. Therefore, it would be more accurate to speak of the batch cooling crystallization step. However, hereafter we refer to the batch cooling crystallization process for simplicity.

In continuous mode, the raw materials are continuously introduced into the unit and the products are continuously removed from the unit. The system is operated at a steady-state condition where the inflow of the raw materials is equal to the outflow of the products, while the transformation takes place continuously inside the unit.²

Conversely, in batch mode, the raw materials are loaded in the unit only at the beginning of the process. Subsequently, the desired transformation takes place inside the unit and the final products are removed altogether after the processing time. In order to achieve the desired production volume, more batches are repeated over time.

Continuous and batch operations have complementary advantages and disadvantages. A first aspect to be considered is the scale of the production. Continuous operations are usually preferable for large-scale productions at a constant rate. Indeed, a continuous process unit is constantly operated around the steady state condition for which it is designed, providing in this condition the maximum efficiency. However, the costs for the design, construction and operation of a continuous unit are generally higher than the ones of a batch unit. These costs are not compensated by the higher efficiency when the production volumes are not high enough and batch operations are preferred in this case. According to Bonvin (2006), continuous processes are required for commodities whose break-even point is on the order of 100000 metric tons per year, while batch processes are attractive for production volumes below 10000 metric tons per year.

Batch operations offer other advantages over continuous ones in terms of flexibility of the production. For certain products, there is a market demand that changes over time or has a seasonal trend. In the case of batch units, it is easy to adjust the production volume simply by repeating the operation a different number of times. Conversely, in the case of continuous units, changes in the production rate may be limited by design, while the shutdown and the startup of the unit may be complex, expensive procedures.

For products that require repeated process steps, the same batch unit can be shared by more steps, while a continuous design would generally require a separate unit for each step. For instance, in the pharmaceutical industry the purification of a drug may require several crystallization steps. The same batch crystallization unit can be used for different crystallization steps.

Batch units can also be easily converted for a different product or even be shared between different products, while continuous units are generally product-specific. Finally, batch processes are often preferred when isolation and lot integrity are necessary for reasons of sterility or safety. Indeed, it is straightforward in batch processes to document and keep track of each lot of raw material that makes up each lot of product (Korovessi and Linninger, 2005).

For all the reasons above, high-volume industries such as petroleum refining typically use continuous operations, while batch operations are far more common in lower-volume industries using specialized types of chemistry, for instance in the pharmaceutical and fine chemicals fields (Tomazi et al., 2006).

²More precisely, in a full process, the inflow of an unit could also consist of intermediate products of a previous step, and not necessarily of raw materials.

1.3 Control-relevant aspects of batch processes

It has been observed that batch processes have specific properties that bring both challenges and opportunities for control (Bonvin, 2006). These challenges and opportunities are described in the following and their implication for control are discussed at the end of this section.

1) Repetitive nature. Batch processes are repeated several times. The control objectives are usually the same for all the batches. Furthermore, the process is usually designed to be operated starting from a fixed initial condition.³

2) Wide dynamical range. Batch processes evolve from an initial condition where the raw materials are loaded inside the vessel to a final one where the products are obtained. Therefore, batch processes have to be operated over a wide range of conditions. Due to the wide operational range, batch processes often exhibit severely nonlinear dynamics.

3) Model uncertainty. First-principles models for most batch process units are available in the literature, as well as in specialized software modeling packages. However, these models often depend on uncertain parameters whose actual values are highly uncertain. In other cases, even the structure of the nonlinear equations describing the process is subject to uncertainty.

4) Limited sensing. While measurements of temperatures, pressures and flows are relatively reliable and inexpensive, measurements of other quantities such as concentrations and particle sizes still pose significant technical challenges. In some cases, reliable measurements are obtained from laboratory analysis of data collected during the process. Therefore, they are not available online while the process is running.

5) Limited actuation. Many batch processes have an irreversible and/or history-dependent behavior, and a strong dependency on the initial state that cannot be corrected using the available inputs. Furthermore, the ability to influence the process usually decreases with time, which limits the impact of corrective actions.

6) Disturbances. Different disturbances may affect the normal operation of a batch process. Some disturbances enter the process as the result of upstream variability such as impurities in the raw materials. The thermal evolution of the process may also be subject to disturbances due to, for instance, variations of the temperature of the cooling medium, interaction with other units in the plant, or to thermal losses to the external environment.

³ In practice, obtaining exactly the same initial condition for all the batches may be difficult due to process-specific technical difficulties. Thus, small batch-to-batch variations of the initial condition cannot be completely ruled out.

7) Slow dynamics. Batch processes are usually characterized by fairly slow dynamics. The time constants of process systems are seldom smaller than a few seconds and have usually the order of several minutes.

On the one hand, points 2, 3, 4, 5, and 6 make batch process control particularly challenging. Due to this wide dynamical range (point 2), we deal most of the times with nonlinear control problems. The use of model-based control techniques is complicated by the uncertainty in the models available at hand (point 3). Due to the lack of online reliable measurements (point 4), it may be difficult to detect that a disturbance (point 6) is moving the batch off the desired specification on time. If the disturbance is detected too late during the batch, it may not be possible to compensate for it any more due to actuation limitations (point 5).

On the other hand, points 1 and 7 offer opportunities that alleviate the difficulties discussed above. Batch processes are repeated several times (point 1) and from the analysis of the data coming from previous batches it is possible for instance to compensate for the effect of repetitive disturbances or to refine the model. Furthermore, batch processes usually have large time constants (point 7). Given the computational power available nowadays, this allows for sophisticated online elaborations.

1.4 Control of batch processes

A comprehensive categorization of the established control strategies for batch processes has been presented in Bonvin (2006). In order to describe these strategies, it is useful to introduce the following mathematical description of a generic batch process

$$\begin{aligned} \frac{dx_k(t)}{dt} &= f(x_k(t), u_k(t)), & x_k(0) &= x_{0,k}, \quad t = [0, t_f], k \in \mathbb{N} \\ y_k(t) &= g(x_k(t), u_k(t)), \\ z_k &= z(x_k(t_f)). \end{aligned} \tag{1.1}$$

The variable t is the time, which spans in a finite interval $[0, t_f]$, k is the batch index, and $x_k(t)$ is the state of the system at time t during batch k .⁴

There are two types of to-be-controlled variables, namely the run-time variables $y_k(t)$ and the run-end variables z_k . Run-end variables represent quantities for which only the final value is of interest. For instance, it is possible that in a batch reaction only the final concentration of a product is interesting, and not the intermediate values attained at different time instants. Run-time variables represent quantities that have to be controlled throughout the whole process. For instance, it may be required for a process to follow certain temperature and pressure profiles in order to guarantee safety and product consistency. Generally speaking, run-end variables are measured only at the end of a process, while run-time variables may or may not be measurable online.

⁴We here assume for notational simplicity that all the batches have the same duration t_f . More in general, different batches may have a different duration $t_{f,k}$.

Implementation	Control objective	
	Run-time reference $\bar{y}_k(t)$	Run-end reference \bar{z}_k
Online	1) Online run-time	2) Online of run-end
Batch to batch	3) B2B run-time	4) B2B run-end

Table 1.1: Control strategies for batch process control.

A controller for a batch process adjusts the input $u_k(t)$ in order to follow a run-time variable $y_k(t)$ or a run-end variable z_k . The controller may update the input either online at every time instant of a batch or off-line, i.e. from one batch to the other.⁵ This leads to the following four control strategies (see also Table 1.1).

1) Online control of run-time variables. Let us assume that the run-time variable $y(t)$ is measured online and a reference trajectory $\bar{y}_k(t)$, $t = [0, t_f]$ for the batch is given. At every time instant, the input $u_k(t)$ is computed by an on-line tracking controller K

$$u_k(t) = K(y(t), \bar{y}_k(t)) \quad (1.2)$$

in order to achieve a small tracking error $e_k(t) \triangleq \bar{y}_k(t) - y(t)$, $t = [0, t_f]$ and to satisfy eventual input, output, or states constraints.

Online tracking controllers are also common for continuous processes where the set-point is a constant corresponding to the desired steady-state and the on-line controller is a (linear) feedback controller. Compared to continuous processes, the set-point in the batch case is often a time-varying trajectory instead of a constant. Furthermore, the dynamics of the batch process and the characteristics of the disturbances may change along that trajectory. For these reasons, a linear feedback controller may not be sufficient in order to follow the reference with the required accuracy in the batch case. Strategies such as gain scheduling (Rugh and Shamma, 2000), feedback linearization (Isidori, 1995), and MPC (Maciejowski and Huzmezan, 1997) may be applied when the performance of a linear feedback controller is not satisfactory. Note that feedback linearization and MPC require the knowledge of the state $x_k(t)$. If $x_k(t)$ is not directly measured, a *state estimator* has to be designed as well.

2) Online control of run-end variables. In this case, the objective of the controller is to steer the run-end variables z_k to the desired set-point \bar{z}_k . Since a prediction in the future is required to evaluate how the run-end variable will evolve, this problem is tackled using an MPC-like control law. Let us assume that a run-time variable $y_k(t)$ is measured online. This variable is used to obtain an estimate $\hat{x}_k(t)$ of the current state $x_k(t)$. The input is determined such that the predicted run-end variable \hat{z}_k is as close as possible to the reference \bar{z}_k . Formally, we write

$$u_k(t) = P(\hat{z}(\hat{x}(t)), \bar{z}_k) \quad (1.3)$$

⁵In practice a digital controller can only elaborate discrete-time, digital signals. In this section, the controllers are introduced using a continuous-time notation for simplicity. Note that due to the usually large time constant of batch processes, the effect of the discretization is not critical for most of the known process control applications.

where P is a model-based optimization routine aiming to steer the predicted run-end variable \hat{z}_k to the desired value \bar{z} .

Note that while in the classic MPC approach for a continuous processes the optimization is typically performed over a receding time horizon (Maciejowski and Huzmezan, 1997), for the case of batch processes the optimization is performed over a *shrinking time horizon* that goes from the current time t to the final time t_f .

3) Batch-to-batch control of run-time variables. The objective is to determine the entire trajectory of the input variable $u_{k+1}([0, t_f])$ for batch $k + 1$, based on all the information collected during the previous batches such that the run-time variable $y_{k+1}([0, t_f])$ follows the desired set-point trajectory $\bar{y}_{k+1}([0, t_f])$ at every time instant. In the literature, a family of algorithms known as Iterative Learning Control (ILC) has been developed for this kind of problems (Bristow et al., 2006). ILC is a fairly common technique for control of mechanical systems, but it has not been widely applied in the field of process control to date. Most of the ILC algorithms compute the input for the next batch as a function of the input/output data relative to the previous batch:

$$u_{k+1}([0, t_f]) = u_k([0, t_f]) + L(u_k([0, t_f]), y_k([0, t_f])). \quad (1.4)$$

The update term L can be computed completely model-free as in the PD-type ILC or based on the nominal process model as in the Quadratically-optimal ILC (Q-ILC) (Bristow et al., 2006). ILC algorithms generally require that the initial condition of the system and the set-point are the same for all batches, and that the real-time disturbances are moderate or absent.

4) Batch-to-batch control of run-end variables. The objective is to determine the entire trajectory of the input variable $u_{k+1}([0, t_f])$ based on data from the previous batch such that the run-end variable z_{k+1} follows the reference \bar{z}_{k+1} . In the literature, algorithms such as Run-to-Run (R2R) have been applied to solve this control problem:

$$u_{k+1}([0, t_f]) = u_k([0, t_f]) + R(z_k, \bar{z}_{k+1}). \quad (1.5)$$

The R2R correction adds to the previous input $u_k([0, t_f])$ an update term R that is expected to decrease the difference between z_{k+1} and \bar{z}_{k+1} . The term R could be for instance a gradient-based optimization step with an objective function such as $\|\bar{z}_{k+1} - \hat{z}_{k+1}\|^2$, where \hat{z}_{k+1} is a prediction of the run-end variable computed based on a nominal process model (Wang et al., 2009).

The control strategies presented above can be combined in different ways. For instance, it is possible to combine online and batch-to-batch control for of run-time variables, see e.g. the Batch-MPC algorithm presented in Lee et al. (1999). Other possibilities could be to use an online controller for the run-time variables combined with a batch-to-batch controller for the run-end variables (Lee and Lee, 2003), or to use the online controller for both the run-time and the run-end variables using a multi-objective MPC scheme (Mesbah, 2010). Other combinations of batch control strategies can be found in the literature.

1.5 Identification of batch processes

The control strategies presented in the previous section are based on a nominal model of the batch process. In this model, the value of a number of physical parameters is typically not known with high precision. This results in a severe model uncertainty, which in turn limits the performance of the model-based control strategies. Intuitively, it should be possible to recover the control performance by identifying a model using previous experimental data.

Identification techniques have been used intensively for improving the model-based control of continuous processes, where the systems are operated around a fixed operating point and the dynamics of the deviation around that point are often approximately linear. Well-established identification techniques using linear, black-box model structures (Ljung, 1999) have been applied in order to obtain the models to be used for control.

In the case of batch processes, the systems have to be operated in a wide dynamical range and nonlinear models are almost always required. Even though nonlinear, black-box structures such as Linear Parameter Varying (LPV) could perhaps be used to model the behavior of batch processes (Lakshmanan and Arkun, 1999), this approach has not yet been widely demonstrated in practice. In the majority of the cases, the model structures for batch processes are derived from a first-principles model, leaving the uncertain physical coefficients as free parameters.

In principle, accurate and reliable models could be obtained by estimating these parameters using the measured data from previous batches. However, parameter estimation for batch processes described in first-principles model structures entails several issues that need to be carefully addressed. These issues are described in the following of this section.

1) Lack of identifiability. In the first-principles model structures, the sensitivities of the measured output to the different parameters may differ by orders of magnitude. Furthermore, these sensitivities depend on the particular input used in the identification data set. The effect of the parameters with lower sensitivity may be hidden in measurement noise. Besides, the effect of a subset of parameters may be highly correlated, or even perfectly collinear. This means that changes of the parameters in certain directions may be difficult to detect from the observations. All these characteristics may lead to lack of identifiability issues: it is possible that more than one set of parameters describe the input/output observation exactly. Even when the “true” parameters are theoretically unique, there could exist a very large set of admissible parameters that describe the input/output observation almost equivalently (even close to numerical precision) (Vajda et al., 1989).

Note that in the case of linear dynamical systems, identifiability issues can be circumvented by adopting standard parameterizations of the transfer functions, such as fractions of polynomials with parametrized coefficients. Conditions on the input such that all the parameters in a standard model structure are identifiable are known in the literature (Ljung, 1999). Even more, the input signals can be designed in order to guarantee a certain accuracy for the estimated parameters

using experiment design tools (Goodwin and Payne, 1977; Jansson and Hjalmarsson, 2005). Unfortunately, these strategies are much less mature in the case of nonlinear dynamical systems (see point 5 in this list).

2) Nonlinear optimization. Parameter estimation generally requires the solution of a nonlinear, nonconvex optimization problem. The solution of such a problem can be computationally intense and specific algorithms may be required. Furthermore, it is in general very hard to guarantee that the algorithm finds a global optimum for the problem. From a numerical perspective, having highly correlated parameters with sensitivities ranging orders of magnitude is a further difficulty.

Note that a nonlinear optimization is required for parameter estimation both in the case of linear and nonlinear dynamical systems. However, for the case of linear dynamical systems, specific algorithms have been developed in order to make this optimization more effective. For instance, subspace identification methods (Verhaegen and Dewilde, 1992), which do not require optimization can be used to generate a rather good initial estimate for the nonlinear optimization.

3) Structural model mismatch. The model structures used to describe batch processes are often approximations of a physical reality that is known to be more complex. Therefore, it is likely that the model structure assumed for identification does not contain the true system for any choice of the model parameters. When this is the case, it is obvious that the identified model cannot converge to the true system. We can only ensure that the identified model is close to the best approximation (defined in some way) of the true system within the model structure. Thus, the actual “distance” between the true system and the identified model is due to the distance between the true system and its best approximation within the model structure (bias error), and the distance between the best approximation and the identified model (variance error).

In the case of linear dynamical systems, the effect of a structural model mismatch has been extensively analyzed in the field of Identification for Control (Gevvers, 2005). In the frequency domain, the distance between the transfer function of the true system and the one of the identified model is simply the sum of the distance between the true system and its best approximation (bias error), and the distance between the identified model and the best approximation (variance error). The best approximation of a linear dynamical system for a certain controller design objective can also be characterized by frequency-domain expressions. Conversely, in the case of nonlinear dynamical systems, there is not a general way to characterize the effect of the bias and the variance error.

5) Need for excitation signals. The accuracy of an identified model depends on the level of information contained in the data used for parameter estimation. In principle, it is possible to increase the level of information by adding an excitation signal to the input in the identification dataset (Ljung, 1999). The parameter estimation procedure will lead to a more accurate model when a suitable excitation signal is applied, which in turn will provide a better control performance once

the model is used to re-design the controller. However, the same excitation signal leads to a performance degradation while it is applied since it acts as a disturbance on the controlled system. Thus, there is a trade-off between the performance degradation due to the excitation signal and the performance improvement that can be obtained having a more accurate model.

The problem of designing excitation signals for linear dynamical systems has been considered in the field of Identification for Control. Experiment design tools for linear dynamical (Jansson and Hjalmarsson, 2005) have been used to obtain excitation signals that guarantee a given model accuracy, while results from robust control set the link between model accuracy and performance improvement.

The classic approaches to the design of the excitation signals consider two distinct phases: an identification phase in which the excitation signal is fed to the system and a model is identified, and a control phase in which a controller based on the identified model is applied to the system. The first approach presented in the literature (Gevers and Ljung, 1986) was to design the excitation signal in order to maximize the performance in the control phase, subject to a constraint on the performance degradation due to the excitation signal in the identification phase. More recently, the reasoning has been reversed in the so-called *least costly* approach (Bombois et al., 2006), where the performance degradation due to the application of the excitation signal in the identification phase is minimized, subject to a constraint on the desired performance in the control phase.

These two-phase frameworks do not fit the situation encountered in batch process control, where several batches are repeated, and the identification and control tasks can be performed at the end of each batch. In the batch control case, excitation signals can be added to the input during each batch. The excitation signal for one batch should be designed taking into account the performance degradation in the current batch, as well as the expected improvement in the all following ones that are planned.

Another complication is that batch processes are almost always severely nonlinear. Unfortunately, there is to date a lack of generally applicable, computational efficient Experiment Design tools which can handle nonlinear dynamical systems. The methods in the literature are either tailor-made for very special and simple nonlinear structures (Barker et al., 2004; Hjalmarsson and Mårtensson, 2007; Larsson et al., 2010), or computationally very expensive and, in general, intractable (Franceschini and Macchietto, 2008). Due to these issues, the design of suitable excitation signals for the identification of batch processes is still a challenge.

1.6 Batch cooling crystallization

As mentioned in Section 1.1, in this thesis we will focus on a specific batch process, namely batch cooling crystallization. Crystallization may be defined as a phase change in which a crystalline product is obtained from a fluid, a gas, or a melt (Myerson, 2002). From an industrial perspective, crystallization processes are utilized with different purposes in a wide range of fields. Crystallization is extensively applied as a separation and purification step in the commodity, petrochemical, specialty, fine-chemical, and pharmaceutical industries. Crystallization

is used in the food industry to give products the desired texture and consistency. It is applied for instance in the industrial production of ice cream, butter, chocolate, cheese, and bread (Larsen et al., 2006). In the semiconductor industry, microelectronic devices are created by a large number of steps, most of which involve either etching or growth of crystalline material (Braatz, 2002).

Different kind of crystallization processes are applied in the industrial practice. Batch cooling crystallization is the most common strategy in the pharmaceutical and the fine chemicals industries (Myerson, 2002). In a batch cooling crystallization process, a chemical solution consisting of a solute dissolved into a solvent is loaded at high temperature into a vessel called crystallizer, and is subsequently cooled down. The cooling is performed by circulating a colder medium inside the jackets surrounding the crystallizer. By cooling, the equilibrium concentration (i.e. the solubility) of the solution is lowered, and part of the solute is transferred from the solution to the solid, crystalline phase. Thus, while the solution is cooled, the concentration of the solute in the solution decreases and the amount of solid crystals increases. When the final temperature is reached, the solid, crystalline product is extracted and the batch ends.

Despite the wide diffusion of batch cooling crystallization in the industry, some of the physical phenomena governing this process are still largely uncertain and debated in the scientific community. As a consequence, the nonlinear dynamic models currently used to describe the process suffer from severe uncertainties. For this reason, besides being directly a relevant industrial application, batch crystallization is a serious benchmark for batch control algorithms.

1.7 Control of batch cooling crystallization

The ultimate objective of a batch cooling crystallization process is to produce crystals satisfying certain requirements in terms of size, morphology, shape, purity, etc. In practice, however, direct control of the crystal properties is severely hindered by modeling and measurement issues. Therefore, in most of the cases the crystal properties are only indirectly controlled by specifying the reference trajectories that a number of process variables such as the temperature, the growth rate and the supersaturation⁶ have to follow during the time of the batch. An accurate tracking of these references can guarantee that the crystals obtained at the end of the batch consistently satisfy the desired properties.

Feedback control of the temperature is the most common control strategy for batch cooling crystallizers in an industrial settings (Fujiwara et al., 2005). The desired cooling profile is given as reference to a feedback controller. Since the thermal part of the dynamics is in general fairly linear, a linear feedback controller such as a PI or PID can provide a sufficiently accurate tracking performance. While the temperature control is easy to implement, in some cases controlling only the temperature is insufficient in order to consistently guarantee the desired product quality.

The supersaturation and the growth rate have a more direct influence on the

⁶The physical meaning of the supersaturation and the growth rate will be given in Chapter 2. The exact definition of these quantities is not crucial for the reasoning of this section.

crystallization process. Feedback control of these quantities has been widely investigated in the literature (Nagy et al., 2008) and different strategies have been applied. Unlike the thermal dynamics, the dynamics of the growth and of the supersaturation are severely nonlinear. For this reason, a nonlinear control strategy is required in this case. In Xie et al. (2002) and Vissers et al. (2011), a feedback linearization strategy has been proposed for supersaturation control. An experimental verification of such a scheme has been presented in Vissers et al. (2012). MPC has also been proposed and experimentally validated for the control of the growth rate (Mesbah et al., 2011). Similar MPC schemes have been applied to control the crystal size in Mesbah (2010).

First attempts of batch-to-batch control for supersaturation using Iterative Learning Control (ILC) have been proposed very recently, see for instance Zhang et al. (2009) and Sanzida and Nagy (2012). However, the results presented in those contributions are limited to simulation studies.

1.8 Problem statement

The following observations have lead us to the problem statement of this thesis

- A wealth of model-based control strategies for batch processes are available in the literature.
- The models describing the dynamics of batch processes suffer from severe uncertainties. Therefore, the performance delivered by these model-based control strategies may be far from optimal.
- In the industrial practice, several batches are repeated over time. The measurements from past batches contain important information on the process dynamics and on the characteristics of the disturbances.

The problem statement is the following:

Problem statement

Develop strategies to improve from batch to batch the performance of model-based control for batch processes and, if possible, validate them for the batch cooling crystallization process.

For the sake of concreteness, we have given particular attention to a specific control objective for a specific batch process, namely the tracking of a constant supersaturation⁷ set-point for a batch cooling crystallization process. More precisely, the control objective for a batch is to design the entire temperature trajectory in order to track the supersaturation set-point, based on the measurements

⁷ We have selected the supersaturation as controlled variable for the batch cooling crystallization process because this quantity can be readily estimated from the measurements of temperature and concentration which were available in our experiments. In general, different process variables such as the growth rate or the CSD could be considered, once these quantities are measured or estimated.

collected in the previous batches. Thus, the control problem classifies as batch-to-batch control of run-time variables in the framework of Section 1.4.

Nevertheless, most of the control methodologies developed in this thesis can be adapted for different batch processes and control objectives.

1.9 Approach

In the pursuit of the main research objective, our general approach is to use the measured data from previous batches in order to improve from batch to batch the model that is used to design the model-based controller. By doing so, the performance of the model-based control system is expected to increase. In order to achieve this result, we have identified and tackled a number of distinct goals. These goals, together with the specific solution approach that we have followed, and an overview of the results achieved of this thesis are presented in this section.

1.9.1 Development of a parametric model update strategy for batch cooling crystallization

As discussed in Section 1.5, a possible approach to perform the batch-to-batch model update is to estimate the uncertain physical coefficients of the process within a model structure obtained from a first-principles modeling. Owing to the structural use of all the a-priori knowledge available on the process, this approach has the potential of delivering an accurate model using a limited amount of data.

The first strategy that we have developed for the batch-to-batch model update is based on this approach and is called in this thesis Iterative Identification Control (IIC). The core element of the IIC approach is a repeated parameter estimation procedure. In IIC, the model of the batch cooling crystallization process is selected within a set of candidates described in a fixed *model structure*. The model structure is obtained from a first-principles modeling of the process, leaving as free parameters a number of uncertain physical constants. The measured batch data are used to update from batch to batch estimates for those uncertain parameters. More precisely, the estimation is performed iteratively after each batch adopting a *maximum likelihood* (Van den Bos, 2007) framework which combines the previous estimate with the data measured in the most recent batch. By doing this (and under certain conditions on the data, see later), the variance of the estimated parameters decreases after each batch, since the estimate is constructed based on the information contained in all the previous batches.

The model updated with the estimated parameters is used to design the input temperature trajectory for the next batch with the objective of tracking the desired supersaturation set-point. The batch is performed using this new input, and so on and so forth for the following batches.

In general, the estimates of the model parameters can really improve only if the data used to update the estimates are *sufficiently informative* (Ljung, 1999). This requires certain condition on the input signal applied to the system while the data are generated. These conditions could be enforced for instance by superposing a special *excitation signal* to the *normal control input* (i.e. the input optimized ac-

ording to the control objective). In our case, we found that the input optimized in order to track the desired set-point already produces sufficiently informative datasets. Therefore, we did not include excitation signals in the IIC algorithm.

IIC is a very powerful and general model learning approach. The measured data, together with the structural a-priori information are used efficiently in order to estimate a limited number of uncertain parameters. Therefore, a very accurate model can be obtained within few iterations of the algorithms (i.e. in a limited number of batches). This model can be utilized in order to design virtually any kind of model-based controller. Even though in this thesis we limited ourself to a supersaturation tracking control problem, other objectives could be similarly selected.

However, a limitation of IIC is that its attractive properties are valid only under the assumption that the process dynamics are truly described within the model structure selected. In the case of a *structural model mismatch*, i.e. when the true process dynamics are not described by any of the models in the model structure, the performance delivered by IIC is in general reduced. Even worse, limited tools are available in order to quantify the performance degradation to be expected in the case of structural model mismatch.

Unfortunately, structural model mismatches cannot be ruled out for the batch crystallization process. Some of the basic principles of the process are debated in the scientific community, and different model structures describing the same phenomena occurring during the process can be found in the literature.

1.9.2 Development of a non-parametric model update approach

In order to cope with the limitations of IIC in the presence of structural model mismatches, we address the use of a non-parametric model update strategy known in the literature as Iterative Learning Control (ILC). ILC is a popular tool for the improvement of the control performance of uncertain dynamic systems that operate repetitively (Bristow et al., 2006). More than a single algorithm, the term ILC actually denotes a class of algorithms specifically designed to solve repeated feed-forward reference tracking problems, such as the supersaturation tracking problem at hand. Unlike IIC, ILC approaches do not usually require strict assumptions on the model structure describing the true process dynamics, nor the presence of excitation signals. However, they do require that the initial condition and the reference trajectories are the same (or at least do not change too much) for all the batches, and that the real-time disturbances are moderate or absent.

In the ILC algorithm that we have adopted, the measured data from previous batches are used to compute a non-parametric, additive correction term for a nominal model of the process dynamics. The nominal model is obtained using the same model structure selected in the IIC algorithm, for a certain nominal choice of the model parameters. Owing to the flexible, non-parametric model correction, the ILC algorithm is suitable to compensate the nominal model for the structural model mismatches that could affect the true process dynamics. This is the crucial advantage of ILC with respect to IIC.

However, it is expected for ILC a slower convergence to the desired set-points than in the case of IIC. Furthermore, the model corrected using the ILC strategy

is a good approximation of the true system dynamics only along one particular trajectory. Therefore, the model is useful only to solve a specific repeated reference tracking problem, with a fixed reference for all the batches.

Another well-known drawback of ILC is the inability to cope efficiently with *real-time disturbances*, i.e. disturbances that are different from batch to batch. To date, most of the successful applications of ILC have appeared in the domain of (electro)-mechanical systems, where the effect of these disturbances is less severe than in the process field (Ahn et al., 2007). For process control application, the use of ILC techniques is more delicate, and generally suggested only in combination with a regular feedback control solution which takes care of these disturbances (Chin et al., 2004). First attempts of using ILC for supersaturation tracking control have appeared recently in the literature, see for instance Zhang et al. (2009) and Sanzida and Nagy (2012). However, these contributions were not dealing with the important issue of the process disturbances, which were not considered in the simulation studies reported therein.

In this thesis, we combine the ILC algorithm for supersaturation tracking with a lower-level feedback temperature controller in a master-slave configuration. Based on the update model, the ILC (master) controller determines from batch to batch a new reference temperature trajectory for a feedback controller in order to track the desired supersaturation set-point. The role of the feedback temperature controller is to reject the real-time disturbances as efficiently as possible, thus decreasing their influence on the supersaturation dynamics. This improves the efficiency of the ILC algorithm.

1.9.3 Validate the batch-to-batch control algorithms

Simulation results. We will first validate the IIC and ILC environment in a simulation environment using a model of the process dynamics obtained from first-principles modeling. In the simulation study, we take into account several issues that may occur in in practice, such as the presence of both parametric and structural model mismatches, as well as process disturbances.

The simulation results show that the two algorithms have complementary advantages and disadvantages. On the one hand, IIC provides the best performance when the assumed model structure can actually describe the *data-generating system* (i.e. the simulation model representing the true process dynamics). On the other hand, ILC is more robust to structural model mismatches. Even though these mismatches slow down the convergence, a satisfactory result is eventually obtained after a number of batches.

Experimental results. We will present the results of an experimental campaign where we have tested the IIC and ILC algorithms on a pilot-scale crystallization set-up. The experiments have been performed in the ACES department of the company DSM (Geleen, The Netherlands) with the support of the ISPT research project PH-00-04 (see Section 1.11). We used for the experiments a 50-liters, jacketed glass vessel as crystallizer and performed batch cooling crystallization experiment from a solution of Succinic acid in water.

These experiments confirm the potential of the IIC and ILC approaches for

the batch-to-batch improvement of model-based control in batch cooling crystallization. However, they also highlight a number of issues that still need to be addressed in order to bring similar techniques to an industrial production environment.

1.9.4 Investigate the role of excitation in an iterative identification/controller design scheme for linear dynamical systems

In the IIC algorithm, the input is designed for each batch in order to satisfy the control objective, i.e. in order to track the desired supersaturation set-point. Once the batch is performed, the measured batch data are used in order to refine the model of the batch dynamics. The improved model is used to design the input for the next batch, and so on and so forth for the following batches.

An interesting observation is that in IIC the choice of the input does not only determine the control performance for the current batch, but also influences the control performance for the following batches. In fact, the accuracy of the identified model (and thus the control performance in the following batches) depends on the input applied to the system in the identification data (Ljung, 2007). For the supersaturation tracking control problem, we verified that the normal control input (i.e. the input designed in order to track the constant supersaturation set-point) was already sufficiently exciting in order to estimate the model parameters with a reasonably good accuracy, and we used this input in the IIC algorithm.

Even though this strategy already guarantees satisfactory results (for the particular control problem at hand), it is possible that different choices could provide an even superior performance. For instance, the use of an additional excitation signal superposed to the normal control input leads after identification to an increased model accuracy, and thus a better control performance. The same excitation signal, however, causes a temporary performance degradation, since it acts as a disturbance while it is applied to control system.

In the thesis, we will study in detail the problem of designing the excitation signals in an iterative identification/controller design scheme with the objective of optimizing the *overall performance*. By overall performance, we mean that we take into account both the performance degradation due to the application of the excitation signal and the improvement due to the more accurate model.

The IIC framework is generalized in order to describe either a continuous or a batch process regulated by a model-based controller. The total time of operation of the model-based control system is divided into a number of *learning intervals*. For the case batch systems, the learning intervals correspond to the different batch runs. After an interval, the measured data are used to refine the estimate of the model parameters, and a new controller is designed. The controller will be applied in the next interval, and so on and so forth for the following intervals.

Excitation signals can be added to the normal control input for all the intervals. We have for each interval a *modeling error* cost due to the current model uncertainty and an *excitation cost* due to the application of an excitation signal. Applying an excitation signal during an interval creates an excitation cost for the current interval, but it also reduces the modeling error cost for all the following intervals. The problem is to design these excitation signals in order to find a good

trade-off between these two costs.⁸

As discussed in Section 1.5, similar design problems have been considered in the Identification for Control literature for linear dynamical systems in a two-phase framework. In the identification phase, the excitation signal is fed to the system and using the data collected from this phase a model is identified. This model is used in order to design an improved controller, that is applied to the system during the control phase. A trade-off is found between the excitation cost in the identification phase and the modeling error cost in the control phase (Bombois et al., 2006; Gevers and Ljung, 1986).

The two-phase framework does not fit the situation encountered in our iterative identification/control scheme. In our case, there is not a clear distinction between identification and control phases. In fact, identification and control design are performed during all the interval. Excitation signals can be injected in every interval and the choice of these signals should be made based on the excitation cost in the current interval, and the modeling error cost for all the following intervals.

In our framework, we define the *total cost* for an interval as the sum of the excitation cost and the modeling error cost. Our approach is to design the excitation signals aiming to minimize the sum of the total cost over all the intervals, possibly satisfying constraints on the total cost for each of the individual intervals.

In principle, we would like to find an efficient way to solve this optimization problem for general nonlinear dynamical systems, such as the one describing the batch cooling crystallization process. However, this is a very tough problem that will not be solved completely in this thesis. To date, in the literature, a general solution to this problem has not yet been presented even for the simpler case of linear dynamical systems.

In this thesis we will show that, in the case of linear dynamical systems, the optimization problem required to find the excitation signals in the iterative identification/controller design framework can be transformed into a convex optimization problem exploiting classic experiment design tools (Goodwin and Payne, 1977; Jansson and Hjalmarsson, 2005). This convex optimization problem can be solved efficiently using standard optimization software (Boyd and Vandenberghe, 2004).

The potential of our method is illustrated in a simulation study. We verified that our approach guarantees a superior overall performance compared to the classic two-phase approaches previously developed in the field of Identification for Control.

1.9.5 Develop experiment design tools for nonlinear systems

The lack of generally applicable, computationally efficient experiment design tools for nonlinear systems is the main bottleneck for the application of the framework discussed in the previous section to batch processes. In fact, as previously remarked in this introduction, batch processes often exhibit severely nonlinear behaviors.

⁸For a continuous system, the length of the intervals is an additional design choice. This is discussed in Chapter 4.

As discussed in Section 1.5, the experiment design tools are used to design excitation signals which guarantee a certain model accuracy. More specifically, the model accuracy is evaluated in the terms of the *information matrix*, which is related⁹ to the inverse of the covariance matrix of the estimated parameters, and is a function of the excitation signal. From a mathematical perspective, experiment design problems are optimization problems where the information matrix appears in the objective function and/or in the constraints, and the excitation signal is the optimization variable. Experiment design tools are used to solve these optimization problems.

Unfortunately, general, computationally efficient experiment design tools are known only for linear dynamical systems (Goodwin and Payne, 1977; Jansson and Hjalmarsson, 2005). For these systems, the information matrix is an affine function of the *spectrum*¹⁰ of the excitation signal. This property is used to solve experiment design problems using a two-step approach. First, the optimal spectrum for the excitation signal is determined. Exploiting the affine relation, the experiment design problem can often be formulated in convex form. Once the convex optimization problem is solved, an excitation signal having the desired spectrum is generated.

Conversely, experiment design for nonlinear dynamical systems is still a very open and challenging research topic. In the nonlinear case, the main difficulty is that the affine relation between the spectrum and the information matrix does not hold. Therefore, the two-step design method used for linear systems cannot be applied. A possibility is to design the entire probability density function of the excitation signal (Hjalmarsson and Mårtensson, 2007). Since the probability density function appears linearly in the information matrix, a similar two-step design approach could be adopted. However, this procedure is much more involved than the one based on the spectrum. So far, it has been successfully applied only to academic examples of very limited complexity.

An alternative approach is to optimize the excitation signal directly in the time domain by solving a dynamic optimization problem (Franceschini and Macchietto, 2008). A drawback of this approach is that the dynamic optimization problem to be solved is in general very hard. Typically, it is severely non-convex and depends on a large number of optimization variables representing a parametrization of the excitation signal. When the optimization problem is solved using standard gradient-based algorithms, chances are high that the numerical solution will lie in the proximity of a local optimum, which is possibly far away from the global one.

Motivated by the limitations of the methods available in the literature, we will conduct research towards the development of a novel experiment design tool applicable to a wide class of nonlinear systems, but still relying on convex optimization routines. We restrict our attention to *multilevel* excitation signals, i.e. signals which admit a finite number of possible levels. A multilevel excitation signal can be described by the *sequence* of the levels appearing therein. Within this se-

⁹For certain estimation criteria such as Maximum Likelihood and Prediction Error Identification, the information matrix is asymptotically equal to the inverse of the covariance matrix (see Ljung (1999) and Van den Bos (2007) for details).

¹⁰The spectrum is a frequency-domain representation of the signal which describe its power content as a function of the frequency.

quence, we recognize a number of shorter *subsequences*. Under certain conditions, we find that the information matrix for the full sequence is proportional to the contribution due to each subsequence, times the frequency at which the subsequence appears in the complete sequence. Owing to the linear relation between the information matrix and these frequencies, we are able to formulate a convex experiment design problem using the frequencies as optimization variables. After solving the problem, we generate an excitation signal in which the subsequences appear in numbers proportional to the optimal frequencies.

The applicability of our method is demonstrated in a simulation study using the model of an irreversible, first-order reaction system.¹¹ We show that the design based on the optimal multilevel excitation signal outperforms the one based on random binary signals, which is a common choice in the current engineering practice.

1.10 Organization of this thesis

This thesis is divided into five main chapters that describe the model of a batch cooling crystallization process, the batch-to-batch algorithms IIC and ILC for supersaturation tracking control, the use of excitation signals in an iterative identification/controller design scheme, a novel experiment design method for nonlinear systems, and the experiments where we applied the IIC and ILC algorithms on a pilot-scale batch crystallizer.

In Chapter 2, the batch cooling crystallization process is introduced. The control-relevant aspect of crystallization are discussed and a mathematical first-principles model of the batch cooling crystallization process is described.

In Chapter 3, the IIC and the ILC algorithm are developed and discussed in detail. The advantages and disadvantages of the two methods are thoroughly investigated in a simulation study using the crystallization model developed in Chapter 2.

In Chapter 4, the use of excitation signals in an iterative identification/controller design inspired from IIC is explored. The framework is here limited to linear dynamical systems due to the current lack of efficient experiment design tools which can handle nonlinear systems. The applicability of the method in the case of linear systems is demonstrated in a simulation study.

In Chapter 5, the novel experiment design method for nonlinear systems based on multilevel excitation signals is presented. The applicability of the method is demonstrated in a simulation study involving an irreversible, first-order reaction system.

In Chapter 6, the results of the experimental campaign performed using a pilot-scale batch cooling crystallization setup are presented. The ILC and IIC control algorithms are applied to a pilot-scale crystallization setup. This gives experimental evidence to the simulation result presented in Chapter 3.

¹¹We could not apply our method to the batch cooling crystallization system since the latter does not satisfy one of the required condition, namely the fading memory property. More details are given in Chapter 5 where the method is presented.

1.11 About the project

The research has been conducted within the research project “PH-00-04: Intelligent observer and control design for industrial batch cooling crystallization”. The project was sponsored and initiated by the Institute for Sustainable Process Technology (ISPT) and the academic/industrial partners within the project.

The main objective of the project was to develop and test novel monitoring and control techniques for batch cooling crystallization processes, with particular focus on applications in the pharmaceutical industry. In order to achieve these objectives, a novel measurement equipment for batch cooling crystallization processes called *skid* was built. The architecture of the skid is briefly discussed in Chapter 6 of this thesis.

The following academic and industrial partners have been cooperating within the project.

- Delft University of Technology - Department DCSC
- Delft University of Technology - Department Process and Energy
- Eindhoven University of Technology - Department of Electrical Engineering
- Albemarle Catalysts
- Merck Sharp & Dohme
- DSM Research
- Friesland Campina DOMO
- Perdix Analytical Systems
- DotX Control Solutions B.V.
- Ipcos Boxtel B.V.

The batch cooling crystallization process

In this chapter, the control-relevant aspect of crystallization are introduced and a first-principles model of the batch cooling crystallization process is presented. Following, the results of a number of simulations using this model are presented and analyzed. Finally, the control strategies developed in the scientific literature and the ones currently applied in the industrial practice for batch cooling crystallization processes are discussed. The model of the batch cooling crystallization process will be utilized as a simulation test case for the batch-to-batch control algorithms in Chapter 3. Experimental results obtained applying these algorithms on a real crystallization system will be presented in Chapter 6.

2.1 Introduction

This chapter provides a control-oriented introduction to the batch cooling crystallization process, which is the main example of a batch process considered in this thesis. The process model presented in this chapter will be used as a simulation test-case for the batch-to-batch control algorithm presented in Chapter 3. Experimental results in which these algorithms have been applied on a real crystallization system will be presented in Chapter 6.

In this chapter, the basic principles of crystallization are first discussed. This part has been written as a first introduction to crystallization for an audience which has little or no familiarity with this topic, and whose background is possibly different than chemical or process engineering. It is required to familiarize with the phenomena that play a major role in a crystallization process, as well as to introduce the field-specific terminology. We do not claim here completeness, and we refer to the crystallization literature (see Myerson (2002) and the references therein) for a more in-depth presentation.

Subsequently, a first-principles, dynamical model for the process is presented. The population balance, mass balance, energy balance and constitutive equations

are used to construct the model. The overall process model is composed of a set of partial integro-differential equations. A model reduction technique known as moment model reduction is used to transform the model into a set of ordinary differential equations which is easier to deal with for optimization and control purposes. The results of a number of simulations along a nominal profile are presented and analyzed. The nonlinearities in the model are quantitatively evaluated by considering the deviation of the output in response to step deviations on the input along the nominal input trajectory.

Finally, the control strategies developed in the scientific literature and the ones currently used in the industrial practice for batch cooling crystallization processes are briefly discussed.

2.2 Principles of Crystallization

2.2.1 Chemical solution

In order to describe a crystallization process, it is necessary to introduce the concept of *chemical solution*. In chemistry, the term solution refers to an homogeneous mixture of at least two substances: a solvent and one (or more) solutes. The solvent constitutes the major fraction of the mixture and the molecules of the solutes are said to be dissolved in the solvent. The solution has the same physical phase as its solvent. Thus, for instance, if the solvent is liquid, then the solution is also liquid.

Hereafter, we consider liquid solutions where the solvent is liquid and the solute dissolved in it is a solid at the condition of interest. An everyday-life example is a glass of water in which a teaspoon of sugar has been dissolved. For a solution, it is possible to define the concentration C of the solute as

$$C = \frac{\text{amount of solute dissolved}}{\text{amount of solution}} \quad \text{or} \quad C = \frac{\text{amount of solute dissolved}}{\text{amount of solvent}}. \quad (2.1)$$

Since the amounts in the formulas above can be represented using different units (volume, mass or moles), the concentration C can be expressed in a variety of different units. In the following, we will consider the concentration as the ratio of the amount of the solute to the total amount of the solution expressing both the quantities in kilograms. In this case, the concentration is a dimensionless quantity.

2.2.2 Objective of the crystallization process

The objective of the crystallization process is to extract the molecules of the solute from the solvent in a solid, crystalline form. Furthermore, certain requirements on a number of properties of the final crystalline product have to be met. The most important industrially-relevant properties are specified in terms of the crystal size, morphology, shape and purity (Vissers, 2012):

Size Crystal of different sizes are produced in a crystallization process. The crystal size influences physical properties of the product such as density and filtrabil-

ity.

Morphology The molecules of the same compound can arrange in different crystalline structures called *polymorphs*. Different polymorphs have different chemical and physical properties.

Shape Crystals can grow in different shapes such as needles, cubes or other polyhedra. Similarly to the size, the shape has an influence on the physical properties of the product.

Purity It is possible that molecules of undesired impurities (e.g. by-products of a previous reaction step) or molecules of the solvent are included in the crystal structure during the growth phase.

The actual specifications depend on the crystallization system at hand, on the following downstream process step and on the final use of the product. For instance, the size of the crystals affects the efficiency of downstream operations such as filtering, milling, mixing, granulation, compactation, etc. (Larsen et al., 2006). A product that includes molecules of the solvent might be difficult to dry (Visser, 2012). For a drug, a final product consisting of a specific polymorph having a very high purity is often required. A thorough characterization of the possible specifications is out of the scope of this thesis. The interested reader is referred to Myerson (2002).

2.2.3 Phase diagram

In the following, we consider in particular a cooling crystallization process. As the name suggests, the extraction of the solute from the solution happens in this case by cooling down the solution. In order to describe the process, it is useful to introduce the so-called *phase diagram* (Figure 2.1). The phase diagram has the temperature and the concentration of a solution on the horizontal and the vertical axes, respectively.

The following *ideal* experiment is useful to understand the main chemical and physical principles governing the process. The experiment takes place in an ideally mixed, closed vessel. Owing to the ideal mixing conditions, all the physical and chemical properties such as temperature and composition are homogeneous at different locations inside the vessel.

2.2.4 Solubility

A certain amount of pure solvent is loaded into the vessel at the start of the experiment. The system can be represented in the phase diagram by point A, which has concentration equal to zero and a given temperature. A certain amount of solute can be added and will dissolve in the solvent forming a solution (point B). The concentration of the solution increases if more solute is added to the solution and dissolves in it. However, for each temperature there is a maximum amount of solute that can be dissolved. The maximum concentration that can be reached

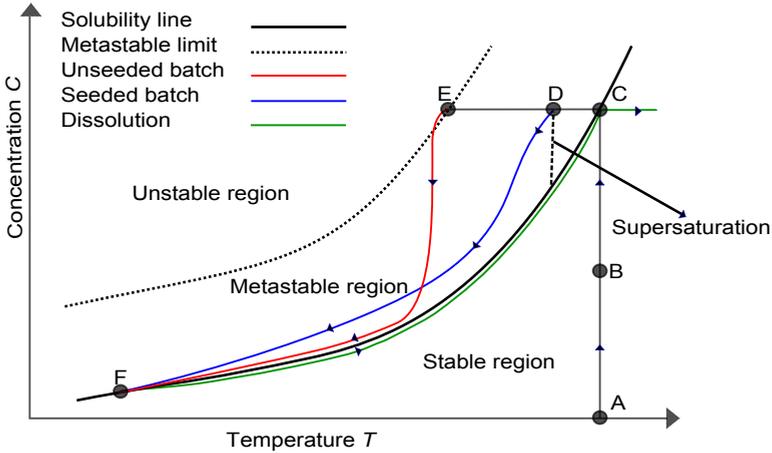


Figure 2.1: The phase diagram of the ideal experiment.

by adding solute to the solution at a constant temperature T is represented in the phase diagram by the *solubility curve* $C_s(T)$. (black continuous line).¹ In most of the cases, the solubility curve $C_s(T)$ is a strictly monotonically increasing function of the temperature T , i.e. $C_s(T_2) > C_s(T_1)$ for $T_2 > T_1$.²

When the value of the concentration C is exactly equal to the solubility C_s , the solution is said to be *saturated*. Conversely, the temperature value T_{sat} such that the solubility C_s is equal to the concentration C is called *saturation temperature*. Formally, we have

$$T_{\text{sat}}(C) = C_s^{-1}(C) \quad (2.2)$$

where $C_s^{-1}(\cdot)$ denotes the inverse of the function $C_s(\cdot)$

In the phase diagram of Figure 2.1, if more solute is added after having reached the point C (which lies exactly on the solubility curve), it will remain in the solid state. Since the vessel is ideally mixed, this solid will be dispersed in the solution in the form of a suspension. The content of the vessel is no longer a clear solution, but a two-phase fluid slurry consisting of the solution and the solid suspension.

2.2.5 Metastable region and Metastable Zone Width

Let us assume that the system is at point C and the solution is perfectly clear, i.e. no solid suspension is present. If the temperature is decreased, the system moves to the so-called *metastable region*. As the name suggests, in this region the chemical system is not at an equilibrium. The concentration is above the solubility curve

¹In the following subsection, we will see that the concentration can actually be brought above the solubility curve following a different experimental procedure. However, the solution is not at a chemical equilibrium point in that case.

²In some special cases, the solubility curve decreases with the temperature. Furthermore, the solubility curve may also depend on other variables such as the pressure, the PH of the solution and the presence of impurities. For the sake of simplicity, here we assume that these variables are constant throughout the process and that the solubility curve is a monotonically increasing function of temperature.

and a certain chemical potential is generated. This potential is often considered in terms of the *supersaturation*, which is defined as the difference between the actual concentration and the solubility at the given temperature. Supersaturation is often described as the *driving force* of crystallization, in the sense that it is necessary (even though not per se sufficient, see later) for all the phenomena involved in crystallization to occur.

The molecules of the solute in a supersaturated solution tend to group together into arrays known as *clusters*. The formation of clusters is thought to be a stochastic process caused by random fluctuation of the energy in the solution (Kashchiev, 2000). The clusters are not stable structures and are continuously forming, changing size and disappearing due to the attachment or the detachment of solute molecules. In principle, there exists a minimum critical size that would make the cluster permanent crystal formations. The value of the critical size decreases for increasing values of the supersaturation (Mullin, 1993). However, in the metastable region none of the cluster reaches the critical size and the crystallization process does not start.

In the phase diagram, if we move from point C to point D no crystals are generated, even though in D a positive supersaturation can be measured. Clusters of solute molecules are continuously forming and disappearing in the supersaturated solution without ever reaching the critical size. From a macroscopic point of view, the properties of the solution do not change.

If the temperature is further reduced in the metastable region, it is possible to hit the *metastable limit* and reach the *unstable region*. In Figure 2.1, if the temperature is further reduced from point D, the system will eventually reach the point E which lies in the unstable region. As we will see in the next subsection, the crystallization process can start once the unstable region is reached.

The width of the metastable region is often evaluated in terms of the MetaStable Zone Width (MSZW). This quantity is defined as the difference between the temperature at which the solution enters the metastable region (which is the saturation temperature for the total amount of solute dissolved in the solution) and the temperature at which the solution leaves the metastable region and reaches the unstable region. In the phase diagram 2.1, the MSZW corresponds to the length of the segment CE.

2.2.6 Unstable region and nucleation

In the unstable region, crystals are spontaneously generated inside the solution due to a mechanism known as *primary nucleation*. For increasing values of supersaturation, the critical size of the cluster decreases. In the unstable region, the supersaturation is so large that some of the clusters can reach the decreased critical size. These cluster become stable crystal structures known as *nuclei* that can grow spontaneously in a supersaturated solution. Primary nucleation inherits the stochastic nature of the cluster formation process. Indeed, the primary nucleation rate depends on the number of clusters that grow up to the critical size, which in turn is a random quantity.

After the first crystals are formed due to the primary nucleation, other crystal formation phenomena known as *secondary nucleation* occur. Several secondary

nucleation mechanisms have been observed, all requiring the presence of preexisting crystals in the solution. A comprehensive categorization of these mechanisms is out of the scope of this thesis. According to (Wissing et al., 1986), the major secondary nucleation mechanism in industrial crystallizers is the result of micro abrasion which occur due to the crystal-crystal, crystal-impeller and crystal-crystallizer wall impacts. This mechanism is known as *contact secondary nucleation*.

In our ideal experiment, starting from the point E, a number of crystals are first generated due to primary nucleation. Afterwards, more crystals will be generated due to both primary and secondary nucleation. The relative amount of primary and secondary nucleation is debated in the scientific community. The classical nucleation theory predicts very high values for the primary nucleation, that would suggest that this is the major crystal generation phenomenon. However, according to recently developed theories and experimental evidences (Kadam et al., 2011), the rate of the primary nucleation is much lower than the one predicted by previous theories and in practice all the crystals that are eventually generated (apart the very first ones) are the result of the secondary nucleation mechanism.

2.2.7 Unseeded batch and crystal growth

A batch crystallization process initialized by a primary nucleation event in the unstable region is referred to as an *unseeded batch*. After a primary nucleation event, the crystallization initially evolves at a dramatic speed due to the large initial supersaturation. More and more nuclei are generated due to both primary and secondary nucleation. Furthermore, the nuclei attract more molecules of solute and start to grow. As for the nucleation, even for the growth several theories exist (Myerson, 2002) and a thorough analysis of those is beyond the scope of this thesis. In general, it is accepted that the growth is an increasing function of the supersaturation.

Due to the growth, the solute is transferred from the solution to the solid crystalline state and consequently its concentration in the solvent rapidly drops towards the solubility curve.

If the temperature is kept constant, the concentration will eventually reach the solubility curve and the crystallization process would stop. In order to continue the crystallization, the temperature is gradually reduced in order to lower the solubility curve and maintain supersaturation. However, in an unseeded batch the concentration is consumed very fast by the growth of the large number of crystals previously generated and for this reason the trajectory followed by the concentration in the phase diagram is always very close to the solubility curve. When the final temperature is reached, the crystallization is finished (point F).

2.2.8 Seeded batch

It is possible to initialize the crystallization process within the metastable region and avoid the primary nucleation event, whose stochastic nature may induce reproducibility issues for the whole process. In order to do so, a certain amount of crystals is introduced in the crystallizer when the system is in the metastable region and the solution is still clear. The crystals used to initialize the process are

known as the *seeds* and the operation is known as the *seeding*.

In the phase diagram of Figure 2.1, seeding is applied in the clear solution at point D. After the seeding, the concentration decreases due to the growth of the seeds and of the newborn nuclei resulting from secondary nucleation. The temperature of the solution is decreased to generate supersaturation and finally the point F is reached, without ever entering the unstable region.

By seeding, it is in principle possible to initialize the process with a known number and type of crystals, and to avoid the primary nucleation event. Therefore, fewer batch to batch variations are expected than in the case of unseeded batch crystallization. In principle, this is a significant advantage for industrial applications.

Furthermore, seeding is applied at lower level of supersaturation compared to the ones required to initiate primary nucleation. For low supersaturation, the growth and the secondary nucleation are expected to have a simpler nature, which is an advantage for modeling and control purposes.

