Optimal and Reproducible Operation of Batch Processes

Dennis Bonné

2005

Department of Chemical Engineering
Technical University of Denmark
Preface

The present thesis was submitted in partial fulfillment of the requirements for obtaining the Ph.D. degree in Chemical Engineering from the Technical University of Denmark. The work presented in the present thesis was carried out from August 2000 to October 2003 in part at the Department of Chemical Engineering, Technical University of Denmark and in part at the Department of Chemical Engineering, University of Wisconsin – Madison. The work was primarily financed by the Danish Energy Authority under contracts ENS 1223/99-0008 and 1273/00-0026.

Throughout my work, a number of people have offered their help and support and for that I am grateful. I would therefore like to thank first of all, my supervisor Professor Sten Bay Jørgensen for his inspiration and guidance and for presenting me with this opportunity, and Professor James B. Rawlings for his supervision during my very pleasant and very productive stay at UW – Madison. In addition to my supervisors I would like to thank Professor Per Christian Hansen for his help on regularization, Associate Professor Niels Kjølstad Poulsen for his help on Kalman filter theory, and Professor Henrik Madsen whom, among others, inspired me to pursue a Ph.D. degree in systems identification and control. Special thanks to my ex-colleagues Niels Rode Kristensen, Lars Gregersen and John Bagterp Jørgensen whom have all in excess of their friendship, contributed their ideas and insight to my work, and Morten Skov Hansen and Peter Mathias Harper for their friendship. And thanks to all those ex-colleagues whom I have not already mentioned — René Skotte, Irene Papaconomou, Mads Thaysen, Krist Gernaey, Martin Hostrup, Erik Bek-Pedersen, Mario Richard Eden, Torben Ravn Andersen — for making my 3+ years at DTU unforgettable. Also thanks to my friend Juan Manuel Guerrero Muñoz for many, oh so many, endless discussions, and to my friends and ex-colleagues in Madison and Porto.

Last but not least, heartfelt thanks to my family and especially my girlfriend for their love and support and for taking such an immense interest in my work, and to all my friends for still being my friends “post-doc”.

Copenhagen, November 2004

Dennis Bonné
Abstract

The present thesis proposes and justifies a model-based methodology for optimal and reproducible operation of chemical batch processes. The development of the proposed methodology was motivated by an industrial need for a more systematic approach to achieving increased reliability and profitability in operation of chemical batch processes. In the quest for a more systematic approach to optimal and reproducible operation of chemical batch processes, the main obstacle faced by industry and academia alike is the deficiency of reliable mathematical batch process models. The methodology proposed in the present thesis thus comprises a modeling methodology as well as a suite of model-based tools for the implementation of optimal and reproducible batch process operation. The proposed methodology has been implemented as a MATLAB toolbox and has been thoroughly tested and validated on several case studies.

To serve as a cost and time efficient alternative (supplement) to the comprehensive and often infeasible, development of phenomena-based models of chemical batch processes, the proposed modeling methodology is data-based and a model can thus be relatively readily obtained from historical process operation data. The causal, linear models obtainable with the proposed modeling methodology, model the differences between the batch evolutions of process variables given changes to the batch operational models according to which batch processes are operated. This modeling of batch-to-batch deviations constitutes a significant contribution to the state-of-the-art of (linearly) modeling differences in batch evolutions of process variables relative to a mean or normal batch evolution, as the inevitable modeling errors are then more convincingly approximated with independent zero-mean white noise sequences, which is an assumption generally made in system identification. More specifically, the proposed modeling methodology models batch processes with relatively large sets of interdependent local Auto-Regressive Moving Average models with eXogenous inputs (ARMAX). The novel introduction of interdependency of these local ARMAX models significantly decreases the sensitivity to measurement noise of the modeling methodology. The Pseudo Linear Regression (PLR) algorithm developed to estimate the ARMAX models from historical process data, does not in contrast to state-of-the-art PLR algorithms, require user supervision, as the parameter space of the underlying linear Least Squares (LS) estimation problem is shrunk by imposed model properties; here amongst the interdependency of the local ARMAX models. In order to assess the quality of ARMAX models obtained with the proposed modeling methodology in terms of how well they generalize, the modeling methodology comprises a novel model quality measure which is based on efficiently computed model prediction errors.
and which can be adapted to the most appropriate prediction horizon for a particular modeling purpose. Furthermore, based on this model quality measure, an algorithm for model quality optimization including model structure identification has been developed. That is, given a well defined modeling purpose and a set of input and output variables chosen according to this modeling purpose, and a historical data set prepared according to the data preparation procedure developed as an integral part of the proposed modeling methodology, the procedure for identifying an optimal batch process model is fully autonomous.

The present thesis furthermore proposes a control methodology for optimal and reproducible operation of chemical batch processes, which comprises a suite of linear model-based tools adapted to the finite horizon and repetitive nature of batch processes. For the achievement of reproducible operation of batch processes, the control methodology proposes an Iterative Learning Control (ILC) algorithm that by utilizing information available from the previous batch run will improve reproducibility in the next batch run. The proposed ILC algorithm thus learns to reject persistent process disturbances. The control methodology furthermore proposes a learning Model Predictive Control (MPC) algorithm which outperforms the proposed ILC algorithm as it in addition to the learning capability of the ILC algorithm, also comprises on-line disturbance attenuation based on information gathered during a batch run. With the purpose of proving the reproducibility increasing properties of the two proposed control algorithms, a novel control design requirement has been developed. When this control design requirement is satisfied, it has been proven in the present thesis that ILC and learning MPC algorithms guarantee improved reproducibility when implemented on known linear batch systems driven by and with observations subject to independent zero-mean white noise disturbance sequences. The state-of-the-art reproducibility proofs for ILC and learning MPC algorithms only apply to implementations on known deterministic batch systems with perfect observations. For the achievement of optimal operation of batch processes, the proposed control methodology defines an optimal batch operational model as the solution to a Linear Program (LP) and furthermore proposes a procedure in which the learning MPC algorithm is used to iteratively bring the operation of a batch process closer to its optimal batch operational model. However, as this optimization procedure is based on local linear models it can not be proven to achieve the optimal batch operation model. Finally the proposed control methodology comprises a combination of a Kernel Smoother and a Kalman filter for state and output estimation given data with multiple sample frequencies.

The combined methodology for optimal and reproducible operation of chemical batch processes proposed in the present thesis has been tested on several case studies some of which included modeling of historical data from industrial production sites. Based on the results of these case studies and the theory presented in the present thesis, it is concluded that the proposed methodology has demonstrated sufficient potential for it partially or in its whole, to at least be subject to further investigation with the aim of potential industrial application.
Resumé


For at tjene som et omkostnings- og tidsbesparende alternativ (supplement) til den omfattende og ofte ikke rentable, udvikling af fænomen-baserede modeller for kemiske batch processer, er den forelagte modellerings metodik data-baseret og en model kan derfor forholdsvis let dannes ud fra eksisterende proces operations data. De kausale, lineære modeller der kan dannes med den forelagte metodik, modellerer afvigelser mellem batch udviklingen af proce-ces variable givet ændringer i de operationelle modeller, ifølge hvilke batch processerne drives. Dette at modellere batch-til-batch afvigelser udgør et signifikant bidrag til state-of-the-art for (lineær) modellering af afvigelser i batch udvikling af proces variable relativt til en middel eller normal batch udvikling, da de umudgåelige modelfejl da er mere overbevisende tilhævet med ukorrelerede hvid-støj processer med gennemsnit hen til nul, hvilket er en almindelig forudsætning i system identifikation. Mere specifikt, så modellerer den forelægte modellerings metodik batch processer med en relativ stor mængde indbyrdes afhængige lokale Auto-Regressive Moving Average modeller med eXogene styreindgreb (ARMAX). Den hidtil usete introduktion af indbyrdes afhængighed mellem disse lokale ARMAX modeller nedbringer modellerings metodikens sensitivitet overfor målestøj betydeligt. Den Pseudo Lineære Regressions (PLR) algoritme der er blevet udviklet til at estimere ARMAX modellerne fra eksisterende proces data, er i modsætning til state-of-the-art PLR algoritmer ikke afhængig af bruger overvågning, da parameter rummet i det underliggende lineære Mindste Kvadraters estimatorings problem er komprimeret af pålagte model egenskaber; heriblandt indbyrdes afhængighed mellem de lokale ARMAX modeller. For at kunne vurdere kvaliteten af ARMAX modellerne er danned med den forelagte modellerings metodik i form af hvor godt de generaliserer, indbefatter modellerings metodikken en nyt model kvalitets mål,
Abstract (in Danish)

der er baseret på effektivt beregnede prædiktionsfejl og som kan tilpasses den mest egne prædiktions horisont til modelleringens formålet. Baseret på dette model kvalitets mål, er der endvidere udviklet en algoritme til model kvalitets optimering herunder også struktur identifikation. Det vil sige at identifikation proceduren til dannelsen af en optimal batch model er helt autonom, når der er givet et vel definert modelleringens formål og et sæt ind- og udgangs signaler udvalgt i henhold til dette modelleringens formål, samt en mængde proces data prepareret i henhold til den data præparerings procedure der er blevet udviklet som en integreret del af den forelagte modelleringens metodik.

Endvidere forelægger denne afhandling en regulerings metodik til optimal og reproducerbar drift af kemiske batch processer, der indbefatter en samling lineære model-baserede værktøjer tilpasset den endelig horisonts og repetitions prægede karakter der kendegør batch processer. Til at opnå reproducerbar drift af batch processer, forelægger regulerings metodikken en Iterative Learning Control (ILC) algoritme, der benytter information fra forrige batch kørsel til at forbedre reproducerbarheden i den næste batch kørsel. Den forelagte ILC algoritme lærer altså hvordan blivende forstyrrelser afvises. Ydermere forelægger regulerings metodikken en lærende Model Predictive Control (MPC) algoritme, der er den forelagte ILC algoritme overlegen, da den i tillegg til ILC algoritmens evne til at lere, også indbefatter on-line forstyrrelsens afvisning baseret på information indsamlet under en batch kørsel. For at bevise de to regulerings algoritmers reproducerbarheds øgende egenskaber, er der blevet udviklet et nyt regulator design krav. Det er blevet bevist i denne afhandling, at når dette regulator design krav er opfyldt, så garanterer ILC og lærende MPC algoritmerne øget reproducerbarhed, når de anvendes på kendte lineære batch systemer, der drives af og med målinger der er påvirket af, ukorrelerede hvid-støjs processer med gennemsnittet nul. Hidtil kendte reproducerbarheds beviser for ILC og lærende MPC algoritmer gælder kun anvendelser på kendte deterministiske batch systemer med perfekte målinger. Til at opnå optimal drift af batch processer, definerer regulerings metodikken en optimal batch operationelle model som løsningen til et Lineært Program (LP) og forelægger endvidere en procedure der benytter den lærende MPC algoritme til iterativt at bringe en batch proces nærmere dens optimale batch operationelle model. Da denne optimierings procedure er baseret på lokale lineære modeller, kan det dog ikke bevises at den optimale batch operationelle model tilnærmes. Endelig indbefatter den forelagte regulerings metodik en kombination af en Kernel Smoother og et Kalman filter til tilstands og udgangs-signal estimering givet data med multiple samplings frekvenser.

Den grundlæggende metodik til optimal og reproducerbar drift af kemiske batch processer der er forelagt i denne afhandling er blevet testet på en mængde cases hvoraf nogle inkluderede modellering af industriel produktions data. På baggrund af resultaterne fra disse cases og den i afhandlingen præsenterede teori, konkluderer at den forelagte metodik har demonstreret tilstrækkeligt potentiale til i det mindste at retfærdiggøre at metodikken eller dele heraf, udforskes nærmere med henblik på industrielle anvendelser.
3.6 Control and Optimization Conclusions .................................................. 106

4 Application Examples .................................................................................. 111
4.1 Pilot Plant Fermentation ........................................................................... 111
4.2 Fermentation of Yeast ............................................................................. 112

5 Conclusion .................................................................................................. 117
5.1 Future Work ............................................................................................. 120

References ...................................................................................................... 121
1 Introduction

Throughout human evolution from the earliest chemical syntheses such as conservation of foods, preparation of paints, fermentation of vine and beer, preparation medical remedies and even alchemy, batch type execution has been the predominant execution of chemical syntheses. Not until the maturing of systems engineering and the ever growing demand for bulk production in the 1900’s did continuous type execution start to dominate the industrial production of foods and chemicals. Batch processes are however experiencing a renaissance as products-on-demand and first-to-market strategies impel the need for flexible and specialized production methods. Furthermore, industries such as food, biochemical, and pharmaceutical depend on the confinement of faults and contaminations to single batches. This renaissance propels the need for modeling and control tools that can facilitate optimal and reproducible operation of batch processes. However, the typically linear modeling and control tools developed for continuous, stationary processes often prove inadequate, when applied to the often highly nonlinear and non-stationary behavior of batch processes.

This thesis attempts to bridge the gap between the non-stationary behavior of batch processes and the linear modeling and control tools developed for continuous type processes. More specifically, it is intended to find reasonable approximations in the form of linear models, of batch processes such that the well established linear model-based state estimation and control tools such as the Kalman filter, Iterative Learning Control and Model Predictive Control, can be implemented in the pursuit of optimal and reproducible operation of batch processes. Furthermore, the thesis will establish how Model Predictive Control and Iterative Learning Control (or optimal decision making) problem formulations can be applied in the search for optimal batch operations models.

1.1 Motivation

Chemical batch processes are typically operated according to operations models or for cultivations, recipes, that specify tasks such as charging/discharging of reactants/products and addition/removal of heat and other utilities as well as the sequencing and timing of these tasks. The batch operations models thus typically consist of a mixture of feed rate profiles specifying the charging of reactants, and reference profiles specifying target values for process parameters
such as temperature, pressure and pH. In industry, batch operations models are more often than not developed based on heuristics that have been tested and improved in laboratories and pilot plants before being implemented in production scale. Industrial batch operations models are thus typically simple and conservative, yet also time consuming and expensive to develop. During batch operation, the target values specified by feed rate and reference profiles are usually achieved in closed-loop (i.e., with the use of data measured on- or at-line during the batch operation) with Single-Input-Single-Output (SISO) control loops implemented at Programmable Logic Controllers (PLC) level. The feed rate and reference profiles themselves, are however, usually implemented without automated closed-loop control, which means that the pre-specified feed rate and reference profiles are implemented independently of the state of the batch operation. When feed rate and reference profiles are indeed adapted to the state of the batch process during operation, this adaptation is done one feed rate or reference profile at a time, disregarding the multivariable complexity of the batch process. This state-of-the-art batch control structure leaves batch process operation exposed to undesired variability induced by process disturbances, in product quality and quantity measures, affecting the profitability of chemical batch production.

Given the description of state-of-the-art chemical batch processing above, it should be clear that there is an imminent industrial need for systematic tools/procedures/algorithms both for design and optimization of batch operations models and for (improved) process disturbance attenuation. In response to this industrial need for systematic tools/procedures/algorithms, academia has developed a broad scope of alternatives, most of which are based on process specific phenomena (or first engineering principles) based models. A detailed review of all of these process specific alternatives falls outside the scope of this thesis and will hence be omitted. Some of the process specific alternatives have been reviewed by Berber (1996); Bonvin (1998); Lübbert and Jørgensen (2001); Jørgensen et al. (2005). Instead, this thesis will focus on and further develop some of the ideas presented by Srinivasan et al. (2003b,a) and Lee et al. (2000). Srinivasan et al. (2003b) proposed that a nominally optimal batch operations model is found by tracking the path and terminal constraints that define optimal operation for a particular batch process given a particular operational objective. The set of path and terminal constraints, which is labeled the solutions model, that define optimal operation or at least a conservative estimate hereof, is found with relatively simple phenomenon based model representations of the batch processes. Srinivasan et al. (2003b); Bonvin and Srinivasan (2003) further more proposed a model-free Run-to-Run control strategy for progressively less conservative implementations of the found solutions models. In a very similar, but completely model-free, framework, Åkesson (1999) proposed determining the solutions models given the heuristics and known phenomena of a batch process, and then tracking the solutions model with a so called probing controller. The two abovementioned methodologies are designed to optimize some operational objective, but have little focus on process disturbance at-
1.2. Hypothesis

The working hypothesis of this thesis, namely that

Constitutive Hypothesis: Batch processes can be modeled from data and these models can be used for model-based state estimation, control and optimization, thus providing optimal and reproducible operation of batch processes.

This constitutive hypothesis is broken down into a modeling hypothesis, a control hypothesis and an optimization hypothesis

Modeling Hypothesis: The non-stationary and most often nonlinear dynamic behavior of batch processes can be approximated with sets of interdependent linear models. These models can be obtained from sparse historical production data and the model sets will approximate batch processes sufficiently well for them to be successfully applicable for the design of model-based state estimation and control tools such as the Kalman filter, Model Predictive Control and Iterative Learning Control. Moreover, as a simulation tool, the models can aid the search for a better or an optimal batch operations model.

Control Hypothesis: Asymptotically with respect to the batch run index, deterministic error-free tracking of a desired batch operations model can be obtained with Iterative Learning Control. In addition to the asymptotically perfect tracking performance of Iterative Learning Control, Model...
Predictive Control can provide on-line attenuation of stochastic disturbances when applied to tracking control of batch processes, and will thus perform superior to Iterative Learning Control. Furthermore, state estimation can be performed with an ordinary Kalman filter.

**Optimization Hypothesis:** The design of optimal batch operations models can be formulated as optimal control problems. The design of deterministic optimal batch operations models can be formulated as Iterative Learning Control problems. Formulating the operation design problems as Model Predictive Control problems however will produce dynamically optimal batch operations models.

### 1.3 Methodology

The contribution of this thesis is presented as three methodologies, each of which justify one of the three hypotheses stated above. The three methodologies modeling, control and optimization, consist of several (sub-) methodologies. That is, the presented methodologies will consist of novel theory and innovative interpretations and combinations of existing methodologies. Combined, these three methodologies constitute one innovative methodology for optimal and reproducible operation of batch processes, justifying the constitutive hypothesis stated above.

#### 1.3.1 Modeling Methodology

The modeling methodology suggested in this thesis, serves the objective of readily providing an input-output description, i.e., a model, of batch processes given historical operation data. The model is sufficiently accurate to be used both for simulation and for design of model-based, reference tracking control tools. The modeling methodology comprises the following elements:

**Data Preparation:** Before historical process data can be used for batch process identification it must be properly prepared. In this thesis a three step procedure is suggested. The first step in data preparation is to remove both data outliers and entire batches that are not consistent with expected batch behavior, by visual inspection of the data. If possible erroneous sensor calibrations should furthermore be corrected. The second step in data preparation is to simultaneously filter and re-sample the data sequences using a Kernel Smoother without reconstructing missing data sequences — i.e., without performing data interpolation and/or extrapolation in time intervals of significant length, where no data is present. In this step high frequency measurement noise is (partially) removed from the data and the data is re-sampled according to the modeling specifications. The third step in data preparation is to reconstruct missing data sequences by estimating the missing data sequences in deviation variables, i.e., the differences between the process variables and
suitably chosen references, with another Kernel Smoother. In the case of no missing data sequences, this procedure for data preparation will produce unbiased estimates of batch data sequences in deviation variables. The statement of unbiased estimates of batch data sequences in deviation variables is valid, when it is assumed that:

**Modeling Assumption #1:** The estimation errors are partly due to persistent (from batch to batch) bias resulting from the choice of Kernel, bandwidth and local regression order and partly due to zero-mean random disturbances.

**Model Development:** When challenged with the nontrivial task of building models of batch processes from their historical operation data, this thesis proposes to approximate the non-stationary and nonlinear behavior of these processes with a set of pseudo linear regression models that are local in time and (state) space. This local, pseudo linear approximation of non-stationary and nonlinear processes is valid, when it is assumed that:

**Modeling Assumption #2:** The changes in the input/output behavior of batch processes, will occur at approximately the same point in (operation) time in every batch.

In addition, it is proposed that each local model must be dependent on its nearest (in time) neighbors. In fact, it is proposed to approximate the process behavior between all consecutive sample instances with each its local model and that each of these local models are dependent on their two neighboring models. With this approximation method, it is the number (and placement) of local models necessary to sufficiently accurately describe the behavior of a specific batch process that determines the necessary number (and placement) of sample instances and not vice versa. It is furthermore proposed that the local models are parameterized as AutoRegressive Moving Average models with eXogenous inputs, which results in a multivariate yet parsimonious parameterization. The AutoRegressive Moving Average model with eXogenous inputs parameterization is valid, when it is assumed that:

**Modeling Assumption #3:** The model approximation errors are partly due to persistent (from batch to batch) modeling and batch operations model bias, and partly due to zero-mean random disturbances.

The selected parameterization provides a model set which is linear time-invariant from batch to batch. Equally important, with this parameterization the set of local models may be converted into a canonical linear time-varying state space model, which can be applied directly in the design of model-based tools such as the Kalman filter, Model Predictive Control and Iterative Learning Control.
**Parameter Estimation:** When batch processes are approximated with large sets of parameterized models, most batch process estimation problems will be rank-deficient and/or ill-conditioned. This means that special measures must be taken in order to reduce the variance of the estimated model parameters, as excessive parameter variance means that identified models generalize poorly. However, the model property, i.e., that neighboring local models must be dependent which is imposed onto the batch process estimation problem in this thesis, improves the conditioning of these problems. It is in fact proposed to imposed further model properties onto the batch process estimation problem, i.e., that the local models have smooth step responses and that the parameter estimates have low variance. All of these desirable model properties can be imposed onto a Least Squares estimate of the model parameters with Tikhonov Regularization. The imposed model properties will unavoidably introduce bias into the model parameter estimates, but due to the reduction in variance, a biased model will generalize in a superior manner compared to its unbiased counterpart.

**Structural Identification:** In order to identify the model structure that best approximate a specific batch process, the quality of models with alternative structures is assessed in terms of their ability to generalize to data independent of the data from which their parameters were estimated. The optimal model structure for a specific batch process is defined as the solution to a Mixed Integer NonLinear Programming problem. This Mixed Integer NonLinear Programming problem is non-convex, but when broken down into a hierarchy of three sub-optimization problems, of which the lowest level is convex, the second level is assumed convex, and the top level is a non-convex integer programming problem, a near optimal model structure can be obtained. The assumption that the second optimization level is convex is valid, when it is assumed that:

**Modeling Assumption #4:** There exists a unique solution to the lowest level of the optimization hierarchy. I.e. the batch model parameter estimation problem has a unique solution.

### 1.3.2 Control and Optimization Methodologies

The control methodology suggested in this thesis, serves the objective of providing tools applicable for reproducible operation of batch processes in the form of linear optimal control formulations of reference tracking control. The suggested control methodology for ensuring reproducible batch operation comprises the constrained Optimal Control Problem formulations Iterative Learning Control for inter-batch implementations and learning Model Predictive Control for intra-batch implementations together with a necessary and sufficient condition for guaranteed reproducible operation. This condition for guaranteed reproducible operation is satisfied if on average the expected reproducibility in an upcoming batch operation is at least as good as the reproducibility achieved in
the prior batch operation. The necessary and sufficient condition for guaranteed reproducible operation can thus be used to verify whether or not batch operation reproducibility will improve before the batch operation is commenced. In addition to the condition for guaranteed reproducible operation, controllability of batch processes is defined. Furthermore, because reproducible operation is guaranteed even for unreachable batch operations models, dynamically optimized batch operation can be realized with a properly designed batch operations model.

The optimization methodology suggested in this thesis, serves the objective of providing tools that are applicable for optimizing operation of batch processes in the form of linear optimal control formulations of optimizing control. The suggested optimization methodology for batch operation optimization comprises a definition of optimal batch operation as a constrained Linear Program and an Iterative Operation Optimization formulation for implementation of the optimal batch operation. The Iterative Operation Optimization formulation guarantees optimal batch operation. For batch processes that can only be locally approximated a procedure is given, which monitors the validity of the local approximation and re-estimates it if/when it becomes invalid. This procedure is unlikely to converge to the validity region of a local approximation, and should thus be monitored closely and stopped if the batch process performance becomes consistently decreasing. To ensure reproducible operation it is suggested that the Iterative Operation Optimization formulation is combined with the control methodology suggested for ensuring reproducible batch operation.

The output profile and state estimations are formulated in common for the control and optimization methodologies as a combination of a Kernel Smoother and a Kalman filter. This means that smooth estimates of output profiles are obtained with a Kernel Smoother between batch operations while state estimates are obtained with a Kalman filter during the individual batch operation. The combination of a Kernel Smoother and a Kalman Filter is suggested in part to allow for multiple sampling rates and in part to reduce the covariance of the estimates. In addition to the combination of a Kernel Smoother and a Kalman filter, observability of finite horizon LTV systems is defined.

1.4 Tools

The development of a methodology for identification of an interdependent set of pseudo linear regression model approximations for batch processes, has materialized into a MATLAB toolbox called Grid of Linear Models (GoLM). The GoLM toolbox, which comprises a commented input script template, several mex’ed dynamic link libraries (also available as mexsol’s), and a demo input script, facilitates automated identification of autoregressive moving average models with exogenous inputs (and any special case hereof) of repeated finite horizon and stationary periodic systems. In addition, the GoLM toolbox includes mex’ed
dynamic link libraries (also available as mexsol’s) with a Kalman filter and tools for initialization of model predictive controllers. The GoLM toolbox is neither validated, documented, nor supported, and the author can not be held liable for any damage and/or loss of value resulting from use hereof.

1.5 Publications

A methodology (a slightly reduced version of the modeling methodology presented in chapter 2 of the present thesis) for data driven time series modeling of batch processes has been submitted for publication (Bonné and Jørgensen, 2005). The methodology develops a dependent model set that approximates the nonlinear and non-stationary behavior of batch processes. It is shown how such a dependent model set can be efficiently identified from historical process data, and how it may be applied in the design of model-based tools such as Iterative Learning Control, Model Predictive Control and the Kalman filter. The methodology is applied to industrial sugar crystallization.

Other contributions to and application examples on optimal and reproducible operation of batch processes (various parts of the methodologies and examples presented in chapters 2 – 4 of the present thesis) have been published in peer-reviewed collections (Jørgensen et al., 2005) and conference proceedings (Bonné and Jørgensen, 2001b, 2002) and presented at several international conferences (Bonné and Jørgensen, 2000, 2001a,c,d, 2003a,b, 2004, Bonné et al., 2004).

1.6 Outline

The remaining part of this thesis falls naturally into two main parts. First, the proposed modeling methodology is presented in chapter 2, and then the proposed control and optimization methodologies are presented in chapter 3. These two main parts are followed by a few application examples in chapter 4 and finally conclusions and discussion are given in chapter 5.
When in pursuit of optimal and reproducible operation of batch processes, the limiting factor is often the lack of reliable models. This chapter presents a methodology for rapid acquirement of discrete-time state space model representations of batch processes based on their historical operation data. The chapter is organized as follows: First the development and parameterization of a batch process model are presented in section 2.1. Then in section 2.2 identification of these batch process models is discussed and efficient algorithms are given for optimal model identification. In section 2.3 the combined methodology presented in sections 2.1 and 2.2 is applied and a model of an industrial vacuum pan used for sugar crystallization, is obtained. Finally, conclusions are given in section 2.4.

Most often the complex and nonlinear dynamics of continuously operated processes can be approximated with a moderate set of local Linear Time-Invariant (LTI) models, each of which describes the process behavior within a characteristic region in the operation window (e.g. Johansen and Foss, 1995; Banerjee et al., 1997). These regions wherein the behavior is described by local models will often be delimited by a set of active constraints (e.g. Bemporad et al., 2002). For batch and semi-batch processes (from here on, batch will cover both batch and semi-batch processes) however, the set of active constraints will change as the batch progresses. In fact, to operate a batch process in an optimal fashion, a specific sequence of constraints is tracked during operation (e.g. Visser et al., 2000). This means that the transition conditions between the locally approximated characteristic regions also need to be modeled to provide a complete description of batch operation. Furthermore, even if specific sets of constraints were active for longer periods; local LTI models can not be expected to describe the time variation due to nonlinear behavior induced by changing hold-ups and/or compositions.

However, the periodic nature and the finite horizon of batch processes from which observations are collected from a grid of sample points in time, make it possible to model the evolution between two consecutive sample points in a batch with local LTI models.

Assumption 2.1

The changes in the input/output behavior of batch processes, will occur at approximately the same point in (operation) time in every batch.
Figure 2.1: The three dimensions of batch data, i.e., output level, time, and batch index. The circles represent the sample points. It is visualized how measurements from the same sample point can be collected from consecutive batches, thus facilitating identification of a grid-point model for each sample point interval.

Such local models valid between two consecutive points in the sample grid, are labeled grid-point models. In this fashion, both the time variation within the characteristic regions and the transitions between these may be approximated with a grid of grid-point models. Thus, such a grid-point model set gives a complete description of a batch. The finite horizon of batch processes means that the model set will be finite. The periodic way in which the same recipe is repeated batch after batch means that several measurements from the individual sample points are available for identification. That is, the time evolution of a process variable is measured/sampled at specific sample points during the batch operation and as the batch operation is repeated several times, then several measurements are collected from every sample point. With multiple data points/measurements from one specific sampling interval a grid-point model can be identified for this interval. Explicitly, in addition to the time dimension, data from batch processes also evolve in a batch index or batch number dimension. This additional dimension in batch data is illustrated in Figure 2.1.
2.1 Multiple ARMAX Modeling

Given the above discussion, batch processes are modeled with sets of dynamic grid-point LTI models. Such a set of grid-point LTI models could also be referred to as one Linear Time Varying (LTV) batch model. These grid-point LTI models can be parameterized in a number of ways – e.g. as Output Error (OE) models, AutoRegressive models with eXogenous inputs (ARX), State Space (SS) models, etc. In the present contribution an AutoRegressive Moving Average model with eXogenous inputs (ARMAX) parameterization was chosen. This choice of parameterization offers a simple multivariable system description with a moderate number of model parameters.

As operation of a batch progresses, different inputs and outputs may be used depending on the current phase of the batch and hence in order to model batch operation it is convenient to define the following variables and references for each time step $t$:

- Input variable $u_t \in \mathbb{R}^{n_u(t)}$ with reference $\bar{u}_t \in \mathbb{R}^{n_u(t)}$
- Output variable $y_t \in \mathbb{R}^{n_y(t)}$ with reference $\bar{y}_t \in \mathbb{R}^{n_y(t)}$
- Disturbance variable $w_t \in \mathbb{R}^{n_y(t)}$

Using an ARX model parameterization, the output deviation from the reference $\bar{y}_t - y_t$ at time $t$ may be given as a weighted sum of $n_A(t)$ past output deviations and $n_B(t)$ past input deviations

$$\begin{align*}
\bar{y}_t - y_t &= -a_{t,t-1} (\bar{y}_{t-1} - y_{t-1}) - \ldots \\
&\quad - a_{t,t-n_A(t)} (\bar{y}_{t-n_A(t)} - y_{t-n_A(t)}) \\
&\quad + b_{t,t-1} (\bar{u}_{t-1} - u_{t-1}) + \ldots \\
&\quad + b_{t,t-n_B(t)} (\bar{u}_{t-n_B(t)} - u_{t-n_B(t)}) + w_t
\end{align*}
$$

where $n_A(t), n_B(t) \in [1, \ldots, t]$ are the grid-point ARX model orders and $a_{i,j} \in \mathbb{R}^{n_y(i) \times n_y(j)}$ and $b_{i,j} \in \mathbb{R}^{n_y(i) \times n_u(j)}$ are the structured grid-point ARX model parameter matrices. The model parameter matrices are structured because the model orders

$$\begin{align*}
{n_A(t)} &= \max \left\{ n_A(i,j,t) | i, j = 1, \ldots, n_y(t) \right\} \\
{n_B(t)} &= \max \left\{ n_B(i,j,t) | i = 1, \ldots, n_y(t), j = 1, \ldots, n_u(t) \right\}
\end{align*}
$$

are the highest model orders at time $t$, for notational simplicity. Note, as the grid point intervals are modeled with individual grid-point models, the sample points do not have to be equidistantly spaced in time. Let $N$ be the batch length in terms of number of samples, and define the input $u$, output $y$. 

shifted output $y^0$, and disturbance $w$ profiles as

$$
\begin{align*}
  u &= [ u'_0 \ u'_1 \ \ldots \ u'_{N-1} ]' \\
  y &= [ y'_1 \ y'_2 \ \ldots \ y'_N ]' \\
  y^0 &= [ y'_0 \ y'_1 \ \ldots \ y'_{N-1} ]' \\
  w &= [ w'_1 \ w'_2 \ \ldots \ w'_N ]'
\end{align*}
$$

(2.3)

Note, the model variables $y$ and $u$ may well be basis functions of the process variables, and that not all initial conditions $y_0$ are measurable and/or physically meaningful — e.g. off-gas measurements. Thus the ARX model set may be expressed in matrix form

$$
\bar{y} - y = -A(\bar{y}_0 - y_0) + B(\bar{u} - u) + w
$$

(2.4)

where $A, B$ are structured lower block triangular matrices. To exemplify, if it is assumed that $n_A(t) = n_A = i$, then $A$ has the following structure

$$
A = 
\begin{bmatrix}
  a_{1,0} & & & \\
  \vdots & \ddots & & \\
  a_{i,0} & \cdots & a_{i,i-1} & \\
  & \ddots & \ddots & \\
  & & a_{N,N-i-1} & \cdots & a_{N,N-1}
\end{bmatrix}
$$

The profile $w$ is a sequence of disturbance terms caused by several effects, which include bias in the reference input profile $\bar{u}$, the effect of process upsets, modeling errors from linear approximations, and errors due to bias in transition times between sets of active constraints. Hence, the disturbance $w$ contains contributions from both batchwise persistent disturbances, such as recipe/input bias, model bias, and erroneous sensor readings, as well as from random disturbances, which occur with no batchwise correlation. It thus seems reasonable to model the disturbance profile $w$ with a random walk model with respect to the batch index $k$

$$
w_k = w_{k-1} + \Delta w_k
$$

(2.5)

where the increment disturbance profile $\Delta w_k$ is modeled with a Moving Average (MA) model with respect to time

$$
\Delta u_{k,t} = v_{k,t} + c_{t,t-1}v_{k,t-1} + \cdots + c_{t,t-nC(t)}v_{k,t-nC(t)}
$$

(2.6)

with model order $nC(t) \in [0, \ldots, t - 1]$

$$
nC(t) = \max\{nC(i,j,t) | i, j = 1, \ldots, n_y(t)\}
$$

(2.7)

In matrix form the disturbance model is expressed as

$$
\Delta w_k = Cv_k
$$

(2.8)
where the sequence \( v_k = [v_{k,1}', v_{k,2}', \ldots, v_{k,N}'] \), \( v_{k,t} \in \mathbb{R}^{n_y(t)} \), represents batch-wise non-persistent disturbances that are assumed to be zero-mean, independent and identically distributed.

