Evaluation of a batch process by means of batch statistical process control and system identification

Sebastian Larsson

Eric Grundén

Automatic Control

Perstorp AB

MSc Thesis
ISRN LUTFD2/TFRT--5974--SE
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2015

Abstract

Batch processes play an important role in the production of high quality specialty chemicals. Examples include the production of polymers, pharmaceuticals and formulated products. In this master thesis, the study of transformation of materials, by batch distillation and mixing is studied. The study is done by means of batch statistical process control and system identification methods in order to build soft sensors that can predict product quality and end-point but also to use the batch trajectory features for early fault detection.

In contrast to a continuous process, a batch process is a finite duration process, from initialization to completion. The physical state of the process is derived from measured variables, for example, temperatures, pressures and flows and comes from on-line measurements of the on-going process.

Since there are many variables, in terms of inputs and outputs, multivariate data analysis is a suitable choice for extracting systematic information which is used to find a relationship among the variables but also to visualize the batch trajectories and deviations from normal batch evolution.

The results suggest that the end point can be predicted during distillation and mixing and it seems like it is possible to separate normal batches from different batches by means of batch statistical process control strategies. However, estimating product purity during distillation was not possible due to limited variation in the output data. Instead, system identification methodologies were a better choice.

Product quality after mixing was poorly estimated with system identification tools due to the lack of variability within the time average of the different variables used, but was better predicted with batch statistical process control.

Keywords System identification, Sub space identification, Kalman filter, Batch statistical process control, Partial least squares, Batch process

Acknowledgements

We would like to extend our gratitude to Perstorp AB, who has been willing to have us as master thesis students, and especially Robert Zuban and Johan Rönnberg for coming up with the project idea and being our mentors throughout this master thesis. We would also like to give Sixten Dahlbom a big thanks for his help and expertise regarding the batch process and general process knowledge.

We would also like to thank Dr Rolf Johansson for helping us with all of the theoretical and administrative issues regarding the thesis project. Every time we ran in to problems, we could always turn to him for guidance.

One last person, we would like to extend our appreciation to is Alexandra Ivanov, for lending her car to us. This made the weekly trips to Perstorp a lot smoother, thus created the possibility to focus more on the thesis project rather than spending time commuting.

List of Abbreviations

Table 0.1 List of Abbreviations.

Abbreviation	**Explanation**
BSPC	Batch Statistical Process Control
MOESP	Multivariable Output Error State sPace
LS	Least Squares
PLS	Partial Least Squares
BEM	Batch Evolution Model
BLM	Batch Level Model
SVD	Singular Value Decomposition
RMSEP	Root Mean Square Error Prediction set
RMSE	Root Mean Square Error

List of Figures

List of Tables

Contents

1

Introduction

Most process industries today use process monitoring to ensure that all of the requirements and specifications regarding quality, safety and economics are fulfilled. The process monitoring is usually visualized as time series charts, where the operators view the variables as historical trends. Since the processes today grow in size as well as in complexity, the different variables measured and stored in a database increases. The interpretation, based on an increasing set of variables, might result in an inconclusive conclusion. Therefore, multivariable methods are needed to cope with the problem of an increasing set of variables, as well as a number of other factors. Some examples of different factors are an increasing number of data sets, colinearity and non-linear relationships between variables as well as non-stationary, noisy and missing data.

In this master thesis a batch process at Perstorp AB has been studied by means of batch statistical process control and system identification which are two multivariate methodologies that can handle the above mentioned difficulties.

1.1 Background

Products manufactured in a batch process at Perstorp AB rely solely on lab samples to determine product compositions. This procedure is something that is expensive to extract and evaluate, and want to be kept to a minimum. The ability to track and measure the product purity in real-time is not possible today and is compensated by previous knowledge and experience of the operators.

One way to estimate and visualize the product quality for the operators would be to use the stored measurements of the different variables from the batch process. The different variables reflect the current state within the process, information that could be used to derive soft sensors, which are on-line estimators that can measure a certain property that cannot be measured on-line, or is very expensive to measure on-line. The soft sensors would be based on different multivariate methodologies such as batch statistical process control and system identification.

The information extracted from the multivariate methodologies can be used for process monitoring, end point- and product purity prediction and fault detection, which would be a valuable tool for process operators.

1.2 Purpose

The main purpose of the project was to investigate a batch process by means of batch statistical process control and system identification. This involved to examine if it were possible to derive models that were able to estimate the product quality of a certain product, or products, and to see if it was possible to estimate a termination time. This was done to see if was possible to minimize any back-off margins in the products manufactured today, and to see if it was possible to shorten the batch duration.

The unit operations in the batch process involved mixing and distillation. The products that were examined are called Product A, and a chemical mixture called Product B. The two products are described more in detail in Section 1.4. The unit operation evaluated for Product A was during its distillation phase and Product B was examined during its mixing phase.

There were several different tasks involved in the thesis. The main issue was to see if it was possible to derive models that were able to describe the product quality with the help of variables describing the physical state of the batch process. An end point prediction, too see if it was possible to estimate batch termination time, depending on how far the batch has gone. The derived model, or models should also be able to work as an early fault detection model, telling the operators whether the current batch is within normal operating conditions. Forward prediction by looking k-steps into the future before batch completion, by using a limited portion of the data from a batch cycle. A methodology comparison reflecting the pros and cons of the different methodologies.