Finally, when the number of seeds used is sufficiently high, the material obtained at the end of the process is expected to be mainly composed of the initial seeds grown up to larger sizes, i.e. the influence of the new particles produced due to the secondary nucleation is expected to be negligible. We speak in this case of a *full seeded* crystallization process.

2.2.9 Dissolution

When the point F is reached (either from a seeded or an unseeded batch crystallization), the solution has a low concentration since most of the compound previously dissolved in the solution is now in the solid crystalline form as a suspension. The physical properties of the slurry are sensibly different from the ones of the solution: the slurry has higher viscosity and a turbid coloration. In a real industrial process, the crystallization step finishes at this point. The solid material is typically separated from the liquid phase using mechanical filtration, dried, and possibly undergoes further process-specific steps (Myerson, 2002).

Instead of performing these operations, in this ideal experiment we now heat up the crystallizer in order to dissolve the solid suspension and bring the system back to the initial state. Let us assume that in point F the concentration lies exactly on the solubility curve. When the slurry is heated up, the solubility increases and the crystal start to dissolve moving the concentration back to the solubility curve. Dissolution is a very fast process compared to crystallization. For this reason, the concentration is always very close to (but in principle lower than) the solubility curve during the heat-up phase. In other words, a negative supersaturation is maintained during the heat-up phase. However, this supersaturation is so small that can hardly be observed using normal process equipment.³

When point C is reached, all the material is dissolved again and the solution is clear. If the temperature is further increased, the concentration remains constant at the same value as in point C.

³ In our experiments, we will exploit this phenomenon and determine the solubility line of the component using the measurement of temperature and concentration collected during an heat-up phase (see Chapter 6).

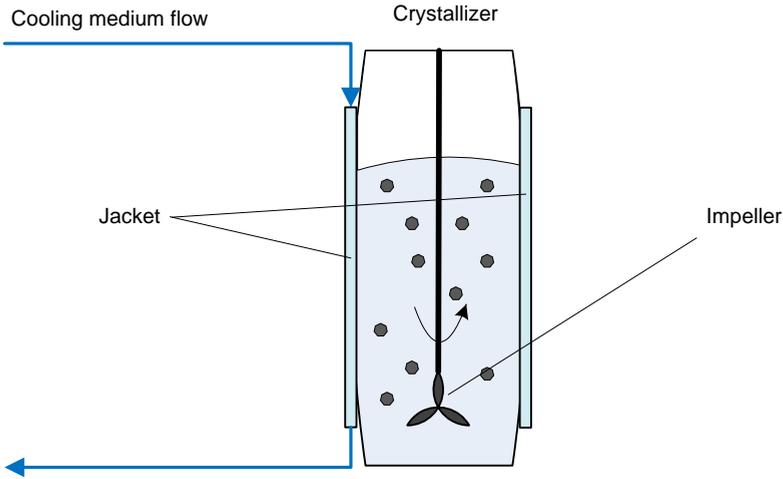


Figure 2.2: Schematic representation of a batch cooling crystallizer. The crystallizer temperature is manipulated by circulating a fluid medium in *jacket*. The content of the crystallizer is mixed with the help of an impeller.

2.3 Seeded batch cooling crystallization

In this section we consider in detail seeded batch crystallization. First, the practical operations required to perform the process are described. Subsequently, a mathematical, first-principles model is presented. Finally, the results of a simulation based on the model are shown and the nonlinearities of the model are analyzed.

2.3.1 Operation

In an industrial environment, a seeded batch cooling crystallization process takes place in a closed stirred vessel called *crystallizer*. The temperature in the crystallizer is usually manipulated by circulating a cooling medium in the so-called *jacket* which surrounds the crystallizer. The heat is removed from the crystallizer by contact with the cooling medium. The composition is kept as homogeneous as possible by mixing the content of the crystallizer with the help of an impeller. A schematic representation of a batch cooling crystallizer is represented in Figure 2.2.

The operations performed to execute the batch are described in the following. In Figure 2.3, the corresponding position of the process in the temperature-concentration plane are shown.

1. The solution is loaded into the crystallizer at the start of the batch. The solution has initial concentration C_i and T^h .
2. The solution is cooled down from temperature T_h to the temperature T_{seed} in order to generate an initial level of supersaturation which is suitable for

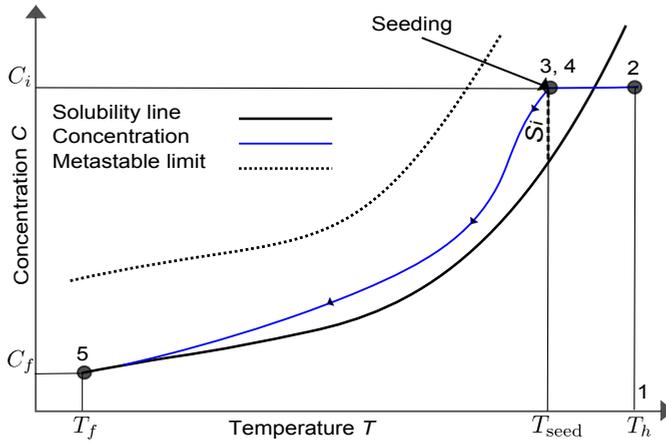


Figure 2.3: Steps of a batch cooling crystallization.

seeding.

3. The seeds are added to the solution and start to grow.
4. The crystallizer is cooled down to the final temperature T_f . The crystals keep growing and new crystals are generated due to the (secondary) nucleation phenomena. The concentration keeps decreasing and reaches the final value C_f .
5. The crystalline product is filtered out from the solution, dried, and undergoes the following process steps.

The most important operational variables that can be directly manipulated in order to influence the result of the crystallization process are

- The initial concentration C_i , supersaturation S_i and the seeding temperature T_{seed} .
- The temperature cooling profile $T(t)$ from the seeding temperature T_{seed} to the final temperature T_f .
- The amount (and possibly the type) of the seeds.

The model that we are going to develop describes the system after the seeding, when the solution is cooled down to the final temperature (i.e. point 4 of the list of operations described above). The model has to describe the time evolution of

- The temperature of the solution.
- The concentration and the supersaturation of the solution.
- The number and the size of the crystals.

2.3.2 Population balance

Modeling the properties of the crystals is the biggest challenge due to the *distributed* nature of the process. A large number of crystals is generated and each of them has different properties. In general, it is not practical to keep track of the properties of each individual crystals. It is more useful (and computationally less involved) to model the distribution of these properties among the whole population of the crystals.

A population balance is a mathematical tool to describe systems consisting of a (usually large) number of separate entities called *particles* dispersed in an environmental phase called *continuous phase* (Ramkrishna, 2000). A number of properties is associated with the particles. These properties are collected in the so-called *particle state*. In general, one would incorporate in the particle state the properties that are of direct interest for the particular application, plus the properties that are necessary to describe the time evolution (e.g. birth and death) of the particles.

In the case of crystallization, the particles are the crystals and the continuous phase is the volume of the crystallizer which is occupied by the slurry. The properties contained in the particle state vector could be the position, the size, the morphology, the shape and the purity and of the crystals. Under the hypothesis of a good mixing, the properties of the slurry can be assumed homogeneous in the crystallizer. Therefore, the position of the crystal in the crystallizer can be removed from the particle state. For control-oriented models of crystallization, crystal shape, morphology and purity are also ignored in the modeling and the size L (m) $\in [0, \infty)$ of the crystals is almost always the only property considered within the particle state vector (Larsen et al., 2006).

As often done in the literature, we define the *Crystal Size Distribution* (CSD) $n(L, t)$ ($1/\text{m}^3 \cdot \text{m}$) as the number density of the crystals at time t and size L , divided by the volume V of the crystallizer. Choosing an arbitrary region $[a, b]$ in the size coordinate, the total number $N_{[a,b]}(t)$ of particles whose size is in the range $[a, b]$ at the time t is given by

$$N_{[a,b]}(t) = V \int_{L=a}^b n(L, t) dL.$$

The CSD evolves over time due to the growth $G(L, t)$ (m/min) of the existing crystals and the birth $B_0(t)$ ($1/\text{min}$) of new crystals, i.e. the nucleation. The growth $G(L, t)$ represents the speed at which the crystals of size L increase at the time t and causes the CSD to shift towards higher size. The birth $B_0(t)$ represents the number of particles having size $L = 0$ that are generated due to nucleation at the time t and implies a boundary condition on the CSD at the size 0, i.e. on $n(0, t)$. Given two arbitrary sizes a, b with $a \neq 0$ and $b > 0$, the growth is the only mechanism such that the number of particles in the size interval $[a, b]$ can instantly change. We have indeed that

$$\frac{d}{dt} N_{[a,b]}(t) = \frac{d}{dt} V \int_a^b n(L, t) dL = G(a, t)n(a, t) - G(b, t)n(b, t).$$

where $G(a, t)n(a, t)$ and $G(b, t)n(b, t)$ represent the flux of crystals through the left

and right boundaries of the interval $[a, b]$ respectively (Ramkrishna, 2000). This last formula can also be written as

$$\int_a^b \left[\frac{\partial n(L, t)}{\partial t} + \frac{\partial G n(L, t)}{\partial L} \right] dL = 0.$$

Since the interval $[a, b]$ in the previous equation is arbitrary, the smoothness of the integrand implies that it vanishes altogether. Thus, we obtain the *Population Balance Equation* (PBE)

$$\frac{\partial n(L, t)}{\partial t} + \frac{\partial G(L, t)n(L, t)}{\partial L} = 0. \quad (2.3)$$

This equation must be supplemented with initial and boundary conditions.

Initial condition Since we apply seeding to the process, an initial distribution of crystals is introduced into the system at the time $t = 0$. Thus we set the initial condition to

$$n(L, 0) = n_0(L)$$

where $n_0(L)$ is the CSD of the seeds.

Boundary condition Assuming that the growth rate $G(L, t)$ is positive, a boundary condition can be set for the size $L = 0$. Let us recall that $B_0(t)$ is the number of crystals having size $L = 0$ which are generated at the time t due to nucleation. Therefore flux of the particles entering the size class $L > 0$ at the time t is

$$G(0, t)n(0, t) = B_0(t). \quad (2.4)$$

The PBE equation together with the initial condition and the boundary condition can be written in the form

$$\frac{\partial n(L, t)}{\partial t} + \frac{\partial}{\partial L} G(L, t)n(L, t) = 0 \quad (2.5)$$

$$n(L, 0) = n_0(t) \quad (2.6)$$

$$n(0, t) = \frac{B_0(t)}{G(L, t)}. \quad (2.7)$$

In the derivation of the PBE above, $B_0(t)$ is considered a generic function of t and $G(L, t)$ a generic function of L and t . In a crystallization process, $B_0(t)$ and $G(L, t)$ actually depend on a number of process variables (e.g. temperature, concentration) through *constitutive* kinetic expressions. In order to determine these process variables, mass and energy balances have first to be included in the model.

Moments of the CSD It is useful to define at this point the i -th *moment* of the CSD as

$$m_i(t) \triangleq \int_{L=0}^{\infty} L^i n(L, t) dL, \quad i = 0, 1, \dots, n. \quad (2.8)$$

The first four moments of the CSD carry a physical meaning regarding the total number, length, area and volume of the crystals inside the crystallizer. We have indeed that at any time t

- The total number $N_c(t)$ of crystals in the crystallizer is

$$N_c(t) = V \int_0^{\infty} n(L, t) dL = V m_0(t). \quad (2.9)$$

- The total length $L_c(t)$ of the crystals in the crystallizer is

$$L_c(t) = V \int_0^{\infty} L n(L, t) dL = V m_1(t). \quad (2.10)$$

- The total surface area $A_c(t)$ of the crystals in the crystallizer is

$$A_c(t) = V k_a \int_0^{\infty} L^2 n(L, t) dL = V k_a m_2(t). \quad (2.11)$$

where k_a (dimensionless) is the area shape factor.

- The total volume $V_c(t)$ of the crystals in the crystallizer is

$$V_c(t) = V k_v \int_0^{\infty} L^3 n(L, t) dL = V k_v m_3(t) \quad (2.12)$$

and the total mass $M_s(t)$ of crystals inside the crystallizer is

$$M_c(t) = V \rho_c k_v \int_0^{\infty} L^3 n(L, t) dL = V \rho_c k_v m_3(t) \quad (2.13)$$

where k_v (dimensionless) is the volume shape factor and ρ_c is the density of the crystalline material.

2.3.3 Mass balance

The solvent is only present in the liquid phase and its total amount M_L (kg) is assumed to be constant⁴. Conversely, the solute is transferred from the liquid to the solid phase during the crystallization process. Let M_C (kg) be the (constant) total mass of the solute in the system. We have that

$$M_C = M_s(t) + M_d(t) \quad (2.14)$$

where M_s (kg) is the mass of the solute in the solid, crystalline phase and M_d (kg) is the mass of the solute which is actually dissolved in the liquid phase. Comparing the right hand side of Equation (2.14) between the time 0 and the generic time t

$$M_s(0) + M_d(0) = M_s(t) + M_d(t). \quad (2.15)$$

⁴We neglect the evaporation of the solvent.

Substituting (2.13) for $M_s(0)$ and $M_s(t)$, solving for $M_d(t)$ we get

$$M_d(t) = M_d(0) - \rho_c k_v V(m_3(t) - m_{3,i}), \quad (2.16)$$

where $m_{3,i} = m_3(0)$ is the third moment of the CSD of the seeds.

Let us define the concentration $C(t)$ (kg/kg) as the mass fraction of the compound to the mass of the solvent:

$$C(t) \triangleq \frac{M_d(t)}{M_L + M_d(t)} \quad (2.17)$$

Using relation (2.16), we can write $C(t)$ as

$$C = \frac{C_i M_L - (1 - C_i) \rho_c k_v V(m_3 - m_{3,i})}{C_i M_L - (1 - C_i) \rho_c k_v V(m_3 - m_{3,i}) + (1 - C_i) M_L}. \quad (2.18)$$

where C_i is the initial concentration

$$C_i = \frac{M_d(0)}{M_d(0) + M_L}. \quad (2.19)$$

Thus, we see that the mass balance implies a static relation between the third moment of the CSD $m_3(t)$ and the liquid concentration $C(t)$.

Crystal mass fraction From the liquid concentration $C(t)$, we can also compute the mass $M_s(t)$ of crystals present in the slurry at the time t . Solving Equation (2.16) for $M_s(t)$ and using the definition of the concentration (2.17):

$$M_s(t) = C_0(M_L + M_d(0)) - C(t)(M_L + M_d(t)) + M_s(0) \quad (2.20)$$

Substituting again $M_d(t)$ from (2.16) we get

$$M_s(t) = \frac{C_i - C(t)}{1 + C(t)}(M_L + M_d(0)) + M_s(0) \quad (2.21)$$

and substituting $M_d(0) = M_C - M_s(0)$:

$$M_s(t) = \frac{C_i - C(t)}{1 + C(t)}(M_L + M_C - M_s(0)) + M_s(0) \quad (2.22)$$

We also define the *crystal mass fraction* m_f as the ratio of the mass $M_s(t)$ of the crystals to the total mass of the solvent and the solute inside the crystallizer

$$m_f(t) \triangleq \frac{M_s(t)}{M_C + M_L} \quad (2.23)$$

substituting $M_s(t)$ from (2.22), we have

$$m_f(t) = \frac{C_i - C(t)}{1 + C(t)} \frac{(M_L + M_C - M_s(0))}{M_C + M_L} + \frac{M_s(0)}{M_C + M_L} \quad (2.24)$$

In all the practical applications, the seed mass $M_s(0)$ is much smaller than the total mass of the solute and the solvent $M_L + M_C$. Therefore, the crystal mass fraction can be simplified as

$$m_f = \frac{C_i - C(t)}{1 + C(t)}. \quad (2.25)$$

It is often useful to evaluate the mass fraction obtained at the final time t_f of the batch. Since the final concentration $C(t_f)$ can often be approximated as the $C_s(T(t_f))$, the final mass fraction can be obtained simply knowing the initial concentration C_i (or, equivalently, the initial saturation temperature $T_{\text{sat}}(C_i)$) and the final temperature C_f :

$$m_f(t_f) \approx \frac{C_i - C_s(t_f)}{1 + C_s(t_f)} \quad (2.26)$$

2.3.4 Energy balance

The total energy in the system is given by

$$E(t) = \rho c_p V T(t). \quad (2.27)$$

where c_p ($J/^\circ\text{C}\cdot\text{kg}$) is the slurry specific heat capacity, ρ (kg/m^3) is the slurry density and $T(t)$ ($^\circ\text{C}$) is the temperature inside the crystallizer. Energy is exchanged through contact with the jackets

$$\frac{dE}{dt} = UA(T_J(t) - T(t)). \quad (2.28)$$

Substituting (2.27) for the energy, the temperature inside the crystallizer is given by the expression

$$\frac{dT}{dt} = \frac{UA}{\rho c_p V} (T_J(t) - T(t)) \quad (2.29)$$

2.3.5 Kinetic Relations

The constitutive kinetic equations link the terms G and B_o describing the growth and the birth, respectively to the other process variables. In this thesis, we use the so-called *power laws* (Myerson, 2002) in order to model these terms.

The growth rate G is assumed to depend on the supersaturation $S(t)$ according to the relation

$$G(t) = k_g S(t)^g \quad (2.30)$$

and the birth B_o is assumed to depend on the supersaturation $S(t)$ and of the third moment $m_3(t)$ according to the relation

$$B_o(t) = k_b S(t)^b m_3(t), \quad (2.31)$$

where the supersaturation $S(t)$ is defined as

$$S(t) = C(t) - C_s(T(t)). \quad (2.32)$$

Note that we assume that the growth rate does not depend on the crystal size L , i.e. $G(L, t) = G(t)$. In other words, the growth rate is the same for all particles, independently on their size. Note also that the term B_o related to the nucleation has an auto catalytic nature since it depends (through m_3) on the mass of the crystals that has already been produced in the crystallization process.

Despite their wide usage for control-oriented modeling of crystallization systems, the equations (2.30)-(2.31) describing the growth and the birth terms are the most uncertain parts of our model, both in terms of the nonlinear structure of the dependencies, and in terms of the numerical values of the coefficients appearing therein.

2.3.6 Reference PBE model

Bringing together the PBE equations with the mass balance (2.16), the energy balance (2.29) and the constitutive kinetic expressions we get the “full” PBE model (2.30, 2.31)

$$\frac{\partial n(L, t)}{\partial t} + \frac{\partial}{\partial L} G n(L, t) = 0 \quad (2.33)$$

$$n(L, 0) = n_0(L) \quad (2.34)$$

$$n(0, t) = \frac{B_0}{G} \quad (2.35)$$

$$\frac{dT}{dt} = \frac{UA}{\rho c_p V} (T_J - T) \quad (2.36)$$

where

$$G = k_g S^g \quad (2.37)$$

$$B_0 = k_b S^b m_3 \quad (2.38)$$

$$S = C - C_s(T) \quad (2.39)$$

$$C = \frac{C_i M_L - (1 - C_i) \rho_c k_v V (m_3 - m_{3,i})}{C_i M_L - (1 - C_i) \rho_c k_v V (m_3 - m_{3,i}) + (1 - C_i) M_L}. \quad (2.40)$$

and the solubility curve $C_s(T)$ is approximated as a polynomial of the temperature T :

$$C_s(T) = a_0 + a_1 T + a_2 T^2 + a_3 T^3. \quad (2.41)$$

Note that from now on we drop the dependency on time of the variables B_o , G , S , C , T , T_J , and m_3 for notational convenience. The system is described by a set of partial integro-differential equations. The input to the system is considered to be the temperature of the cooling medium inside the jacket T_J . The measured output depends on the particular measurement set-up available. In general, it could contain the temperature T , the concentration C and the some information about the CSD $n(L, t)$. In this thesis, we will assume that only the temperature T and the concentration C are measured.

In order to simulate this system, a numerical scheme has to be applied. In Appendix B, a finite volume scheme that can be used for this purpose is presented.

2.3.7 Moment model reduction

A simple model reduction technique is commonly used in order to transform the PBE model to a set of Ordinary Differential Equations (ODE).

Let us multiply both sides of Equation (2.5) by L^i and integrate over the crystal size domain:

$$\int_{L=0}^{\infty} \frac{\partial n(L, t)}{\partial t} L^i dL = -G \int_{L=0}^{\infty} \frac{\partial n(L, t)}{\partial L} L^i dL.$$

Bringing the derivative out of the integral on the left hand side⁵ and integrating by parts on the right hand side

$$\frac{d}{dt} \int_{L=0}^{\infty} \overbrace{n(L, t) L^i}^{m_i} dL = -G n(L, t) L^i \Big|_0^{\infty} + iG \int_{L=0}^{\infty} \overbrace{n(L, t) L^{i-1}}^{m_{i-1}} dL.$$

$$\frac{dm_i}{dt} = \lim_{L \rightarrow \infty} (-G n(L, t) L^i) + G n(0) 0^i + iG m_{i-1}$$

where in the previous formula $0^i = 1$ for $i = 0$. The limit $\lim_{L \rightarrow \infty} (-G n(L, t) L^i)$ is 0 since $n(L, t) = 0, \forall t$ for L large enough. Therefore, using the boundary condition (2.7), we can write

$$\frac{dm_0}{dt} = G n(0, t) = B_0 \quad (2.42)$$

$$\frac{dm_i}{dt} = iG m_{i-1}, \quad i = 1, 2, \dots, \infty. \quad (2.43)$$

Note that (2.42) is valid only when the growth rate G is positive, since the boundary condition (2.7) is used in the derivation. In order to describe the system for negative growth rate, the moment reduction cannot be applied and the PBE model has to be used.

2.3.8 Reference moment model

Making use of the moment transformation, we can write the ODE system

$$\frac{dm_0}{dt} = B_0 \quad (2.44)$$

$$\frac{dm_1}{dt} = G m_0 \quad (2.45)$$

$$\frac{dm_2}{dt} = 2G m_1 \quad (2.46)$$

$$\frac{dm_3}{dt} = 3G m_2 \quad (2.47)$$

$$\frac{dT}{dt} = \frac{1}{\rho c_p V} (UA(T - T_J)) \quad (2.48)$$

⁵This mathematical operation is known as Leibnitz integral rule and it requires the function $n(L, t)$ and its first derivative to be continuous.

where

$$G = k_g S^g \quad (2.49)$$

$$B = k_b S^b m_3 \quad (2.50)$$

$$S = C - C_s(T) \quad (2.51)$$

$$C = \frac{C_i M_L - (1 - C_i) \rho_c k_v V (m_3 - m_{3,i})}{C_i M_L - (1 - C_i) \rho_c k_v V (m_3 - m_{3,i}) + (1 - C_i) M_L}. \quad (2.52)$$

Note that we need to include the moments at least up to the third order in order to obtain a *closed* ODE system. Indeed, the third moment m_3 is required in order to compute the concentration C , which in turn is used to compute S , G and B .

The moment reduction does not introduce further approximations, i.e. the system (2.44-2.48) describes the time evolution of the moments and of the temperature exactly as the PBE model (2.33-2.36). However, the exact shape of the distribution cannot be reconstructed exactly from a finite number of moments. Furthermore, the moment model cannot be applied when the growth rate is negative (i.e. during the dissolution phase) and when B_0 and G have a more complex form⁶.

In this thesis, we will not model the system during the dissolution phase. Furthermore, we will assume that the simple kinetic expressions (2.30-2.31) hold. Therefore, the moment model will be applied for identification and control purposes.

2.3.9 Measured output and control output

We will assume throughout this thesis that the concentration C and the crystallizer temperature T are the only measured outputs of the system. The actual measurements of concentration \tilde{C} and reactor temperature \tilde{T} are collected at the measurement rate $t_s = 5$ s and corrupted by the additive noise signals e_C and e_T , respectively. The noise signals e_C and e_T are modeled as the realization of two independent white Gaussian processes having standard deviation $\sigma_T = 0.1$ °C and $\sigma_C = 0.001$ kg/kg, respectively.

In the following chapter, we will consider the supersaturation S as the control variable for batch-to-batch control. An estimate \tilde{S} of the supersaturation S can be readily obtained from the measurement \tilde{C} and \tilde{T} using (2.32), i.e.

$$\tilde{S} = \tilde{C} - C_s(\tilde{T}) \quad (2.53)$$

since the coefficients of the solubility curve $C_s(\cdot)$ are assumed to be known exactly.

⁶The method cannot not be applied for instance when G depends on the size L (except for the case of linear dependency) or when B depends on the the CSD by a quantity that cannot be represented as an expression involving the moments.

Symbol	Description	Value	Units
ρ_c	Crystal density	1130	kg/m ³
k_v	Crystal shape factor	0.1	dimensionless
ρ	Slurry density	789	kg/m ³
c_p	Slurry specific heat capacity	4185	J/(°C kg)
V	Crystallizer volume	0.905	m ³
UA	Product heat-transfer area	$1.5 \cdot 10^5$	J/(min °C)
a_0	Coefficient 0 solubility	27.8428	dimensionless
a_1	Coefficient 1 solubility	2.0891	1/°C
a_2	Coefficient 2 solubility	-0.0311	1/°C ²
a_3	Coefficient 3 solubility	0.0017	1/°C ³
k_b	Nucleation base	$1.57 \cdot 10^{13}$	1/(m ³ min)
b	Nucleation exponent	1.7	dimensionless
k_g	Growth base	$5.0 \cdot 10^{-4}$	m/min
g	Growth exponent	1.1	dimensionless

Table 2.1: Model coefficients of Succinic acid in water.

2.3.10 Model coefficients and scaling

The nominal value of the coefficients of the crystallization model are reported in the Table 2.1. These coefficient are relative to a crystallization process where the solute is Succinic acid and the solvent is water. Real experiments based on this crystallization system will be presented in Chapter 6.

In our model, the order of magnitude of the CSD $n(L, t)$ is about 10^{15} (see Figure 2.5). The order of magnitude of the first four moments m_0, m_1, m_2, m_3 is $10^{10}, 10^6, 10^3, 1$ respectively. The order of magnitude of k_b, b, k_g, g (which are very uncertain quantities, and often considered as to-be-estimated parameters) is $10^{13}, 1, 10^{-4}, 1$. For numerical reasons, it is convenient to work with a model where all the states and the other quantities of interest are in the same numerical range. This is particularly important when the model is used for optimization or parameter estimation.

Therefore, the model is modified by scaling the CSD by a factor s_N and crystal length by a factor s_L in the software implementation. In the new variables, the scaled CSD $n'(L', t)$ is expressed in $(s_N/(m^3 \cdot s_L m))$ with $s_N = 10^{-10}$ and $s_L = 10^4$. The original moments m_i are in 1:1 correspondence with the scaled moments m'_i according to the formula

$$m'_i = s_N s_L^i m_i. \quad (2.54)$$

System equations can be derived in the scaled variables by modifying the kinetic parameters:

$$k'_g = s_L k_g \quad k'_b = \frac{k_b}{s_L^3}. \quad (2.55)$$

Adopting this transformation, all the model states and the to-be-estimated parameters are kept in the same numerical range (approximately from 0 to 100).

In this thesis, we performed all the simulations and optimizations using the scaled model, while we present the results in their original physical units (unless

explicitly stated).

2.3.11 Simulations

In this subsection we analyze the response of the model developed in the previous subsection subject to a nominal input profile. The initial values of temperature and concentration and supersaturation are $T_i = T_{seed} = 38\text{ }^\circ\text{C}$, $C_i = 0.0147\text{ kg/kg}$ and $S_i = 0.0025\text{ kg/kg}$ respectively. The CSD of the seeds $n_o(L)$ (Figure 2.5a) has the shape of a parabola centered in the mean size $L_0 = 40\text{ }\mu\text{m}$ and having width $v_s = 20\text{ }\mu\text{m}$ and total mass of the seeds if $M_s = 1\text{ kg}$. The first four moments of the seed CSD distributions are (in scaled units) $m_{0,i} = 14.7$, $m_{1,i} = 5.9$, $m_{2,i} = 2.38$, $m_{3,i} = 0.97$.

The crystallizer is cooled down from the initial temperature T_i to the final temperature $T_f = 20\text{ }^\circ\text{C}$ in 100 minutes following an approximately linear temperature profile⁷. Afterwards, the crystallizer temperature is kept constant at the value T_f for 20 more minutes.

Both the PBE model (2.33-2.36) and the moment model (2.44-2.48) are simulated. The moment model is integrated in time using a fixed-step Runge-Kutta method with sampling time $t_s = 5\text{ sec}$. The PBE model is first discretized in the size domain using the finite volume scheme presented in Appendix B and then integrated in time using the same fixed-step Runge-Kutta method used for the moment model.

A number of time profiles for this simulation are reported in Figure 2.4. In 2.4a the crystallizer temperature T , the measured crystallizer temperature \tilde{T} and the jacket temperature T_J are reported. The jacket temperature is lower than the crystallizer temperature during the time of the cooling in order to remove heat from the crystallizer. At the end of the cooling, the temperatures T and T_J reach the final equilibrium value $T_f = 10\text{ }^\circ\text{C}$. Note that the noise e_T is rather small compared to the range of variation of the temperature T and therefore the signals T and \tilde{T} are almost overlapping in the plot.

In Figure 2.4b the concentration C , the measured concentration \tilde{C} and the solubility C_s are reported. The concentration C decrease from the initial value C_i to a final value C_f which is approximately on the solubility of the final temperature T_f , i.e. $C_f \approx C_s(T_f)$. Unlike the temperature noise e_T , the concentration noise e_C is rather large compared to the range of variation of the concentration C and therefore its effect on the measurement \tilde{C} is more important.

In 2.4c the supersaturation S and the supersaturation \tilde{S} estimated according to (2.53) are reported. The effect of the measurement noise signals e_T and e_C is significant on \tilde{S} . Note however that the dynamics of the temperature and the concentration are rather slow (in the range of several minutes) compared to the measurement rate $t_s = 5\text{ s}$. Therefore, it is possible to obtain a more accurate estimate of the supersaturation simply by low-pass filtering the measurements \tilde{T} and \tilde{C} before applying (2.53). (see Section 3.3.3 in Chapter 3).

⁷Note that the actual input of the system is the jacket temperature T_J . In this simulation, jacket temperature T_J was driven in such a way that the reactor temperature follows the desired temperature trajectory by simulating the system in a feedback loop with a PI controller. The PI controller in this simulation receives the noise-free temperature signal T .

The supersaturation has a peak around the time $t = 20$ min and decreases afterwards. This behavior for the supersaturation is expected in the case of a linear temperature decrease due to the combination of two effects. First, the rate at which the total crystal mass M_s increases (or conversely the concentration C decreases) depends on the total surface area of the crystals. At the start of the batch only few, small crystals (i.e. the seeds) having a small total surface area are present. Therefore, the concentration C decreases slowly. In the equations of the moment model, this is reflected by the fact that the derivative of the third moment m_3 (which is related to the crystal mass) is proportional to the growth rate G times the second moment m_2 (which is related to the total surface area of the crystals).

Second, the solubility $C_s(T)$ is steeper at higher temperature (see Figure 2.6). Therefore, for a linear cooling rate the solubility decreases faster at the start of the batch. Let us recall that the supersaturation S is defined as the difference $C - C_s$. At the start of the batch, the small concentration consumption is not compensated by the large decrease in the solubility. Therefore, the supersaturation increases. Afterwards, the concentration decreases faster and the solubility slower. Therefore, the concentration approaches the solubility and the supersaturation decreases towards 0. This behavior is also evident from the Figure 2.4b where the concentration C and the solubility $C_s(T)$ are reported.

The moments m_0, m_1, m_2, m_3 are reported in 2.4d. The moments are here reported in scaled units, so that they can be conveniently shown in the same plot. All the moments increase throughout the process reflecting the increase in the number, length, surface area and volume/mass of the crystals. Note that the increase in the number of the crystals (which is the result of the nucleation) is not particularly significant for this system. The scaled zeroth moments m_0 increases by a factor 1.3, i.e. from $m_0(0) = 14.7$ to $m_0(t_f) = 18.5$. In fact, we are here simulating a full seeded crystallization process (see section (2.2.8)) where most of the final crystals are the result of the outgrowth of the initial seeds. For different crystallization systems, the zeroth moment could increase by a much larger factor, e.g. larger than 100. For instance, it was recently shown that a crystallization process can be initialized by seeding with a single particle (Kadam et al., 2011). In this case, this factor is practically unbounded.

In Figure 2.5 the initial (a) and the final (b) CSD are shown. Note that while the other results presented in this section can be obtained by simulating either the moment model or the CSD model, the evolution of the CSD can only be simulated using the PBE model. We see that the final CSD can be seen as the sum of the initial CSD shifted to a higher mean size due to the growth of the seeds and a "tail" of smaller particles created throughout the process due to the nucleation. Even from these plots it is evident that we are here considering a full seeded crystallization process. Indeed, the total area under the curve in Figure 2.5b (which is proportional to the total number of crystals obtained at the end of the process, i.e. $m_0(t_f)$) is only 1.3 times the the area of the seed CSD in Figure 2.5a) (which is proportional to the number of the seeds, i.e. $m_0(t_0)$).

The relative importance of the crystals which are the results of nucleation is even lower if they are considered in terms of their total mass instead of their total number. Indeed, the crystals resulting from the nucleation are smaller than the ones resulting from the growth of the initial seeds, and the contribution to the total

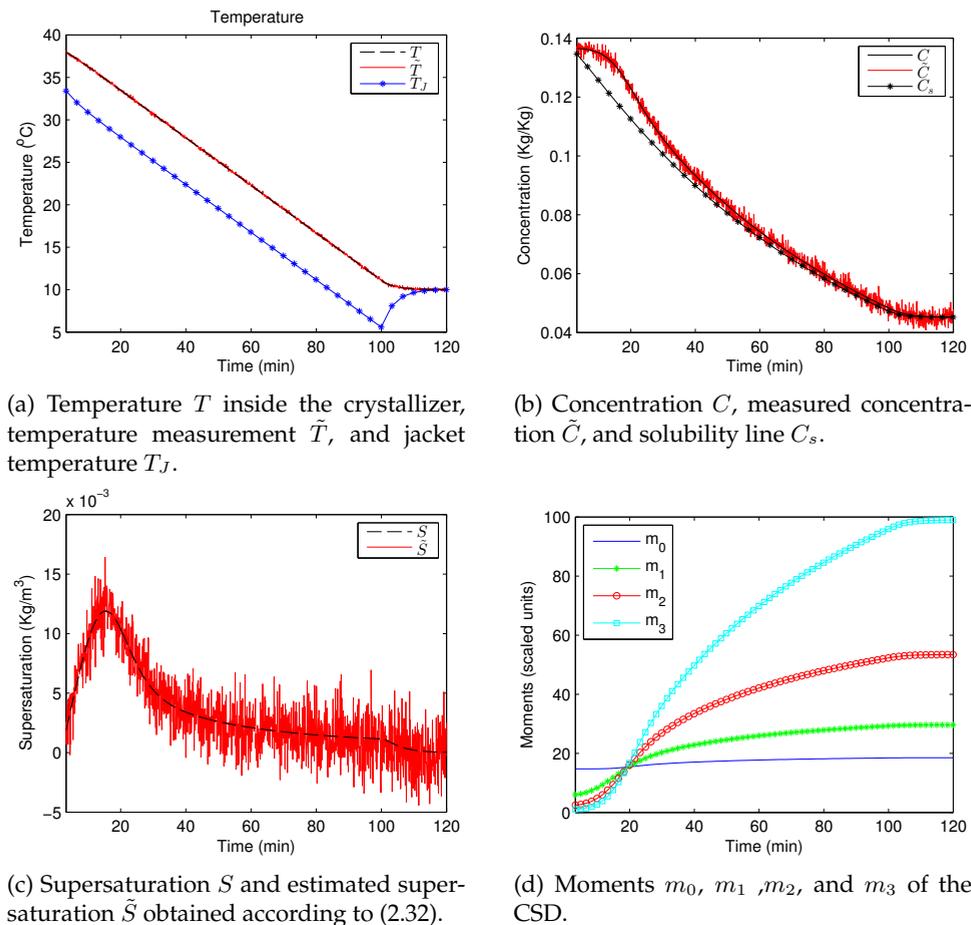


Figure 2.4: Simulated time profiles.

mass of the crystals is proportional to the cube of the crystal length (see formula (2.13)). In our simulation, the mass of crystals which are the result of nucleation at the final time less the 0.1% the total mass produced at the final time in this simulation.

2.3.12 Nonlinearity study

The presence of the nonlinearities in the model can be visualized by considering the *output deviation* with respect to a nominal output in response to an input signal constructed as the sum of a nominal input and a step input deviation. In this nonlinearity study, we consider as nominal input and output the trajectories of $T_J(t)$ and $S(t)$ of the previous simulation, respectively (Figure 2.4). Step deviations on the input having different amplitudes and applied at different time instants along the nominal input $T_J(t)$ trajectory are considered. For each input $\bar{T}_J(t)$ built as the sum of the nominal input $T_J(t)$ and a step signal, the model is simulated and

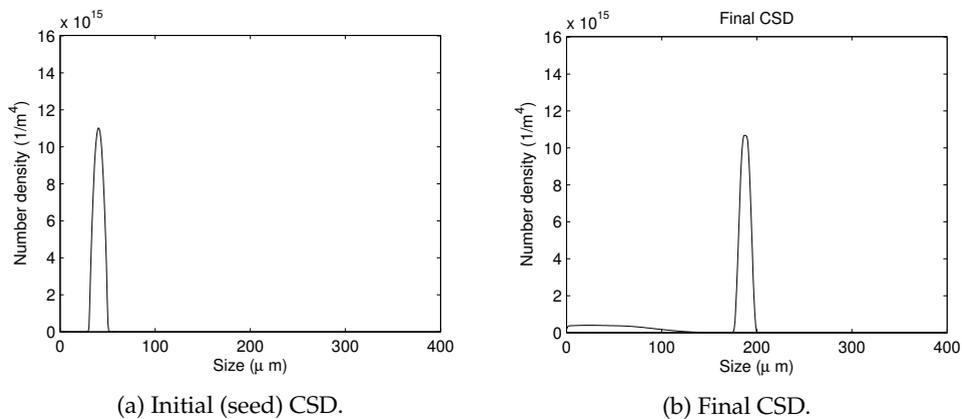


Figure 2.5: Simulated initial and final CSD.

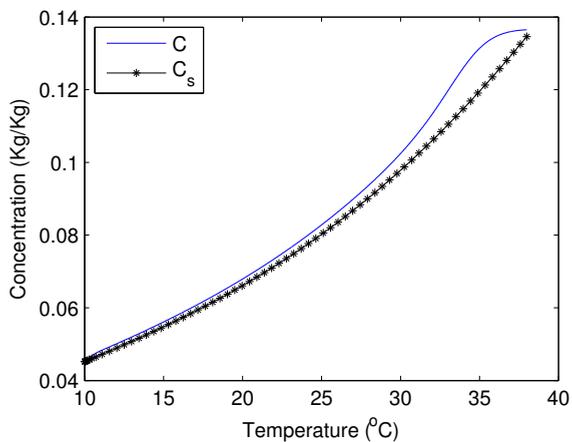
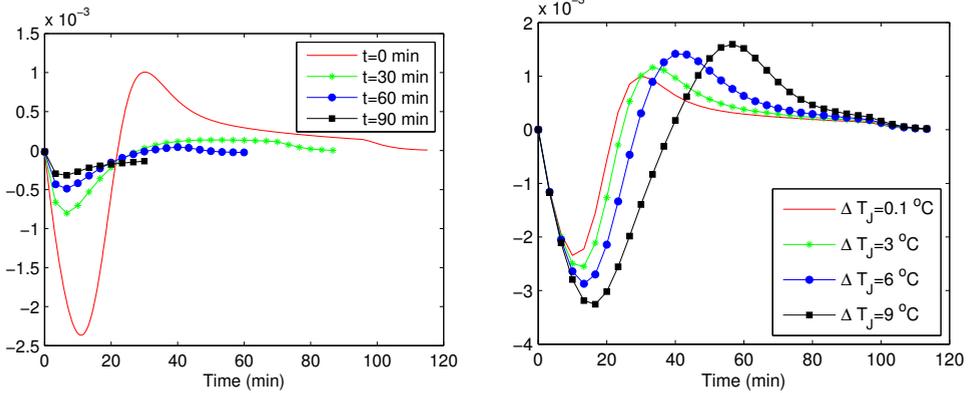


Figure 2.6: Simulated C-T plane.



(a) Normalized step responses to input deviation having amplitude $\Delta T_J = 0.1^\circ\text{C}$ applied at different time instants.

(b) Normalized step responses to input deviation applied at $t = 0$ having different amplitudes ΔT_J .

Figure 2.7: Normalized step responses.

the output $\bar{S}(t)$ is obtained. The output deviation caused by the step input deviation is thus $\bar{S}(t) - S(t)$. In order to compare the effect of the different steps, the *normalized step responses* are constructed by dividing the output deviations by the amplitude of the corresponding steps and shifting them in time to the origin. If the system dynamics from T_J to S were linear, all the normalized step responses would perfectly overlap. Therefore, the level of nonlinearity can be evaluated by considering the mismatch between the normalized responses.

In Figure 2.7a, we report the normalized responses to steps having the same amplitude $\Delta T_J = 0.1^\circ\text{C}$, but applied at different time instants $t = 0, 30, 60, 90$ min along the trajectory $T_J(t)$. In Figure 2.7b, we report the normalized responses in front of steps applied at the same time $t = 0$, but having different amplitudes $\Delta T_J = 0.1, 3, 6, 9$. The dramatic effect of the nonlinearities is evident in these plots.

2.4 Control of batch cooling crystallization processes

As previously mentioned in this chapter, the ultimate objective of a batch cooling crystallization process is to produce crystals having certain properties in terms of size, morphology, shape, purity, etc. In general, one is interested in the properties that the crystals have at the end of the process, and not in the intermediate values that they take at other instants during the process. For this reason, they classify as a run-end control variables in the batch control framework presented in Chapter 1.

In practice, direct control of the crystal properties is severely hindered by modeling and measuring limitations. In fact, the morphology, shape and purity of the crystals are not even included in our modeling framework. In the literature, few models have been proposed to predict the shape of the crystals and their poly-

morphic nature (see e.g. Datta and Grant (2004); Rohl (2003)). However, they are not considered reliable enough to be used in a control scheme. To the best of the author's knowledge, quantitative models describing the crystal purity have not been developed to date.

The crystal size is described in our model in terms of the CSD. Despite recent advances in the sensor technology, measuring this quantity is still a great technical challenge. To date, direct control of the CSD has been investigated in a certain number of publications with promising results (see e.g. Abu Bakar et al. (2009); Mesbah (2010); Nagy and Braatz (2004)). However, in our experimental campaign, we only had access to temperature and concentration measurements. As recently shown in Vissers (2012), these process variables do not contain sufficient information in order to estimate the CSD. Therefore, we could not pursue further research in the direction of CSD control.

In the absence of direct measurements of the crystal properties of interest, the latter can be controlled only indirectly by specifying the reference trajectories that a number of other process variables such as the temperature, the growth rate and the supersaturation have to follow during the time of the batch. An accurate tracking of the references can guarantee that the crystals obtained at the end of the batch consistently satisfy the production requirements. For these process variables, the entire trajectory that is followed throughout the process has an influence on the properties of the crystals that are generated. Therefore, they are considered run-time control variables in the framework introduced in Chapter 1.

Temperature control is the most common control strategy for batch cooling crystallizers in an industrial settings (Fujiwara et al., 2005). The desired cooling profile is given as reference to a feedback controller. Since the thermal part of the dynamics is fairly linear, a linear feedback controller such as a PI or PID can provide a sufficiently accurate tracking performance. While temperature control is generally easy to implement, it is not always sufficient in order to guarantee the desired product quality. In fact, the supersaturation and the growth rate have a more direct influence on the crystallization process.

Supersaturation and growth-rate control have been widely investigated in the literature (Nagy et al., 2008) using different strategies. Unlike the thermal dynamics, the dynamics of the growth and of the supersaturation are severely nonlinear (see Section 2.3.12). For this reason, a nonlinear control strategy is required in this case. In Xie et al. (2002) and Vissers et al. (2011) a feedback linearization strategy has been proposed for supersaturation control. An experimental verification of such scheme has been presented in Vissers et al. (2012). MPC has also been proposed and experimentally verified for the control the growth rate in Mesbah et al. (2011).

In the control strategies described above, a nominal model of the process dynamics is used to design the model-based controller. The quality of this model used to synthesize the controller influences the tracking performance, and thus the properties of the crystals. Unfortunately, models of batch cooling crystallization processes suffer from significant uncertainties, notably in the parts describing the growth and the nucleation. In the next chapter we will show that, when several batches are repeated, it is possible to improve the model (and consequently, the performance delivered by a model-based control solution) using the input/output

measurements from previous batches.

2.5 Summary

In this chapter, we have introduced the batch cooling crystallization process from a control-oriented perspective. The objective of the process is to extract the molecules of the solute from the solution in the solid, crystalline phase. Furthermore, a number of specifications in terms of properties such as size, morphology, shape and purity have to be met.

First, we have presented general principles regarding crystallization and presented a first-principle mathematical model of the batch cooling crystallization process. The model is constructed using the population balance framework and is described by a set of partial integro-differential equations. The well-known moment model reduction technique has been applied in order to transform the original model to a set of ordinary differential equations known as moment model. Due to the simpler structure, the moment model can more easily be used for identification and control purposes. Even though it is not possible to reconstruct the full crystal size distribution from a finite number of moment, the moment model can still describe the temperature, concentration and supersaturation dynamics without introducing any approximations in our case.

Following, we have presented and discussed the results of a number of simulations. The presence of severe nonlinearities in the dynamics from the jacket temperature to the supersaturation has been visualized by considering the output response of the model when step changes are applied along a nominal input trajectory.

Finally, we have discussed a number of control strategies for batch cooling crystallization. In general, direct control of the desired crystal properties such as the CSD is hindered by measurement and modeling limitations. Therefore, in most of the cases the crystal properties are only indirectly controlled by specifying the reference trajectories that a number of process variables such as the temperature, the growth rate and the supersaturation have to follow during the time of the batch. An accurate tracking of the references can guarantee that the crystals obtained at the end of the batch consistently satisfy the desired properties.

In the state-of-the-art approaches, the tracking of these variables is achieved using model-based control scheme. The quality of the model used to synthesize the controller influences the tracking performance, and thus the properties of the crystals. In the next chapter we will show that, when several batches are repeated, it is possible to improve the model using the input/output measurements from previous batches.

2.6 Symbols and units

Symbol	Description	Units
a_0	Coefficient 0 solubility	dimensionless
a_1	Coefficient 1 solubility	$1/^\circ\text{C}$
a_2	Coefficient 2 solubility	$1/^\circ\text{C}^2$
a_3	Coefficient 3 solubility	$1/^\circ\text{C}^3$
b	Nucleation exponent	dimensionless
c_p	Slurry specific heat capacity	$\text{J}/(^\circ\text{C kg})$
C	Concentration of Succinic acid in water	kg/kg
C_s	Solubility of Succinic acid in water	kg/kg
E	Energy of the system	J
g	Growth exponent coefficient	dimensionless
k_b	Nucleation base	$1/(\text{m}^3 \text{ min})$
k_g	Growth base coefficient	m/min
k_a	Area shape factor	dimensionless
k_v	Crystal shape factor	dimensionless
L	Crystal size	m
m_0	0th moment of the CSD	$1/\text{m}^3$
m_1	1st moment of the CSD	m/m^3
m_2	2nd moment of the CSD	m^2/m^3
m_3	3rd moment of the CSD	m^3/m^3
M_L	Total mass of solvent	kg
M_C	Total mass of solute	kg
M_c	Mass of solute in the solid, crystalline phase	kg
M_d	Mass of solute dissolved in the solution	kg
n	CSD	$1/(\text{m}^3 \text{ m})$
S	Supersaturation	kg/kg
σ_C	Standard deviation concentration noise	kg/kg
σ_T	Standard deviation temperature noise	$^\circ\text{C}$
T	Crystallizer temperature	$^\circ\text{C}$
T_J	Jacket temperature	$^\circ\text{C}$
UA	Product heat-transfer area	$\text{J}/(\text{min } ^\circ\text{C})$
V	Crystallizer volume	m^3
ρ	Slurry density	kg/m^3
ρ_c	Crystal density	kg/m^3

Table 2.2: Symbols and units used in this chapter.

Batch-to-batch control for cooling crystallization

In this chapter, we investigate the use of batch-to-batch model update techniques for the batch cooling crystallization process with the objective of improving the supersaturation tracking performance. When a model structure describing the true process dynamics is available, the use of a parametric model update provides a fast and efficient learning mechanism. However, the performance of a parametric approach is reduced in the case of structural model mismatches. Thus, when severe structural model mismatches are suspected, the use of a nonparametric approach may be advantageous. Two batch-to-batch control algorithms for supersaturation control in cooling crystallization, namely Iterative Identification Control (IIC) and Iterative Learning Control (ILC) are developed in this chapter. IIC is based on a parametric model update. In IIC, estimates of the uncertain coefficients of the process model are recursively estimated adopting a Maximum Likelihood estimation framework. Conversely, ILC is based on a nonparametric model update. In ILC, the nominal process model is adjusted with a nonparametric, additive correction term which depends on the error in the last batch. The supersaturation tracking performance achieved using the two algorithms is investigated in a simulation study. Similar control schemes will be tested on a real experimental setup in Chapter 6.¹

3.1 Introduction

As discussed in the previous chapter, the ultimate objective of a batch cooling crystallization process is to produce crystals having specified properties defined in terms of size, morphology, shape, composition, purity, etc. However, direct

¹This chapter encompasses the results presented in Forgione et al. (2012b) and Forgione et al. (2012a).

control of these properties is often not possible due to measurement and modeling limitations.

In most of the cases, the crystal properties are translated into reference trajectories that a number of process variables such as the temperature, the supersaturation and the growth rate have to follow during the time of the batch. An accurate tracking of these references can guarantee that the desired properties are consistently met (Fujiwara et al., 2005; Gutwald and Mersmann, 1990).

In industrial batch cooling crystallizers, the crystallizer temperature is often the only process variable that is controlled (Fujiwara et al., 2005). The jacket temperature is the manipulated variable used to steer the crystallizer temperature to the desired trajectory. Since accurate, on-line temperature measurements can be readily obtained, the crystallizer temperature is usually controlled in a closed-loop setting. In this configuration, the desired cooling profile is given as set-point of a feedback loop. This strategy is known as T-control in the literature (Fujiwara et al., 2005). In general, the thermal part of the dynamics is fairly linear, and a linear feedback controller such as a PI or PID provides a sufficiently accurate tracking performance. This controller is also useful in order to reject real-time disturbances affecting the temperature dynamics, caused for instance by variations of the ambient temperature, and of the temperature of the cooling medium circulated in the jacket of the reactor.

However, even when the temperature is effectively controlled, the final product of a batch might not show all the desired properties. In fact, even though the temperature is an important process variable, it is not the one most closely related to the crystallization dynamics.

A process variable having a more significant effect on the properties of the final product is the supersaturation, which is the driving force for important phenomena involved in crystallization such as the growth and the nucleation. On-line feedback control of the supersaturation has been widely investigated (Nagy et al., 2008). This strategy is known as C-control in the literature (Fujiwara et al., 2005).² In general, C-control was shown to give better performance compared to T-control, particularly in terms of reproducibility.

In most of the C-control strategies, a nominal model of the supersaturation dynamics is used to design the model-based controller. Therefore, the quality of the model has a direct influence on the tracking performance that the model-based controller can achieve. Unfortunately, the models describing this dynamics often suffer from severe uncertainties. Due to these uncertainties, the performance delivered by the model-based controller can significantly deteriorate in the case of a model-plant mismatch.

In this chapter, we investigate the opportunity of using input/output data from previous batches in order to improve this model, and consequently the performance delivered by a model-based control solution. We consider in particular a situation in which the concentration measurements are available only at the end of a batch. This situation can occur in an industrial environment, where the measurements are often obtained through the off-line analysis of samples collected

²The term C-control refers to the concentration, which is required (together with the temperature) in order to compute the supersaturation.

throughout the batch.³ Due to this limitation, on-line feedback control for the supersaturation is not feasible and feedforward solutions have to be applied. The quality of the model is even more important than in the case of feedback control. Indeed, feedforward control solutions are known to be more sensitive to model-plant mismatches than the ones based on on-line feedback.

We present a Batch-to-Batch (B2B) control framework conceived in order to track a given supersaturation profile under the presence of disturbances and model uncertainties. Based on the off-line concentration measurements collected from the previous batches, the B2B algorithm updates a model of the uncertain process dynamics. Following, it uses the updated model in order to compute an improved reference profile T^r for the temperature in the crystallizer. This profile is fed to the lower-level PI temperature controller in the next batch. Two B2B algorithms, namely Iterative Identification Control (IIC) and Iterative Learning Control (ILC) are here presented. While IIC is based on a parametric model update, ILC performs a more flexible, nonparametric model correction. Due to the different nature of the model update, the two algorithms have complementary advantages and disadvantages, which are investigated in this chapter in a simulation study.

In the IIC algorithm, the measurements are used to estimate the uncertain coefficients of the process described in a fixed model structure. In our case, the model structure used for IIC is given by the moment model (2.44)-(2.48) introduced in the previous chapter, leaving the coefficients of the growth and the nucleation as free parameters. The estimation of these parameters is performed iteratively after each batch adopting a Maximum Likelihood (Van den Bos, 2007) framework which combines the previous estimate with the data measured in the most recent batch. By doing this (and under certain conditions on the data used to perform the estimation, see later), the accuracy of the model increases after each batch, since the estimates are constructed using an increasing amount of information. Consequently, the control performance also improves.

In general, the accuracy of the parameter estimates can really increase only if the data used to perform the estimation are *sufficiently informative* (Ljung, 1999). This requires certain conditions on the input signal applied to the system while the data are generated. These conditions could be enforced for instance by superposing a special excitation signal to the normal control input (i.e. the input optimized in order to track the desired supersaturation set-point). In our case, we found that the normal control input already leads to sufficiently informative datasets. Therefore, we did not include excitation signals in the IIC algorithm.

IIC is a very powerful and general model learning approach. Owing to the structural use of the a-priori knowledge available on the process, this approach has the potential of delivering an accurate model using a limited amount of data, i.e. within a limited number of batches. This model can be utilized in order to design virtually any kind of model-based controller. Even though here we consider specifically a supersaturation tracking control problem, other objective could be similarly selected.

However, a limitation of IIC is that its attractive properties are valid only under the assumption that the process dynamics are truly described within the model

³ In other cases, the measurement are actually collected on-line, but they are not considered reliable enough for an on-line feedback control solution.

structure selected. In the case of a structural model mismatch, i.e. when the true process dynamics are not described by any of the models in the model structure, the performance delivered by IIC is in general diminished. Unfortunately, structural model mismatches cannot be ruled out for the batch crystallization process. As discussed in the previous chapter, some of the basic principles of the process are debated in the scientific community, and different model structures describing the same phenomena occurring during the process can be found in the literature.