**Assumption 2.2**

The model approximation errors are partly due to persistent (from batch to batch) modeling and batch operations model bias, and partly due to zero-mean random disturbances.

Considering the difference between two successive batches

\[
\Delta y_k = y_k - y_{k-1} = A(y_k^0 - y_{k-1}^0) - B(u_k - u_{k-1}) + w_k - w_{k-1} \quad (2.9)
\]

A batch ARMAX model (2.9) that is independent of the reference profiles \((\bar{y}, \bar{u})\) and batchwise persistent disturbances has been obtained. With such a batch ARMAX model the path is prepared for multivariable, model-based monitoring, control, optimization, and of course simulation.

During the model derivation above it was assumed that the outputs are known. This is however not the case in practice, where only a sequence \( z_k = [z_{k,0}', z_{k,1}', \ldots, z_{k,N}'] \), \( z_{k,t} \in \mathbb{R}^{n_y(t)} \) of noisy observations of the outputs is available

\[
z_k = [y_{k,0}' \quad y_k'] + \epsilon_k \quad (2.10)
\]

where \( \epsilon_k = [\epsilon_{k,0}', \epsilon_{k,1}', \ldots, \epsilon_{k,N}'] \), \( \epsilon_{k,t} \in \mathbb{R}^{n_y(t)} \) is a sequence of measurement noise terms that are assumed to be zero-mean, independent and identically distributed.

### 2.1.1 Application Specific Models

The batch ARMAX model (2.9) may be converted into different representations dependent on the particular application task. If the task at hand is to predict (or simulate) the behavior of a batch before it is started the following form is convenient

\[
\Delta y_k = A\Delta y_k^0 - B\Delta u_k + Cv_k = \hat{A}\Delta y_{k,0} + [ \hat{A} \quad 0 ] \Delta y_k - B\Delta u_k + Cv_k = E^{-1}\hat{A}\Delta y_{k,0} - E^{-1}B\Delta u_k + E^{-1}Cv_k = H\Delta y_{k,0} - G\Delta u_k + Fv_k \quad (2.11)
\]

where \( \hat{A} \) and \( \hat{A} \) are partitions of \( A \) corresponding to \( y_0 \) and \( [y_1', y_2', \ldots, y_{N-1}'] \), respectively, and

\[
E = I - [ \hat{A} \quad 0 ] \quad (2.12)
\]
The matrix $E$ is obviously lower-triangular and positive definite and its inverse can thus always be efficiently computed by forward-substitution. Note that the disturbance matrix $F$ models the propagation of batchwise non-persistent disturbances — including batchwise non-persistent model-plant mismatch. The initial condition $\Delta y_{k,0}$ can be considered as either an input/control variable or a disturbance. The distinction between the two possibilities will of course depend on the information on and control of the outputs prior to a batch. If the initial conditions are considered disturbances, it is necessary to model them. Then the initial output deviation from the reference is also modeled as a random walk with respect to batch index

$$w_{k,0} = w_{k-1,0} + v_{k,0}$$

$y_0 - y_{k,0} = w_{k,0}$

where $v_{k,0}$ is assumed zero-mean, independent of, and distributed identically with $v_k$. Which means that the initial condition

$$\Delta y_{k,0} = y_{k,0} - y_{k-1,0}$$

$$= w_{k,0} - w_{k-1,0}$$

$$= v_{k,0}$$

is given as a zero-mean random variable.

The form (2.11) above is also convenient for the task of classification/monitoring (e.g. normal or not) of a batch after it has been completed. Furthermore, the form (2.11) can be used to determine open-loop optimal recipes in the sense of optimizing an objective for the batch. If such an objective is to minimize the deviations $e = [e_1', e_2', \ldots, e_N']'$, $e_t \in \mathbb{R}^{n_y(t)}$, from a desired trajectory $\bar{y}$, then (2.11) can be modified into

$$e_k = \bar{y} - y_k$$

$$= \bar{y} - y_{k-1} - \Delta y_k$$

$$= e_{k-1} - H\Delta y_{k,0} + G\Delta u_k - Fv_k$$

There are two important points to be made about the trajectory tracking model form (2.15). First of all, as the error profile $e_k$ in batch $k$ depends on the error profile $e_{k-1}$ from batch $k-1$, the effects of the batchwise persistent disturbances are integrated with respect to batch index. This means that a properly designed controller will be able to reject the effects of the batchwise persistent disturbances asymptotically with respect to batch index — e.g. removing the effects of recipe and model bias. Secondly, given the above mentioned asymptotic behavior and as the control moves/actions generated by such a controller are deviations from the control/input profile realized in the previous batch, the control actions due to batchwise persistent disturbances will converge asymptotically to zero with respect to batch index. In literature (e.g. Moore and

\[\text{Note that the matrices } H, G, \text{ and } F \text{ can in fact be computed without inverting } E -- \text{ see page 78.}\]
it is said that the controller learns to reject the batchwise persistent disturbances — i.e. the resulting controller is an Iterative Learning Control (ILC) scheme. A more precise formulation would be that both output and input errors are modeled using integrators with respect to the batch index. The trajectory tracking model representation (2.15) is similar to that of Lee et al. (2000), but the model representation differ significantly since (2.15) includes the effect of initial conditions \((H \Delta y_k, 0)\) and disturbance propagation \((F v_k)\). Another important difference is that (2.15) does not include double dependence on the batchwise persistent disturbances — i.e., the trajectory tracking model representation (2.15) only includes the batchwise persistent disturbances as represented by \(e_{k-1}\) and not as both the part of \(e_{k-1}\) caused by the batchwise persistent disturbances and the batchwise persistent disturbances themselves.

The two forms (2.11) and (2.15) of the batch ARMAX model above are applicable to off-line or inter-batch type applications. For on-line estimation, monitoring, feedback control, and optimization however, it is convenient to use a SS realization of the batch ARMAX model. To achieve such a realization in a simple manner it is desirable to simplify the batch ARMAX model structure with the assumption that the number of outputs is constant \(n_y(t) = n_y\) for \(t = 0, 1, \ldots, N\). In an observer canonical form, which is structurally a minimal realization, the SS realization is given as

\[
x_{k,t} = A_t x_{k,t-1} + B_t \Delta u_{k,t-1} + E_t v_{k,t} \\
\Delta y_{k,t} = C x_{k,t}
\]

(2.16)

with the SS model dimension \(n_x = n_y \max (n_i(t) \mid 1 \leq t \leq N, i = A, B, C)\) and initial condition

\[
x_{k,0} = C' \Delta y_{k,0}
\]

(2.17)

The SS model matrices \(A_t \in \mathbb{R}^{n_x,n_x}\), \(B_t \in \mathbb{R}^{n_x,n_u(t)}\), \(E_t \in \mathbb{R}^{n_x,n_y}\), and \(C \in \mathbb{R}^{n_y,n_x}\) contain the corresponding block columns in the batch ARMAX model \((A, B, C)\). To exemplify, assume that \(n_A(t) = n_B(t) = n_C(t) + 1 = 3\) and the SS model matrices will be given as

\[
A_t = \begin{bmatrix}
a_{t,t-1} & I & 0 \\
a_{t+1,t-1} & I & 0 \\
a_{t+2,t-1} & 0 & 0
\end{bmatrix}, \quad E_t = \begin{bmatrix}
I \\
a_{t+1,t} \\
a_{t+2,t}
\end{bmatrix}
\]

\[
B_t = \begin{bmatrix}
-b_{t,t-1} \\
-b_{t+1,t-1} \\
-b_{t+2,t-1}
\end{bmatrix}, \quad C = \begin{bmatrix}
I & 0 & 0
\end{bmatrix}
\]

Just as (2.11), the SS model form (2.16) is convenient for prediction, monitoring, and optimization type applications, and it facilitates on-line implementation of such applications. Furthermore, the SS model form (2.16) is particularly...
well suited for closed-loop or feedback control applications. For tracking control applications the SS model form (2.16) can be modified to
\[
\begin{align*}
x_{k,t} &= A_t x_{k,t-1} + B_t \Delta u_{k,t-1} + E_t v_{k,t} \\
e_{k,t} &= \bar{y}_t - \bar{y}_{k,t} - C_t x_{k,t-1, t} - \Delta y_{k,t} \\
e_{k-1, t} &= C_{k-1, t} - C_t x_{k,t}
\end{align*}
\] (2.18)

Following the discussion above, a multivariable feedback controller properly designed using the trajectory tracking SS model form (2.18), will be able to reject the effects of the batchwise persistent disturbances asymptotically with respect to batch index. A controller designed to reject disturbances with respect to time within a batch will hence also, due to the integration of output and input errors into the model framework, asymptotically reject the effects of batchwise persistent disturbances with respect to batch index.

2.2 Identification

With the batch ARMAX model (2.9) derived above, the parameterization of the batch model is in place, however the model orders and the model parameters still need to be determined from process data. One seemingly major drawback of the proposed parameterization is the immense dimensionality of the resulting set of models. In practice this immense dimensionality will render any standard Least Squares (LS) identification problem rank-deficient, and the resulting model will generalize poorly. It turns out however, that the conditioning of the identification problem improves when the grid-point models are mutually dependent and that the stronger the dependency is between the grid-point models, the better the conditioning of the identification problem becomes. In fact, it is this correlation between the forced similarity of neighboring grid-point models and predictive capability of the model set that forms the basis for the identification framework presented below.

The present section on model identification, comprises three subsections. The first subsection presents a Kernel Smoother based procedure for data preparation. The second subsection presents efficient algorithms for model parameter estimation given sparse data sets. Finally, the third subsection presents a selection of quality measures for model assessment and algorithms for optimal model selection.

The first step in any modeling work should of course be the definition of the modeling objective — what problems are to be solved? and are they to be solved through monitoring and/or control?, how is the most value added with the least effort?, etc. Once the modeling objective is well defined, the appropriate inputs and outputs can be chosen as basis functions, e.g. \{x, y\}, can be expanded with basis functions of the basis, e.g. to \{x, y, x^2, xy, \exp(y), \ldots\}, offering considerable flexibility to linear regression modeling.
or sensors. Should the modeling objective be monitoring, control, and/or optimization of product quality variables only measured sparsely off-line, then these variables can once they have been re-sampled appropriately, be included as on-line measurements. If all the required inputs and outputs are chosen among the existing process variables, then historical process data can be used for model identification. Otherwise, the necessary new actuators and/or sensors are installed on the process and a few subsequent batch runs are planned and their data collected.

2.2.1 Data Preparation

Having selected the inputs and outputs and obtained the data, these must be filtered and re-sampled in accordance with the modeling objective. According to the process dynamics, the data is re-sampled sufficiently densely to comply with the modeling objective. It must be noted that the sample intervals do not have to be of constant length, which means that the samples can be placed in time, where they do the most good in terms of approximating the system behavior. Once outliers have been removed from the data, a Kernel smoother can be used to 1) re-sample the data, 2) remove most of the high frequency measurement noise, and 3) reconstruct missing data sequences. In the present work the following three step data preparation procedure is proposed:

1. Data outliers and complete batches that are inconsistent with normal operation, are removed by visual inspection of the data. If data from several identical vessels is used, then incompatible sensor calibration and/or scaling is corrected if possible. It may also be necessary to move the origin of some signals in order to avoid negative values of strictly positive variables close to zero — this is however only necessary when it is desirable to work with the logarithm of these signals.

2. The data is re-sampled with a Kernel smoother without reconstructing missing data sequences. In the same operation, most of the high frequency measurement noise is removed from data. Care should be taken however, to avoid erroneously removing system dynamics while filtering measurement noise. Any remaining measurement noise is taken care of in the model estimation procedure described below. It may be desirable to re-sample the logarithm of some signals in order to avoid smoothing or “rounding” the corners of abrupt step changes in the values of these signals. E.g. heuristically optimized feed rate profiles for semi- or fed-batch operation will often comprise several step changes and working with the logarithm of such signals allows for high frequency noise filtering without excessive smoothing of the step changes themselves.

3. Missing data sequences are reconstructed by re-sampling the difference profile between a re-sampled batch with missing data sequences and the previous or subsequent re-sampled batch with no missing data sequences,
Figure 2.2: This figure illustrates the difference between reconstructing missing data sequences in “regular” versus batch increment deviation variables. The true variables (solid lines) and their noisy realizations (thin solid lines) from two consecutive batches, one of which has a missing sequence of data, are plotted along with their Kernel Smoother estimates based on the two batches separately (dash-dotted lines), and the Kernel Smoother estimate based on the deviations (dashed line) between the two batches.

with a Kernel smoother. This is done under the assumption that the difference profile will have lower order dynamics than the original profiles. Missing data sequences often originate from sequences of outliers and/or temporary failures in the data acquisition system. Again, care should be taken to avoid erroneously removing system dynamics in the process of reconstructing missing data sequences. An illustrative example of why reconstruction should be performed in deviation variables, is given in figure 2.2. In this example, the local (second order) polynomial regression models are unable to satisfactorily extrapolate (i.e., when all of the (100) Nearest Neighbors are on the same side of the missing sequence) the higher order system. The (first order) difference profile is however, satisfactorily extrapolated by the Kernel Smoother.
Kernel Smoother

In industry, the process variables \( \tilde{z}(p) \in \mathbb{R} \) for \( p = 1, \ldots, N_p \), are most often logged individually at times \( \tilde{T}(k, i, p) \), giving \( N_z(k, p) + 1 \) observations of variable \( \tilde{z}_k(p) \) in batch \( k \). What is needed however, is up to \( N + 1 \) noise free observations of the variables \( \bar{z}(p) \) for \( p = 1, \ldots, N_p \) at times \( T(t) \) in the \( N_B \) batches available for identification — see figure 2.1 on page 10, in which one process variable (\( N_p = 1 \)) is sampled \( N = 10 \) times in \( N_B = 5 \) batches. These noise free or expected observations can be estimated using local polynomial regression of order \( d(p) \)

\[
\hat{\bar{z}}_{k,t}(p) = \hat{\alpha}(k, t, p) + \sum_{j=1}^{d(p)} \hat{\beta}_j(k, t, p) T(t)^j
\]

for \( p \in M_t \), where \( M_t \) is the set of variables considered at time \( t \). The local parameters \( \hat{\alpha}(k, t, p) \in \mathbb{R} \) and \( \hat{\beta}_j(k, t, p) \in \mathbb{R} \) are the arguments that minimize

\[
\min_{\alpha(k, t, p), \{\hat{\beta}_j(k, t, p)\}_{j=1}^{d(p)}} \left[ \sum_{i=0}^{N_z(k, p)} K_{\gamma(p)}^p(T(t), \tilde{T}(k, i, p)) (\tilde{z}_{k,i}(p) - \hat{z}_{k,i}(p))^2 \right]

s.t. \( \tilde{z}_{k,i}(p) = \alpha(k, t, p) + \sum_{j=1}^{d(p)} \beta_j(k, t, p) T(t)^j \)

If the profile of process variable \( \tilde{z}_k(p) \) in batch \( k \) is defined as

\[
\tilde{z}_{k,p} = [\tilde{z}_{k,0}(p), \tilde{z}_{k,1}(p), \ldots, \tilde{z}_{k,N_z(k, p)}(p)]'
\]

then the estimation problem (2.19)–(2.20) can be given explicitly as

\[
\hat{z}_{k,t}(p) = t_{p,t} \left( \tilde{T}'_{k,p} K_{\gamma(p),k,t} \tilde{T}_{k,p} \right)^{-1} \tilde{T}'_{k,p} K_{\gamma(p),k,t} \tilde{z}_{k,p}
\]

where

\[
K_{\gamma(p),k,t} = \left[ \begin{array}{cccc} K_{\gamma(p)}^p(T(t), \tilde{T}(k, 0, p)) & \cdots & \cdots & K_{\gamma(p)}^p(T(t), \tilde{T}(k, N_z(k, p), p)) \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ K_{\gamma(p)}^p(T(t), \tilde{T}(k, N_z(k, p), p)) & \cdots & \cdots & K_{\gamma(p)}^p(T(t), \tilde{T}(k, N_z(k, p), p)) \end{array} \right]
\]

and

\[
t_{p,t} = \left[ \begin{array}{cccc} 1 & T(t)^1 & \cdots & T(t)^{d(p)} \end{array} \right]
\]

\[
t_{k,p,i} = \left[ \begin{array}{cccc} 1 & \tilde{T}(k, i, p)^1 & \cdots & \tilde{T}(k, i, p)^{d(p)} \end{array} \right]
\]

\[
\tilde{T}_{k,p} = \left[ \begin{array}{cccc} \tilde{T}_{k,p,0} & \tilde{T}_{k,p,1} & \cdots & \tilde{T}_{k,p,N_z(k, p)} \end{array} \right]
\]

(2.24)
Note, efficient implementation of explicit Least Squares solutions such as \((2.22)\), are dealt with in detail beginning on page \(29\). If it is further assumed that process variable \(\bar{z}(p)\) will be used throughout the batch, then the estimated profile of variable \(\hat{\bar{z}}(p)\) in batch \(k\) is given explicitly as

\[
\hat{\bar{z}}_k(p) = \left[ \hat{\bar{z}}_{k,0}(p) \quad \hat{\bar{z}}_{k,1}(p) \quad \ldots \quad \hat{\bar{z}}_{k,N}(p) \right]'
\]

\[
= \left[ s'_{k,p,0} \quad s'_{k,p,1} \quad \ldots \quad s'_{k,p,N} \right]' \bar{z}_k(p) \\
= S_{k,p} \tilde{z}_k(p) \\
\]

(2.25)

The type (e.g., Epanechnikov, tri-cube, or Gaussian) and bandwidth \(\gamma(p)\) of the Kernels \(K_p^{\gamma(p)}\) are selected individually for the process variables. The Kernel bandwidth \(\gamma(p)\) and the local polynomial order \(d(p)\) determine the trade-off between the bias and variance of the estimates — i.e., the smoothness of the estimates. The Kernel type also influences the bias and variance of the estimates. The local polynomial order \(d(p)\) is particularly sensitive at the boundaries, but the Kernel bandwidth \(\gamma(p)\) is however the most sensitive tuning parameter in Kernel smoothing. A detailed discussion of tuning of the Kernel bandwidth \(\gamma(i)\) falls outside this thesis, but the Kernel bandwidth \(\gamma(i)\) should be chosen conservatively to avoid removing system dynamics.

Let the true noise free observation \(\bar{z}_k = [\bar{z}_k,0, \ldots , \bar{z}_k,N]'\), with \(\bar{z}_k,t \in \mathbb{R}^{(n_y(t)+n_u(t))}\) be given as

\[
\bar{z}_k = \hat{z}_k + \omega_k \\
\]

(2.26)

where \(\hat{z}_k\) is the estimated observation and \(\omega_k = [\omega_{k,0}', \omega_{k,1}', \ldots , \omega_{k,N}']\), \(\omega_k,t \in \mathbb{R}^{(n_y(t)+n_u(t))}\) is a sequence of estimation errors. The estimation error \(\omega_k\) will consist of both random errors and systematic errors such as the height of a characteristic peak being underestimated due to excessive smoothing due to too high bandwidth and/or trimming the hills and filling the valleys due to too low local regression order. The estimation error \(\omega_k\) can thus be modeled with a random walk with respect to the batch index \(k\)

\[
\omega_k = \omega_{k-1} + \nu_k \\
\]

(2.27)

where \(\nu_k = [\nu_{k,0}', \nu_{k,1}', \ldots , \nu_{k,N}']\), \(\nu_k,t \in \mathbb{R}^{(n_y(t)+n_u(t))}\), represents a sequence of batchwise non-persistent estimation errors that are assumed to be zero-mean.

**Assumption 2.3**

The estimation errors are partly due to persistent (from batch to batch) bias resulting from the choice of Kernel, bandwidth and local regression order and partly due to zero-mean random disturbances.

The expected difference between two successive batches is then given as the
difference between their respective estimates

\[
E\{\Delta \hat{z}_k\} = E\{\hat{z}_k - \hat{z}_{k-1}\} \\
= \hat{z}_k - \hat{z}_{k-1} + E\{\omega_k - \omega_{k-1}\} \\
= \Delta \hat{z}_k + E\{v_k\} \\
= \Delta \hat{z}_k
\]  

(2.28)

That is, the estimated difference profile \(\Delta \hat{z}_k\) is an unbiased estimate of its mean \(E\{\Delta \bar{z}_k\}\). The expected output and input difference profiles which are all contained in \(\Delta \bar{z}_k\), are thus given as

\[
E\{\Delta y_k\} = \Delta \hat{y}_k \\
E\{\Delta y_k^l\} = \Delta \hat{y}_k^l \\
E\{\Delta u_k\} = \Delta \hat{u}_k
\]  

(2.29)

In fact, the expected difference between any two observations

\[
E\{\bar{z}_k - \bar{z}_{k-l}\} = \hat{z}_k - \hat{z}_{k-l} + E\{\omega_k - \omega_{k-l}\} \\
= \hat{z}_k - \hat{z}_{k-l}
\]  

(2.30)

is given as the difference between their respective estimates. Furthermore, as the disturbance profile \(\tilde{v}_{k,l} = w_k - w_{k-l}\) is zero-mean for any \(l\), the batch ARMAX model (2.9) can describe the difference between any two batches — i.e., the difference profiles between any two batches can be used for identification.

### 2.2.2 Parameter Estimation

Several suggestions to how (sets of) LTI or (periodic) LTV models should be identified from data can be found in literature (e.g., Akaike, 1975; Larimore, 1990, 1996; Verhaegen, 1994; Verhaegen and Yu, 1995; van Overschee and de Moor, 1994; Nomikos and MacGregor, 1994; Wenfu et al., 1995; Chen and McAvoy, 1998). All these authors employ some or other coefficient shrinkage or subspace method to improve the conditioning of the identification problem and hence reduce the variance of the model parameter estimates. Comparisons of some of these methods may be found in Negiz and Çınar (1997); Kalivas (2001); Juricek et al. (2001); Hastie et al. (2001); Chiang et al. (2001). In a modeling framework similar to the one employed here, Dorsey and Lee (2003) proposed to estimate both a set of Finite Impulse Response (FIR) models and a LTI SS model using Principle Component Analysis (PCA) and N4SID (van Overschee and de Moor, 1994), but such a set of non-parametric FIR models is obviously non-parsimonious and would exhibit poor predictive capabilities (Larimore, 1996). Simoglou et al. (2002) suggested estimating a set of independent, overlapping local LTI SS models using Canonical Variant Analysis (CVA). Instead the present contribution proposes estimating a set of interdependent grid-point LTI ARMAX models using a novel interpretation of Tikhonov Regularization.
The batch ARMAX model (2.9) can be formulated as a pseudo linear regression model

\[ \Delta y_k = \Delta x_k \theta + v_k \]  

(2.31)

where \( \Delta x_k = \Delta x_k(\Delta y^0_k, \Delta u_k, v_k) \in \mathbb{R}^{n_x \times n_y} \) is a structured regressor matrix with past outputs, inputs and disturbances and \( \theta = (A, B, C) \in \mathbb{R}^{n_x} \) is a column parameter vector with the model parameters from the batch ARMAX model. Taking the expectation of the linear regression model (2.31) and recalling (2.29) we find that

\[ \Delta \hat{y}_k = E\{\Delta y_k\} = E\{\Delta x_k\} \theta + E\{v_k\} = \Delta \hat{x}_k \theta \]  

(2.32)

with \( \Delta \hat{x}_k = \Delta x_k(\Delta \hat{y}^0_k, \Delta \hat{u}_k, \hat{v}_k) \) and where \( \hat{v}_k \) is an estimate of the disturbance or One-Step-Ahead (OSA) prediction error profile. This means that, if the process variable estimation error model (2.27) is a valid approximation, then estimation of model parameters from the pretreated data will give unbiased model parameter estimates. A multivariate weighted Least Squares (wLS) estimate of the model parameters in the linear regression model (2.31) is found by solving the following minimization problem

\[
\min_{\theta} J_{wLS} \\
\text{s.t. } J_{wLS} = \|\Delta \hat{y}_k - \Delta \hat{x}_k \theta\|^2_W \\
= (\Delta \hat{y}_k - \Delta \hat{x}_k \theta)^\prime W (\Delta \hat{y}_k - \Delta \hat{x}_k \theta) \\
= \sum_{t=1}^{N} \left((\Delta \hat{y}_{k,t} - \Delta \hat{x}_{k,t} \theta_t)^\prime W_t (\Delta \hat{y}_{k,t} - \Delta \hat{x}_{k,t} \theta_t)\right) 
\]  

(2.33)

where \( \| \cdot \| \) denotes the 2-norm and \( W \) is a block diagonal weighting matrix with symmetrical, positive definite, block elements \( W_t \in \mathbb{R}^{n_y \times n_y} \). Note that \( \theta_t \in \mathbb{R}^{n_x} \) for \( t = 1, \ldots, N \), are the model parameters in the local grid-point models and that they are mutually independent in the estimation problem (2.33). The weighting matrices \( W_t \) for \( t = 1, \ldots, N \), can be designed in a number of ways, e.g. as:

- The inverse of the covariance \( \Sigma_t \in \mathbb{R}^{n_y \times n_y} \), of the one-step-ahead prediction error \( v_t \) at times \( t = 1, \ldots, N \). I.e.,

\[ W_t = \Sigma_t^{-1} \]  

(2.34)

This choice of weighting matrices will produce the so called Minimum Variance Unbiased Estimator (MVUE) of the estimation problem (2.33).
• The inverse of the covariance of the Kernel smoother estimates (see pages 20 and 97) at times $t = 1, \ldots, N$. I.e.,

$$W_{k,t}(i,i) = 
\left(\|s_{k,i,t}\|_2 + \|s_{k-1,i,t}\|_2\right) R_{\epsilon_t}(i,i)^{-1} \quad (2.35)$$

for $i = 1, \ldots, n_y(t)$, where $R_{\epsilon_t}$ is the covariance of the measurement noise $\epsilon_t$ at time $t$.

• The inverse of maximum output deviation between two batches. E.g., if the same $n_y = n_y(t), t = 1, \ldots, N$, outputs are used throughout the batch, then

$$W_t(i,i) = \left(\max\{|\Delta \hat{y}_{k,j}(i)| \mid k = 1, \ldots, N_B; j = 1, \ldots, N \right) \right)^{-1} \quad (2.36)$$

for $i = 1, \ldots, n_y$ and $t = 1, \ldots, N$.

The latter two choices of diagonal weighting matrices are particularly tractable from a computational point of view — see page 34.

The linear estimation problem (2.33) is in general underdetermined, i.e., $n_p < n_y$, and will thus in general have infinitely many solutions. The minimum norm (or variance) and unique estimator $\hat{\theta}$ is however found by solving (2.33) with Principle Component Regression (PCR) (also referred to as Truncated Singular Value Decomposition (TSVD)) or Ridge Regression (RR) (see e.g. Huffel and Vandewalle [1991], Hansen [1996], Kailath et al. [2000], Hastie et al. [2001]). Both PCR and RR weights the principle components in an orthogonal decomposition of the input space according to their contribution to the explanation of variance in the input space. But while PCR selects derived input directions by assigning binary weights (0 or 1) to the principle components, RR shrinks the coefficients (or parameters) by assigning continuous weights to the principle components. An alternative to PCR is Partial Least Squares (PLS), which assigns binary weights to the principle components of a orthogonal decomposition of the cross-covariance between the output and input spaces, and not the covariance of the input space as in PCR (see e.g. Hastie et al. [2001], Chiang et al. [2001]). In common, for PCR and PLS is however the truncated Singular Value Decomposition (SVD). Of the three regression methods, i.e., PCR, PLS and RR, RR is reportedly the most numerically stable and the method that produces model estimates with the best predictive capabilities (Hastie et al. [2001]). Using RR the minimum norm estimator $\hat{\theta}_{RR}$ is given as

$$\hat{\theta}_{RR}(\hat{v}_k, W, \lambda) = \arg \min_{\theta} \left[ J_{wLS}^{RR} \right]$$

s.t. $J_{wLS}^{RR} = \|\Delta \hat{y}_k - \Delta \hat{x}_k \theta\|_W^2 + \lambda^2\|\theta\|^2 \quad (2.37)$

where the scalar weight $\lambda$ determines the trade-off between the variance of the model parameter estimates and their bias.
To reduce the variance of model parameter estimates without introducing bias, as much data (of sufficiently high quality) as possible should be used for the model parameter estimation. Typically, the available data set of \(N_B = N_{\text{est}}^B + N_{\text{val}}^B + N_{\text{test}}^B\) batches is split up into sets of \(N_{\text{est}}^B\) batches for model parameter estimation, \(N_{\text{val}}^B\) batches for model validation and \(N_{\text{test}}^B\) batches for model testing. The linear system (2.32) is thus augmented as

\[
Y = \begin{bmatrix}
\Delta \hat{y}_1' \\
\Delta \hat{y}_2' \\
\vdots \\
\Delta \hat{y}_{N_{\text{est}}^B}'
\end{bmatrix}' = \begin{bmatrix}
\Delta \hat{x}_1' \\
\Delta \hat{x}_2' \\
\vdots \\
\Delta \hat{x}_{N_{\text{est}}^B}'
\end{bmatrix}' \theta
\]

The linear system (2.38) will however, most likely still be rank-deficient and solving it in a wLS sense

\[
\hat{\theta}_{\text{RR}}(\hat{V}, W, \lambda) = \arg \min_{\theta} \left[ J_{\text{wLS}}^{\text{RR}} \right]
\]

s.t. \( J_{\text{wLS}}^{\text{RR}} = \| Y - X\theta \|^2_W + \lambda^2 \| \theta \|^2 \) (2.39)

where \(\hat{V} = [\hat{v}_1', \hat{v}_2', \ldots, \hat{v}_{N_{\text{est}}^B}]'\) and \(W\) is a block diagonal matrix with block elements \(W\), will still produce model parameter estimates with excessive variance or bias given the regularization weight \(\lambda\). In fact, if it is assumed that \(n_i(t) = n_i\) for \(i = y, u, A, B, C\) and \(a_{t,i}, b_{t,j}\) and \(c_{t,l}\) are dense \(\forall t, i, j, l\), then it would take at least

\[
N_{\text{est}}^B = \frac{1}{2N} \left( n_y n_A(2N - n_A + 1) + n_u n_B(2N - n_B + 1) \right) \\
+ n_y n_C(2N - n_C - 1)
\]

\[
\approx n_y (n_A + n_C) + n_u n_B
\]

batches for (2.39) to be structurally overdetermined and hence even more batches to improve the conditioning and hence reduce the variance of the model parameter estimates for a given level of bias. Such excessive model parameter variance or bias will yield models with poor predictive capabilities (Larimore, 1996). Thus, to improve the predictive capabilities of an estimated model, additional measures must be taken to further reduce the variance (and hence the necessary bias) of the estimated model parameters.

A possible alternative approach to reduce the variance of model parameter estimates is to enforce that the estimated model possesses some desired model properties. One such model property could be that neighboring gridpoint models are analogous in the sense that they exhibit similar behavior. In fact, without this property, the model would constitute a set of independent models and not an entangled grid of interdependent models. Enforcing model properties however, inevitably introduce bias into the model parameter estimates. There will thus be a trade-off between the bias and variance of
the model parameter estimates and this trade-off will determine the predictive
capabilities of estimated models. One coefficient shrinkage based parameter
estimation method that can incorporate model properties into wLS estimates
is Tikhonov Regularization (TR). The derived input direction selection based
Truncated Generalized SVD (TGSVD) regression method [Hansen 1996] could
in principle be an alternative to TR, but in practice the truncated SVD would
be computationally infeasible. It is thus proposed to estimate the model para-
meters by solving the TR problem

$$\hat{\mathbf{\theta}}_{\text{TR}}(\mathbf{V}, W, \mathbf{\Lambda}) = \arg \min_{\mathbf{\theta}} \left( J_{\text{wLS}}^{\text{TR}} \right)$$

subject to

$$J_{\text{wLS}}^{\text{TR}} = \| \mathbf{Y} - \mathbf{X} \mathbf{\theta} \|_W^2 + \| L \mathbf{\theta} \|_{\mathbf{\Lambda}}^2$$

(2.41)

where the structured penalty matrix \( L \) maps the parameter vector \( \mathbf{\theta} \) into the
desired parameter differences and the diagonal weighting matrix \( \mathbf{\Lambda} \) weights
the parameter differences, such that the penalty \( \mathbf{A} \mathbf{\theta} \) is a column vector of
weighted differences between parameters in neighboring grid-point models. The
penalty matrix \( L \) consists of five sub-matrices \( L_i \) for \( i = 1, \ldots, 5 \)

$$L = [ L'_1 \quad L'_2 \quad L'_3 \quad L'_4 \quad L'_5 ]'$$

(2.42)

each of which are individual weighted by block diagonal weighting matrices \( \Lambda_i \)
for \( i = 1, \ldots, 5 \)

$$\Lambda = \begin{bmatrix} \Lambda_1 & & \\ & \ddots & \\ & & \Lambda_5 \end{bmatrix}$$

(2.43)

To reduce the complexity associated with design of the penalty matrix \( L \)
and the weighting matrix \( \Lambda \), the following assumption is introduced at this
point.

- The number of inputs and outputs are constant throughout the batch –
  \( n_y = n_y(t), \quad n_u = n_u(t) \ \forall t \).

This assumption does not limit the applicability of the modeling methodology,
but it will lead to larger estimation problems.