In order to cope with the limitations of IIC in the presence of structural model mismatches, we have investigated the use of a nonparametric model update approach, namely ILC. Unlike IIC, ILC does not require strict assumptions on the model structure describing the true process dynamics, nor the presence of excitation signals. However, it does require that the initial condition and the reference trajectories are the same (or at least do not change too much) for a number of consecutive batches, and that the real-time disturbances are moderate or absent.

Different algorithms are categorized as ILC in the literature. The term ILC actually denotes a class of algorithms specifically designed to solve repeated feed-forward reference tracking problems (Bristow et al., 2006), such as the supersaturation tracking problem at hand. The ILC algorithm presented in this chapter is based on the two-step procedure first introduced in Volckaert et al. (2010). After a batch, a nominal model of the dynamics between the temperature reference T^r and the supersaturation S is updated in a non-parametric way using an additive correction term. In our case, the nominal model for the ILC algorithm is the (non-linear) moment model (2.44)-(2.48) introduced in the previous 2 with a nominal values for the uncertain physical parameters. The correction term is obtained in such a way that the updated model matches more closely the actual supersaturation during the previous batch, which is estimated using the measurement of temperature and concentration.

Owing to the flexible, non-parametric model correction, the ILC algorithm is suitable to compensate the nominal model for the structural model mismatches that could affect the true process dynamics. This is the crucial advantage of ILC with respect to IIC. However, being a nonparametric approach, ILC generally requires a larger amount of data than IIC in order to obtain an accurate model. Therefore, the convergence of the supersaturation to the desired set-point can only be achieved in a certain number of batches. Furthermore, the model corrected using ILC can be a valid approximation of the true process dynamics only along one particular trajectory. Therefore, this model is useful only to solve a specific repeated reference tracking problem, where the set-point is kept the same for a number of consecutive batches.

Another well-known drawback of ILC is the inability to cope efficiently with *real-time disturbances*, i.e. the disturbances that are different from batch to batch. To date, most of the successful applications of ILC have indeed appeared in the domain of (electro)-mechanical systems, where these disturbances are less severe than in the case of process systems (Ahn et al., 2007). For process control, the use of ILC is known to be more delicate, and generally suggested only in combination with a regular feedback control solution (Chin et al., 2004). However, the issue of these disturbances was completely ignored in previous contributions of ILC for supersaturation control in batch cooling crystallization, (see Zhang et al. (2009)

and Sanzida and Nagy (2012)), where the jacket temperature was directly adjusted from batch to batch using a similar ILC algorithm. In our B2B framework, the presence of the lower-level PI temperature controller is meant to mitigate the effect of these disturbances on the ILC.

3.2 Model of the batch cooling crystallization process

The dynamics of the batch cooling crystallization process is described by the moment model (2.44)-(2.48) already presented in Chapter 2. A compact state space representation of the model is

$$\begin{aligned} \dot{x} &= \mathcal{F}(x) + \mathcal{G}(T_J) \\ y &= \mathcal{H}(x) \\ S &= \mathcal{M}(y) \end{aligned} \quad (3.1)$$

where the states $x = (m_0 \ m_1 \ m_2 \ m_3 \ T)^\top$ are the first four moments of the CSD, and the crystallizer temperature, the input is the jacket temperature T_J , the measured outputs $y = (T \ C)^\top$ are the crystallizer temperature and the solute concentration, the control output S is the supersaturation. The state and the output mappings are given by

$$\mathcal{F}(x) = \begin{pmatrix} k_b (C - C_s(T))^b m_3 \\ k_g (C - C_s(T))^g m_0 \\ 2k_g (C - C_s(T))^g m_1 \\ 3k_g (C - C_s(T))^g m_2 \\ -\frac{UA}{\rho c_p V} T \end{pmatrix}, \mathcal{G}(T_J) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{UA}{\rho c_p V} T_J \end{pmatrix}, \mathcal{H}(x) = \begin{pmatrix} T \\ C \end{pmatrix}, \quad (3.2)$$

$$\mathcal{M}(y) = C - C_s(T) \quad (3.3)$$

where

$$C = \frac{C_i M_L - (1 - C_i) \rho_c k_v V (m_3 - m_{3,i})}{C_i M_L - (1 - C_i) \rho_c k_v V (m_3 - m_{3,i}) + (1 - C_i) M_L} \quad (3.4)$$

is the liquid concentration and

$$C_s(T) = a_o + a_1 T + a_2 T^2 + a_3 T^3 \quad (3.5)$$

represent the solubility at the temperature T .

Note that the control output S can be computed from the measured outputs y using the static relation $\mathcal{M}(y)$ defined in (3.3), i.e. $S = C - C_s(T)$.

The parameters $\theta = [k_g \ g \ k_b \ b]^\top$ in the state equation are in practice often not known with good accuracy. We assume that a nominal parameter vector $\hat{\theta}_1 \triangleq [\hat{k}_{g,1} \ \hat{g}_1 \ \hat{k}_{b,1} \ \hat{b}_1]^\top$ is available a priori. However, the *data-generating* system which represents the true crystallization system is described by a different parameter vector $\theta_o \triangleq [k_{g,o} \ g_o \ k_{b,o} \ b_o]^\top$ in order to study the performance of the algorithm in the presence of a *parametric model mismatch*.⁴ More precisely, the

⁴Later in this chapter, we will also consider the situation of a *structural model mismatch* and modify

nominal parameter vector is chosen as $\hat{\theta}_1 = [5 \ 1.1 \ 10.57 \ 1.7]^\top$, while the true parameter vector is $\theta_o = [4 \ 1 \ 12 \ 1.4]^\top$. The other coefficients appearing in the model equations are assumed to be known exactly and their numerical values are the ones reported in Table 2.1.

It will be convenient to consider the system dynamics in a finite, discrete-time representation. By applying an integration method with fixed step $t_d = 5 \text{ s}$, the input/output relation from T_J to S is represented by the static mapping $\mathbf{S} = F_{ST_J}(\mathbf{T}_J, \theta)$ relating the input vector $\mathbf{T}_J \in \mathbb{R}^N$ containing the N values of the jacket temperature T_J to the output vector $\mathbf{S} \in \mathbb{R}^N$ containing the N values of the supersaturation S at the sample points, where N is the number of samples corresponding to a batch.⁵ In this chapter, we shall adopt the bold-face notation \mathbf{W} for vectors of sampled variables $\mathbf{W} \in \mathbb{R}^N$. Furthermore, the notation $F_{WV}(\cdot)$ will be used in general to describe the mapping from an input vector \mathbf{V} to an output vector \mathbf{W} , i.e. $\mathbf{W} = F_{WV}(\mathbf{V})$. The dependence of the mapping on the uncertain parameter vector θ will be made explicit when required.

Measurements \tilde{C} and \tilde{T} of C and T are collected at the same rate $t_s = t_d$ and corrupted by the additive noise signals e_C and e_T , respectively:

$$\tilde{C} = C + e_C, \quad \tilde{T} = T + e_T. \quad (3.6)$$

The noise signals e_C and e_T are modeled as the realization of two independent white Gaussian processes having standard deviation $\sigma_T = 0.1 \text{ }^\circ\text{C}$ and $\sigma_C = 0.002 \text{ kg/kg}$. The jacket temperature T_J is perturbed by an additive, low-frequency disturbance δ_T (see Figure 3.1), which is modeled as a first-order autoregressive process having standard deviation $\sigma_{AR} = 0.25 \text{ }^\circ\text{C}$:

$$\delta_T(t+1) = a\delta_T(t) + e(t) \quad (3.7)$$

where $a = 0.9895$ and $e(t)$ is white noise with standard deviation $\sigma_e = \sigma_{AR}\sqrt{1-a^2}$. The realizations e_C, e_T and δ_T are different for all the batches.

There are several reasons to include the disturbance δ_T into the nominal temperature dynamics. Such a disturbance could be the effect of undesired interactions between the crystallizer and other pieces of equipment connected to the same utility network. For instance, the demand of coolant of the complete site can cause a disturbance on the jacket temperature (Visser, 2012). The disturbance δ_T can also be used to take into account the effect of unmodeled dynamics such as the thermal losses to the environment, which in turn depend on the crystallizer temperature and the ambient temperature.

3.3 Batch-to-batch supersaturation control

In this section we present the B2B strategies ILC and IIC. Both strategies are based on the same B2B+PI configuration that is presented in the first subsection. Subsequently, the role of the B2B controller is discussed and finally a detailed description of the ILC and IIC algorithms is given.

the structure of the model equations representing the data-generating system.

⁵Hereafter, we ignore the error which is due to the discretization of the continuous-time system.

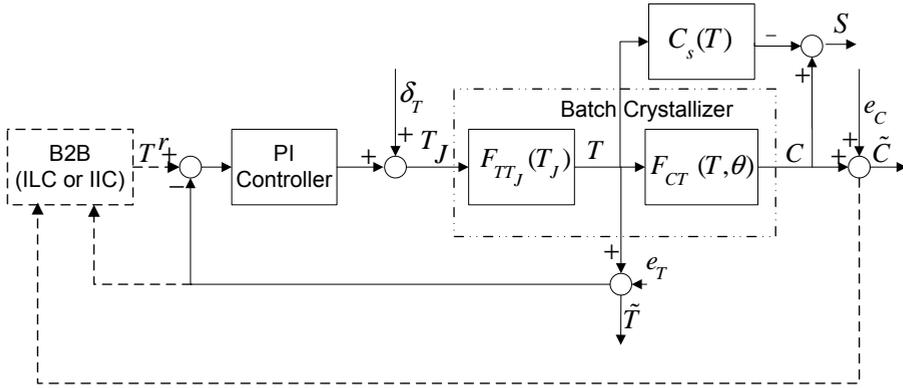


Figure 3.1: The overall B2B+PI control scheme.

3.3.1 B2B+PI Configuration

As discussed in the introduction, we included in the control scheme a PI controller for the crystallizer temperature. This controller is beneficial in order to reject the disturbance δ_T that we introduced in the previous section.

The B2B algorithm will be designed to provide a reference trajectory T^r for the PI controller, instead of providing a trajectory for the jacket temperature T_J directly. The overall B2B+PI control scheme is sketched in Figure 3.1. The two leftmost blocks represent the control system: the PI temperature controller and the B2B controller, which drives the reference of the latter. The signals coming and departing from the B2B block are updated off-line only, i.e. from one batch to the other, and are indicated by dashed lines. All other signals represented by continuous lines are updated during the batch. Note that the supersaturation S , which is the control output of the process, is the static function of the measured outputs T and C defined in Equation (3.3), i.e. $S = C - C_s(T)$ where $C_s(\cdot)$ is the solubility line (see Section 2.2.4).

3.3.2 Design of the PI controller

The design of the PI controller is here presented in the continuous-time framework for simplicity. The continuous-time transfer function of the PI controller is

$$C(s) = K_P + \frac{K_I}{s} \quad (3.8)$$

where

$$K_P = \frac{\rho c_p V}{t_{cl} U A} \quad \text{and} \quad K_I = \frac{1}{t_{cl}} \quad (3.9)$$

and $t_{cl} = 2$ min. Note that the PI controller is designed in order to move the pole of the closed-loop system to the location $\frac{1}{t_{cl}}$. Indeed, the closed-loop transfer

function $\mathcal{F}_{TT^r}(s)$ from T^r to T is

$$\mathcal{F}_{TT^r}(s) = \frac{1}{1 + \frac{s}{t_{cl}}}. \quad (3.10)$$

A discrete-time version of this PI controller having the same sampling time $t_d = 5$ s as the one used to discretize the crystallization system is implemented in the simulation model:

$$C(z) = K_P + t_s \frac{K_I}{z - 1}. \quad (3.11)$$

3.3.3 Nominal cooling trajectory

The temperature reference T^r for the first batch will be set to a linear trajectory from 38 to 10 °C in a total time of $t_f = 150$ min. The corresponding temperature and supersaturation profiles are reported in Figure 3.2. From the temperature plot, we see that the actual temperature T follows the reference T^r with a very small steady-state tracking error. Note that a nonzero steady state tracking error has to be expected for this closed-loop system, which has one integrator in the loop transfer function, in response to a linear reference trajectory. The effect of the slow disturbance δ_T is efficiently rejected by the PI controller and is not visible on the temperature T . The effect of the additive measurement noise e_T is also almost negligible and the signals \tilde{T} and T almost overlap on the plot.

On the contrary, the effect of the measurement noise e_C and e_T is much more significant on the estimate \tilde{S} of the supersaturation computed as

$$\tilde{S} = \tilde{C} - C_s(\tilde{T}). \quad (3.12)$$

A more accurate estimate of the supersaturation can be obtained noting that the measurement rate $t_s = 5$ s is rather fast compared with the dynamics of the system, which are in the order of several minutes. Therefore, it is possible to reject a significant portion of the measurement noise by low-pass filtering the measurements \tilde{T} and \tilde{C} . For this purpose, we make use of a fourth order low-pass Butterworth filter $B(z)$ having a cutoff frequency of $\frac{1}{t_f}$ with $t_f = 5$ min

$$B(z) = \frac{6.5785 \cdot 10^{-6}(z + 1)^4}{(z^2 - 1.814z + 0.8239)(z^2 - 1.913z + 0.9231)}. \quad (3.13)$$

First, the signals \tilde{T}_f and \tilde{C}_f are obtained by filtering \tilde{T} and \tilde{C} , respectively through $B(z)$. A zero-phase lag is obtained by processing the signals in both the forward and reverse directions (Mitra, 2000).⁶ Following, a filtered estimate \tilde{S}_f of the supersaturation S is computed as

$$\tilde{S}_f = \tilde{C}_f - C_s(\tilde{T}_f). \quad (3.14)$$

It is evident from the plot that \tilde{S}_f is a much more accurate estimate of actual su-

⁶Note that this filtering technique can only be performed *off-line* having the entire measurement vectors $\tilde{\mathbf{T}}_f$ and $\tilde{\mathbf{C}}_f$. However, this is not a limitation for batch-to-batch control.

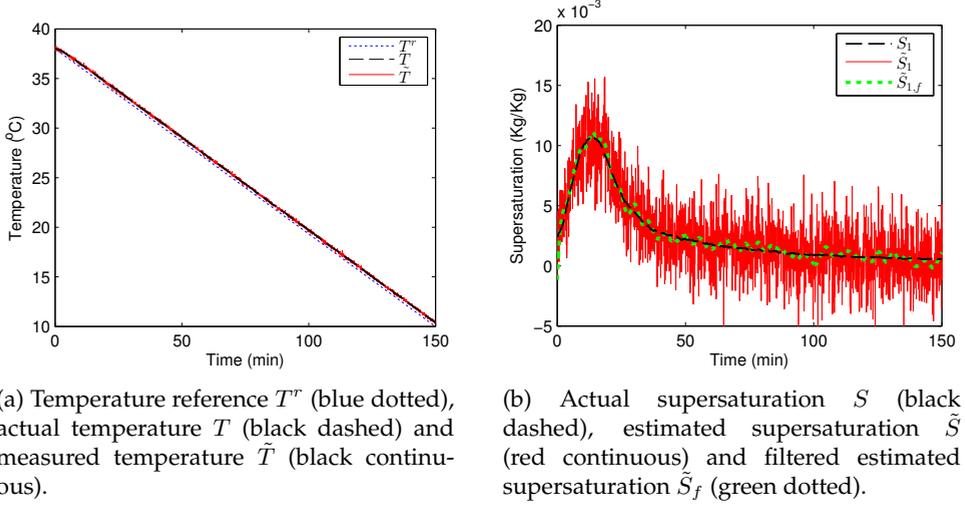


Figure 3.2: Temperature and supersaturation profiles for the first batch.

persaturation S than \tilde{S} . For this reason, in the ILC algorithm we will estimate the supersaturation according of (3.14).

3.3.4 Batch-to-Batch Control

After batch k , the corrupted measured outputs $\tilde{\mathbf{y}}_k = (\tilde{\mathbf{T}}_k, \tilde{\mathbf{C}}_k)^\top$ are available. The role of the B2B controller is to design an improved input \mathbf{T}_{k+1}^r in order to track a reference $\bar{\mathbf{S}}_{k+1}$ in the batch $k+1$. The design can be based on all the information collected up to batch k , that is given by the data collected from the previous batches and the a priori information about the system. In our case, the a priori information consists of an assumed *model structure* $F_{ST^r}(\cdot, \theta)$, the *nominal parameter vector* $\hat{\theta}_1$ and the properties of the disturbances.

The B2B algorithms IIC and ILC are described in the following.

3.3.5 Iterative Identification Control

In the IIC algorithm, the measurement $\tilde{\mathbf{y}}_k = (\tilde{\mathbf{T}}_k, \tilde{\mathbf{C}}_k)^\top$ collected from one batch are used to update an estimate $\hat{\theta}_{k+1} \triangleq [\hat{k}_{g,k+1} \hat{g}_{k+1} \hat{k}_{b,k+1} \hat{b}_{k+1}]$ for the unknown parameter vector θ_o . The IIC-updated model from the temperature reference T^r to the supersaturation S for the batch $k+1$ is defined as

$$\hat{S}_{k+1}^{\text{IIC}}(\cdot) = F_{ST^r}(\cdot, \hat{\theta}_{k+1}) \quad (3.15)$$

and is used to compute the temperature reference for the next batch as

$$\mathbf{T}_{k+1}^r = \arg \min_{\mathbf{T}^r \in \mathbb{R}^N} \left\| \bar{\mathbf{S}}_{k+1} - \hat{S}_{k+1}^{\text{IIC}}(\mathbf{T}^r) \right\|^2. \quad (3.16)$$

The estimation of the updated parameter $\hat{\theta}_{k+1}$ is performed in a Maximum Likelihood (ML) framework (Van den Bos, 2007) based on the previous estimate $\hat{\theta}_k$ and the data $\tilde{\mathbf{y}}_k = [\tilde{\mathbf{T}}_k \ \tilde{\mathbf{C}}_k]^\top$ measured in the current batch.⁷

Using the notation introduced in Section 3.2, the dynamics of the data-generating system from T to C is described by the function $F_{CT}(\cdot, \theta)$ for $\theta = \theta_o$. Formally, we should solve an Errors-in-Variables (EIV) estimation problem (Soderstrom, 2007) since both $\tilde{\mathbf{T}}_k$ (which is the input variable of the to-be-estimated dynamics) and $\tilde{\mathbf{C}}_k$ (which is the output) are corrupted by measurement noise. However, computing the ML estimator for an EIV problem is rather complicated, in particular for a generic nonlinear system such as the one describing the batch crystallization process considered here.

In this work, we construct the ML estimator ignoring the effect of the measurement error e_T on the temperature, since its standard deviation is very small compared to the amplitude of the temperature T . For this purpose, let us assume for the moment that the noise-free temperature \mathbf{T}_k is available. Then, the ML estimator is given by

$$\hat{\theta}_{k+1} = \arg \min_{\theta} \frac{1}{\sigma_C^2} \left\| \tilde{\mathbf{C}}_k - F_{CT}(\mathbf{T}_k, \theta) \right\|^2 + \left\| \theta - \hat{\theta}_k \right\|_{P_k^{-1}}^2. \quad (3.17)$$

where $F_{CT}(\cdot, \theta)$ is the uncertain dynamics from T to C , the matrices P_k for $k \geq 2$ are the covariances matrices of the estimated parameter vectors $\hat{\theta}_k$ (the covariance matrices will also be iteratively estimated, see later), and the matrix P_1^{-1} is set to 0.

For a sufficiently large value of N , the estimated parameter vector $\hat{\theta}_{k+1}$ is approximately normally distributed around θ_o with a covariance matrix P_{k+1} given by

$$P_{k+1}^{-1} = P_k^{-1} + \overbrace{\frac{\partial F_{CT}(\mathbf{T}_k, \theta)}{\partial \theta}^\top \Sigma_e^{-1} \frac{\partial F_{CT}(\mathbf{T}_k, \theta)}{\partial \theta}}^{I_k} \Bigg|_{\theta=\theta_o} \quad (3.18)$$

where $I_k \succeq 0$ is the *information matrix* relative to the experiment k (see Van den Bos (2007)).

In practice, the true parameter θ_o is not known. Therefore, the information matrix is approximated as

$$I_k = \frac{\partial F_{CT}(\mathbf{T}_k, \theta)}{\partial \theta}^\top \Sigma_e^{-1} \frac{\partial F_{CT}(\mathbf{T}_k, \theta)}{\partial \theta} \Bigg|_{\theta=\hat{\theta}_k}, \quad (3.19)$$

i.e. the derivatives of $F_{CT}(\mathbf{T}_k, \theta)$ with respect to the parameter vector θ are computed in the point $\hat{\theta}_k$, instead that in the unknown point θ_o .

As mentioned before, the noise-free signal \mathbf{T}_k is not available in practice. Therefore, we will replace \mathbf{T}_k with $\tilde{\mathbf{T}}_k$ in the formulas (3.17), (3.18), (3.19) in the IIC

⁷The previous parameter $\hat{\theta}_k$ could be interpreted as a *prior* in a Bayesian estimation framework. However, since $\hat{\theta}_k$ is obtained from a previous parameter estimation procedure using data from previous batches, we refer to our framework as Maximum Likelihood given the past and the most recent batch data (see also Remark 3.1.)

algorithm.⁸

Remark 3.1 In the ML estimation framework, P_1^{-1} is set to 0 and therefore the initial estimate $\hat{\theta}_1$ has no influence on the future estimates $\hat{\theta}_k$, $k \geq 2$ in (3.17). In general, it is also possible to set P_1^{-1} to a positive definite matrix, i.e. $P_1^{-1} > 0$. This choice would act as a regularization term penalizing the deviation of the estimates $\hat{\theta}_k$, $k \geq 2$ from the initial estimate $\hat{\theta}_1$. In a Bayesian framework, the estimator (3.17) with $P_1^{-1} > 0$ is a Maximum a Posteriori estimator where a Gaussian prior distribution having mean value $\hat{\theta}_1$ and covariance P_1 is assigned to the true parameter vector θ_o .

Remark 3.2 The inverse P_{k+1}^{-1} of the parameter covariance matrix increases from batch to batch of a term which is equal to the information matrix I_k . (see Equation 3.18). The control performance increases from batch to batch only if the eigenvalues of the information matrix are large in certain directions in the parameter space that most influence the control performance.

Summarizing, in the IIC algorithm at each batch k these steps are executed:

1. The temperature reference \mathbf{T}_k^r is set as the input to the temperature controller and the noisy measurements $\tilde{\mathbf{C}}_k, \tilde{\mathbf{T}}_k$ are collected.
2. A corrected model of the dynamics from T^r to S is defined as

$$\hat{S}_{k+1}^{\text{IIC}}(\cdot) = F_{ST^r}(\mathbf{T}^r, \hat{\theta}_{k+1}). \quad (3.20)$$

The updated parameter vector $\hat{\theta}_{k+1}$ is computed as

$$\hat{\theta}_{k+1} = \arg \min_{\theta \in \mathbb{R}^4} \left(\frac{1}{\sigma_C^2} \left\| \tilde{\mathbf{C}}_k - F_{CT}(\tilde{\mathbf{T}}_k, \theta) \right\|^2 + \left\| \theta - \hat{\theta}_k \right\|_{P_k^{-1}}^2 \right) \quad (3.21)$$

where $\Sigma_e = \sigma_C^2 I_N$. The inverse of the covariance P_{k+1}^{-1} is set to $P_{k+1}^{-1} = P_k^{-1} + I_k$ where I_k is an approximation of the information matrix computed as

$$I_k = \left. \frac{\partial F_{CT}(\tilde{\mathbf{T}}_k, \theta)}{\partial \theta} \Sigma_e^{-1} \frac{\partial F_{CT}(\tilde{\mathbf{T}}_k, \theta)}{\partial \theta} \right|_{\theta = \hat{\theta}_k}.$$

and P_1^{-1} is set to a zero matrix.

3. The corrected model is used to compute the temperature profile for the next batch as

$$\mathbf{T}_{k+1}^r = \arg \min_{\mathbf{T}^r \in \mathbb{R}^N} \left\| \bar{\mathbf{S}}_{k+1} - \hat{S}_{k+1}^{\text{IIC}}(\mathbf{T}^r) \right\|^2. \quad (3.22)$$

The optimization problems (3.21) and (3.22) are solved numerically using the active-set method of the Matlab function `fmincon`. The same single shooting

⁸We verified the validity of this approximation by performing the estimation using both the noise-free temperature signal \mathbf{T} and the noisy signal $\tilde{\mathbf{T}}$. The parameters estimated using the noise-free temperature signal \mathbf{T} were almost identical to the ones obtained using the noisy signal $\tilde{\mathbf{T}}$. The results presented in the simulation study are obtained using the noisy signal $\tilde{\mathbf{T}}$.

strategy used to solve problem (3.28) is used for problem (3.22). For problem (3.21), the derivatives of the objective function with respect to model parameters are computed analytically by integrating the *sensitivity equations* along with the model equations Rabitz et al. (1983).

3.3.6 Iterative Learning Control

In the literature, different B2B control algorithms are categorized as ILC (Bristow et al., 2006). In general, they can be described as mappings defining the input in the next batch based on the nominal model and the previous input-output data.

The ILC algorithm presented here uses as nominal model the dynamics $\hat{F}_{ST^r}(\cdot)$ obtained from the model equations (3.1)-(3.2) using the nominal parameter vector $\hat{\theta}_1$. An additive correction to the nominal model is computed after each batch considering the measurement collected in the previous batch.

Once the batch k is performed, the measurement $\tilde{\mathbf{y}}_k = (\tilde{\mathbf{T}}_k \tilde{\mathbf{C}}_k)^\top$ is used to estimate the control output \mathbf{S}_k according to (3.14). The estimate $\tilde{\mathbf{S}}_{k,f}$ is thus computed as $\tilde{\mathbf{S}}_{k,f} = \tilde{\mathbf{C}}_{k,f} - C_s(\tilde{\mathbf{T}}_{k,f})$ where $\tilde{\mathbf{C}}_{k,f}$ and $\tilde{\mathbf{T}}_{k,f}$ are obtained by filtering the signals $\tilde{\mathbf{C}}_k$ and $\tilde{\mathbf{T}}_k$, respectively through the filter $B(z)$ in both the forward and backward direction.

The ILC-updated model $\hat{S}_{k+1}^{\text{ILC}}(\cdot)$ from the temperature reference T^r to the supersaturation S for the batch $k+1$ is defined as

$$\hat{S}_{k+1}^{\text{ILC}}(\cdot) = F_{ST^r}(\cdot, \hat{\theta}_1) + \boldsymbol{\alpha}_{k+1} \quad (3.23)$$

The correction vector $\boldsymbol{\alpha}_{k+1}$ is an estimate obtained using $\tilde{\mathbf{S}}_{k,f}$ of the difference $\mathbf{S}_k - F_{ST^r}(\mathbf{T}_k, \hat{\theta}_1)$ (see later in this section for the exact definition of $\boldsymbol{\alpha}_{k+1}$). Let us assume now for simplicity that $\boldsymbol{\alpha}_{k+1}$ is exactly equal to $\mathbf{S}_k - F_{ST^r}(\mathbf{T}_k, \hat{\theta}_1)$. In this case, the output of the ILC-corrected model $\hat{S}_{k+1}^{\text{ILC}}(\cdot)$ in response to the temperature reference \mathbf{T}_k^r is equal to the output \mathbf{S}_k given by the uncertain data-generating system in response to the same temperature reference profile \mathbf{T}_k^r .

Next, the ILC-updated model (3.23) is used to optimize the reference temperature profile \mathbf{T}_{k+1}^r for the next batch according to the quadratic criterion

$$\mathbf{T}_{k+1}^r = \arg \min_{\mathbf{T}^r \in \mathbb{R}^N} \left\| \mathbf{S}^r - \hat{S}^{\text{ILC}}(\mathbf{T}^r) \right\|^2 \quad (3.24)$$

where \mathbf{S}^r is the supersaturation set-point.

Note that the ILC-corrected model $\hat{S}_{k+1}^{\text{ILC}}(\cdot)$ is just an approximation of the of the data-generating system for the batch $k+1$. In fact, the output of the data-generating system and the ILC-corrected model are the same only for the temperature reference \mathbf{T}_k^r , while a different reference \mathbf{T}_{k+1}^r obtained through (3.24) will be actually applied in the batch $k+1$. Nonetheless, the supersaturation output \mathbf{S}_{k+1} obtained with the reference profile \mathbf{T}_{k+1}^r is still expected to be closer to the set-point \mathbf{S}^r than \mathbf{S}_k . Therefore, after a number of batches, the supersaturation is expected to approach the set-point, while the temperature reference will approach the optimal value for the uncertain data-generating system.

The correction vector is actually computed as

$$\alpha_{k+1} = \arg \min_{\alpha \in \mathbb{R}^N} \left\| \tilde{\mathbf{S}}_{k,f} - (F_{STr}(\mathbf{T}_k^r, \hat{\theta}_1) + \alpha) \right\|^2 + \lambda_k \|\alpha - \alpha_k\|^2. \quad (3.25)$$

The first term in (3.25) forces α_{k+1} to be close to the difference $\tilde{\mathbf{S}}_{k,f} - F_{STr}(\cdot, \hat{\beta}_1)$. The second term in (3.25) has a regularization effect and prevents large variation for the correction vector α_{k+1} with respect to the previous correction vector α_k . On the one hand, this regularization term is beneficial in order to reduce the influence of the noise affecting $\tilde{\mathbf{S}}_{k,f}$. On the other hand, the regularization term should not be too large because the actual difference $\mathbf{S}_{k,f} - F_{STr}(\mathbf{T}_k^r, \hat{\beta}_1)$ changes from batch to batch, since the temperature reference \mathbf{T}_k also changes from batch to batch. Thus, the correction vector α_k also has to adapt from batch to batch. The scalar λ_k is a tuning parameter, possibly iteration-dependent, which represents a compromise between the two objectives.

Summarizing, for each batch k the following steps are executed:

1. The temperature reference \mathbf{T}_k^r is set as the input to the temperature controller and the noisy measurements $\tilde{\mathbf{y}}_k = (\tilde{\mathbf{T}}_k \tilde{\mathbf{C}}_k)^\top$ are collected.
2. $\tilde{\mathbf{S}}_k$ is computed as $\tilde{\mathbf{S}}_{k,f} = \tilde{\mathbf{C}}_{k,f} - C_s(\tilde{\mathbf{T}}_{k,f})$. $\tilde{\mathbf{C}}_{k,f}$ and $\tilde{\mathbf{T}}_{k,f}$ are obtained by filtering the signals $\tilde{\mathbf{C}}_k$ and $\tilde{\mathbf{T}}_k$, respectively through the filter $B(z)$ in both the forward and backward direction.
3. The updated model for the batch $k + 1$ is defined as

$$\hat{S}_{k+1}^{ILC}(\cdot) = F_{STr}(\cdot, \hat{\theta}_1) + \alpha_{k+1} \quad (3.26)$$

where α_{k+1} is computed as

$$\alpha_{k+1} = \arg \min_{\alpha \in \mathbb{R}^N} \left\| \tilde{\mathbf{S}}_k - (F_{STr}(\mathbf{T}_k^r, \hat{\theta}_1) + \alpha) \right\|^2 + \lambda_k \|\alpha - \alpha_k\|^2. \quad (3.27)$$

The correction vector used in the first batch α_1 is set to zero.

4. The corrected model is used to compute the temperature profile for the next iteration

$$\mathbf{T}_{k+1}^r = \arg \min_{\mathbf{T}^r \in \mathbb{R}^N} \left\| \bar{\mathbf{S}}_{k+1} - \hat{S}_{k+1}^{ILC}(\mathbf{T}^r) \right\|^2 \quad (3.28)$$

where $\bar{\mathbf{S}}_{k+1}$ is the supersaturation set-point for the batch $k + 1$.

The optimization problem (3.28) is solved numerically using the active-set method of the Matlab function `fmincon`. The optimization is based on a *single shooting* strategy that is discussed in Appendix A.

3.4 Simulation Results

In this section we evaluate the performance of the B2B scheme described in the previous section on two different test cases, namely in presence of parametric

model mismatch only (Case 1) and in presence of a structural mismatch (Case 2). The batch time is $t_f = 150$ min and the total number of batches is $n = 30$. The desired supersaturation is the constant value $\bar{S}_k = 0.0025$ kg/kg for the first 10 batches and is changed from batch 11 to 30 to a parabola passing through the points $(t, S) = \{(0, 0.0025), (100, 0.0012), (150, 0.005)\}$ (min, g/L).⁹

The temperature reference at the first batch T_1^r is set to a linear cooling trajectory from 38 to 10 °C in the time interval $[0, t_f]$, as already shown in Figure 3.2. For the ILC algorithm, the parameter λ_k is chosen as

$$\lambda_k = \begin{cases} 0, & k = 1 \text{ and } k = 11 \\ 1, & k = 2, \dots, 5 \text{ and } k = 12, 13, \dots, 15 \\ 5, & k = 6, 7, \dots, 10 \text{ and } k = 16, 17, \dots, 29. \end{cases} \quad (3.29)$$

Note that λ_k is zero for the first batch and is kept “small” ($\lambda_k = 1$) in the following 4 batches. Indeed, a large mismatch is expected for the initial model and the objective of matching the previous measurement is considered more important than rejecting the disturbances in (3.25). For the batches 6, 7, ..., 10, the mismatch is expected to be lower and the tuning parameter is increased ($\lambda_k = 5$) in order to further improve the tracking performance by filtering out the disturbances from the correction vector. After the set-point change, the model mismatch is expected to be large again and λ_k is set to zero for the batch 11. Similarly, we set $\lambda_k = 1$ for the batches 12, 13, ..., 15 and $\lambda_k = 5$ for the batches 16, 17, ..., 29.

For the IIC algorithm, we found that for this system, at the given experimental conditions, variations in the parameters k_b and b related to the nucleation behavior have an almost negligible effect (close to numerical precision) on the measured output $y = (C T)^\top$.¹⁰ As a consequence, the result of the estimation for these parameters is highly unreliable and strongly depends on the choice of the initial point for the optimization. This was reflected by values close to zero for the third and fourth diagonal entries of the information matrix I_k computed according to (3.19) for the full parameter vector θ computed using the data of the first batch.

Performing parameter estimation with a model whose parameters are close to being unidentifiable may lead to numerical issues in the optimization. Furthermore, the same covariances estimated according to (3.18) for the full parameter vector θ may be very inaccurate in this case. In order to avoid these issues, we fixed the value of k_b and b to their nominal value and performed the parameter estimation for a *reduced* parameter vector $\beta = (k_g g)$.¹¹

⁹Note that a constant supersaturation is a common objective for a batch cooling crystallization process (Gutwald and Mersmann, 1990). Conversely, the parabolic set-point is a rather arbitrary choice used here to study the effect of a significant change in the set-point on the performance of the two B2B algorithms.

¹⁰From a physical perspective, this is explained by the fact that we are simulating a *full seeded* crystallization process (see Section 2.2.8 in the previous chapter) where most of the crystal mass produced is the result of the growth of the initial seeds. Therefore, the measured concentration (which is related to the crystal mass produced) contains little information about the parameters related to the nucleation phenomenon.

¹¹Alternatively, one could estimate for the full parameter vector θ by adding a regularization term

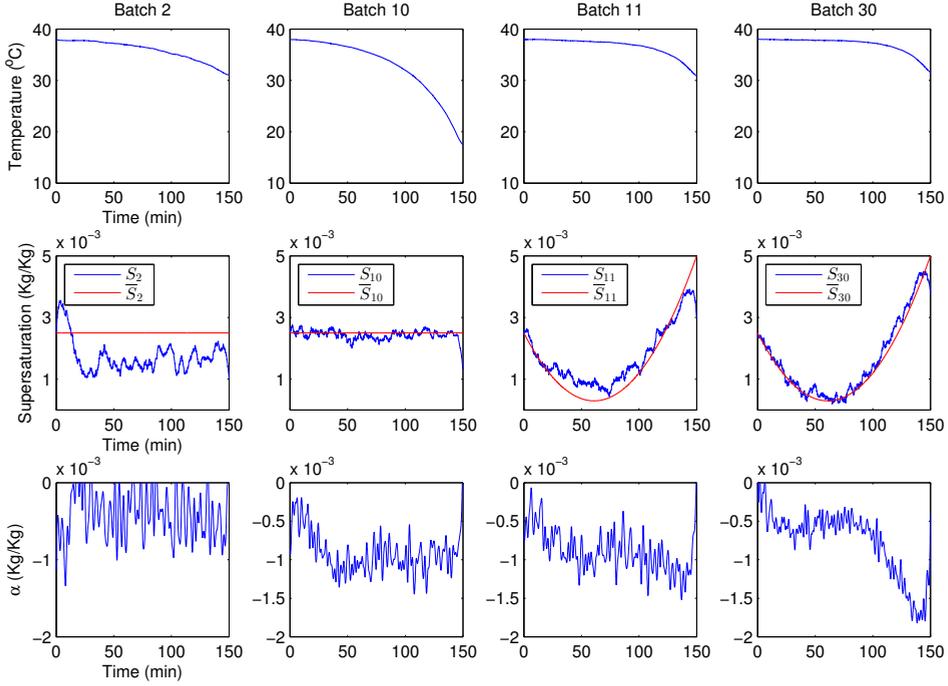


Figure 3.3: Case 1 ILC: temperature reference T^r , supersaturation S , and correction vector α for the batches 2, 10, 11, and 30.

3.4.1 Case 1

In this subsection, the supersaturation tracking performance obtained using the two B2B algorithms in presence of parametric model mismatch is evaluated. The IIC algorithm, which performs a parametric update based on a correct assumption on the correct model structure, is expected in this case to provide better results compared to ILC, which performs a more generic nonparametric correction.

As mentioned in Section 3.2, the data generating system is constructed based on the system equations (3.1)-(3.2) with the true parameter vector $\theta_o = [4 \ 1 \ 12 \ 1.4]^\top$, while the nominal parameter vector $\hat{\theta}_1 = [5 \ 1.1 \ 10.57 \ 1.7]^\top$ is used in the design of the ILC algorithms. For the IIC algorithm, as discussed in the previous subsection the parameters k_b, b are fixed to the nominal values $\hat{k}_{b,1}, \hat{b}_1$, respectively, and the reduced parameter vector $\beta = (k_g \ g)^\top$ is estimated after each batch.

The results of the simulations are shown in Figure 3.3 for ILC and Figure 3.4 for IIC. We report for different batches the temperature reference profile T_k^r and the supersaturation S_k , together with the supersaturation set-point \bar{S}_k . In the case of ILC, we also report the correction vector α_k .

In Batch 2 the tracking error is already small for IIC, while it is still appreciable for ILC. This confirms the intuition that IIC can be more efficient than ILC in the

penalizing the deviation of k_b and b from their nominal value, as discussed in the Remark 3.1.

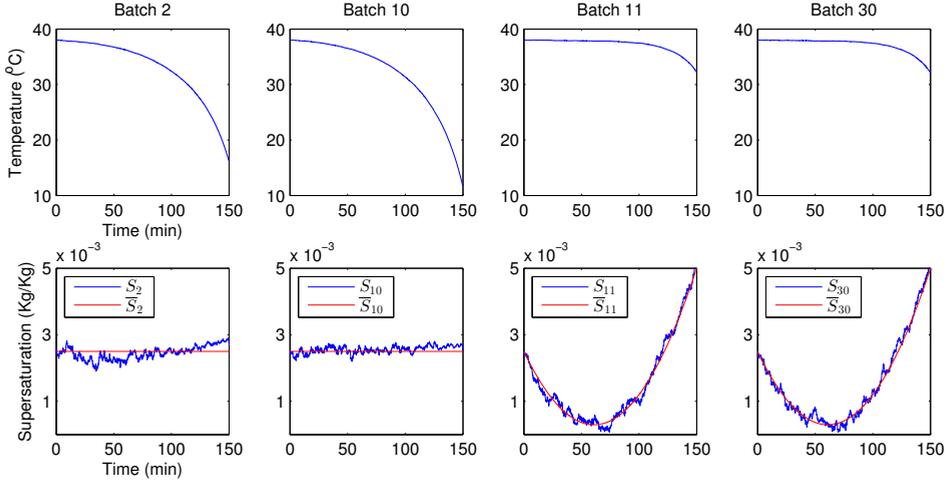


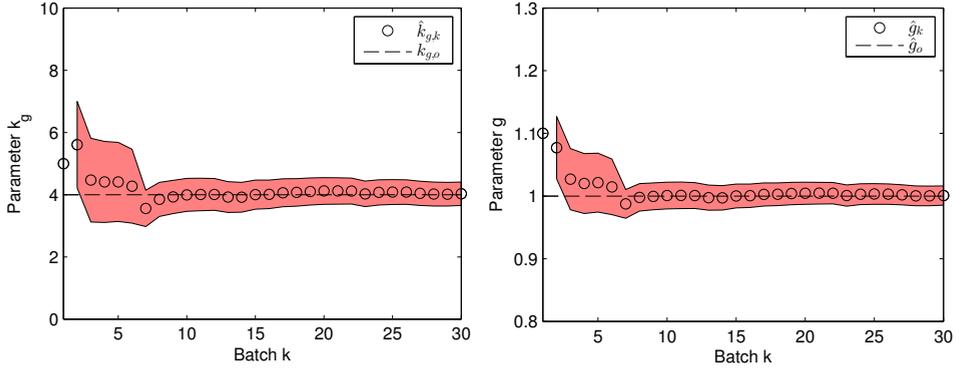
Figure 3.4: Case 1 IIC: temperature reference T^r and supersaturation S for the batches 2, 10, 11, and 30.

case where the model-structure used for parameter estimation can describe the data-generating system. After some iterations, also ILC approaches the set-point more closely (Batch 10). However, the performance sensibly degrades in Batch 11 for ILC due to the set-point change. This can be expected since the additive model correction based on the correction vector α_k is accurate only around one particular trajectory. Few iterations of the ILC algorithm are required in order to adapt the correction vector in order to compensate for the model mismatch along the new trajectory, and eventually a good tracking performance is recovered (Batch 30). Note also that the vector α_k is smoothed during the iterations owing to larger value of the tuning parameter λ_k . In the IIC case, the set-point change does not lead to any performance loss in Batch 11. Indeed, the parametric model correction is valid for all the possible system trajectories.

For the IIC algorithm, we also computed the standard 95% *confidence intervals* of the parameters $\hat{k}_{g,k}$ and \hat{g}_k for $k \geq 2$.¹² The confidence intervals are numerical intervals centered around the estimated parameters $\hat{k}_{g,k}$ and \hat{g}_k , respectively where the true parameters $k_{g,o}$ and g_o are expected to lie with probability 95%. The confidence interval are obtained based on the covariance matrix P_k which in turn is computed according to (3.18) (for the reduced parameter vector β instead of θ). For instance, the 95% confidence interval \mathcal{D}_{g_k} of \hat{g}_k is computed as $\mathcal{D}_{g_k} = [\hat{g}_k - 1.96\sigma_{g_k}, \hat{g}_k + 1.96\sigma_{g_k}]$ where $\sigma_{g_k} = \sqrt{P_k(2,2)}$ is the standard deviation of \hat{g}_k . The true coefficient g_o is thus expected to lie with probability 95% inside this interval.

In Figure 3.5 we report the estimated parameters, together with their 95% confidence interval, and the true value of the parameters. It is shown that k_g and g are effectively estimated in the IIC algorithm. The width of the confidence inter-

¹²The uncertainty intervals cannot be defined for $k = 1$ since there is no probability distribution associated to the nominal parameter vector.



(a) Estimated parameter \hat{k}_g vs batch number. (b) Estimated parameter \hat{g} vs batch number.

Figure 3.5: Estimated parameters $\hat{k}_{g,k}$ and \hat{g}_k of the model (3.2) in the IIC algorithm vs batch number k in Case 1. The colored area represents the standard 95% confidence intervals of the estimates.

val reduces significantly over the first 7 batches. The true parameters lie inside the 95% confidence intervals for all the batches, apart from the batch $k = 2$.

3.4.2 Case 2

On top of the parametric mismatch already considered in Case 1, a structural model mismatch is here considered. A temperature-dependent crystal growth mechanism is introduced in the data-generating system. In order to model such behavior, the second, third, and fourth elements of the state-space map (3.2) of data-generating system are multiplied by a term $A_0 \exp(-E_a/R(T + K_0))$, with $A_0 = 1.3 \times 10^7$, $E_a = 4.2 \times 10^4$ J/mol, $R = 8.3144$ J/mol·K and $K_0 = 273.15$ °C.¹³ The modified state equations are

$$\mathcal{F}(x) = \begin{pmatrix} k_b (C - C_s(T))^b m_3 \\ A_0 \exp(-E_a/R(T + K_0)) k_g (C - C_s(T))^g m_0 \\ A_0 \exp(-E_a/R(T + K_0)) 2k_g (C - C_s(T))^g m_1 \\ A_0 \exp(-E_a/R(T + K_0)) 3k_g (C - C_s(T))^g m_2 \\ -\frac{UA}{\rho c_p V} T \end{pmatrix}. \quad (3.30)$$

This structural model mismatch is expected to diminish the performance of the IIC algorithm, even though the extent of the performance degradation is difficult to be quantified a-priori. On the contrary, ILC is expected to be less sensitive to the structural mismatch, since the algorithm does not explicitly use any assumptions on the model structure in the computation of the model update.

The results of the simulations are reported in Figure 3.6 for ILC and Figure 3.7 for IIC. As in Case 1, we report for a number of batches the temperature reference T_k^r , the supersaturation S_k , and the supersaturation set-point \bar{S}_k . For the ILC

¹³This kind of temperature dependence is known as Arrhenius-type in literature (Luyben, 2007).

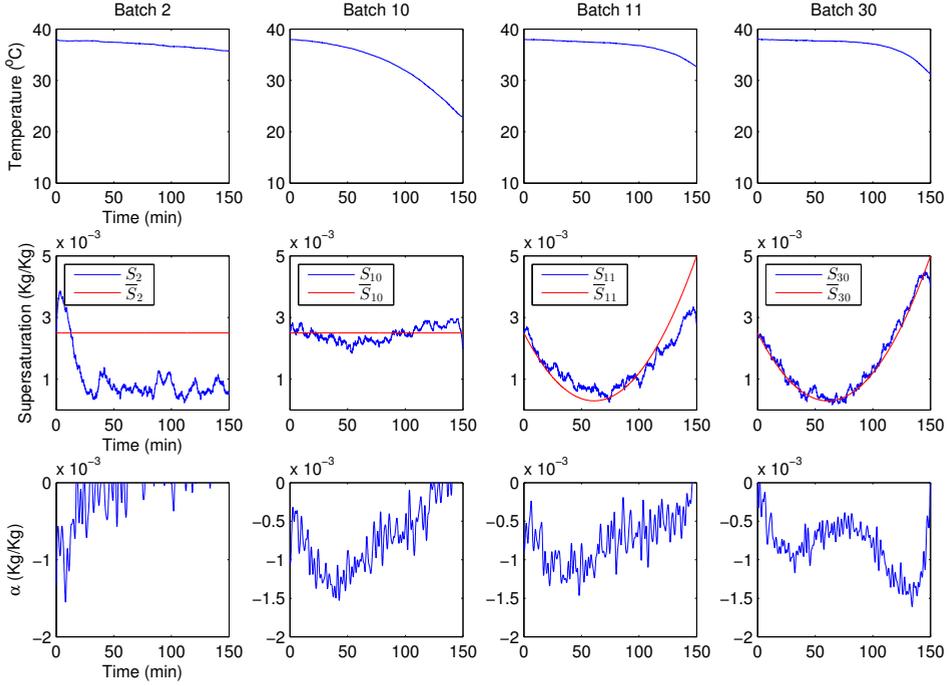


Figure 3.6: Case 2 ILC: temperature reference T^T , supersaturation S , and correction vector α for the batches 2, 10, 11, and 30.

algorithm, we also report the correction vector α_k .

The IIC algorithm leads to a rather poor tracking performance in the first 10 batches (Batch 1, 10). Apparently, due to the structural model mismatch the estimated models cannot represent the dynamics of the data-generating system with a sufficient accuracy in order to track the desired supersaturation set-point. In Batch 11 a better result is achieved. The cause of the better result is that the tracking of the new set-point requires a lower temperature variation compared to the previous case. Therefore, the mismatch of the nominal model (that does not incorporate the temperature-varying growth behavior) is less detrimental in these conditions. Note that this better result is not caused by a learning mechanism. Indeed, no further improvement is obtained in the following batches (Batch 30).

On the contrary the ILC algorithm is still capable to approach the set-point, even though a number of iterations are required (Batches 10, 30). The flexible, nonparametric model correction performed in ILC using the correction vector α_k can compensate the nominal model for a large class of structural and/or parametric model mismatches. For the IIC algorithm, we report in the Figure 3.8 the estimated parameters $\hat{k}_{g,k}$, $\hat{g}_{k,r}$ together with their 95% confidence interval (for $k \geq 2$), and the true coefficient $k_{g,o}$, g_o .

Note that the confidence intervals do not have a rigorous mathematical meaning in this case since the expression of the covariance P_k (3.18) is not valid when the model structure used cannot describe the data-generating system. Moreover,

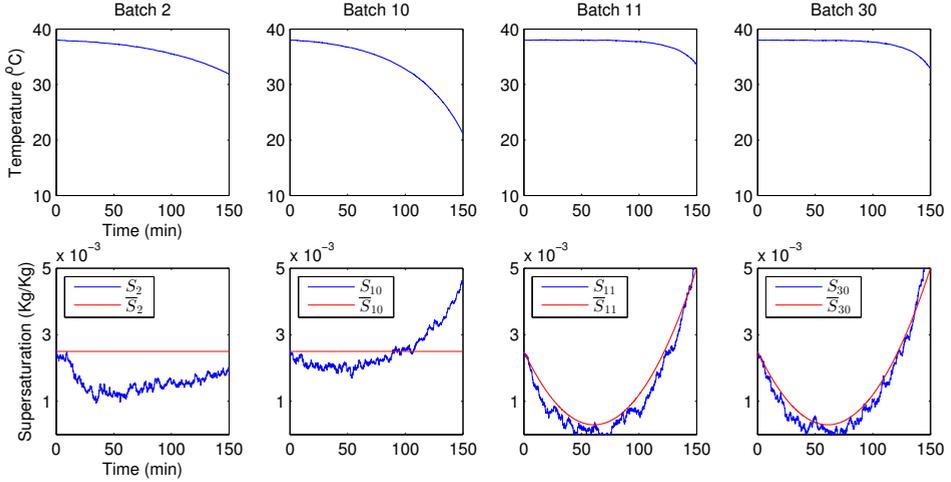
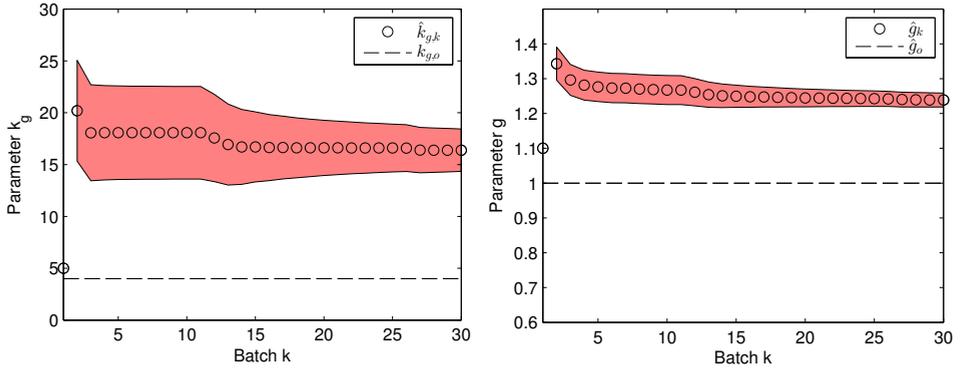


Figure 3.7: Case 2 IIC: temperature reference T^r and supersaturation S for the batches 2, 10, 11, and 30.

the parameters $\hat{k}_{g,k}$, \hat{g}_k cannot be formally compared to the true parameters $k_{g,o}$, g_o since the latter represent a model in a different model structure than the one assumed for the estimation. Nonetheless, it is still interesting to observe the results given by this kind of analysis, which is often performed in practice without having the possibility of verifying whether or not the model structure selected for the estimation can describe the true system.

The confidence intervals appear to be larger than in Case 1, but they are still reducing throughout the iteration. However, the estimated parameters $k_{g,k}$, b_k seem to converge to values that are different than the “true” ones $k_{g,o}$, b_o . In fact, the true parameters are not contained in the confidence intervals for any of the iterations.

This result is not surprising, since we are in the case of a structural model mismatch. The estimated parameters converge to the ones describing a model that is the *best approximation* of the data-generating system within the assumed model structure. The best approximating model depends in general on the experimental conditions (i.e. input and initial condition of the system) and on the estimation criterion. Formal results about estimation in the case of structural model mismatches are well established in the situation when both the assumed model structure and the data-generating system are linear, and a Prediction Error Identification criterion is used (Ljung, 1999). In this case, the best approximating model is characterized by frequency-domain integral expressions involving the *spectrum* of the input signal. Some results also exist for the case when the data-generating system is nonlinear, but the assumed model structure is linear. The properties of the best approximating linear model (also called *best linear approximation*) of a nonlinear system have been analyzed in Schoukens et al. (2005). Interestingly, it was found that (unlike in the linear case) the input spectrum is not sufficient to characterize the best linear approximations of the nonlinear system, and also higher



(a) Estimated parameter \hat{k}_g vs batch number. (b) Estimated parameter \hat{g} vs batch number.

Figure 3.8: Estimated parameters $\hat{k}_{g,k}$ and \hat{g}_k of the model (3.30) in the IIC algorithm vs batch number k in Case 2. The colored area represents the standard 95% confidence interval of the estimates. A bias can be observed for the parameters k_g and g due to the structural model mismatch.

order statistical properties of the input influence the result.

However, in a full nonlinear framework as the one presented here, it is very hard to analyze the properties for the best approximating model.

Remark 3.3 If the “correct” structure (3.30) of the state equations was known in Case 2, the additional parameters A_0 and E_a could be estimated together with the growth parameters k_g and g in the IIC algorithm. By doing this, the IIC-updated models would converge to the data-generating system and a similar performance as in Case 1 would be achieved. However, in practice, the correct model structure is not always known. For this reason, it is interesting to study the performance of the IIC algorithm in presence of a significant structural model mismatch.

3.4.3 Overall Results and Discussion

The Root Mean Square value of the supersaturation tracking Error (RMSE) is plotted against the iteration number for all cases in Figure 3.9. The different behavior of the two algorithms is evident in this plot. IIC provides the best performance when the model structure selected can describe the data-generating system. A good result is already obtained after the first iteration of the algorithm. Furthermore, the performance is not influenced by the set-point change since the algorithm learns the full model structure (Case 1 IIC). However, the behavior of the algorithm is hard to predict (and the performance is in general lower) in the case of structural model mismatches.