The block diagonal weighting matrices \( \Lambda_m \) for \( m = 1, \ldots, 5 \), each consists of

$$N_c = 2n^2_y + n_y n_u$$

(2.44)
diagonal weighting matrices \( \Lambda_{m,i} \), i.e.,

$$\Lambda_m = \begin{bmatrix} \Lambda_{m}^a & & \\ & \Lambda_{m}^b & \\ & & \Lambda_{m}^c \end{bmatrix}$$

(2.45)

with block elements

$$\Lambda_{m,i}^n = \begin{bmatrix} \Lambda_{m,i}^{n,1} & & \\ & \ddots & \\ & & \Lambda_{m,i}^{n,N_c} \end{bmatrix}$$

(2.46)
MODELING METHODOLOGY

where

\[ \Lambda_{m,i,j}^{a} = \lambda_{m,i,j}^{a} I, \quad i, j = 1, \ldots, n_y \]

\[ \Lambda_{m,i,j}^{b} = \lambda_{m,i,j}^{b} I, \quad i = 1, \ldots, n_y, \quad j = 1, \ldots, n_u \]

\[ \Lambda_{m,i,j}^{c} = \lambda_{m,i,j}^{c} I, \quad i, j = 1, \ldots, n_y \]

(2.47)

and \( \lambda_{m,i,j}^{n} \) are scalar weights. I.e., for every penalty matrix \( L_{m} \) (structured accordingly), every output-output (\( N_{a} = n_{y}^{2} \)), input-output (\( N_{b} = n_{y} n_{u} \)), or disturbance-output (\( N_{c} = n_{y}^{2} \)) connection in the model set is weighted with an individual scalar weight, and there are hence a total of \( 5N_{c} \) weights \( \lambda_{m,i,j}^{n} \).

Each of the sub-matrices \( L_{m} \) for \( m = 1, \ldots, 5 \) penalize violations of specific model (parameter) properties:

- **L1**\( \theta \)** Approximates the 1st order time derivative of the parameters \( \theta \). It thus incorporates the local model interdependency by penalizing the model parameter time evolution. E.g., for the \( A \) matrix this means that the quantities

  \[ (\lambda_{1,i,j}^{a})^{2} (\hat{\alpha}_{t,t-l}(i,j) - \hat{\alpha}_{t+1,t-l-1}(i,j))^{2} \]

  for \( i, j = 1, \ldots, n_y, \quad t = 1, \ldots, N - 1 \) and \( l = 1, \ldots, n_A(i,j,t) \), are penalized.

- **L2**\( \theta \)** Approximates the 2nd order time derivative of the parameters \( \theta \). It thus incorporates the local model interdependency by penalizing non-smoothness of the model parameter time evolution. E.g., for the \( A \) matrix this means that the quantities

  \[ (\lambda_{2,i,j}^{a})^{2} (\hat{\alpha}_{t-1,t-l-1}(i,j) - 2\hat{\alpha}_{t,t-l}(i,j) + \hat{\alpha}_{t+1,t-l+1}(i,j))^{2} \]

  for \( i, j = 1, \ldots, n_y, \quad t = 2, \ldots, N - 1 \) and \( l = 1, \ldots, n_A(i,j,t) - 1 \), are penalized.

- **L3**\( \theta \)** Approximates the 1st order time derivative of the impulse response of the local models \( \theta \). It thus enforces dampened impulse responses onto the local model parameter estimates by penalizing the time evolution of the impulse responses. E.g., for the \( B \) matrix this means that the quantities

  \[ (\lambda_{3,i,j}^{a})^{2} (\hat{\beta}_{t+1,t-l-1}(i,j) - \hat{\beta}_{t+1,t-l}(i,j))^{2} \]

  for \( i = 1, \ldots, n_y, \quad j = 1, \ldots, n_u, \quad t = 1, \ldots, N - 1 \) and \( l \in \{ l \mid n_B(i,j,t + l) > 0, \quad l \in \{1, \ldots, N - t\} \} \), are penalized.

- **L4**\( \theta \)** Approximates the 2nd order time derivative of the impulse response of the local models \( \theta \). It thus enforces smooth impulse responses onto the local model parameter estimates by penalizing non-smoothness of the impulse responses. E.g., for the \( B \) matrix this means that the quantities

  \[ (\lambda_{4,i,j}^{a})^{2} (\hat{\beta}_{t+1,t-l-1}(i,j) - 2\hat{\beta}_{t+1,t-l-1}(i,j) + \hat{\beta}_{t+1,t-l-1}(i,j))^{2} \]
for $i = 1, \ldots, n_y$, $j = 1, \ldots, n_u$, $t = 1, \ldots, N - 2$ and $l \in \{ l \mid n_B(i, j, t + l) > l, l \in [1, \ldots, N - t] \}$, are penalized.

$L_5 \theta$ penalizes the variance of the parameter estimates $\widehat{\theta}$, and thus enforces minimum variance estimates — that is $L_5 = I$. E.g., for the $C$ matrix this means that the quantities

$\left(\lambda_{5, i, j}^2\right)^2 \widehat{c}_{t+1-t(i, j)}^2$

for $i, j = 1, \ldots, n_y$, $t = 2, \ldots, N$ and $l = 1, \ldots, n_C(i, j, t)$, are penalized.

The estimated parameter vector $\hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda)$ is a function of the weighting matrix $\Lambda$ (and hence the $5N_c$ scalar weights $\lambda_{i, j}$) that determines the coefficient shrinkage and hence the trade-off between bias and variance. This means that the regularization matrix $\Lambda$ can be used to tune the predictive capabilities of the model estimate for the specific modeling purpose. Through the particular choice of penalty matrix $L$, the regularization matrix $\Lambda$ also determines the interdependency between the grid-point models in the model grid. Note, if $L^T \Lambda L$ is designed to be positive definite then a unique solution to the estimation problem (2.41) is guaranteed. In general, if the Hessian matrix $\mathcal{H}$

$\mathcal{H} = X'WX + L^T \Lambda L$ (2.48)

is positive definite, then the estimation problem (2.41) has a unique solution. Note, when used alone, the penalty matrix $L_5$ reduces the TR problem (2.41) to a standard form TR problem or a RR problem. Furthermore, with the proper choice of weighting matrix $\Lambda_5^2$ (which for unbounded uncertainties will have to be negative definite (Golub et al. 1999; Kailath et al. 2000)), this penalty formulation also gives the solutions to Total Least Squares (TLS) problems (Himmelblau and Vandewalle 1991; Hansen 1996; Fierro et al. 1997; Golub et al. 1999; Kailath et al. 2000), i.e., to estimation problems with uncertain inputs. In fact, it is for this reason that LS is applied instead of TLS in the modeling methodology presented here, although the inputs ($X$) are indeed uncertain.

**Example 2.1**

This example will illustrate the differences between Least Squares, Ridge Regression, and Tikhonov Regularization. The two first are special cases of the latter, for particular choices of the penalty matrix $L$.

For simplicity, the system in this example is a Single Input Single Output (SISO), first order, Linear Time-Varying (LTV) system

$y_t = -a_{t-1}y_{t-1} + b_{t-1}u_{t-1}$

with the parameters

$a_{t, t-1} = 8(t - 0.42)^2 - 1$

$b_{t, t-1} = 2t - 1$
Figure 2.3: Batch Input-Output data used in example 2.1. The top figure shows the output $y_t$, and the bottom figure shows the input $u_t$. Notice the lack of excitation and hence insufficient information in the data from the end part of the batches.

for $t = 0, 0.01, 0.02, \ldots, 1$, where $y$ is the output and $u$ is the input. The batches available for identification, $N_B = 15$, are shown in figure 2.3. Note the lack of excitation, and hence lack of information in the output towards the end of the batches. Actually, the information is there, but it is concealed by the measurement noise drawn from normal distribution $N(0, 0.001)$, that is added to the outputs.

The system is modelled with a correct model structure, i.e., $n_A = n_B = 1$ and $n_C = 0$, and the model parameters are estimated from data from $N_{est}^B = 10$ batches and “Pure-Simulation” validated on data from the remaining $N_{val}^B = 5$ batches. The system is identified with Least Squares ($L = 0$), Ridge Regression ($L = L_5 = I$), and Tikhonov Regularization ($L = L_2$), and the respective parameter estimates are shown in figure 2.4. The regularizations with respect to the parameters $a_{t, t-1}$ and $b_{t, t-1}$ are weighted separately with the scalar weights $\lambda_a$ and $\lambda_b$, respectively. The weights $\lambda_a$ and $\lambda_b$ are chosen for minimum “Pure-Simulation” validation prediction errors. The optimal weights $\lambda_a$ and $\lambda_b$, and the normed sum-of-squared-prediction-errors (FIT) for the three regression methods are shown in table 2.1.

From figure 2.4 on page 30, it is obvious that Tikhonov Regularization is a
tractable regression method for this system, whereas Least Squares and Ridge Regression are not. It is however also obvious that all three regression methods fail when there is insufficient information in the data – as there indeed is for this system towards the end of the batches.

Implementation Issues

By reformulating the TR objective function $J_{w,LS}^{TR}$ in (2.41) into

$$
J_{w,LS}^{TR} = \|Y - X\theta\|^2_W + \|L\theta\|^2_\Lambda
= \|\begin{bmatrix} WY \\ 0 \end{bmatrix} - \begin{bmatrix} WX \\ \Lambda L \end{bmatrix}\theta\|^2
= \|Y - X\theta\|^2
$$

where $W = W'W$, it is obvious that if the Hessian matrix $H$ is positive definite, then the optimal model parameter estimate $\hat{\theta}_{TR}^{(2.41)}$ is the unique solution to the linear normal equations

$$
X' \left( Y - X\hat{\theta}_{TR}(V, W, \Lambda) \right) = 0
$$

From a geometrical point of view, the interpretation of the normal equations is that LS estimates are obtained when the (OSA) prediction errors $(Y - X\hat{\theta}_{TR}(V, W, \Lambda))$ are orthogonal to the linear subspace spanned by the regressors $(X)$. Note that, the normal equations are simply the First Order Necessary Optimality Condition, i.e., that the 1st order derivative of the cost function is zero, and that for convex optimization problems the First Order Necessary Optimality Condition is both necessary and sufficient. With a little algebra, the normal equations (2.50) can be reformulated as

$$
Q'Y = R\hat{\theta}_{TR}(V, W, \Lambda)
$$

Table 2.1: This table shows the optimal regularization weights and the corresponding normed sum-of-squared-prediction-errors (FIT) for the three regression methods applied in example 2.1. As should be expected, Tikhonov Regularization outperforms both Least Squares and Ridge Regression for this particular system.

<table>
<thead>
<tr>
<th>Regression Method</th>
<th>FIT</th>
<th>$\lambda^a$</th>
<th>$\lambda^b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Least Squares</td>
<td>1.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ridge Regression</td>
<td>0.97</td>
<td>6.70E-10</td>
<td>0.0100</td>
</tr>
<tr>
<td>Tikhonov Regularization</td>
<td>0.77</td>
<td>11.1</td>
<td>126</td>
</tr>
</tbody>
</table>
Figure 2.4: Model parameter estimates obtained using Least Squares (marker +), Ridge Regression (marker o), and Tikhonov Regularization (marker ♦), respectively, as well as the true model parameters (bold lines) of the system in example 2.1. As should be expected, Tikhonov Regularization outperforms both Least Squares and Ridge Regression, when estimating model parameters in rank-deficient systems with smooth model parameter time (or space) evolutions.

where $Q_1$ and $R$ are the truncated factors of the orthogonal QR factorization of the augmented input space $\bar{X}$

$$\bar{X} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R$$  \hspace{1cm} (2.52)

The truncated QR factor $Q_1$ is orthogonal, i.e., $Q_1'Q_1 = I$, and the truncated QR factor $R$ is square and upper-triangular. This means that the optimal
model parameter estimate $\hat{\theta}_{TR}$ can be computed as

$$
\hat{\theta}_{TR}(\hat{V}, W, \Lambda) = R^{-1}Q'_1Y \\
= R^{-1} \begin{bmatrix} Q'_{11} & Q'_{21} \end{bmatrix} \begin{bmatrix} WY \\ 0 \end{bmatrix}
$$

(2.53)

where the inverse $R^{-1}$ is efficiently computed by backwards-substitution as $R$ is upper-triangular. This furthermore means that, if desired, the Hessian matrix $\mathcal{H}$ and its inverse can be computed efficiently as

$$
\mathcal{H}^{-1} = (R'R)^{-1} = R^{-1}R'^{-1}
$$

(2.54)

In fact, for sparse systems, i.e., when $X$ is structured, which is the case for the methodology presented here, the column vector $Q'_1Y$ is computed directly without computing the often dense factor $Q_1$, i.e., with “Q-less” QR factorization. This means that (2.53) constitutes a numerically very efficient and stable way of computing the optimal model parameter estimates (see Golub and van Loan, 1996; Matstoms, 1997, for further details on (“Q-less”) QR factorization and its applications to linear LS).

The general Multiple Input Multiple Output (MIMO) system (2.32) is easily de-coupled into $n_y$ dependent Multiple Input Single Output (MISO) systems. That is, the individual output profiles are expressed solely by the regressors (/inputs) and can hence be de-coupled. Note that, the dependency between the MISO systems stem from the estimated OSA prediction error profiles that are part of the regressors — see the paragraph below. The same principle applies to the model parameter estimates (2.53), but the estimation problem (2.41) only de-couples into $n_y$ dependent sub-estimation problems if the weighting matrices $W_t$, for $t = 1, \ldots, N$, are all diagonal. This means that choosing diagonal weighting matrices $W_t$, for $t = 1, \ldots, N$, is a particularly tractable special case as the computational burden is reduced by a factor $n_y$. This computational simplification is illustrated in figure 2.5, which shows the structure of the upper-triangular matrix $R^{-1}$ in terms of non-zero elements, for the general case and the special case of all diagonal weighting matrices, respectively.

The solution $\hat{\theta}_{TR}(V, W, A)$ to the estimation problem (2.41) given by (2.53) is obviously dependent on the estimated OSA prediction error profiles $\hat{V}$, which until now have been assumed known. These OSA prediction error profiles are however not measurable and hence not known. This means that the OSA prediction error profiles have to be estimated from data simultaneously with the model parameters, and consequently that the estimation problem is a pseudo linear estimation problem. The combined model parameter and OSA prediction error profile estimation problem can be formulated as a general nonlinear optimization problem. This approach is however considered intractable in the
Figure 2.5: The structure, in terms of non-zero elements, of the upper-triangular matrix $\hat{R}^{-1} = PR^{-1}$, where $P$ is a permutation matrix, for five coupled (a) or de-coupled (b) Multiple Input Single Output (MISO) systems. The coupled MISO systems constitute one truly Multiple Input Multiple Output (MIMO) system.

modeling methodology presented here as it would present a considerable computational burden. Instead, it is attempted to utilize the numerically efficient explicit solution to the linear model parameter estimation problem \[2.41\] and to solve the pseudo linear estimation problem by solving a sequence of linear LS problems. That is, it is proposed to estimate the model parameters in a Pseudo Linear Regression (PLR) model with the iterative algorithm \[2.1\].

**Algorithm 2.1 (Estimation of PLR models)**

1. Set the iteration index $n$ to zero, and initialize the estimated prediction error with a zero vector.

   $$\hat{V}^n = 0$$

2. Construct a regressor matrix $X^n$ with the estimated prediction error profiles $\hat{V}^n$.

   If estimating ARMA(X) models, then the regressor matrix $X^n$ can be constructed with the model orders $\tilde{n}_A(i,j,t)$, $\tilde{n}_B(i,j,t)$ and $\tilde{n}_C(i,j,t)$, where

   $$n_A(i,j,t) \leq \tilde{n}_A(i,j,t) < \infty$$
   $$\tilde{n}_B(i,j,t) = n_B(i,j,t)$$
   $$\tilde{n}_C(i,j,t) = 0$$

   I.e., estimating truncated AR(X) approximations of the ARMA(X) models.
3. Compute the optimal solution $\hat{\theta}_{TR}^n(\hat{V}^n, W, \Lambda)$ to the linear regression model parameter estimation problem (2.41), as (2.53).

4. Update the estimated prediction error profiles

$$\hat{V}^{n+1} = Y - X^n\hat{\theta}_{TR}^n(\hat{V}^n, W, \Lambda)$$

and propagate the iteration index $n = n + 1$.

5. Construct a regressor matrix $X^n$ with the estimated prediction error profiles $\hat{V}^n$.

6. Compute the optimal solution $\hat{\theta}_{TR}^n(\hat{V}^n, W, \Lambda)$ to the linear regression model parameter estimation problem (2.41), as (2.53).

7. Update the estimated prediction error profiles

$$\hat{V}^{n+1} = Y - X^n\hat{\theta}_{TR}^n(\hat{V}^n, W, \Lambda)$$

8. Check for convergence in the estimated prediction error profiles

$$\|\hat{V}^{n+1} - \hat{V}^n\| \leq \varsigma$$

where $\varsigma$ is a user defined scalar tolerance. If convergence is not satisfied and the iteration index has not exceeded a user defined maximum number of iterations, then propagate the iteration index $n = n + 1$ and proceed to step #5. Otherwise, stop.

Algorithm 2.1 belongs to a class of PLR algorithms presented by [Sliidi] (1983) amongst others, that gives fast, but less accurate, approximations of Prediction Error Method (PEM) (see e.g. [Ljung] 1999) estimates. Note however, that algorithm 2.1 does not require monitoring as the model parameter estimates are already shrunk in the TR model parameter estimation problem. [Sliidi] (1983) considers ARMA(X) model estimation explicitly, and sets the autoregressive model orders in the initial AR(X) approximation to $\tilde{n}_A(i, j, t) = n_A(i, j, t) + n_C(i, j, t)$, which also seems to work well in practice. This means that [Sliidi] (1983) can and do, check for convergence in the model parameters instead of in the prediction error profiles as in algorithm 2.1. Checking for convergence in the prediction error profiles will however save one execution of step #6 in algorithm 2.1, which is where the main computational burden is located. [Stoica et al] (1984, 1985) discuss convergence and asymptotic accuracy of algorithms similar to algorithm 2.1 for LTI systems, but these questions remain unresolved for LTV systems. Another class of fast PLR algorithms is the Two-Step algorithms and modifications hereof (see e.g. [Mayne and Firoozan] 1982). Also, see e.g. [He] (2003) for a Bayesian approach to ARMAX estimation.
2.2.3 Model Assessment and Selection

Models are in general assessed in terms of their quality in order to provide a basis for model selection and to determine their validity. The quality of a model is most often evaluated by how well the model generalizes to an independent data set. That is, how well the model predicts the regressors/outputs given the predictors/inputs, in a data set unseen by the model.

The simplest and probably most widely used method for quantitative assessment of the quality of a model is cross-validation, which estimates the generalization error $G$ when a model $\hat{\theta}$ ($\theta$ will be used as short notation for $\theta_{\text{TR}}(\mathbf{V},W,\Lambda)$) is applied to an independent data set

$$G(\hat{\theta}) = \frac{1}{N_{\text{val}}^B} \sum_{k=1}^{N_{\text{val}}^B} \| \Delta\hat{y}_k - \Delta\hat{y}_k(\hat{\theta}) \|^2_W$$

(2.55)

where $\Delta\hat{y}_k$ is the model prediction of the regressors in batch $k$ of the independent validation data set and $W$ is a block diagonal weighting matrix with symmetrical, positive definite, block elements $W_t \in \mathbb{R}^{n_y(t),n_y(t)}$. Note, $G$ is the expectation of the true generalization error in a batch. Traditionally, the weighting matrix $W$ is set to $W = W^\top$, but this need not be the case. Indeed, if the same $n_y = n_y(t), t = 1, \ldots, N$, outputs are used throughout the batch, then for ease of comparison to alternative modeling methodologies, the weighting matrix $W$ should be chosen as

$$W_t(i,i) = ((N-1)n_y \max\{ |\Delta\hat{y}_{k,j}(i)| \mid k = 1, \ldots, N_B, j = 1, \ldots, N \})^{-1}$$

(2.56)

for $i = 1, \ldots, n_y$ and $t = 1, \ldots, N$, in which case the estimated generalization error $G$ would replicate the quality measure (goodness-of-fit) as defined by Ljung [1999] — i.e., a measure of the estimated variance of normalized deviation variables, not described by a model estimate. The size of the validation data set $N_{\text{val}}^B$ is typically 50% of the size of the data set used for model parameter estimation $N_{\text{est}}^B$. When data is too scarce to form two independent data sets ($N_{\text{est}}^B$ and $N_{\text{val}}^B$), $K$-fold cross-validation is used to estimate the generalization error. In $K$-fold cross-validation the combined data set with $N_{\text{est+val}}^B = N_{\text{est}}^B + N_{\text{val}}^B$ batches, is split up into $N_K$ equal-sized parts and for each part a model is fitted to the remaining $N_K - 1$ parts of the data set. These $N_K$ models are then cross-validated on their respective and unseen parts of the data set. $K$-fold cross-validation is typically not applicable to time-series analysis, but given their finite time dimension, $K$-fold cross-validation is easily applied to batch processes. E.g., let $\kappa(k)$ index the model $\hat{\theta}_{\kappa(k)}$ that has not been fitted to data from batch $k$, then the generalization error of a model $\hat{\theta}$ fitted to the entire data set, is estimated as

$$G_K(\hat{\theta}) = \frac{1}{N_{\text{est+val}}^B} \sum_{k=1}^{N_{\text{est+val}}^B} \| \Delta\hat{y}_k - \Delta\hat{y}_k(\hat{\theta}_{\kappa(k)}) \|^2_W$$

(2.57)
The number of parts $N_K$ determines the trade-off between bias and variance in the estimate of the generalization error — typically five- or tenfold cross-validation is recommended (Hastie et al., 2001).

Generalized cross-validation provides an approximation of the leave-one-batch-out cross-validation and gives the estimated generalization error as

$$
\tilde{G}(\hat{\theta}) = \frac{1}{N_{est+val}^B} \sum_{k=1}^{N_{est+val}^B} \left[ \frac{\| \Delta \hat{y}_k - \Delta \hat{y}_k(\hat{\theta}) \|^2_W}{(1 - \text{trace}(S)/N_{est+val}^B)^2} \right]
$$

(2.58)

where $S$ is the projection of $Y$ onto the linear subspace spanned by the regressors, i.e.,

$$
Y(\hat{\theta}) = \begin{bmatrix} \Delta \hat{y}_1(\hat{\theta})' & \Delta \hat{y}_2(\hat{\theta})' & \ldots & \Delta \hat{y}_{N_{est+val}^B}(\hat{\theta})' \end{bmatrix}'
= XR^{-1} R^{-1}' X' W Y
= SY
$$

(2.59)

where it is assumed that $N_{est+val}^B = N_{est}^B$. For batch processes, the accuracy of generalized cross-validation can be improved by replacing the trace of $S$ with the trace of the part of $S$ that corresponds to the individual batches

$$
\tilde{G}(\hat{\theta}) = \frac{1}{N_{est+val}^B} \sum_{k=1}^{N_{est}^B} \left[ \frac{\| \Delta \hat{y}_k - \Delta \hat{y}_k(\hat{\theta}) \|^2_W}{(1 - \text{trace}(S_k))^2} \right]
$$

(2.60)

where $S_k$ is the projection of $\Delta \hat{y}_k$ onto the linear subspace spanned by the regressors, i.e.,

$$
\Delta \hat{y}_k(\hat{\theta}) = \Delta \hat{x}_k R^{-1} R^{-1}' \Delta \hat{x}_k' W \Delta \hat{y}_k
= S_k \Delta \hat{y}_k
$$

(2.61)

The efficient computation makes generalized cross-validation tractable when a large number of models are to be assessed. The trace of the projection matrix $S$ (also referred to as the smoother and hat matrix) is the number of effective model parameters, i.e., the number of significant model parameters given the level of regularization. That, along with the approximation $1/(1-x)^2 \approx 1 + 2x$, shows the similarity between generalized cross-validation and the widely used Akaike Information Criteria (AIC) (Hastie et al., 2001). When generalized cross-validation is used for selection of regularization weights it can alleviate the tendency of cross-validation to under-smooth/shrink (Hastie et al., 2001).

When it comes to assessment of ARMA(X) models however, the drawback of cross-validation is that the accuracy of the model parameters in the MA part of a model is not properly assessed as the stochastic signals driving the MA part are unknown for the independent data set. This means that the estimated
generalization error $G$ on its own, is intractable for selection of ARMA(X) models. To overcome this intractability in the assessment of ARMA(X) models, the estimated generalization error $G$ is supplemented (as discussed below) with the fitting error $F$ from the model parameter estimation

$$F(\hat{\theta}) = \frac{1}{N_{\text{est}}} \sum_{k=1}^{N_{\text{est}}} \| \Delta \hat{y}_k - \Delta \hat{y}_k(\hat{\theta}) \|_W^2$$  \hspace{1cm} (2.62)

i.e., the mean batch prediction error when the model $\hat{\theta}$ is fitted to the estimation data set.

Thus far the nature of the model predictions $\Delta \hat{y}$ has not been discussed. The most general model prediction for model assessment is the $\tau$-step-ahead prediction $\Delta \hat{y}_{k,t+\tau}$ given the input profiles of batch $k$ and the outputs up to and including time $t$ in batch $k$. The number of steps ahead $\tau$ should be chosen such that a model suitable for the intended application is selected based on the model assessment. Most often OSA prediction (i.e. $\tau = 1$) is the method of choice. Two factors provide strong motivation for this choice of prediction horizon. First of all, if a model $\hat{\theta}$ can not predict one step ahead, it is not very likely to perform better over longer horizons. Secondly, it is a very tractable implementation

$$\Delta \hat{y}_k^{\text{osa}}(\hat{\theta}) = \Delta \hat{x}_k \hat{\theta}$$  \hspace{1cm} (2.63)

which is of significant importance when a considerable number of models are to be assessed. Another computationally tractable choice of prediction horizon is Pure-Simulation (PS), which means that the output profiles are predicted given the initial conditions, the input profiles, and for the estimation data set, the estimated OSA prediction error profiles (for the validation data set the OSA prediction error profile is unknown and hence $\hat{v}_k = 0$)

$$\Delta \hat{y}_k^{\text{ps}}(\hat{\theta}) = \hat{H}(\hat{\theta}) \Delta \hat{y}_{k,0} - \hat{G}(\hat{\theta}) \Delta \hat{u}_k + \hat{F}(\hat{\theta}) \hat{v}_k$$  \hspace{1cm} (2.64)

That is, PS prediction represents the most demanding application of a batch process model. Thus, if a models quality is assessed to be good in terms of the estimated generalization error based on PS model predictions $G^{\text{ps}}$, the model will have significant credibility in any application. In general, OSA model prediction error based model assessment and selection has a tendency to under-smooth, while PS model prediction error based model assessment and selection has a tendency to over-smooth. However, as OSA and PS are the limit points for $\tau$-step-ahead prediction, combining these two provides a computationally tractable alternative to traditional $\tau$-step-ahead prediction error based model assessment and selection. This alternative to $\tau$-step-ahead prediction error based model assessment uses the following weighted fitting and generalization errors

$$F_\rho(\hat{\theta}) = (1 - \rho)F^{\text{ps}}(\hat{\theta}) + \rho F^{\text{osa}}(\hat{\theta})$$

$$G_\rho(\hat{\theta}) = (1 - \rho)G^{\text{ps}}(\hat{\theta}) + \rho G^{\text{osa}}(\hat{\theta})$$  \hspace{1cm} (2.65)
where superscripts indicate the type of model predictions (i.e., $\Delta \hat{y}_{\text{ps}}$ or $\Delta \hat{y}_{\text{osa}}$) and $\rho$ is a scalar between 0 and 1. The scalar $\rho$ should be chosen such that a model suitable for its application is selected based on the model assessment. In practice, the golden cut, i.e., $\rho = 0.382$, seems to work well for general purpose models.

**Optimal Model selection**

In order to facilitate a more automated model selection, the optimal model $\hat{\theta}^*$ is mathematically defined as the argument that solves the Multiple Objective (MO) optimization problem

$$
\begin{align*}
\min_{\theta_{\text{ARX}}} & \left[ G_{\rho}(\hat{\theta}) \right] \\
\min_{\theta_{\text{MA}}} & \left[ F_{\rho}(\hat{\theta}) \right] \\
\text{s.t.} & \quad \theta = P_{\text{ARX}}\hat{\theta}_{\text{ARX}} + P_{\text{MA}}\hat{\theta}_{\text{MA}}
\end{align*}
$$

(2.66)

where $\hat{\theta}_{\text{ARX}}$ and $\hat{\theta}_{\text{MA}}$ are partitions of $\hat{\theta}$, that corresponds to the ARX and MA parts of the model, respectively. The matrices $P_{\text{ARX}}$ and $P_{\text{MA}}$, are appropriate permutation matrices. The MO optimization problem (2.66) is a non-convex Mixed Integer NonLinear Programming (MINLP) problem and solving it would almost certainly be infeasible — recall that a model is dependent on the discrete model orders. However, dividing the MO MINLP problem into subproblems and solving these subproblems in a hierarchical order, a near optimal model $\hat{\theta}^*$ can be found.

**Level 0** For a given set of model orders ($n_A, n_B, n_C$), weighting matrices $W$ and $\Lambda$, use Algorithm 2.1 to estimate a model $\hat{\theta}_{\text{TR}}(V, W, \Lambda)$ and the corresponding OSA prediction error profiles $V$.

**Level 1** For a given set of model orders ($n_A, n_B, n_C$) and the weighting matrix $W$, find the diagonal weighting matrix $\Lambda^*$ that minimizes (2.66). An optimal diagonal weighting matrix $\Lambda^*$ is found by dividing it into two column vectors $\lambda_{\text{ARX}} \in \mathbb{R}^{5(N_a^2+N_a)}$ and $\lambda_{\text{MA}} \in \mathbb{R}^{5N_c}$, which corresponds to the ARX and MA parts of the model, respectively, such that

$$
\Lambda = P_{\text{ARX}}\lambda_{\text{ARX}} + P_{\text{MA}}\lambda_{\text{MA}}
$$

(2.67)

where $P_{\text{ARX}}$ and $P_{\text{MA}}$ are permutation matrices, and solving the following MO optimization problem

$$
\begin{align*}
\min_{\lambda_{\text{ARX}}} & \left[ G_{\rho}(\hat{\theta}_{\text{TR}}(V, W, \Lambda)) \right] \\
\min_{\lambda_{\text{MA}}} & \left[ F_{\rho}(\hat{\theta}_{\text{TR}}(V, W, \Lambda)) \right] \\
\text{s.t.} & \quad \Lambda = P_{\text{ARX}}\lambda_{\text{ARX}} + P_{\text{MA}}\lambda_{\text{MA}}
\end{align*}
$$

(2.68)
The two optimization problems in the MO optimization problem (2.68), are not guaranteed convex, but are assumed convex for ease of implementation. A Pareto optimal solution to (2.68) is given as

$$\Lambda^* = P_{\text{ARX}} \lambda_{\text{ARX}}^* + P_{\text{MA}} \lambda_{\text{MA}}^*$$  \hspace{1cm} (2.69)

where

$$\lambda_{\text{ARX}}^* = \arg\min_{\lambda_{\text{ARX}}} \left[ G_\rho \left( \hat{\theta}_{\text{TR}}(\hat{V}, W, A) \right) \right] \hspace{1cm} s.t. \hspace{0.5cm} \Lambda = P_{\text{ARX}} \lambda_{\text{ARX}} + P_{\text{MA}} \lambda_{\text{MA}}$$ \hspace{1cm} (2.70)

and

$$\lambda_{\text{MA}}^* = \arg\min_{\lambda_{\text{MA}}} \left[ F_\rho \left( \hat{\theta}_{\text{TR}}(\hat{V}, W, A) \right) \right] \hspace{1cm} s.t. \hspace{0.5cm} \Lambda = P_{\text{ARX}} \lambda_{\text{ARX}} + P_{\text{MA}} \lambda_{\text{MA}} \hspace{1cm} (2.71)$$

where the scalar $\mu_\lambda \in [0; 1]$ is a user defined minimum improvement on the estimated generalization error, e.g. if $\mu_\lambda = 0.05$, then $G_\rho(\hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda^*))$ will improve by a minimum of 5% compared to $G_\rho(\hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda))$. Obviously, if $\mu_\lambda > 0$, a solution to (2.71) need not exist. For alternative approaches to determining optimal regularization parameters see e.g. Hansen (1996), Sugiyama and Ogawa (2002), Hagiwara (2002), Belge et al. (2002).

**Level 2** For a given weighting matrix $W$, find the set of model orders $(n_A^*, n_B^*, n_C^*)$ that minimize (2.66). An optimal set of model orders $(n_A^*, n_B^*, n_C^*)$ is obviously found by solving

$$\begin{align*}
\min_{(n_A, n_B)} & \left[ G_\rho \left( \hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda^*, n_A, n_B, n_C) \right) \right] \\
\min_{n_C} & \left[ F_\rho \left( \hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda^*, n_A, n_B, n_C) \right) \right] \\
\text{s.t.} & \Lambda^* \text{ given by (2.69)} \\
& 0 \leq n_i \leq \overline{n}_i, \text{ for } i = A, B, C
\end{align*}$$

(2.72)

where $\overline{n}_i$ are the maximum model orders allowed by the user. A Pareto
optimal solution to (2.72) is given as

\[
(n_A^*, n_B^*) = \arg \min_{(n_A, n_B)} \left[ G_{\rho} \left( \hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda^*, n_A, n_B, n_C) \right) \right]
\]

s.t. \( \Lambda^* \) given by (2.69)
\( n_C \) is known
\( 0 \leq n_i \leq \pi_i, \quad \text{for} \ i = A, B \) \hspace{1cm} (2.73)

\[
n_C^* = \arg \min_{n_C} \left[ J_{\rho} \left( \hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda^*, n_A^*, n_B^*, n_C) \right) \right]
\]

s.t. \( \Lambda^* \) given by (2.69)
\( 0 \leq n_C \leq \pi_C \)
\( \frac{G_{\rho} \left( \hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda^*, n_A^*, n_B^*, n_C) \right)}{G_{\rho} \left( \hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda^*, n_A^*, n_B^*, n_C) \right)} \leq 1 - \mu_n \) \hspace{1cm} (2.74)

where the scalar \( \mu_n \in [0; 1] \) is a user defined minimum improvement on the estimated generalization error. Obviously, if \( \mu_n > 0 \), a solution to (2.74) need not exist. The two integer programming problems (2.73) and (2.74) are non-convex and solving them by searching every possible combination of the elements in \( (n_A, n_B) \) and \( n_C \), respectively, will most certainly be infeasible due to the curse of dimensionality. However, by repeatedly manipulating the individual elements in the model order set \( (n_A, n_B, n_C) \) one by one, until convergence, a reasonable, but suboptimal solution \( (n_A^*, n_B^*, n_C^*) \) can be reached – see algorithm 2.2.

Algorithm 2.2

1. Guesstimate an initial set of model orders \( (n_A^0, n_B^0, n_C^0) \) and initialize the iteration index \( m = 0 \).
2. Given the model orders \( (n_A^0, n_B^0, n_C^0) \), find the optimal weighting matrix \( \Lambda^* \) as given by (2.69), and initialize the iteration index \( n = 0 \).
3. Compute the estimated generalization error \( G_{\rho}^0 \) given the model orders \( (n_A^0, n_B^0, n_C^0) \) and the optimal weighting matrix \( \Lambda^* \).
4. Propagate the iteration index \( n = n + 1 \) and specify local search ranges, i.e., \( (n_A^n, \pi_A^n) \), \( (n_B^n, \pi_B^n) \) and \( (n_C^n, \pi_C^n) \) such that
\[
0 \leq n_i^n \leq n_i^{n-1} \leq n_i^n \leq \pi_i, \quad \text{for} \ i = A, B, C
\]
5. Improve the ARX model orders

\[
\text{for} \ i = 1 \ \text{to} \ n_y \ \text{do}
\]
for $j = 1$ to $n_y$ do

$$n_A^n(i, j) = \arg \min_{n_A^n(i, j)} \left[ G_\rho \left( \hat{\theta}_{TR}(\hat{V}, W, A^*, n_A^n, n_B^n, n_C^n) \right) \right]$$
\hspace{1cm} s.t. $n_A^n(i, j) \leq n_A^n(i, j) \leq \bar{n}_A^n(i, j)$

end

for $j = 1$ to $n_u$ do

$$n_B^n(i, j) = \arg \min_{n_B^n(i, j)} \left[ G_\rho \left( \hat{\theta}_{TR}(\hat{V}, W, A^*, n_A^n, n_B^n, n_C^n) \right) \right]$$
\hspace{1cm} s.t. $n_B^n(i, j) \leq n_B^n(i, j) \leq \bar{n}_B^n(i, j)$

end

do compute the temporary estimated generalization error $\hat{G}_\rho$

$$\hat{G}_\rho = G_\rho \left( \hat{\theta}_{TR}(\hat{V}, W, A^*, n_A^n, n_B^n, n_C^n) \right)$$

6. Improve the MA model orders

for $i = 1$ to $n_y$ do

for $j = 1$ to $n_y$ do

$$n_C^n(i, j) = \arg \min_{n_C^n(i, j)} \left[ F_\rho \left( \hat{\theta}_{TR}(\hat{V}, W, A^*, n_A^n, n_B^n, n_C^n) \right) \right]$$
\hspace{1cm} s.t. $n_C^n(i, j) \leq n_C^n(i, j) \leq \bar{n}_C^n(i, j)$

$$\hat{G}_\rho = G_\rho \left( \hat{\theta}_{TR}(\hat{V}, W, A^*, n_A^n, n_B^n, n_C^n) \right)$$

$$\hat{G}_\rho \leq 1 - \mu_n$$

end

end

7. Check for convergence in the model orders — i.e., if

$$(n_A^n, n_B^n, n_C^n) \neq (n_A^{n-1}, n_B^{n-1}, n_C^{n-1})$$

and $n + 1$ does not exceed a user specified maximum number of iterations, then go to step #4. Otherwise, reinitialize the model orders

$$(n_A^0, n_B^0, n_C^0) = (n_A^n, n_B^n, n_C^n)$$

and proceed.
8. Check for convergence in the estimated generalization error

\[ \| G^0_\rho - G_\rho \left( \hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda^*, n^A_A, n^B_B, n^C_C) \right) \| \leq \tilde{\varsigma} \]

where \( \tilde{\varsigma} \) is a user specified convergence tolerance. If convergence is not satisfied and the propagated iteration index \( m = m + 1 \) does not exceed a user specified maximum number of iterations, then go to step #2. Otherwise, set the suboptimal model orders to

\( (n^*_A, n^*_B, n^*_C) = (n^A_A, n^B_B, n^C_C) \)

and stop.

Most often the outer loop (steps 2 to 8) in algorithm 2.2 will converge in just a few iterations \((< 5)\) provided that the inner loop (steps 4 to 7) converges inside the allowed number of iterations.

A fourth and fifth level could be finding an optimal weighting matrix \( W^* \) and an optimal experimental design, respectively, but these fall outside the present thesis.

Model Testing

As the selected near optimal model

\[ \hat{\theta}^* = \hat{\theta}_{\text{TR}}(\hat{V}, W, \Lambda^*, n^*_A, n^*_B, n^*_C) \]  

(2.75)

is obviously dependent on both the estimation and the validation data sets, it must be tested on a third data set, namely the test data set, which should be approximately equal in size compared to the validation data set. For this purpose, the estimated test generalization error \( G^*_\rho \) is defined as

\[ G^*_\rho (\hat{\theta}^*) = (1 - \rho) G^{ps*} (\hat{\theta}^*) + \rho G^{osa*} (\hat{\theta}^*) \]

(2.76)

where

\[ G^{i*} (\hat{\theta}^*) = \frac{1}{N_{\text{test}}} \sum_{k=1}^{N_{\text{test}}} \left\| \Delta \hat{y}_k - \Delta \hat{y}_k^i (\hat{\theta}^*) \right\|_W^2, \quad i = \text{ps, osa} \]

(2.77)

This means that the estimated test generalization error \( G^*_\rho \) is the key deliverable when assessing to which extent a model can be trusted to fulfill its purpose and finally, whether or not the model can be implemented. Note, the estimated test generalization error \( G^*_\rho \) is only the key deliverable in terms of quantified model quality, and it should always be supplemented with visual inspection/validation of the model predictions and parameters.
2.3 Modeling Example

In collaboration with an industrial partner it has been investigated how multi-
variable model-based control could aid optimal utilization of constrained util-
ities in a sugar plant. One aspect of this problem is control of sugar crystal-
ization in vacuum pans. During operation the pans are under vacuum and
are continuously fed with concentrated/thick sugar juice as water evaporates
and sugar crystals grow. The energy for evaporation is supplied through steam
heated coils. To obtain the desired sugar quality, i.e., crystal size distribution
and purity, a desired sugar supersaturation profile is tracked throughout the
sugar crystallization operation cycle/batch duration. The crystal size distri-
bution is important not only for esthetic reasons, but equally important, to
maintain efficient operation of down-stream centrifuges. The esthetically de-
sired whiteness of the sugar crystals is determined by the purity, i.e., embedded
impurities e.g. between two agglomerated crystals will color the crystals. The
sugar crystal size distribution and purity will depend on the degree of supersat-
uration experienced during the batch. The degree of supersaturation is tracked
by manipulating one or more of four main control valves, i.e., manipulating the
sugar juice feed rate, stirrer work, heat supply, and/or vacuum pressure. The
setpoint profiles tracked by the decentralized PID controllers manipulating the
control valves are independent (non-)linear functions of either the pan level or
batch time. Depending on the current stage of the operation cycle/batch, the
operators manipulate usually just one of these setpoint functions to accom-
modate for disturbances in the sugar juice and/or utilities. As the operators
oversee several pans, which are all at different stages of operation, corrective
actions are both sparse, inconsistent and prone to errors due to operator defi-
ciencies. Thus, the sugar quality may suffer accordingly. Moreover, the control
ability of three of the four control valves is constrained by the utilities, i.e.,
the limited availability of sugar juice, sufficient steam pressure, and sufficient
vacuum pressure. Whether or not a valve is at or close to its constraint will de-
pend on the current stage of operation of the entire sugar plant. Consequently,
an automated multivariable model-based control scheme, which can handle the
control constraints in an optimal fashion, is desired if the sugar quality is to
be improved. Furthermore, given frequent and consistent corrective actions,
the operation of the sugar pans will be more smooth and thus more energy
efficient.

For this reason, the applicability of the suggested modeling methodology
has been investigated in a preliminary study. The modeling objective is to pre-
dict the conductivity as a measure of the pan concentration, and the pan weight
as a measure of the pan level, from the initial charging of a vacuum pan and
the subsequent 112 minutes (i.e., $N = 112$), given the realized setpoint profiles
of the five controllers (two of the controllers interchangeably manipulate the
sugar juice feed rate given the operation stage — switching from pan weight
control to conductivity control between 10 – 40 minutes from the initial charge
of a pan). Note that, predicting the pan weight is equivalent to predicting the
cycle time. This means that, an upper-level controller based on these 2 
outputs and 5 inputs (setpoint profiles for conductivity, stirrer work, level, vapor 
pressure, steam pressure, in that order), pan model could be added as a multi-
variable extension or add-on to the existing decentralized PID control system. 
To comply with the modeling objective, data from 51 batches was filtered and 
re-sampled in 113 sample points one minute apart, starting from the end of the 
initial pan changing operation stage. The 51 batches were chosen at random 
and included batches from production periods where the individual pan control 
systems were saturated due to insufficient utility capacity. The vacuum pan 
model parameters were fitted to data from 26 batches and the regularization 
weights and model orders (the maximum allowed model order was 4) were iden-
tified through cross-validation on 15 independent batches. The TR estimation 
problem was weighted with the weighting matrix $W$ given by (2.36). The (near) 
optimal model orders and regularization weights are given in tables 2.2 and 2.3 
respectively. It is interesting to note that there is not a direct correlation (the 
model orders are zero) between neither the level setpoint profile nor the vapor 
pressure setpoint profile and the level profile model prediction. With the model 
orders given in table 2.2, the model set of 113 local models has a total of 2,229 
model parameters, but with the regularization weights given in table 2.3 the 
model set has just 133 effective model parameters. This means that the regu-
larization has shrunk the space spanned by the model parameter by 94%. The 
optimized sugar pan model was tested on 10 independent batches and achieved 
an estimated test generalization error of $G^\star_{0,382} = 0.0540$, with the weighting 
matrix $\bar{W}$ given as (2.56). For completeness, the OSA prediction based esti-
mated test generalization error is $G^\star_1 = 0.0394$ and the PS prediction based 
estimated test generalization error is $G^\star_0 = 0.0855$. The model PS prediction 
of one of these test batches, is shown in Figure 2.6. The discrepancies between 
the PS model predictions and the realized measurements in the interval 10 – 40 
minutes from the initial charging of the pan, are believed to be caused by the 
fact that the switching time between weight control and conductivity control 
although known, has been ignored in the modeling work and that the model 
hence during this interval, considers input signals that have had no influence 

Table 2.2: The 18 model orders of the sugar pan model. The near optimal model 
orders were found by simultaneously minimizing the estimated generalization 
error and the fitting error. The maximum allowed model order was 4. 

<table>
<thead>
<tr>
<th>Model Orders</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
<th>$j = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_A(1, j, t), \ t = 1, \ldots, N$</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\alpha_A(2, j, t), \ t = 1, \ldots, N$</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\alpha_B(1, j, t), \ t = 1, \ldots, N$</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha_B(2, j, t), \ t = 1, \ldots, N$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha_C(1, j, t), \ t = 1, \ldots, N$</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\alpha_C(2, j, t), \ t = 1, \ldots, N$</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
The quality of the obtained sugar pan model is judged to indicate significant potential for industrial implementation of the proposed modeling methodology.

2.4 Modeling Conclusions

The present chapter presented a methodology for data driven time series modeling of batch processes. The modeling methodology comprises the development of a batch Auto-Regressive Moving Average model with eXogenous inputs (ARMAX) and its transformation into State Space (SS) representations designed for specific applications in control and optimization of batch processes. The presented modeling methodology produces both a Linear Time-Invariant (LTI) SS model capable of inter-batch prediction as well as a Linear Time-Varying (LTV) SS model capable of intra-batch prediction. With such two SS models, the design and implementation of model-based tools such as Iterative Learning
Control (ILC), Model Predictive Control (MPC) and the Kalman filter, constitute an imminent and indeed relatively effortless, next step on the path towards optimal and reproducible operation of batch processes. The batch ARMAX model consists of a set of local, but interdependent ARMAX models. Each of these local ARMAX models approximates the behavior of the batch process locally in both batch time and state space and as these local models are distributed over batch time, they are hence only valid if the set of phenomena that comprise the batch behavior locally in batch time and state space is approximately given exclusively by batch time. This constitutes the main assumption of the presented modeling methodology, namely that the batch behavior can be approximately characterized by batch time. This main assumption is also the main limitation of the applicability of the modeling methodology, as the assumption can only be assumed valid locally in state space. However, if the intended use of the batch ARMAX model is the design of stabilizing, reference tracking controllers, the reference being local in state space, then the assumption that the batch behavior can be approximately characterized by batch time, may indeed be valid.

The modeling methodology furthermore comprises the steps from data preparation to model assessment and selection, necessary for identification of batch ARMAX models. The first step is data preparation, where the modeling methodology proposes that a Kernel smoother with local polynomial regression, is applied to filter and re-sample, as well as reconstruct missing sequences of, historical data. Note however, that unless the reconstructed variable is strongly correlated to other known variables and these correlations are also known, that it is not possible to validate reconstructed data sequences and data reconstruction should hence only be considered as a last resort to obtaining sufficient data. The modeling methodology then proposes a computationally efficient algorithm for estimation of a set of dependent Pseudo Linear Regression (PLR) models. At the core of this PLR algorithm is the explicit solution to a Linear Regression (LR) estimation problem. This LR estimation problem is formulated as a Tikhonov Regularization (TR) problem. The proposed TR problem formulation enforces desirable model properties such as local model interdependency and minimum variance into the model parameter estimates. The explicit solution to the TR problem is efficiently computed with “Q-less” QR factorization. The proposed PLR algorithm belongs to a class of PLR algorithms that gives fast, but less accurate, approximations of Prediction Error Method (PEM) estimates, but unlike most of these, this PLR algorithm does not require monitoring to avoid divergence, as the parameter estimates are already shrunk in the TR problem formulation. The modeling methodology finally proposes three quantitative measures of model quality for model assessment and selection, and defines the optimal model in terms of these quality measures. Furthermore, a model estimate the quality of which, will most likely be inferior but close to that of the optimal model, can be found with a hierarchical optimization algorithm proposed as part of the modeling methodology. This optimization algorithm consists of three levels of increasing mathematical
programming complexity, but with the main computational burden located at the mathematically simpler levels. The main assumption of this optimization algorithm is that the optimization problem on the second level (level 1) has only one (global) extremum, when the solution to the lowest optimization level (level 0 — the PLR algorithm) exists and is unique. The optimization problem on the second level is not subject to an explicit requirement of uniqueness of the lowest optimization level, but uniqueness is implicitly given locally by the TR formulation — i.e., by the penalty on the variance of the model parameter estimates, as a model with excessive parameter variance will be of poor quality. Uniqueness on the lowest optimization level is however not proven sufficient to guarantee only one (global) extremum for the second optimization level.
Table 2.3: The 90 regularization weights of the sugar pan model. The optimal regularization weights were found by simultaneously minimizing the estimated generalization error and the fitting error. Note, out of the possible 90 just 51 regularization weights are in use — unused regularization weights are marked by x.

<table>
<thead>
<tr>
<th>Weights</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
<th>$j = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_{a,1}^{1,j}$</td>
<td>1.88e-5</td>
<td>3.64</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{a,2}^{1,j}$</td>
<td>1.16e-6</td>
<td>1.13e-2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{a,1}^{2,j}$</td>
<td>1.00e-6</td>
<td>9.69e-7</td>
<td>1.50e-5</td>
<td>6.10e-6</td>
<td>2.30e1</td>
</tr>
<tr>
<td>$\lambda_{a,2}^{2,j}$</td>
<td>1.57e-6</td>
<td>1.29e-6</td>
<td>x</td>
<td>x</td>
<td>8.74e-3</td>
</tr>
<tr>
<td>$\lambda_{b,1}^{1,j}$</td>
<td>1.00e-6</td>
<td>1.00e-6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{b,2}^{1,j}$</td>
<td>1.00e-6</td>
<td>1.00e-6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{c,1}^{1,j}$</td>
<td>5.44</td>
<td>8.32e-1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{c,2}^{1,j}$</td>
<td>1.12e1</td>
<td>3.27e-1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{b,1}^{1,j}$</td>
<td>3.76e9</td>
<td>5.46e2</td>
<td>8.30e2</td>
<td>2.68e1</td>
<td>4.38e-2</td>
</tr>
<tr>
<td>$\lambda_{b,2}^{1,j}$</td>
<td>1.15e3</td>
<td>1.59e-5</td>
<td>x</td>
<td>x</td>
<td>4.06e-3</td>
</tr>
<tr>
<td>$\lambda_{b,1}^{2,j}$</td>
<td>1.00</td>
<td>1.00</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{b,2}^{2,j}$</td>
<td>1.00</td>
<td>1.00</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{b,1}^{3,j}$</td>
<td>x</td>
<td>1.00e-6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{b,2}^{3,j}$</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{c,1}^{1,j}$</td>
<td>9.00e-7</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\lambda_{c,2}^{1,j}$</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\lambda_{b,1}^{4,j}$</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{b,2}^{4,j}$</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{c,1}^{2,j}$</td>
<td>9.77e-7</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\lambda_{c,2}^{2,j}$</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\lambda_{a,1}^{3,j}$</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{a,2}^{3,j}$</td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{a,1}^{4,j}$</td>
<td>8.75e-2</td>
<td>4.12e-1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{a,2}^{4,j}$</td>
<td>8.82e-2</td>
<td>1.09e-2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{b,1}^{5,j}$</td>
<td>5.21e-1</td>
<td>4.41e-1</td>
<td>3.42e-2</td>
<td>4.52e-1</td>
<td>9.29e-1</td>
</tr>
<tr>
<td>$\lambda_{b,2}^{5,j}$</td>
<td>2.95e-1</td>
<td>8.57</td>
<td>x</td>
<td>x</td>
<td>2.78</td>
</tr>
<tr>
<td>$\lambda_{b,1}^{6,j}$</td>
<td>1.00</td>
<td>1.00</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\lambda_{b,2}^{6,j}$</td>
<td>1.00</td>
<td>1.00</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Control and Optimization Methodologies

Batch processes are typically an integrated part of continuously operated chemical production plants, and as such, their operation is both dependent on and influencing the operation the production plants. Chemical plants that utilize batch processing in continuous production in the sense of continuous and batch type processes combined serially and/or in parallel, are labeled hybrid plants. Operation of chemical plants is typically designed to comply with two possibly conflicting, desirable operation qualities, i.e., *optimal and reproducible operation*. Optimal operation usually involve optimizing some business objective (e.g. maximum production rate and/or minimum production costs) subject to constraints (e.g. environmental emission and/or capacity constraints), while reproducible batch operation usually means minimum variability in e.g. production rates and/or product qualities. If optimal operation entails operation in close proximity to bifurcation points, optimal operation may induce high sensitivity and is hence potentially in conflict with the desire for reproducible operation. When optimal operation induces variability, optimal *and* reproducible operation can only be obtained simultaneously through the application of advanced process control.

When in pursuit of optimal and reproducible operation of chemical hybrid plants, one possible step may be to pursue reproducible operation of the individual batch processes. By reproducible operation of a batch process, is meant the ability to produce products with similar specifications and/or production costs in several consecutive batch runs. Achieving reproducible operation of the individual batch processes, will significantly simplify forecasting the demand on upstream processes, the utility consumption, and the load on downstream processes, which in turn will significantly simplify production planning and batch process scheduling, which may ultimately lead to optimal and reproducible operation of the production plant. The reproducibility of a batch process can be expressed in terms the deviations between a desired (optimal) operation trajectory and the trajectories realized during operation of the batch process. These deviations between a desired or reference operation trajectory and the trajectories realized during operation are labeled trajectory tracking errors. Optimal reproducibility of a batch process is thus achieved by minimizing the trajec-
tory tracking errors. This leads to the formulation of optimal reproducibility of batch processes as a stochastic Optimal tracking Control Problem (OCP).

Another step in the pursuit of optimal and reproducible operation of chemical hybrid plants is to optimize the operations models/recipes according to which the batch processes are operated. The operation of a batch determines the consumption of raw materials and utilities such as heating and cooling as well as the production of desired and undesired products. These batch operations models are designed to comply with and should also best utilize, constraints on utilities and up- and down-stream capacity bottlenecks. Thus, by assigning costs to raw materials, utilities, products and down-stream recovery, an optimal batch operations model/recipe can be formulated and pursued as a constrained OCP.

In state-of-the-art batch control systems chemical batch processes are typically operated according to operations models implemented at Programmable Logic Controllers (PLC) level. This state-of-the-art batch control structure leaves batch process operation exposed to undesired variability induced by process disturbances, in product quality and quantity measures, affecting the profitability of chemical batch production. The present chapter will however present OCP formulations that will replace the open-loop implementation of the operations model thus improving the profitability of chemical batch production. This optimal control extension of the state-of-the-art batch control system is shown in figure 3.1.

The present chapter first presents an iterative control algorithm for guaranteed reproducible repetitive batch process operation in section 3.1. This iterative control algorithm which comprises an optimal tracking control problem formulation and a control design criterion, is proven to guarantee closed-loop stability. The chapter then proceeds in section 3.2 with a dual dimension formulation of the optimal tracking control problem of section 3.1. Combined with the control design criterion of section 3.1 this dual dimension optimal tracking control problem formulation is proven to outperform the one-dimensional optimal tracking control problem formulation of section 3.1 and also to guarantee closed-loop stability. In section 3.3 a general objective function for optimal operations model design is presented. Section 3.3 furthermore presents an iterative control algorithm for operations model optimization, which is proven to converge to the optimal operations model. Algorithms for output and state estimation which are required for implementation of the control algorithms in sections 3.1 – 3.3 are presented in section 3.4. The performance of the proposed algorithms is demonstrated in examples in section 3.5 and finally conclusions are given in section 3.6.

Throughout the chapter, it will be assumed that optimal solutions to stochastic OCP formulations will be independent of the replacement of stochastic variables with their conditional means. That is, it is assumed that the certainty equivalence principle \cite{Bertsekas2000} holds for all OCP formulations in the chapter. Furthermore, it will be assumed that OCP formulations can be separated into an optimal regulator problem and an optimal state estimation
3.1 Iterative Learning Control

This section first introduces Iterative Learning Control (ILC) and defines a ILC control design requirement for stochastic batch processes. The section then proceeds by proving that closed-loop stability is guaranteed when this ILC control design requirement is satisfied and some ILC OCP formulations are given. Next, optimizing implementation of the ILC OCP formulations is discussed and finally, an infinite horizon ILC OCP formulation with an optimal problem, and that this separation is optimal. That is, it is assumed that the separation theorem for linear systems with quadratic costs \cite{Bertsekas2000} holds for all OCP formulations in the present chapter. Note that the separation theorem for linear systems with quadratic costs is a manifestation of the certainty equivalence principle. A detailed discussion of these two assumptions fall outside this thesis, but can be found in \cite{Simon1956} and \cite{Striebel1965}. Note however, that the certainty equivalence principle does not hold if the system description is uncertain \cite{Bertsekas2000}, which it will indeed be if it is obtained with the methodology given in chapter \ref{chapter2} and that in this case assuming certainty equivalence principle may lead to instability \cite{deKoning1982}.

Figure 3.1: In state-of-the-art batch control systems, the nominal operations model is implemented in open-loop at PLC level. To achieve optimal and reproducible batch process operation, the batch control system is extended with optimal control problem formulations of reference tracking and operations model optimization.
learning rate is proposed.

In its traditional form (see e.g. Moore et al. 1992; Amann et al. 1998; Longman 2000; Moore 2000; Moore and Xu 2000; Gorinevsky 2002), ILC seeks to determine a control law such that the (weighted) cost of the known tracking error profile $e_{k-1}$ realized in batch $k-1$ is reduced in batch $k$. In the linear, deterministic case this means finding a correction profile $\Delta u_k$ to the control profile $u_{k-1}$ implemented in batch $k-1$, such that when the corrected control profile

$$u_k = u_{k-1} + \Delta u_k$$

is implemented in batch $k$, the traditional ILC control design requirement

$$\|e_k\|_Q^2 \leq \|e_{k-1}\|_Q^2$$

is satisfied. If the weighting matrix $Q_k$ is symmetrical and positive definite for all $k$, then the traditional ILC control design requirement (3.2) will obviously guarantee closed-loop stability for controllable, unconstrained systems. For chemical batch processes, the error profile $e_{k-1}$ is however, not known and most often the control problem formulation will be constrained.

Francois et al. (2003) discuss convergence properties of (linear) ILC for a class of nonlinear systems, whereas this section considers convergence properties of (linear) ILC of batch processes that are described by the following stochastic Linear Time-Invariant (LTI) system

$$e_k = \bar{y} - y_k = e_{k-1} - H \Delta y_{k,0} + G \Delta u_k - F v_k$$

Let the $n_y$ outputs and $n_u$ inputs of the batch process, be sampled $N + 1$ and $N$ times during a batch, respectively. Then, the error profile $e_k \in \mathbb{R}^{N n_y}$ in batch $k$ is given by a weighted sum of the error profile in the previous batch $e_{k-1}$, the effect of corrections made to the control profile $\Delta u_k \in \mathbb{R}^{N n_u}$, the effect of changes to the initial conditions $\Delta y_{k,0}$, and the effect of a stochastic zero-mean disturbance profile $v_k \in \mathbb{R}^{N n_y}$. The output profile $y_k \in \mathbb{R}^{N n_y}$ and hence its deviation from its reference profile $\bar{y}$, is observed with measurement noise

$$z_k = \begin{bmatrix} y_{k,0}' & y_k' \end{bmatrix} + \epsilon_k$$

where $\epsilon_k, \in \mathbb{R}^{(N+1) n_y}$ is a zero-mean measurement noise profile, and the error profile $e_k$ is hence uncertain.

Considering the change in initial conditions $\Delta y_{k,0}$ as an unknown zero-mean disturbance, and combining (2.10) and (3.3), the following error profile model is obtained

$$e_k = e_{k-1} + G \Delta u_k - [H \ F][v_{k,0}' \ v_k']'$$

$$= e_{k-1} + G \Delta u_k - \tilde{F} \tilde{v}_k$$

(3.5)
3.1. Iterative Learning Control

The system (3.5) is obviously controllable, i.e., the ability to move the system to a given output profile with a finite sequence of input profile corrections, if the matrix $G$ has full row-rank.

**Definition 3.1 (LTI Controllability)**
The system (3.5) is said to be controllable if and only if the finite impulse response matrix $G$ has full row-rank.

As the system (3.5) is a pure integrator, the system (3.5) is furthermore stabilizable, i.e., possesses the ability to move the unstable modes of the system to a given output profile with a finite sequence of input profile corrections, if and only if it is controllable.

Given the stochastic model (3.5) of the error profile propagation from batch to batch, the traditional ILC control design requirement (3.2) is extended to

$$E \{ \| e_k \|_{Q_k} | \mathcal{I}_k \} \leq E \{ \| e_{k-1} \|_{Q_k} | \mathcal{I}_k \}$$  \hspace{1cm} (3.6)

where $\mathcal{I}_k = \{ z_{k-1}, \Delta u_k, \mathcal{I}_{k-1} \}$, for $k \geq 0$ and $\mathcal{I}_{-1} = \{ u_{-1}, \bar{y} \}$, is the information available before batch $k$, and $Q_k$ are symmetric, positive definite weighting matrices. That is, a control law should be found such that given the information available before batch $k$, the expected tracking error cost in batch $k$ is less than or equal to the conditional mean of the tracking error realized in batch $k - 1$. In practical implementations however, it is often more convenient to consider the cost of the expected tracking error

$$||E\{e_k | \mathcal{I}_k\} ||_{Q_k}^2 = ||E\{e_{k-1} | \mathcal{I}_k\} ||_{Q_k}^2 + ||G\Delta u_k||_{Q_k}^2 + 2E\{e_{k-1} | \mathcal{I}_k\}^T Q_k G \Delta u_k$$  \hspace{1cm} (3.7)

rather then the expected cost of the tracking error

$$E \{ \| e_k \|_{Q_k} | \mathcal{I}_k \} = E \{ \| e_{k-1} \|_{Q_k} | \mathcal{I}_k \} + ||G\Delta u_k||_{Q_k}^2 + 2E\{e_{k-1} | \mathcal{I}_k\}^T Q_k G \Delta u_k + \Psi_k$$  \hspace{1cm} (3.8)

where given the covariance of the disturbance profile $\tilde{\Sigma}_k$ and the cross-covariance between the disturbance profile and the measurement noise profile $S_k$ in batch $k$, $\Psi_k$ is the expected cost of the effect of the disturbance profile $\tilde{v}$ and the effect of fed back measurement noise $\epsilon_{k-1}$

$$\Psi_k = E \{ \| \tilde{F} \tilde{v}_k \|_{Q_k}^2 | \mathcal{I}_k \} - 2E \{ \epsilon'_{k-1} Q_k \tilde{F} \tilde{v}_k | \mathcal{I}_k \}$$

$$= \text{Trace} \left( Q_k \tilde{F} \tilde{\Sigma} \tilde{F}' Q_k^\dagger \right) - 2\text{Trace} \left( Q_k S' \tilde{F}' Q_k^\dagger \right)$$  \hspace{1cm} (3.9)

---

\(^1\)For a zero-mean stochastic process $x$ and a deterministic matrix $A$ it follows that

$$E \{ x' A x \} = \text{Trace} \left( \text{Cov} (Ax) \right)$$

$$= \text{Trace} \left( A \text{Cov} (x) A' \right)$$

$$= \text{Trace} \left( A E \{ x x' \} A' \right)$$
The square-root factor $Q_k^{1/2}$ is the Cholesky factor of $Q_k$. From equations (3.7) and (3.8) it then follows that
\[
E \left\{ \|e_k\|_{Q_k}^2 \mid I_k \right\} - E \left\{ \|e_{k-1}\|_{Q_{k-1}}^2 \mid I_k \right\} = \|E \{e_k \mid I_k\}\|_{Q_k}^2 + \Psi_k - \|E \{e_{k-1} \mid I_k\}\|_{Q_{k-1}}^2
\]
(3.10)
and hence that the stochastic ILC design requirement (3.6) can be formulated as
\[
\|E \{e_k \mid I_k\}\|_{Q_k}^2 \leq \|E \{e_{k-1} \mid I_k\}\|_{Q_{k-1}}^2 - \Psi_k
\]
(3.11)
That is, a control law should be found such that given the information available before batch $k$, the cost of the expected tracking error in batch $k$ is less than or equal to the cost of the conditional mean of the tracking error realized in batch $k-1$. In theorem 3.1 it is proven that given assumption 3.1

Assumption 3.1
The norm of the weighting matrices is non-increasing
\[
\|Q_k\|_2 \leq \|Q_{k-1}\|_2
\]
then a system (3.5) that satisfy the stochastic ILC design requirement (3.11) (and thus (3.6)) will converge in the sense that the cost of the expected tracking error is non-increasing on average(!), from batch to batch. First two auxiliary lemmas are given.

**Lemma 3.1**
Let $\{c_k\}_{k=1}^{\infty}$ be a bounded sequence, then for any $1 \leq m < \infty$
\[
\lim_{M \to \infty} \frac{1}{M-m+1} \sum_{k=m}^{M} c_k = \lim_{M \to \infty} \frac{1}{M} \sum_{k=1}^{M} c_k
\]
Proof. As any one element in the sum $\lim_{M \to \infty} \frac{1}{M} \sum_{k=1}^{M} c_k$ is assigned the weight $\lim_{M \to \infty} \frac{1}{M} = 0$, it has no significance, and can thus be omitted. ■

That is, the mean of an infinite bounded sequence is independent of the initialization of the sequence. And as a consequence hereof an infinite bounded sequence with a limit point will converge to its mean.

**Lemma 3.2**
Let $\{c_k\}_{k=1}^{\infty}$ be a bounded sequence with a limit point
\[
\lim_{k \to \infty} c_k = c_\infty
\]
then
\[
c_\infty = \lim \inf_{k \to \infty} c_k = \lim \sup_{k \to \infty} c_k = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} c_k
\]
Proof. Since the mean of a sequence must lie between the minimum and maximum of that sequence
\[
\min\{c_i \mid i \in [k; \infty]\} \leq \lim_{N \to \infty} \frac{1}{N - k + 1} \sum_{i=k}^{N} c_i \leq \max\{c_i \mid i \in [k; \infty]\}
\]
and convergence implies that
\[
c_\infty = \lim_{k \to \infty} \min\{c_i \mid i \in [k; \infty]\} = \lim_{k \to \infty} \max\{c_i \mid i \in [k; \infty]\}
\]
From lemma 3.1 it follows that
\[
\lim_{k \to \infty} \lim_{N \to \infty} \frac{1}{N - k + 1} \sum_{i=k}^{N} c_i = \lim_{k \to \infty} \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} c_i = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} c_i
\]
and hence that
\[
\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} c_i = c_\infty
\]

Next the convergence properties of a system (3.5) satisfying the stochastic ILC design requirement (3.11) are proven in theorem 3.1.