The IIC framework indeed strongly relies on the assumption of the model structure. In the situation considered where a temperature-dependent growth behavior is incorporated in the data-generating system and the model structure for estimation is incorrect, IIC leads to a worse overall performance compared to ILC.

Indeed, ILC is much more robust to model structure mismatches due to the

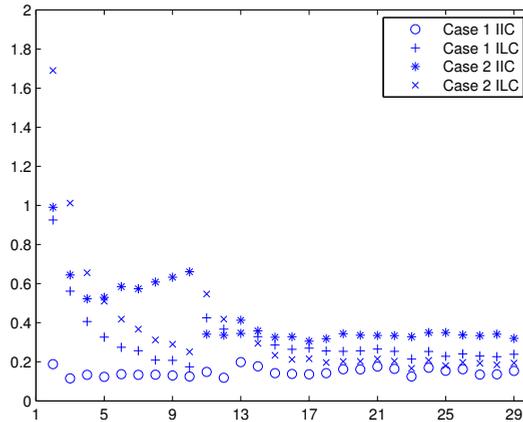


Figure 3.9: RMSE of the supersaturation vs. batch number for IIL and ILC in Case 1 and 2.

more generic additive correction performed herein. Even though these mismatches make the convergence somewhat slower in Case 2, a satisfactory result is eventually reached. However, after the set-point change more iterations are needed in order to come close to the optimum again since the additive correction is a valid approximation of the real dynamics only locally around a particular trajectory.

The complexity of the two solutions has to be considered both in terms of the computational effort required and in terms of the ease of design. The design of the parameter update in IIC is rather straightforward by adopting the Maximum Likelihood approach. However, such an estimation requires the numerical solution of a nonlinear least-squares problem. On the other hand, the model update in ILC does not require particularly complex computations. However, the tuning of the algorithm is more delicate and still requires some trial-and-error.

3.5 Iterative control and dual control

A model update based on parameter estimation generally requires the use of a sufficiently informative dataset. Intuitively, the influence of all the to-be-estimated parameters has to be clearly distinguishable in the measured output in order to be able to perform an effective estimation. More formally, the information content of a dataset can be quantified in the terms of the information matrix (3.19), which depends on the system input. A large information matrix guarantees that the covariance of the estimated parameter vector is small, i.e. the model parameters can be accurately estimated.

In practice, ad-hoc experiments where the input is chosen e.g. as a step, a (pseudo)random binary signal or a white noise signal are often executed in order to obtain an informative dataset. A more advanced solution would be to explicitly design the input in order to optimize a measure of the information matrix, as often done in the field of Experiment Design (Atkinson and Donev, 1992; Fedorov and Hackl, 1996).

Note however that in the IIC algorithm the parameter estimation procedure is performed using the data collected from regular batch experiments. In these experiments, the input is determined according to the control objective only, i.e. in order to track the desired supersaturation. In our case, such input led to datasets which happened to be sufficiently informative. Even though the input trajectory was optimized for the control objective, it also provided enough excitation in order to estimate with good accuracy the parameters k_g and g , which determine the supersaturation dynamics to a large extent in our case. Therefore, the models updated through the parameter estimation procedure were adequate in order to satisfy the control objective.

A more advanced approach would be to design the input signal with the dual objective of tracking the desired set-point (control objective) and generating an informative dataset for parameter estimation (identification objective). In an iterative identification/control framework such as the one presented for batch-to-batch control, this approach is particularly appealing if one recognizes that the accuracy of the updated model influences the control performance that can be achieved *for the future batches*, where the updated model will be used to define the control action. The input is said to have a *dual effect* (Bar-Shalom and Tse, 1974; Tse and Bar-Shalom, 1973; Tse et al., 1973) since it influences both directly the performance of the current batch, and indirectly (through the identified model) the performance for the future batches. Optimizing the overall performance taking explicitly into account the dual effect of the input is a tough problem known as the *dual control* problem. For this problem, approximate solutions have been found only in few specific cases of model structures and control objectives (Kulcsar et al., 2009; Pronzato et al., 1996).

The link between the accuracy of an identified model and the performance expected from a control strategy based on such models has also been studied in the field of *Identification for control* (Bombois et al., 2006; Gevers, 2005). One of the findings was that often few control-relevant properties of the model influence the control performance to a large extent. Furthermore, these properties are often naturally emphasized by performing the identification in a situation which resembles the desired controlled behavior of the system. For these reasons, iterative schemes consisting of successive identification and controller re-design step were found to be suitable in order to gradually improve the most important control-related characteristic of the identified model (Garatti et al., 2010; Hjalmarsson et al., 1996). In this sense, the use of iterative identification/control schemes is not only limited to batch-to-batch control applications, but can also be beneficial for the gradual improvement of the control performance of a continuous process.

In Chapter 4, we will build up on the results established in the field of Identification for Control and, based on modern experiment design tools (Bombois et al., 2006; Jansson and Hjalmarsson, 2005), we will develop an approximate solution to the dual control problem in an iterative identification/control scheme. Compared to the previous solution appeared in the literature, our solution can be applied for a fairly large class of model structures describing linear dynamical system, and for different control objectives.

The bottleneck for the application of similar techniques to general, nonlinear dynamical systems such as the one describing the batch cooling crystallization is

the lack of efficient experiment design tools which can handle these systems. In this direction, preliminary results of experiment design for nonlinear dynamical systems have been developed in this thesis, and will be presented in Chapter 5.

3.6 Conclusion

We have presented a batch-to-batch (B2B) solution for supersaturation control in batch cooling crystallization. The B2B controller drives the reference of the PI temperature controller in the B2B+PI configuration. Two B2B algorithms are examined in this paper, namely an Iterative Learning Control (ILC) and an Iterative Identification Control (IIC). The behavior of system in the B2B+PI configuration with the two solutions has been discussed and analyzed in a simulation study.

The IIC and ILC algorithms are shown to have complementary advantages and disadvantages. On the one hand, IIC provides the best performance when the assumed model structure can actually describe the data-generating system. Furthermore, the performance is not sensitive to set-point changes. However, the performance of IIC is hard to predict (and generally lower) when the true system is not contained in the assumed model structure, i.e. in the case of a structural model mismatches. On the other hand, ILC is more robust to structural model mismatches. Even though these mismatches slow down the convergence, a satisfactory result is eventually obtained after a number of batches. However, the performance deteriorates after a set-point change and more iterations are required in order to recover the previous performance level. Furthermore, the tuning of the ILC algorithm is more delicate than the one of IIC and may require some trial-and-error. For the reasons above, it would be useful to design a supervisory algorithm in order to switch from one strategy to the other based on the results obtained in the previous batches.

Finally, we have noticed that in an iterative control framework such as batch to batch control, the input for one batch has a dual effect on the performance in the current and future batches. On the one hand, it influences directly the performance achieved in the current batch. On the other hand, it influences indirectly the performance that will be achieved in the future batches through the model that will be used to define the control action for the future batches, which is updated based on the data measured in the current batch. The B2B algorithms presented in this chapter ignore the dual effect of the input as they design the input only according to the control criterion for the batch. In the following chapter, we will present an iterative control algorithm where the input is specifically designed in order to optimize the performance in the current and future iterations taking the dual effect explicitly into account.

Iterative model improvement for model-based control

In this chapter, we present a framework for the gradual improvement of model-based controllers. The total time during which the system will be operated is divided into a number of learning intervals. After a learning interval, the model is refined based on the measured data. This model is used to synthesize the controller that will be applied during the next learning interval. Excitation signals can be injected into the control loop during each of the learning intervals. On the one hand, the introduction of an excitation signal worsens the control performance during the current learning interval since it acts as a disturbance. On the other hand, the informative data generated owing to the excitation signal are used to refine the model using a closed-loop system identification technique. Therefore, the control performance for the next learning interval is expected to improve. Our objective is to maximize the overall control performance taking the effect of the excitation signals explicitly into account. However, this is in general an intractable optimization problem. For this reason, a convex approximation of the original problem is derived using standard relaxations techniques for Experiment Design. The approximated problem can be solved efficiently using common optimization routines. The applicability of the method is demonstrated in a simulation study.¹

4.1 Introduction

It is well known that the performance of a model-based controlled system largely depends on the quality of the model that is used to synthesize the controller. System identification provides tools that can be used to construct models using measured input/output data. Together with the model, most of the identification methods also provide a measure of the model uncertainty.

¹This chapter is based on the results in Forgione et al. (2013) and Forgione et al. (2014b).

The relation between the uncertainty of an identified model and the expected control performance has been studied in the field of Identification for Control (Gevers, 2005). An important finding was that often few dominant features of the model determine the performance of the controller to a large extent (Gevers, 2002). Therefore, the identification experiments have to be designed in such a way that these control-relevant features can be accurately identified. In the earliest contributions, the identification experiments and the normal operation of the system were considered as completely distinct phases. In fact, the identification experiments were collected in open loop, while the model was used to design a closed-loop controller for the system (Gevers and Ljung, 1986).

During the nineties, closed-loop identification techniques (Van den Hof and Schrama, 1995) gained increasing attention in the Identification for Control community. An immediate advantage of closed-loop identification is that the normal (closed-loop) operation can continue while the identification data are collected. In this sense, the closed-loop identification procedures are intrinsically less intrusive than the ones based on open-loop data.

The use of closed-loop identification was also supported by the intuition that the control-relevant features of the system are naturally emphasized when the system is operating in a condition which resembles the desired controlled behavior. In specific cases, it was formally proven that the optimal experimental condition for control-oriented identification are met when the optimal controller (with respect to the same control objective) is present in the loop (Gevers, 2002; Hjalmarsson et al., 1996).

Note however that this optimal controller is always unknown. In fact, its determination is the ultimate goal of the user. Therefore, the optimal experimental conditions can only be approached by adopting *iterative* schemes consisting of repeated closed-loop identification and model-based control design steps (De Callafon and Van den Hof, 1997; Schrama, 1992).

An aspect that was not thoroughly investigated in these contributions was the choice of the *excitation signals* fed to the system during the closed-loop experiments, often simply taken as white noise signals for ease of analysis. However, a careful choice of the excitation signals can be beneficial both to improve the model accuracy and to limit the cost of the identification (Bombois et al., 2006).

On the one hand, a high level of excitation leads to informative data sets which can be used to identify accurate models. On the other hand, the excitation signals also act as disturbances on the controlled system and consequently lead to a (temporary) performance degradation when they are applied. Therefore, there is a trade-off between the performance degradation due to the application of the excitation signals and the improvement that is expected due to the increased model accuracy. In the literature, the excitation is said to have a *dual effect* on the control performance (Tse and Bar-Shalom, 1973) for this reason.

In this chapter, we consider the problem of designing the excitation signals in an iterative identification/controller design scheme aiming to maximize the overall control performance, while guaranteeing a minimum performance level at all times. In our framework, the total time of the closed-loop operation is divided into a number of *learning intervals*. Excitation signals can be injected into the control loop during each of these intervals. After an interval, the measured data are

used to refine the model using closed-loop identification. Based on that model, a new controller is designed. The controller is applied during the next learning interval, and so on and so forth for the following ones.

We define the cost \mathcal{T}_k of one interval k as the sum of the performance degradation due to the difference between the true system and the identified model (modeling error cost), and the one due to the presence of the excitation signal (excitation cost) during that interval. Following, we determine the excitation signals to be applied in each interval by minimizing the sum $\sum_k \mathcal{T}_k$ of the cost over all the intervals, subject to constraints $\mathcal{T}_k \leq \bar{\mathcal{T}}_k$ on the cost for each interval. In practice, these constraints could be useful in order to ensure the safety of the plant and in order to honor specifications which guarantee minimum product quality.

Note that the modeling error cost in one interval depends on the excitation signals applied during all the previous intervals since the model is identified based on the previous data, while the excitation cost depends on the excitation signal applied during the current interval. Thus, we are here taking the dual effect of the excitation signals explicitly into account.

It has to be mentioned that the optimization problem that we would like to solve in order to find these excitation signal is intractable as such. However, using established relaxation techniques and tools developed in the field of Experiment Design, we can derive an approximation of the original optimization problem that is convex and can be solved efficiently.

In fact, the problem of designing excitation signals which guarantee desired properties for the identified model has been extensively studied in the Experiment Design field. The classic approaches for experiment design consider only two distinct phases: an identification phase in which the excitation signal is fed to the system and a model is identified, and a control phase in which a controller based on the identified model is applied. The objective generally considered is to find a compromise between the excitation cost in the identification phase and the modeling error cost in the control phase (Bombois et al., 2006; Gevers and Ljung, 1986).

A limit of the classic approaches is that if the identification phase is too short, it might not be possible to satisfy the performance requirement for the control phase without violating constraints on maximum level of excitation in the identification phase. It is possible circumvent this issue by extending the duration of the identification phase, but this implies that one has to wait a longer time before having any improvement in the control performance.

Our approach can be seen as an extension of the classic experiment design approaches to a situation with several phases (i.e. the learning intervals), and in which the dual effect of the excitation signal is considered altogether for all the learning intervals. As we will show in the numerical example, by considering several learning intervals we can gradually improve the controller and achieve a better overall performance than in a classic two-phase framework.

Our approach has also a certain analogy with the *actively adaptive* learning algorithms discussed in Pronzato et al. (1996), since it takes explicitly into account the dual effect of the excitation. However, the approach in Pronzato et al. (1996) leads to stochastic dynamic optimization problems that can be solved only for very specific model structures and control objectives, while our approach can be

applied to (almost) any LTI model structure and control objective. There are indeed important differences between our approach and the one in Pronzato et al. (1996). First, in Pronzato et al. (1996) the input is optimized for the identification and the control objective altogether, while in our framework the controller takes care of the control objective and the superposed excitation signal takes care of the identification objective. Second, the model is updated at each time instant in Pronzato et al. (1996), while we perform the identification only at the end of a learning interval. These simplifications allow us to use the classical experiment design tools to tackle this complicated problem in a wider range of cases.

The rest of this chapter is organized as follows. In Section 4.2 the framework is discussed in details. In Section 4.3 the experiment design problem is introduced and the approximated convex optimization problem is derived. The framework is applied to a simulation study in Section 4.5 and conclusions are drawn in Section 4.6.

4.2 The Framework

The *true system* S_o is the linear time-invariant system

$$y = G_o(q^{-1})u + \overbrace{H_o(q^{-1})e}^{=v} \quad (4.1)$$

where u is the input, y is the output, e is white noise with variance σ_e^2 , and q^{-1} is the unit-delay operator. G_o and H_o are stable discrete-time transfer functions; H_o is monic and minimum phase.

S_o is known to belong to a *model set* $\mathcal{M} = \{M(\theta), \theta \in \mathbb{R}^p\}$ where θ is the model parameter. We assume that the true system S_o is described in \mathcal{M} by a (unique) *true parameter* θ_o , i.e. $\exists! \theta_o \mid S_o = M(\theta_o)$.

In order to reject the disturbance $v = H_o e$, we would like to operate (4.1) in closed loop. We assume that an initial controller $C(\hat{\theta}_1)$ is available and that it stabilizes the true system. However, this controller has been designed with a model $M_1 = M(\hat{\theta}_1)$ that is a relatively poor representation of the true system, and the performance of the initial loop $[C(\hat{\theta}_1) S_o]$ is rather poor.

In the sequel, we will present an iterative model and controller update procedure whose objective is to gradually improve the accuracy of the model and the control performance. For this purpose, the period of time during which (4.1) will be operated is divided into n learning intervals of duration N . During each interval, a specially tailored excitation signal r_k is applied to the loop. At the end of the interval, input and output data are collected and are used to gradually improve the control performance by finding a better model (with respect to the desired control application) within the model set \mathcal{M} .

Based on the improved model, the controller is updated and applied to the true system for the next interval. The procedure is illustrated in Figure 4.1 and will be presented in more details in the next subsection.

Remark 4.1 *If the initial model M_1 is too poor for the design of a stabilizing controller for S_o , the first interval can be performed in open loop.*

Remark 4.2 Our framework can be readily extended to the case of reference tracking. A nonzero set-point for the controller would provide additional excitation to the system, while no cost is associated to its application. For this reason, the excitation signals r_k are generally smaller or even are not required in this case.

Remark 4.3 In principle, a better performance can always be obtained by reducing the interval length N . The controller and the excitation signals could be updated more frequently using a more recent parameter estimate. In this sense, the best choice would be to set N to just one time sample. However, in the development of the experiment design procedure we will make use of properties of prediction error identification which are asymptotic in N . Therefore, we need to choose N sufficiently large for these properties to hold. Dropping this condition would lead to the same (generally intractable) formulations obtained in the actively adaptive learning algorithms (Pronzato et al., 1996). It is hard to quantify the effect of the interval length N on the overall performance in analytical form. Simulation results with different values of N will be presented in the Section 4.5.

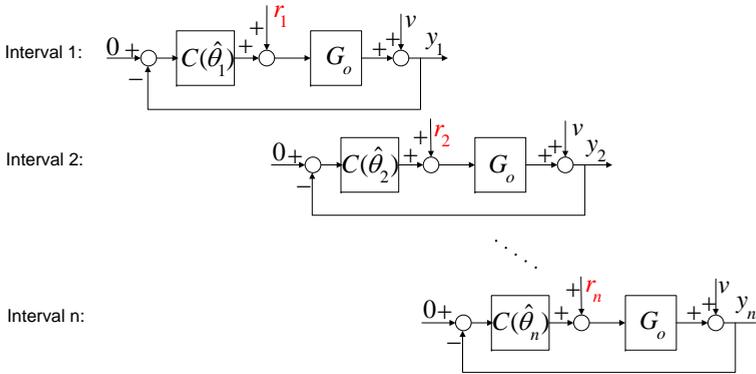


Figure 4.1: The n learning intervals with successive model and controller updates.

4.2.1 Iterative identification

In interval k , the true system is operated with a controller $C(\hat{\theta}_k)$ which has been designed based on an identified model $M_k = M(\hat{\theta}_k)$. As will become clear in the sequel, the identified parameter vector $\hat{\theta}_k$ is normally distributed around θ_o with a covariance matrix P_k , i.e. $\theta_o - \hat{\theta}_k \sim \mathcal{N}(0, P_k)$.

During the N time samples of interval k , an excitation signal r_k is applied to the closed-loop system: $u_k = r_k - C(\hat{\theta}_k)y_k$ (see Figure 4.1). Even though other choices are possible, we will restrict in this chapter to excitation signals r_k which are generated by filtering white noise through a certain FIR filter.

At the end of interval k , the data set $Z_k = \{u_k(t), y_k(t) \mid t = 1, \dots, N\}$ is collected and is used to obtain a more accurate model $M_{k+1} = M(\hat{\theta}_{k+1})$. For this purpose, the parameter vector $\hat{\theta}_{k+1}$ is identified using not only the new data set

Z_k , but also all previous data sets $Z_{k-1}, Z_{k-2}, \dots, Z_1$. Since $\hat{\theta}_k$ has been determined using the previous data sets, this can be done by using $\hat{\theta}_k$ and its covariance matrix P_k in a regularization term. Hence, $\hat{\theta}_{k+1}$ is determined as

$$\hat{\theta}_{k+1} = \arg \min_{\theta \in \mathbb{R}^p} \left(\frac{1}{\sigma_e^2} \sum_{t=1}^N \varepsilon_k^2(t, \theta) + (\theta - \hat{\theta}_k)^\top P_k^{-1} (\theta - \hat{\theta}_k) \right) \quad (4.2)$$

where $\varepsilon_k(t, \theta) = H(q^{-1}, \theta)^{-1} (y_k(t) - G(q^{-1}, \theta) u_k(t))$. Since $\hat{\theta}_{k+1}$ is determined based on kN data and N has been chosen relatively large, it is acceptable to use the asymptotic properties of the prediction error estimate (Ljung, 1999). Consequently, the parameter vector $\hat{\theta}_{k+1}$ identified in this way is (approximately) normally distributed around θ_o with a covariance matrix P_{k+1} given by

$$P_{k+1}^{-1} = I_k + P_k^{-1} \quad (4.3)$$

where $I_k \succeq 0$ is the *information matrix* corresponding to the data Z_k defined as

$$I_k(\theta_o) = \frac{N}{\sigma_e^2} \bar{\mathbb{E}} (\psi_k(t, \theta_o) \psi_k(t, \theta_o)^\top) \quad (4.4)$$

with $\psi_k(t, \theta_o) = - \left. \frac{\partial \varepsilon_k(t, \theta)}{\partial \theta} \right|_{\theta=\theta_o}$ (see Ljung (1999)). Using the recursive nature of (4.3), we have also that

$$P_{k+1}^{-1} = I_k + I_{k-1} + \dots + I_1 + P_1^{-1} \quad (4.5)$$

where P_1 is the covariance matrix of the initial parameter vector $\hat{\theta}_1$.²

Information matrix

The information matrix I_k defined in (4.4) is an affine function on the spectrum $\Phi_k^r(\omega)$ of the excitation signal r_k , and it is in general positive semidefinite, i.e. $I_k \succeq 0$. In this chapter, we will consider in particular excitation which can be obtained by filtering white noise through an FIR filter. The reason of this choice is that for this class of excitation signals the information matrix is guaranteed to be strictly positive definite, i.e. $I_k \succ 0$.

The spectrum $\Phi_k^r(\omega)$ of a FIR-filtered white noise signal $r_k(t)$ can be written as

$$\Phi_k^r(\omega) = R_k(0) + 2 \sum_{j=1}^m R_k(j) \cos(j\omega). \quad (4.6)$$

where $R_k \in \mathbb{R}^{(m+1) \times 1}$ are the autocorrelation coefficients of the signal r_k . For such an excitation signal, the information matrix can be written as (Bombois et al.,

²The covariance matrix P_1 is available for instance if the initial model had been previously identified using the prediction error framework. If that is not the case, and no measure of the accuracy of the initial parameter is available, P_1^{-1} can be set to 0 in (4.5). In this way, the information of the initial will be simply ignored in the computation of the future estimates.

2006)

$$I_k(R_k, \theta_o) = \bar{F}(\theta_o) + \sum_{j=0}^m F_j(\theta_o, \sigma_e^2) R_k(j) \quad (4.7)$$

where the matrices \bar{F} and F_j are nonlinear functions of the $\theta_o, \hat{\theta}_k$ and σ_e^2 . Note that since I_k is affine in R_k , P_{k+1}^{-1} is affine in R_1, R_2, \dots, R_k due to (4.5).

Uncertainty region

Since the updated parameter is estimated with an increasing number of data after each interval, its accuracy will increase for increasing k . In fact, we can define for each estimate $\hat{\theta}_k$ an *uncertainty ellipsoid* \mathcal{D}_k where the modeling error $\theta_o - \hat{\theta}_k$ lies in with a certain probability α as

$$\mathcal{D}_k \triangleq \{\delta \in \mathbb{R}^p \mid \delta^\top P_k^{-1} \delta \leq \chi_p^2(\alpha)\} \quad (4.8)$$

where $\chi_p^2(\alpha)$ is the α -percentile of the chi-squared distribution having p degrees of freedom.

Due to (4.3) and since $I_k > 0$, the volume of the uncertainty regions decreases after each interval. Note that this property always holds if r_k is designed as FIR-filtered white noise and does not require r_k to be designed in an optimal sense. Therefore, even though we will introduce a number of approximations in the experiment design for the determination of these signals, a decrease in the uncertainty after each interval is still guaranteed.

4.2.2 Controller design

Using the data Z_k , we have obtained a more accurate model $M_{k+1} = M(\hat{\theta}_{k+1})$. This new model can be used to design an updated controller $C(\hat{\theta}_{k+1})$. We assume that a control design method has been fixed a-priori and thus the controller is a function $C(\cdot)$ of the parameter vector. The choice for the particular controller design criterion is not important in the development of our framework.

Remark 4.4 *Since the model $M(\hat{\theta}_{k+1})$ is expected to be more accurate than $M(\hat{\theta}_k)$, the controller $C(\hat{\theta}_{k+1})$ is very likely to perform better than the controller $C(\hat{\theta}_k)$ that was in the loop during interval k . However, before applying this new controller to the true system, it is safer to verify whether $C(\hat{\theta}_{k+1})$ stabilizes all the loops $[C(\hat{\theta}_{k+1}) M(\hat{\theta}_{k+1} + \delta)]$ with $\delta \in \mathcal{D}_{k+1}$. A necessary and sufficient condition to perform this robust stability test can be found in Bombois et al. (2001). If the controller $C(\hat{\theta}_{k+1})$ passes this test, we have the guarantee that the loop $[C(\hat{\theta}_{k+1}) S_o]$ is stable (with at least the probability α related to \mathcal{D}_{k+1}) and the controller $C(\hat{\theta}_{k+1})$ can be applied during the interval $k + 1$. In the rare eventuality that robust stability is not validated for the controller $C(\hat{\theta}_{k+1})$, the controller $C(\hat{\theta}_k)$ will be kept for the interval $k + 1$.*

4.2.3 Total cost, modeling error cost, excitation cost

As mentioned in the introduction, one of our objectives is to minimize the overall cost over the n intervals. We evaluate total cost \mathcal{T}_k of interval k considering the difference between the output y_k obtained during this interval, and the output y_o that would have been obtained in an ideal situation, i.e. a situation where we would perfectly know the true system S_o . The output y_k during interval k is given by

$$y_k(t) = \overbrace{\frac{H_o}{1 + C(\hat{\theta}_k)G_o} e(t)}^{=y_{e,k}} + \overbrace{\frac{G_o}{1 + C(\hat{\theta}_k)G_o} r_k(t)}^{=y_{r,k}}, \quad (4.9)$$

while the output y_o of the ideal loop is

$$y_o(t) = \frac{H_o}{1 + C(\theta_o)G_o} e(t). \quad (4.10)$$

The differences between (4.9) and (4.10) are the presence of the term $y_{r,k}$ (which is due to the excitation signal r_k) in (4.9), and the different controllers present in the two loops. Indeed, in the ideal loop the controller $C(\theta_o)$ based on the true parameter θ_o is present, while the controller $C(\hat{\theta}_k)$ actually present in the loop during the interval k is designed based on the parameter $\hat{\theta}_k$.

Total cost

The total cost \mathcal{T}_k for one interval can now be defined as the power of the difference $y_o - y_k$ between these two outputs:

$$\mathcal{T}_k \triangleq \mathbb{E}_{e_k, r_k} [(y_o - y_k)^2] \quad (4.11)$$

where the operator \mathbb{E}_{e_k, r_k} is the *marginal* expectation on the noise source e_k and the excitation signal r_k .

Since r_k and e are independent, we can write the total cost \mathcal{T}_k as

$$\overbrace{\mathbb{E}_{e_k} (y_o - y_k)^2}^{\mathcal{T}_k} = \overbrace{\mathbb{E}_{e_k, r_k} (y_o - y_{e,k})^2}^{\mathcal{V}_k} + \overbrace{\mathbb{E}_{e_k, r_k} (y_{r,k})^2}^{\mathcal{E}_k} \quad (4.12)$$

Modeling error cost

The term \mathcal{V}_k in (4.12) is called *modeling error cost* and represents the performance degradation caused by the use of the controller $C(\hat{\theta}_k)$ instead of the optimal controller C_o . Using the Parseval relation (Ljung, 1999), we can write \mathcal{V}_k as

$$\mathcal{V}_k(\theta_o, \hat{\theta}_k) = \left\| \frac{H(\theta_o)}{1 + C(\theta_o)G(\theta_o)} - \frac{H(\theta_o)}{1 + C(\hat{\theta}_k)G(\theta_o)} \right\|_{\mathcal{H}_2}^2 \sigma_e^2. \quad (4.13)$$

The modeling error cost $\mathcal{V}_k(\theta_o, \hat{\theta}_k)$ is a nonlinear function of θ_o and $\hat{\theta}_k$ having a global minimum in $\theta_o = \hat{\theta}_k$. Indeed, the function takes the value 0 for $\theta_o = \hat{\theta}_k$, while it is equal to or greater than 0 otherwise.

Excitation cost

The term \mathcal{E}_k is called *excitation cost* and represents the performance degradation caused by the introduction of the excitation signal r_k . Using the Parseval relation we can also write \mathcal{E}_k as

$$\mathcal{E}_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \frac{G(e^{i\omega}, \theta_o)}{1 + C(e^{i\omega}, \hat{\theta}_k)G(e^{i\omega}, \theta_o)} \right|^2 \Phi_k^r(\omega) d\omega. \quad (4.14)$$

If we use the parametrization (4.6) for the spectrum Φ_{r_k} , the excitation cost \mathcal{E}_k is linear in the coefficients $R_k(j)$ (Bombois et al., 2006). Thus, it can be written as

$$\mathcal{E}_k(\theta_o, \hat{\theta}_k, R_k) = R_k^\top c(\theta_o, \hat{\theta}_k) \quad (4.15)$$

where $c(\theta_o, \hat{\theta}_k) \in \mathbb{R}^{(m+1) \times 1}$ is a nonlinear vector function of θ_o and $\hat{\theta}_k$ (Bombois et al., 2006).

Worst-case settings

The terms \mathcal{V}_k and \mathcal{E}_k cannot be evaluated since they both depend on the unknown true parameter vector θ_o . However, we can consider these quantities in a worst-case sense by computing their maximum value over the ellipsoid \mathcal{D}_k :

$$\mathcal{V}_k^{\text{wc}} \triangleq \max_{\delta \in \mathcal{D}_k} \mathcal{V}_k(\hat{\theta}_k + \delta, \hat{\theta}_k), \quad (4.16)$$

$$\mathcal{E}_k^{\text{wc}} \triangleq \max_{\delta \in \mathcal{D}_k} \mathcal{E}_k(\hat{\theta}_k + \delta, \hat{\theta}_k, R_k). \quad (4.17)$$

Even though robust design tools (Bombois et al., 2010) could be used to find upper bounds to (4.16) and (4.17), we will here use a simpler approach introduced in (Hjalmarsson, 2009) and based on a second-order Taylor approximation of the functions $\mathcal{V}_k(\cdot, \hat{\theta}_k)$ and $\mathcal{E}_k(\cdot, \hat{\theta}_k, R_k)$ around $\hat{\theta}_k$:

$$\mathcal{V}_k(\theta_o, \hat{\theta}_k) \approx \mathcal{V}_k(\hat{\theta}_k + \delta, \hat{\theta}_k) = \frac{1}{2} \delta^\top V''(\hat{\theta}_k) \delta, \quad (4.18)$$

$$\begin{aligned} \mathcal{E}_k(\theta_o, \hat{\theta}_k, R_k) \approx \mathcal{E}_k(\theta_k + \delta, \hat{\theta}_k, R_k) = \\ R_k^\top c(\hat{\theta}_k, \hat{\theta}_k) + R_k^\top J_c(\hat{\theta}_k) \delta + \frac{1}{2} \delta^\top \left(\sum_{j=0}^m E_j(\hat{\theta}_k) R_k(j) \right) \delta \end{aligned} \quad (4.19)$$

with $\delta = \theta_o - \hat{\theta}_k$. In the last formulas, $V''(\hat{\theta}_k)$ and $E_j(\hat{\theta}_k)$ are the Hessian matrices of $\mathcal{V}_k(\cdot, \hat{\theta}_k)$ and of the j^{th} entry of $c(\cdot, \hat{\theta}_k)$ computed in $\hat{\theta}_k$, respectively:

$$V''(\hat{\theta}_k) = \left. \frac{\partial^2 \mathcal{V}(\vartheta, \hat{\theta}_k)}{\partial^2 \vartheta} \right|_{\vartheta=\hat{\theta}_k}, \quad E_j = \left. \frac{\partial^2 c(\vartheta, \hat{\theta}_k)}{\partial^2 \vartheta} \right|_{\vartheta=\hat{\theta}_k} \quad (4.20)$$

and where $J_c(\hat{\theta}_k)$ is the Jacobian of $c(\cdot, \hat{\theta}_k)$ computed in $\hat{\theta}_k$:

$$J_c = \left. \frac{\partial c(\vartheta, \hat{\theta}_k)}{\partial \vartheta} \right|_{\vartheta=\hat{\theta}_k}. \quad (4.21)$$

Note that the Hessian matrix $V''(\hat{\theta}_k)$ is positive semidefinite since the function $\mathcal{V}_k(\cdot, \hat{\theta}_k)$ has a global minimum in $\hat{\theta}_k$.

Using the second order approximations (4.18) and (4.19), the optimization problems (4.16) and (4.17) can be written as quadratic optimization problems with a single quadratic constraint.

$$\mathcal{V}_k^{\text{wc}} = \max \frac{1}{2} \delta^\top V''(\hat{\theta}_k) \delta \quad \text{such that} \quad \delta^\top P_k^{-1} \delta \leq \chi_p^2(\alpha) \quad (4.22)$$

and

$$\begin{aligned} \mathcal{E}_k^{\text{wc}} = \max & R_k^\top c(\hat{\theta}_k, \hat{\theta}_k) + R_k^\top J_c(\hat{\theta}_k) \delta + \frac{1}{2} \delta^\top \left(\sum_{j=0}^m E_j(\hat{\theta}_k) R_k(j) \right) \delta \quad \text{such that} \\ & \delta^\top P_k^{-1} \delta \leq \chi_p^2(\alpha), \end{aligned} \quad (4.23)$$

respectively.

These problems are not in general convex. However, as proven in (Boyd and Vandenberghe, 2004), *strong duality* holds for this class of problems, i.e. the solution is equal to the solution of the dual problem (which is always convex) (Boyd and Vandenberghe, 2004). Therefore, the dual of problem (4.22) will be used to compute $\mathcal{V}_k^{\text{wc}}$ as

$$\mathcal{V}_k^{\text{wc}} = \min_{\lambda_k} \lambda_k \quad \text{such that} \quad P_k^{-1} \succeq \frac{1}{\lambda_k} \frac{V''(\hat{\theta}_k) \chi_\alpha^2(p)}{2}, \quad (4.24)$$

and the dual of problem (4.23) will be used to compute $\mathcal{E}_k^{\text{wc}}$ as

$$\begin{aligned} \mathcal{E}_k^{\text{wc}} = \min_{\gamma_k, \tau_k} & \gamma_k \quad \text{such that} \\ & \tau_k \geq 0 \\ & \left[\begin{array}{cc} \frac{1}{2} \sum_j E_j(\hat{\theta}_k) R_k(j) - \tau_k \frac{P_k^{-1}}{\chi^2} & \frac{1}{2} (R_k^\top J_c(\hat{\theta}_k))^\top \\ \frac{1}{2} R_k^\top J_c & R_k^\top c(\hat{\theta}_k, \hat{\theta}_k) + \tau_k - \gamma_k \end{array} \right] \preceq 0 \end{aligned} \quad (4.25)$$

Remark 4.5 The second-order approximations (4.18)-(4.19) will become more accurate

when $\hat{\theta}_k$ is closer to θ_o . Consequently, the effects of this approximation will (automatically) decrease for increasing k .

4.3 Experiment Design

Before interval $k = 1$, our objective is to determine the excitation signals in the n learning intervals which will minimize (in a worst-case sense) the overall cost over the n learning intervals, while guaranteeing that the cost of each interval remains below a given threshold. Defining $\mathcal{T}_k^{\text{wc}} \triangleq \mathcal{V}_k^{\text{wc}} + \mathcal{E}_k^{\text{wc}}$, our objective can be mathematically formulated using the following experiment design problem

Problem 4.1 (Experiment Design Problem)

$$R^{\text{opt}} = \arg \min_R \sum_{k=1}^n \mathcal{T}_k^{\text{wc}} \quad \text{such that} \quad (4.26)$$

$$\mathcal{T}_k^{\text{wc}} \leq \bar{\mathcal{T}}_k, \quad \text{for } k = 1, 2, \dots, n. \quad (4.27)$$

where the variable $R \triangleq \{R_1, R_2, \dots, R_n\}$ contains the coefficients parametrizing the excitation spectra for all the learning intervals and $\bar{\mathcal{T}}_k$ is the threshold for the cost during batch k .

The dual effect of the excitation signal is incorporated in the problem formulation (4.26)-(4.27). On the one hand, if the excitation spectrum Φ_{r_k} during the interval k is “large”, the worst-case total cost $\mathcal{T}_k^{\text{wc}}$ will also be large due to the contribution of $\mathcal{E}_k^{\text{wc}}$. On the other hand, this large excitation leads to a “small” covariance matrix P_{k+1} which in turn leads to a small modeling error cost $\mathcal{V}_{k+1}^{\text{wc}}$ for the next interval (see Equation (4.24)).

Further approximations of $\mathcal{V}_k^{\text{wc}}$ and $\mathcal{E}_k^{\text{wc}}$

In order to formulate the experiment design problem as a convex problem, we will use the affine relation existing between the decision variable R and P_k^{-1} and the fact that P_k^{-1} appears linearly in the constraint of (4.24).

In Equation (4.25), it is unfortunately not the case since P_k^{-1} appears in a product with the decision variable τ_k . To convexify the constraint in (4.25), we can redefine $\mathcal{E}_k^{\text{wc}}$ as the worst-case cost over the initial uncertainty ellipsoid \mathcal{D}_1 (instead of \mathcal{D}_k). This is equivalent with replacing P_k by the initial covariance matrix³ P_1 in (4.25). This introduces a conservatism, but $\mathcal{E}_k^{\text{wc}}$ remains an upper bound on \mathcal{E}_k . It is furthermore an acceptable approximation since the actual cost \mathcal{E}_k (unlike \mathcal{V}_k) is not related to the modeling error and is thus not expected to reduce after each interval.

Furthermore, we also have to tackle the so-called “chicken-and-egg” problem, that is a characteristic of most optimal experiment design frameworks (Ljung, 1999). Indeed, the constraints in (4.24) and (4.25) are functions of the identified parameter vectors $\hat{\theta}_k$ that are not available (for $k > 1$) before the first interval i.e. the moment when (4.26)-(4.27) has to be solved. As is generally done, $\hat{\theta}_k$ for all $k > 1$

³If P_1 is not available, the nominal excitation cost $R_k^{\top} c(\hat{\theta}_k, \hat{\theta}_k)$ can be used instead of $\mathcal{E}_k^{\text{wc}}$.

will be replaced by an initial estimate in the optimization problem. This initial estimate will be $\hat{\theta}_1$ in this case. We also use $\hat{\theta}_1$ as an estimate of θ_o in the expression of the information matrix I_k : $I_k(\theta_o, R_k) \approx I_k(\hat{\theta}_1, R_k)$. It is to be noted that, as opposed to other experiment design frameworks, the effects of the approximations introduced to tackle the chicken-and-egg problem (and of the replacement of P_k by P_1 in (4.25)) will be mitigated by the receding horizon mechanism proposed later in this paper (see Section 4.4).

Summarizing, a (further) approximation $\tilde{\mathcal{V}}_k^{\text{wc}}$ of the worst-case modeling error cost $\mathcal{V}_k^{\text{wc}}$ is obtained using (4.24) with $\hat{\theta}_k = \hat{\theta}_1$ and $P_k^{-1} = \tilde{P}_k^{-1}$:

$$\mathcal{V}_k^{\text{wc}} = \min_{\lambda_k} \lambda_k \text{ such that } \tilde{P}_k^{-1} \succeq \frac{1}{\lambda_k} \frac{V''(\hat{\theta}_1) \chi_\alpha^2(p)}{2}, \quad (4.28)$$

where \tilde{P}_k^{-1} is obtained similarly to (4.3), but replacing θ_o with $\hat{\theta}_1$ in the computation of the information matrices I_1, I_2, \dots, I_{k-1} (which is performed according to (4.4)).

A (further) approximation $\tilde{\mathcal{E}}_k^{\text{wc}}$ of the worst-case excitation cost $\mathcal{E}_k^{\text{wc}}$ is obtained using (4.25) with $\hat{\theta}_k = \hat{\theta}_1$ and $P_k = P_1$:

$$\begin{aligned} \tilde{\mathcal{E}}_k^{\text{wc}} &= \min_{\gamma_k, \tau_k} \gamma_k \text{ such that} & (4.29) \\ \tau_k &\geq 0 \\ \begin{bmatrix} \frac{1}{2} \sum_j E_j(\hat{\theta}_1) R_k(j) - \tau_k \frac{P_1^{-1}}{\chi^2} & \frac{1}{2} (R_k^\top J_c(\hat{\theta}_1))^\top \\ \frac{1}{2} R_k^\top J_c & R_k^\top c(\hat{\theta}_k, \hat{\theta}_k) + \tau_k - \gamma_k \end{bmatrix} &\preceq 0 \end{aligned}$$

We are now able to formulate our experiment design problem as a convex optimization problem:

Proposition 4.1 *Let us introduce n scalar variables $\lambda \triangleq \{\lambda_1, \lambda_2, \dots, \lambda_n\}$, n scalar variables $t \triangleq \{t_1, t_2, \dots, t_n\}$, n scalar variables $\gamma \triangleq \{\gamma_1, \gamma_2, \dots, \gamma_n\}$, n scalar variables $\tau \triangleq \{\tau_1, \tau_2, \dots, \tau_n\}$ and n matrix variables $Q \triangleq \{Q_1, Q_2, \dots, Q_n\}$, $Q_k \in \mathbb{R}^{m \times m}$. Using the approximation $\tilde{\mathcal{T}}_k^{\text{wc}} \triangleq \tilde{\mathcal{V}}_k^{\text{wc}} + \tilde{\mathcal{E}}_k^{\text{wc}}$ for the worst-case total cost $\tilde{\mathcal{T}}_k^{\text{wc}}$, the experiment design problem (4.26)-(4.27) is equivalent to the following convex semidefinite optimization problem*

Problem 4.2 (Convex approximation of (4.26)-(4.27))

$$R^{\text{opt}} = \arg \min_{R, \lambda, \gamma, \tau, Q} \sum_{k=1}^n \lambda_k + \sum_{k=1}^n \gamma_k \quad (4.30)$$

such that

$$\lambda_k + \gamma_k \leq \tilde{T}_k \quad (4.31)$$

$$\underbrace{\tilde{P}_{k-1}^{-1}}_{\tilde{P}_k^{-1}} + I_k(\hat{\theta}_1, R_k) \succeq t_k \frac{\chi_\alpha^2(p) V''(\hat{\theta}_1)}{2} \quad (4.32)$$

$$\tau_k \geq 0$$

$$\begin{bmatrix} \frac{1}{2} \sum_j E_j R_k(j) - \tau_k \frac{P_1^{-1}}{\chi^2} & \frac{1}{2} (R_k^\top J_c)^\top \\ \frac{1}{2} R_k^\top J_c & R_k^\top c(\hat{\theta}_1, \hat{\theta}_1) + \tau_k - \gamma_k \end{bmatrix} \succeq 0$$

$$\begin{bmatrix} \lambda_k & 1 \\ 1 & t_k \end{bmatrix} \succeq 0 \quad (4.33)$$

$$\begin{bmatrix} Q_k - A^\top Q_k A & C_k^\top - A^\top Q_k B \\ C_k - B^\top Q_k A & D_k + D_k^\top - B^\top Q_k B \end{bmatrix} \succeq 0 \quad (4.34)$$

for $k = 1, \dots, n$

with $A = \begin{bmatrix} 0 & 0 \\ I_{m-1} & 0 \end{bmatrix}$, $B = [1 \ 0 \ \dots \ 0]$,

$C_k = [R_k(1) \ R_k(2) \ \dots \ R_k(m)]$, and $D_k = \frac{R_k(0)}{2}$.

Proof: The objective function (4.30) is the sum of the worst-case modeling error costs λ_k and the worst-case excitation costs γ_k . The LMIs (4.33) guarantee the conditions $\lambda_k \geq 0$, $t_k \geq 0$ and $t_k \geq \frac{1}{\lambda_k}$ simultaneously. This in turn implies that (4.32) is equivalent with the constraint in (4.28). Finally, (4.34) guarantees that $\Phi_{r_k}(\omega) \geq 0$ (see Bombois et al. (2006)). \square

4.4 Receding horizon

By solving the optimization problem (4.30)-(4.34) before the first interval we can design not only the spectrum $\Phi_{r_1}^{\text{opt}}$ of the signal r_1 that will be applied during this first interval, but also the spectra of the excitation signals r_2, r_3, \dots, r_n for all the following intervals. During the first interval, we generate a signal r_1 having the desired spectrum $\Phi_{r_1}^{\text{opt}}$ and apply this signal as excitation to the loop $[C(\hat{\theta}_1) S_o]$. After the execution of this first interval, the data Z_1 are collected and a new parameter vector $\hat{\theta}_2$ is identified using (4.2). Based on $\hat{\theta}_2$, a new controller $C(\hat{\theta}_2)$ is designed and applied to the true system S_o (after the robust stability check). We could then proceed with interval 2 by applying to the closed loop $[C(\hat{\theta}_2) S_o]$ a signal r_2 having the the spectrum $\Phi_{r_2}^{\text{opt}}$ obtained from the solution of the previous optimization problem. However, a better approach is to redesign the spectra Φ_{r_k} for $k = 2, 3, \dots, n$ using the newly identified parameter vector $\hat{\theta}_2$. Indeed, this parameter vector is a more accurate estimate of θ_o than the initial estimate $\hat{\theta}_1$ since it has been estimated with twice as much data (and consequently $P_2 < P_1$). Consequently, evaluating $\mathcal{E}_k^{\text{wc}}$ at interval $k \geq 2$ as the worst case of \mathcal{E}_k over the new uncertainty ellipsoid \mathcal{D}_2 is less conservative than doing it over \mathcal{D}_1 . Moreover, replacing θ_o and $\hat{\theta}_k$ (for $k > 2$) by $\hat{\theta}_2$ instead of $\hat{\theta}_1$ is also more appropriate in order to tackle the chicken and egg problem since $\hat{\theta}_2$ will be generally closer to θ_o and $\hat{\theta}_k$ (for $k > 2$) than $\hat{\theta}_1$.

For the reasons above, the spectra Φ_{r_k} , $k = 2, 3, \dots, n$ will be redesigned using a similar optimization problem as the one presented in Section 3, but using the new estimate $\hat{\theta}_2$ and its covariance matrix P_2 in order to evaluate $\mathcal{E}_k^{\text{wc}}$ and to deal with the chicken-and-egg problem. This spectrum redesign procedure, inspired by the receding horizon mechanism in MPC control (Maciejowski and Huzmezan,

1997), will be performed after each interval. Note that a similar receding horizon mechanism for the adaptive solution of an experiment design problem was adopted in (Stigter et al., 2006).

Owing to the spectrum redesign procedure, the effects of the approximations on the obtained spectra will become smaller after each interval and the obtained spectra will become increasingly more effective to achieve the objectives of the experiment design. Note that even though the effect of the approximation may be significant for the very first intervals, our approach will still lead to models having and increasing accuracy. This is indeed guaranteed whatever the excitation signal is, as long as it is chosen (like in our case) as filtered white noise.

Remark 4.6 *It can happen that, from a certain value of k onwards, the to-be-applied excitation r_k is (almost) zero. This is the sign that the optimal performance has been reached and the controller update procedure can be stopped. This situation will occur in the example of the next section.*

4.5 Simulation Study

In this section, the iterative identification/controller design framework is applied in a simulation example. The experiment design procedure developed in Section 4.3 is used to generate the excitation signals for the intervals. We consider the Box-Jenkins (BJ) model structure $\mathcal{M} = \{M(\theta), \theta \in \mathbb{R}^6\}$. A model $M(\theta)$ in this structure has $G(q^{-1}, \theta) = \frac{\theta_1 q^{-1} + \theta_2 q^{-2}}{1 + \theta_5 q^{-1} + \theta_6 q^{-2}}$, $H(q^{-1}, \theta) = \frac{1 + \theta_3 q^{-1}}{1 + \theta_4 q^{-1}}$. The true system $S_o = M(\theta_o)$ is described by

$$\theta_o = [0.8 \ 0 \ 0 \ -0.6 \ 0.985 \ 0.819]^\top$$

and the variance of e is $\sigma_e^2 = 1$. The controllers design function is based on the \mathcal{H}_2 criterion

$$C(\hat{\theta}_k) = \arg \min_K \left\| \left[\begin{array}{cc} \frac{H(\hat{\theta}_k)}{1 + K G(\hat{\theta}_k)} & \frac{\sqrt{\beta} K H(\hat{\theta}_k)}{1 + K G(\hat{\theta}_k)} \end{array} \right]^\top \right\|_{\mathcal{H}_2}^2 \quad (4.35)$$

with $\beta = 0.1$.

The initial (stabilizing) controller $C_1 = C(\hat{\theta}_1)$ is designed based on the initial model $M_1 = M(\hat{\theta}_1)$ where

$$\hat{\theta}_1 = [0.676 \ 0.464 \ 0.099 \ 0.6 \ 1.24 \ 0.858]^\top.$$

The initial covariance matrix P_1 is:

$$P_1 = \begin{bmatrix} 0.044 & -0.022 & 0 & 0 & 0.007 & -0.009 \\ -0.022 & 0.056 & 0 & 0 & 0.008 & 0.003 \\ 0 & 0 & 0.0006 & 0.0004 & 0 & 0 \\ 0 & 0 & 0.0004 & 0.0004 & 0 & 0 \\ 0.007 & 0.008 & 0 & 0 & 0.007 & -0.003 \\ -0.009 & 0.003 & 0 & 0 & -0.003 & 0.005 \end{bmatrix}.$$

The parameter $\hat{\theta}_1$ and covariance matrix P_1 were obtained from a preliminary

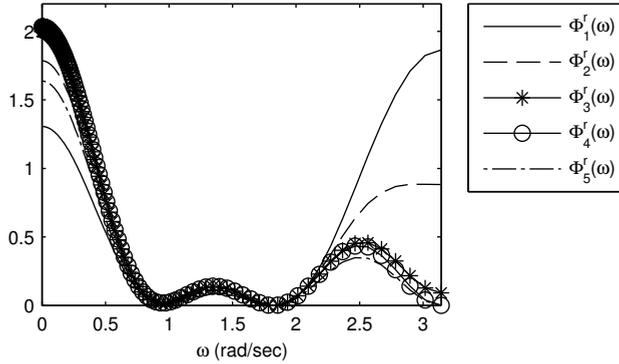


Figure 4.2: Spectra Φ_k^r solution of the first experiment design problem in Case 1

identification based on prediction error identification, using a white noise excitation signal. The worst-case terms are computed with probability $\alpha = 0.99$. The experiment design problems formulated in SDP form are solved numerically using the LMI Lab toolbox (Gahinet et al., 1993).

The true system will be operated in closed loop for a total time corresponding to $N_{\text{tot}} = 2400$ samples. For the first case considered (Case 1), the total time N_{tot} is divided into $n = 12$ intervals having equal length $N = 200$. The constraints on the worst-case total cost are set to $\bar{T}_k = \bar{T}^h = 0.7$ for $k = 1, \dots, 6$ and $\bar{T}_k = \bar{T}^l = 0.005$ for $k = 7, \dots, 12$. Thus, a rather large cost is allowed for the first half of the experiment. This leaves the possibility to introduce a significant level of excitation in the corresponding intervals, in order to satisfy the tighter performance constraint defined for the second half of the experimental time.

A first experiment design problem based on the initial model $M(\hat{\theta}_1)$ is performed before the execution of interval 1 and the optimal sequence of excitation spectra $\{\Phi_{r_1}, \Phi_{r_2}, \dots, \Phi_{r_n}\}$ is found. The first five spectra of this sequence are reported in Figure 4.2. The following spectra are zero up to numerical precision. An excitation signal r_1 with spectrum Φ_{r_1} is generated and it is applied to the system during interval 1.

After the execution of interval 1, the data Z_1 are collected and used to estimate the parameter $\hat{\theta}_2$ and its covariance P_2 . Following, the controller $C_2 = C(\hat{\theta}_2)$ is also designed, and the robust stability of the uncertain closed loop system $[C_2 M(\hat{\theta}_2 + \delta)]$ with $\delta \in \mathcal{D}_2$ is verified using the robust stability tools (Bombois et al., 2001). Subsequently, a new experiment design problem involving the remaining intervals is formulated and solved. The result is a new sequence of excitation spectra $\{\Phi_{r_2}, \Phi_{r_3}, \dots, \Phi_{r_n}\}$. The first element of this new optimal sequence is used to realize the excitation signal r_2 implemented in the interval 2 and the procedure is iterated for all the following intervals.

The spectra of the excitation signals actually fed to the system in the receding horizon implementation are reported in Figure 4.3. Note that these spectra are significantly different from the ones computed before the first interval (Figure 4.2). Furthermore, only the spectra relative to the first 3 intervals are non-zero. Owing

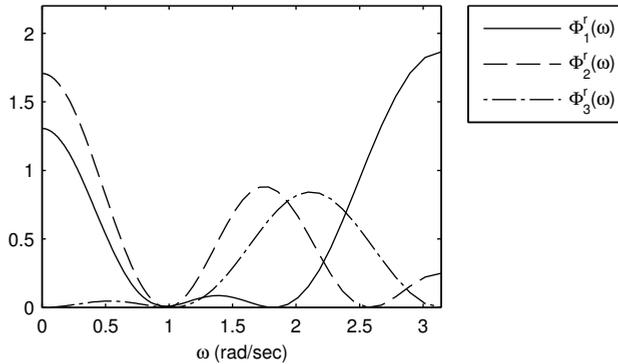


Figure 4.3: Spectra Φ_k^r of the excitation signals actually applied to the system in Case 1.

to the receding horizon mechanism, the algorithm detects that the excitation in the intervals 4 and 5 is no longer required. By removing the excitation in these intervals, the overall performance improves significantly.

In the top left plot of Figure 4.4 the *experimental total cost* \mathcal{T}_k^e is reported. \mathcal{T}_k^e is the sample-based approximation of total cost \mathcal{T}_k and is defined as

$$\mathcal{T}_k^e \triangleq \frac{1}{N_{\text{tot}}} \sum_{t=1}^{N_{\text{tot}}} (y_o(t) - y_k(t))^2.$$

In the same plot, the constraint $\bar{\mathcal{T}}_k$ and the worst-case total cost $\mathcal{T}_k^{\text{wc,RH}}$ are shown. The latter is the worst-case total cost relative to interval k computed during the experimental design performed at the start of interval k .⁴ $\mathcal{T}_k^{\text{wc,RH}}$ reaches the constraint $\bar{\mathcal{T}}_k$ for the first three intervals and decreases in the following ones, so that it can satisfy the new level of the constraint for $k = 7, \dots, 12$. The experimental total cost \mathcal{T}_k^e is always below $\bar{\mathcal{T}}_k$ as expected. \mathcal{T}_k^e increases from interval 1 to 2, since more excitation is applied in the second interval. In the following intervals, \mathcal{T}_k^e decreases and is close to zero for $k \geq 4$. In Figure 4.5 the Bode diagrams of G_o and of the identified models $G(\hat{\theta}_1)$, $G(\hat{\theta}_2)$, $G(\hat{\theta}_3)$, and $G(\hat{\theta}_4)$ are reported. For the identified models, the 99% uncertainty region is indicated by the colored area. The improvement of the models and the reduction of their uncertainty regions over the first four learning intervals is evident in this plot.