Theorem 3.1
Let there exist four bounded sequences \(\{y_k\}_{k=1}^{\infty}, \{\Psi_k\}_{k=1}^{\infty}, \{\|E\{e_k \mid I_k\}\|_{Q_k}^2\}_{k=1}^{\infty}\) and \(\{\|E\{e_{k-1} \mid I_k\}\|_{Q_k}^2\}_{k=1}^{\infty}\) for which the following applies
\[
\Psi = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} y_k = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \Psi_k
\]
\[
\Psi_\infty = \min \left\{ \|E\{e_k \mid I_k\}\|_{Q_k}^2 \mid k \in [1, \infty] \right\}
\]
such that the following two inequalities are true for \(k \geq 2\)
\[
\|E\{e_{k-1} \mid I_k\}\|_{Q_k}^2 \geq \|E\{e_k \mid I_k\}\|_{Q_k}^2 + \Psi_k
\]
\[
\|E\{e_{k-1} \mid I_k\}\|_{Q_k}^2 \leq \|E\{e_{k-1} \mid I_{k-1}\}\|_{Q_k_{k-1}}^2 + y_{k-1}
\]

\(^2\)Note that the term ‘convergence’ is used in the meaning of ‘convergence to some ball in state space’.
Then the sequence \( \{ \| E_e | I_k \|_{Q_k}^2 \}_{k=1}^\infty \) will converge
\[
\lim_{k \to \infty} \| E_e | I_k \|_{Q_k}^2 = \Psi_e
\]
and the sequence \( \{ \| E_e \mid I_k \|_{Q_k}^2 \}_{k=1}^\infty \) will be attracted to and distributed around the mean
\[
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^N \| E_e \mid I_k \|_{Q_k}^2 = \Psi_e + \bar{\Psi}
\]

**Proof.** From the two inequalities
\[
\| E_e - 1 | I_k \|_{Q_k}^2 \geq \| E_e | I_k \|_{Q_k}^2 + \Psi_k
\]
\[
\| E_e - 1 | I_k \|_{Q_k}^2 \leq \| E_e - 1 | I_k - 1 \|_{Q_k - 1}^2 + y_{k-1}
\]
it follows for \( k \geq 2 \), that
\[
\| E_e - 1 | I_k - 1 \|_{Q_k - 1}^2 - \| E_e | I_k \|_{Q_k}^2 \geq \Psi_k - y_{k-1}
\]
and given lemma 3.1, hence that
\[
\lim_{N \to \infty} \frac{1}{N} \sum_{k=m+1}^{N+m} \| E_e - 1 | I_k - 1 \|_{Q_k - 1}^2 - \| E_e | I_k \|_{Q_k}^2 \geq \Psi - \bar{\Psi}
\]
\[
\geq 0
\]
for any \( m \geq 1 \). Consequently, the sequence \( \{ \| E_e | I_k \|_{Q_k}^2 \}_{k=1}^\infty \) is decreasing on average. As the weighting matrices \( Q_k \) are positive definite and given assumption 3.1, their norm is non-increasing, there must exist a limit point
\[
\lim_{k \to \infty} \| Q_k \|_2 = \| Q_{\infty} \|_2 > 0
\]
which in turn means that the sequence \( \{ \| E_e | I_k \|_{Q_k}^2 \}_{k=1}^\infty \) has an attractor, and hence given lemma 3.2, that
\[
\lim_{k \to \infty} \| E_e | I_k \|_{Q_k}^2 = \Psi_e
\]
It furthermore follows that
\[
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^N \| E_e - 1 | I_k \|_{Q_k}^2 \geq \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^N \| E_e | I_k \|_{Q_k}^2 + \Psi_k
\]
\[
\geq \Psi_e + \bar{\Psi}
\]
and that
\[
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^N \| E_e - 1 | I_k \|_{Q_k}^2 \leq \lim_{N \to \infty} \frac{1}{N} \sum_{k=2}^N \| E_e - 1 | I_k - 1 \|_{Q_k - 1}^2 + y_{k-1}
\]
\[
\leq \Psi_e + \bar{\Psi}
\]
and consequently it must follow that

\[
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \|E\{e_{k-1} \mid I_k\}\|^2_{Q_k} = \Psi_k + \Psi
\]

\[\blacksquare\]

Remark 3.1

Let \( y_{k-1} \) represent the weighted variance contribution from realized process disturbances and fed back estimation errors, then the inequality

\[
\|E\{e_{k-1} \mid I_k\}\|^2_{Q_k} \leq \|E\{e_{k-1} \mid I_{k-1}\}\|^2_{Q_{k-1}} + y_{k-1}
\]

is a manifestation of assumption 3.1 and the system eq. (3.5), and it furthermore follows that

\[
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} y_k = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \Psi_k
\]

That is, theorem 3.1 states that given assumption 3.1, the system (3.5) is guaranteed to converge if the stochastic ILC design requirement (3.6) is satisfied for all \( k \geq 0 \). Furthermore, combining eq.'s (3.7) and (3.11) it is found that the stochastic ILC design requirement (3.6) is satisfied for batch \( k \) if the control profile correction \( \Delta u_k \) satisfies the following inequality

\[
\Psi_k \leq -\|G\Delta u_k\|^2_{Q_k} - 2E\{e_{k-1} \mid I_k\}'Q_kG\Delta u_k
\]

which is easily evaluated before the control profile correction \( \Delta u_k \) is implemented in batch \( k \). That is, based on the evaluation of the inequality (3.12), a control profile correction \( \Delta u_k \) can either be accepted as stabilizing and implemented or rejected, redesigned and reevaluated.

If the batch control problem at hand is not input and/or output constrained, one way of determining control profile corrections \( \Delta u_k \), that obey the stochastic ILC design requirement (3.6), is to solve the following (standard ILC) OCP

\[
\Delta u^*_k = \arg \min_{\Delta u_k} \left[ E\{e_k'Q_k e_k \mid I_k\} + \Delta u_k'R_k \Delta u_k \right] \\
\text{s.t.} \quad e_k = e_{k-1} + G\Delta u_k - F\tilde{v}_k
\]

where the symmetrical weighting matrices \( Q_k \) and \( R_k \) are positive definite.

Remark 3.2

There are a couple of important remarks to be made about the above proposed objective function. Firstly, the objective function is a Tikhonov Regularization
(and Ridge Regression if $R_k$ is diagonal) problem formulation and as such the optimal control profile corrections $\Delta u_k^*$ are found in a lower dimensional space. Secondly, a quadratic objective function of the form

$$J_k = \tilde{x}_k' \tilde{Q} \tilde{x}_k + u_k' \tilde{R} u_k$$

where $\tilde{x}_k$ is an augmented state vector and $u_k$ is a parameter vector, is quite general and comprises more elaborate parameter penalty functions such as the widely used and strongly recommended, penalty on the approximated first order derivative of the parameter vector, i.e.,

$$J_k = x_k' Q x_k + (u_k - u_{k-1})' S (u_k - u_{k-1}) + u_k' R u_k$$

where $x_k$ is a state vector. Thirdly, the weighting matrix $\tilde{Q}$ need note be positive definite if the state vector $\tilde{x}_k$ is augmented, but the weighting matrices $Q$ and $\tilde{R}$ must be positive definite (otherwise some theorems and propositions below will not be true).

It can be easily shown (Bertsekas, 2000) that the certainty equivalence principle holds for the OCP (3.13) and that the solution is explicitly given as

$$\Delta u_k^* = - (G' Q_k G + R_k)^{-1} G' Q_k E\{e_{k-1} \mid I_k\}$$

$$= - K_{ILC}^k E\{e_{k-1} \mid I_k\}$$

(3.14)

The OCP (3.13) is well described in ILC literature, but it has only been proven to converge for deterministic systems. The present thesis has however already presented a design requirement which when satisfied, guarantees that the OCP (3.13) converges for the stochastic system (3.5). In proposition 3.1 given below, the optimal solution (3.14) to the OCP (3.13), is proven to satisfy the stochastic ILC design requirement (3.6) and thus guarantee convergence, if the weighting matrix pairs $(Q_k, R_k)$, $\forall k$, are designed such that

$$\Psi_k \leq E\{e_{k-1} \mid I_k\}' \Gamma_k E\{e_{k-1} \mid I_k\}$$

(3.15)

where $\Gamma_k$ is given as

$$\Gamma_k = K_{ILC}^k G' Q_k + K_{ILC}^k R_k K_{ILC}^k$$

(3.16)

Note that the inequality (3.15) can be used to evaluate and if necessary, redesign the weighting matrices $Q_k$ and $R_k$ for guaranteed convergence, before every batch.

**Proposition 3.1**

Let the system (3.5) be governed by the ILC control law (3.14), then the closed-loop system

$$e_k = e_{k-1} + G \Delta u_k^* - \tilde{F} \tilde{v}_k$$

will satisfy the ILC design requirement (3.6) if and only if

$$\Psi_k \leq E\{e_{k-1} \mid I_k\}' \Gamma_k E\{e_{k-1} \mid I_k\}$$
Proof. First, assume that
\[ \Psi_k \leq E \{ e_{k-1} \mid I_k \} \Gamma_k E \{ e_{k-1} \mid I_k \} \]
\[ \leq E \{ e_{k-1} \mid I_k \} \left( 2K^{ILC}_k G'Q_k - K^{ILC}_k G'Q_k G_k K^{ILC}_k \right) E \{ e_{k-1} \mid I_k \} \]
\[ \leq -2\Delta u_k^\ast G'Q_k E \{ e_{k-1} \mid I_k \} - \Delta u_k^\ast G'Q_k G_k \Delta u_k^\ast \]
and it follows that
\[ E \left\{ \| e_{k-1} \|_{Q_k}^2 \mid I_k \right\} \leq \Psi_k \]
and it follows that
\[ E \left\{ \| e_{k-1} \|_{Q_k}^2 \mid I_k \right\} \geq E \left\{ \| e_{k-1} \|_{Q_k}^2 \mid I_k \right\} + 2\Delta u_k^\ast G'Q_k E \{ e_{k-1} \mid I_k \} \]
\[ + \Delta u_k^\ast G'Q_k G_k \Delta u_k^\ast + \Psi_k \]
Then, assume that
\[ E \{ \| e_{k-1} \|_{Q_k}^2 \mid I_k \} < E \{ \| e_{k-1} \|_{Q_k}^2 \mid I_k \} \]
\[ \leq E \{ \| e_{k-1} \|_{Q_k}^2 \mid I_k \} + 2\Delta u_k^\ast G'Q_k E \{ e_{k-1} \mid I_k \} \]
\[ + \Delta u_k^\ast G'Q_k G_k \Delta u_k^\ast + \Psi_k \]
and it follows that
\[ \Psi_k > -2\Delta u_k^\ast G'Q_k E \{ e_{k-1} \mid I_k \} - \Delta u_k^\ast G'Q_k G_k \Delta u_k^\ast \]
\[ > E \{ e_{k-1} \mid I_k \} \Gamma_k E \{ e_{k-1} \mid I_k \} \]

Proposition 3.1 thus states that the ILC design requirement (3.6) is satisfied by the ILC control law (3.14), when the new design requirement (3.15) is satisfied. Proposition 3.1 furthermore states that the ILC control law (3.14) is stabilizing, when the new design requirement (3.15) is satisfied. This closed-loop stability of the ILC control law (3.14) is proven in proposition 3.2.

Proposition 3.2
Let the system (3.5) be governed by the ILC control law (3.14), then given assumption 3.1, the closed-loop system
\[ e_k = e_{k-1} + G\Delta u_k^\ast - \tilde{F} \tilde{v}_k \]
will converge a.e. (almost everywhere) to its origin, in the sense that
\[ \lim_{k \to \infty} E \{ e_k \mid I_k \} = 0 \]
if \( \forall k \)
\[ \Psi_k \leq E \{ e_{k-1} \mid I_k \} \Gamma_k E \{ e_{k-1} \mid I_k \} \]
\textbf{Proof.} It follows from theorem \ref{thm:3.1} and proposition \ref{prop:3.1} that
\[ \lim_{k \to \infty} \|E\{e_k \mid I_k\}\|_Q^2 = 0 \]
and finally, as $Q_k$ is positive definite \forall $k$, that
\[ \lim_{k \to \infty} E\{e_k \mid I_k\} = 0 \]
if \forall $k$
\[ \Psi_k \leq E\{e_{k-1} \mid I_k\}^\Gamma_k E\{e_{k-1} \mid I_k\} \]
\[ \square \]

\textbf{Remark 3.3}
The closed-loop system is only convergent a.e. because the cost of the expected tracking error may be constant and nonzero. In any case, the closed-loop system is non-divergent.

\textbf{Remark 3.4}
Note that, a weighting matrix pair $(Q_k, R_k)$ that satisfy \eqref{eq:3.15}, need not exist. E.g. if the system \eqref{eq:3.5} is not controllable, i.e., $G$ does not have full row-rank, it will obviously not be stabilizable.

\textbf{Remark 3.5}
The inequality
\[ \Psi_k \leq E\{e_{k-1} \mid I_k\}^\Gamma_k E\{e_{k-1} \mid I_k\} \]
can only be true for \forall $k$ if the covariances of the disturbances signals converge to zero (i.e., if $\lim_{k \to \infty} \tilde{\Sigma}_k = 0$ and $\lim_{k \to \infty} S_k = 0$) or if the weighting matrices $R_k$ converge to zero (i.e., if $\lim_{k \to \infty} R_k = 0$). These restrictions are discussed further below.

If on the other hand, the batch control problem at hand is input and/or output constrained (constrained ILC is discussed in e.g. \cite{Chen99, Lee00} and the admissible input space $\mathcal{U}$ in batch $k$ is defined as
\[
\mathcal{U}_k(I_k) = \{ \Delta u \mid u_{\min} \leq u_k \leq u_{\max}, y_{\min} \leq E\{y \mid I_k\} \leq y_{\max} \}
\]
\[
= \{ \Delta u_k \mid h(I_k) \leq 0 \}
\]
(3.17)
where $h(I_k)$ is given as
\[
h(I_k) = \begin{bmatrix}
    u_{\min} - u_k \\
    -u_{\max} + u_k \\
    y_{\min} - E\{y \mid I_k\} \\
    -y_{\max} + E\{y \mid I_k\}
\end{bmatrix}
\]
(3.18)
another way of determining control profile corrections $\Delta u_k$, that obey the stochastic ILC design requirement (3.6) and thus achieve the convergence stated in theorem 3.1, is to solve assuming that a solution exists, the following constrained OCP

$$
\Delta u_k^* = \arg\min_{\Delta u_k} \left[ E\{e'_kQ_k e_k \mid T_k\} + \Delta u'_k R_k \Delta u_k \right] \\
\text{s.t.} \ e_k = e_{k-1} + G\Delta u_k - F\hat{v}_k
$$

(3.19)

where the weighting matrix pairs $(Q_k, R_k)$, $\forall k$, are designed such that eq. (3.12) is satisfied for every batch. The constrained OCP (3.19) is solved as a Quadratic Programming (QP) problem (see e.g. Gill et al., 1981). Alternatively, the inequality (3.12) can be added as an additional constraint to the OCP (3.19) and the augmented OCP is then solved as a quadratically constrained QP problem (see e.g. Boyd and Vandenberghe, 2004).

Thus far, the two ILC OCP formulations (3.13) and (3.19) have been proven to guarantee closed-loop convergence if the stochastic ILC design requirement (3.6) is satisfied for all batches. But, is it feasible for the two ILC OCP formulations (3.13) and (3.19) to satisfy the stochastic ILC design requirement (3.6) for all batches? The answer is unfortunately, no. Because the weighting matrices $R_k$ are positive definite, in batch $k$, the optimal control profile correction $\Delta u_k^*$ given by (3.13) or (3.19), will never completely reject the effect of the disturbance profile in the previous batch $\hat{v}_{k-1}$. The next question then is, whether it is necessary to satisfy the stochastic ILC design requirement (3.6) for all batches in order to guarantee closed-loop convergence? The answer to that question is fortunately, no. What is necessary and sufficient for guaranteed closed-loop convergence is that the stochastic ILC design requirement (3.6) is satisfied on average. I.e., that the following necessary and sufficient stochastic ILC design requirement is satisfied

$$
\lim_{M \to \infty} \frac{1}{M} \sum_{k=1}^{M} E\left\{ \|e_k\|^2_{Q_k} \mid T_k \right\} - E\left\{ \|e_{k-1}\|^2_{Q_k} \mid T_k \right\} \leq 0
$$

(3.20)

or given (3.12), equivalently

$$
\lim_{M \to \infty} \frac{1}{M} \sum_{k=1}^{M} \rho_k(\Delta u_k) \leq 1
$$

(3.21)

where the convergence indicator $\rho_k$ based on eq. (3.12), is given as

$$
\rho_k(\Delta u_k) = \frac{-\Psi_k}{\|G\Delta u_k\|^2_{Q_k} + 2E\{e_{k-1} \mid T_k\}' Q_k G \Delta u_k}
$$

(3.22)

Although the necessary and sufficient stochastic ILC design requirement (3.21) is infeasible to compute, it can be estimated and monitored based on an appropriately sized window of past and present batches. The convergence indicator
\(\rho_k\) is however easily computed and when \(\rho_k \leq 1\), then the two ILC OCP formulations (3.13) and (3.19) will guarantee convergence. More importantly, as \(\Psi_k\) is the average/expected cost of the disturbance profile \(v_k\) (and a feedback part of the measurement noise \(e_{k-1}\)), the two ILC OCP formulations (3.13) and (3.19) have the potential to satisfy the inequality \(\rho_k(\Delta u_k^*) \leq 1\) on average and thus satisfy the necessary and sufficient stochastic ILC design requirement (3.20). The definition of the necessary and sufficient stochastic ILC design requirement (3.20) leads to the convergence theorem 3.2 that follows below.

**Theorem 3.2**

Let there exist two bounded sequences \(\{y_k\}_{k=1}^\infty\) and \(\{\Psi_k\}_{k=1}^\infty\) for which the common mean is given as

\[
\bar{\Psi} = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} y_k = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \Psi_k
\]

and a bounded sequence \(\{\|E\{e_k | I_k\}\|_{Q_k}^2\}_{k=1}^\infty\) for which the minimum feasible mean is given as

\[
\Psi_k = \min_{\{E\{\hat{e}_k | I_k\}\}_{k=1}^\infty} \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \|E\{\hat{e}_k | I_k\}\|_{Q_k}^2
\]

\[
\leq \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \|E\{e_k | I_k\}\|_{Q_k}^2
\]

and a bounded sequence \(\{\|E\{e_{k-1} | I_k\}\|_{Q_k}^2\}_{k=1}^\infty\), such that the following two inequalities are true for \(k \geq 2\)

\[
\frac{1}{N} \sum_{k=2}^{N} \|E\{e_{k-1} | I_k\}\|_{Q_k}^2 \leq \|E\{e_k | I_k\}\|_{Q_k}^2 - \Psi_k \geq 0
\]

\[
\|E\{e_{k-1} | I_k\}\|_{Q_k}^2 \leq \|E\{e_{k-1} | I_{k-1}\}\|_{Q_{k-1}}^2 + y_{k-1}
\]

Then the sequence \(\{\|E\{e_k | I_k\}\|_{Q_k}^2\}_{k=1}^\infty\) will be attracted to and distributed around the minimum feasible mean

\[
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \|E\{e_k | I_k\}\|_{Q_k}^2 = \Psi_k
\]

and the sequence \(\{\|E\{e_{k-1} | I_k\}\|_{Q_k}^2\}_{k=1}^\infty\) will be attracted to and distributed around the mean

\[
\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \|E\{e_{k-1} | I_k\}\|_{Q_k}^2 = \Psi_k + \bar{\Psi}
\]
Proof. The proof is analogous to that of theorem 3.1.

That is, as long as the convergence indicator \( \rho_k \) is less than or equal to one or the necessary and sufficient stochastic ILC design requirement (3.21) is satisfied, then the two ILC OCP formulations (3.13) and (3.19) guarantee convergence for the closed-loop system

\[
e_k = e_{k-1} + G\Delta u^*_k - \tilde{F}\tilde{v}_k
\]  

(3.23)

The convergence properties of the closed-loop system stated in theorem 3.2 are demonstrated in examples 3.1 and 3.2 below.

Example 3.1
In this example, the properties of the above proposed ILC methodologies are demonstrated through control of the following unconstrained LTI system

\[
e_k = e_{k-1} + G\Delta u^*_k - \tilde{F}\tilde{v}_k
\]

where \( \Delta u^*_k \) is given by the ILC OCP (3.13) and the historic profiles \( e_k \) for \(-10,000 \leq k \leq 0\), are simulated with \( \Delta u^*_{k-1} = 0 \). The system matrices are given in detail in example 3.4 on page 84. The disturbance profiles \( \tilde{v}_k \) are drawn from a zero-mean Gaussian distribution with covariance \( \tilde{\Sigma} = 0.001I \) and the error profile estimate \( E\{e_{k-1} | I_k\} \) is modeled as

\[
E\{e_{k-1} | I_k\} = e_{k-1} + \epsilon_{k-1}
\]

where the estimation error profiles \( \epsilon_k \) are drawn from a zero-mean Gaussian distribution with covariance \( S = 0.0001I \). The controller is designed using the true system and the weighting matrices \( Q_k \) and \( R_k \) are designed as

\[
Q_k = R_k = I
\]

The controller performance in terms of the cost \( \| e_k \|^2_{Q_k} \) and the convergence indicator \( \rho_k \), is plotted in figure 3.2 for 5000 batches. The convergence indicator \( \rho_k \) is clearly less than one on average and in concurrence herewith, the cost \( \| e_k \|^2_{Q_k} \) decreases to a steady state level. As this system is driven by random disturbances there is not much for the controller to learn, and the learning rate or cost reduction is thus rather slow. The batch process reproducibility is however greatly improved with the application of the proposed ILC OCP formulation.

Example 3.2
In this next example, the properties of the above proposed ILC methodologies are demonstrated through control of the following unconstrained LTI system

\[
e_k = e_{k-1} - H\Delta y_{k,0} + G\Delta u^*_k - Fv_k
\]
Figure 3.2: This figure shows the weighted tracking error cost $\|e_k\|_Q^2$ and the convergence indicator $\rho_k(\Delta u_k^*)$ achieved on the system in example 3.1, with the ILC OCP formulation (3.13). The system runs in open-loop up to batch 1, hereafter the system is under closed-loop control from batch to batch. As seen from both the tracking error cost and the convergence indicator, the system converges to an improved steady state reproducibility level.

where the control corrections $\Delta u_k^*$ are given by the ILC OCP (3.13) and the initial conditions are $y_{k,0} = \begin{bmatrix} 1 & -1 \end{bmatrix}'$ for $k \geq 6$. The tracked reference profile is $\bar{y} = 0$. The historic profiles $e_k$ for $-12,500 \leq k \leq 5$, are simulated in closed-loop with the initial conditions $y_{k,0} = 0$. The system matrices are given in detail in example 3.4 on page 84. The disturbance profiles $v_k$ are drawn from a zero-mean Gaussian distribution with covariance $\Sigma = 0.001I$ and the error profile estimate $E\{e_{k-1} | I_k\}$ is modeled as

$$E\{e_{k-1} | I_k\} = e_{k-1} + \epsilon_{k-1}$$

where the estimation error profiles $\epsilon_k$ are drawn from a zero-mean Gaussian distribution with covariance $\mathcal{S} = 0.0001I$. The controller is designed using the true system and the weighting matrices $Q_k$ and $R_k$ are designed as

$$Q_k = 0.01R_k = I$$

The controller performance in terms of the cost $\|e_k\|_Q^2$ and the convergence indicator $\rho_k$, is plotted in figure 3.3 for 500 batches. The convergence indicator
3.1. Iterative Learning Control

Figure 3.3: This figure shows the weighted tracking error cost $\|e_k\|^2_{Q_k}$ and the convergence indicator $\rho_k(\Delta u_k^*)$ achieved on the system in example 3.2 with the ILC OCP formulation (3.13). The system is subject to a persistent change in the initial conditions in batch 6 and subsequent batches. As seen from both the tracking error cost and the convergence indicator, the system converges to the steady state reproducibility level achieved prior to the persistent change of initial conditions.

$\rho_k$ is clearly less than one on average and in concurrence herewith, the cost $\|e_k\|^2_{Q_k}$ decreases to a steady state level. As this system is driven by random disturbances and by a persistent perturbation of the initial conditions $y_{k,0}$, the controller learns during the course of approximately 20 batches, how to reject the effects of this persistent disturbance. The system consequently returns to level of reproducibility achieved prior to the persistent perturbation of the initial conditions.

**Complete Disturbance Rejection**

Alternatives to the two ILC OCP formulations (3.13) and (3.19), that have the ability to (possibly) completely reject the effect of the disturbance profile in the previous batch $\tilde{v}_k$, are the following ILC OCP formulation

$$
\Delta u_k^*_k = \arg \min_{\Delta u_k} [E\{e_k'Q_k e_k \mid I_k\}]
$$

$$
s.t. \quad e_k = e_{k-1} + G\Delta u_k - \tilde{F}\tilde{v}_k
$$

(3.24)
and its constrained analogue
\[
\Delta \hat{u}_k^* = \arg \min_{\Delta u_k} \left[ E\{e_k'Q_k e_k \mid I_k\} \right] \\
\text{s.t. } e_k = e_{k-1} + G\Delta u_k - \tilde{F}\tilde{v}_k \\
h(I_k) \leq 0
\] (3.25)

These two new ILC OCP formulations (3.24) and (3.25) are however, intractable for the following three reasons. Firstly, for the two new ILC OCP formulations (3.24) and (3.25), to have a solution it is necessary to assume that the matrix product \( G'G \) is positive definite.

**Assumption 3.2**
The matrix product \( G'G \) is positive definite.

Secondly, if solutions exist to the two new ILC OCP formulations (3.24) and (3.25), these solutions are not unique. For the abovementioned reasons it seems that an ILC OCP formulation that fuse the positive while eliminating the negative qualities of the ILC OCP formulations stated above, will be a tractable alternative. It is thus proposed to substitute the above stated the ILC OCP formulations with the following blended ILC OCP formulation
\[
\Delta \hat{u}_k = (1 - \psi_k)\Delta u_k^* + \psi_k \Delta \hat{u}_k^* \\
\] (3.26)

where the weighting function \( \psi_k \) is given as a sigmoidal function
\[
\psi_k = \frac{1}{1 + \exp\left(1 - \rho_k(\Delta u_k^*)^{\alpha_\psi}\right)}
\] (3.27)

the convergence indicator \( \rho_k(\Delta u_k^*) \) is computed given the solution to the ILC OCP formulation (3.13) or (3.19), and \( \alpha_\psi \) is a scalar nonnegative tuning parameter. An alternative optimal blended ILC OCP formulation is
\[
\Delta \hat{u}_k = \arg \min_{\Delta u_k} \left[ E\{e_k'Q_k e_k \mid I_k\} + (1 - \psi_k)\Delta u_k'R_k \Delta u_k \right] \\
\text{s.t. } e_k = e_{k-1} + G\Delta u_k - \tilde{F}\tilde{v}_k \\
h(I_k) \leq 0
\] (3.28)

with or without the inequality constraints. The convergence properties of the two blended ILC OCP formulations (3.26) and (3.28) can then be monitored by estimating and monitoring the convergence indicator \( \rho_k(\Delta \hat{u}_k) \) based on an appropriately sized window of past and present batches.

### 3.1.1 Optimizing Learning Control

At this point it is relevant return to theorem 3.2 (on page 62) and recall that convergence is guaranteed independent of whether or not the output reference profile \( \bar{y} \) is reachable. Intuitively, it may not seem feasible to design an
unreachable output reference profile, but when properly designed, such an unreachable output reference profile may actually improve the performance of a batch process. This is significant as most controller convergence proofs in literature assume reachability. E.g., assume that a best case, i.e., the case with no disturbances, optimal output reference profile $\bar{y}^*$ is known and that this reference profile can only be reached if no disturbances influence the batch process. Then this best case optimal output reference profile would not be reachable in practice, but it would certainly be desirable and may be optimal to attempt to track it. One such scenario could be a best case optimal output reference profile reachable when utilities are available at 100% of their capacities, but unreachable when due to scheduled or unscheduled downtime (or due to the lack of an advanced control system, the utility production capacity is not fully utilized, or due to the lack of an advanced batch scheduling system, the utility production capacity is insufficient), the utilities are available at less than 100% of their capacities. The standard approach is to replace the best case optimal output reference profile with a worst case reachable output reference profile, but tracking this worst case reachable output reference profile will obviously cause sub-optimal batch process performance. Instead the present thesis proposes to track the best case optimal output reference profile or a estimate hereof, with the proposed ILC OCP formulation (3.19), and hence utilize favorable disturbances to get close to the optimal output reference profile while rejecting unfavorable disturbances. Tracking the best case optimal output reference profile with the ILC OCP formulation (3.19) will thus in the worst case scenario yield the same batch process performance as would have been achieved tracking the worst case reachable output reference profile, and in the best case scenario yield the optimal batch process performance. I.e., tracking the best case optimal output reference profile or a guesstimate hereof, with the ILC OCP formulation (3.19) comprises an optimizing ILC OCP formulation, that achieves the best possible batch process operation given the disturbance scenario. Obviously the optimizing ILC OCP formulation will not (necessarily) yield reproducible batch process operation. The properties of this optimizing ILC OCP formulation are demonstrated in the example given below.

**Example 3.3**
In this example, the properties of the above proposed optimizing ILC methodologies are demonstrated through control of the following constrained LTI system

$$e_k = e_{k-1} + G\Delta u^*_k - \tilde{F}\bar{v}_k$$

where the control corrections $\Delta u^*_k$ are given by the ILC OCP (3.19) with the input constraints $-250 \leq u_k \leq 250$. The tracked reference profile is $\bar{y} = 0$ for $-10,000 \leq k \leq 0$. The system matrices are given in detail in example 3.4 on page 84. The control objective in this example is to maximize the cost function...
\( q y_k \), where the weighting vector \( q \) is given as
\[
q = [1 \quad -1 \quad 1 \quad -1 \quad \ldots \quad 1 \quad -1]
\]
in every batch \( k \). To achieve this control objective, the following two reference profiles are implemented in batches \( k \geq 1 \):

- A worst case reachable reference profile \( \tilde{y} = 2q \) for which the input constraints are inactive \( \forall k \).
- An unreachable reference profile \( \tilde{y} = 20q \) for which the input constraints are active.

The disturbance profiles \( \tilde{v}_k \) are drawn from a zero-mean Gaussian distribution with covariance \( \Sigma = 0.001I \) and it is furthermore assumed that the error profile estimate \( E\{e_{k-1} | I_k\} \) is given as
\[
E\{e_{k-1} | I_k\} = e_{k-1} + \hat{e}_{k-1}
\]
where the estimation error profiles \( \hat{e}_k \) are drawn from a zero-mean Gaussian distribution with covariance \( S = 0.0001I \). The controller is designed using the true system and the weighting matrices \( Q_k \) and \( R_k \) are designed as
\[
Q_k = R_k = I
\]
The controller performance in terms of the cost \( q y_k \) and the convergence indicator \( \rho_k \), for the two reference profiles is plotted in figure 3.4 for 1000 batches. The convergence indicator \( \rho_k \) is clearly less than one on average and in concurrence herewith, the cost \( \|e_k\|_{Q_k}^2 \) decreases to a steady state level. It is also obvious from figure 3.4 that the unreachable reference profile achieves the control objective significantly better than the worst case reachable reference profile.

### 3.1.2 Optimally Learning Control

Although it guarantees convergence, the stochastic ILC design requirement (3.6) is not optimal in terms of learning or convergence rates (see e.g. Xu and Tan, 2002), especially when batch processes are subject to (active) input and/or output constraints. To ensure optimal learning, i.e., the fastest convergence rate of \( E\{e_k | I_k\} \), the stochastic ILC control design requirement (3.6) is relaxed to
\[
\lim_{k \to \infty} \|E\{e_k | I_k\}\|_{Q_k}^2 = \Psi_{e}
\]
where \( \Psi_{e} \) is the minimum achievable cost of the expected tracking error profile
\[
\Psi_{e} = \min \left\{ \|E\{e_i | I_i\}\|_{Q_i}^2 \mid i \in [0; \infty] \right\}
\]
Furthermore, assume that there exists a sequence of admissible input profiles.
Figure 3.4: This figure shows the cost function $qy_k$ and the convergence indicator $\rho_k(\Delta u^*_k)$ achieved on the system in example 3.3 with the constrained ILC OCP formulation (3.19). The level of the cost function $qy_k$ achieved with an unreachable reference profile (top figure on the right hand side) is significantly better than that achieved with a reachable reference profile (top figure on the left hand side). As seen from the convergence indicators (bottom figure on the left hand side and bottom figure on the right hand side for the reachable and unreachable reference profiles, respectively), the system converges to steady state reproducibility levels for both reference profiles.

**Assumption 3.3**

There exists a sequence of admissible control profile corrections

$$\Delta u_k \in \mathcal{U}_k(I_k), \quad k = 0, 1, \ldots, \infty$$

That is, assume that the system (3.5) can be kept within its constraints in closed-loop. Then for input-output constrained batch processes, ILC with an optimal learning rate, obeying the relaxed stochastic ILC design requirement
can be formulated as the following infinite horizon OCP

\[
\{\Delta u^*_k\}_{k=0}^\infty = \arg \min \left\{ \sum_{k=0}^\infty E\{e_k | I_k\}'QE\{e_k | I_k\} + \Delta u^*_k'R\Delta u_k \right\}
\]

s.t. \( e_k = e_{k-1} + G\Delta u_k - \tilde{F}\tilde{v}_k \)

\[
h(I_k) \leq 0
\]

(3.31)

Note that it is assumed that the weighting matrices \( Q \) and \( R \) as well as the covariance matrices \( \tilde{\Sigma} \) and \( S \), are constant (which means that also \( \Psi \) will be constant). In proposition 3.4 the solution \( \{\Delta u^*_k\}_{k=0}^\infty \) to the infinite horizon OCP (3.31), is proven to satisfy the relaxed stochastic ILC design requirement (3.29) and thus, to guarantee convergence. At this point the following important lemma (see e.g. Rockafellar, 1970) for convex functions is introduced

**Lemma 3.3 (Jensen’s Inequality)**

If \( \lambda_i \geq 0, i = 1, 2, \ldots, m, \sum_{i=1}^m \lambda_i = 1 \) and \( f \) is a convex function, then

\[
f \left( \sum_{i=1}^m \lambda_i x_i \right) \leq \sum_{i=1}^m \lambda_i f(x_i)
\]

**Proof.** For the case \( m = 2 \), Jensen’s Inequality is true by the definition of a convex function. The proof can thus be given by induction for \( m > 2 \), by assuming that Jensen’s Inequality holds for \( m-1 \). That is, for a convex function it follows that

\[
f \left( \sum_{i=1}^m \lambda_i x_i \right) = f \left( \sum_{i=1}^{m-1} \lambda_i x_i + \lambda_m x_m \right)
\]

\[
= f \left( \frac{1 - \lambda_m}{1 - \lambda_m} \sum_{i=1}^{m-1} \lambda_i x_i + \lambda_m x_m \right)
\]

\[
\leq (1 - \lambda_m) f \left( \frac{1}{1 - \lambda_m} \sum_{i=1}^{m-1} \lambda_i x_i \right) + \lambda_m f(x_m)
\]

\[
\leq (1 - \lambda_m) f \left( \sum_{i=1}^{m-1} \frac{\lambda_i}{1 - \lambda_m} x_i \right) + \lambda_m f(x_m)
\]
and by induction it follows that
\[
f\left(\sum_{i=1}^{m} \lambda_i x_i \right) \leq (1 - \lambda_m) \sum_{i=1}^{m-1} \lambda_i \frac{1}{1 - \lambda_m} f(x_i) + \lambda_m f(x_m)
\]
\[
\leq \sum_{i=1}^{m-1} \lambda_i f(x_i) + \lambda_m f(x_m)
\]
\[
\leq \sum_{i=1}^{m} \lambda_i f(x_i)
\]

\[\blacksquare\]

**Remark 3.6**

Jensen’s Inequality is perhaps best known for the following special (but important) case

\[
f(E\{x\}) \leq E\{f(x)\}
\]

where \(f\) is a convex function and \(x\) is a stochastic process.