From the result of this simulation, it appears that our experiment design procedure allows us to satisfy performance constraints and to optimize the overall performance (for a fixed choice of the interval length) in the iterative identification/controller design framework. The advantage of performing the design of the excitation signals in receding horizon over the intervals is highlighted by the fact that the spectra actually implemented using the receding horizon mechanism

⁴ Note that since the experiment design is repeated in receding horizon, the worst-case for interval k computed in the experiment design k may be different from the worst-case computed in the previous experiment design steps.

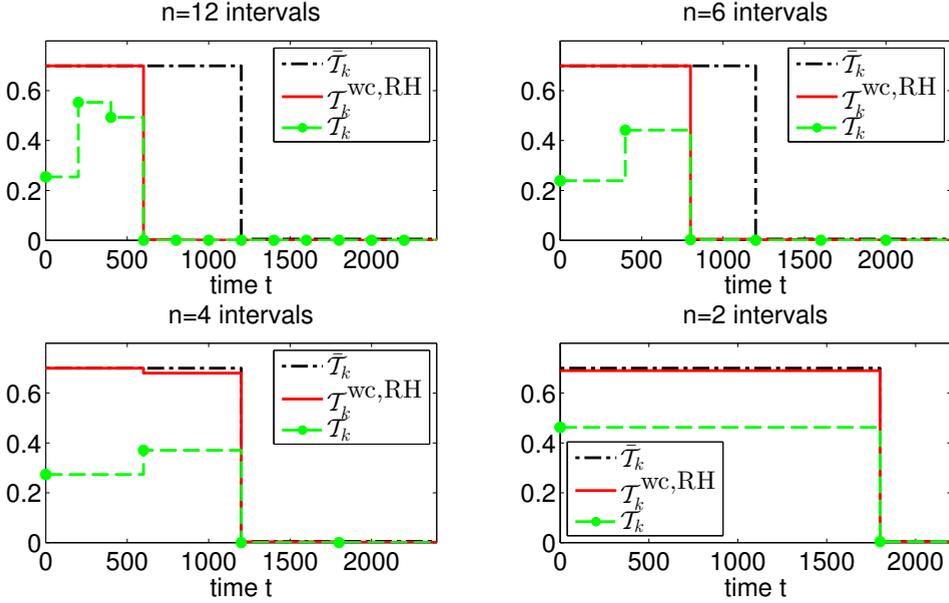


Figure 4.4: Total cost \mathcal{T}_k^e , worst-case total cost $\mathcal{T}_k^{\text{wc,RH}}$, and constraint $\bar{\mathcal{T}}_k$ vs. time t in Case 1 ($n = 12$), Case 2 ($n = 6$), Case 3 ($n = 4$) and Case 4 ($n = 2$) intervals. The green circles denote the time instants corresponding to the start of a learning interval.

are significantly different from the ones that are obtained as solution of the first experiment design problem.

As mentioned in Remark 4.3, the interval length may have a significant impact on the performance that can be achieved in the iterative identification/controller design framework. In general, a better performance is expected using a larger number of shorter intervals, as long as these intervals are sufficiently long for the asymptotic assumptions to be (approximately) satisfied. Unfortunately, we do not have to date the instruments to perform a formal and quantitative analysis clarifying the exact relation between the the interval length and performance. Nonetheless, we investigated this relation in this simulation study where the iterative identification/controller design framework is applied for different choices of the interval length.

We applied the same framework to the same system in three more cases dividing the total time $N_{\text{tot}} = 2400$ into $n = 6$ intervals having equal length $N = 400$ (Case 2), into $n = 4$ intervals having equal length $N = 600$ (Case 3), and into $n = 2$ intervals having lengths $N_1 = 1800$, $N_2 = 600$, respectively (Case 4). The constraints $\bar{\mathcal{T}}_k$ are set to $\bar{\mathcal{T}}^h = 0.7$ in the first $n/2$ intervals and to $\bar{\mathcal{T}}^l = 0.05$ for the last $n/2$ intervals. All the other settings are kept the same as in Case 1. The experimental total cost \mathcal{T}_k^e , the worst-case $\mathcal{T}_k^{\text{wc,RH}}$ and the constraint $\bar{\mathcal{T}}_k$ for all the cases are reported in Figure 4.4. It appears that the average over time of both \mathcal{T}_k^e and $\mathcal{T}_k^{\text{wc,RH}}$ is lower when a larger number of shorter intervals is selected. The

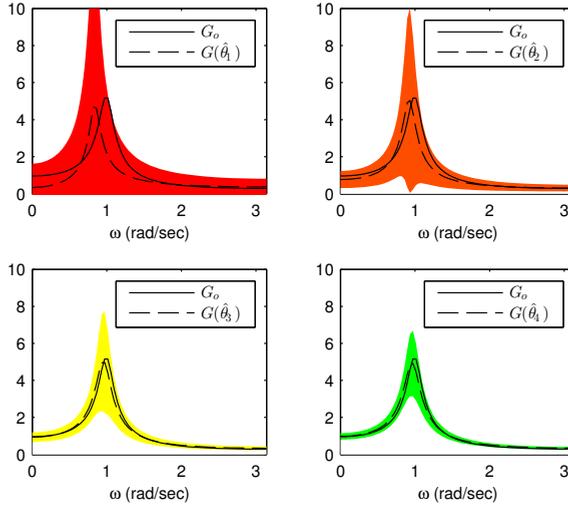


Figure 4.5: Bode plot of G_o , $G(\hat{\theta}_1)$, $G(\hat{\theta}_2)$, $G(\hat{\theta}_3)$, $G(\hat{\theta}_4)$. The colored areas represent the 99% uncertainty regions.

best performance is obtained in Case 1, followed by the Cases 2, 3, 4. Note that Case 4, where only $n = 2$ intervals are considered, corresponds to a classical experiment design problem (Bombois et al., 2006; Gevers and Ljung, 1986) made up of an identification phase (interval 1) and a control phase (interval 2). Indeed, the excitation in the last interval is always zero in our framework. In this case, the constraint $\bar{\mathcal{T}}^h = 0.7$ is kept for a longer time than in the other cases since the length of the first interval is chosen as $N_1 = 1800$. This was done because it was not possible to satisfy the constraint $\mathcal{T}_2^{\text{wc}} \leq 0.005$ without violating the constraint $\mathcal{T}_1^{\text{wc}} \leq 0.7$ having two intervals of equal length $N = 1200$ samples.

These results suggest that the iterative identification/controller design framework can lead to a superior performance compared to the classic two-phase frameworks documented in the literature.

4.6 Conclusions

In this chapter, we have presented a new procedure for the gradual update of a model-based controller. Our approach combines the objective of classic and least costly identification and is based on an iterative identification/controller design framework where the model and the controller are gradually improved during the whole time of the operation. Such an approach allows us to maximize a measure of the overall performance, while guaranteeing at all time a minimum performance level.

The applicability of our procedure has been verified in a simulation study. The simulation study also shows that the interval length has a significant impact on the performance that our procedure can deliver. In particular, for a fixed time

of operation, having a large number of short intervals leads to a superior performance compared to a smaller number of long intervals. Therefore, the iterative identification/controller design framework has the potential of delivering a better performance compared to the classic two-phases experiment design frameworks.

However, the intervals also need to be chosen long enough for the asymptotic properties of the estimated models (which are used in the experiment design) to hold. Thus, there is a lower bound on the interval length. In order to overcome this limitation, system identification and experiment design tools which guarantee uncertainty bounds on *finite* data sets need to be developed. This is a very tough challenge, that we did not tackle in this thesis. Some contributions in this direction are given in Bombois et al. (2008); Dalai et al. (2007); den Dekker et al. (2008).

Another limitation of our framework is the restriction to linear dynamical systems. As already mentioned in Chapters 1 and 3, the main limitation for the application of similar techniques to general nonlinear systems is the lack of efficient Experiment Design tools which can handle these systems. In this direction, preliminary results of experiment design for nonlinear dynamical systems have been developed in this thesis, and are presented in Chapter 5.

Experiment design for parameter estimation in nonlinear systems based on multilevel excitation

The accuracy of a model whose parameters are estimated from measured data depend on the experimental conditions under which these data are collected. A careful choice of these experimental conditions can lead to a significant increase in the accuracy of the parameter estimates, and consequently in the performance of a controller based on the identified model. While for linear dynamical systems generally applicable, computationally efficient experiment design tools for the determination of the optimal experimental conditions are available, the approaches for nonlinear dynamical systems available in the literature to date have still several shortcomings and limitations. In this chapter, a novel experiment design procedure for nonlinear dynamical systems is presented. The input to the system is designed in such a way that the information content of the data, as measured by a scalar function of the information matrix, is maximized. By restricting the input to a finite number of possible levels, the experiment design is formulated as a convex optimization problem which can be solved efficiently. Our method is discussed in relation with other approaches for nonlinear experiment design that have recently appeared in the literature and is applied to the model of a Continuous Stirred Tank Reactor in a simulation study. The estimation based on the input signal obtained in our procedure is shown to outperform the one based on random binary signals.¹

¹This chapter is based on the results presented in Forgiione et al. (2014c).

5.1 Introduction

In many fields of engineering, we interact with system whose behavior is partially unknown. Prior information is often available, but it is either incomplete or not accurate enough for the desired application. We can perform one or more experiments on the system and collect measurements, which are possibly corrupted by sensor noise or perturbed by other disturbances. In this thesis, this situation has already been encountered in the framework for batch-to-batch supersaturation control presented in Chapter 3, and in the iterative identification/controller design framework presented in Chapter 4.

The general problem considered in the Experiment Design field is to determine conditions such that the data collected from the experiments, together with the prior information available, can be used to construct an accurate model of the system. In this chapter, we consider in particular a situation where a *model structure* which can represent the dynamical system is known (e.g. from first-principles knowledge), but the values of certain numerical *coefficients* determining the “right” model within the structure are uncertain and need to be estimated from the experimental data. The objective of the experiment design in this case is to guarantee that accurate estimates for those parameters can be reconstructed from the data.

A measure of the parameter accuracy can be defined in terms of the so-called *information matrix*. A well-known result is indeed that when a statistically efficient estimation method such as *maximum likelihood* is used, the variance of the estimated parameters is asymptotically equal to the inverse of the information matrix (Van den Bos, 2007).

In the Systems Identification field, the experiment design task has been extensively studied for dynamical systems which are *linear* in the input (Goodwin and Payne, 1977). In this case, the information matrix is an affine function of the *power spectrum* of the input signal. This property has been widely used in the design of the input signal adopting a two-step design procedure.

Firstly, a power spectrum for the excitation signal which is optimal (according to the desired criterion) is determined. Exploiting the affine relation between the information matrix and the spectrum, the latter can often be found as the solution of a convex optimization problem (Bombois et al., 2006; Jansson and Hjalmarsson, 2005). Secondly, an input signal having the desired power spectrum is generated. The signal generation can be performed by filtering white noise (which has a flat power spectrum) through the *shaping filter* relative to the optimal spectrum. This two-step design procedure has also been applied in Chapter 4 of this thesis.

However, linear models are often approximations of more complex, nonlinear phenomena. Therefore, the validity of linear models is usually limited to a certain operational range. In some cases, linear models may not suffice in order to describe the underlying dynamics accurately enough for the desired application and a nonlinear approach may be required. For instance, as seen in the Chapters 2 and 3, a linear model could hardly represent the dynamics of the batch cooling crystallization process with reasonable accuracy for control applications.

Nonetheless, performing experiment design for nonlinear dynamical systems is still an open and challenging research topic. Still, the objective can be formu-

lated in terms of the information matrix as in the linear case. However, in the nonlinear case the power spectrum is not sufficient for characterizing the information matrix completely, i.e. input signals having the same power spectrum can lead to different information matrices. Therefore, the two-step design procedure based on the power spectrum cannot be applied.

A possibility is to design the entire probability density function of the input signal.² Since the probability density function appears linearly in the information matrix, a similar two-step procedure based on the entire probability density function of the input signal could be adopted. However, this procedure is much more involved than the one based on the power spectrum (see Hjalmarsson and Mårtensson (2007); Larsson et al. (2010); Valenzuela et al. (2013)). To date, it has been successfully applied only to nonlinear dynamical systems of very limited complexity such as nonlinear FIR.

An alternative approach is to optimize the input signal directly in the time domain by solving a dynamic optimization problem involving the information matrix (Franceschini and Macchietto, 2008). This approach has been followed for different applications such as a cooling crystallization in Chung et al. (2000) and a semibatch RODTOX process in Stigter et al. (2006). However, a drawback of this approach is that the dynamic optimization problems involving the information matrix are in general very hard to be solved. Typically, the dependency of the information matrix on the input signal is severely non-convex. When the optimization problem is solved using standard gradient-based algorithms, chances are high that the numerical solution will lie in the proximity of a local optimum, which is possibly far away from the global one. Convex relaxation for experiment design problems posed in the time domain have only been developed for model structure representing linear dynamical systems (see Larsson et al. (2013); Manchester (2010)).

In this chapter, we present an experiment design procedure which can be applied to a fairly large class of nonlinear systems, but still relies on convex optimization. More precisely, the method can be applied to the class of *fading memory* nonlinear systems, in the sense defined in Boyd and Chua (1985). Loosely speaking, this means that the output of the system mostly depends on the values of the input in the recent past, while the influence of the input in the remote past (and the one of the initial condition of the system) gradually fades out. In the field of process engineering, continuous processes generally satisfy the fading memory property. However, batch processes may not satisfy this condition. Unfortunately, the fading memory property does not hold for the batch cooling crystallization process, where the initial condition of the system has a significant influence on the future behavior of the system up to the final time. For this reason, in the numerical example we will apply our framework to a different process engineering system, namely a Continuous Stirred Tank Reactor (CSTR) where a first-order, irreversible reaction takes place. For the CSTR system, the fading memory property holds.

Our procedure can be seen as a deterministic version of the classical *approximate discrete design* (Fedorov and Hackl, 1996; Pronzato, 2008) here extended to

²Note that the power spectrum (which is sufficient to characterize the information matrix for linear dynamical system) only describes the second order statistical properties of the input signal.

the case of dynamical systems. We restrict the range of the input signal to a finite number of possible *levels* and divide the time of the experiment in a number N of consecutive intervals. During each of the intervals, we keep the input signal constant at one of the levels. This *piecewise constant* input signal can be described by the sequence of N levels that are encountered. We call this sequence the *input sequence*. In the input sequence, we can recognize $N - m + 1$ shorter *subsequences* of length m , partially overlapping each other in the input sequence as shown in Figure 5.1.

We show that if the fading memory of the system is shorter than m intervals, the information matrix for the entire experiment is a linear function of the *relative frequency* of occurrence of each possible pattern of length m (i.e. each possible subsequence) in the full input sequence. Owing to linearity, we will be able to formulate the experiment design problem optimizing a convex measure of the information matrix using the frequencies as design variables. After solving the problem, we will generate an input sequence in which the subsequences appear in numbers proportional to the relative optimal frequencies obtained as solution of the convex optimization problem.

It has to be mentioned that the use of multilevel signals has been investigated for a long time in the field of experiment design. The goal usually considered is to generate multilevel signals whose power spectrum is close to an arbitrary target (see Rojas et al. (2007)). The objective is to replace the input generation approach based on filtered white noise in the two-step experiment design procedure for linear systems discussed above. In fact, even though it is straightforward to generate an input signal having an arbitrary power spectrum by filtering white noise, a multilevel realization is often more attractive since multilevel signals can satisfy *by design* input amplitude constraints, which are common in many real-life applications.

Multilevel signals have also been used in Wong et al. (2013) in order to obtain the *best linear approximation* of a nonlinear system, defined in response to a white Gaussian input signal, using a multilevel signal as input for the identification. Note that the best linear approximation depends on the probability distribution of the input and for this reason a bias has to be expected when a multilevel signal (instead of a white Gaussian signal) is used as input for the identification. In Wong et al. (2013), a moment matching technique was used to obtain a multilevel sequence whose probability distribution approximates the one of the white Gaussian noise as close as possible. It was shown that the number of moments of the Gaussian distribution that is possible to match using a multilevel signal grows linearly with the number of levels in the multilevel signal. Therefore, the bias between the best linear approximation defined for a white Gaussian input signal and the one identified using the multilevel input signal can be arbitrarily reduced by increasing the number of values allowed in the multilevel input signal.

Compared to the contributions above, we are here using multilevel input sequences for the estimation of the parameters of a nonlinear system in a full order model structure. Our approach is closely related to the *probabilistic input design* first proposed in Larsson et al. (2010) for nonlinear FIR systems. In that contribution, a convex measure of the expected value of the information matrix is optimized over the probability of the occurrence of the different subsequences within

the input sequence. Compared to (Larsson et al., 2010), we adapt the framework in such a way that it can be applied to the larger class of nonlinear *fading memory* systems. Moreover, we here follow a deterministic procedure for the generation of the input signal. For this reason, even though the experiment design optimization problem solved in Larsson et al. (2010) and the one solved here are formally equivalent, the probability of the subsequence is here interpreted as the frequency of its occurrence in the input sequence.

Another similar experiment design method based on a multilevel input signal has been recently derived in De Cock et al. (2013). Interestingly, the work in De Cock et al. (2013) was developed as a generalization of a previous multisine experiment design method for linear systems described in Schoukens and Pintelon (1991). The approach is described in De Cock et al. (2013) for the particular case of Wiener systems consisting of a linear FIR filter followed by a static polynomial nonlinearity. Compared to De Cock et al. (2013) (beside the more general framework allowing nonlinear fading memory systems), we formulate the experiment design as a convex optimization problem (as was also done in Larsson et al. (2010)) which can be solved efficiently using standard software and algorithms without introducing further approximations.

5.2 The Framework

5.2.1 Data-generating system and model structure

We assume that a *model structure* which can describe the *data-generating system* is given in state-space ODE representation

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t), \theta) \\ y(t) &= g(x(t), \theta)\end{aligned}\tag{5.1}$$

where $x(t)$ is the state, $u(t)$ is the input, $y(t)$ is the output and $\theta \in \mathbb{R}^p$ is the uncertain parameter vector. The initial state $x_0 = x(0)$ is fixed and known. We consider here the SISO and SIMO cases, i.e. $u(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}^q$, $q \geq 1$.³

We assume that there exists one (and only one) *true parameter* θ_o such that the output of the model constructed using θ_o is equal to the output of the data generating system for every possible input signal.

Measurements \tilde{y}_k of the output y are collected at a constant rate t_s and are corrupted by an additive white Gaussian noise source e_k having mean 0 and known covariance Σ_e : $\tilde{y}_k = y(kt_s) + e_k$ with $e_k \sim \mathcal{N}(0, \Sigma_e)$. We assume that the data-generating system has the *fading memory* property, in the sense defined in Boyd and Chua (1985). Loosely speaking, this means that the output of the system mostly depends on the values of the input in the recent past, while the influence of the input in the remote past (and the one of the initial condition of the system) gradually fades out.

³The extension to the MIMO case is possible, but would add notational complexity.

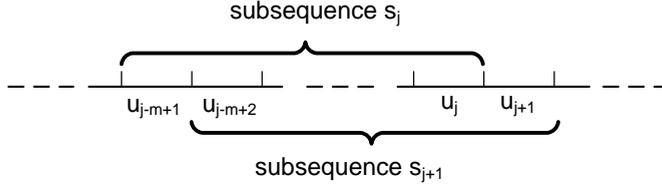


Figure 5.1: Two consecutive subsequences s_j and s_{j+1} share $m - 1$ elements.

5.2.2 Input signal

We restrict the range of the input signal $u(t)$ to a finite number of possible levels $\alpha = \{\alpha_0, \alpha_1, \dots, \alpha_{\ell-1}\}$. The time of the experiment is divided into a number of consecutive intervals I_j , $j = 0, 1, \dots, N - 1$. Each interval has a fixed duration t_I which is a multiple of the sampling time t_s : $t_I = nt_s$ where n is a positive, integer number. The input signal is kept constant to one of the levels in α during each interval.

The input signal during the experiment can be described by an input sequence of N levels $S = \{u_0, u_1, \dots, u_{N-1}\}$, $u_j \in \alpha$. Each element u_j represents the value of the input signal during the interval I_j . The output $y(t)$ of the system is in general a function of the input sequence S , the initial condition x_0 , and the true parameter θ_o : $y(t) = y(t; S, x_0, \theta_o)$.

Since the system has fading memory, the output during a certain interval I_j mostly depends on the input during the previous m intervals $I_{j-m+1}, I_{j-m+2}, \dots, I_j$. For this reason, we recognize in the input sequence S a number $N - m + 1$ of shorter subsequences $s_j = \{u_{j-m+1}, u_{j-m+2}, \dots, u_j\}$ having length m . Note that neighboring subsequences overlap each other. In fact, two consecutive subsequences s_j and s_{j+1} share $m - 1$ elements (see Figure 5.1).

The number m of elements in a subsequence is chosen large enough to describe the transient of the system with good accuracy. Thus, we can approximate the output of the system during interval I_j as a function of input values contained in the subsequence s_j , neglecting the influence of the input values outside s_j and of the initial condition⁴, i.e. $y(t) \approx y(t; s_j, \theta_o)$ for $t \in I_j$ and $j \geq m - 1$.

5.2.3 Subsequence patterns

From a combinatorial consideration, it is easy to see that the total number of possible patterns of length m which can take ℓ distinct values is $M = \ell^m$. For instance, in the case of binary subsequences (i.e. $\alpha = \{\alpha_0, \alpha_1\}$) and with $m = 2$, there are $2^2 = 4$ possible *subsequence patterns*, namely $\{\{\alpha_0, \alpha_0\}, \{\alpha_0, \alpha_1\}, \{\alpha_1, \alpha_0\}, \{\alpha_1, \alpha_1\}\}$.

For the development and the implementation of the experiment design problem, it will be useful to enumerate all the possible patterns. To this end, it is convenient to put the set of these patterns in a 1:1 relation with the set of integer numbers ranging from 0 to $\ell^m - 1$. We shall denote as $s[h]$ the subsequence pattern which corresponds to the integer number h according to the following 1:1 relation.

⁴The approximation is not accurate for $j < m - 1$, since the influence of the initial condition cannot be neglected for these intervals.

Definition 5.1 Given an integer number h , we derive its representation in the base ℓ and build the subsequence $s[h]$ by appending the levels corresponding to the digits of this number from left to right. Conversely, given a subsequence $s[h]$, we construct the number h considering the elements of the subsequences as the digits of an integer number written in the base ℓ .

For instance, in the case of binary subsequences with $m = 2$ discussed above, the 4 possible patterns are in 1:1 relation with the integer numbers 0, 1, 2, 3. These integer numbers can be represented in the base 2 as $(00)_2, (01)_2, (10)_2, (11)_2$. We have

$$\begin{aligned} s[0] &= s[(00)_2] = \{\alpha_0, \alpha_0\}, \\ s[1] &= s[(01)_2] = \{\alpha_0, \alpha_1\}, \\ s[2] &= s[(10)_2] = \{\alpha_1, \alpha_0\}, \\ s[3] &= s[(11)_2] = \{\alpha_1, \alpha_1\}. \end{aligned}$$

The notation $(abc)_d$ stands hereafter for the integer number having digits abc when represented in the base d .

5.2.4 Information Matrix

The information matrix F for the entire estimation problem is

$$F = \sum_{j=0}^{N-1} \overbrace{\sum_{i=0}^{n-1} \psi((jn+i)t_s) \Sigma_e^{-1} \psi((jn+i)t_s)^\top}^{F_j}. \quad (5.2)$$

where $\psi(t) = \left. \frac{d}{d\theta} y(t; S, x_0, \theta) \right|_{\theta=\theta_0}$ is the output parameter sensitivity. We see that the information matrix F is the sum over the intervals of the information matrices F_j relative to the data contained in the intervals I_j .

For $j > m - 1$, owing to the fading memory property of the system, the output of the system in the interval I_j mostly depends on the input values during the most recent m intervals. Therefore, F_j is a function of the subsequence $s_j = \{u_{j-m+1}, u_{j-m+2}, \dots, u_j\}$. Note that the information matrices F_l and F_m relative to two different intervals l, m are identical if the subsequences s_l, s_m have the same pattern.

Definition 5.2 For each of the possible subsequence patterns $s[h], h = 1, 2, \dots, M - 1$, we define

$$p([h]) = \frac{\text{number of occurrences of the subsequence pattern } s[h] \text{ in } S}{N - m + 1} \quad (5.3)$$

as the relative frequency of the subsequence pattern $s[h]$ in the input sequence S .

Definition 5.3 For each of the possible subsequence patterns $s[h], h = 1, 2, \dots, M - 1$, we also define $F(s[h])$ as the contribution to the information matrix relative to a subsequence that has pattern $s[h]$.

In order to compute a term $F(s[h])$, we can simply simulate the nonlinear system together with the parameter sensitivities $\psi(t)$ feeding as input a signal which corresponds to the subsequence pattern $s[h]$.⁵ Following, we can compute the information matrix $F(s[h])$ for the last interval of $s[h]$ as

$$F(s[h]) = \sum_{i=(m-1)n}^{mn-1} \psi(it_s)\Sigma_e^{-1}\psi(it_s)^\top. \quad (5.4)$$

Using the definitions above, we can easily show that the information matrix F for the entire estimation problem is proportional to the relative frequency $p(h)$ of the pattern $s([h])$ in the input sequence S times the contributions $F(s[h])$ relative to the pattern $s(h)$.

Proposition 5.1 *Ignoring the contribution from data in the first $m - 1$ intervals, the information matrix F in (5.2) can be written as*

$$F = (N - m + 1) \sum_{h=0}^{M-1} p(s[h])F(s[h]). \quad (5.5)$$

Proof: This proposition is easily proven by grouping in (5.2) the contributions due to the terms which have the same pattern. □

5.3 Experiment Design

5.3.1 Relative frequencies as design variables

For a *given* input sequence S , it is probably more natural to compute the information matrix using (5.2). Nonetheless, (5.5) is useful for the *design* of the input sequence.

We see indeed that the information matrix F is a *linear function* of the relative frequencies $p(s[h])$. Thus, it is convenient to consider $p(s[h])$ as design parameters of the input sequence. Owing to the linear relation, a large class of experiment design problems can be posed as convex optimization problems in the decision variables $p(s[h])$.

However, a number of constraints on the relative frequencies have to be set in order to obtain a solution which can actually be implemented:

1. The relative frequencies $p(s[h])$ need to be nonnegative numbers summing up to 1.
2. An *exact discrete design* (Fedorov and Hackl, 1996; Pronzato, 2008) requires the relative frequencies $p(s[h])$ to be rational numbers in the range $\mathbb{Q}_N = \{\frac{k}{N}, k = 0, 1, \dots, N\}$.

⁵ The parameter sensitivities can be obtained either from finite differences or integrating the *sensitivity equations* (Rabitz et al., 1983) associated with the ODE (5.1).

3. The relative frequencies have to be chosen in such a way that the subsequences can be concatenated in the input sequence S as shown in Figure 5.1.

Addressing point 1) is immediate. It implies M linear inequalities and one linear equality, which can be easily included in a convex optimization problem.

On the contrary, addressing point 2) is not straightforward. Restricting $p(s[h])$ to \mathbb{Q}_N would lead to a hard combinatorial optimization problem. In practice, it is possible to relax (i.e. neglect) this constraint and allow the frequencies $p(s[h])$ to be *real* numbers. In the experiment design literature, this corresponds to the so-called *approximate discrete design* (Fedorov and Hackl, 1996; Pronzato, 2008).

Point 3 is also delicate to address. We have seen that two consecutive subsequences s_j, s_{j+1} in the input sequence S have $m - 1$ elements in common. Let $*$ denote any level $\alpha_h \in \alpha$. If the subsequence s_j is in the form $\{*, u_0, \dots, u_{m-2}\}$, the next subsequence s_{j+1} in S has to be in the form $\{u_0, u_1, \dots, u_{m-2}, *\}$. Similarly, a subsequence in the form $\{u_0, u_1, \dots, u_{m-2}, *\}$ can only appear after by one the form $\{*, u_0, u_1, \dots, u_{m-2}\}$. Clearly, this sets a constraint on the number of subsequences which have these two patterns in the input sequence.

A *necessary* condition for the existence of an input sequence in which the subsequences appear in numbers proportional to the frequencies $p(s[h])$ is that

$$\sum_{h=0}^{\ell-1} p(\{\alpha_h, u_0, u_1, \dots, u_{m-2}\}) = \sum_{h=0}^{\ell-1} p(\{u_0, u_1, \dots, u_{m-2}, \alpha_h\}) \quad (5.6)$$

for all the ℓ^{m-1} possible of subsequences of length $m-1$ $\{u_0, u_1, \dots, u_{m-2}\}, u_j \in \alpha$. It is easy to show that the condition (5.6) is necessary, but not sufficient for the existence of an input sequence satisfying the ordering constraints. For instance, for a binary sequence of length 2, the solution $p(\{\alpha_0, \alpha_0\}) = p(\{\alpha_1, \alpha_1\}) = 0.5$, $p(\{\alpha_1, \alpha_0\}) = p(\{\alpha_0, \alpha_1\}) = 0$ satisfies the constraints (5.6). However, it is not possible to switch from the subsequence $\{\alpha_0, \alpha_0\}$ to $\{\alpha_1, \alpha_1\}$ without introducing at least one subsequence $\{\alpha_0, \alpha_1\}$.

Remark 5.1 *The same issue is present in the approach presented in De Cock et al. (2013) where the same condition (5.6) has been used. A similar issue occurs in the probabilistic approach as in Larsson et al. (2010). The condition (5.6) in the probabilistic approach guarantees the existence of a stationary distribution for the input sequence such that the subsequences have marginal probabilities $p(\{\cdot\})$ (Grillenberger and Krengel, 1976). However, the Markov Chain defining this stationary probability distribution may not be irreducible. Therefore, it may not be possible to generate the subsequences in numbers asymptotically proportional to $p(s[h])$ sampling from a single realization in the Markov Chain (i.e. we may not be in the condition of applying the ergodic theorem (Norris, 1998)). These issues were not discussed in the contributions De Cock et al. (2013); Larsson et al. (2010).*

At the moment, we are not aware of the existence of a general, convex condition which is necessary and sufficient in order to satisfy the ordering constraints. Thus, we only include the necessary condition (5.6) in the experiment design and solve the issues related to transitions similar to the one above by including a (minimum) number of additional elements in the input sequence.

Proposition 5.2 *The set of constraints (5.6) can be written conveniently as*

$$\sum_{h=0}^{\ell-1} p(s[j+h\ell^{m-1}]) = \sum_{h=0}^{\ell-1} p(s[\ell j+h]) \text{ for } j = 0, 1, \dots, \ell^{m-1}-1. \quad (5.7)$$

Proof: The equivalence can be verified exploiting the correspondence between the subsequence $\{u_0, u_1, \dots, u_{m-2}\}$ and the integer number $(u_0u_1 \dots u_{m-3}u_{m-2})_\ell$, i.e. the integer number having digits $u_0u_1 \dots u_{m-3}u_{m-2}$ represented in the base ℓ . \square

5.3.2 Experiment Design Problem

We here consider a D-optimal experiment design problem which aims to maximize the determinant of the information matrix. Other experiment design problems considering a convex measure of the information matrix such as E-optimal, A-optimal and L-optimal (Atkinson and Donev, 1992) could be similarly implemented.

Problem 5.1 *The D-optimal experiment design problem is*

$$\max_{p(s[0]), \dots, p(s[M-1])} \log \det \left(\sum_{h=0}^{M-1} p(s[h])F(s[h]) \right) \text{ subject to} \quad (5.8)$$

$$p(s[h]) \geq 0 \quad \text{for } h = 0, 1, \dots, M-1 \quad (5.9)$$

$$\sum_{h=0}^{M-1} p(s[h]) = 1 \quad (5.10)$$

$$\sum_{h=0}^{\ell-1} p(s[j+h\ell^{m-1}]) = \sum_{h=0}^{\ell-1} p(s[\ell j+h]) \text{ for } j = 0, 1, \dots, \ell^{m-1}-1 \quad (5.11)$$

The logarithm of the determinant of the information matrix (which is a concave function) is maximized (5.8) subject to the constraints that the relative frequencies are positive (5.9) and sum up to 1 (5.10). The necessary ordering constraint (5.11) for the subsequences is also included.

The overall optimization problem (5.8)-(5.11) is convex and can be solved using standard software and algorithms. In this work, we used the optimization modeling software CVX (Grant et al., 2008) with the solver SDPT3 (Toh et al., 1999).

Remark 5.2 *The experiment design framework presented above is computationally attractive when the memory of the system is significantly shorter than the total time of the experiment. In this case, the length of the subsequences m can be chosen significantly shorter than the length of the input sequence M . For systems that do not satisfy the fading memory property, the effect of the initial condition and of the input in the remote does not decay over time. Therefore, we would need to choose m equal to M in this case. However, in this case our approach reduces to a trivial, brute-force optimization of all the possible input sequences. Unfortunately, the fading memory property does not hold for the batch cooling crystallization process, where the initial condition of the system is known*

to have a significant influence on the future behavior of the system up to the final time. For this reason, in the numerical example we will apply our framework to another system from process engineering, namely a continuous stirred tank reactor, for which the fading memory property holds.

5.3.3 Chicken and the egg issue

In order to solve the optimization problem (5.8)-(5.11), we need to compute the terms $F(s[h])$. However, these terms actually depend on the true parameters θ_o , which are in general unknown. This situation, already encountered in Chapter 4, is actually common in most of the experiment design problems (Ljung, 1999) and is known as the “chicken and the egg” issue. A common workaround to this issue is to replace the true parameter θ_o by an initial estimate $\hat{\theta}_{\text{init}}$ in the experiment design problem.

5.3.4 Input Realization

Once the problem (5.8)-(5.11) is solved, an input sequence containing the subsequences in numbers (approximately) proportional to the optimal frequencies has to be generated. The input generation problem for dynamical systems is more complicated than in the static case considered in classical approximate discrete design (Fedorov and Hackl, 1996) due to the ordering constraints that the subsequences have to satisfy. In the numerical example, we will present an ad-hoc input generation for optimal frequencies obtained in that particular case. The development of a general algorithm for the input generation starting from the optimal frequencies is left for future work. Some ideas in this direction are presented in the conclusions of this chapter.

5.3.5 Parameter Estimation

Once the input sequence S is found, the corresponding input signal $u(t)$ is given to the system and the measurements \tilde{y}_k are collected. The parameter is estimated according to a maximum likelihood criterion. As stated in the introduction, this will lead (asymptotically in the number of samples) to a normal parameter estimate whose covariance is equal to the inverse of the information matrix. In our framework, the maximum likelihood estimator is the *weighted least squares estimator*

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^p} \sum_{k=0}^{Nn-1} (\tilde{y}_k - y(kt_s; S, x_0, \theta))^T \Sigma_e^{-1} (\tilde{y}_k - y(kt_s; S, x_0, \theta)). \quad (5.12)$$

The nonlinear optimization problem (5.12) is solved numerically using the *active-set* algorithm embedded in the Matlab function `fmincon`. The derivatives of the objective function with respect to model parameters are computed analytically by integrating the *sensitivity equations* along with the model equations (Rabitz et al., 1983).

5.4 Numerical Example

We consider a CSTR system with jacket cooling in which a first-order, irreversible reaction $A \rightarrow B$ takes place. Details on this system can be found in Chapter 3 of Luyben (2007). An ODE representation of the system is

$$\dot{C}_A = \frac{F}{V_R}(C_{A0} - C_A) - C_A k_0 e^{-\frac{E}{RT_R}} \quad (5.13)$$

$$\dot{T}_R = \frac{F}{V_R}(T_0 - T_R) - \lambda \frac{C_A}{\rho c_p} k_0 e^{-\frac{E}{RT_R}} - \frac{UA_J(T_R - T_J)}{V_R \rho c_p} \quad (5.14)$$

$$\dot{T}_J = \frac{F_J}{V_J}(T_{cin} - T_J) + \frac{UA_J}{V_J \rho_J c_J}(T_R - T_J). \quad (5.15)$$

where C_A (kmol/m³) is the concentration of the reactant A in the reactor, T_R (K) is the temperature inside the reactor and T_J (K) is the temperature of the cooling medium inside the jackets. The input $u = F_J$ (m³/s) is the flow rate of the cooling medium inside the jackets, the output vector is $y = [C_A \ T_R]^\top$ and the state vector is $x = [C_A \ T_R \ T_J]^\top$.

The symbols $\theta = [k_0 \ E \ \lambda \ UA_J]^\top$ represent the uncertain parameters which are to be estimated. The other symbols represent known, fixed coefficients. The numerical values of the true parameters θ_o and of the fixed coefficients are reported in Table 5.1.

Measurements $\tilde{y} = [\tilde{C}_A \ \tilde{T}_R]^\top$ of $y = [C_A \ T_R]^\top$ are taken at a rate $t_s = 10$ min and are corrupted by additive white Gaussian noise terms having variance $\sigma_C^2 = 0.05^2$ and $\sigma_T^2 = 0.1^2$ respectively: $\Sigma_e = \text{diag}(\sigma_C^2, \sigma_T^2)$. Each interval contains $n = 30$ time samples. An input subsequence is formed by $m = 10$ consecutive time intervals. A binary excitation signal is considered: $\alpha = \{\alpha_0, \alpha_1\}$ where $\alpha_0 = 0.6\bar{u}$, $\alpha_1 = 1.4\bar{u}$ and $\bar{u} = 11.26 \cdot 10^{-3}$. Thus, we have $M = 2^{10} = 1024$ subsequence patterns $s[h]$, $h = 0, 1, \dots, 1023$. For all the possible subsequences, the information matrix of the data collected in the last element of the subsequence is computed.

The experiment design problem (5.8)-(5.11) is implemented and solved numerically. In order to avoid the chicken-and-the-egg issue, the experiment design is here based on the true parameters θ_o .⁶

Interestingly, out of the 1024 subsequence patterns $s[h]$, only 5 have strictly positive optimal frequencies $p(s[h])$. These five patterns, together with their optimal frequencies, are the following

$$\begin{aligned} A &= \{\alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0\}, & p(A) &= 0.1 \\ B &= \{\alpha_0, \alpha_1, \alpha_1, \alpha_1, \alpha_0, \alpha_1, \alpha_1, \alpha_1, \alpha_0, \alpha_1\}, & p(B) &= 0.225 \\ C &= \{\alpha_1, \alpha_0, \alpha_1, \alpha_1, \alpha_1, \alpha_0, \alpha_1, \alpha_1, \alpha_1, \alpha_0\}, & p(C) &= 0.225 \\ D &= \{\alpha_1, \alpha_1, \alpha_0, \alpha_1, \alpha_1, \alpha_1, \alpha_0, \alpha_1, \alpha_1, \alpha_1\}, & p(D) &= 0.225 \\ E &= \{\alpha_1, \alpha_1, \alpha_1, \alpha_0, \alpha_1, \alpha_1, \alpha_1, \alpha_0, \alpha_1, \alpha_1\}, & p(E) &= 0.225. \end{aligned}$$

⁶ Of course, this approach would not be feasible in practice since θ_o is always unknown. Nonetheless, the performance that we obtain using θ_o represents an upper bound to the performance that can be achieved adopting a different approximation. Therefore, it is interesting to consider this situation in this preliminary study on nonlinear experiment design.

Name	Description	Value	Units
F	Flow rate of the feed	0.0044	m^3/s
V_R	Reactor volume	101.6	m^3
C_{A0}	Concentration of A in the feed	8.01	kmol/m^3
T_0	Temperature of the feed	294	K
ρ	Density of product stream	801	kg/m^3
c_p	Heat capacity product	3137	$\text{J}/(\text{K kg})$
V_J	Jacket volume	33.75	m^3
T_{cin}	Temperature of cooling medium	294	K
c_J	Coolant density	1000	kg/m^3
ρ_J	Coolant heat capacity	4183	kg/m^3
k_0	Preexponential factor	$20.75 \cdot 10^6$	$1/\text{s}$
E	Activation energy	$69.771 \cdot 10^6$	J/kmol
UA_J	Heat transfer coefficient · jacket area	$8.6163 \cdot 10^4$	W/K
λ	Heat of reaction	$69.771 \cdot 10^6$	J/kmol

Table 5.1: Parameters and fixed coefficients of the CSTR model.

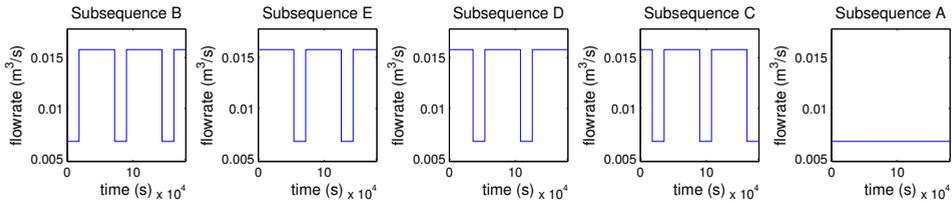


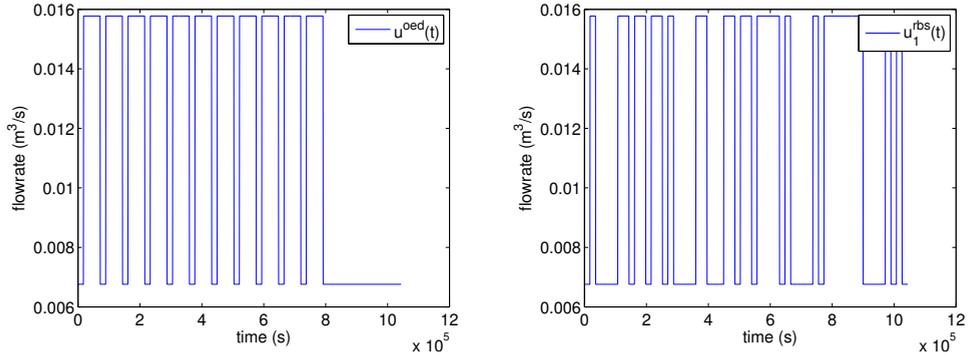
Figure 5.2: The subsequence patterns A, B, C, D, E . The patterns $L_1 = \{B, E, D, C\}$ and $L_2 = \{A\}$ form two distinct loops. The loop L_1 corresponds to a square wave, while L_2 is the constant “low” value.

Note that we have renamed the patterns to A, B, C, D , and E for notational convenience. The patterns are also shown in Figure 5.2.

Given the patterns and the optimal frequencies, an input sequence containing the patterns in numbers (approximately) proportional to their optimal frequencies has to be generated. As mentioned in Section 5.3.4, we did not develop in this thesis a general input generation algorithm. However, we show that for this particular case such an input sequence can be readily obtained.

Indeed, it is easy to verify that the patterns A, B, C, D, E can be concatenated in two separated *cycles*: $L_1 = \{B, E, D, C\}$ and $L_2 = \{A\}$. For instance, the elements from 2 to 10 inside B are equal to the elements from 1 to 9 in E . Therefore, a subsequence having pattern E can be placed after one having pattern B in an input sequence. Examining the patterns, it is evident that the loop L_1 corresponds to a square wave, while L_2 is simply the constant “low” value (see Figure 5.2).

Furthermore, if we could repeat 9 times the cycle L_1 and 4 times the cycle L_2 ,



(a) Optimal input signal $u^{\text{oed}}(t)$ used for all the Monte Carlo runs in Case 1. (b) Random binary signal $u_1^{\text{rbs}}(t)$ used for the first Monte Carlo run in Case 2.

Figure 5.3: Optimal input signal $u^{\text{oed}}(t)$ (left) and realization of a random binary signal $u_1^{\text{rbs}}(t)$ (right).

we would have an input sequence S in the form

$$S = \underbrace{BEDCBEDC \dots BEDC}_{9 \text{ loops } L_1} \underbrace{AAAA}_{4 \text{ loops } L_2}$$

containing $9 * 4 + 4 * 1 = 40$ subsequences. In this sequence, the subsequences would appear in numbers proportional to the desired frequencies. For instance, the pattern A would appear with frequency $4/40 = 0.1$, while the pattern B would appear with frequency $9/40 = 0.225$.

The only problem is that it is not possible to concatenate directly the subsequence having pattern C in the last loop L_1 with the first subsequence A of the first loop L_2 . This is a consequence of the use of the necessary (and not sufficient) ordering constraints (5.6) in the optimization problem. As discussed in Section 5.3.1, we here circumvent this issue including a (minimum) number of transition elements in the input sequence. We build an input sequence that contains 9 cycles of L_1 , 4 cycles L_2 , and a minimum number of transition elements.

Our input sequence starts with a full subsequence with pattern B , i.e. with the elements $\{\alpha_0, \alpha_1, \alpha_1, \alpha_1, \alpha_0, \alpha_1, \alpha_1, \alpha_1, \alpha_0, \alpha_1\}$. Following, the last elements of the patterns E, D, C , i.e. $\{\alpha_1, \alpha_1, \alpha_0\}$, are appended to the input sequence. In this way, 3 the subsequences with pattern E, D, C are concatenated to the input sequence and the first loop L_1 is formed. Next, the last elements of the patterns B, E, D, C , i.e. $\{\alpha_1, \alpha_1, \alpha_1, \alpha_0\}$, are appended to the input sequence 8 more times, forming 8 more loops L_1 . The input sequence defined so far contains 9 loops L_1 and terminates with α_0 , i.e. the last element of a C subsequence. At this point, nine elements α_0 are appended, forming a full subsequence A , which is also a full cycle L_2 . Three more elements α_0 are appended to the sequence, forming 4 cycles L_2 in total. The input signal $u^{\text{oed}}(t)$ corresponding to this sequence consists in square wave followed by a constant part at the value α_0 (Figure 5.3a).

In order to verify the effectiveness of the experiment design procedure, we

perform two Monte Carlo studies where the parameter estimation is repeated $n_{mc} = 100$ times for different realizations of the measurement noise. In the first Monte Carlo study (Case 1), the input signal is $u^{\text{oed}}(t)$ for all the Monte Carlo runs. In the second Monte Carlo study (Case 2), the input $u_k^{\text{rbs}}(t)$ is a random binary signal which can take the same values $\{\alpha_0, \alpha_1\}$ as the optimal input signal and switches its value randomly with clock period t_I , which is the duration of an interval in the optimal input signal. In this case, the input $u_k^{\text{rbs}}(t)$ is stochastic and the realizations are different for all the Monte Carlo runs.

The objective of these Monte Carlo studies is twofold. First, we want to verify whether (and to what extent) the optimal signal leads to a more informative experiment than the random binary signals, as measured by the optimized metric of the information matrix. The random binary signal is a good candidate for the comparison since it is a common, non-optimal engineering choice for the generation informative identification datasets. Furthermore, the particular choice of the random binary signals that we have performed corresponds to a random search in the same space of signals that our optimal procedure can generate.

Second, we want to verify whether the metric of the information matrix that is optimized is actually related to the accuracy of the estimated parameters. For this to be verified, the relation between the information matrix and the inverse of the parameter covariance matrix, which holds under asymptotic assumptions, has to be approximately valid for our finite-time experiment.

The sample-scaled determinant of the information matrices $\frac{1}{Nn} I_k^{\text{oed}}$ and $\frac{1}{Nn} I_k^{\text{rbs}}$ obtained for the different Monte Carlo iterations in the two cases are reported in Figure 5.4. Note that the information matrix in our case only depends on the input and therefore $I_k^{\text{oed}} = I^{\text{oed}}$ is the same for all the realizations in Case 1, while I_k^{rbs} depends on the particular realization of the random binary signal $u_k^{\text{rbs}}(t)$. The scaled determinant $\frac{1}{Nn} \det I^{\text{oed}}$ is $2.2 \cdot 10^{14}$, while the average of the scaled determinants $\frac{1}{n_{mc}} \sum_{k=1}^{n_{mc}} \frac{1}{Nn} \det I_k^{\text{rbs}}$ over the different Monte Carlo iterations is $1.2 \cdot 10^{14}$. Thus, in average, the optimal input is approximately 1.8 times more efficient than the random binary signal.

Under asymptotic assumptions, the covariance matrix of the estimated parameters equals the inverse of the information matrix. Nonetheless, it is important to verify how accurate this relation is for a finite-time data set. For this reason, we also compute the *sample covariance*

$$\hat{\Sigma}^{\text{oed}} = \frac{1}{n_{mc} - 1} \sum_{k=1}^{n_{mc}} (\hat{\theta}_k^{\text{oed}} - \bar{\theta})(\hat{\theta}_k^{\text{oed}} - \bar{\theta})^\top \quad (5.16)$$

where $\bar{\theta} = \frac{1}{n_{mc}} \sum_{k=1}^{n_{mc}} \hat{\theta}_k^{\text{oed}}$ is the *sample mean*. The sample covariance $\hat{\Sigma}^{\text{oed}}$ has

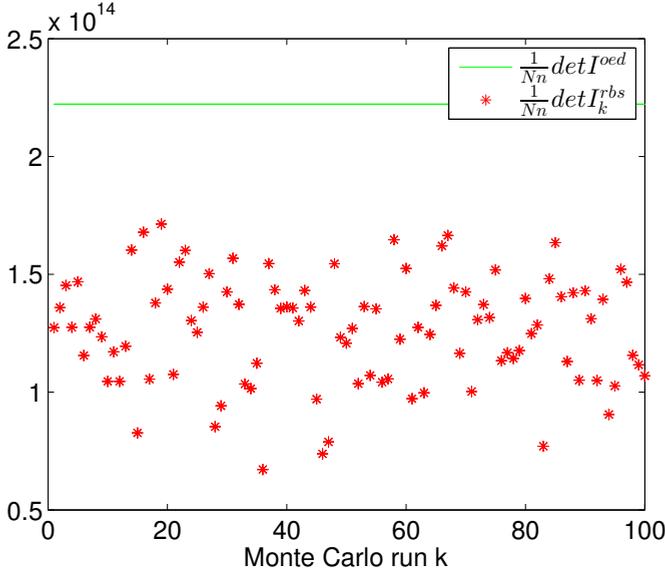


Figure 5.4: Determinant of I_k^{oed} and I_k^{rbs} vs. iteration number k .

to be compared with the *theoretical covariance* $\Sigma_{\text{theo}}^{\text{oed}} \triangleq \text{inv}(I^{\text{oed}})$. We find that

$$\hat{\Sigma}^{\text{oed}} = \begin{bmatrix} 7.2 \cdot 10^{-2} & 9.5 \cdot 10^{-3} & 7.5 \cdot 10^{-4} & -4.2 \cdot 10^{-5} \\ 9.5 \cdot 10^{-3} & 1.3 \cdot 10^{-3} & 1.0 \cdot 10^{-4} & -1.0 \cdot 10^{-5} \\ 7.5 \cdot 10^{-4} & 1.0 \cdot 10^{-4} & 9.1 \cdot 10^{-5} & -4.5 \cdot 10^{-5} \\ -4.2 \cdot 10^{-5} & -1.0 \cdot 10^{-5} & -4.5 \cdot 10^{-5} & 2.6 \cdot 10^{-5} \end{bmatrix},$$

$$\Sigma_{\text{theo}}^{\text{oed}} = \begin{bmatrix} 8.6 \cdot 10^{-2} & 1.1 \cdot 10^{-2} & 5.7 \cdot 10^{-4} & 1.0 \cdot 10^{-4} \\ 1.1 \cdot 10^{-2} & 1.4 \cdot 10^{-3} & 8.0 \cdot 10^{-4} & 1.0 \cdot 10^{-5} \\ 5.7 \cdot 10^{-4} & 8.0 \cdot 10^{-5} & 7.3 \cdot 10^{-5} & -3.5 \cdot 10^{-5} \\ 1.0 \cdot 10^{-4} & 1.0 \cdot 10^{-5} & -3.5 \cdot 10^{-5} & 2.0 \cdot 10^{-5} \end{bmatrix}.$$

The matrices $\hat{\Sigma}^{\text{oed}}$ and $\Sigma_{\text{theo}}^{\text{oed}}$ are reasonably close to each other, e.g. the relative difference $\frac{\|\Sigma_{\text{theo}}^{\text{oed}} - \hat{\Sigma}^{\text{oed}}\|_{\mathcal{F}}}{\|\Sigma^{\text{oed}}\|_{\mathcal{F}}}$ of their Frobenius norm is 0.15.

Finally, a scatter plot for the first two coordinates of the n_{mc} estimated parameters for the two cases is reported in Figure 5.5. The estimates $\hat{\theta}_k^{\text{oed}}$ obtained in Case 1 with the optimal input signal appear to be closer to the true parameter than the estimates $\hat{\theta}_k^{\text{rbs}}$ obtained in Case 2 with the random binary input signal. This confirms that our experiment design approach leads to an increase in the accuracy of the estimated parameters compared to a non-optimal, random binary design.

5.5 Conclusions

We have presented an experiment design method that can be applied to a rather wide class of nonlinear systems, namely to fading memory nonlinear systems.

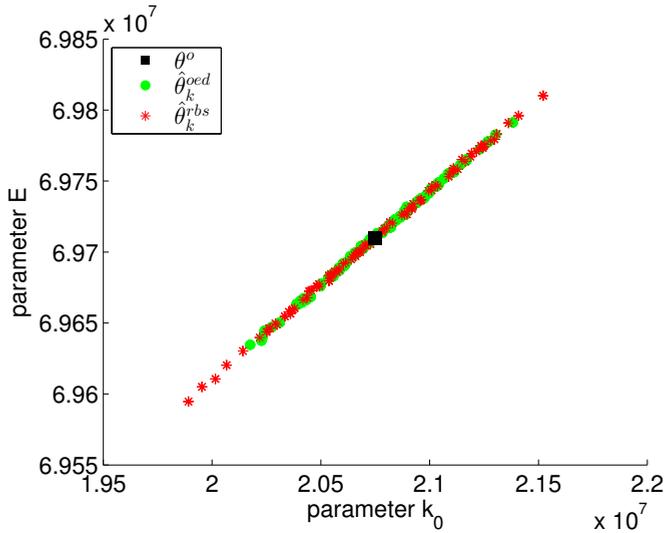


Figure 5.5: Scatter plot for the first two coordinates of θ_o , $\hat{\theta}_k^{\text{oed}}$ and $\hat{\theta}_k^{\text{rbs}}$.

The method has been successfully tested on the model of a first-order, irreversible CSTR system. The signal generated through this method is shown to compare favorably with random binary signals in a simulation case.

The promising results of this preliminary work leave a number of open questions and space for future research. First, it would be useful to develop a general algorithm for the generation of the input sequence starting from the optimal frequencies. In fact, the input generation could be rather involved when several cycles of subsequences are possible and when the transition between the cycles requires appending additional elements to the input sequence. Tools from graph theory could be used to tackle the problem, e.g. to discover all the possible cycles of subsequences in the optimal solution.