Next, Jensen’s Inequality is restated in proposition 3.3 in the present context.

**Proposition 3.3**

Let \(\{x_k\}_{k=1}^{M}\) be a bounded state sequence and let the strictly convex function \(f\) be the cost associated with a state \(x_k\). Furthermore, let the mean state and the mean cost be defined as

\[
x_m(M) = \frac{1}{M} \sum_{k=1}^{M} x_k
\]

\[
f_m(M) = \frac{1}{M} \sum_{k=1}^{M} f(x_k)
\]

and the state bounds as

\[
\underline{x}(1,M) = \min\{x_k \mid k \in [1;M]\}
\]

\[
\overline{x}(1,M) = \max\{x_k \mid k \in [1;M]\}
\]

Then

\[
f(x_m(M)) \leq f_m(M)
\]

and, if and only if

\[
\underline{x}(1,M) = x_m(M) = \overline{x}(1,M)
\]
then
\[ f(x_m(M)) = f_m(M) \]

**Proof.** The first statement follows directly from Lemma 3.3. The second statement also follows from Lemma 3.3 as the cost function \( f \) is strictly convex. ■

Theorem 3.3 then states a property which is inherent to systems associated with strictly convex cost functions, namely the optimality of converging sequences in terms of their associated mean costs.

**Theorem 3.3**
Let \( \{x_k\}_{k=1}^{\infty} \) be a bounded state sequence and let the strictly convex function \( f \) be the cost associated with a state \( x_k \). Furthermore, let the mean state and the mean cost be defined as

\[
x_m(\infty) = \lim_{M \to \infty} x_m(M) = \lim_{M \to \infty} \frac{1}{M} \sum_{k=1}^{M} x_k
\]

\[
f_m(\infty) = \lim_{M \to \infty} f_m(M) = \lim_{M \to \infty} \frac{1}{M} \sum_{k=1}^{M} f(x_k)
\]

Then
\[ f(x_m(\infty)) \leq f_m(\infty) \]

and, if and only if the state sequence converges
\[ \lim_{k \to \infty} x_k = x_m(\infty) \]

then
\[ f(x_m(\infty)) = f_m(\infty) \]

**Proof.** The non-increasing upper bounds on the states and their associated costs at time \( k \), are defined as

\[
\overline{x}(k) = \max\{x_i \mid i \in [k; \infty]\}
\]

\[
\overline{f}(k) = \max\{f(x_i) \mid i \in [k; \infty]\}
\]

and the non-decreasing lower bounds on the states and their associated cost at time \( k \), are defined as

\[
\underline{x}(k) = \min\{x_i \mid i \in [k; \infty]\}
\]

\[
\underline{f}(k) = \min\{f(x_i) \mid i \in [k; \infty]\}
\]
Proposition 3.3 gives that
\[ f(\lim_{M \to \infty} x_m(M)) = \lim_{M \to \infty} f(x_m(M)) \leq f_m(M) \leq f_m(\infty) \]

Given their definitions and lemma 3.1, it then follows for the upper and lower cost bounds that
\[ f(k) \leq f(\lim_{M \to \infty} x_m(M)) \leq f_m(\infty) \leq f(k), \text{ for } k \in [1; \infty[ \]

If the state sequence converges, then given lemma 3.2,
\[ \lim_{k \to \infty} x(k) = \lim_{k \to \infty} x_m(M) = \lim_{k \to \infty} \pi(k) \]

and consequently
\[ \lim_{k \to \infty} f(k) = f(\lim_{M \to \infty} x_m(M)) = f_m(\infty) = \lim_{k \to \infty} \tilde{f}(k) \]

If the state sequence does not converge, then
\[ \lim_{k \to \infty} x(k) < \lim_{k \to \infty} x_m(M) < \lim_{k \to \infty} \pi(k) \]

and proposition 3.3 gives that
\[ f(\lim_{M \to \infty} x_m(M)) < f_m(\infty) \]

That is, theorem 3.3 states that the mean cost associated with a non-converging sequence is greater than or equal to, the mean cost associated with a converging sequence. Then closed-loop convergence for systems governed by the optimal ILC control law (3.31) is stated in theorem 3.4.

Theorem 3.4
Let the controllable system (3.5) (i.e., \( G \) has full row rank) be governed by the optimal ILC control law (3.31), then given assumption 3.3, the closed-loop system
\[ e_k = e_{k-1} + G\Delta u_k^* - \tilde{F}\tilde{v}_k \]

will converge to the fixpoint \( x_\infty \), in the sense that
\[ \lim_{k \to \infty} \left[ \|E_k\| Q \| \Delta u_k^* \| R \right] = x_\infty \]
**Proof.** Consider the state sequence

\[ x_k = \begin{bmatrix} \|E\{e_k^* I_k\}\|_Q \\ \|\Delta u_k^*\|_R \end{bmatrix} \]

with the associated strictly convex cost function

\[ f(x_k) = x_k^T x_k = \|E\{e_k^* I_k\}\|_Q^2 + \|\Delta u_k^*\|_R^2 \]

According to theorem 3.3, non-convergence of the state sequence \( x_k \) would contradict the optimality of the control sequence \( \{\Delta u_k^*\}^\infty_{k=0} \) and it hence follows that the state sequence converges

\[ \lim_{k \to \infty} x_k = \lim_{k \to \infty} \begin{bmatrix} \|E\{e_k^* I_k\}\|_Q \\ \|\Delta u_k^*\|_R \end{bmatrix} = x_\infty \]

Theorem 3.4 thus states that given assumption 3.3 the optimal ILC control law (3.31) guarantees convergence for the controllable system (3.5). In order to identify the fixpoint for the controllable system (3.5) a few more assumptions are however necessary. Thus, assume that the output reference profile is feasible

\[ y_{\min} \leq \bar{y} \leq y_{\max} \quad (3.32) \]

and that the associated input profile \( \bar{u} \) is admissible

\[ u_{\min} \leq \bar{u} \leq u_{\max} \quad (3.33) \]

That is, assume that the output reference profile is reachable.

**Assumption 3.4**

The output reference profile is reachable. For a controllable system this means that the output reference profile is feasible

\[ y_{\min} \leq \bar{y} \leq y_{\max} \]

and the associated input profile \( \bar{u} \) is admissible

\[ u_{\min} \leq \bar{u} \leq u_{\max} \]

Then the solution \( \{\Delta u_k^*\}^\infty_{k=0} \) to the infinite horizon OCP (3.31), can be proven to satisfy the relaxed stochastic ILC design requirement (3.29).
3.1. Iterative Learning Control

Proposition 3.4
Let the controllable system (3.5) (i.e., $G$ has full row rank) be governed by the optimal ILC control law (3.31), then given assumptions 3.3 and 3.4, the closed-loop system

$$e_k = e_{k-1} + G\Delta u_k^* - \tilde{F}\tilde{v}_k$$

will satisfy the relaxed stochastic ILC design requirement (3.29).

Proof. Consider the state sequence

$$x_k = \left[ \begin{array}{c} \|E\{e_k | I_k\}\|_Q \\ \|\Delta u_k^*\|_R \end{array} \right]$$

with the associated strictly convex cost function

$$f(x_k) = x_k^tx_k = \|E\{e_k | I_k\}\|_Q^2 + \|\Delta u_k^*\|_R^2$$

According to theorem 3.4 it then follows that the state sequence converges

$$\lim_{k \to \infty} x_k = \lim_{k \to \infty} \left[ \begin{array}{c} x_1^k \\ x_2^k \end{array} \right] = x_\infty$$

Given controllability and assumptions 3.3 and 3.4

$$x_1^\infty = 0$$

is a feasible convergence point. According to theorem 3.4, any other convergence point/mean $\tilde{x}_\infty$ would contradict optimality of the control sequence $\{\Delta u_k^*\}_k=0$, i.e., it would be associated with a greater mean cost

$$f(\tilde{x}_\infty) > f([0^t \ x_2^\infty]^t)$$

as $\|\Delta u_k^*\|_R$ is proportional to $\|E\{e_{k-1} | I_k\}\|_Q$ unless all input constraints are active, and thus $\tilde{x}_2^\infty \geq x_2^\infty \iff \tilde{x}_1^\infty > 0$. Consequently it must follow that

$$\lim_{k \to \infty} \|E\{e_k | I_k\}\|_Q = 0$$

and hence as $Q$ is positive definite, that

$$\lim_{k \to \infty} E\{e_k | I_k\} = 0$$
This means that the infinite horizon OCP (3.31) guarantees asymptotically perfect reproducibility given the off-line information from past and future batches. However, solving the infinite horizon OCP (3.31) is obviously infeasible given its infinite dimension and the causal unavailability of information from future batches. Hence, it is attempted in the following to approximate the infinite horizon OCP (3.31) with a dual-mode finite horizon OCP based on information from past batches.

Traditionally, finite horizon dual-mode formulations of infinite horizon OCPs are based on the assumption that there exists a horizon $M$ beyond which, none of the input-output constraints are active (see e.g. Gilbert and Tin Tan, 1991).

**Assumption 3.5**
The exists a horizon $M$ and an admissible control sequence $\{\Delta u_k\}^\infty_{k=1}$ for the closed-loop system

$$e_k = e_{k-1} + G\Delta u_k - \tilde{F}\tilde{v}_k$$

such that

$$h(I_k) < 0$$

for $k \geq M$ and any bounded initial condition $e_0$.

Given assumption [3.5] the infinite horizon OCP (3.31) can be approximated with the following finite horizon OCP

$$\{\Delta u_{k,l}\}^{k+M-1}_{l=k} = \arg \min_{\{\Delta u_{i}\}^M_{i=1}} \begin{bmatrix} \hat{e}'_M P \hat{e}_M + \sum_{i=1}^{M-1} \hat{e}'_i Q \hat{e}_i + \sum_{j=1}^{M} \Delta u'_j R \Delta u_j \end{bmatrix}$$

s.t. $\hat{e}_i = \hat{e}_{i-1} + G\Delta u_i$

$$h(\tilde{I}_j) \leq 0$$

$$\hat{e}_0 = E\{e_{k-1} | \tilde{I}_0\}$$

$$\tilde{I}_j = \{\Delta u_j, \tilde{I}_{j-1}\}, \tilde{I}_0 = \{I_{k-1}, z_{k-1}\}$$

(3.34)

where it has been further assumed that $G$ has full row-rank. The terminal penalty matrix $P$ is the solution to the stationary Discrete-time Algebraic Riccati Equation (DARE)

$$P = P - PG(PGP + R)^{-1}G'P + Q$$

(3.35)

As known from the Linear Quadratic Regulator (LQR) and Linear Quadratic Gaussian (LQG) Regulator literature (see e.g. Bertsekas, 2000), the penalty
term with the positive definite symmetric weighting matrix $\mathbf{P}$, gives the optimal cost of the following unconstrained infinite horizon OCP

$$\hat{e}_M' \mathbf{P} \hat{e}_M = \min_{\{\Delta \mathbf{u}_l\}_{l=M+1}^{\infty}} \left[ \sum_{l=M}^{\infty} E\{e_l | \hat{I}_l\}' Q E\{e_l | \hat{I}_l\} + \Delta \mathbf{u}_{l+1}' R \Delta \mathbf{u}_{l+1} \right]$$

s.t.

$$\begin{aligned}
\hat{e}_{l+1} &= e_l + G \Delta \mathbf{u}_{l+1} - \hat{F} \hat{v}_{l+1} \\
\hat{I}_l &= \{\Delta \mathbf{u}_l, \hat{I}_{l-1}\}, \quad \hat{I}_0 = \{I_{k-1}, z_{k-1}\}
\end{aligned}$$

(3.36)

Note, the fact that the terminal penalty term is bounded for any bounded $\hat{e}_M$, means that given assumptions 3.4, the sequence $\{(E\{e_l | \hat{I}_l\}, \Delta \mathbf{u}_l)\}_{l=M}^{\infty}$ in (3.36) converges to its origin $(0, 0)$. This means that under assumptions 3.4 and 3.5, the difference between the infinite horizon OCP (3.31) and the finite horizon OCP (3.34) (for $k = 0$) is the information on which the expected error profiles are conditioned. That is, as the infinite horizon OCP (3.31) includes all future information it is said to be closed-loop optimal, while the finite horizon OCP (3.34), which only includes past information, is said to be open-loop optimal (Bertsekas, 2000). Furthermore, if the finite horizon OCP (3.34) is re-computed as $k$ propagates and more information becomes available, then the sub-optimal sequence $\{\Delta \mathbf{u}_{k,k}\}_{k=0}^{\infty}$ given by (3.34), will approximate the closed-loop optimal sequence $\{\Delta \mathbf{u}_k^*\}_{k=0}^{\infty}$ given by (3.31).

### 3.2 Learning Model Predictive Control

The general idea of learning Model Predictive Control (MPC) is to combine the asymptotically error-free performance of ILC with the approximately closed-loop performance of MPC (see e.g. Mayne et al., 2000; Cannon and Kouvaritakis, 2002). The learning MPC formulation of ILC often referred to as internal-model-based ILC, and its convergence properties are discussed for deterministic LTI systems by Hillenbrand and Pandit (1999); de Roover et al. (2000) amongst others. Although Lee et al. (2000) propose learning MPC of stochastic LTV systems, they only discuss convergence properties for deterministic systems. This section considers learning MPC of batch processes that are described by the stochastic Linear Time-Varying (LTV) system

$$\begin{aligned}
x_{k,t} &= A_t x_{k,t-1} + B_t \Delta \mathbf{u}_{k,t-1} + E_t v_{k,t} \\
e_{k,t} &= \hat{y}_t - y_{k,t} \\
&= e_{k-1,t} - C x_{k,t}
\end{aligned}$$

(3.37)

for $t = 1, \ldots, N$ with the initial condition

$$x_{k,0} = C' v_{k,0}$$

(3.38)

That is, the state $x_{k,t} \in \mathbb{R}^{n_x}$ of the system (3.37) at time $t$ in batch $k$, is given by linear mappings of the system history $x_{k,t-1}$, the control correction
\[ \Delta u_{k,t-1} = u_{k,t-1} - u_{k-1,t-1} \in \mathbb{R}^{n_x(t-1)} \] and a zero-mean Gaussian disturbance \( v_{k,t} \in \mathbb{R}^{n_v} \). The tracking error \( e_{k,t} \in \mathbb{R}^{n_y} \) of the system (3.37) at time \( t \) in batch \( k \), is given as the difference between the system output \( y_{k,t} \in \mathbb{R}^{n_y} \) and a desired output reference \( \bar{y}_t \in \mathbb{R}^{n_y} \). The LTV description of the system (3.37) and (3.38) is equivalent to the LTI system description (3.5) as the finite impulse response matrix \( G \) is given as

\[
G = \begin{bmatrix}
\tilde{g}_{1,1}B_1 & \tilde{g}_{2,1}B_2 & \cdots & \tilde{g}_{N,1}B_N \\
\tilde{g}_{2,1}B_1 & \tilde{g}_{2,2}B_2 & \cdots & \tilde{g}_{N,2}B_N \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{g}_{N,1}B_1 & \tilde{g}_{N,2}B_2 & \cdots & \tilde{g}_{N,N}B_N 
\end{bmatrix} \quad (3.39)
\]

and the finite impulse response matrix \( F \) is given as

\[
F = \begin{bmatrix}
\tilde{g}_{1,1}E_1 & \tilde{g}_{2,1}E_2 & \cdots & \tilde{g}_{N,1}E_N \\
\tilde{g}_{2,1}E_1 & \tilde{g}_{2,2}E_2 & \cdots & \tilde{g}_{N,2}E_N \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{g}_{N,1}E_1 & \tilde{g}_{N,2}E_2 & \cdots & \tilde{g}_{N,N}E_N 
\end{bmatrix} \quad (3.40)
\]

and the matrix \( H \) is given as

\[
H = \begin{bmatrix}
(\tilde{g}_{1,1}C)’ & (\tilde{g}_{2,1}C)’ & \cdots & (\tilde{g}_{N,1}C)’
\end{bmatrix}' \quad (3.41)
\]

where the matrices \( \tilde{g} \) are given as

\[
\tilde{g}_{i,j} = C \prod_{m=j+1}^{i} A_m \quad (3.42)
\]

The fact that the LTV description of the system (3.37) is equivalent to the LTI system description (3.3) means that output-controllability, i.e., the ability to move the system to a given output profile with a finite sequence of input profile corrections, of the LTV system (3.37) is determined by the row-rank of the finite impulse response matrix \( G \).

**Definition 3.2 (Finite Horizon LTV Controllability)**

The system (3.37) is said to be output-controllable if and only if the finite impulse response matrix

\[
G = \begin{bmatrix}
Cg_{1,1} & Cg_{2,1} & Cg_{2,2} \\
\vdots & \vdots & \ddots \\
Cg_{N,1} & Cg_{N,2} & \cdots & Cg_{N,N}
\end{bmatrix}
\]

\[3\]The matrix product is defined as \( \prod_{j=x}^{m} A_j = A_m \cdot A_{m-1} \cdot \ldots \cdot A_{n+1} A_n \).
3.2. Learning Model Predictive Control

has full row-rank where the impulse response coefficient matrices are given as

\[ g_{i,j} = \prod_{m=j+1}^{i} A_m B_j \]

Furthermore, the system (3.37) is said to be controllable if and only if the finite impulse response matrix

\[ G_x = \begin{bmatrix} g_{1,1} & g_{2,1} & g_{2,2} \\ \vdots & \vdots & \ddots \\ g_{N,1} & g_{N,2} & \cdots & g_{N,N} \end{bmatrix} \]

has full row-rank.

Assume that during a batch \((k)\), observations \(z_{k,t}\) are collected at times \(t = 0, 1, \ldots, N\) and let the optimal estimate of the state \(x_{k,t}\) in batch \(k\) at time \(t_1\) given data up to and including time \(t_2\) be given as the conditional mean

\[ \hat{x}_{k,t_1|t_2} = E\{x_{k,t_1} | I_{k,t_2}\} \] (3.43)

where the information \(I_{k,t}\) is given as

\[ I_{k,t} = \{z_{k,t}, \Delta u_{k,t}, I_{k,t-1}\} \] (3.44)

with initial conditions

\[ I_{k,-1} = I_{k-1,N}, \quad I_{0,-1} = \{y_{-1}, u_{-1}, z_{-1}\} \] (3.45)

Then the tracking error \(e_{k,t_1}\) in batch \(k\) at time \(t_1\) given data up to and including time \(t_2\) is estimated as

\[ \hat{e}_{k,t_1|t_2} = \hat{e}_{k-1,t_1|N} - C \hat{x}_{k,t_1|t_2} \] (3.46)

where the smoothened estimate of the error profile in batch \(k - 1\) is given as

\[ \hat{e}_{k-1,k-1} = \begin{bmatrix} \hat{e}_{k-1,1|N} & \hat{e}_{k-1,2|N} & \cdots & \hat{e}_{k-1,N|N} \end{bmatrix}' \] (3.47)

and \(\hat{y}_{k-1,k-1}\) is the smoothened output profile estimate from batch \(k - 1\) (see page 97).

Because of the equivalence between the LTI and LTV descriptions of the system, the ILC OCP formulation

\[
\{\Delta u_{k,t_1-1}\}_{t_1=0}^{N-1} = \arg \min_{\Delta u_k} [E\{e_k'Q_k e_k | I_k\} + \Delta u_k' R_k \Delta u_k] \\
\text{s.t. } e_k = e_{k-1} + G \Delta u_k - \tilde{F} \hat{v}_k \quad h(I_{k,-1}) \leq 0
\] (3.48)
is equivalent to the following learning MPC OCP formulation

\[
\{\Delta u_{k,t-1}\}_{t=0}^{N-1} = \arg \min_{\{\Delta u_{k,i}\}_{i=0}^{N-1}} \left[ \sum_{t=1}^{N} \hat{e}_{k,i}[t]^T Q_{k,i} \hat{e}_{k,i}[t] + \Delta u'_{k,t-1} R_{k,i} \Delta u_{k,i-1} \right]
\]

s.t. \[
\begin{align*}
\dot{x}_{k,i}[t] &= A_i \hat{x}_{k,i-1} + B_i \Delta u_{k,i-1} \\
\hat{e}_{k,i}[t] &= \hat{e}_{k,i}[N] - C \hat{x}_{k,i-1} \\
u_{\text{min}_i-1} &\leq \Delta u_{k,i-1} + u_{k-1,i-1} - u_{\text{max}_i-1} \\
y_{\text{min}_i} &\leq \hat{y}_i - \hat{e}_{k,i-1} \leq y_{\text{max}_i}
\end{align*}
\]

(3.49)

if the weighting matrices \(Q_k\) and \(R_k\) are block diagonal

\[
Q_k = \text{diag}(Q_{k,t}) \\
R_k = \text{diag}(R_{k,t})
\]

(3.50)

and the weighting matrices \(Q_{k,t}\) and \(R_{k,t}\) are all symmetric and positive definite. In proposition 3.5, the optimal solution \(\Delta u_{k,-1} = \{\Delta u_{k,i-1}\}_{i=0}^{N-1}\) to the learning MPC OCP \(3.49\), is proven to satisfy the stochastic ILC design requirement \(3.6\) and thus guarantees convergence, if the weighting matrix pairs \((Q_k, R_k), \forall k\), are designed such that

\[
\Psi_k \leq -2\hat{e}'_{k-1,k-1} Q_k G \Delta u_{k,-1} - \Delta u'_{k,-1} G' Q_k G \Delta u_{k,-1}
\]

(3.51)

Proposition 3.5

Let the system \(3.37\) be governed by the learning MPC OCP \(3.49\), then given assumption 3.3, the closed-loop system

\[
\begin{align*}
x_{k,t} &= A_t x_{k,t-1} + B_t \Delta u_{k,t-1} - C_t v_{k,t} \\
\hat{e}_{k,t} &= e_{k-1,t} - C x_{k,t}
\end{align*}
\]

for \(t = 1, \ldots, N\), will satisfy the ILC design requirement \(3.6\) if

\[
\Psi_k \leq -2\hat{e}'_{k-1,k-1} Q_k G \Delta u_{k,-1} - \Delta u'_{k,-1} G' Q_k G \Delta u_{k,-1}
\]

Proof. The proof is analogous to that of proposition 3.7 on page 58.

Remark 3.7

Note that even if the system \(3.37\) is (output-) controllable, a weighting matrix pair \((Q_k, R_k)\) for which \(3.51\) is satisfied need not exist.
Thus, proposition 3.5 specifies a sufficient condition, but not a necessary condition, for convergence of the closed-loop system. Furthermore, assuming that the reference output profile \( \bar{y} \) is reachable and that the system (3.37) is output-controllable, convergence analogous to that of proposition 3.2 is proven in proposition 3.6.

**Proposition 3.6**

Let the output-controllable system (3.37) be governed by the learning MPC OCP formulation (3.49), then given assumption 3.1, the closed-loop system

\[
x_{k,t} = A_t x_{k,t-1} + B_t \Delta u_{k,t-1} + E_t v_{k,t}
\]

\[
e_{k,t} = e_{k-1,t} - C x_{k,t}
\]

for \( t = 1, \ldots, N \), will converge to its origin, in the sense that

\[
\lim_{k \to \infty} E\{e_k | I_{k-1}\} = 0
\]

if

\[
\Psi \leq -2 \hat{e}_{k-1,k-1} Q G \Delta u_{k,-1} - \Delta u_{k,-1} G' Q G \Delta u_{k,-1}
\]

**Proof.** Follows from propositions 3.1, 3.2 and 3.5, given output-controllability and assumption 3.1. \(\blacksquare\)

The learning MPC OCP (3.49) does not however, include the information gathered during the batch operation and hence, will not give improved closed-loop performance as desired. Thus, to take full advantage of the information gathered during the batch operation an open-loop optimal learning MPC OCP

\[
\{\Delta u_{k,t}\}_{t=1}^{N-1} = \arg \min_{\{\Delta u_{k,i}\}_{i=1}^{N-1}} \left[ \sum_{i=t+1}^{N} \hat{e}_{k,i,t} Q_i \hat{e}_{k,i,t} + \Delta u_{k,i-1} R_i \Delta u_{k,i-1} \right]
\]

\[
s.t. \quad \hat{x}_{k,i,t} = A_i \hat{x}_{k,i-1} + B_i \Delta u_{k,i-1}
\]

\[
\hat{e}_{k,i,t} = \hat{e}_{k-1,i,N} - C \hat{x}_{k,i,t}
\]

\[
u_{\text{min} - 1} \leq \Delta u_{k,i-1} + u_{k-1,i-1} \leq u_{\text{max} - 1}
\]

\[
y_{\text{min}} \leq y_{k} - \hat{e}_{k,i,t} \leq y_{\text{max}}
\]

is resolved at times \( t = 0, 1, \ldots, N - 1 \) in batch \( k \) and the control sequence

\[
\Delta u_{k}^* = [ \Delta u_{k,0,0}^* \Delta u_{k,1,1}^* \ldots \Delta u_{k,N-1,N-1}^* ]'
\]

will then approximate a closed-loop optimal control sequence. In order to compare the achievable performance of the ILC OCP formulation (3.48) to that of the learning MPC OCP formulation (3.52) the following intermediate
sequences are defined. Define the intermediate control sequence $\Delta u_{k,t}$ at time $t$ in batch $k$ as

$$
\Delta u_{k,t} = \begin{bmatrix}
\Delta u_{k,0,0} \\
\Delta u_{k,1,1} \\
\vdots \\
\Delta u_{k,t,t} \\
\Delta u_{k,t+1,t} \\
\vdots \\
\Delta u_{k,N-1,t}
\end{bmatrix}, \quad t = -1, 0, \ldots, N - 1 \tag{3.54}
$$

with $\Delta u_{k,N-1} = \Delta u_k^*$. Note that, if no information about the initial condition of a batch is available (which is the implicit assumption behind eq. (3.38)), then

$$
\Delta u_{k,-1} = \{\Delta u_{k,l,-1}\}_{l=0}^{N-1} = \{\Delta u_{k,l,0}\}_{l=0}^{N-1} = \Delta u_k \tag{3.55}
$$

Then given information up to and including time $t$, the state of the system (3.37) at time $i \in [t; N]$ in batch $k$ is predicted as

$$
\hat{x}_{k,i|t}(\Delta u_{k,j}) = \prod_{j=t+1}^{i} A_j \hat{x}_{k,t|t}(\Delta u_{k,j}) + \sum_{l=t+1}^{i} \prod_{m=l+1}^{i} A_m B_l \Delta u_{k,l-1,j} \tag{3.56}
$$

for $t = 0, \ldots, N - 1$, and

$$
\hat{x}_{k,i|t}(\Delta u_{k,j}) = \sum_{l=1}^{i} \prod_{m=l+1}^{i} A_m B_l \Delta u_{k,l-1,j} \tag{3.57}
$$

for $t = -1$, and $0 \leq j < N$. The corresponding tracking error is hence predicted as

$$
\hat{e}_{k,i|t}(\Delta u_{k,j}) = \hat{e}_{k-1,i|t,N} - C \hat{x}_{k,i|t}(\Delta u_{k,j}) \tag{3.58}
$$

for $t = -1, 0, \ldots, N - 1$ and $-1 \leq j < N$. Next, define the intermediate error profile $\hat{e}_{k,t}$ as

$$
\hat{e}_{k,t}(\Delta u_{k,j}) = \begin{bmatrix}
\hat{e}_{k,1|t}(\Delta u_{k,j}) \\
\hat{e}_{k,2|t}(\Delta u_{k,j}) \\
\vdots \\
\hat{e}_{k,t|t}(\Delta u_{k,j}) \\
\hat{e}_{k,t+1|t}(\Delta u_{k,j}) \\
\vdots \\
\hat{e}_{k,N|t}(\Delta u_{k,j})
\end{bmatrix} \tag{3.59}
$$
for \( t = -1, 0, \ldots, N \) and \(-1 \leq j < N\). Then the cost function \( J \) can be defined as

\[
J_k(\hat{e}_{k,t}(\Delta u_{k,j}), \Delta u_{k,j}) = \|\hat{e}_{k,t}(\Delta u_{k,j})\|^2_{Q_k} + \|\Delta u_{k,j}\|^2_{R_k}
\]

(3.60)
such that the achieved ILC cost \( J_{aILC} \) is given in batch \( k \) as

\[
J_{aILC}^k = J_k(\hat{e}_{k,N}(\Delta u_{k,-1}), \Delta u_{k,-1})
\]

(3.61)
and the achieved learning MPC cost \( J_{aMPC}^k \) is given in batch \( k \) as

\[
J_{aMPC}^k = J_k(\hat{e}_{k,N}(\Delta u_*)^k), \Delta u_*^k)
\]

(3.62)
Furthermore, under the assumption that the state and tracking error estimation errors are zero-mean, it is proven in theorem 3.5 that learning MPC will outperform ILC, that is

\[
J_{aMPC}^k \leq J_{aILC}^k
\]

(3.63)

Assumption 3.6

The state estimation errors \( \tilde{x}_{k,t} = x_{k,t} - \hat{x}_{k,t|t} \) and the tracking error estimation errors \( \tilde{e}_{k,t} = e_{k,t} - \hat{e}_{k,t|t} \) are zero-mean, i.e.,

\[
E\{x_{k,t} | \hat{x}_{k,t|t}\} = \hat{x}_{k,t|t}
\]

\[
E\{e_{k,t} | \hat{e}_{k,t|t}\} = \hat{e}_{k,t|t}
\]

Note that assumption 3.6 can be relaxed into (monotonically decreasing) bounds on the norm of the estimation errors [Scokaert et al. 1997].

Theorem 3.5

Let the system (3.37) be governed by the learning MPC OCP formulation (3.52), then given assumption 3.6, the closed-loop system

\[
x_{k,t} = A_t x_{k,t-1} + B_t \Delta u_{k,t-1,t-1} + E_t v_{k,t}
\]

\[
e_{k,t} = e_{k-1,t} - C_t x_{k,t}
\]

for \( t = 1, \ldots, N \), will satisfy

\[
J_{aMPC}^k \leq J_{aILC}^k
\]

Proof. Let \( \tilde{\Psi}_{k,t} \) for \( t = -1, 0, \ldots, N \), be defined such that the achieved ILC cost \( J_{aILC}^k \) and the achieved learning MPC cost \( J_{aMPC}^k \) are given as

\[
J_{aILC}^k = J_k(\hat{e}_{k,t}(\Delta u_{k,-1}), \Delta u_{k,-1}) + \tilde{\Psi}_{k,t}
\]

\[
J_{aMPC}^k = J_k(\hat{e}_{k,t}(\Delta u_*^k), \Delta u_*^k) + \tilde{\Psi}_{k,t}
\]
Given assumption 3.9 it then follows that

\[ J_k(\dot{e}_{k,t}, \Delta u_{k,j}) + \tilde{\Psi}_{k,t} = J_k(\dot{e}_{k,t+1}(\Delta u_{k,j}), \Delta u_{k,j}) + \tilde{\Psi}_{k,t+1} \]

and from optimality of the optimal control profiles given by the learning MPC OCP (3.49), it follows that

\[ J_k(\dot{e}_{k,t}(\Delta u_{k,t}), \Delta u_{k,t}) \leq J_k(\dot{e}_{k,t}(\Delta u_{k,t-1}), \Delta u_{k,t-1}) \]

and hence that

\[ J_k^{\text{MPC}} = J_k(\dot{e}_{k,N}(\Delta u_{k,N-1}), \Delta u_{k,N-1}) \\
= J_k(\dot{e}_{k,N-1}(\Delta u_{k,N-2}), \Delta u_{k,N-2}) + \tilde{\Psi}_{k,N-1} \\
\leq J_k(\dot{e}_{k,N-2}(\Delta u_{k,N-2}), \Delta u_{k,N-2}) + \tilde{\Psi}_{k,N-2} \\
\vdots \\
\leq J_k(\dot{e}_{k,-1}(\Delta u_{k,-1}), \Delta u_{k,-1}) + \tilde{\Psi}_{k,-1} \\
\leq J_k^{\text{ILC}} \]

That is, utilizing the information gathered during a batch it is firstly, guaranteed that learning MPC achieves better reproducibility than ILC, and secondly, possible to achieve reproducibility with learning MPC even when the ILC design requirement (3.6) is not satisfied. Thus, when the (optimizing) ILC OCP (3.19) guarantees convergence, so does the (optimizing) learning MPC OCP (3.52).

**Example 3.4**
In the following example, the properties of the above proposed learning MPC methodology are demonstrated through control of the following unconstrained LTV system

\[ x_{k,t} = A_t x_{k,t-1} + B_t \Delta u_{k,t-1,t-1} + E_t v_{k,t} \]
\[ e_{k,t} = e_{k-1,t} - C x_{k,t} \]
\[ y_{k,t} = y_{k-1,t} + C x_{k,t} \]

for \( t = 1, \ldots, 100 \), with unknown initial conditions \( x_{k,0} = C^t y_{k,0} \), and where the control moves \( \Delta u_{k,t,t} \) are given by the learning MPC OCP (3.52) and the historic profiles \( e_k \) for \(-10,000 \leq k \leq 0\), are simulated with \( \Delta u_{k-1} = 0 \). The
system matrices are given as

\[ A_t = \begin{bmatrix} ((t/50) - 1)^2 + 0.5 & t/100 + 0.5 \\ -0.3 & 1.2/\sqrt{t} + 0.6 \end{bmatrix} \]

\[ B_t = \begin{bmatrix} 0.05 \sin(t/50) + 0.075 & 0 \\ 0 & 0.05 \cos(t/50) + 0.075 \end{bmatrix} \]

\[ E_t = C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

The disturbance profiles \( \tilde{v}_k \) are drawn from a zero-mean Gaussian distribution with covariance \( \tilde{\Sigma} = 0.001I \) and it is furthermore assumed that the output profile estimate \( E\{y_{k-1} \mid I_k\} \) is given as

\[ E\{y_{k-1} \mid I_k\} = y_{k-1} + \hat{e}_{k-1} \]

where the estimation error profiles \( \hat{e}_k \) are drawn from a zero-mean Gaussian distribution with covariance \( S = 0.0001I \). The controller is designed using the true system and the weighting matrices \( Q_k \) and \( R_k \) are designed as

\[ Q_k = R_k = I \]

The controller performance in terms of the cost \( \|e_k\|_Q^2 \) and the convergence indicator \( \rho_k \), is plotted on the right hand side in figure 3.5 for 2000 batches. The controller performance that would have been achieved with ILC is plotted on the left hand side in figure 3.5 for comparison. The convergence indicator \( \rho_k \) is clearly less than one on average and in concurrence herewith, the cost \( \|e_k\|_Q^2 \) decreases to a steady state level. The learning MPC OCP formulation (3.52) clearly outperforms the ILC OCP formulation (3.13) in terms of reproducibility. As this system is driven by random disturbances there is not much for the controller to learn, and the learning rate or cost reduction is thus rather slow. The batch process reproducibility is however greatly improved with the application of the proposed learning MPC OCP formulation.