Second, the complexity of our method is proportional to the number of possible subsequences, which in turn is equal to the number of levels raised to the power of the number of elements per subsequence. Increasing the number of levels allows one to switch between a larger number of input values. Increasing the number of elements in a subsequence allows one to switch more often between the input values. Both choices increase the degrees of freedom in the design and thus possibly lead to a more effective excitation signal. However, given the limitations on the computational power available, a trade-off between considering more levels or more elements per subsequence has to be found.

Finally, in the numerical example we have observed that out of the many possible subsequences, only a few of them have strictly positive optimal frequencies, and they correspond to two different regimes for the system. This result is very interesting in relation with the experiment design problems involving linear systems, where on the contrary it is known that the optimal excitation signal can be generally chosen as the realization of an ergodic, quasi-stationary stochastic process. It would be interesting to investigate whether this property is really induced

by the nonlinearity of the system, or it is just due to the particular experiment design method that we have devised, which allows this kind of solutions.

Experimental Results

This chapter describes the results of an experimental campaign on batch cooling crystallization of Succinic acid from water. The ultimate goal of the experiments was to test the batch to batch control algorithms previously developed in this thesis. The results of the experiments are very promising. The Root Mean Square of the supersaturation tracking error is shown to decrease owing to the application of the batch to batch control algorithms. Further investigation is required in order to make these control techniques robust enough to be applied in an industrial production environment.¹

6.1 Introduction

This chapter describes the results of an experimental campaign on batch cooling crystallization performed in the ACES department of DSM (Geleen, The Netherlands) within the ISPT project PH-00-04. In this campaign, batch cooling crystallization experiments of Succinic acid in water were performed in a 50-liters glass vessel. The vessel was connected to a skid that was previously developed within the same project (Kadam et al., 2012) in order to collect process measurements from a number of different instruments. The skid consists of two parts: a pump skid and an instrument skid. The slurry is circulated from the crystallizer through the pump skid to the measurement skid, and then back to the crystallizer. On the instrument skid analytical instrument are mounted in order to collect different process measurements.

The objective of the campaign was to test the batch to batch control experiments developed in Chapter 3. In order to achieve this result, the behavior of the water-succinic crystallization process was first characterized. The solubility line and the MSZW for the crystallization system at hand have been estimated performing heat-up and unseeded batch crystallization experiments, respectively. Next, a seeding procedure was determined and seeded batch crystallization experiments were performed.

¹Part of the results presented in this chapter have been used in Forgione et al. (2014a).

The skid appeared to be of great importance both for the process characterization and for the implementation of the B2B control strategies.

The rest of this Chapter is organized as follows. The experimental setup and the software architecture are described in Section 6.2. Next, the design of the temperature controller is presented in Section 6.3. The results of preliminary crystallization experiments using a pure water-succinic system are presented in Section 6.4. Subsequently, the result of crystallization experiments in the presence of Fumaric acid as an additional impurity to the water-succinic system are presented in Section 6.5. The system in the presence of the impurity appears to be more suitable for the testing of the batch-to-batch supersaturation control algorithm. Finally, batch-to-batch supersaturation control experiments are presented in Section 6.5. Overall conclusions on this experimental campaign and on the applicability of batch-to-batch control for cooling crystallization are drawn in Section 6.7. Symbols and units of all the quantities used in this chapter are defined in Section 6.8.

6.2 The experimental setup

A schematic representation of the experimental setup is given in Figure 6.1. The crystallizer used in the experiments is a 50-liters, jacketed glass vessel (Figure 6.2). The crystallizer is agitated by an impeller driven by a DC motor. The temperature in the crystallizer is manipulated by circulating a fluid medium (a solution of water and glycol) in the jackets of the vessel. A thermostatic bath (LAUDA RUK 90 S, Figure 6.3) is used to heat up and cool down the fluid medium. The same thermostatic bath is equipped with a pump that allows the circulation of the fluid in the jackets.

The crystallizer is connected from the bottom valve to the skid through thermally traced, insulated hoses having an internal diameter of 2.54 cm. As previously mentioned, the skid actually consists of two parts called pump skid and measurement skid, respectively.

The slurry is circulated from the bottom valve of the crystallizer to the pump skid, the instrument skid and finally back to the top of the crystallizer with the help of a lobe pump (Omac BF330) which is mounted on the pump skid. Four instruments are mounted on the instrument skid in order to collect different process measurements:

1. The refractive index sensor K-Patents PR-23 [K-Patents].
2. The ATR-FTIR Spectroscope Bruker MATRIX-MF [Bruker].
3. The Ultrasound extinction sensor Sympatec OPUS/G [Opus].
4. A Perdix ISPV Inline Vision Probe [ISPV].

We will often refer to the instruments using the shorthand name reported between square brackets.

The K-Patents and the Bruker instruments are meant to measure concentration, while the Opus and the ISPV provide information about the CSD. Along

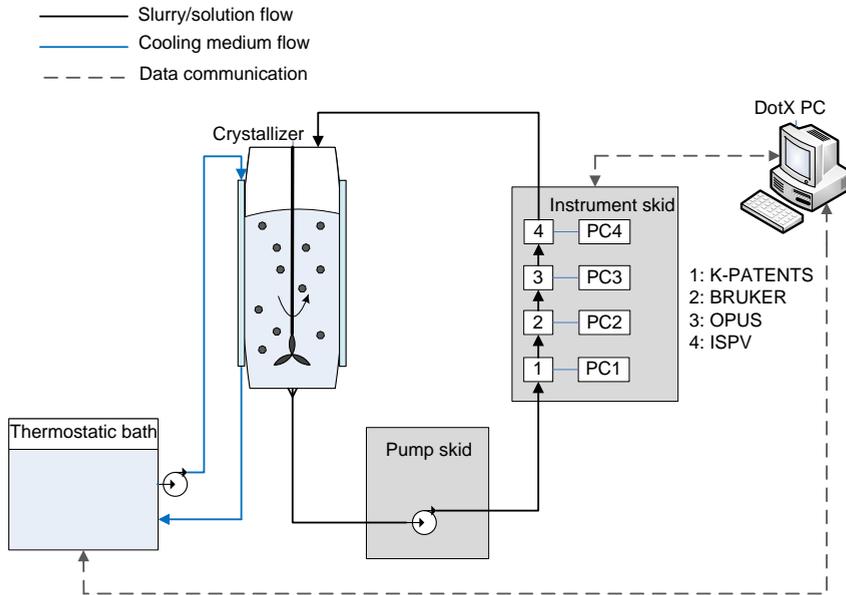


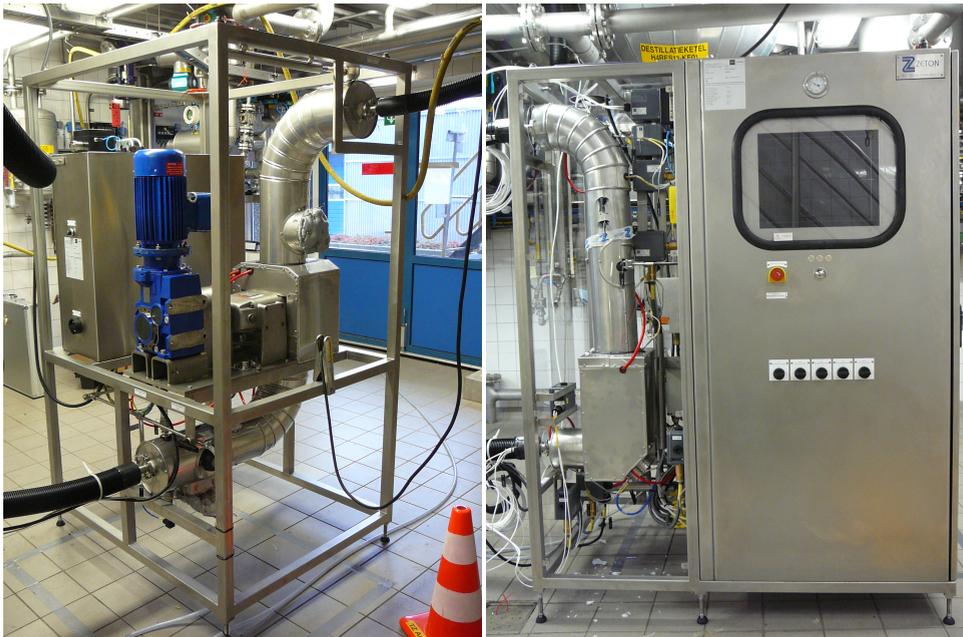
Figure 6.1: A schematic representation of the experimental setup.



Figure 6.2: The 50-liters vessel used as crystallizer for the experiments.



Figure 6.3: The thermostatic bath LAUDA RUK 90 S.



(a) Pump skid.

(b) Measurement skid.

Figure 6.4: Pictures of the pump skid (left) and the measurement skid (right).

the circulation line, it is possible to measure the temperature at different locations with PT100 sensors. A PT100 sensor is also mounted inside the vessel in order to measure the slurry temperature accurately. Each of the instruments is equipped with a dedicated computer. These computers are located in a pressurized cabinet located on the instrument skid. The cabinet also hosts a Eurotherm PLC system which performs a number of low-level control tasks.

The high-level monitoring and control tasks are performed on a separate computer which we will refer to as the DotX PC. The DotX PC is connected with the computers inside the skid and to the thermostatic bath through Ethernet and serial interfaces, respectively. Data communication with the computers in the cabinet is provided by an Ethernet interface positioned on the outside of the cabinet, while data communication with the thermostatic bath is provided by a serial (RS-232) interface.

6.2.1 The instruments

K-Patents PR-23. The K-Patents instrument measures the refractive index of the liquid phase in the slurry. The refractive index depends on both the temperature of the solution and the concentration of a solute dissolved in it. Since the temperature can be measured independently (for instance from the PT100 sensor), the concentration can be reconstructed using refractive index measurements (Rozsa, 2006). In principle, the sensor is not sensitive to the presence of a solid phase in the slurry and for this reason it is suitable for measuring the concentration during a crystallization process. However, the measurements may be affected by crystal formations growing on the instrument itself (fouling). After a fouling event, the measurement is not reliable any more until the solution is heated up and the crystals growing on the sensors are dissolved. In order to reduce the chances of fouling events, the K-Patents instrument was installed at an angle of 45° with respect to the flow of the slurry. This allows for the self-cleaning effect of the prism. Even though it is difficult to prevent fouling completely, this phenomenon can be detected by inspecting a quality measure called *slope* provided by the instrument itself. More details on this instrument can be found in K-Patents (2013).

Bruker MATRIX-MF. The Bruker instrument measures the absorption of mid-infrared radiation of the liquid phase in the range $3000 - 300\text{cm}^{-1}$. From the analysis of the absorption spectra, it is possible to reconstruct the composition and the concentration of the chemicals dissolved in the liquid phase (Kadam et al., 2010). Like the K-Patents, the Bruker is not sensitive to the properties of the solid phase phase. Therefore, it is suitable for measuring the concentration during a crystallization process. However, during the experimental campaign we experienced a large number of technical failures of the instrument that prevented its usage in the experiments reported later.

Sympatec OPUS/G. The Opus instrument measures the absorption of ultrasound waves in the range $1 - 100\text{ Mhz}$ that are passed through a small gap ($4 - 8\text{ mm}$) where the slurry is circulated. The absorption of ultrasound waves depends on the amount and the size of the particles that are present in the solid phase, as well

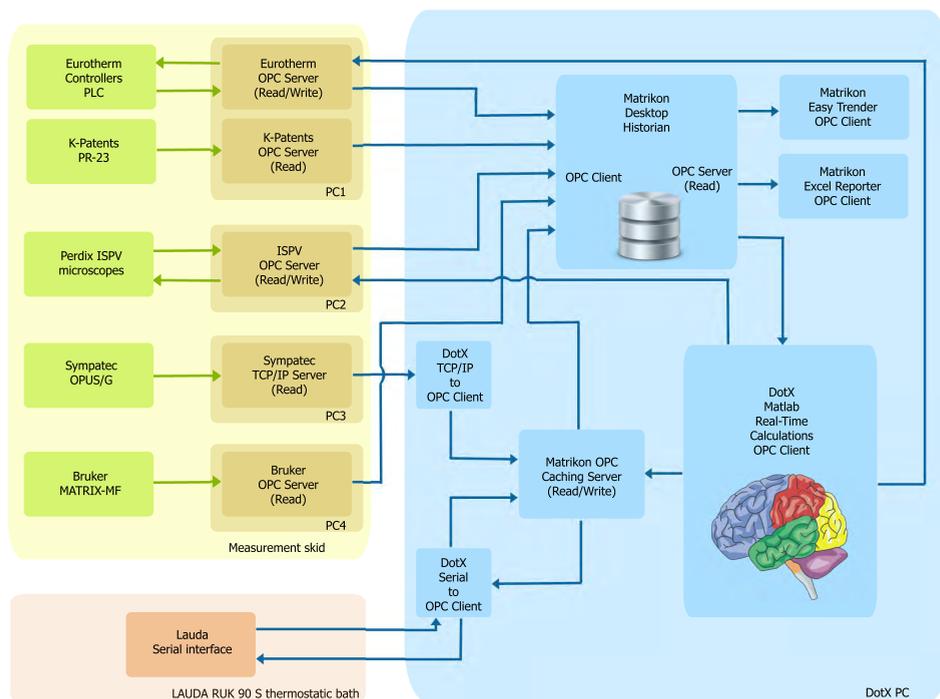


Figure 6.5: Software architecture of the experimental setup.

as on the properties of the liquid phase (Allegra and Hawley, 1972). However, the dependency is rather complex and the reconstruction of the CSD from ultrasound absorption measurements is not straightforward. The OPUS instrument has not been used in this experimental campaign.

Perdix ISPV Inline Vision Probe. The ISPV instrument consists of two optical microscopes with a resolution of $10\mu\text{m}/\text{pixel}$ (low magnification) and $2\mu\text{m}/\text{pixel}$ (high magnification) respectively. In principle, the CSD could be reconstructed using the images collected by the instrument (Li et al., 2008). However, a reliable image analysis software was not available during the experiment. For this reason, the ISPV was only used in a qualitative way to detect the presence of crystals. This allowed us to observe the onset of nucleation in the unseeded batch experiments and to verify that the seeds do not dissolve in the solution in the seeded batch experiments.

6.2.2 Software architecture

A scheme of the software architecture is depicted in Figure 6.5. The design and implementation of the software architecture was performed in cooperation with the company DotX.

The control algorithms are implemented in a Matlab program which runs on

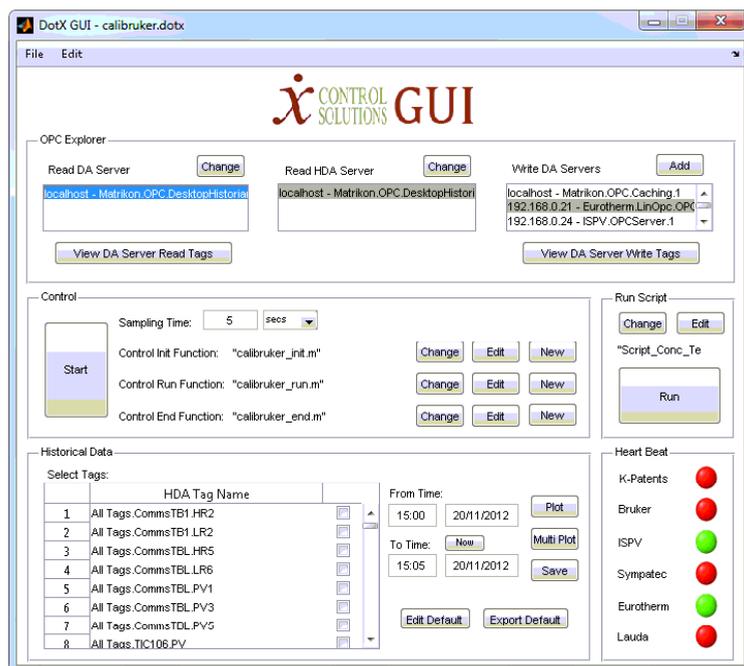


Figure 6.6: The Matlab GUI.

the DotX PC. The Open Platform Communication (OPC) protocol is used to implement read/write communication within the DotX PC, the skid, the instruments and the thermostatic bath. The Eurotherm PLC, the K-Patents, the ISPV and the Bruker are equipped with OPC Servers delivered by the respective manufacturer. For the Opus and the thermostatic bath, two ad-hoc OPC clients were implemented by DotX. In turn, these OPC clients write the data to a Matrikon OPC Caching Server. Therefore, the OPC Caching Server acts as an OPC server for the Opus and the Lauda.

An OPC database (Matrikon OPC Desktop Historian) collects measurements from the different servers at a constant rate and is used to store the data permanently on the DotX PC for off-line analysis.

The Desktop Historian is also used as a read buffer between the Matlab program and the OPC server of the instruments. Whenever the Matlab program needs a measurement (either real-time or historical), it queries the Desktop Historian instead of the OPC server of the instrument directly.

A screenshot of the graphical user interface (GUI) of the Matlab program is depicted in Figure 6.6. The GUI allows the user to check the status of the instruments, plot the data and modify the control algorithm. The latter is implemented as a Matlab script that runs periodically at a constant sampling rate.

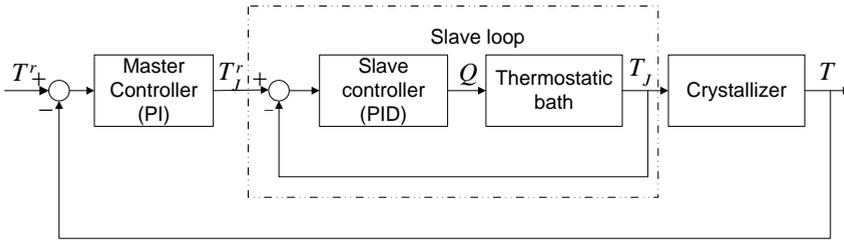


Figure 6.7: The master-slave control scheme for the temperature of the crystallizer.

6.3 Design of the temperature controller

The temperature inside the crystallizer is controlled using the master-slave control configuration that is represented in Figure 6.7.

A slave PID controller regulates the temperature T_J inside the thermostatic bath in order to follow the set-point T^r_j by providing a heat input Q . The signal Q is not directly accessible. The user can only provide the set-point T_J for the PID temperature controller which is directly integrated in the thermostatic bath. The parameters of this controller can also be tuned. However, the default factory tuning was found to be satisfactory and was not changed in our experiments. From a preliminary data analysis, the dynamics of the slave loop were found to be rather fast (in the order of a minute) compared to the ones of the crystallizer (which are in the order of an hour). For this reason, the presence of an internal loop was ignored for the design of the master controller.

From physical considerations, the dynamics from T_J to T is known to be stable and first-order. A first-order model was estimated based on measured data of T_J and T using the Matlab System Identification Toolbox (Ljung, 2007). The identified model is

$$T(s) = F_{TT_J}(s)T_J = \frac{2.925 \cdot 10^{-4}}{s + 2.53 \cdot 10^{-4}}T_J \quad (6.1)$$

where s denotes the Laplace variable. The time constant of the model is $T_{cl} = 39526 \text{ s} = 1.1 \text{ hours}$. Given the model, the following master PI controller was designed

$$\text{PI}(s) = \frac{5.698(s + 2.53 \cdot 10^{-4})}{s}. \quad (6.2)$$

The closed-loop system F_{TT^r} from T^r to T has a linear, first-order behavior² with a time constant $T_{cl} = 600 \text{ s} = 10 \text{ min}$ and static gain exactly equal to one owing to the integral action of the PI controller

$$T(s) = F_{TT^r}(s)T^r = \frac{1}{1 + \frac{s}{600}}T^r. \quad (6.3)$$

²Note that the zero of the PI controller is used to cancel the pole of the open loop system $T = F_{TT_J}(T_J)$.

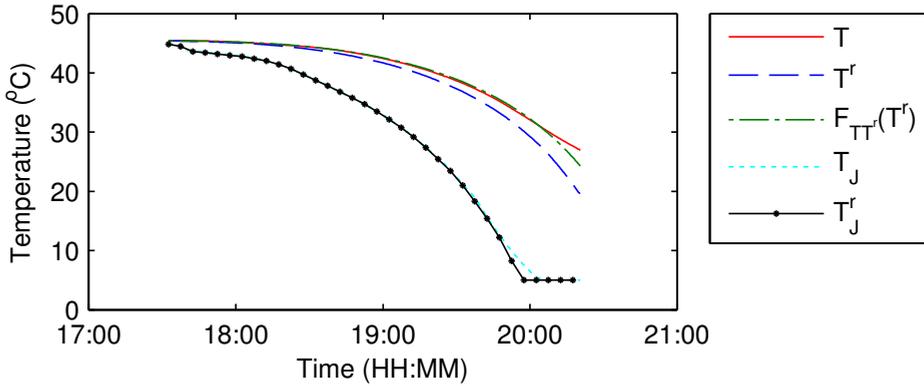


Figure 6.8: The behavior of the closed-loop temperature dynamics.

The actual command T_J^r is generated by a discrete-time version of the controller $\text{PI}(s)$ having sampling time $t_s = 30$ s. Furthermore, the command is saturated in the range $[5 - 90 \text{ }^\circ\text{C}]$ for safety considerations, i.e.

$$T_J^r = \min(\max(\text{PI}(z)(T - T^r), 5 \text{ }^\circ\text{C}), 90 \text{ }^\circ\text{C}). \quad (6.4)$$

where

$$\text{PI}(z) = \frac{5.698(z - 0.9924)}{z - 1}. \quad (6.5)$$

The behavior of the overall closed-loop system during a typical cooling down trajectory for batch crystallization is analyzed in the Figure 6.8. The reference T^r is chosen such that the crystallizer temperature T follows the trajectory $F_{TT^r}(T^r)$. Such trajectory has a parabolic shape and is almost flat at the start of the batch (at high temperature) and gets steeper and steeper towards the end of the batch (at low temperature). From the plot, we see that the jacket temperature T_J follows its set-point T_J^r very closely for most of the batch time. A small deviation between the two signals can be seen around 19:45 and 20:05 due to the limitation of the cooling power Q that the thermostatic bath can generate.

The dynamics between the temperature reference T^r and the crystallizer temperature T is accurately described by the relation $T = F_{TT^r}(T^r)$ up to 20:00. Afterwards, the relation is not accurate any more due to the saturation of the jacket temperature reference T_J^r to the minimum value $5 \text{ }^\circ\text{C}$. Note that even if we would not have this saturation the relation $T = F_{TT^r}(T^r)$ would eventually become less accurate since already from 19:45 the jacket temperature T_J does not follow the reference T_J^r very closely due to the limitation of the cooling power Q . In general, for batch crystallization experiments an accurate temperature tracking at the start of the batch is of the highest importance, while it is less critical towards the end. For this reason, the overall temperature control architecture was considered satisfactory and was kept for the following experiments.

6.4 Crystallization experiments using the pure water-succinic system

6.4.1 Preliminary data and process design

The solubility of Succinic acid in water was first estimated from laboratory analysis by DSM before the start of the experimental campaign. The solubility was given in terms of the molar fraction (mol/mol) i.e. the ratio between the moles of the Succinic acid and the sum of the moles of the Succinic acid and the water as

$$x_s(T) = e^{-\frac{\Delta H_{\text{sol}}}{R} \left(\frac{1}{T_{\text{sol}} + K_o} - \frac{1}{T + K_o} \right)} \quad (6.6)$$

where $K_o = 273.15 \text{ K}$ is the absolute temperature corresponding to $0 \text{ }^\circ\text{C}$, $R = 8.314 \frac{\text{J}}{\text{mol } ^\circ\text{C}}$ is the universal gas constant, ΔH (J/mol) is the molar heat of dissolution of Succinic acid in water and $T_{\text{sol}} = 180.72 \text{ }^\circ\text{C}$. For experimental purpose, it is convenient to express the solubility and the concentrations in terms of the mass fraction (kg/kg), i.e. the ratio between the mass of the Succinic acid and the sum of the masses of the Succinic acid and the water. The solubility in terms of mass fraction can be written as

$$C_{s,\text{lab}}(T) = \frac{x_s(T)M_s}{x_s(T)M_s + (1 - x_s(T))M_w} \quad (6.7)$$

where $M_s = 118.089 \text{ (g/mol)}$ and $M_w = 18.01 \text{ (g/mol)}$ are the molar mass of Succinic acid and water respectively. The solubility line (6.7) is plotted in Figure 6.9.

In practice, it is well known that the measurement of the solubility are rather sensitive and may change from laboratory to the pilot and industrial scales. The results may also depend on the sensors that are used and the experimental procedures that are followed. For this reason, we will verify the solubility using the instruments of the skid at an early stage in our experiments.

Nonetheless, even a rough approximation of the solubility is a valuable tool for a preliminary process design. Using the solubility, we can design the saturation temperatures relative to the initial and the final concentration of the batch in order to obtain a certain solid crystal content at the end of the batch (see Section 3.2). We decided for the experiments to have a saturation temperature around $T_{\text{sat}}(C_i) = 45 \text{ }^\circ\text{C}$ and a final temperature $T_{\text{sat}}(C_f) = 20 \text{ }^\circ\text{C}$. The solid content due to the crystals at the end of the experiment will be approximately $(C_{s,\text{lab}}(T_{\text{sat}}) - C_{s,\text{lab}}(T_f))/C_{s,\text{lab}}(T_f) = 0.1$, which is largely within the specifications that can be safely handled by the equipment at hand.

6.4.2 Calibration of the K-Patents

An experiment was performed in order to calibrate the K-patents for measuring the concentration of the Succinic acid in water. In this experiment, the crystallizer was filled with a known amount of water. Next, a number of loads of Succinic acid were weighed and poured in the crystallized.

After one load of Succinic acid is added, the concentration is known if all the crystals are dissolved in the solution, i.e. if the solution is clear. This condition can

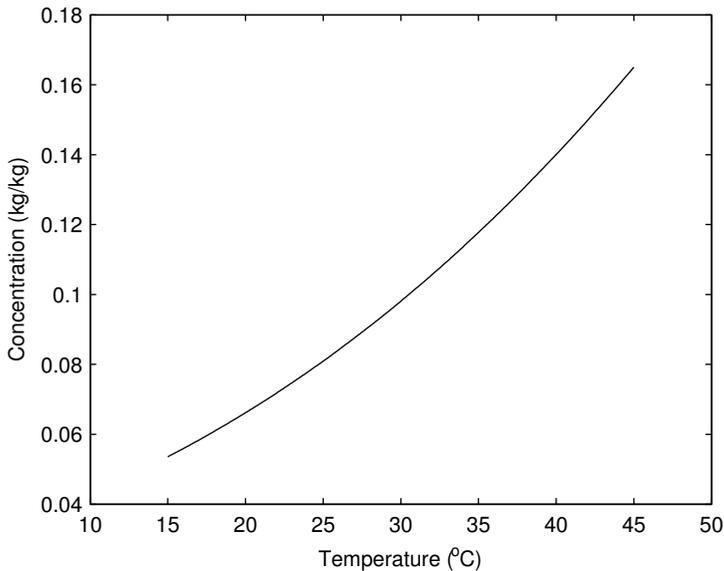


Figure 6.9: The solubility line of Succinic acid in water estimated from laboratory analysis by DSM.

be accurately verified using the ISPV images. The concentration C in this case is known by definition

$$C = C_{\text{ref}} = \frac{S_T}{S_T + S_W}. \quad (6.8)$$

where S_T (kg) is the total amount of Succinic acid and S_W (kg) is the total amount of water present in the crystallizer.

Therefore, when the solution inside the crystallizer is clear the measured value of the refractive index can be used for calibration.

A difficulty arising in the calibration of the K-Patents is that the refractive index depends on both the temperature and concentration. Therefore, the measured values of refractive index have to be calibrated against both the measured values of temperature and the known values of concentration.

The solubility (6.7) was used in order to design the amounts of the loads and the temperature at which they were added. Since the temperature range for the experiments is $45\text{ }^\circ\text{C} - 20\text{ }^\circ\text{C}$, the concentration range of interest is approximately $0.18 - 0.04\text{ kg/kg}$ (see the solubility line in Figure 6.9). Thus, the crystallizer was filled with water and six loads of Succinic acid were prepared in order to obtain concentration values in the desired range. The loads were added at a temperature close to the saturation for the total amount of Succinic present in the crystallizer.

For each load of Succinic acid, the following operations were performed

1. Weigh the load of Succinic acid and pour it in the crystallizer.
2. Check whether the solution is clear. If so, the concentration is valid and the data can be used for calibration. Otherwise, heat up until the solution is

clear. Take note of the instant at which the solution is clear and use the data for calibration starting from that instant.

3. Heat up to a temperature close to the expected saturation temperature for the total amount of the Succinic acid in the vessel after the next load.

The amounts of water and Succinic acid actually fed into the crystallizer are shown in Table 6.1. Note that together with the Succinic acid, water was added for the first three loads. That was done in order to rinse the walls of the vessel and the impeller from a portion of the Succinic acid that accidentally was sticking on them after the load of Succinic acid were poured in.

A number of time profiles relative to this experiments are reported in Figure 6.10. In Figure 6.10a the measured temperature inside the crystallizer is reported. The temperature during the experiment is increased from 22 °C to 50 °C. In Figure 6.10b the refractive index n_D measured by the K-Patents is shown, together with an indicator function of the calibration region which takes the value 1 when the solution was observed to be clear, and 0 otherwise. The data collected by the instrument in the calibration region were used to calibrate the K-Patents. In Figure 6.10c the reference concentration C_{ref} is shown. The values of the refractive index n_d were calibrated against the reference concentration C_{ref} and the measured temperature T in the region where the the indicator function has the value 1. A linear model was found to describe the relation between C_{ref} , n_D , and T and in the calibration region:

$$C_{\text{ref}} \approx \tilde{C}(n_D, T) = c_0 + c_1 n_D + c_2 T. \quad (6.9)$$

The parameters $c_0 = -12.7117$, $c_1 = 9.5056$, $c_2 = 0.0017$ were found through a least squares fitting procedure.

The quality of the fit of this model can be appreciated in the same Figure 6.10c where the estimated concentration $\tilde{C}(n_D, T)$ is reported. The estimated concentration follows the reference C_{ref} in the calibration region very well. The Root Mean Square of the residual $C_{\text{ref}} - \tilde{C}(n_D, T)$ is

$$\hat{\sigma}_C = \sqrt{\frac{1}{N} \|C_{\text{ref}} - \tilde{C}(n_D, T)\|_2^2} = 8.84 \cdot 10^{-4}.$$

The quantity $\hat{\sigma}_C$ can be seen an estimate of the standard deviation of the concentration noise. Since the typical values of the concentration C are in the order of 10^{-1} kg/kg, the concentration can be monitored accurately using the K-Patents sensor. However, our ultimate objective is to use the concentration measurements in order to estimate the supersaturation S , whose typical values are in the order of 10^{-3} kg/kg. Since $\hat{\sigma}_C$ has the same order of magnitude as the supersaturation S , estimating the latter using the K-Patents could be somewhat involved.

Outside the calibration region, the actual concentration is not known because part of the Succinic acid is not dissolved in the solution and therefore the actual concentration is less than C_{ref} . Thus, the model $\tilde{C}(n_D, T)$ is the only information available about the concentration and for this reason a quantitative analysis of the fit performance is not possible. From a qualitative perspective the model $\tilde{C}(n_D, T)$

Time	Amounts added (kg)	S_T (kg)	W_T (kg)	C_{ref} (kg/kg)
11:16	16.16 W	0	16.16	0
11:34	0.800 S + 0.290 W	0.800	18.89	0.0406
12:18	0.816 S + 0.355 W	1.616	19.18	0.0777
13:49	0.718 S + 0.244 W	2.335	19.53	0.1068
14:05	0.500 S	2.835	19.78	0.1254
15:24	0.799 S	3.635	19.78	0.1553
15:44	0.635 S	4.270	19.78	0.1775

Table 6.1: Loads of water (W) and Succinic acid (S) introduced in the crystallizer for the calibration experiment.

delivers a result which is in line with our physical insight about the process. The concentration has step increases in correspondence with the loads of Succinic acid and it gradually increases during the heat-up when the Succinic acid is being dissolved. After all the Succinic acid is dissolved, the concentration remains constant even if the temperature is increased.

From now on, we will refer to the concentration model $\tilde{C}(n_d, T)$ as the measured concentration C for simplicity.

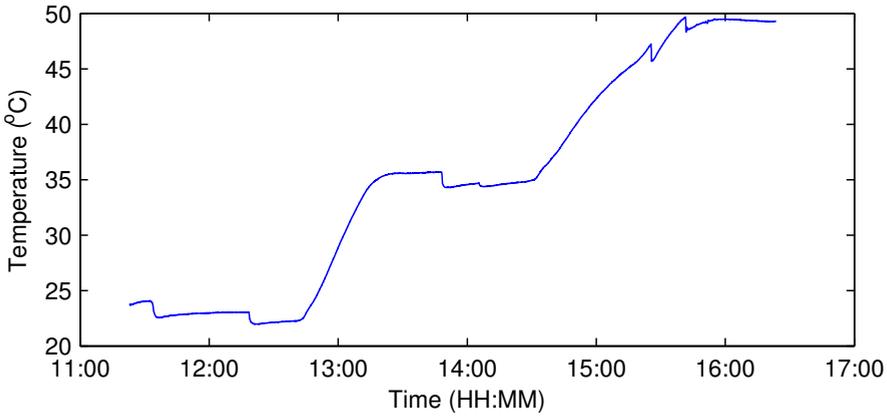
6.4.3 Determination of the solubility

As already mentioned, the solubility of a component is sensitive to several factors and the information available from laboratory analysis has to be validated on the real experimental setup. A precise knowledge of the solubility line is required in order to characterize the crystallization system, e.g. to compute the supersaturation and evaluate the MSZW.

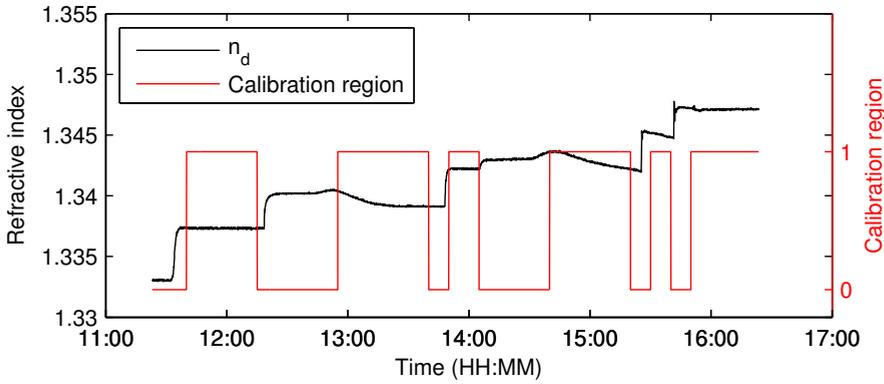
For this reason, the solubility was redetermined performing an heat-up experiment after the calibration model for the K-Patents was obtained. The experiment starts with the crystallizer at the temperature $T_i = 20.6$ °C containing 4.27 kg of Succinic acid and approximately 19.78 kg of water³. The initial measured concentration is $C_i = 0.057$. Thus, 1.38 kg of Succinic acid are dissolved in the water, while the remaining 2.89 kg are present in form of crystals in the suspension. The crystallizer is slowly heated up and the crystals that were in suspension start to dissolve. Since the dissolution dynamics is very fast, the concentration remains always very close to the solubility during the heat-up phase (see Figure 2.1 in Chapter 2). For this reason, the temperature and concentration data collected in this experiment can be used in order to reconstruct the solubility.

The time profiles of temperature and concentration of this experiments are reported in Figure 6.11, while the $C - T$ plane is reported in Figure 6.12. From these plots it is clear that there is a mismatch between the solubility obtained from laboratory data $C_{s,\text{lab}}(T)$ and the actual one. Therefore, the solubility has been redetermined based on the experimental data. The solubility was modeled using

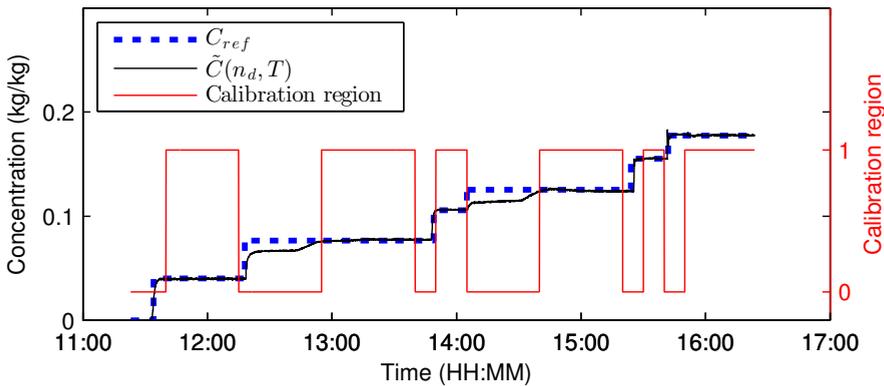
³These conditions were obtained by cooling down the crystallizer after the K-Patents calibration experiment. The uncertainty on the amount of water is due to the evaporation of part of the water.



(a) Temperature T inside the crystallizer.

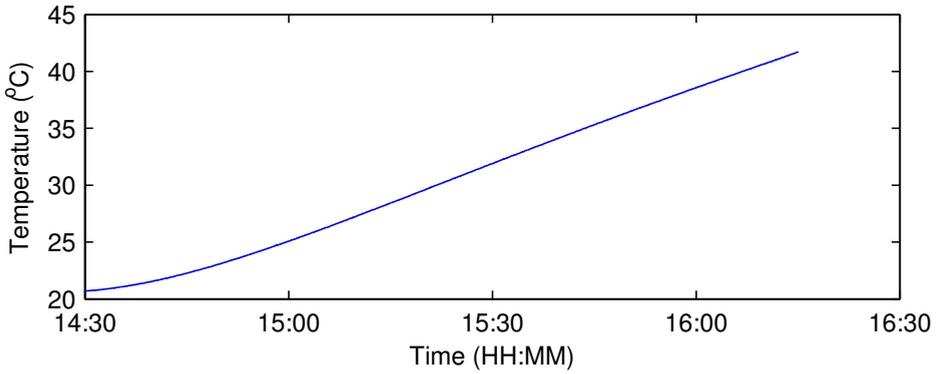


(b) Refractive index n_d from the K-Patents (black) and clear solution indicator function (red).



(c) Reference concentration C_{ref} (blue), concentration model $\tilde{C}(n_D, T)$ (black) and clear solution indicator function (red).

Figure 6.10: Time profiles for the calibration experiment.



(a) Measured temperature inside the crystallizer.

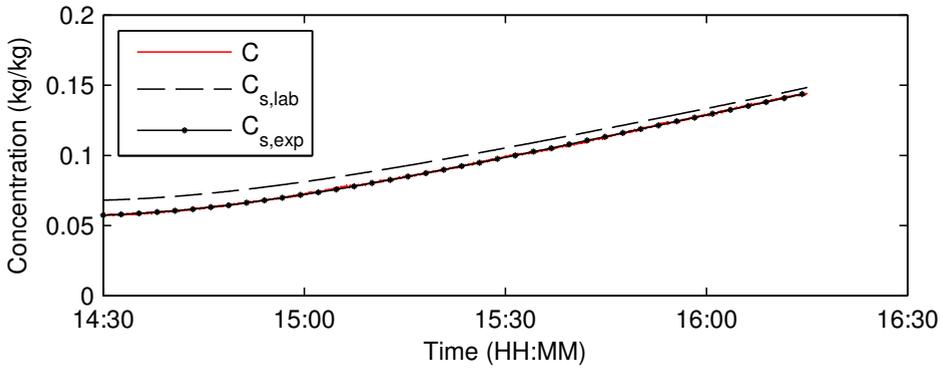
(b) Solubility from laboratory analysis $C_{s,\text{lab}}(T)$, measured concentration C , and experimental solubility $C_{s,\text{exp}}(T)$.

Figure 6.11: Time profiles of the heat-up experiment for the determination of the solubility.

the second order polynomial

$$C_{s,\text{exp}}(T) = a_0 + a_1T + a_2T^2. \quad (6.10)$$

The parameters $a_0 = 1.3388$, $a_1 = -6.250 \cdot 10^{-5}$, $a_2 = 5.2164 \cdot 10^{-6}$ were estimated using a least squares fitting procedure based on the concentration C and temperature T measured during the heatup phase until the solution became clear. The curve $C_{s,\text{exp}}(T)$ is also reported in the $C-T$ plane in Figure 6.12.

For the rest of this section, will refer to the experimental solubility $C_{s,\text{exp}}(T)$ as $C_s(T)$ and we will use it to compute the supersaturation as $S = C - C_{s,\text{exp}}(T)$.

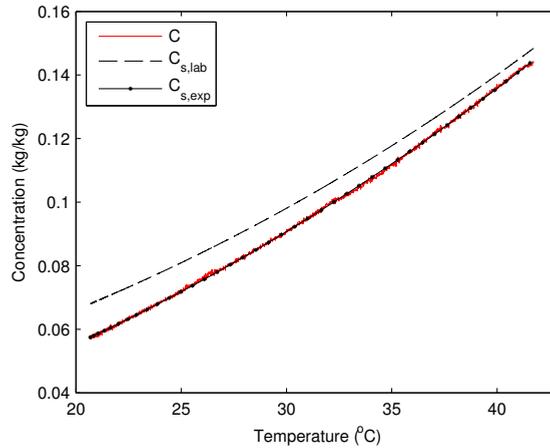


Figure 6.12: $C-T$ plane of the heatup experiment for the determination of the solubility. The solubility $C_{s,\text{exp}}$ is interpolated based on the measured concentration and temperature data. There is a mismatch between $C_{s,\text{exp}}$ and the solubility obtained from laboratory data $C_{s,\text{lab}}$.

6.4.4 Unseeded batch cooling experiment

An unseeded batch experiment was performed to build up insight in the process and the measuring equipment. The crystallizer was cooled down following a linear profile from $T_i = 47^\circ\text{C}$ to $T_f = 21^\circ\text{C}$ in approximately 4 h. The initial concentration is $C_i = 0.1541$ kg/kg and the solution in the crystallizer is clear. The saturation temperature is indeed $T_{\text{sat},i} = C_s^{-1}(C_i) = 43.5^\circ\text{C}$. The final concentration is expected to be very close to $C_s(T_f) = 0.0585$ kg/kg. Therefore, the final mass fraction of the crystals will be $C_i - C_s(T_f) = 0.0956$ kg/kg.

The time profiles of temperature, concentration and supersaturation are presented in Figure 6.13. The nucleation event is evident from the concentration and supersaturation plots. Indeed, the concentration is constant until 10:43 and rapidly drops close to the solubility in the following few minutes. Since the solution was previously clear, this is a consequence of the nucleation event and of the subsequent growth of the crystals. Afterwards, the concentration remains very close to the solubility for the rest of the experiment.

The supersaturation takes its maximum value of 0.01 kg/kg at 10:43 and rapidly drops close to 0 where it remains for the rest of the experiment.

The $C-T$ plane of the experiment is reported in Figure 6.14. From the $C-T$ plane we observe that the concentration starts to drop when the temperature is 41.5°C . Since the initial saturation temperature was $T_{\text{sat},i} = 43.5^\circ\text{C}$, the MSZW is 2°C .

The images collected by the ISPV microscope (low magnification) are reported in Figure 6.15. The first crystal is observed at 10:35, few minutes before the drop of the concentration was observable.

We also see that it is possible to distinguish single crystals in the images only when the solid content is rather low (0 – 1.5%), while for higher densities the

images are very dark and crystals overlap on each other in the images. Note that this is not a severe limitation since in general in a crystallization experiment the first part of the process (where the crystal content is low) is the most critical.

6.4.5 Seeded batch experiment

An experiment was performed to verify the effect of seeding on the crystallization process. The crystallizer was cooled down following a linear profile from $T_i = 47\text{ }^\circ\text{C}$ to $T_f = 21\text{ }^\circ\text{C}$ in approximately 4 h. The initial concentration is $C_i = 0.1688\text{ kg/kg}$ and the solution in the crystallizer is clear. The saturation temperature is $T_{\text{sat}} = C_s^{-1}(C_i) = 46.2\text{ }^\circ\text{C}$.

Seeding with 163 g of coarse Succinic acid (mean size $\approx 150\mu\text{m}$) was applied at 16:23 PM. The seed material was directly poured into the crystallizer. This form of seeding where the solid crystals are poured directly in the crystallizer is known as dry seeding. At the moment of the seeding, the solution was clear and the temperature was $45.9\text{ }^\circ\text{C}$.

Few seconds after the seeding, crystals were observed with both the ISPV microscopes. This is a clear indication that the seeds actually survived in the crystallizer. The concentration rapidly drops and approaches the solubility in the following minutes. Therefore, the supersaturation drops to 0. Afterwards, the concentration remains close to the solubility for the rest of the experiment and the supersaturation remains always close to 0. The supersaturation seems to be actually lower than 0 in certain time instants, which has no physical explanation for a crystallization experiment. Possible causes could be a minor shift in the solubility or a small bias of the concentration measurements.

6.4.6 Conclusions on the preliminary experiments

The preliminary experiments were useful to calibrate the K-Patents instrument to measure the concentration and to build up preliminary experience about the process and the equipment. The following conclusions were drawn:

- The K-Patents can be used in order to measure the concentration of the Water-Succinic crystallization system throughout the batch crystallization process. However, estimating the supersaturation using the K-Patents could be complicated since the supersaturation has the same order of magnitude of the standard deviation of the noise of the concentration measurements.
- The ISPV can be used in order to observe crystal formations when the solid content of the vessel is in the range (0 – 1.5%). This is sufficient in order to observe the onset of nucleation in an unseeded batch and to verify that the seeds do not dissolve in a seeded batch.
- The actual solubility is considerably different from the one obtained from laboratory analysis. Furthermore, minor changes in the solubility may have happened during the experiments. It is necessary to verify the solubility throughout the experiments in order to be able to interpret the results correctly. The solubility can be easily verified performing an heat-up experiment.

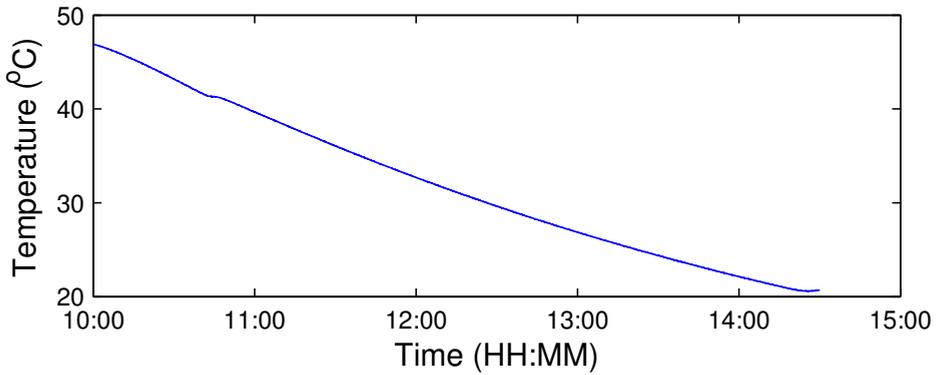
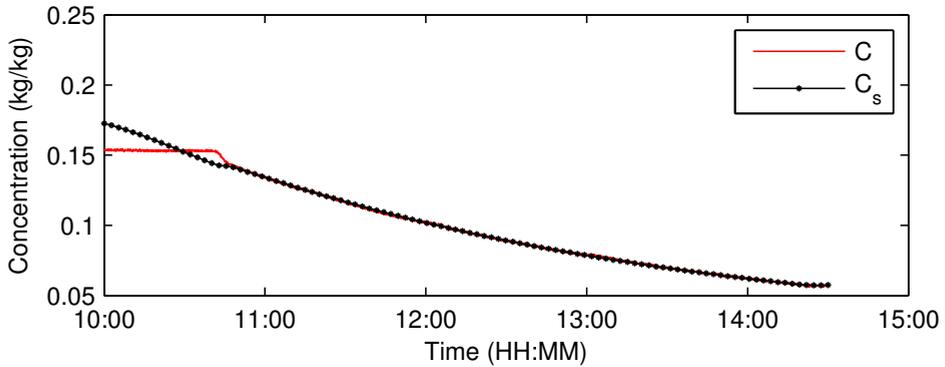
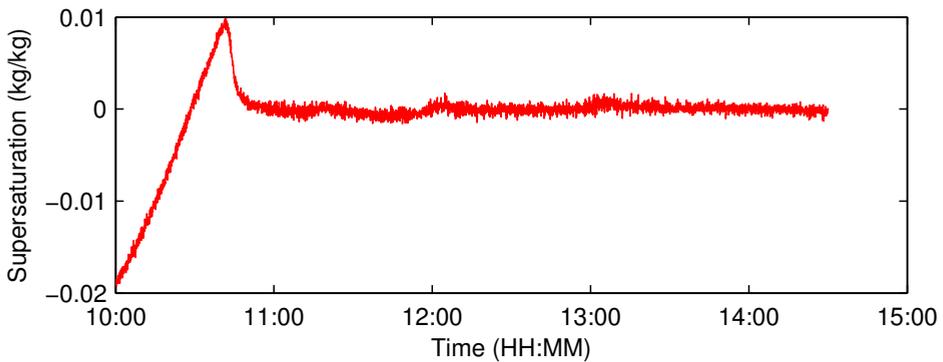
(a) Temperature T inside the crystallizer.(b) Concentration C (red) and solubility C_s (black line). At around 10:45 the concentration drops and remains very close the solubility for the rest of the experiment.(c) The supersaturation S reaches a maximum at around 10:45 and drops to 0 afterwards.

Figure 6.13: Time profiles of the unseeded batch experiment.

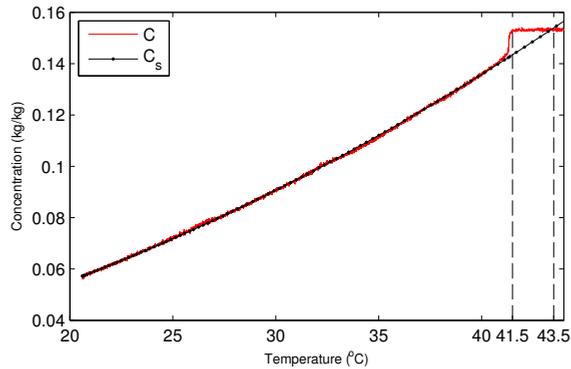
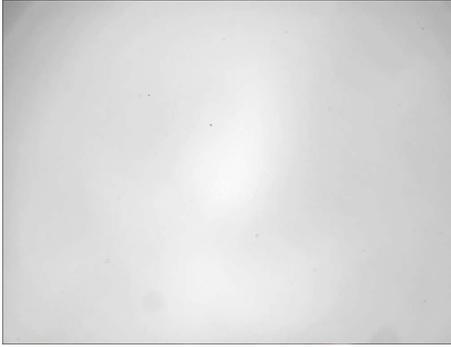
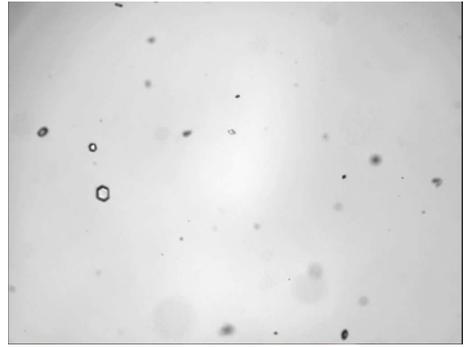


Figure 6.14: $C - T$ plane of the unseeded batch crystallization experiment. The Metastable Zone Width is approximately $2\text{ }^{\circ}\text{C}$.

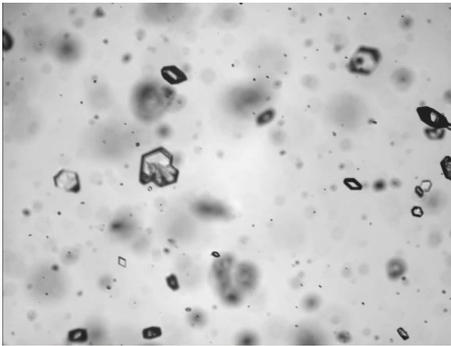
- The MSZW is rather narrow (approximately $2\text{ }^{\circ}\text{C}$). Performing seeded experiments with such narrow MSZW is technically challenging. Furthermore, the choice of the initial supersaturation is limited.
- The crystallization dynamics of the crystals is very fast. In the seeded and the unseeded experiments, the concentration dropped in few minutes close to the solubility after seeding and primary nucleation respectively. Therefore, the supersaturation is very small during most of the experiment and it is difficult to measure with the given equipment, even when cooling at the maximum speed. This leaves limited possibilities for supersaturation control.



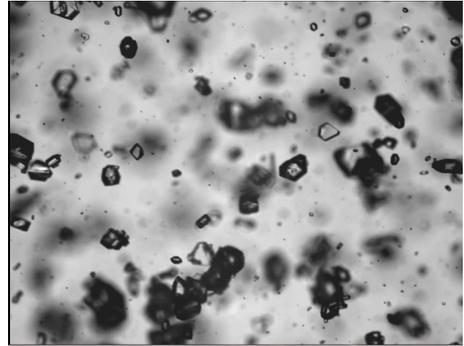
(a) $t=10:30$, $T = 41.71$ °C, $C_{\text{solid}} = 0$.
The solution is clear.



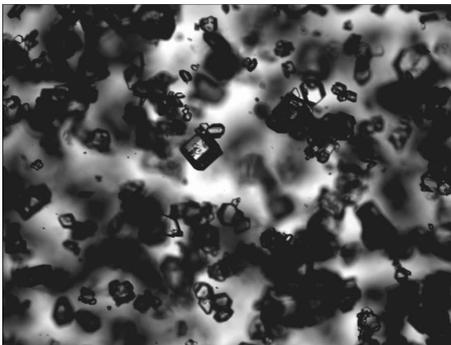
(b) $t=10:40$, $T = 41.71$ °C, $C_{\text{solid}} \approx 0$.
The first crystal formations are visible.



(c) $t=10:45$, $T = 41.3$ °C, $C_{\text{solid}} = 0.8\%$.
The crystals grow and increase in number due to nucleation. Most of the crystals are still individually distinguishable.



(d) $t=10:52$, $T = 40.64$ °C, $C_{\text{solid}} = 1.4\%$.
More and more crystals are present. Some of the crystal start to overlap on each other on the image.



(e) $t=11:19$, $T = 37.29$ °C, $C_{\text{solid}} = 3\%$.
The crystals keep on increasing in number and in size. The image get darker and it becomes more and more difficult to distinguish individual crystals.