3.3 Iterative Operation Optimization

In the previous sections convergence of ILC and learning MPC to a desired batch operations model was proven. In the present section it is of interest to investigate the possibilities for optimizing the batch operations model. Explicitly, this section considers Iterative Operations Model Optimization (IOMO) of batch processes that are described by the stochastic LTI system

\[ y_k = y_{k-1} + H(y_{k-1} - y_{k-1,0}) - G(u_k - u_{k-1}) + Fv_k \]

\[ = y_{k-1} + \begin{bmatrix} H & -G \end{bmatrix} \begin{bmatrix} \Delta y_{k,0}^t & \Delta u_{k}^t \end{bmatrix} + Fv_k \]

\[ = y_{k-1} + G\Delta u_{k} + Fv_k \] (3.64)
Figure 3.5: This figure shows the weighted tracking error cost $\|e_k\|^2_{Q_k}$ and the convergence indicator $\rho_k(\Delta u_k^*)$ achieved on the system in example 3.4, with the ILC OCP formulation (3.13) (left hand side figures) and the learning MPC OCP formulation (3.52) (right hand side figures). The learning MPC OCP formulation (3.52) clearly outperforms the ILC OCP formulation (3.13) in terms of reproducibility. The system runs in open-loop up to batch 1, hereafter the system is under closed-loop control. As seen from both the tracking error cost and the convergence indicator, the system converges to an improved steady state reproducibility level.

Let the $n_y$ outputs and $n_u$ inputs of the batch process, be sampled $N$ times during a batch. Then, the output profile $y_k \in \mathbb{R}^{Nn_y}$ in batch $k$ is given by a weighted sum of the output profile in the previous batch $y_{k-1}$, the effect of corrections made to the batch operations model $\Delta \tilde{u}_k \in \mathbb{R}^{Nn_u+n_y}$, and the effect of a stochastic zero-mean disturbance profile $v_k \in \mathbb{R}^{Nn_y}$. The output profile $y \in \mathbb{R}^{Nn_y}$ is observed with measurement noise

$$z_k = y_k + \tilde{\epsilon}_k$$

(3.65)

where $\tilde{\epsilon} \in \mathbb{R}^{Nn_y}$ is a zero-mean measurement noise profile.

Let the optimal operations model $\tilde{u}_{\text{opt}}$ be defined by the following con-
strained Linear Program (LP)

\[
\tilde{u}_{\text{opt}} = \arg \min_{\tilde{u}} \left[ qy + r\tilde{u} \right]
\]
\[\text{s.t. } \begin{align*}
    y &= \bar{y} + \tilde{G}(\tilde{u} - \tilde{\bar{u}}) \\
    \tilde{u}_{\text{min}} &\leq \tilde{u} \leq \tilde{u}_{\text{max}} \\
    y_{\text{min}} &\leq y \leq y_{\text{max}}
\end{align*} \tag{3.66}
\]

where the weighting matrices \( q \) and \( r \) have full column-rank and \((\bar{y}, \tilde{\bar{u}})\) is a pair of reference profiles. Obviously, the optimal operations model \( \tilde{u}_{\text{opt}} \) could be implemented directly, but what is sought in the present section is to move the batch process operation towards its optimum with a desired learning rate. By implementing the optimal operations model \( \tilde{u}_{\text{opt}} \) at a desired (/conservative) rate the user/control system has time to evaluate and in-/validate that the system is indeed being steered towards an improved operations model, when the system description \( [3.64] \) is uncertain, which it will indeed be in any chemical process application. A procedure for IOMO of uncertain batch systems is discussed further below. This problem definition gives rise to the following OCP formulation of IOMO

\[
\Delta \tilde{u}_k^* = \arg \min_{\Delta \tilde{u}_k} \left[ J_{k}^{\text{IOMO}} \right]
\]
\[\text{s.t. } \begin{align*}
    y_k &= y_{k-1} + \tilde{G}\Delta \tilde{u}_k + Fv_k \\
    \Delta y_k &= y_k - y_{k-1} \\
    \tilde{u}_k &= \Delta \tilde{u}_k + \bar{u}_{k-1} \\
    \tilde{u}_{\text{min}} &\leq \tilde{u}_k \leq \tilde{u}_{\text{max}} \\
    y_{\text{min}} &\leq E\{y_k \mid I_k\} \leq y_{\text{max}}
\end{align*} \tag{3.67}
\]

where \( J_{k}^{\text{IOMO}} = E[\Delta y_k \mid I_k]^{\prime} \tilde{Q}_k E[\Delta y_k \mid I_k] + \Delta \tilde{u}_k^{\prime} \tilde{R}_k \Delta \tilde{u}_k \\
+ qE\{y_k \mid I_k\} + r\tilde{u}_k \tag{3.68}
\]

and the symmetric weighting matrices \( \tilde{Q}_k \) and \( \tilde{R}_k \), which are positive semi-definite and positive definite, respectively, determines the learning rate. It is furthermore assumed that the initial input profile \( u_0 \) is admissible, i.e., that the constraints are satisfied before the optimization is initiated. In proposition 3.7 it is proven for systems without uncertainty, that the IOMO control law \( [3.67] \) guarantees asymptotic convergence to the optimal operations model given by \( [3.66] \).

**Proposition 3.7**

Let the controllable system (i.e., \( \tilde{G} \) has full row-rank) \( [3.64] \) be governed by the IOMO control law \( [3.67] \), then the closed-loop system

\[
y_k = y_{k-1} + \tilde{G}\Delta \tilde{u}_k^* + Fv_k
\]
will converge to its optimal operations model given by (3.66), in the sense that

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \tilde{u}_k = \tilde{u}_{opt}$$

**Proof.** For the system

$$y_k(\Delta \tilde{u}_k) = y_{k-1} + \tilde{G} \Delta \tilde{u}_k + F v_k$$

define the expected batch cost function \(J_k\) as

$$J_k(\Delta \tilde{u}_k) = \|E\{\Delta y_k(\Delta \tilde{u}_k) \mid \mathcal{I}_k\}\|^2_{\tilde{Q}_k} + \|\Delta \tilde{u}_k\|^2_{\tilde{R}_k} + qE\{y_k(\Delta \tilde{u}_k) \mid \mathcal{I}_k\} + r \tilde{u}_k(\Delta \tilde{u}_k)$$

and it follows that

$$J_k(\Delta \tilde{u}_k) \leq qE\{y_k(\Delta \tilde{u}_k) \mid \mathcal{I}_k\} + r \tilde{u}_k(\Delta \tilde{u}_k) \leq J_k(0) \leq \frac{1}{N} \sum_{k=1}^{N} \tilde{u}_k = \tilde{u}_{opt}$$

It then must follow that

$$qE\{y_k(\Delta \tilde{u}_k) \mid \mathcal{I}_k\} + r \tilde{u}_k(\Delta \tilde{u}_k) \leq J_k(\Delta \tilde{u}_k) \leq qE\{y_{k-1} \mid \mathcal{I}_k\} + r \tilde{u}_{k-1}$$

which, recalling that

$$E\{y_k \mid \mathcal{I}_k\} = E\{y_{k-1} + \tilde{G} \Delta \tilde{u}_k + F v_k \mid \mathcal{I}_k\} = E\{y_{k-1} \mid \mathcal{I}_k\} + \tilde{G} \Delta \tilde{u}_k$$

and

$$\tilde{u}_k(\Delta \tilde{u}_k) = \Delta \tilde{u}_k + \tilde{u}_{k-1}$$
3.3. Iterative Operation Optimization

is equivalent to

\[ q \hat{G} \Delta \hat{u}_k^* + r \Delta \hat{u}_k^* \leq 0 \]

Consequently, as the system is controllable it follows that

\[ \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} J_k = J^* \]

where \( J^* \) is the optimal cost of (3.66), and thus that

\[ \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \hat{u}_k = \hat{u}_{\text{opt}} \]

as \( q \) and \( r \) have full column-rank.

Disturbance Attenuation

The IOMO OCP formulation (3.67) is a pure feedforward control formulation with no attenuation of the effect of the disturbance profile \( v \). This means that although according to proposition 3.7, the input profile sequence \( \{ \hat{u}_k \}_{k=0}^{\infty} \) will converge to the optimal input profile \( \hat{u}_{\text{opt}} \), there is no attenuation of the variability of the output profile sequence \( \{ y_k \}_{k=0}^{\infty} \). To minimize the variability of the output profile sequence \( \{ y_k \}_{k=0}^{\infty} \) it is thus proposed to combine the disturbance attenuation capabilities of the approximately closed-loop optimal learning MPC OCP formulation (3.52) with the IOMO OCP formulation (3.67). This combination of learning MPC and IOMO is formulated in procedure 3.1.

Procedure 3.1

1. Obtain an output profile estimate \( E\{ y_k \mid I_{k+1} \} \) by implementing an admissible input profile \( u_k \) and propagate the batch index \( k = k + 1 \).

2. Obtain an improving input correction profile \( \Delta \hat{u}_k^* \) from (3.67).

3. Update the reference profile

\[ \tilde{y} = E\{ y_{k-1} \mid I_k \} + \hat{G} \Delta \hat{u}_k^* \]

and the estimated tracking error profile

\[ \hat{e}_{k-1,k-1} = \hat{G} \Delta \hat{u}_k^* \]

from batch \( k - 1 \).

4. Implement the improved initial conditions \( y_{k,0} = \Delta y_{k,0} + y_{k-1,0} \).
5. Track the reference profile $\hat{y}$ with the learning MPC OCP formulation \[ (3.52) \]. That is, implement the input profile $u_k = \Delta u_k^* + u_{k-1}$. 

6. Propagate the batch index $k = k + 1$ and obtain an improving input correction profile $\Delta\hat{u}_k^*$ from \[ (3.67) \]. 

7. Check for convergence in the input correction profile, in the sense that 
   \[ \|\Delta\hat{u}_k^*\|_2 \leq \varsigma \] 
   where $\varsigma$ is a user defined scalar tolerance. If convergence go to step #5. Otherwise go to step #3.

Note, although procedure 3.1 will only bring the input profile sequence $\{\hat{u}_k\}_{k=0}^\infty$ close to the optimal input profile $\hat{u}_{\text{opt}}$, the decrease in the variability of the output profile sequence $\{y_k\}_{k=0}^\infty$ may lead to achieved costs \[ \tilde{J}_k^{\text{IOMO}} = qE\{y_k \mid I_{k+1}\} + r\hat{u}_k \] (3.69) 
that are lower than the achieved IOMO optimal costs \[ \tilde{J}_k^{\text{IOMO, opt}} = qE\{y_k \mid I_{k+1}\} + r\hat{u}_{\text{opt}} \] (3.70)
for $k$ greater than some positive scalar $\tilde{M}$. If procedure 3.1 has this property of producing achieved costs that are lower than the achieved IOMO optimal costs for $k$ greater than some positive scalar $\tilde{M}$, then the procedure 3.1 is said to be overtakingly optimal (Carlson et al., 1991) when compared to the IOMO OCP formulation \[ (3.67) \] implemented on its own.

Uncertain System Descriptions

Assume that the true system is unknown and that $\hat{\theta}_i$ is a LTI model approximation of the behavior of the true system, local to the reference profile pair $(\hat{y}_i, u_i)$ and with the following stochastic representation 
\[ y_k = y_{k-1} + \hat{H}(\hat{\theta}_i)(y_{k,0} - y_{k-1,0}) - \hat{G}(\hat{\theta}_i)(u_k - u_{k-1}) + \hat{F}(\hat{\theta}_i)v_k \] 
\[ = y_{k-1} + \hat{G}(\hat{\theta}_i)\Delta\hat{u}_k + \hat{F}(\hat{\theta}_i)v_k \] (3.71) 

Then applying the IOMO OCP formulation \[ (3.67) \], may move the system out of the region in which the local model $\hat{\theta}_i$ is valid. If the learning rate designed by manipulating the weighting matrices $\hat{Q}_k$ and $\hat{R}_k$, is low however, the system may stay within the region of validity of the local model $\hat{\theta}_i$. In this way, if based on the local model estimate $\hat{\theta}_i$, the IOMO OCP formulation \[ (3.67) \] produce an input correction profile $\Delta\hat{u}_k$ such that both the input profile $\hat{u}_k = \Delta\hat{u}_k + \hat{u}_k$ and the corresponding realized output profile $y_k$ are inside the region of validity.
of the local model estimate \( \hat{\theta}_i \), then the realized profile pair \((y_k, u_k)\) may form a new reference profile pair

\[
(y_{i+1}, u_{i+1}) = (y_k, u_k)
\]  

(3.72)

Locally to this new reference pair \((\bar{y}_{i+1}, \bar{u}_{i+1})\), a new model estimate could then be obtained and the IOMO OCP formulation (3.67) could be reapplied. This Ad-Hoc procedure is stated in procedure 3.2.

**Procedure 3.2**

1. Given a local model \( \hat{\theta}_i \), apply procedure 3.1 while monitoring the model validity. The model validity is monitored by computing the model quality \( G^*_\rho \) given by eq. (2.76) on page 41, for each completed batch. When/If the model quality decreases (i.e., \( G^*_\rho \) increases) to a level at which the model is no longer considered valid by the user, procedure 3.1 is stopped.

2. Update the reference pair \((\bar{y}_i, \bar{u}_i)\) to the realized input-output profiles from the last batch where the model estimate was deemed valid by the user. Conduct designed experiments to generate data local to the new reference pair \((y_{i+1}, u_{i+1})\). Propagate the model index \( i = i + 1 \).

3. Identify a model \( \hat{\theta}_i \) from data local to the reference pair \((\bar{y}_i, \bar{u}_i)\). Based on the model quality \( G^*_\rho \), the local model estimate \( \hat{\theta}_i \) is judged to be valid or not by the user. If the model estimate \( \hat{\theta}_i \) is judged to be valid, then proceed to step #1. Otherwise, stop.

Note however, that the region of validity of a local linear model estimate \( \hat{\theta}_i \) of a nonlinear system, will most likely decrease as the modeled system approaches its optimal operational state. The postulated decreased region of validity of a local linear model estimate close to the optimum of a nonlinear system, is based on an assumption of optima being defined by nonlinearities/bifurcations (typically in the form of system and/or product quality constraints). This means that procedure 3.2 will most likely fail to bring a nonlinear system to its true optimum. Procedure 3.2 should thus be monitored and stopped when sufficient improvement has been achieved. Sufficient improvement can often be achieved with just a few (\( \leq 5 \)) iterations (Srinivasan and Bonvin 2004), but the improvement (learning) rate will obviously depend on the ratio between the norms of the weighting matrices \( \tilde{Q}_k \) and \( \tilde{R}_k \).

### 3.4 State and Output Estimation

Consider a batch system given by the following periodically LTV, observer canonical state space model in \( \mathbb{R}^{n_x} \) (see equations (2.10) and (2.16) in chapter 2)

\[
x_{k,t} = A_t x_{k,t-1} + B_t \Delta u_{k,t-1} + E_t v_{k,t}
\]

\[
z_{k,t} = C_t x_{k,t} + y_{k-1,t} + \epsilon_{k,t}
\]  

(3.73)
with noisy observations $z_{k,t} \in \mathbb{R}^{n_x}$ at times $t = 0, 1, \ldots, N$. Furthermore, let the initial conditions of the system (3.73) be given as (see equations (2.10), (2.14), and (2.16) in chapter 2)

$$
\begin{align*}
    x_{k,0} &= C' v_{k,0} \\
    z_{k,0} &= C x_{k,0} + y_{k-1,0} + \epsilon_{k,0}
\end{align*}
$$

(3.74)

Assume that the stochastic sequences $v_{k,t}$ and $\epsilon_{k,t}$ are zero-mean, independent and drawn from normal distributions with covariances $\Sigma_t$ and $R_{\epsilon_t}$, respectively. That is

$$
\begin{bmatrix}
    v_{k,i} \\
    \epsilon_{k,i} \\
    1
\end{bmatrix}
\begin{bmatrix}
    v_{l,j} \\
    \epsilon_{l,j} \\
    1
\end{bmatrix} = \begin{bmatrix}
    \Sigma_t \delta_{i,j,k,l} \\
    S_t \delta_{i,j-1,k,l} \\
    R_t \delta_{i,j,k,l}
\end{bmatrix} \delta_{i,j,k,l} - 1
$$

(3.75)

where $\delta_{i,j,k,l}$ is a (modified) Kronecker delta function, i.e., for $i = j$ and $k = l$ simultaneously, $\delta_{i,j,k,l} = 1$ and for $i \neq j$ or $k \neq l$, $\delta_{i,j,k,l} = 0$. The cross-covariance $S_t = E\{v_{k,t} \epsilon_{k,t}^t \}$ may be nonzero for systems under feedback control. Assume that the covariance matrices $\Sigma_i$ and $R_{\epsilon_i}$ for $i = 0, 1, \ldots, N$, are positive definite. Furthermore, assume that the measurement noise on the individual outputs is uncorrelated. That is, the covariance matrices $R_{\epsilon_i}$ are diagonal. Note, for continuous discrete-time systems, the formulations of the assumptions (3.75) will usually include independence of the initial state $x_{k,0}$, but this independence is implicitly given for batch systems as they are defined in the present methodology — see eq.’s (3.74) and (3.82). The LTV system (3.73) is assumed observable, i.e., the initial condition can be estimated from a given output profile.

**Definition 3.3 (Finite Horizon LTV Observability)**

The system (3.73) is said to be observable if and only if the observability matrix $O = \begin{bmatrix} O_1 & O_2 \end{bmatrix}$ has full column-rank, when

$$
O_1 = \begin{bmatrix}
    CP_1 & CP_2 & \cdots & CP_{n-1} \\
    C g_{2,1} P_2 & \cdots & C g_{2,n-1} P_{n-1} \\
    \vdots & \vdots & \vdots \\
    C g_{n-1,n-1} P_{n-1}
\end{bmatrix}
$$

The assumption of uncorrelated measurement noise on the individual outputs is necessary for the below proposed output estimation procedure, while the below proposed state estimation procedure is valid for general (/dense) covariance matrices $R_{\epsilon_i}$ for $i = 0, 1, \ldots, N$. 

---

---

---
and

\[
O_2 = \begin{bmatrix}
C \\
C g_{2,n_o} \\
C g_{3,n_o} & C g_{3,n_o+1} & \ddots \\
\vdots & \vdots & \ddots & C \\
C g_{n_o,n_o} & C g_{n_o,n_o+1} & \cdots & C g_{N-n_o+1,N-n_o-1} & C \\
C g_{n_o+1,n_o+1} & \cdots & C g_{N-n_o+2,N-n_o-1} & C g_{N-n_o+2,N-n_o} & \cdots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \ddots \\
C g_{N-1,N-n_o-1} & C g_{N-1,N-n_o} & \cdots & C g_{N,N-n_o} & \cdots & \cdots & C g_{N,N-n_o}
\end{bmatrix}
\]

where \( n_o = \frac{n_x n_y}{n_o} \) is the input/output model order, the coefficient matrices \( g_{i,j} \) are given as

\[
g_{i,j} = \prod_{t=j+1}^{i} A_t
\]

and the permutation matrices \( P_t \) are designed such that \( B_t = A_t P_t \) is the first \( n_y i \) columns of \( A \).

Because of the correlated noise processes \( v \) and \( \epsilon \), in (3.73), a Gram-Schmidt procedure is used to derive the following system with uncorrelated noise processes, for which the state estimation problem is equivalent to that of (3.73)

\[
x_{k,t} = A_s^t x_{k,t-1} + B_t \Delta u_{k,t-1} + \epsilon_t v_{k,t} + \epsilon_t S_t R_{\epsilon_t}^{-1} \epsilon_{k,t}
\]

\[
z_{k,t} = C x_{k,t} + y_{k-1,t} + \epsilon_{k,t}
\]

where

\[
A_s^t = A_t - \epsilon_t S_t R_{\epsilon_t}^{-1} C
\]

\[
v_{k,t} = v_{k,t} - S_t R_{\epsilon_t}^{-1} \epsilon_{k,t-1}
\]

For the assumed observable LTV systems (3.73) and (3.76), the optimal state estimate at time \( t \) in batch \( k \), given information up to and including time \( t \), namely the conditional mean

\[
\hat{x}_{k,t|t} = E\{x_{k,t} | I_{k,t}\}
\]

is then given by the Kalman filter recursions

\[
\hat{x}_{k,t|t} = \hat{x}_{k,t|t-1} + K_{k,t}(z_{k,t} - C \hat{x}_{k,t|t-1} - \hat{y}_{k-1,t|N})
\]

\[
\hat{x}_{k,t|t-1} = A_s^t \hat{x}_{k,t-1|t-1} + B_t \Delta u_{k,t-1}
\]

\[
+ \epsilon_t S_t R_{\epsilon_t}^{-1} (z_{k,t-1} - \hat{y}_{k-1,t-1|N})
\]

(3.79)
for \( t = 0, 1, \ldots, N \), with the initial condition

\[
\hat{x}_{k,0|0} = 0
\]  

(3.80)

In most textbooks the corresponding Kalman filter gain matrix \( K_{k,t} \) and state estimate covariance matrix \( P_{k,t|t} \) are propagated by the following recursions

\[
P_{k,t|t} = P_{k,t|t-1} \left( I - C'K_{k,t} \right)
\]

\[
K_{k,t} = P_{k,t|t-1}C' \left( CP_{k,t|t-1}C' + R_{\epsilon,t} + P_{k-1,t|N} \right)^{-1}
\]

(3.81)

\[
P_{k,t|t-1} = A_t^s P_{k,t-1|t-1} A_t^{s'} + \mathcal{E}_t \Sigma_t^s \mathcal{E}_t'
\]

with the initial condition

\[
P_{k,0|0} = C' \Sigma_0 C
\]  

(3.82)

where

\[
\Sigma_t^s = \Sigma_t + S_t \left( R_{\epsilon,t}^{-1} P_{k-1,t|N} R_{\epsilon,t}^{-1} - R_{\epsilon,t}^{-1} \right) S_t'
\]  

(3.83)

If during a batch, the observation \( z_{k,t} \) is missing at time \( t \), the covariance matrix \( R_{\epsilon,t} \) is set to infinity — in practice this means \( \| R_{\epsilon,t} \|_2 \gg 0 \). Furthermore, in situations where there are no observations of the initial conditions, which is indeed the most common situation for chemical batch processes, then obviously

\[
\hat{x}_{k,0|0} = 0
\]

\[
P_{k,0|0} = C' \Sigma_0 C
\]  

(3.84)

One major difficulty when implementing the “textbook Kalman filter” (equations (3.81) and (3.82)) is round-off errors arising from finite-precision requirements. One consequence of round-off errors is that the computed covariance matrix may be non-Hermitian, i.e., not equal to its Hermitian transpose. Non-Hermitian covariance matrices can be compensated for by averaging computed covariance matrices with their Hermitian transpose, or preferably by propagating only half of the covariance elements, e.g. only the lower block-triangle. Since covariance matrices must be positive semi-definite a more serious consequence of round-off errors is loss of definiteness. Since the diagonal entries of covariance matrices are the mean-square errors, computation will be seriously off if they are not nonnegative. Although the Kalman filter can recover from loss of definiteness (Kailath et al., 2000), it is preferable to eliminate the numerical operations that may cause indefiniteness, i.e., eliminate matrix inversions and subtractions. One formulation of the Kalman filter recursions that is both numerically stable and efficient, is the array algorithms (Kailath et al., 2000), which in short, are orthogonal triangularizations analogous to the UD algorithm (see e.g. Thornton and Bierman, 1978; Grewal and Andrews, 1993; Kailath et al., 2000), originally developed for deterministic least squares problems (see pages 29 – 31). The main advantages of the array algorithms are that
they operate on lower triangular Cholesky (or square-root) factors (denoted by $P^\dagger$, such that $P = P^\dagger P^\dagger$) of the covariance matrices and that these factors are computed with an efficient (“Q-less”) QR factorization\(^5\). In doing so, the updated covariance matrices are less likely to lose their definiteness and will always be Hermitian and have nonnegative diagonal elements. That is, let $Q_{k,t}$ and $R_{k,t}$ be the QR factors of $\Theta_{k,t}$

$$\Theta_{k,t} = \begin{bmatrix} R_{t}^\dagger & CA_{t} P_{k,t-1|t-1}^{\dagger} \Sigma_{t}^{\dagger} & \Sigma_{t}^{\dagger} & P_{k,t-1|t-1}^{\dagger} \end{bmatrix}^\dagger$$

(3.85)

where $A_{t}$ and $\Sigma_{t}$ are efficiently computed (off-line) as

$$\begin{align*}
A_{t} &= A_{t} - E_{t} S_{t} R_{t}^{-1} R_{t}^{-1} C \\
\Sigma_{t} &= \Sigma_{t} + S_{t} R_{t}^{-1} \left( R_{t}^{-1} P_{k,t-1|t}^{\dagger} - I \right) R_{t}^{-1} S_{t}^{\dagger}
\end{align*}$$

(3.86)

then, recalling that $R_{k,t}$ is the Cholesky factor of $\Theta_{k,t}^{\dagger} \Theta_{k,t}$, $R_{k,t}$ is found to be

$$R_{k,t} = \begin{bmatrix} R_{k,t}^{\dagger} & 0 & P_{k,t}^{\dagger} \end{bmatrix}$$

(3.87)

from which the propagated state estimate covariance matrix $P_{k,t|t}$ is obtained directly and the Kalman filter gain matrix $K_{k,t}$ is obtained as

$$K_{k,t} = K_{k,t} \left( R_{k,t}^{\dagger} \right)^{-1}$$

(3.88)

where the inverse of the lower triangular innovation covariance matrix $R_{t}^{\dagger}$ is efficiently computed with forward-substitution. For completeness $\Theta_{k,t}$ is given for the initial condition

$$\Theta_{k,0} = \begin{bmatrix} R_{0} & 0 & \Sigma_{0}^{\dagger} & P_{k-1,N}^{\dagger} \end{bmatrix}^\dagger$$

(3.89)

$$= Q_{k,0} \begin{bmatrix} R_{k,0}^{\dagger} \end{bmatrix}$$

The output profile estimate $\hat{y}_{k}$ with covariance $P_{k,N|t}$, is a smooth estimate of the output profile $y_{k}$ in batch $k$ given information up to and including the end ($t = N$) of batch $k$. That is,

$$\hat{y}_{k,t|N} = E \{ y_{k,t} | \tilde{z}_{k} \}$$

(3.90)

\(^5\)When properly implemented, array algorithms are not computationally more expensive than the usual Kalman filter recursions (see e.g. Bierman and Thornton [1977].
where \( \tilde{z}_k \) is the observed profile from batch \( k \). Note, \( \tilde{z}_k \) may contain fewer or more likely, more observations than \( z_k \). In the present methodology, \( \hat{y}_{k,t|N} \) will be estimated with a Kernel Smoother (see e.g. Hastie et al., 2001), while the covariance matrix \( \bar{P}_{k,t|N} \) is estimated as

\[
\bar{P}_{k,t|N} = \tilde{W}_{k,t} R_{\epsilon_t} \tag{3.91}
\]

where \( \tilde{W}_{k,t} \) is a diagonal weighting matrix with diagonal elements

\[
0 < \tilde{W}_{k,t}(i,i) \leq 1 \tag{3.92}
\]

for \( i = 1, \ldots, n_y \). Assume that for batch \( k \), the output variables \( y(i) \in \mathbb{R} \) for \( i = 1, \ldots, n_y \), have been observed at times \( \tilde{T}(k,\tilde{t},i) \), giving \( N(k,i) + 1 \) observations \( \tilde{z}_{k,\tilde{t},i}(i) \) of variable \( y_k(i) \). The smoothened output estimates \( \hat{y}_{k,t|N}(i) \) are then given at times \( T(t), t \in \{0,1,\ldots,N\} \) in batch \( k \), by the following local polynomial regression model of order \( d(i) \)

\[
\hat{y}_{k,t|N}(i) = \hat{\alpha}(k,t,i) + \sum_{j=1}^{d(i)} \hat{\beta}_j(k,t,i) T(t)^j \tag{3.93}
\]

where the local parameters \( \hat{\alpha}(k,t,i) \in \mathbb{R} \) and \( \hat{\beta}_j(k,t,i) \in \mathbb{R} \) are the arguments that minimize

\[
\min_{\alpha(k,t,i),\{\beta_j(k,t,i)\}_{j=1}^{d(i)}} \left[ \sum_{t=0}^{\tilde{N}(k,i)} \tilde{K}_{ij}^i(T(t),\tilde{T}(k,\tilde{t},i)) \left( \tilde{z}_{k,\tilde{t},i}(i) - \hat{y}_{k,t|N}(i) \right)^2 \right] \tag{3.94}
\]

s.t. \( \hat{y}_{k,t|N}(i) = \alpha(k,t,i) + \sum_{j=1}^{d(i)} \beta_j(k,t,i) \tilde{T}(k,\tilde{t},i)^j \)

Let the observed profile of output variable \( y_k(i) \) in batch \( k \) be defined as

\[
\tilde{z}_k(i) = [\tilde{z}_{k,0}(i), \tilde{z}_{k,1}(i), \ldots, \tilde{z}_{k,N(k,i)}(i)]'
\]

then the estimation problem (3.93)–(3.94) can be given explicitly as

\[
\hat{y}_{k,t|N}(i) = t_{k,t} \begin{pmatrix} \tilde{T}_{k,\tilde{t}_i}^i \tilde{K}_{\gamma(i),k,t}^i \tilde{T}_{k,\tilde{t}_i}^i \end{pmatrix}^{-1} \begin{pmatrix} \tilde{T}_{k,\tilde{t}_i}^i \tilde{K}_{\gamma(i),k,t}^i \end{pmatrix} \tilde{z}_k(i) = s_{k,t,i} \tilde{z}_k(i) \tag{3.96}
\]
where

$$
\tilde{K}_{\gamma(i),k,t} = \begin{bmatrix}
\tilde{K}_{\gamma(i)}(T(t), \tilde{T}(k,0,i)) \\
\vdots \\
\tilde{K}_{\gamma(i)}(T(t), \tilde{T}(k,N(i),i))
\end{bmatrix}
$$

$$
t_{i,t} = \begin{bmatrix}
1 & T(t)^1 & \ldots & T(t)^{d(i)}
\end{bmatrix}
$$

$$
\hat{t}_{k,i,i} = \begin{bmatrix}
1 & \tilde{T}(k,\hat{t},i)^1 & \ldots & \tilde{T}(k,\hat{t},i)^{d(i)}
\end{bmatrix}
$$

$$
\bar{T}_{k,i} = \begin{bmatrix}
\hat{t}_{k,i,0}' & \hat{t}_{k,i,1}' & \ldots & \hat{t}_{k,i,N}'(k,i)
\end{bmatrix}'
$$

(3.97)

Note, efficient implementation of explicit Least Squares solutions such as (3.96), are dealt with in detail beginning on page 29 in chapter 2. Furthermore, with this explicit solution (3.96), the weighting matrix elements \( \tilde{W}_{k,t}(i,i) \) for \( i = 1, \ldots, n_y \), are given as

$$
\tilde{W}_{k,t}(i,i) = \| s_{k,i,t} \|_2
$$

(3.98)

for \( t = 0, \ldots, N \). The smoothed profile of output \( y_{k,t}(i) \) in batch \( k \) is then given explicitly as

$$
\hat{y}_{k|k}(i) = \begin{bmatrix}
\hat{y}_{k,0}(i) & \hat{y}_{k,1}(i) & \ldots & \hat{y}_{k,N}(i)
\end{bmatrix}'
$$

$$
= \begin{bmatrix}
s_{k,i,0}' & s_{k,i,1}' & \ldots & s_{k,i,N}'(k,i)
\end{bmatrix}' \tilde{z}_{k}(i)
$$

(3.99)

The type (e.g. Epanechnikov, tri-cube, or Gaussian) and bandwidth \( \gamma(i) \) of the Kernels \( \tilde{K}_{\gamma(i)} \) are selected individually for the output variables. The Kernel bandwidth \( \gamma(i) \) and the local polynomial order \( d(i) \) determine the trade-off between the bias and variance of the estimates — i.e., the smoothness of the estimates. The Kernel type also influences the bias and variance of the estimates. The local polynomial order \( d(i) \) is particularly sensitive at the boundaries, but the Kernel bandwidth \( \gamma(i) \) is however the most sensitive tuning parameter in Kernel smoothing. A detailed discussion of tuning of the Kernel bandwidth \( \gamma(i) \) falls outside this thesis.

### 3.5 Control and Optimization Example

This example considers modeling, control and optimization of the production of a secreted protein in a fed-batch reactor. This nonlinear batch system which has been subject to several optimization studies (see e.g. Banga et al. 1998, where the system model may also be found), should not be considered a complete or even realistic, representation of a cultivation process, but rather as a simple illustrative example of a nonlinear batch system. Cultivation of the secreted
protein is not a LTV process, but it will be modeled as such with the modeling methodology proposed in chapter 2 and based on this LTV model subjected to some the above proposed control algorithms.