(f) $t=12:05$, $T = 32.10$ °C, $C_{\text{solid}} = 5.3\%$.
The solid content is very high and the image is almost completely dark. single crystals are hardly distinguishable even for a human eye.

Figure 6.15: Images captured by the ISPV probe (low magnification) for the unseeded batch experiment.

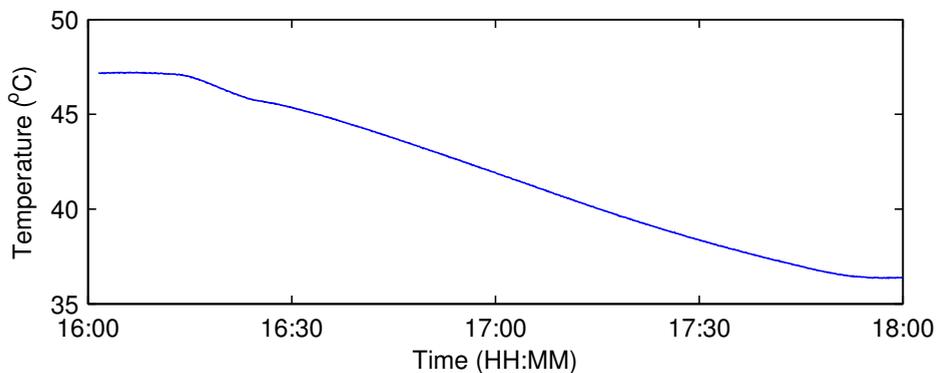
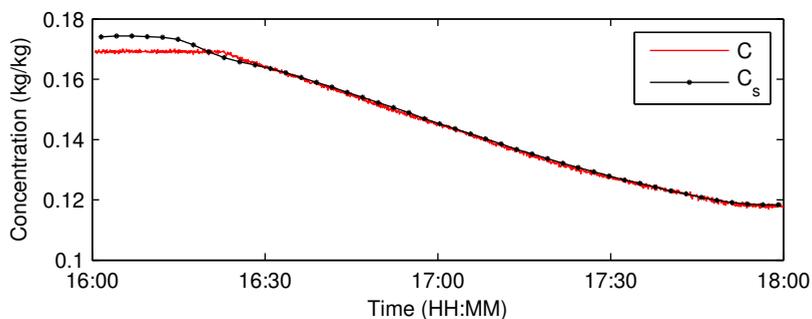
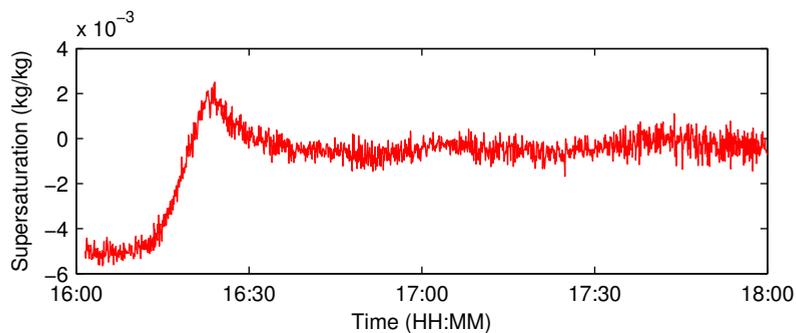
(a) Temperature T inside the crystallizer.(b) Concentration C (red) and solubility C_s (black line). Soon after the seeding the concentration drops and remains very close the solubility for the rest of the experiment.(c) The supersaturation S reaches a maximum soon after the seeding and drops to 0 afterwards.

Figure 6.16: Time profiles of the first seeded batch crystallization experiment.

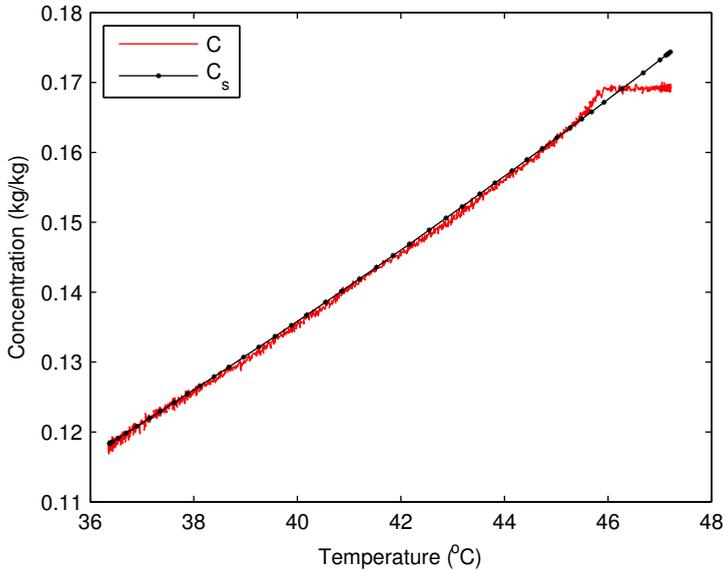


Figure 6.17: $C-T$ plane of the first seeded crystallization experiment.

6.5 Crystallization experiments in the presence of Fumaric acid as impurity

In the following experiments, a small amount of impurity (Fumaric acid) was added to the water-succinic system. The presence of the impurity is expected to slow down the growth rate of the crystals. In turn, this would slow down the dynamics of the supersaturation giving better chances for control. Furthermore, it is known that impurities are often present in industrial crystallization processes and for this reason it is relevant to consider their presence in the control experiments.

For the following experiments, the crystallizer was loaded with 45.221 kg of water, 8.646 kg of Succinic acid and 0.400 kg of Fumaric acid.

6.5.1 Determination of the solubility line in the presence of Fumaric Acid

Due to the change in the composition, a change in the solubility line is to be expected. For this reason, the solubility was redetermined based on an experiment similar to the one presented in Section 6.4.3.

The new experimental solubility is

$$C_{s,\text{exp,fum}}(T) = a_0 + a_1T + a_2T^2 + a_3T^3. \quad (6.11)$$

with $a_0 = 0.0152$, $a_1 = 0.0013$, $a_2 = 4.8803 \cdot 10^{-5}$, $a_3 = -1.5245 \cdot 10^{-08}$. The new solubility $C_{s,\text{exp,fum}}(T)$ is reported in the Figure 6.18 together with the previous experimental solubility based on the pure water-succinic system $C_{s,\text{exp}}(T)$ and

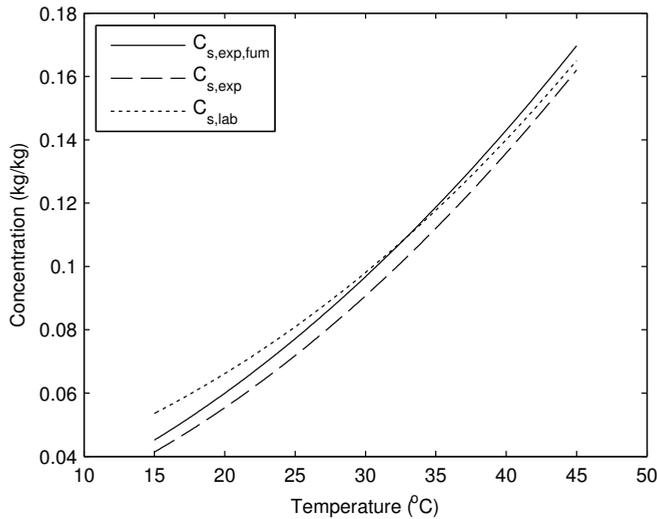


Figure 6.18: Experimental solubility with Fumaric acid $C_{s,exp,fum}(T)$, experimental solubility based on the pure system $C_{s,exp}(T)$ and solubility from laboratory analysis $C_{s,lab}(T)$.

the initial solubility line obtained from laboratory analysis $C_{s,lab}(T)$.

The mismatch between the three curves is evident in the plot. From now on, we will refer to the new solubility $C_{s,exp,fum}(T)$ as $C_s(T)$ for simplicity and we will use it to compute the supersaturation as $S = C - C_{s,exp,fum}(T)$.

The solubility was verified several times during the experimental campaign by repeating the heat-up experiment and it was found to be consistent.

6.5.2 Unseeded batch experiments

An increase of the MSZW is also to be expected for the crystallization in presence of the impurity. Different unseeded batch experiments similar to the one presented in Section 6.4.4 were performed. The measured width of the MSZW for these experiments was around 2.7 °C. Note that we have a certain increase with respect to the 2.0 °C MSZW of the pure system.

6.5.3 Seed preparation

In the rest of the experimental campaign, seeded batch crystallization experiments with the impure system were performed. Milled crystals of mean size 25 μm were made available by DSM for seeding. For each seeded batch experiment 5 g. of seeding material was used. The seeds were introduced into the vessel in the form of a suspension in a saturated solution. This form of seeding is called wet seeding and it is expected to be more effective in order to limit the secondary nucleation and obtain a growth-dominated crystallization process. The procedure followed in order to obtain the seeds is the following



Figure 6.19: The second flask is stirred for 10 minutes with a magnetic stirrer. A well-dispersed seed suspension is obtained.

1. Weigh 10 gram of coarse Succinic acid crystals in a 100 ml Erlenmeyer flask. Add water up to 100 ml at room temperature.
2. Add 4 drops of Polysorbate 80 using a syringe-needle.
3. Stir for 30 minutes at room temperature to obtain a saturated solution.
4. Weigh 5.0 gram of seed material having mean size $25 \mu\text{m}$ in a second 100 ml Erlenmeyer flask.
5. Decant about 80 ml of saturated solution from the first to the second flask.
6. Stir the second flask for 10 minutes (Figure 6.19).

The saturated solution is prepared in the first flask and it is poured to the second flask where the seeds were previously added. The Polysorbate 80 is used to favor the dispersion of the seeds in the saturated solution. Thus, at the end of the procedure, seeds in the form of a well-dispersed suspension in a saturated solution are obtained in the second flask.

6.5.4 Nominal seeded batch experiment

A seeded batch crystallization was performed to verify the behavior of the crystallization system in the presence of the impurity and the new seeding procedure. This experiment is also considered as a “nominal case” for the following B2B control experiments.

The initial concentration was $C_i = 0.1739 \text{ kg/kg}$ and the corresponding saturation temperature is $T_{\text{sat},i} = C_s^{-1}(C_i) = 45.7 \text{ }^\circ\text{C}$. The seeds were prepared according to the procedure presented in the previous subsection and introduced in the

crystallizer at the temperature $T_{\text{seed}} = 45 \text{ }^\circ\text{C}$. After the seeding, the crystallizer was cooled from $45 \text{ }^\circ\text{C}$ to $20 \text{ }^\circ\text{C}$ in 3 hours following a linear temperature profile. The time profiles for temperature, concentration and supersaturation starting from the moment of the seeding to the end of the experiment are reported in Figure 6.20, while the $C-T$ plane is reported in Figure 6.21. A number of pictures of the crystallizer collected during this experiment are reported in Figure 6.22.

We observe that the dynamics of the supersaturation is now slower compared to the one of the seeded experiment with pure Succinic acid. After the seeding, it takes approximately one hour for the supersaturation to go to zero.

The data collected from this experiment were also used in order to identify a model of the system dynamics to be used in the B2B control experiments. The measured temperature T and concentration C were used to estimate the kinetic parameters k_g and g of the model (2.44)-(2.48). The estimated parameters were $\hat{\theta}_0 = [\hat{k}_{g,0} \hat{g}_0]^\top = [6.77 \ 1.05]^\top$. These parameters are used to construct a model $F_{ST}(\cdot, \hat{\theta}_0)$ from the crystallizer temperature T to the supersaturation S . The output of supersaturation given by the identified model $F_{ST}(T, \hat{\theta}_0)$ fed by the measured temperature T is also reported in Figure 6.20c. From this plot, the model seems to describe the dynamics rather accurately.

The model $F_{ST}(\cdot, \hat{\theta}_0)$ is integrated with the model of the temperature dynamics $F_{TT^r}(\cdot)$ already estimated in Section 6.3 and an input/output model $F_{ST^r}(\cdot, \theta_0)$ from the temperature reference T^r to the supersaturation S is obtained:

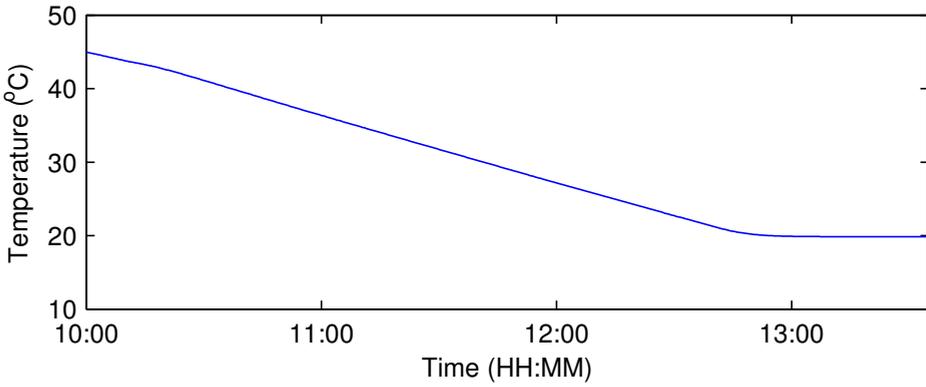
$$S = F_{ST^r}(\cdot) = F_{ST}(F_{TT^r}(\cdot)). \quad (6.12)$$

6.5.5 Conclusions on the experiments in the presence of the impurity

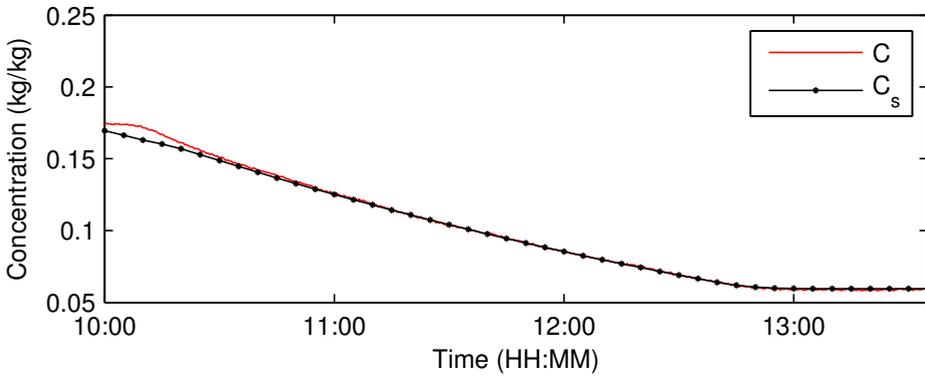
The behavior of the system in the presence of Fumaric acid as an impurity is rather different from the one of the pure water-succinic system. In particular, the following observations are made

- The supersaturation dynamics is considerably slowed down. After the seeding, the supersaturation takes approximately one hour to move close to the solubility compared to the few minutes required for the pure system.
- The MSZW is increased from $2 \text{ }^\circ\text{C}$ to $2.7 \text{ }^\circ\text{C}$.
- The solubility has significantly changed and it has been redetermined experimentally.

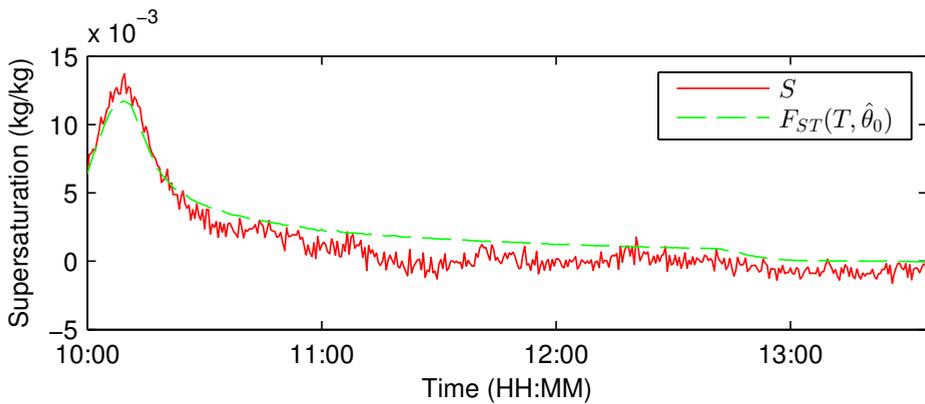
The slower supersaturation dynamics and the larger MSZW leave better chances for control. For these reasons, the B2B control experiments presented in the following section are based on the crystallization system in the presence of Fumaric acid.



(a) Temperature T inside the crystallizer.



(b) Concentration C and solubility C_s .



(c) The supersaturation S reaches a maximum soon after the seeding and drops to 0 afterwards.

Figure 6.20: Time profiles of the nominal seeded batch experiment.

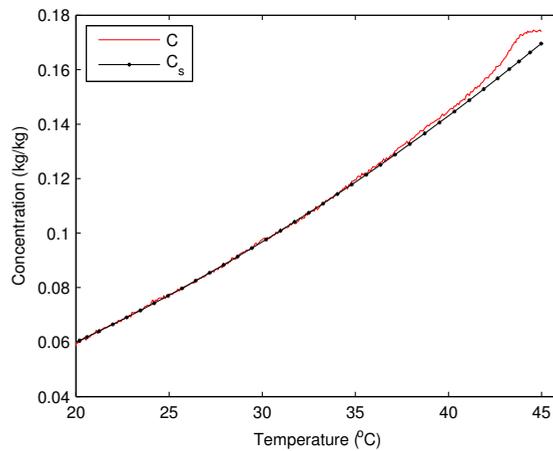


Figure 6.21: $C-T$ plane of the nominal seeded batch experiment.



(a) The temperature in the crystallizer is $45\text{ }^{\circ}\text{C}$ and the solution is clear.



(b) The seeds are poured from the flask into the crystallizer.



(c) Soon after seeding, the seeds are visible as a turbid cloud inside the crystallizer.



(d) During the cooling down, the solid content in the crystallizer increases and the slurry gets more and more turbid.

Figure 6.22: Pictures of the crystallizer during the nominal seeded batch experiment.

6.6 Batch to batch control experiments

The batch to batch control experiments were performed with the water-succinic crystallization in the presence of Fumaric acid as an impurity described in the previous section. The temperature set-point T^r for each batch was optimized in order to keep the supersaturation at a constant value S^r for the time of the batch. The model used to optimize the temperature trajectory for one batch was updated based on the the data collected from the previous batch performing IIC-type or ILC-type corrections.

6.6.1 Control architecture

The control architecture used for the batch to batch experiments is reported in Figure 6.23. In the scheme, continuous lines represent signals that are updated on-line during a batch, while dashed lines represent signal that are updated only from one batch to the other. The B2B controller optimizes the set-point T^r of the master temperature controller in order to keep the supersaturation S close to the reference S^r for the time of the batch. After a batch is performed, the B2B controller first updates the model based on the data collected during the batch. Subsequently, it uses the model to optimize the temperature reference profile T^r for the next batch in order to fulfill the control objective.

The performance of the controller is evaluated by considering the Root Mean Square (RMS) of the supersaturation tracking error $S - S^r$ for the first two hours of the experiment. The data after 2 hours was ignored for the performance analysis because the measured supersaturation dropped to values close to 0 (or even negative) after 2 hours for all the experiments.

6.6.2 Experimental Procedure

The following experimental procedure was followed for each of the B2B experiments (see Figure 6.24)

1. Heat up the crystallizer to the temperature $T_h = 50$ °C such that all the crystals have been dissolved in the reactor.
2. Keep the temperature at T_h for 15 min.

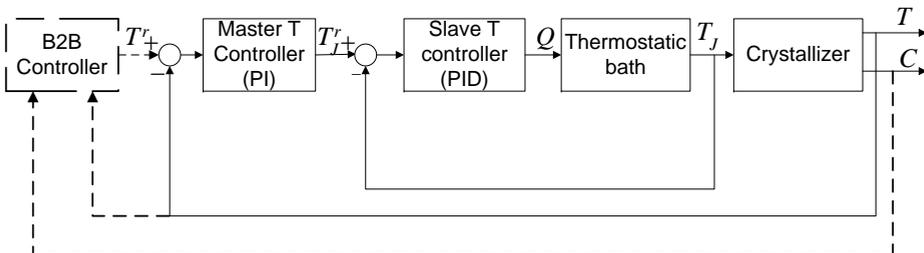


Figure 6.23: The control architecture for batch to batch control.

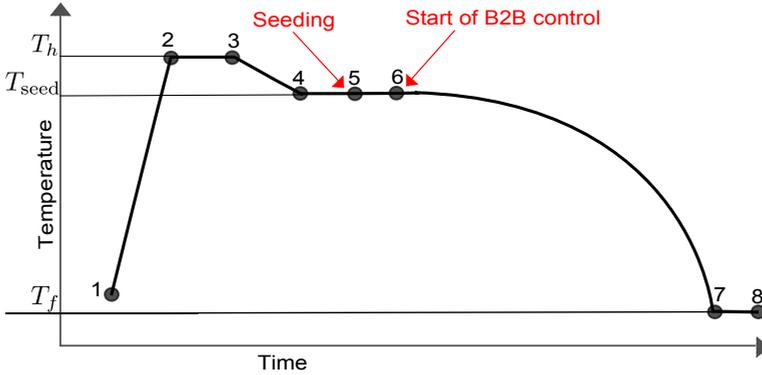


Figure 6.24: The steps of the experimental procedure for the B2B experiments. The B2B algorithm is used to determine the cooling trajectory at the step 6 of the procedure.

3. Cool down from T^h to $T_{\text{seed}} = 45\text{ }^\circ\text{C}$ in 30 min.
4. Keep the temperature at T_{seed} for 15 min.
5. Introduce the seeds prepared according to the procedure described in Section 6.5.3.
6. Cool down from T_{seed} to T_f according to the optimal input profile generated by the B2B algorithm.

The initial concentration was around $C_i = 0.1731$ and the corresponding saturation temperature is $T_{\text{sat},i} = C_s^{-1}(C_i) = 45.6\text{ }^\circ\text{C}$. The initial supersaturation is $S_i = C_i - C_s(T_{\text{seed}}) = 0.0034\text{ kg/kg}$. However, it has to be noted that due to technical difficulties the actual values of T_h , T_{seed} , C_i and S_i have a certain variability from one batch to the other. The final temperature T_f is not fixed and depends on the particular temperature reference profile T^r that is generated by the B2B algorithm.

6.6.3 Batch to batch supersaturation control experiments

B2B experiment 1: tracking of $S^r = 0.0028\text{ kg/kg}$ using IIC. In this experiment, the objective was to track a constant supersaturation $S^r = 0.0028\text{ kg/kg}$. The reference temperature profile was optimized based on the model estimated on the nominal seeded batch experiment (see Section 6.5.4) in order to track the supersaturation set-point $S^r = 0.0028\text{ kg/kg}$. This value for the supersaturation set-point was chosen as the average of the supersaturation in the first two hours of the nominal seeded experiment. This can be considered as an IIC model correction starting from the nominal process trajectory.

The time profiles of the reference temperature T_1^r , the measured temperature T_1 , the temperature model $F_{TT^r}(T_1^r)$, and the supersaturation S_1 for this experiment are reported in Figure 6.25. The temperature T_1 is described by the model $F_{TT^r}(T_1)$ as expected for most of the experiment and moves slightly off towards

the end (after 20:00) indicating that the constraint on the maximum cooling power of the thermostatic bath has been reached.

Compared with the nominal seeded batch experiment, we see that the maximum of the supersaturation is slightly lower (0.012 kg/kg instead of 0.014 kg/kg). However, the supersaturation is still rather far from the set-point $S^r = 0.0028$ kg/kg. The RMS of the supersaturation tracking error for the first 2 hours of experiment ⁴ is 0.0042 kg/kg. In the supersaturation plot the output of the previously identified model $F_{ST}(T_1, \hat{\theta}_0)$ fed with the measured reactor temperature T_1 is reported. We see that the output of the model $F_{ST}(T_1, \hat{\theta}_0)$ is very far from the measured supersaturation S_1 . This suggests that there might be a severe mismatch between the model and the actual system dynamics.

B2B experiment 2: same input as B2B experiment 1. The unexpected result of the B2B experiment 1 was tested for reproducibility. The B2B experiment 2 was performed with the same temperature reference as in the B2B experiment 1, i.e. $T_2^r = T_1^r$.

The time profiles of the reference temperature T_2^r , the measured temperature T_2 , the temperature model $F_{TTr}(T_2^r)$ and the supersaturation S_2 for this experiment are reported in Figure 6.26. The results of this experiment is similar to the previous case. The RMS of the supersaturation tracking error for the first 2 hours of experiments is 0.0040 kg/kg. This consolidates the hypothesis of a model mismatch.

The measured temperature T_2 and concentration C_2 collected from 17:30 to 19:10 were used to update the estimates for the parameters k_g and g .⁵ The estimated parameters were $\hat{\theta}_2 = [\hat{k}_{g,2} \hat{g}_2]^\top = [1.78 \ 1.00]^\top$. The output of the new model $F_{ST}(T_2, \hat{\theta}_2)$ and of the previous model $F_{ST}(T_2, \hat{\theta}_0)$ are reported in the Figure 6.26b. Note that even the new model $F_{ST}(T_2, \hat{\theta}_2)$ does not interpolate the data very closely and it seems to be a compromise to describe a growth dynamics that is slower in the first part of the experiment (where the model predicts lower supersaturation than the measurements) and larger towards the end (where the model predicts a lower supersaturation than the measurements).

In order to confirm this hypothesis, we estimated the model parameters again using only the data from 17:30 to 18:30 where the growth dynamics seems to be slow. The estimated parameters were $\tilde{\theta}_2 = [\tilde{k}_{g,2} \tilde{g}_2]^\top = [0.55 \ 1.03]^\top$. The response of the model $F_{ST}(T_2, \tilde{\theta}_2)$ is also represented in the Figure 6.26b and it fits accurately the identification data (i.e. the response from 17:30 to 18:30), but is completely off the measurement afterwards where it underestimates the actual growth. Apparently, the actual growth behavior is more complicated than the one that the model can capture. In other words, we are in the presence of a structural model mismatch.

In the B2B experiments 1 and 2, it appeared to be difficult to follow the super-

⁴In all the experiments, the measured supersaturation is very close to 0 (or even negative) after 2 hours. For this reason, we consider the supersaturation tracking only in the first 2 hours.

⁵The data after 19:10 was ignored because the measured supersaturation is lower than zero, which has no physical meaning that can be captured by our model.

saturation set-point $S^r = 0.0028$ kg/kg for this crystallization system. At the start of the batch, the growth seems to be very small and the supersaturation builds up to values much higher than $S^r = 0.0028$ kg/kg before it starts being consumed. For this reason, in the following B2B experiments we increased the supersaturation set-point to an higher value $S^r = 0.006$ kg/kg.

B2B experiment 3: tracking of $S^r = 0.006$ kg/kg using IIC+ILC. The set-point of the supersaturation was increased to $S^r = 0.006$ kg/kg. In order to compensate for the structural model mismatch verified in B2B experiments 1 and 2, an ILC correction based on the model $F_{ST}(\cdot, \hat{\theta}_2)$ was performed. The correction vector α_3 was computed as $\alpha_3 = H(q)(S_2 - F_{ST^r}(T_2, \hat{\theta}_2))$ where S_2 is the measured output from the previous experiment and $H(q)$ is a low-pass filter having a bandwidth of 5 minutes. The ILC-corrected model is $F_{ST}(\cdot, \hat{\theta}_2) + \alpha_3$. This can be considered as a combined IIC+ILC step from the B2B experiment 2 since the parameters of the model $F_{ST}(T, \hat{\theta}_2)$ are also estimated on the data of the B2B experiment 2. The temperature profile was optimized based on the the model $F_{ST}(\cdot, \hat{\theta}_2) + \alpha_3$. The time profiles of the reference temperature T_3^r , the measured temperature T_3 , the temperature model $F_{TTr}(T^r)$ and the supersaturation S_3 for this experiment are reported in Figure 6.27. Note that the reference T_3^r decreases very slowly for the first part of the experiment, where the consumption of concentration due to the growth of the crystal is also low. As a result, the supersaturation remains relatively close to the reference for the first part of the batch. Towards the end of the batch, the reference T_3^r decreases very fast since a much higher growth is expected. However, the actual crystallizer temperature cannot decrease as fast as required due to the limitations of the thermostatic bath (note that T_3 moves away from the model $F_{TTr}(T_3^r)$) and the supersaturation S_3 drops to values lower than the reference S^r . The RMS of the supersaturation tracking error for the first 2 hours of experiments is 0.0025 kg/kg, which is a significant improvement with respect to batch 2.

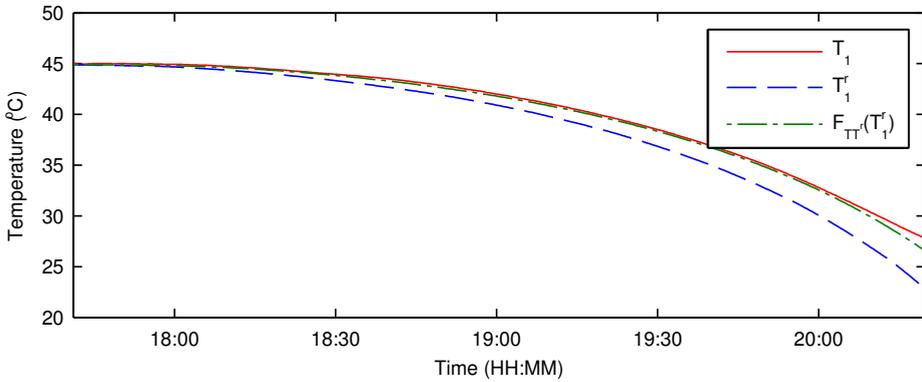
B2B experiment 4: tracking of $S^r = 0.006$ kg/kg using ILC. A final B2B iteration was performed with the same set-point $S^r = 0.006$ kg/kg. An ILC correction based on the data from the previous experiment was performed. The correction vector α_4 was computed as $\alpha_4 = H(q)(S_3 - F_{ST^r}(T_3, \hat{\theta}_2))$.

The ILC-corrected model was found as $F_{ST}(\cdot, \hat{\theta}_2) + \alpha_4$ and the temperature profile T_4^r was found based on this model.

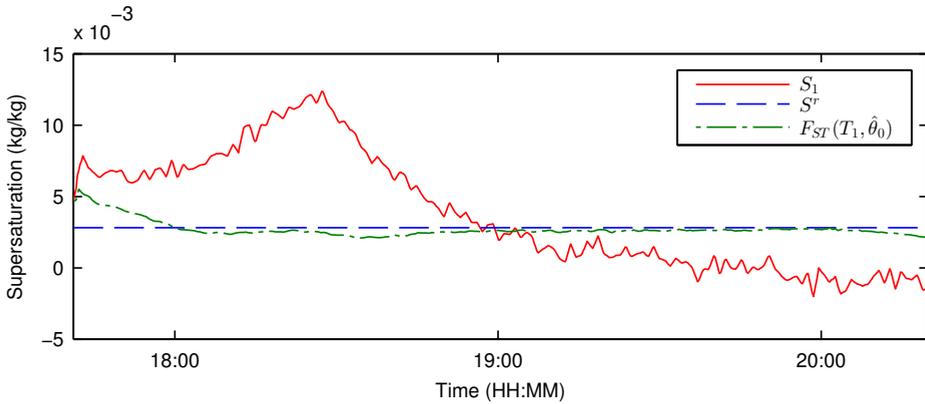
The time profiles of the reference temperature T_4^r , the measured crystallizer temperature T_4 , the temperature model $F_{TTr}(T_4)$ and the supersaturation S_4 for this experiment are reported in the Figure 6.28. The cooling is still very slow at the start of the batch. Owing to the ILC correction, the temperature reference T^r decreases very slowly for an even longer time than in the previous experiment. Subsequently, it decreases even faster for the last part of the experiment.

However, the actual temperature T_4 still cannot decrease as fast as required towards the end of the batch and therefore the supersaturation tracking is lost. The RMS of the supersaturation tracking error for the first 2 hours is now reduced to 0.0013 kg/kg which is a significant improvement with respect to the B2B experi-

ment 3.

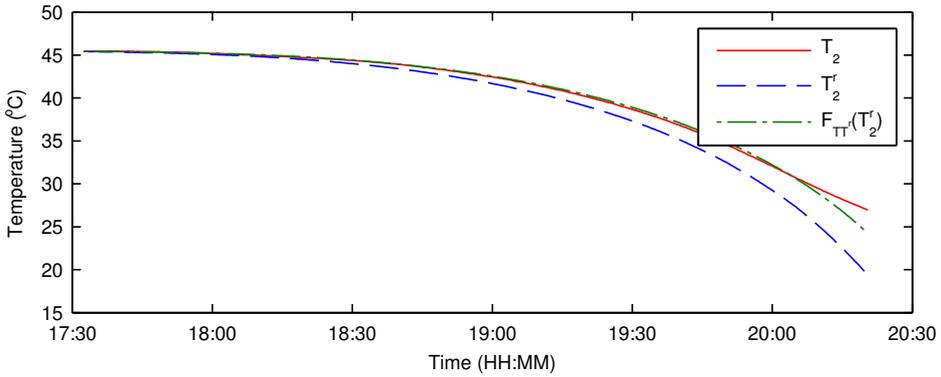


(a) Reference temperature T_1^r , measured crystallizer temperature T_1 and temperature model $F_{TTr}(T_1^r)$. The measured crystallizer temperature T_1 is close to the temperature model $F_{TTr}(T_1^r)$, as desired.

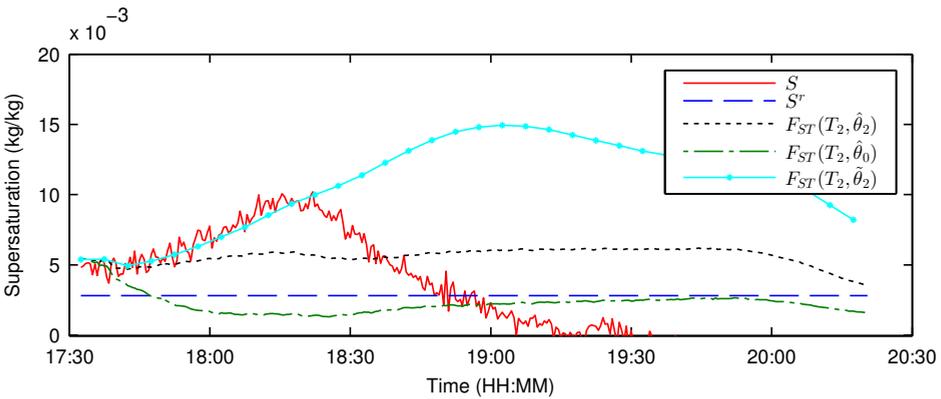


(b) The supersaturation S_1 reaches a maximum value of 0.012 kg/kg at 18:30 and decreases afterwards. The trajectory is rather far from the set-point S^r .

Figure 6.25: Time profiles of the B2B experiment 1.

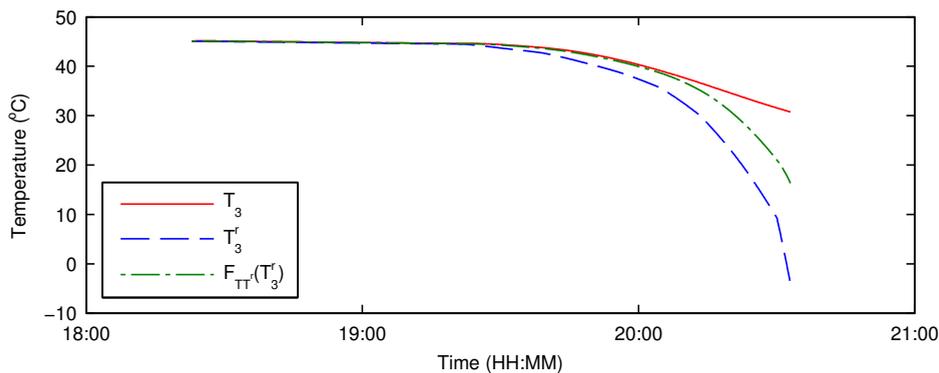


(a) Reference temperature T_2^r , measured temperature T_2 and temperature model $F_{TTr}(T_2^r)$.

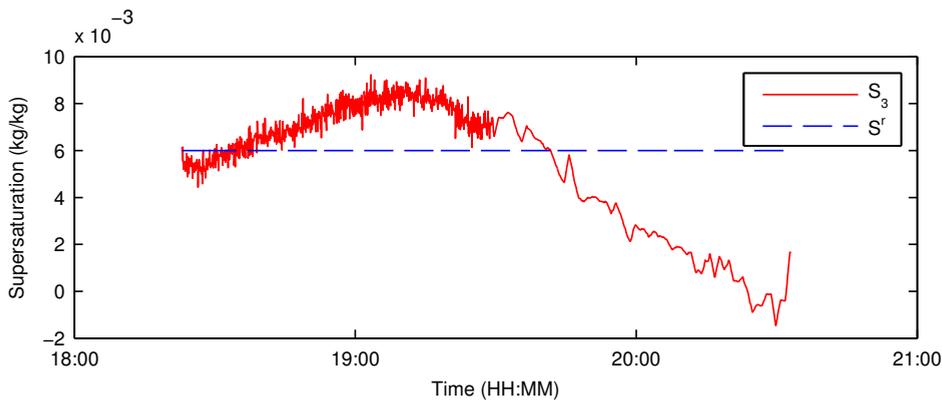


(b) The supersaturation S_2 reaches a maximum of 0.1 kg/kg at 18:20 and decreases afterwards. The tracking performance is similar to the one of B2B experiment 1. None of the supersaturation models can describe the actual supersaturation dynamics with reasonable accuracy.

Figure 6.26: Time profiles of the B2B experiment 2.

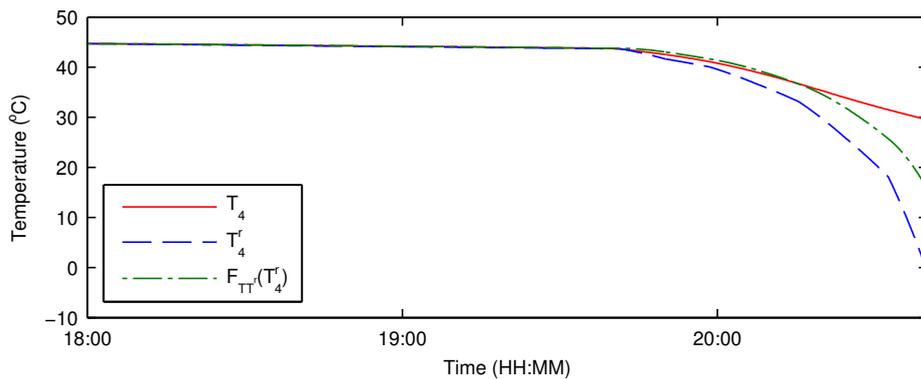


(a) Reference temperature T_3^r , measured temperature T_3 and temperature model $F_{TT}(T_3^r)$.

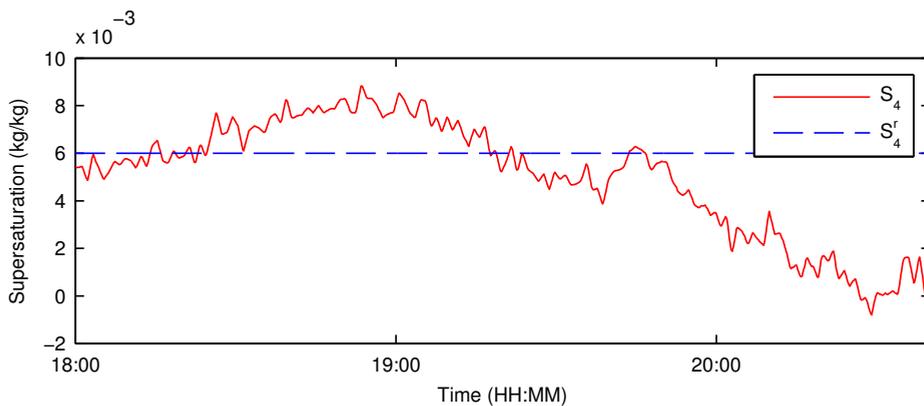


(b) The supersaturation S_3 is relatively close to the set-point S^r for the first part of experiment.

Figure 6.27: Time profiles of the B2B experiment 3.



(a) Reference temperature T_4^r and measured temperature T_4 inside the crystallizer.



(b) The supersaturation S_3 gets closer to the set-point S^r .

Figure 6.28: Time profiles of the B2B experiment 4.

6.6.4 Conclusions for the B2B experiments

The RMSE of the supersaturation tracking for the B2B experiment is reported in Table 6.2.

Experiment	RMSE	S^*
B2B experiment 1	0.0042	0.0028
B2B experiment 2	0.0040	0.0028
B2B experiment 3	0.0025	0.006
B2B experiment 4	0.0013	0.006

Table 6.2: The RMSE of the supersaturation tracking error for the experiments.

The RMSE is similar for the first two experiments where the same input was used. Subsequently, it decreases for the experiment 3 due to the combined IIC+ILC step (and the set-point change) and for experiment 4 due to the ILC step. These results are very encouraging, even though further investigation is required.

First, achieving constant supersaturation seems to be an unfeasible objective for this process towards the end of the batch. The required temperature trajectory in order to obtain constant supersaturation is almost flat at the start of the batch and very steep towards the end. However, a very fast cooling cannot be implemented in practice due to the limitations on the minimum temperature and the cooling power of the thermostatic bath.

Second, we found that there is a severe structural mismatch between the available crystallization model and the actual process dynamics. This limits to a large extent the performance of the IIC model updates. In order to overcome the problem of the structural model mismatch, we combined IIC-type and ILC-type model corrections. However, the ILC algorithm requires the same initial conditions for each batch. From a practical perspective, we found that bringing the system to the same initial condition of temperature, concentration and supersaturation is rather difficult. The initial supersaturation is very sensitive to small changes of the seeding temperature. The temperature dynamics are rather slow and therefore adjusting the temperature with precision requires a long settling time. However, waiting a long time in the metastable region before the seeding is not desirable since it may cause an undesired primary nucleation event. Starting with the same initial concentration is also not straightforward due to evaporation of the water from the crystallizer, which has to be accurately compensated for before each batch.

More experiments would have been required in order to evaluate the use of a different control objective and to further investigate the nature of the model mismatch. However, we could not perform more experiments due to the limited time availability of the experimental set-up.

6.7 Conclusions

In the first part of this experimental campaign we investigated the behavior of the water-succinic crystallization system. The preliminary experiments aimed to build up experience about the process and the instruments in the skid. The work

included the calibration of the K-Patents sensor, the determination of the solubility line, the use of Fumaric acid as an impurity and the design of the seeding procedure. Finally, a well-behaved nominal seeded batch crystallization process suitable for the investigation of the B2B control algorithms was obtained.

In the second part of the experimental campaign, B2B control algorithms were applied in order to achieve a constant supersaturation. The results of these experiments are promising, but further investigation is still required. Different practical and conceptual challenges have to be solved in order to bring similar B2B control strategies to an industrial environment.

First, achieving a constant supersaturation seems to be an unfeasible objective for this process towards the end of the batch since a very fast cooling rate is required. In practice, a very fast cooling cannot be implemented due to limitations on the actuator. Therefore it may be questioned whether a different control objective could be chosen (at least for the last part of the batch).

Second, we found that there is a severe structural mismatch between the available crystallization model and the actual process dynamics. This limits to a large extent the performance of the IIC model updates. Therefore, there is the need for a model structure able to capture the complex growth behavior that we verified in our experiments.

In order to overcome the problem of the structural model mismatch, we combined IIC-type and ILC-type model corrections. However, the ILC algorithm requires the same initial conditions for each batch. From a practical perspective, we found that bringing the system to the same initial condition is rather involved even in a laboratory environment and would hardly be possible to incorporate into standard industrial procedures. In this sense, it would be useful to develop an ILC strategy that is robust to changes in the initial condition.

6.8 Symbols and units

Symbol	Description	Units
a_0	Coefficient 0 solubility	dimensionless
a_1	Coefficient 1 solubility	1/°C
a_2	Coefficient 2 solubility	1/°C ²
C	Concentration of Succinic acid in water	kg/kg
C_s	Solubility of Succinic acid in water	kg/kg
$C_{s,lab}$	Solubility of Succinic acid in water based on laboratory data	kg/kg
$C_{s,exp}$	Solubility of Succinic acid in water based on experiments	kg/kg
g	Growth exponent coefficient	dimensionless
k_g	Growth base coefficient	m/min
K_o	Absolute temperature corresponding to 0 °C	K
n_D	Refractive index	dimensionless
R	Universal gas constant	J/(mol °C)
S	Supersaturation	kg/kg
S^r	Reference supersaturation	kg/kg
T	Crystallizer temperature	°C
T^r	Crystallizer reference temperature	°C
T_J	Jacket temperature	°C
T_J^r	Jacket reference temperature	°C
x_s	Solubility of Succinic acid in water	mol/mol
ΔH_{sol}	Heat of dissolution of Succinic acid in water	J/mol

Table 6.3: Symbols and units used in this chapter.

Conclusions

In this chapter, the research objective and the most important results achieved in this thesis are revisited. This helps us to draw conclusions and to identify further objectives and opportunities for future research.

7.1 Contribution of this thesis

The primary objective of this thesis was to develop strategies to improve the performance of model-based control for batch processes using data measured from previous batches. We have given particular focus to a specific control objective for a specific batch process, namely the tracking of a supersaturation set-point for a batch cooling crystallization process.

Data-based modeling of batch processes is hindered by the nonlinear dynamic behavior and the possible presence of significant structural mismatches with respect to the first-principles models. In this thesis, we have developed the parametric model update strategy Iterative Identification Control and the nonparametric model update strategy Iterative Learning Control for the batch cooling crystallization process. We have applied these techniques to a pilot-scale crystallizer in a campaign of experiments. Furthermore, we have investigated the opportunity to ameliorate the performance of parametric strategies similar to Iterative Identification Control using excitation signals.

The main contributions of this thesis are here summarized in the following:

1. *Development and experimental validation of the batch-to-batch model improvement approaches Iterative Identification Control and Iterative Learning Control for model-based control of batch cooling crystallization.* We have developed in Chapter 3 two strategies for batch-to-batch model improvement in a batch cooling crystallization process, namely Iterative Identification Control (IIC) and Iterative Learning Control (ILC). Both algorithms update a model representing the uncertain process dynamics based on the measurement collected from the previous batches. Subsequently, they use the updated model to determine the temperature reference profile for the next batch in order to track the desired supersaturation set-point. This temperature reference profile is

given to a lower-level temperature controller whose role is to reduce the effect of possible disturbances affecting the temperature dynamics.

The nature of the model update is different in the two algorithms. In IIC, the physical parameters of the process model are iteratively estimated using the measured data in such a way that their accuracy increases from batch to batch. In ILC, the process model is adjusted by an additive, nonparametric correction term which depends on the difference between the measured output and the model output in the previous batch.

We have shown in a simulation study that the two approaches have complementary advantages and disadvantages. On the one hand, IIC provides the best performance (measured in terms of the root mean square of the tracking error) when the assumed model structure can actually describe the data-generating system. However, the performance of IIC is hard to predict (and generally lower) when the data-generating system is not contained in the assumed model structure, i.e. in the case of a structural model mismatches. On the other hand, ILC is more robust to structural model mismatches. Even though these mismatches slow down the learning process, a satisfactory result is eventually obtained after a number of batches.

We have demonstrated the applicability of the two strategies performing experiments on a pilot-scale crystallization setup. The experimental results are presented in Chapter 6. These experiments confirm the potential of the IIC and ILC strategies for the batch-to-batch improvement of model-based control in batch cooling crystallization.

2. *Investigation of the use of excitation in an iterative identification/controller design scheme.* Inspired by the IIC approach, we have investigated the use of excitation signals in an iterative identification/controller design scheme. In the framework developed in Chapter 4, the total time of operation of the model-based control system is divided into a number of learning intervals. For the case of batch systems, the learning intervals correspond to the different batch runs. After an interval, the measured data are used to refine the estimate of the model parameters, and a new controller is designed. The controller will be applied in the next interval, and so on and so forth for the following intervals. Excitation signals can be superposed to the normal control input in all the intervals. On the one hand, applying an excitation signal during an interval leads to a performance degradation during the current interval, since it acts as an disturbance on the control system. On the other hand, the informative dataset obtained owing to the excitation signal can be used to identify a more accurate model, and thus improve the performance for the following intervals. The problem considered is to design the excitation signal in order to maximize the overall performance taking this dual effect of the excitation signals explicitly into account.

For the case of linear dynamical systems, we have developed a general, tractable solution to the optimal design of the excitation signals in the iterative identification/controller design framework. Our solution uses methodologies from the field of Identification for Control and standard experiment

design tools for linear systems. We have shown in a simulation example that the design of the excitation signal in the iterative identification/controller design framework guarantees a superior overall performance than the one performed in the classic two-phases approaches documented in the literature.

3. *Development of experiment design tools for nonlinear systems.* A limitation of the approach discussed in the previous point is the restriction to linear dynamical systems. This restriction was necessary since generally applicable and computationally efficient experiment design tools for nonlinear dynamical systems are not available to date. Experiment Design for nonlinear dynamical systems is still a very open and challenging research area. The methods suggested in the literature are either computationally expensive, or only applicable for very specific classes of nonlinear systems. Motivated by the limitations of the state-of-the-art methods, we conducted research towards the development of novel Experiment Design tools applicable to a fairly wide class of nonlinear systems, but still relying on computationally efficient convex optimization routines.

The method that we devised is presented in Chapter 5 of this thesis. We restrict our attention to multilevel excitation signals, i.e. signals which admit a finite number of possible levels. A multilevel excitation signal can be described by the sequence of the levels appearing in it. Within this sequence, we recognize a number of shorter subsequences. Under certain conditions, the information matrix for the full sequence is proportional to the contribution due to each subsequence, times the frequency at which the subsequence appears in the full sequence. Owing to the linear relation between the information matrix and these frequencies, we are able to formulate a convex experiment design problem using the frequencies as optimization variables. The convex problem can be solved efficiently using standard software and algorithms.

The applicability of our method is demonstrated in a simulation study using the model of a first-order, irreversible reaction system. We have shown that the optimal multilevel excitation signal found using our approach for this system has two distinct regimes. This result is very interesting in relation to Experiment Design problems for linear systems, where on the contrary the optimal excitation signal can always be found in the class of the stationary signals. Furthermore, we have shown that the design based on the optimal multilevel excitation signal outperforms the one based on random binary signals, which is a common choice in engineering practice.

7.2 Recommendations for future research

In light of the research objective of the thesis and the results achieved, we have identified the following opportunities for future research:

1. *Modeling of the batch cooling crystallization process.* In our experiments, we have verified that there is a severe, structural mismatch between our crys-

tallization model and the actual process dynamics. Moreover, the presence of impurities (which are common in industrial environments) modifies the process dynamics dramatically (see Chapter 6). For this reason, the model of a batch cooling crystallization process should be maintained during the whole life-time of the operation. Unfortunately, the presence of structural model mismatches limits to a large extent the applicability of parametric model learning approaches such as IIC.

The development of more accurate model structures describing the batch cooling crystallization process is of paramount importance for the improvement of the related model-based control techniques. These model structures need to be validated both for pure crystallization systems and in the presence of impurities, since the dynamics are substantially different in the two cases. For the model development and validation, the availability of additional process measurements such as the CSD would be of great help.

2. *Control objective for the batch cooling crystallization process.* In our experiments, we have fixed as control objective the tracking of a constant supersaturation set-point. However, we observed that this objective does not seem to be feasible for the experimental set-up at hand towards the end of the process (see Chapter 6). Therefore, it may be questioned whether a different control objective should be selected.

Defining the control objective directly in terms of the CSD would probably be the most effective solution, since the product requirements are also often specified in the terms of this quantity. However, direct control of the CSD also requires the availability of adequate measurements of this process variable.

3. *Effect of structural model mismatches on the parametric model learning approach.* The properties of a parametric model learning approach such as IIC are well understood (and optimal) only when the assumed model structure contains the data-generating system. However, this is an ideal situation that never really occurs in practice. In fact, all model structures used in engineering are either obtained under some simplifying hypotheses and assumptions (in the case of physical modeling), or are simply general approximation of dynamical systems (in the case of black-box modeling). For batch cooling crystallization in particular, we have observed in this thesis that the structural model mismatch is a serious issue. Since the structural model mismatches cannot completely be avoided, it would be important to quantify their impact on the model accuracy, and thus on the control performance given by a model-based controller designed using an identified model.

In the case of structural model mismatches, the identified model converges to a system that is the best approximation (with respect to the particular identification dataset and identification criterion) of the data-generating system within the model structure. To date, the properties of the best approximating model are well understood when both the assumed model structure and the data-generating system are linear in the input (Ljung, 1999). In this case, the best approximating model is characterized by frequency-domain

integral expressions involving the spectrum of the input signal. Some results also exist for the situation where the data-generating system is nonlinear, but the assumed model structure is linear. The properties of the best approximating linear model of a nonlinear system have been discussed in Schoukens et al. (2005). However, in a full nonlinear context, it is very hard to analyze the properties for the best approximating model, and thus the effect of a structural model mismatch on the performance of a model-based controller based on an identified model. Due to the complexity of the required analysis, it would be sensible to first tackle it considering very simple nonlinear structures such as low-order, nonlinear FIR. A formal result in this direction would still be a novel contribution to the field of Systems and Control.

4. *Combination of parametric and nonparametric learning approaches.* The use of a parametric and a nonparametric learning approach has been shown to have complementary advantages (see Chapter 3). On the one hand, the parametric approach allows for a faster learning, since it produces a parsimonious representation of the system. Furthermore, a physically parametrized model retains all the physical insight. On the other hand, the nonparametric approach can cope effectively with the serious issue of structural mismatches owing to the use of a more flexible representation. A natural question is whether it is possible to combine the two approaches in a convenient way in order to benefit from the advantages of both worlds.

Different black-box model structures where the whole dynamics is represented adopting very general parameterizations, or in a completely nonparametric way have been presented in the literature (Ljung et al., 2011; Sjöberg et al., 1995). In principle, these structures can approximate wide classes of nonlinear dynamical systems with arbitrary precision. It may look tempting to use these structures to model batch processes, since the first-principles approach often leads to severe structural mismatch issues. However, in our opinion, a fully nonparametric, black-box approach has limited possibility of success in the field of process engineering. Learning a fully nonparametric model requires huge amounts of data, since all the a priori knowledge is disregarded and the model is searched in a vast search space. Furthermore, nonparametric, black-box models would not be easily accepted by process experts, who are often interested in the physical interpretation of their models, beside of their prediction capability.

In the systems usually encountered in the field of process engineering, there are parts such as mass and energy balances that are very well understood. For these parts, the measured data should only be used to estimate (if any) the uncertain constants of their first-principles representation. Other parts, such as the growth and the birth dynamics in the crystallization process, are subject to more severe uncertainties. In order to model the behavior of these parts with the required accuracy, a non-parametric approach may be really required.

A convenient representation for a process system could be an interconnection of physically parametrized blocks representing the behavior of the first

parts, and nonparametric blocks representing the behavior of the latter. Such representation would exploit as much as possible the a priori knowledge and retain the physical insight, while using the flexibility of the non-parametric approach in order to describe the most uncertain parts. The difficulty in identifying such a system is that the different blocks interact with each other and all contribute to the observed input/output behavior of the system. So far, similar identification problems have been considered for systems described as the interconnection of linear dynamics with known structure, and static nonlinearities represented in a non-parametric way (see Hsu et al. (2008)). It would be interesting to investigate whether a similar modeling framework could be useful to describe process systems.

5. *Experiment design for nonlinear systems.* In this thesis, we have presented a novel experiment design method for nonlinear systems based on multilevel excitation signals. The merit of our method in comparison with similar ones that have recently appeared in the literature is its applicability to a wider class of nonlinear systems.

However, several questions related to our method were not answered in this thesis (see the conclusions of Chapter 5). In particular, we have shown that the optimal multilevel excitation signal for the first-order, irreversible reaction system considered in the numerical example is composed of two distinct regimes. This result is very interesting in relation with the experiment design problems involving linear systems, where on the contrary it is known that the optimal excitation signal can always be found in the class of the stationary signals. Solutions having multiple regimes were not even found in the previous contributions on experiment design for nonlinear systems, in which much simpler nonlinear model structures were examined. We would like to understand whether this property is really induced by the nonlinearity of the system, or it is just due to the particular experiment design method that we have devised, which allows this kind of solutions.

A possibility to address this question could be to study the experiment design problem for classes of nonlinear systems that still have significant descriptive capability, but present some characteristic that ease a formal analysis. Good candidates for this study could be LPV or nonlinear ARX model structures.

A

APPENDIX

Applied Dynamic Optimization

Dynamic Optimization Problems occur in several fields of science and engineering. In process engineering they are used for instance for control, parameter estimation, reactor design and optimization, optimal start-up and shut-down, and scheduling (Biegler, 2010). In this appendix, we briefly present three approaches for the solution of Dynamic Optimization Problems. The approaches considered are the Pontryagin Maximum Principle, the simultaneous optimization approach, and the sequential optimization approach. We describe a possible implementation of these methods using the software environments Matlab and GAMS.

In this appendix, we briefly present three approaches for the solution of Dynamic Optimization Problems (DOP). For the sake of concreteness, we apply these approaches to a specific DOP:

Problem A.1 (Dynamic Optimization Problem)

$$\min_u \overbrace{\int_0^{t_f} u^2(t) dt}^E \quad (\text{A.1})$$

such that

$$\begin{aligned} \dot{x} &= ax + bu \\ x(0) &= x_0 \\ x(t_f) &= x_f. \end{aligned} \quad (\text{A.2})$$

In this problem, we look for an optimal input $u^o(t)$, $t \in [0, t_f]$ that steers the system from the initial state x_0 to the final state x_f in the time $[0, t_f]$ minimizing the total energy E of the input signal. We fix in this problems the numerical quantities, $t_f = 1$, $x_0 = 1$, $x_f = 5$, $a = -2$, and $b = 1$. Note that the DOP at stake is infinite dimensional since the input $u(t)$ is a function of the continuous time. For this reason, generic optimization strategies cannot be directly applied.

In this appendix, we obtain a solution of this DOP using three different approaches, namely the Pontryagin Maximum Principle, the simultaneous optimization approach, and the sequential optimization approach. We here focus on the implementation aspects of these approaches using the software environments Matlab and GAMS. Our implementation strategies are inspired from (Huesman, 2003, 2005). For a more general presentation of the methods, see also (Cervantes and Biegler, 1999).

1. *Pontryagin Maximum Principle (PMP)*. The first-order necessary conditions for optimality are obtained using the PMP principle. This results in a two-point boundary value problem which can be solved using different techniques.
2. *Simultaneous optimization approach*. The input and states trajectories are discretized altogether by applying an ODE integration scheme. By doing this, the dynamic model equations of the system are approximated with a set of equality constraints and the original DOP is transformed into a finite-dimensional (approximated) optimization problem. The latter can be solved using standard optimization software.
3. *Sequential optimization approach*. A finite-dimensional parametrization of the input signal is first selected. For a certain parameter describing an input signal, the objective function and the constraints of the problem can be evaluated by integrating the system equations with the help of an ODE solver. In the sequential approach, the DOP is solved restricting the input signal to the space of signal that can be described by the parametrization. The optimization routine “calls” the ODE solver in order to evaluate the objective function and the constraints for different parameters describing different input signals.

A.1 Pontryagin Maximum Principle

In this section, we obtain the solution of the DOP (A.1) using the PMP approach. As mentioned before, the PMP approach leads to a two-point BVP problem which can be solved in general using different techniques. For the particular DOP (A.1), the resulting BVP can be solved analytically.¹ This allows us to verify the correct implementation of the other two methods, which conversely always provide an approximated numerical solution. The analytical solution obtained using the PMP approach is useful in order to verify the correct implementation of the two other approaches, which instead always provide an approximated numerical solution.

First, we write the Hamiltonian of the DOP

$$H = u^2 + \lambda_1(ax + bu).$$

Minimizing the Hamiltonian with respect to u , we have for the optimal input u^o the condition

$$\frac{\partial H}{\partial u} = 0 \Rightarrow u^o = -\frac{\lambda_1 b}{2}. \quad (\text{A.3})$$

¹Note that this is not in general the case and is due to the very simple structure of the DOP at stake.

In order to determine u^o , we have to solve the following two-point Boundary Value Problem (BVP):

$$\begin{aligned} \dot{x}^o &= ax^o + bu^o \\ \dot{\lambda}_1^o &= -\frac{\partial H}{\partial x} = -\lambda_1^o a \\ x^o(0) &= x_0 \\ x^o(t_f) &= x_f. \end{aligned} \tag{A.4}$$

Solving the differential equation for λ_1 , we have

$$\lambda_1(t) = c_1 e^{-at}$$

where c_1 is a generic real-valued constant. Using (A.3), we can write the optimal input signal $u^o(t)$ as

$$u^o(t) = -\frac{bc_1}{2} e^{-at}.$$

Thus, we can simplify the two-point BVP (A.4) to another BVP that involves the state variables x only:

$$\dot{x}^o = ax^o - \frac{b^2 c_1}{2} e^{-at} \tag{A.5}$$

$$x^o(0) = x_0 \tag{A.6}$$

$$x^o(t_f) = x_f. \tag{A.7}$$

Let us first assume that $a \neq 0$. In this case, the general integral of the differential equation A.5 is

$$x^o(t) = c_2 e^{at} + \frac{b^2 c_1}{4a} e^{-at}.$$

We find the actual values of c_1 and c_2 by requiring that the state x^o has the desired initial and final values x_0 and x_f :

$$x^o(0) = c_2 e^0 + \frac{b^2 c_1}{4a} = x_0 \tag{A.8}$$

$$x^o(t_f) = c_2 e^{at_f} + \frac{b^2 c_1}{4a} e^{-at_f} = x_f.$$

The last equation can be written as

$$\begin{bmatrix} \frac{b^2}{4a} & 1 \\ \frac{b^2}{4a} e^{-at_f} & e^{at_f} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_f \end{bmatrix}. \tag{A.9}$$

Since we assumed $a \neq 0$, the matrix is invertible and the problem has a unique solution for any $t_f \neq 0$.

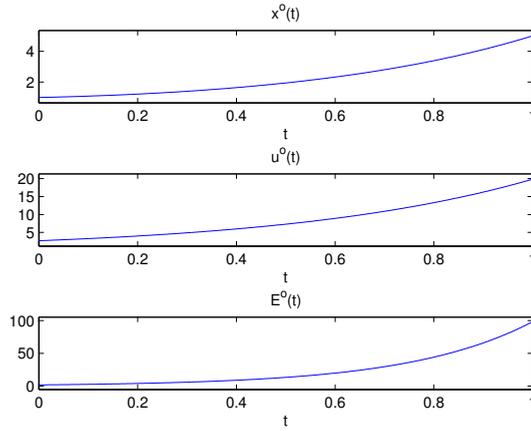


Figure A.1: Optimal input $u^o(t)$, state $x^o(t)$, and energy $E^o(t)$ obtained using the PMP principle and solving the associated BVP analytically.

If on the contrary $a = 0$, the resulting BPV reduces to:

$$\dot{x}^o = -\frac{b^2 c_1}{2} \quad (\text{A.10})$$

$$x^o(0) = x_0 \quad (\text{A.11})$$

$$x^o(t_f) = x_f. \quad (\text{A.12})$$

The general integral of (A.10) is

$$x^o(t) = c_2 - \frac{b^2 c_1}{2a} t. \quad (\text{A.13})$$

In this case, c_1 and c_2 are given by the solution of the linear system

$$\begin{bmatrix} 0 & 1 \\ -\frac{b^2 c_1}{2} t_f & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_f \end{bmatrix}.$$

For the numerical values of the problem $t_f = 1$, $x_0 = 1$, $x_f = 5$, $a = -2$, and $b = 1$, we find using (A.9) that $c_1 = -5.3652$ and $c_2 = 0.3294$. The optimal input $u^o(t)$, state $x^o(t)$, and energy $E^o(t)$ are shown in Figure A.1. The Matlab code used to obtain and plot the solution is reported below.

Listing A.1: 'Exact solution using the PMP principle'

```
addpath('export_fig');
```

```
%% PROBLEM PARAMETERS %%
```

```
a=-2;
b=1;
x0=1;
xf=5;
```

```

tf=1;

%% EXACT SOLUTION – MAXIMUM PRINCIPLE %%
syms C2 C1 t;

% exact solution x = dsolve('Dx = a*x -b^2*c/2*exp(-a*t)', 't');
x=C2*exp(a*t) +(b^2*C1)/(4*a)*exp(-a*t);

% Boundary problem solution %
A = [b^2/(4*a) 1 ; b^2/(4*a)*exp(-a*tf) exp(a*tf)];
B = [x0 xf]';
P = linsolve(A,B);
C1 = P(1);
C2 = P(2);

syms l1 u t E;
l1 = C1*exp(-a*t); %lambda1
u = -l1*b/2; %u
x=subs(x,{ 'C2' , 'C1' },{C2,C1}); %x
E = u*u;

f=figure(1);
set(f, 'color', 'w');
h=subplot(3,1,1);
ezplot(x,[0 1],f);
title(h, 'x^o(t)');

h=subplot(3,1,2);
ezplot(u,[0 1]);
title(h, 'u^o(t)');

h=subplot(3,1,3);
ezplot(int(E), [0 1]);
title(h, 'E^o(t)');

export_fig('analytical_solution', '-pdf');

```

A.2 Numerical solution using the simultaneous optimization approach

In the simultaneous approach, the DOP is first discretized by applying an ODE integration scheme. By doing this, the dynamic model equations are transformed into a set of equality constraints and the original problem is transformed into a finite-dimensional optimization problem, which can be solved using standard optimization software.

In this example, we have selected the Explicit Euler (EE) integration scheme. The problem A.1 discretized using the EE integration scheme becomes

Problem A.2 (Dynamic Optimization Problem - simultaneous approach)

$$\min_u \sum_{i=1}^N u(i-1)^2 \quad (\text{A.14})$$

such that

$$\begin{aligned}x(i) &= \Delta T(ax(i-1) + bu(i-1)) \quad \forall i \in [1 \dots N] \\x(0) &= x_0 \\x(N) &= x_f\end{aligned}\tag{A.15}$$

with

$$\Delta T = \frac{t_f}{N}.\tag{A.16}$$

Note that the system equations and the input vector are discretized altogether. This is the reason why this method is called simultaneous. The simultaneous approach leads to a problem with a large number of equality constraints (in our case $2N - 2$) and optimization variables (in our case $3N - 3$);

Two software implementations of the problem are here proposed. The first one is based on Matlab and the function `fmincon`, the second one is based on the optimization modeling software GAMS.

A.2.1 Matlab Implementation

The Matlab implementation is based on the files `DOsim1.m`, `objsim1.m`, and `consim1.m`.

Listing A.2: 'DOsim1.m'

```
clear;
global con nx;
nx = 50; % Number of discretization points

%% Optimization options %%
Options = optimset;
Options.Display = 'iter';
Options.MaxSQPIter = 200;
Options.MaxIter = 4000;
Options.TolFun = 1e-4;
Options.MaxFunEvals = 4e5;

x0 = 2;
xf = 5;
f0 = 0;
tf = 1;
a = -2;
b = 1;
l = -a*xf/b;

%% Start value decision variables %%
px = 0.9*xf*ones((nx-2),1);
pe = 0*ones(nx-1,1);
pu = 1*ones(nx,1);
p = [px; pe; pu];

%% Bounds on the decision variables %%
lb = [];
ub = [];
```

```

%% Call optimization routine %%
[pm,fval,exitflag,output] = fmincon('objsim1',p,[],[],[],[],lb,ub,'consim1',Options);

```

Listing A.3: 'objsim1'

```

function obj = objsim1(p)
global con nx

%% Set-up decision variables %%
x0 = 2;
xf = 5;
a = -2;
b = 1;
t0 = 0;
tf = 1;

x = zeros(nx,1);
e = zeros(nx,1);
u = zeros(nx,1);

x(1) = x0;
x(2:nx-1) = p(1:nx-2);
x(end) = xf;

e(1) = 0;
e(2:nx) = p((nx-1):(2*nx-3));

u = p((2*nx-2):end);

%% Objective Function %%
obj = e(end);

%% Constraints %%
h = (tf-t0)/(nx-1);
for i = 1:(nx-1)
    conX(i) = (x(i+1)-x(i))/(h) - (a*x(i) + b*u(i));
    conE(i) = (e(i+1) - e(i))/(h) - u(i)*u(i);
end

con = [conX conE];
assignin('base','conX',conX);
assignin('base','conE',conE);

```

Listing A.4: 'consim1.m'

```

function [conin,coneq] = consim1(p)
global con Np
conin = [];
coneq = con;

```

A.2.2 GAMS Implementation

The GAMS implementation is based on the files `simplest_rocket.gms`. A Matlab interface used to set up the problem variables and export the result of the GAMS optimization into Matlab is implemented in the file `simplest_rocket.m`.

Listing A.5: 'simplest_rocket.gms'

```

$set matout "'rocket_output.gdx', t, x, u, E ";

$if      set n $set nh %n%
$if not set nh $set nh 300

Set h intervals /h0*h%nh%/

Scalars
    a      /-2/
    b      /1/
    nh     /%nh%/
    x0     /2/
    xf     /5/
    tf     /1/
    umax   /1000/
    step   step size ;

Parameters
    t(h)   time;

$if exist matdata.gms $include matdata.gms

Variable EF total energy consumed;

Positive variables
    x(h)   position at time h
    u(h)   input at time h
    E(h)   energy consumed at time h;

Equations
    x_eqn(h)
    E_eqn(h)
    obj;

step = tf/nh;
t(h) = (ord(h))*step;

obj.. EF =e= E('h%nh%');

x_eqn(h-1).. x(h) =e= x(h-1) + step*a*x(h-1) + step*b*u(h-1);
E_eqn(h-1).. E(h) =e= E(h-1) + step*u(h-1)*u(h-1);

*x.lo(h) = x0;
*x.up(h) = xf;
u.up(h) = umax;
u.lo(h) = -umax;
x.fx(h)$(ord(h) = 1) =x0 ;
* BUGFIX even x(h-1) = xf for euler integration ...;
*x.fx(h)$(ord(h) = card(h)-2) = xf ;
*x.fx(h)$(ord(h) = card(h)-1) = xf ;

```

```

x.fx(h)$(ord(h) = card(h)) = xf ;
* Initial guess for u: steady state xf
*u.l(h) = -a*xf/b;

model simplest_rocket /all/;
solve simplest_rocket using nlp minimizing EF;
*display x.l, u.l;
*display c;
execute_unload %matout%;

```

Listing A.6: 'simplest_rocket.gms'

```

gamso.form='full';
gamso.input='exec';

a.val = -2;
a.type = 'parameter';
a.name = 'a';

b.val = 1;
b.type = 'parameter';
b.name = 'b';

x0.val = 1.00;
x0.type = 'parameter';
x0.name = 'x0';

xf.val = 5.00;
xf.type = 'parameter';
xf.name = 'xf';

tf.val = 1.00;
tf.type = 'parameter';
tf.name = 'tf';

%x0.load = 'replace';

um.val = 120;
um.type = 'parameter';
um.name = 'umax';
%um.load = 'replace';

[ tg , xg , ug , Eg ] = gams('simplest_rocket', x0, um, a, b, tf);

figure(1);
subplot(3,1,1);
% position
plot(tg.val, xg.val, 'r');
subplot(3,1,2);
% fuel
plot(tg.val, ug.val, 'g');
subplot(3,1,3);
% energy consumption
plot(tg.val, Eg.val, 'b');

```

Some considerations:

- The initial guess for u is set to the equilibrium value for x_f

- Different integration schemes are also possible.
- The integration step ΔT has to be chosen wisely

A.3 Numerical solution using the sequential optimization approach

As explained before, in the sequential approach a finite-dimensional parametrization of the input signal is first selected. In our case, the input sequence is parametrized as a piecewise linear function interpolating $n_p = 11$ points. Thus, the to-be-optimized input parameter is a quantity $p \in \mathbb{R}^{11}$.

For a certain parameter p describing an input signal u , the objective function and the constraints of the DOP can be (approximately) evaluated by integrating the system equations with the help of an ODE solver. The system equations are here integrated using the EE integration scheme fixed step $\Delta T = \frac{t_f - t_0}{N}$.

The Matlab function `fmincon` is used to optimize the parameter p in order to minimize the energy E and to satisfy the constraint $x(t_f) = x_f$.² The optimization problem solved in the sequential approach is the following:

Problem A.3 (Dynamic Optimization Problem - sequential approach)

$$\begin{aligned} p^o &= \arg \min_p E(p) && \text{s.t.} \\ x(t_f) &= x_f(p) \end{aligned} \tag{A.17}$$

where $E(p)$ and $x_f(p)$ are computed by integrating the state equation using the EE scheme.

A.3.1 Matlab Implementation

The Matlab implementation is based on the files `DOsim.m`, `objsim.m`, and `consim.m`.

Listing A.7: 'DOsim.m'

```

global con;
Options = optimset;
Options.Display = 'iter';
Options.MaxSQPIter = 200;
Options.TolFun = 1e-6;
Options.MaxFunEvals = 2500;

np = 11;

x0 = 2;
xf = 5;
a = -2;
b = 1;
l = -a*xf/b;

```

² Note also that the initial condition $x(0) = x_0$ can be satisfied by construction setting the initial state of the simulation in the ODE solver to x_0 .

```

p = 1*ones(1,np);
lb = [];% -3000.*ones(1,np);
ub = [];% 3000.*ones(1,np);
[pm,fval,exitflag,output] = fmincon('objsim',p,[],[],[],[],lb,ub,'consim',
    Options);

```

Listing A.8: 'objsim.m'

```

function obj = objsim(p)
global con;
% Discretisation, initialisation and parameters
t(1) = 0;
tf = 1;
x0 = 2;
xf = 5;
a = -2;
b = 1;
np = length(p);
dt = linspace(t(1), tf, np);
du = p;
N = 300;
h = (tf - t(1))/N;
x(1) = x0;
E(1) = 0;
uu(1) = 0;
% Euler
for i = 2:N+1
t(i) = t(1) + h*(i-1);
u = interp1(dt,du,t(i));
uu(i) = u;
x(i) = x(i-1) + h*(a*x(i-1) + b*u);
E(i) = E(i-1) + h*(u*u);
end
% States
obj = E(N+1);
% Constraints
con = x(N+1)-xf;

```

Listing A.9: 'objsim.m'

```

function [conin, coneq] = consim(p)
conin = [];
coneq = con;

```


B

APPENDIX

Numerical solution of Partial Differential Equations

We describe in this appendix the finite-volume discretization scheme used in this thesis to obtain numerical solution for the Partial Differential Equation used in the modeling of the batch crystallization process as the Population Balance Equation.

B.1 Problem Formulation

Consider the one-dimensional Partial Differential Equation (PDE) integration problem

Problem B.1 (PDE integration problem)

$$\frac{\partial n(L, t)}{\partial t} + \frac{\partial f(n(L, t))}{\partial L} = 0, \quad (\text{B.1})$$

$$n(t, 0) = n_0(L) \quad (\text{B.2})$$

$$n(0, t) = h(t) \quad (\text{B.3})$$

with $L \in [0, \infty]$, $t \in [0, \infty]$, and $f \geq 0$.

The PDE (B.1) is complemented by an initial condition (B.2) and a boundary condition (B.3). The solution of the PDE integration problem B.1 is a function $n(L, t)$. Note that this PDE is used in this thesis as to describe the amount of crystals of different lengths in the batch cooling crystallization process adopting the Population Balance modeling framework (Ramkrishna, 2000).

B.2 Finite volume discretization scheme

In order to obtain a numerical solution to the problem B.1, we adopt a finite-volume discretization scheme. We first define the points

$$L_i \triangleq i\Delta L$$

and divide the spatial domain $[0, L_{\max}]$ into a finite number of *volumes* $v_i, i = 1, 2, \dots, N$ having equal size $\Delta L = L_{\max}/N$. Each volume v_i is centered around the length L_i and spans the interval $[L_{i-1/2}, L_{i+1/2}]$.

For a certain volume v_i , we can write the average value $n_i(t_1)$ of the function $n(L, t)$ at the time t_1 inside the volume as

$$n_i(t_1) = \frac{1}{\Delta L} \int_{v_i} n(L, t_1) dL. \quad (\text{B.4})$$

For the same volume v_i , at a different time instant t_2 , we have

$$n_i(t_2) = \frac{1}{\Delta L} \int_{v_i} n(L, t_2) dL. \quad (\text{B.5})$$

Integrating the expression (B.1) in time, we get

$$n(L, t_2) = n(L, t_1) - \int_{t_0}^{t_f} \frac{\partial f}{\partial L} dt. \quad (\text{B.6})$$

Thus, the Equation (B.5) can be written as:

$$\begin{aligned} n_i(t_2) &= \frac{1}{\Delta L_i} \int_{c_i} \left[n(L, t_1) - \int_{t_0}^{t_f} \frac{\partial f}{\partial L} dt \right] dL = \\ &= n_i(t_1) - \frac{1}{\Delta L_i} \int_{t_0}^{t_f} \overbrace{f(n(L_{i+1/2}), t)}^{f_i^+} - \overbrace{f(n(L_{i-1/2}), t)}^{f_i^-} dL. \end{aligned} \quad (\text{B.7})$$

Differentiating with respect to time, we get semi-discrete equations in the form:

$$\frac{dn_i}{dt} = -\frac{1}{\Delta L_i} [f_i^+ - f_i^-] \quad (\text{B.8})$$

which in principle can be solved numerically by applying an ODE integration scheme.

Note that Equation (B.8) holds exactly. However, the terms f_i^+ and f_i^- cannot be evaluated exactly whenever the function f depends on the solution $n(L, t)$. In these cases, f_i^+ and f_i^- have to be approximated using a finite-volume approximation scheme.

B.3 First-order upwind approximation scheme

The simplest finite-volume approximation scheme is the first-order upwind. Let us introduce for convenience the notation $f_i \triangleq f(n(L_i, t))$. In the first-order upwind scheme, the terms f_i^+ and f_i^- are approximated as

$$\begin{aligned} f_i^+ &= f_i \\ f_i^- &= f_{i-1} = f_{i-1}^+ \end{aligned}$$

The first-order upwind scheme is given by:

$$\begin{aligned} \frac{dn_i(t)}{dt} &= -\frac{1}{\Delta L_i} (f_i - f_{i-1}), \quad i = 1, 2, \dots, N. \\ n_0(t) &= h(t) \\ n_i(0) &= \int_{v_i} n_0(L) dL \end{aligned} \tag{B.9}$$

Thus, using the first-order integration scheme, a set of ODEs can be generated. Subsequently, the set of ODE can be integrated in time using an ODE solver, providing a numerical solution to the problem B.1.

B.4 Second-order approximation schemes

The accuracy of the first-order upwind approximation scheme may be limited due to numerical diffusion. In order to overcome this limitation, we used in this thesis a second-order approximation scheme.

A *family* of second-order approximation schemes may be defined as

$$f_i^+ = \left(f_i + \frac{1+\kappa}{4} (f_{i+1} - f_i) + \frac{1-\kappa}{4} (f_i - f_{i-1}) \right), \tag{B.10}$$

$$f_i^- = f_{i-1}^+, \tag{B.11}$$

$$\kappa \in [-1, 1]. \tag{B.12}$$

For $\kappa = -1$, one gets the second-order accurate fully one-sided upwind scheme, while for $\kappa = 1$, one gets the standard second-order accurate central scheme. For all other values of $\kappa \in [-1, 1]$, a weighted blend between the two schemes is obtained. In this thesis, we used a second-order approximation scheme with $\kappa = 1/3$ as in Koren (1993). Substituting $\kappa = 1/3$ in (B.10), we get

$$f_i^+ = f_i + \frac{1}{3} (f_{i+1} - f_i) + \frac{1}{6} (f_i - f_{i-1}). \tag{B.13}$$

B.4.1 Flux limiter function

In spite of the fact that second-order schemes suppress numerical diffusion to a large extent, they often leads to numerical oscillations (wiggles) due to shocks, discontinuities, or sharp changes in the flux.

The problem can be circumvented by switching to a first-order scheme in the regions where the flux presents these characteristics. For this purpose, the scheme (B.13) is rewritten in the form

$$f_i^+ = f_i + \left[\frac{1}{2} \left(\frac{1}{3} + \frac{2}{3} r_i^+ \right) (f_i - f_{i-1}) \right] \quad (\text{B.14})$$

where

$$r_i^+ = \frac{f_{i+1} - f_i + \epsilon}{f_i - f_{i-1} + \epsilon} \quad (\text{B.15})$$

and ϵ is a small number which is introduced to avoid the possibility of a division by zero in a uniform flow region (Koren, 1993).

When the ratio r_i^+ is close to one, the flux is regular. Therefore, the second-order approximation is reliable and has to be preferred since it limits the numerical diffusion. Conversely, when the ratio r_i^+ negative, close to 0, or much larger than one, the flux has a sharp change and a first order scheme has to be preferred.

For this reason, a *flux limiter* function is introduced. Equation (B.14) is modified to

$$f_i^+ = f_i + \left[\frac{1}{2} \phi(r_i^+) (f_i - f_{i-1}) \right] \quad (\text{B.16})$$

where $\phi(r_i^+)$ is the flux limiter function.

In this thesis, we adopted the Koren flux limiter function (Koren, 1993) defined as

$$\phi(r_i^+) = \max \left(0, \min \left(2r_i^+, \min \left(\frac{1}{3} + \frac{2}{3} r_i^+, 2 \right) \right) \right). \quad (\text{B.17})$$

Bibliography

- M.R. Abu Bakar, Z.K. Nagy, A.N. Saleemi, and C.D. Rielly. The impact of direct nucleation control on crystal size distribution in pharmaceutical crystallization processes. *Crystal Growth and Design*, 9(3): 1378–1384, 2009.
- H.S Ahn, Y.Q. Chen, and K.L. Moore. Iterative learning control: brief survey and categorization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1099–1121, 2007.
- J.R. Allegra and S.A. Hawley. Attenuation of sound in suspensions and emulsions: Theory and experiments. *The Journal of the Acoustical Society of America*, 51:1545, 1972.
- K.J. Astrom. Pid controllers: theory, design and tuning. *Instrument Society of America*, 1995.
- A.C. Atkinson and A.N. Donev. *Optimum Experimental Designs*. Clarendon. Oxford, 1992.
- Y. Bar-Shalom and E. Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transaction on Automatic Control*, 19(5):494–500, 1974.
- H.A. Barker, A.H. Tan, and K.R. Godfrey. Design of multilevel perturbation signals with harmonic properties suitable for nonlinear system identification. *IEE Proceedings-Control Theory and Applications*, 151(2):145–151, 2004.
- L.T. Biegler. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*, volume 10. SIAM, 2010.

- X. Bombois, M. Gevers, G. Scorletti, and B.D.O. Anderson. Robustness analysis tools for an uncertainty set obtained by prediction error identification. *Automatica*, 37(10):1629–1636, 2001.
- X. Bombois, G. Scorletti, M. Gevers, P.M.J. Van den Hof, and R. Hildebrand. Least costly identification experiment for control. *Automatica*, 42(10):1651–1662, 2006.
- X. Bombois, M. Barenthin, and P.M.J. Van den Hof. Finite-time experiment design with multisines. In *Proc. of the 17th IFAC World Congress, Seoul, Korea*, 2008.
- X. Bombois, H. Hjalmarsson, and G. Scorletti. Identification for robust H_2 deconvolution filtering. *Automatica*, 46(3):577–584, 2010.
- D. Bonvin. Control and optimization of batch processes. *IEEE Control Systems Magazine*, 26(6):34–45, 2006.
- S. Boyd and L. Chua. Fading memory and the problem of approximating nonlinear operators with volterra series. *IEEE Transactions on Circuits and Systems*, 32(11):1150–1161, 1985.
- S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- R.D. Braatz. Advanced control of crystallization processes. *Annual Reviews in Control*, 26(1):87–99, 2002.
- D.A. Bristow, M. Tharayil, and A.G. Alleyne. A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96–114, 2006.
- A. Cervantes and L.T. Biegler. Optimization strategies for dynamic systems. *Encyclopedia of optimization*, 4:216–227, 1999.
- I. Chin, S.J. Qin, K.S. Lee, and M. Cho. A two-stage iterative learning control technique combined with real-time feedback for independent disturbance rejection. *Automatica*, 40(11):1913–1922, 2004.
- S.H. Chung, D.L. Ma, and R.D. Braatz. Optimal model-based experimental design in batch crystallization. *Chemometrics and Intelligent Laboratory Systems*, 50(1):83–90, 2000.
- M. Dalai, E. Weyer, and M.C. Campi. Parameter identification for nonlinear systems: Guaranteed confidence regions through lscr. *Automatica*, 43(8):1418–1425, 2007.
- S. Datta and D.J.W. Grant. Crystal structures of drugs: advances in determination, prediction and engineering. *Nature Reviews Drug Discovery*, 3(1):42–57, 2004.
- R.A. De Callafon and P.M.J. Van den Hof. Suboptimal feedback control by a scheme of iterative identification and control design. *Mathematical Modelling of Systems*, 3(1):77–101, 1997.

- A. De Cock, M. Gevers, and J. Schoukens. A preliminary study on optimal input design for nonlinear systems. In *Proc. of the 51st IEEE Conference on Decision and Control*, pages 4931 – 4936, 2013.
- A.J. den Dekker, X. Bombois, and P.M.J. Van den Hof. Finite sample confidence regions for parameters in prediction error identification using output error models. In *Proc. of the 17th IFAC world congress (pp. 5024-5029)*, 2008.
- V. V. Fedorov and P. Hackl. *Model-Oriented Design of Experiments*. Lecture Notes in Statistics; 125. Springer Verlag, Berlin, 1996.
- M. Forgione, A. Mesbah, X. Bombois, and P.M.J. Van den Hof. Batch-to-batch strategies for cooling crystallization. In *Proc. of the 50th IEEE Conference on Decision and Control (CDC)*, pages 6364–6369, 2012a.
- M. Forgione, A. Mesbah, X. Bombois, and P.M.J. Van den Hof. Iterative learning control of supersaturation in batch cooling crystallization. In *Proc. of the American Control Conference (ACC)*, pages 6455–6460, Fairmont Queen Elizabeth, Montreal, Canada, 2012b.
- M. Forgione, X. Bombois, and P.M.J. Van den Hof. Experiment design for batch-to-batch model-based learning control. In *Proc. of the American Control Conference (ACC)*, pages 3918–3923, Renaissance Hotel, Washington, D.C., USA, June 2013.
- M. Forgione, G. Birpoutsoukis, X. Bombois, A. Mesbah, P. Daudey, and P.M.J. Van den Hof. Batch-to-batch model improvement for cooling crystallization. *Submitted to Control Engineering Practice*, 2014a.
- M. Forgione, X. Bombois, and P.M.J. Van den Hof. Data-driven model improvement for model-based control. *Submitted to Automatica*, 2014b.
- M. Forgione, X. Bombois, P.M.J. Van den Hof, and H. Hjalmarsson. Experiment design for parameter estimation in nonlinear systems based on multilevel excitation. In *Proc. of the 13th European Control Conference*, 2014c.
- G. Franceschini and S. Macchietto. Model-based design of experiments for parameter precision: State of the art. *Chemical Engineering Science*, 63(19):4846–4872, 2008.
- M. Fujiwara, Z.K. Nagy, J.W. Chew, and R.D. Braatz. First-principles and direct design approaches for the control of pharmaceutical crystallization. *Journal of Process Control*, 15(5):493–504, 2005. ISSN 0959-1524.
- P. Gahinet, A. Nemirovskii, and A. Laub. LMI Lab: a package for manipulating and solving LMIs, 1993.

- S. Garatti, M.C. Campi, and S. Bittanti. Iterative robust control: Speeding up improvement through iterations. *Systems & Control Letters*, 59(2):139–146, 2010.
- M. Gevers. A decade of progress in iterative process control design: from theory to practice. *Journal of Process Control*, 12(4):519–531, 2002.
- M. Gevers. Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, 11:1–18, 2005.
- M. Gevers and L. Ljung. Optimal experiment designs with respect to the intended model application. *Automatica*, 22(5):543–554, 1986.
- G. C. Goodwin and R. L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York, 1977.
- M. Grant, S. Boyd, and Y. Ye. CVX: Matlab software for disciplined convex programming, 2008.
- C. Grillenberger and U. Krengel. On marginal distributions and isomorphisms of stationary processes. *Mathematische Zeitschrift*, 149(2): 131–154, 1976.
- T. Gutwald and A. Mersmann. Batch cooling crystallization at constant supersaturation: technique and experimental results. *Chemical Engineering & Technology*, 13(1):229–237, 1990.
- H. Hjalmarsson. System identification of complex and structured systems. *European Journal of Control*, 15(3):4, 2009.
- H. Hjalmarsson and J. Mårtensson. Optimal input design for identification of non-linear systems: Learning from the linear case. In *Proc. of the American Control Conference*, New York City, USA, July 11-13 2007.
- H. Hjalmarsson, M. Gevers, and F. De Bruyne. For model-based control design, closed-loop identification gives better performance. *Automatica*, 32(12):1659–1673, 1996.
- K. Hsu, K. Poolla, and T.L. Vincent. Identification of structured non-linear systems. *IEEE Transactions on Automatic Control*, 53(11):2497–2513, 2008.
- A. Huesman. Research note: Solving and optimization of ode's. Research Note, 2003.
- A. Huesman. Short note: How to implement dynamic optimisation in matlab -the sequential and simultaneous approach-. Research Note, 2005.

- A. Isidori. *Nonlinear control systems*, volume 1. Springer, 1995.
- J. Jansson and H. Hjalmarsson. Input design via LMIs admitting frequency-wise model specifications in confidence regions. *IEEE Transactions on Automatic Control*, 50(10):1534–1549, 2005.
- K-Patents. Process refractometer pr-23 instruction manual. available on <http://www.kpatents.com/pdf/downloads/pr-23.pdf> (visited October 7, 2013), 2013.
- S.S. Kadam, E. van der Windt, P.J. Daudey, and H.J.M. Kramer. A comparative study of ATR-FTIR and FT-NIR spectroscopy for in-situ concentration monitoring during batch cooling crystallization processes. *Crystal Growth & Design*, 10(6):2629–2640, 2010.
- S.S. Kadam, H.J.M. Kramer, and J.H. ter Horst. Combination of a single primary nucleation event and secondary nucleation in crystallization processes. *Crystal Growth & Design*, 11(4):1271–1277, 2011.
- S.S. Kadam, J.A.W. Vissers, M. Forgione, R.M. Geertman, P.J. Daudey, A.I. Stankiewicz, and H.J.M. Kramer. Rapid crystallization process development strategy from lab to industrial scale with pat tools in skid configuration. *Organic Process Research & Development*, 16(5):769–780, 2012.
- D. Kashchiev. *Nucleation*. Butterworth-Heinemann, 2000.
- B. Koren. *A robust upwind discretization method for advection, diffusion and source terms*. Centrum voor Wiskunde en Informatica Amsterdam, 1993.
- E. Korovessi and A.A. Linninger. *Batch processes*. CRC Press, 2005.
- C. Kulcsar, L. Pronzato, and E. Walter. Dual control of linearly parametrised models via prediction of posterios densities. *European Journal of Control*, 2(135):135–143, 2009.
- N.M. Lakshmanan and Y. Arkun. Estimation and model predictive control of non-linear batch processes using linear parameter varying models. *International Journal of Control*, 72(7-8):659–675, 1999.
- P.A. Larsen, D.B. Patience, and J.B. Rawlings. Industrial crystallization process control. *IEEE Control Systems Magazine*, 26(4):70–80, 2006.
- C.A. Larsson, H. Hjalmarsson, and C.R. Rojas. On optimal input design for nonlinear FIR-type systems. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 7220–7225. IEEE, 2010.
- C.A. Larsson, H. Hjalmarsson, C.R. Rojas, X. Bombois, A. Mesbah, and P.-E. Modén. Model predictive control with integrated experiment design for output error systems. In *European Control Conference, Zurich, Switzerland*, 2013.

- K.S. Lee and J.H. Lee. Iterative learning control-based batch process control technique for integrated control of end product properties and transient profiles of process variables. *Journal of Process Control*, 13(7):607–621, 2003.
- K.S. Lee, I. Chin, H.J. Lee, and J.H. Lee. Model predictive control technique combined with iterative learning for batch processes. *AIChE Journal*, 45(10):2175–2187, 1999.
- R.F. Li, R. Penchev, V. Ramachandran, K.j. Roberts, X.Z. Wang, R.J. Tweedie, A. Prior, J.W. Gerritsen, and F.M. Hugen. Particle shape characterisation via image analysis: from laboratory studies to in-process measurements using an in situ particle viewer system. *Organic Process Research & Development*, 12(5):837–849, 2008.
- L. Ljung. *System identification*. Wiley Online Library, 1999.
- L. Ljung. System identification toolbox. For Use with MATLAB, 2007.
- L. Ljung, H. Hjalmarsson, and H. Ohlsson. Four encounters with system identification. *European Journal of Control*, 17(5):449–471, 2011.
- W.L. Luyben. *Chemical reactor design and control*. Wiley-Interscience, 2007.
- W.L. Luyben, B.D. Tyr eus, and M.D. Luyben. *Plantwide process control*. McGraw-Hill New York, 1998.
- J.M. Maciejowski and M. Huzmezan. *Predictive control*. Springer, 1997.
- I.R. Manchester. Input design for system identification via convex relaxation. In *Proc. of the 49th IEEE Conference on Decision and Control*, pages 2041–2046, Atlanta, GA, USA, 2010.
- A. Mesbah. *Optimal operation of industrial crystallizers*. PhD thesis, Delft University of Technology, 2010.
- A. Mesbah, A.E.M. Huesman, H.J.M. Kramer, Z.K. Nagy, and P.M.J. Van den Hof. Real-time control of a semi-industrial fed-batch evaporative crystallizer using different direct optimization strategies. *AIChE journal*, 57(6):1557–1569, 2011.
- S.K.K. Mitra. *Digital signal processing: a computer-based approach*. McGraw-Hill Higher Education, 2000.
- J.W. Mullin. *Crystallization*, volume 4. Butterworth-Heinemann, 1993.
- A.S. Myerson. *Handbook of Industrial Crystallization*. Butterworth-Heinemann, 2002.
- Z.K. Nagy and R.D. Braatz. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *Journal of process control*, 14(4):411–422, 2004.

- Z.K. Nagy, J.W. Chew, M. Fujiwara, and R.D. Braatz. Comparative performance of concentration and temperature controlled batch crystallizations. *Journal of Process Control*, 18(3-4):399–407, 2008.
- S.Y. Nof. *Springer handbook of automation*. Springer, 2009.
- J.R. Norris. *Markov chains*. Cambridge university press, 1998.
- L. Pronzato. Optimal experimental design and some related control problems. *Automatica*, 44(2):303–325, 2008.
- L. Pronzato, C. Kulcsár, and E. Walter. An actively adaptive control policy for linear models. *IEEE Transactions on Automatic Control*, 41(6):855–858, 1996.
- H. Rabitz, M. Kramer, and D. Dacol. Sensitivity analysis in chemical kinetics. *Annual review of physical chemistry*, 34(1):419–461, 1983.
- D. Ramkrishna. *Population balances: Theory and applications to particulate systems in engineering*. Academic press, 2000.
- A.L. Rohl. Computer prediction of crystal morphology. *Current Opinion in Solid State and Materials Science*, 7(1):21–26, 2003.
- C.R. Rojas, J.S. Welsh, and G.C. Goodwin. A receding horizon algorithm to generate binary signals with a prescribed autocovariance. In *American Control Conference, 2007. ACC'07*, pages 122–127. IEEE, 2007.
- L. Rozsa. Seedmaster 2: A universal crystallization transmitter and automatic seeding device. *International sugar journal*, 1296:683, 2006.
- W.J. Rugh and J.S. Shamma. Research on gain scheduling. *Automatica*, 36(10):1401–1425, 2000.
- N. Sanzida and Z. Nagy. Iterative learning control of a batch cooling crystallization process based on linear time-varying perturbation models. In *Proc. of the 22nd European Symposium on Computer Aided Process Engineering, London.*, June 2012.
- J. Schoukens and R. Pintelon. *Identification of linear systems: a practical guideline to accurate modeling*. Pergamon press New York, 1991.
- J. Schoukens, R. Pintelon, T. Dobrowiecki, and Y. Rolain. Identification of linear systems with nonlinear distortions. *Automatica*, 41(3):491–504, 2005.
- R.J.P Schrama. Accurate identification for control: The necessity of an iterative scheme. *IEEE Transactions on Automatic Control*, 37(7):991–994, 1992.
- D.E. Seborg, D.A. Mellichamp, T.F. Edgar, and F.J. Doyle III. *Process dynamics and control*. John Wiley & Sons, 2010.

- J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.Y. Glorenec, H Hjalmarsson, and A Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.
- T. Soderstrom. Errors-in-variables methods in system identification. *Automatica*, 43(6):939–958, 2007.
- J.D. Stigter, D. Vries, and K.J. Keesman. On adaptive optimal input design: a bioreactor case study. *AIChE journal*, 52(9):3290–3296, 2006.
- K.C. Toh, M.J. Todd, and R.H. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, 11(1-4):545–581, 1999.
- K. Tomazi, A. Linninger, and J. Daniel. *Batch processing industries*. Boca Raton, FL: CRC Press, 2006.
- E. Tse and Y. Bar-Shalom. An actively adaptive control for linear systems with random parameters via the dual control approach. *IEEE Transactions on Automatic Control*, 18(2):109–117, 1973.
- E. Tse, Y. Bar-Shalom, and L. Meier III. Wide-sense adaptive dual control for nonlinear stochastic systems. *IEEE Transactions on Automatic Control*, 18(2):98–108, 1973.
- S. Vajda, H. Rabitz, E. Walter, and Y. Lecourtier. Qualitative and quantitative identifiability analysis of nonlinear chemical kinetic models. *Chemical Engineering Communications*, 83(1):191–219, 1989.
- P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. Optimal input design for non-linear dynamic systems: a graph theory approach. In *Proc. of the 51st IEEE Conference on Decision and Control*, 2013.
- A. Van den Bos. *Parameter estimation for scientists and engineers*. Wiley-Interscience, 2007.
- P.M.J. Van den Hof. Advanced autonomous model-based operation of industrial process systems, February 2014. URL <http://www.fp7-autoprofit.eu/>.
- P.M.J. Van den Hof and R.J.P. Schrama. Identification and control – closed-loop issues. *Automatica*, 31(12):1751–1770, 1995.
- M. Verhaegen and P. Dewilde. Subspace model identification part 1. the output-error state-space model identification class of algorithms. *International journal of control*, 56(5):1187–1210, 1992.
- J. Vissers. *Model-based estimation and control methods for batch cooling crystallization*. PhD thesis, Eindhoven University of Technology, 2012.

- J. Vissers, P. Jansen, and S. Weiland. Control of supersaturation in batch cooling crystallization by robust state feedback linearization. In *Proc. of the 9th IEEE International Conference on Control and Automation (ICCA)*, pages 1114–1120. IEEE, 2011.
- J. Vissers, M. Forgione, S. Kadam, P. Daudey, T. Backx, A. Huesman, H. Kramer, and P.M.J. Van den Hof. Novel control of supersaturation on an industrial scale pharmaceutical batch crystallizer. In *Proc. of the 18th International Symposium on Industrial Crystallization*, pages 141–142, 2012.
- M. Volckaert, M. Diehl, and M. Swevers. A two step optimization based iterative learning control algorithm. In *Dynamic Systems and Control Conference*, pages 579–581, Cambridge, Massachusetts, USA, September 2010.
- Y. Wang, F. Gao, and F.J. Doyle III. Survey on iterative learning control, repetitive control, and run-to-run control. *Journal of Process Control*, 19(10):1589–1600, 2009.
- R. Wissing, M. Elwenspoek, and B. Degens. In situ observation of secondary nucleation. *Journal of Crystal Growth*, 79(1):614–619, 1986.
- H.K. Wong, J. Schoukens, and K.R. Godfrey. Design of multilevel signals for identifying the best linear approximation of nonlinear systems. *IEEE Transactions on Instrumentation and Measurement*, 2013.
- W. Xie, S. Rohani, and A. Phoenix. Extended kalman filter based nonlinear geometric control of a seeded batch cooling crystallizer. *The Canadian Journal of Chemical Engineering*, 80(1):167–172, 2002.
- J. Zhang, J. Nguyen, Z. Xiong, and J. Morris. Iterative learning control of a crystallization process using batch wise updated linearized models. In *Computer Aided Chemical Engineering*, pages 387–392, 2009.

List of Abbreviations

ARX	AutoRegressive Exogenous
B2B	Batch-to-batch
BVP	Boundary Value Problem
CSD	Crystal Size Distribution
CSTR	Continuous Stirred Tank Reactor
DOP	Dynamic Optimization Problem
FIR	Finite Impulse Response
IIC	Iterative Identification Control
ILC	Iterative Learning Control
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
LTI	Linear Time-Invariant
MPC	Model Predictive Control
MSZW	MetaStable Zone Width
OPC	Open Platform Communication
ODE	Ordinary Differential Equation
PBE	Population Balance Equation
PDE	Partial Differential Equation
PI	Proportional Integral
PID	Proportional Integral Derivative
PMP	Pontryaging Maximum Principle

RH	Receding Horizon
RMS	Root Mean Square
MIMO	Multiple-Input Multiple-Output
SDP	SemiDefinite Problem
SISO	Single-Input Single-Output
SIMO	Single-Input Multiple-Output

Summary

Batch-to-batch learning for model-based control of process systems with application to cooling crystallization

Marco Forgione

From an engineering perspective, the term process refers to a conversion of raw materials into intermediate or final products using chemical, physical, or biological operations. Industrial processes can be performed either in continuous or in batch mode. There exist for instance continuous and batch units for reaction, distillation, and crystallization. In batch mode, the raw materials are loaded in the unit only at the beginning of the process. Subsequently, the desired transformation takes place inside the unit, and the products are eventually removed altogether after the processing time. In order to obtain the desired production volume, several batches are repeated.

In an industrial process, several variables such as temperatures, pressures, and concentrations have to be regulated in order to ensure safety, maintain the product quality, and optimize economic criteria. In principle, model-based control techniques available in the literature could be systematically utilized in order to achieve these goals. However, a limitation to the applicability of model-based techniques for batch process control is that the available models of batch processes often suffer from severe uncertainties.

In this thesis, we have investigated the use of measured data in order to improve the performance of model-based control of batch processes. Our approach consists in using the measured data in order to refine from batch to batch the model that is used to design the controller. By doing so, the performance delivered by the model-based controller is expected to improve. We have developed the parametric

model update technique Iterative Identification Control (IIC) and non-parametric model update technique Iterative Learning Control (ILC). While in IIC the measured batch data are used to update from batch to batch parameter estimates for the uncertain physical coefficients, in ILC the data are used to compute a non-parametric, additive correction term for a nominal process model.

We have tested the ILC and IIC algorithms for the batch cooling crystallization process both in a simulation environment and on a real pilot-scale crystallization setup. We have shown that the two approaches have complementary advantages. On the one hand, the parametric approach allows for a faster learning since it produces a parsimonious representation of the process. On the other hand, the nonparametric approach can cope effectively with the serious issue of structural mismatches owing to the use of a more flexible representation.

Furthermore, we have investigated the use of excitation signals to enhance the performance of parametric model update techniques in an iterative identification/controller design scheme similar to IIC. The excitation signals have a dual effect on the overall control performance. On the one hand, the application of an excitation signal superposed to the normal control input leads after identification to an increased model accuracy, and thus a better control performance. On the other hand, the excitation signal also causes a temporary performance degradation, since it acts as a disturbance while it is applied to control system. For linear dynamical systems, we have shown that the problem of designing the excitation signals aiming to maximize the overall control performance can be approximated as a convex optimization problem.

The lack of generally applicable and computationally efficient experiment design tools for nonlinear systems is the main bottleneck for the optimal design of the excitation signals in the case of batch processes. In this thesis, we have developed a novel experiment design method applicable to the class of fading memory nonlinear system. Limiting the excitation signals to a finite number of levels, the information matrix can be expressed as a linear function of the frequency of occurrence of each possible pattern having duration equal to the memory of the system. Exploiting the linear relation between the frequencies and the information matrix, several experiment design problems can be formulated as convex optimization problems.

Samenvatting

Batch-na-batch modelverfijning voor de modelgebaseerde regeling van processystemen met als toepassing cooling kristallisatie

Marco Forgione

Vanuit het oogpunt van een ingenieur duidt de term “proces” een conversie aan van onbewerkte materialen naar tussen- of eindproducten door middel van chemische, fysische, of biologische handelingen. Industriële processen kunnen in een continue of in een batch modus uitgevoerd worden. Voorbeelden hiervan zijn eenheden voor distillatie en kristallisatie. In de seriegewijze modus worden de onbewerkte materialen aan het begin van het proces de eenheid ingebracht, waarna binnen de eenheid de conversie plaatsvindt. Na een zekere tijd worden de eindproducten uit de eenheid gehaald. Om voor voldoende productie volume te zorgen worden verschillende seriegewijze handelingen herhaald.

In een industrieel proces moeten verschillende variabelen zoals temperatuur, druk, en concentraties nauwkeurig gereguleerd worden teneinde veiligheids-, kwaliteits-, en winstnormen te waarborgen. In zekere zin bestaan er in de literatuur al modelgebaseerde regeltechnieken die aan deze eisen kunnen voldoen. Een beperking van deze modelgebaseerde regeltechnieken is echter hun toepasbaarheid op batch-processen, gezien het feit dat huidige modellen voor deze processen onderhevig zijn aan enorme onzekerheden.

Dit proefschrift bevat nieuwe methodieken die de prestatie van de modelgebaseerde regeling van batch-processen verhoogd. De methodes maken gebruik van meetdata om het model dat gebruikt wordt om de regelaar te ontwerpen te verfijnen na elke batch. Twee verschillende technieken zijn ontwikkeld: *Iterative Identification Control* (IIC)

en *Iterative Learning Control* (ILC). In de IIC methode worden de meetdata, die tijdens de laatste batch zijn verzameld, gebruikt om de nauwkeurigheid van de parameters van het procesmodel na elke batch te vergroten, waarentegen de ILC methode deze data gebruikt om een niet-parametrische additieve correctie term te berekenen voor een nominaal model van het proces.

We hebben de ILC en IIC methodieken getest op een batch-kristallisatie proces in een simulatie omgeving alsook in een echte, doch kleinschalige, kristallisator. Hieruit bleek dat elk algoritme zijn voordelen heeft. De IIC methodiek heeft een snelle leercurve door de vaste modelstructuur waarbinnen de parameters verfijnd worden. Daarentegen heeft de ILC methodiek als voordeel zijn flexibeler representatie van het proces om zo effectief om te gaan met structurele afwijkingen.

Verder is onderzocht hoe, door gebruik te maken van speciaal ontworpen signalen, de prestatie van verfijntechnieken voor parametrische modellen in een iteratieve identificatie/regelaar ontwerp omgeving te verbeteren. Deze signalen hebben twee effecten op de regulatieprestatie. Aan de ene kant zorgt de toepassing van de ontworpen signalen voor een verhoogde precisie van het model na een identificatie procedure en hierdoor voor een verbeterde regulatieprestatie. Aan de andere kant zorgen deze signalen, wanneer toegepast op het systeem, voor een tijdelijke verslechtering van de prestaties. Voor lineaire dynamische systemen tonen we dat het probleem van het ontwerpen van de signalen met als doel het maximaliseren van de regulatieprestatie, kan worden omgezet in een convex optimalisatie probleem.

Voor de niet-lineaire batch-processen was er een tekort aan algemeen toepasbare en efficiënte methodes om optimale signalen te ontwerpen. In dit proefschrift is een nieuwe methodiek ontworpen die toepasbaar is op de klasse van niet-lineaire systemen met een langzaamverdwijnd geheugen (*fading memory*). Door ons te beperken tot signalen met een eindig aantal niveaus kan de informatie matrix uitgedrukt worden als een lineaire functie van de frequentie van voorvallen van elk mogelijk patroon met een lengte die gelijk is aan het geheugen van het systeem. Daarnaast kunnen, door gebruik te maken van het lineaire verband tussen deze frequenties en de informatie matrix, verschillende optimalisatie problemen geformuleerd worden als convexe optimalisatie problemen.

Curriculum Vitae



Marco Forgione was born on July 6, 1986 in Varese, Italy. He obtained his B.Sc. and M.Sc. degree in Computer Engineering “cum laude” from the University of Pavia. In his B.Sc. thesis, he contributed to the design, implementation, and control of a laboratory-scale gantry crane. In his M.Sc. thesis, he worked on the black-box modeling through identification of the glucose-insulin metabolism for the development of an automated insulin delivery system (Artificial Pancreas) in type-1 diabetes. In March 2010, he started his Ph.D. at the Delft Center for Systems and Control. The primary objective of his Ph.D. research was to investigate the use of measured batch data to improve model-based control strategies for industrial batch processes. During his Ph.D., he performed experiments on an industrial-scale batch crystallizer at Merck Sharp and Dohme in Apeldoorn and on a pilot-scale batch crystallizer at DSM in Geleen. Furthermore, he obtained the DISC certificate for completing the graduate course program of the Dutch Institute for Systems and Control. His main research interests are iterative control, system identification, experiment design, and robust control with applications in different engineering domains.