During operation the cultivation reactor is continuously fed substrate at feed rate $u_{k,t}$ (L/h) in batch $k$ from time $t$ to time $t + 1$, and the process variables secreted protein ($y_{k,t}(1)$), total protein ($y_{k,t}(2)$), glucose ($y_{k,t}(3)$) and substrate ($y_{k,t}(4)$) concentrations (g/L), and the reactor volume ($y_{k,t}(5)$) (L) are sampled at discrete time instances. The duration of a batch is fixed to 15 hours during which the process variables ($y_{k,t}$) and the feed rate ($u_{k,t}$) are equidistantly sampled 101 times; i.e., the sample time is 9 minutes and the batch duration is $N = 100$ samples. The observations $z_{k,t}$ of the process variables are subject to normally distributed measurement noise $\epsilon_{k,t} \in \mathcal{N}(0, 0.01I)$, the covariance of which is assumed known. The initial conditions $y_{k,0}$ are stochastic and given as:

$$ y_{k,0} = \begin{bmatrix} 0.009 \\ 0.01 \\ 1 \\ 5 \\ 1 \end{bmatrix} + \begin{bmatrix} 0.009 \\ 0.009 \\ 0 \\ 2.5 \\ 9/16 \end{bmatrix} p_k $$

where $p_k$ is a vector of random variables drawn from a uniform distribution between -1 and 1. Observations of the initial conditions are not available for control implementations, but are collected in experimental batch runs. Operation of the cultivation reactor is constrained by downstream purification capacity and turn-down ratio and the feed pump capacity. I.e., the final reactor volume is constrained to $10 \leq y_{k,N}(5) \leq 20$ L and the feed rate is constrained to $0 \leq u_{k,t} \leq 2$ L/h. The nominal feed rate profile $\bar{u}$ is given as a staircase ramp

$$ \bar{u}_t = \frac{t}{N} + 0.5 $$

for $t = 0, 1, \ldots, N-1$. The evolution of 25 batches fed according to the nominal feed rate profile are plotted in figure 3.6. Due to the stochastic initial conditions significant variability in the final secreted protein concentration is experienced in these batch runs. In the first part of this control example excessive variability in the final product concentration is considered intractable as it complicates downstream purification. The first control objective of this control example will thus be to reduce the variability in the final secreted protein concentration by manipulating the substrate feed rate.

**Remark 3.8**

In a real fed-batch production scenario the control objective would most likely be to reduce variability in the final product mass and not the final product concentration. This more realistic control objective is however not feasible for the simplified process in the present example, as any variability removed from
Figure 3.6: This figure shows the evolution of 25 batch runs on a cultivation reactor producing secreted protein. The cultivation reactor is operated in open-loop and according to the nominal batch operations model. Due to stochastic initial conditions, an undesired variability in the final secreted protein concentration is experienced. One control objective will thus be to reduce the variability in the final secreted protein concentration.

the secreted protein concentration by manipulation of the feed rate, is introduced as variability in the reactor volume, and the product mass variability consequently remains unchanged. In a real fed-batch production scenario, the substrate feed rate would however not be the sole manipulated variable. Firstly, other process variables such as temperature, pressure, and pH, could be manipulated in order to contribute to the achievement of the control objective. Secondly, the substrate feed would hopefully be divided into a glucose feed and one or more feeds introducing other nutrients (some of these are not available for use in the overall control problem as they are used to control e.g. pH), as this would add more degrees of freedom to the control problem. Finally, it is recommended that fed-batch cultivation reactors are (re-) designed with a pure water feed as manipulation of this feed stream would make it possible to manipulate the other feed streams with a minimum of variability being introduced into the reactor volume.

In the second part of this control example, the control objective will be to maximize the final secreted protein mass by manipulating the substrate feed
rate. In common for the two control objectives is the following constrained OCP formulation

$$\{ \Delta u_{k,i,t} \}_{i=t}^{N-1} = \underset{\{ \Delta u_{k,i} \}_{i=t}^{N-1}}{\arg\min} \left[ J_{k,i,t} \right]$$

$$\text{s.t. } J_{k,i,t} = \sum_{i=t+1}^{N} \left( u_{k,i-1} - u_{k,i-2} \right)^{T} \hat{T}_{i} \left( u_{k,i-1} - u_{k,i-2} \right)$$

$$+ \hat{e}_{k,i,t}^{T} \hat{Q}_{i} \hat{e}_{k,i,t} + \Delta u_{k,i-1}^{T} \hat{R}_{i} \Delta u_{k,i-1}$$

$$\hat{x}_{k,i,t} = A_{i} \hat{x}_{k,i-1} + B_{i} \Delta u_{k,i-1}$$

$$\hat{e}_{k,i,t} = \hat{e}_{k-1,i|N} - C_{i} \hat{x}_{k,i,t}$$

$$u_{k,i-1} = \Delta u_{k,i-1} + u_{k-1,i-1}$$

$$0 \leq u_{k,i-1} \leq 2$$

$$y_{min} \leq \hat{y}_{i} - \hat{e}_{k,i|t} \leq y_{max}$$

with $u_{k,-1} = u_{k-1,0}$ and the output constraints given as

$$y_{min} = \left\{ \begin{array}{ll} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{T} & \text{for } i = 1, \ldots, N-1 \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 10 \end{bmatrix}^{T} & \text{for } i = N \end{array} \right. \quad (3.102)$$

$$y_{max} = \begin{bmatrix} 100 & 100 & 100 & 100 & 20 \end{bmatrix}^{T} \text{ for } i = 1, \ldots, N$$

The scaled weighting matrices $\hat{T}_{i}$, $\hat{Q}_{i}$, and $\hat{R}_{i}$ are given as

$$\hat{T}_{i} = W_{input}^{T} T_{i} W_{input}$$

$$\hat{Q}_{i} = W_{input}^{T} Q_{i} W_{input}$$

$$\hat{R}_{i} = W_{input}^{T} R_{i} W_{input}$$

(3.103)

where $W$ and $W_{input}$ are scaling/weighting matrices (see below). For ILC OCP formulations the open-loop optimal input profile $\Delta u_{k,0}$ is implemented, while the closed-loop optimal approximation $\Delta u_{k,N-1}$ is implemented for lMPC OCP formulations (see eq. (3.54)).

Model Identification

Once the control objective has been defined the next step in any model-based control implementation is to obtain a model. In this control example the batch process will be modeled following the modeling methodology proposed in chapter 2. To do so, data from 40 experimental batch runs was collected and prepared according to the methodology stated in section 2.2.1. To ensure sufficient information in the experimental data set, a Pseudo Random Binary Sequence (PRBS) was superimposed onto the nominal feed rate profile. The cost of running these designed experiments is a 3% drop in the mean productivity — see table 3.2. The noise-free(!) data from these experimental batch runs is shown in
Figure 3.7: This figure shows the outcome of 50 designed experiments run on a cultivation reactor. The process excitation is generated by a Pseudo Random Binary Sequence (PRBS) dither signal. The cost in terms of lost productivity, associated with these experiments is a modest 3% drop in the mean final secreted protein mass.

According to the methodologies proposed in sections 2.2.2 and 2.2.3, the model parameters in a set of AutoRegressive models with eXogenous inputs (ARX), were fitted to data from 20 batches and the regularization weights and model orders (the maximum allowed model order was 4) were identified through cross-validation on data from 10 independent batches. The estimation problem was weighted with the weighting matrix $W$ given by (2.36) (as is $W_{input}$). As part of the model cross-validation, the one-step-ahead estimation error covariance matrices $\Sigma_t$, for $t = 0, \ldots, N$, were also estimated. The (near) optimal model orders are given in table 3.1. The optimized cultivation reactor model was tested on 10 independent batches and achieved an estimated test generalization error of $G_0^{\star} = 0.0268$, with the weighting matrix $W$ given as (2.56). Given the ARX representation of the cultivation reactor, a 20 state LTV model was obtained for regulator and state estimator design.

State And Output Estimation

The states of the identified LTV model were estimated with a Kalman filter as specified in section 3.4 under the assumption that the process disturbances are
independent of the measurement noise, i.e., the cross-covariance matrices are given as $S_t = 0$, for $t = 0, \ldots, N$. The output profile estimation was performed with a tri-cube Kernel Smoother based on 35 nearest neighbors.

**Reproducibility Increasing Control**

To achieve the first control objective, namely to reduce the variability in the final secreted protein concentration by manipulating the substrate feed rate, a nominal reference output profile, which is assumed known, was tracked with ILC and learning MPC algorithms. Given the control objective the following weighting matrices were designed

$$Q_t = \begin{bmatrix} t^2/N & t^2/N \\ 0.1 & 0.01 \end{bmatrix}$$

$$R_t = 0.1$$

$$T_t = 10$$

for $t = 1, \ldots, N$. This controller design firstly uses the reactor volume as a variance sink, secondly uses the glucose and substrate concentration profiles to steer the process in the right direction in the initial production phase, and thirdly places significant emphasis on reducing the variability in the final protein concentrations. As the dominant source of variance (the controller itself will also introduce variance due to model-plant mismatch and erroneous state estimates) in the secreted protein concentration is the stochastic initial conditions, it is not expected that the ILC algorithm will improve reproducibility as there are no significant persistent disturbances. It is however expected that the learning MPC algorithm will improve reproducibility as it utilizes process information as it becomes available during a batch run. The closed-loop performance of the ILC and learning MPC algorithms is shown in figures 3.8 and 3.9.

<table>
<thead>
<tr>
<th>Model Orders</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
<th>$j = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_A(1, j, t)$, $t = 1, \ldots, N$</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$n_A(2, j, t)$, $t = 1, \ldots, N$</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$n_A(3, j, t)$, $t = 1, \ldots, N$</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>$n_A(4, j, t)$, $t = 1, \ldots, N$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$n_A(5, j, t)$, $t = 1, \ldots, N$</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$n_B(j, 1, t)$, $t = 1, \ldots, N$</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3.1: The 30 active model orders of the cultivation reactor model. The Moving Average model orders were fixed to $n_C = 0$. The near optimal model orders were found by minimizing the estimated generalization error. The maximum allowed model order was 4.
3.5. Control and Optimization Example

Figure 3.8: Closed-loop performance of an ILC algorithm implemented on a nonlinear batch process with stochastic initial conditions. The control objective is improved reproducibility. The ILC algorithm does not improve the reproducibility of this particular batch process, just as should be expected when it is implemented on a batch process without persistent disturbances.

3.9 for the same 25 batch runs as shown in figure 3.6. Just as expected the ILC algorithm does not improve the reproducibility of the cultivation reactor, while the learning MPC algorithm improves the reproducibility significantly. It is concluded that the learning MPC algorithm achieved the first control objective.

Optimizing Control

The process data generated during the abovementioned 50 identification experiments contains more information than that summarized in the identified model. In fact, these identification experiments can be considered as a stochastic optimization experiment and the experimental batch with the highest productivity produced 15.1 g secreted protein (Banga et al., 1998, have reported a theoretically optimal productivity of 32.6 g). Although it is infeasible to reproduce this batch as the initial conditions are stochastic, an attempt to reproduce it could increase productivity. The nominal output reference profile is therefore replaced by the output profile of this optimal batch run. To achieve the second control objective, namely to maximize the final secreted protein mass by
manipulating the substrate feed rate, the new unreachable(!) reference output profile, was thus tracked with ILC and learning MPC algorithms. Given the control objective the following weighting matrices were designed

$$Q_t = \begin{bmatrix}
0.05 & -0.0005 & -0.00125 & 0 & 0.2125 \\
-0.0005 & 0.05 & 0 & 0 & 0.2125 \\
-0.00125 & 0 & 0.0125 & 0 & 0.02 \\
0 & 0 & 0 & 0.05 & -0.075 \\
0.2125 & 0.2125 & 0.02 & -0.075 & 2
\end{bmatrix} \quad (3.105)$$

$$R_t = 0.5$$

$$T_t = 1$$

for \( t = 1, \ldots, N \). This controller design is significantly different from the controller design for the first control objective, as it has the main emphasis on rejecting variability in the reactor volume, while it encourages productivity improving variability in the remaining output profiles. This means that e.g. elevated protein concentrations are encouraged as long as they are not achieved.
Figure 3.10: Closed-loop performance of an ILC algorithm implemented on a nonlinear batch process with stochastic initial conditions. The control objective is to maximize the final secreted protein mass. For this particular batch process, the ILC algorithm increases the mean protein productivity by 11%.

at the expense of decreased reactor volume and vice versa. With this reactor design high secreted protein productivity is thus favored. As the ILC algorithm will continuously try to move the batch process closer to the infeasible reference profile, it is expected that the ILC algorithm improves the mean secreted protein productivity marginally as it is unlikely to cause significant drops in the secreted protein productivity. It is however expected that the learning MPC algorithm will significantly improve the secreted protein productivity as it utilizes process information as it becomes available during a batch run. The closed-loop performance of the ILC and learning MPC algorithms is shown in figures 3.10 and 3.11 for the same 25 batch runs as shown in figure 3.6. As expected the learning MPC algorithm increased the protein productivity significantly with 12%, but unexpectedly the ILC algorithm performed equally good with an 11% increase in the protein productivity. It is concluded that both the ILC and the learning MPC algorithms achieved the second control objective.

The in-/decreases in the protein productivity before, during and after the different controller implementations are summarized in table 3.2, from which it is apparent that pursuing minimum variance in the final secreted protein concentration will cause a drop in productivity of the cultivation reactor and that further studies are necessary in order to determine if simplified downstream
Figure 3.11: Closed-loop performance of a learning MPC algorithm implemented on a nonlinear batch process with stochastic initial conditions. The control objective is to maximize the final secreted protein mass. For this particular batch process, the learning MPC algorithm increases the mean protein productivity by 12%.

purification justifies this drop. With respect to feasibility of the implementation of optimizing controllers on the cultivation reactor, then the increased productivity obtained in closed-loop clearly justifies the cost associated with the identification experiments.

3.6 Control and Optimization Conclusions

The present chapter presented constrained linear control and optimization methodologies for optimal and reproducible operation of batch processes. The control and optimization methodologies comprise Optimal Control Problem (OCP) formulations of disturbance attenuating control for reproducible and batch operations model optimization for optimal batch process operation and a formulation of output and state estimation for batch processes.

The control methodologies for reproducible batch operation comprise the constrained OCP formulations Iterative Learning Control (ILC) for Linear Time-Invariant (LTI) batch representations and learning Model Predictive Control (MPC) for Linear Time-Varying (LTV) batch representations and a neces-
sary and sufficient design condition for guaranteed closed-loop stability of these constrained OCP formulations. For LTI batch representations, the ILC OCP formulation has been proven to guarantee stability if and only if the above necessary and sufficient design condition is satisfied. For LTV batch representations, the learning MPC OCP formulation has been proven to outperform the ILC OCP formulation in terms of minimum tracking error variability, and the learning MPC OCP formulation has thus been proven to guarantee stability if the necessary and sufficient closed-loop stability condition is satisfied. When the necessary and sufficient closed-loop stability condition is satisfied, stability is guaranteed for the ILC and the learning MPC OCP formulations even if the desired batch operations model is unreachable. This means that unreachable batch operations models may be deliberately implemented in an attempt to achieve dynamically optimized batch operation instead of reproducible batch operation. The necessary and sufficient closed-loop stability condition is satisfied if on average the expected reproducibility in terms of minimum tracking error variability, in the coming batch operation is at least as good as the reproducibility achieved in the prior batch operation. The necessary and sufficient closed-loop stability condition can thus be used to verify whether or not closed-loop batch operation will be stabilizing before the batch operation is commenced. And if it can not be verified, the necessary and sufficient closed-loop stability condition can be used to guide alternative controller tuning — i.e., redesign of the weighting matrices in the optimal control problem formulations. In addition to the necessary and sufficient closed-loop stability condition, controllability of finite horizon LTV systems has been defined. As a final element in the control methodologies for reproducible batch operation, it has been shown that ILC with an optimal learning rate can be formulated as an infinite horizon MPC OCP and convergence of this infinite horizon MPC OCP formulation has been proven. As implementation of an infinite horizon MPC OCP is infeasible it has furthermore been shown how the infinite horizon MPC OCP formulation can be approximated with a finite horizon MPC OCP formulation.

The control methodology for batch operation optimization comprise a definition of optimal batch operation as a constrained Linear Program (LP) and an Iterative Operations Model Optimization (IOMO) OCP formulation to track

<table>
<thead>
<tr>
<th>Operation / Algorithm</th>
<th>Mean Product Mass</th>
<th>Relative Productivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal / Non</td>
<td>10.4 g</td>
<td>100 %</td>
</tr>
<tr>
<td>Identification / PRBS</td>
<td>10.1 g</td>
<td>97 %</td>
</tr>
<tr>
<td>Reproducible / ILC</td>
<td>10.0 g</td>
<td>96 %</td>
</tr>
<tr>
<td>Reproducible / MPC</td>
<td>10.0 g</td>
<td>96 %</td>
</tr>
<tr>
<td>Optimizing / ILC</td>
<td>11.6 g</td>
<td>111 %</td>
</tr>
<tr>
<td>Optimizing / MPC</td>
<td>11.7 g</td>
<td>112 %</td>
</tr>
</tbody>
</table>

Table 3.2: The protein productivity of a nonlinear cultivation reactor before, during and after the implementation of model-based, linear controllers.
optimal batch operation. The definition of optimal batch operation as a con-
strained LP may seem quite restrictive, but recall that a linear basis can be
augmented at will with nonlinear expansions (e.g. the product of two or more
system parameters), which means that constrained LP need not be overly re-
strictive. For LTI batch representations, the IOMO OCP formulation for batch
operation optimization has been proven to converge on average to the optimal
batch operation. For batch processes that can only be locally approximated by
a LTI representation, a procedure has been given, which suggests monitoring
the validity of a local representation and re-estimating it if/when it becomes
invalid. This procedure is unlikely to converge to the validity region of a lo-
cal representation, and should thus be monitored closely and stopped when/if
the batch process performance becomes consistently decreasing. As the IOMO
OCP formulation does not include disturbance attenuation, a procedure has
been given, which combines the disturbance attenuation of the control method-
ologies for reproducible batch operation with the batch operation optimization
abilities of the IOMO OCP formulation.

The output profile and state estimation has been formulated as a combina-
tion of a Kernel Smoother and a Kalman filter. That is, smooth estimates of
output profiles are obtained with a Kernel Smoother between batch operations
while state estimates are obtained with a Kalman filter during batch operation.
The combination of a Kernel Smoother and a Kalman Filter is suggested in
part to allow for multiple sample rates and in part to reduce the covariance of
the estimates. If the output profiles were not estimated separately as smoothen
estimates with a Kernel smoother, but instead estimated as filtered estimates
with a Kalman filter, the output error integrator build into the batch models
that have been proposed throughout this thesis, would inadvertently integrate
the state and hence output profile, covariance, which would obviously not be
tractable. Furthermore, batch processes are typically sampled much more fre-
cently than their model representations used to design Kalman filters (and
ILC or MPC controllers), and when the output profiles are estimated sepa-
rately with a Kernel Smoother, which is model-free, all available process data
can be utilized resulting in better estimates with lower covariances. Before
state estimation can be attempted the observability of a batch representation
must be determined. To serve this purpose, observability of finite horizon LTV
systems has been defined.

Unless otherwise is stated, the methodologies presented in the present chap-
ter are based on the assumptions that 1) batch processes can be represented
by LTI models describing the batch process evolution from batch to batch
and LTV models describing the batch process evolution both from batch to
batch and in time, and 2) that optimal solutions to optimal control problem
formulations are independent of the replacement of stochastic variables with
their conditional means (i.e., the certainty equivalence principle applies) and
that optimal control formulations can be separated into an optimal regulator
problem and an optimal state estimation problem, and that this separation is
optimal (i.e., the separation theorem for linear systems with quadratic costs
applies). These two general assumptions are most likely not valid in practical implementations, but they are necessary for proving guaranteed closed-loop stability (i.e., non-decreasing reproducibility or batch operation performance) which is the main contribution in the present chapter. Furthermore, awareness of the theoretical limitations aids diagnose of and helps avoid, failures in practical control implementations. If a batch process in closed-loop is stable, it is however, of less importance whether or not it can be accurately described by a linear model or if the certainty equivalence principle applies. Although the validity of the two general assumptions is necessary for the proofs presented in the present chapter, it is not necessary for successful practical batch process control implementations. Invalidity of the two general assumptions should however always be cause for caution as instability may well be the result of a batch process control implementation in such a case. That practical batch process control implementations can be successful despite invalidity of the two general assumptions is demonstrated in the control and optimization example given in section 3.5. In this example the two general assumptions are invalid; in spite hereof it is shown that both the reproducibility and the operation performance of the batch process can be improved with the application of the control and optimization methodologies presented in the present chapter.
Application Examples

To demonstrate the applicability of the proposed modeling and control methodologies, two batch processes have been investigated. These case studies were conducted early on in the collective work presented in the present thesis, the modeling and control methodologies as well as the nomenclature are therefore not necessarily identical to that of the preceding chapters. The first case study presented here is an industrial pilot plant fermentation, which was modeled with a set of Finite Impulse Response (FIR) models. The second case study is a demonstration of the application of both the modeling and control methodologies to a simulated yeast fermentation which is modeled with a set of FIR models extended with initial conditions.

4.1 Pilot Plant Fermentation

In collaboration with an industrial partner, data from pilot plant fermenters was investigated for the purpose of modeling. Such models could potentially be used for operator training, multivariable control design, and as was the case in this case study, as a screening tool in the ongoing search for the optimal (or at least improved) recipe. To optimize any batch process it is necessary to track or push against process constraints. To find these constraints, exploratory recipes are tested in pilot scale. One such pilot scale fermenter was modeled with a set of FIR models. For the sake of simplicity only the on-line measurements of carbon dioxide, oxygen, substrate feed rate, and the off-line measurements product activity were considered. The substrate feed rate was selected as the input or manipulated variable and the rest, i.e., the carbon dioxide, oxygen, and product activity, were selected as output variables. During all the batch runs, growth was limited by insufficient oxygen concentrations caused by excessive feeding. A FIR model set was identified using data from 12 batch runs with high diversity, and cross-validating on data from 4 batch runs. The identified model showed significant potential for industrial use, when cross-validated through “Pure-Simulation”. As shown in Figure 4.1 an experiment with an exploratory recipe resulting in a 50% decrease in product activity as a direct result of constraint violations, could have been simulated using the model and it would not have been necessary to run the experiment. Hence, the experimental costs could have been avoided. Note that the poor predictive performance in the
initial phase of the batch depicted in Figure 4.1 is caused by a batch phase preceding the fed-batch phase, i.e., no input signal during the initial two time intervals.

4.2 Fermentation of Yeast

To simulate fed-batch fermentation of yeast a biochemically structured model presented by Lei et al. (2001) was used. The model describes aerobic growth of *Saccharomyces cerevisiae* on glucose and ethanol and focuses on the pyruvate and acetaldehyde branch points, where overflow metabolism occurs when the growth changes from oxidative to oxido-reductive.

For development of the set of FIR models extended with extra initial conditions, the following output variables were chosen: Ethanol concentration (EtOH), Carbon dioxide Evolution Rate (CER), Oxygen Uptake Rate (OUR), and Respiratory Quotient (RQ). The feed rate was selected as control variable. In fact, $RQ \equiv \frac{\text{CER}}{\text{OUR}}$, but here RQ was simulated as a measurement to
see if its dynamics could be captured with the suggested modeling framework. Gaussian white measurement noise was added to all inputs and outputs: EtOH max ±3%, OUR max ±2%, CER max ±2%, RQ max ±2%, feed rate max ±1%. The level of measurement noise was based on laboratory experiments and industrial data. It was assumed that there were no time delays.

The model set was identified on data from 10 normal batch runs and cross-validated through “Pure-Simulation” on 3 normal data sets. For the identification task a strongly exciting input sequence was chosen. The input sequence was constructed by adding a Pseudo Random Binary Sequence (PRBS) to the nominal input trajectory. At each sample the PRBS was uniformly scaled by maximum ±10% of the local nominal trajectory value. Note, strongly exciting input sequences are not system friendly and thus seldom applicable in practice. For further discussion of input sequences see Pearson (1998). The identification was regularized to ensure smooth model parameter evolutions and smooth impulse responses. The regularization weights were optimized by trial and error, minimizing the mean prediction error from pure simulation model cross-validation. The optimal regularization weights (λ1 and λ2 on L1 and L2, respectively; see page [26]) are shown in Table 4.1 along with their respective mean prediction fits. Figure 4.2 shows that the identified model set possesses
Figure 4.3: Time evolution of the first impulse response coefficient $g_{t,t-1}$ ($= \tilde{g}_{t,B_t}$; see eq.'s (3.39) and (3.42)) for each of the four output variables. It is evident that the fermentation (and hence the model) is time-varying and that the time evolution of the model parameters are nonlinear. The relative smoothness of the four model parameter time evolutions are in good agreement with the relative size of the regularization weights $\lambda_1$ in Table 4.1.

Based on the identified FIR model set the reference profiles of the four outputs were tracked for further model validation. In the control simulation scenario the initial conditions were randomly perturbed by maximum $\pm10\%$ in
4.2. Fermentation of Yeast

Each batch run and the feed concentration was perturbed by a persistent 10% reduction in all batch runs.

In Figure 4.5 it can be observed that the controller rejected approximately 80% of the Summed Squared Error Sequence (SSES), that is $e_k^T e_k$, in the first batch run and after having been trained on the first three batch runs more than 90% of SSES was rejected with the exception of batch run no. 6. The sluggish performance in batch run no. 6 was due to a particularly intractable disturbance direction, i.e., the yeast had been starved before seeding. However, the controller still rejected approximately 40% of SSES and the performance in the subsequent batch run was not affected despite the iterative learning of the controller. In addition to SSES, also the end-product quality (final yeast mass minus the mass of the impurities substrate and EtOH) and the product yield (the ratio between produced yeast and consumed substrate on a mass basis) are shown in Figure 4.5, and it can be observed that the end-product quality can be improved by output reference tracking alone and that the product yield may endure a minute decrease, which is the cost for high tracking performance.
Figure 4.5: Summed squared error sequence, quality, and yield evolution in noisy reference tracking of EtOH, OUR, CER, and RQ. No control corresponds to 100%.

Table 4.1: Optimal regularization weights and mean relative prediction fit from the identification of a FIR model set extended with initial conditions. \( fit = \frac{||y_k - y_{k-1} - G\Delta u_k||_2}{\sqrt{N}} \).

<table>
<thead>
<tr>
<th>Output variable</th>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
<th>( fit )</th>
</tr>
</thead>
<tbody>
<tr>
<td>EtOH</td>
<td>1.18</td>
<td>0.05</td>
<td>( 1.5708 \times 10^{-2} )</td>
</tr>
<tr>
<td>OUR</td>
<td>0.05</td>
<td>0.01</td>
<td>( 2.4224 \times 10^{-2} )</td>
</tr>
<tr>
<td>CER</td>
<td>0.05</td>
<td>0.01</td>
<td>( 2.0479 \times 10^{-2} )</td>
</tr>
<tr>
<td>RQ</td>
<td>0.26</td>
<td>0.02</td>
<td>( 2.3902 \times 10^{-2} )</td>
</tr>
</tbody>
</table>
The present chapter summarizes and discusses the main results and contributions presented in the present thesis. Suggestions for future work in related areas where issues remain unresolved are given in section 5.1. The main objective and result of the present thesis is the development of a generally applicable methodology for optimal and reproducible batch process operation. The development of this toolbox was commenced because of a need for systematic tools to alleviate some of the complications of batch processing reported by industry. The application of conventional tools often fail to reduce the complications of batch processing because of either the prohibitive complexity of the work needed to derive an accurate nonlinear dynamic model or the shortcomings in dealing with non-stationary and nonlinear systems of conventional linear system tools. The main objective of this thesis was therefore to develop a methodology with non-stationary enhancements of the well established conventional and easily obtainable, linear systems tools for modeling, control and optimization. This objective has been met with the development of an empirical time-series modeling methodology and complementary control and optimization methodologies.

The modeling methodology proposed in the present thesis comprises the development and an algorithm for identification of a batch Auto-Regressive Moving Average model with eXogenous inputs (ARMAX) and its transformation into state space representations designed for specific applications in control and optimization of batch processes. Inspired by the batch Finite Impulse Response (FIR) modeling ideas presented by [Lee et al. (2000)], the Linear Time-Invariant (LTI) batch model presented in the present thesis, comprise a set of local ARMAX models that model the incremental differences between operations on a batch process. The fact that it is the incremental differences between operations on a batch process that is modeled means that the batch ARMAX model includes input and output error integrators, which will prove advantageous in the design of control and optimization algorithms. Moving from a FIR model representation to an ARMAX model representation means that a broader class of systems can be adequately modeled and that the resulting models are significantly more parsimoniously parameterized and hence generalize better. Equally important, the local ARMAX models can be transformed into local state space (SS) representations, which means that established SS model based formulations of linear tools such as the Kalman filter,
Iterative Learning Control (ILC), and Model Predictive Control (MPC), are easily designed with the local models in a batch ARMAX model. Although the ARMAX model structure is significantly more parsimoniously parameterized than the original FIR model representation, estimating all the model parameters in the set of local models would require an infeasible amount of process data. As documented by Kourti et al. (1996) however, the number of effective parameters generally needed to reasonably accurately model batch processes with linear models, is very moderate. Therefore it is proposed to shrink the number of effective model parameters in the batch ARMAX model by penalizing excessive parameter variance and independency between local and thus similar, models when the parameters in the model set are estimated. Applying this type of regularized regression shrinks the number of effective parameters down to a level which is comparable to the level documented by Kourti et al. (1996). The two main methodological disadvantages of the proposed ARMAX model structure and regularized regression are the dimension of the regression problem and a significant number of tuning parameters that have to be chosen in some optimal manner. The dimension of the regression problem is easily handled with an efficient QR factorization based array algorithm. To overcome the significant number of tuning parameters that have to be chosen in some optimal manner, a model quality measure for model assessment and a multiobjective optimization algorithm for tuning parameter optimization have been developed specifically for ARMAX models. The two main practical disadvantages of the proposed modeling methodology are firstly, that the proposed batch ARMAX model is a local linear approximation and consequently that the use of the model is limited to a region local to the data set used for model identification, and secondly, that the availability of consistent process data is very limited in industry. Although the proposed modeling methodology comprises a data preparation procedure, lost or biased process information remains lost or biased.

The control and optimization methodologies proposed in the present thesis, comprise constrained Optimal Control Problem (OCP) formulations of disturbance attenuating control for reproducible operation and batch operations model optimization for optimal operation of batch processes and a procedure for output and state estimation for batch processes. The proposed OCP formulations for optimal and reproducible batch operation have been developed specifically for use with SS representations of the batch ARMAX model from the proposed modeling methodology. The control methodologies for reproducible batch operation comprise the constrained OCP formulations ILC for LTI SS model representations and learning MPC for LTV SS model representations. Under the general assumption of SS model representations being exact representations of batch processes, a necessary and sufficient condition for guaranteed closed-loop stability of the proposed constrained ILC and learning MPC OCP formulations has been developed. When the developed necessary and sufficient stability condition is satisfied, the proposed ILC and learning MPC algorithms have been proven to converge to a steady state level of repro-
ducibility, and it has furthermore been proven that the proposed learning MPC algorithm will outperform the proposed ILC algorithm in terms of maximum reproducibility. Such a necessary and sufficient stability condition and stability proofs for constrained ILC and learning MPC implementations with imperfect state information and unreachable references is most likely unprecedented. The necessary and sufficient stability condition was developed by reverse engineering the concept of a successful closed-loop control implementation, i.e., convergence to a lower level of batch process variability. That the proposed learning MPC algorithm outperforms the proposed ILC algorithm, is to be expected as the proposed learning MPC algorithm utilizes the process data that becomes available during operation of a batch process, whereas the propose ILC algorithm relies on information available before a batch operation is commenced. The same learning MPC superiority need however not be the case if the batch model is uncertain. The control methodology for batch operation optimization comprise a definition of optimal batch operation as a constrained Linear Program (LP) and an Iterative Operations Model Optimization (IOMO) OCP algorithm to track optimal batch operation. For LTI SS batch representations, the IOMO OCP algorithm for batch operation optimization has been proven to converge to the optimal batch operation. The main disadvantage of the proposed control methodology for batch operations model optimization is the definition of optimal batch operation as a constrained LP which is not particularly flexible but which was chosen because it complements the proposed OCP algorithms for reproducible batch operation. The main disadvantage of the proposed collective control and optimization methodologies, is that stability and convergence have “only” been proven for batch processes which can be accurately approximated globally with a local linear model set. Chemical batch processes can however almost certainly not be globally approximated with linear models and identified linear models such as those proposed in the present thesis, will be local to the data from which they were identified. The proposed collective control and optimization methodologies also do not consider model uncertainty which will always be present if the applied models are obtained with the proposed modeling methodology. It has been demonstrated through simulated case studies however, that the application of the proposed linear OCP algorithms designed from uncertain linear models, can improve operational performance and reproducibility of nonlinear batch processes. Finally, the output profile and state estimation required by the proposed control and optimization methodologies, has been developed as a combination of a Kernel Smoother and a Kalman filter. The contribution here is, that smooth estimates of output profiles are obtained between batch operations while state estimates are obtained during batch operation with a combination of a Kernel Smoother and a Kalman filter. Furthermore, with this combination it is possible to utilize data sampled at multiple frequencies and with different and significant, time-delays.
5.1 Future Work

Although the present thesis proposes a coherent methodology for optimal and reproducible batch process operation, there are still several issues that should be investigated further in the future. The unresolved issues that could bring the most value to this methodology are:

When compared to the simple yet successful and very similar, approach to batch process modeling proposed by Flores-Cerrillo and MacGregor [2004], it should be investigated whether the approach to batch process modeling proposed in the present thesis, would benefit from being simplified. As part of this investigation it should also be investigated how often and when during its operation a batch process should be sampled in order to obtain the best closed-loop performance.

The issue of the viability of the proposed model quality measures and tuning parameter optimization algorithm remains unresolved and should be a focus area for future investigations. The model quality measures and tuning parameter optimization algorithm could potentially find use outside the present modeling methodology.

The issue of how to monitor and/or guarantee validity/stability of closed-loop implementations based on uncertain local model approximations remains unresolved. This is the most critical issue with respect to practical implementations of optimal and reproducible batch process operation, and should therefore be key focus area investigations in the near future. A related issue is to establish guidelines for the design of the weighting matrices, covariance matrices, Kernel Smoothers, etc. when the model representation of the batch process is uncertain.
References