The different objectives were tackled from two different perspectives. From a system identification and a batch statistical process control point of view. This created the possibility to work from different angles, thus creating a better way to interpret, analyze and draw conclusions based on a larger set of methodologies.

1.3 Perstorp AB

Perstorp AB is a chemical company with a focus towards the specialty chemicals market. With over 125 years of experience, Perstorp AB is considered as one of the worlds leading specialty chemical companies in many sectors. Their products are used in many different industries, such as aerospace, coatings, chemicals and plastics. For more information about their products, or the company, see [*Perstorp*].

1.4 Products

The thesis focused on two different products that Perstorp AB currently produces. The two products are Product A and Product B. Product A was chosen as the first product to examine, since the properties of the product is easy to interpret and analyze, as compared to other products. Product A contains about 99.5% of a certain substance called "II". Product B was the second product that was examined. The reason for this was that the variation of the different properties were higher, and a correlation pattern might be easier to find. Product B contains several constituents and are named "I"-"XIV".

1.5 Process description

The process used in the thesis is a batch distillation column. The two products are both formed in the batch distillation process, but in different unit operations. The first stage of the batch process, is to mix different raw materials under low pressure and elevated temperature to change the chemical composition of the mixture. When the mixing phase is completed, the mixture is either sent to a tank and departed for sale, or heated once more, separating the light chemicals from the heavy ones. The separated chemicals is then sent to different tanks for storage and is either ready to be sold or resent back to the process for separation once more. Product B is studied when mixing and Product A during distillation.

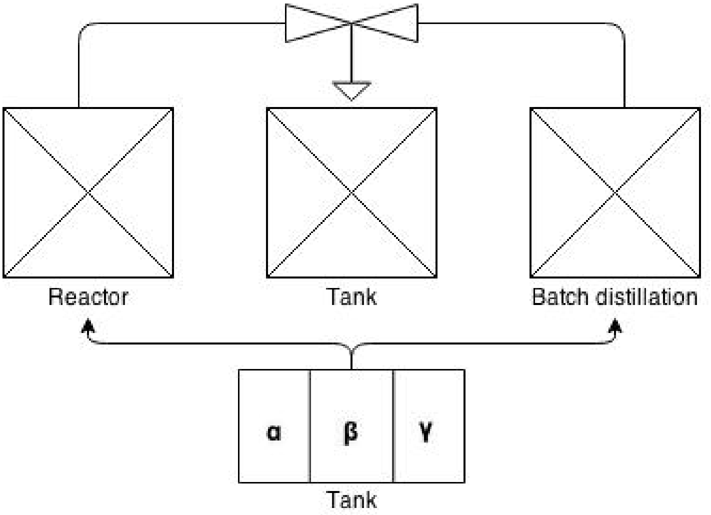


Figure 1.1 Simplified figure of the reactor and distillation. The raw material comes from the different tanks, α, β and γ. The product is put in an output tank.

1.6 Software

1.6.1 Matlab®

Matlab® is a numerical computing environment developed by Mathworks. The program is used for doing various of different mathematical matrix- and vector operations and computations. The output can be visualized in numerous ways with the help of different methods implemented in the program. [*Matlab*]

1.6.2 SIMCA

SIMCA is a program used for analyzing uni- and multivariate data sets. The graphical nature of the program makes it easy to analyze and interpret large data sets. The program is developed by Umetrics. For more information about the program, or the company, see [*SIMCA*].

2

Theory

In this chapter, the theory used in the project is briefly described. The basis of the different methodologies used in the project relies on different versions of linear regression. The regression method used in the system identification methodology was least-squares estimation and the counter-part in batch statistical process control was partial least-squares. Since batch processes has multiple data sets, a data structuring and unfolding is required. This was mainly used when working with the batch statistical process control concept, and is also explained in the chapter. To be able to find the appropriate model order for the different system identification models derived, singular value decomposition was done. To be able to simulate the different models derived with the system identification concept, a Kalman-filter together with a recursive prediction algorithm was implemented and is described below.

2.1 Linear regression

Linear regression is used to describe an output as a function of inputs [Johansson, 1993]. The output is often denoted y and the inputs are denoted with $\phi = [\phi_1, \phi_2, \ldots, \phi_J]^T$. The task will thus be to find some parameters $\theta = [\theta_1, \theta_2, \ldots, \theta_J]^T$ that describe the relationship between the output as a function of inputs. The relationship can be described via the function

$$y(k) = \phi^T(k)\theta + e(k) \tag{2.1}$$

If one assumes there are K number of input- and output observations available, the task will then be to find optimal parameter estimates $\hat{\theta}$ of the parameter vector θ that describes the output as a function of inputs.

By re-arranging the available observations in vector- and matrix notation, the resulting model for linear regression will then be

$$Y_N = \Phi_K \theta + e \tag{2.2}$$

where

$$Y_K = \begin{bmatrix} y1 \\ y2 \\ \vdots \\ y_K \end{bmatrix}, \Phi_K = \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_K^T \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_K \end{bmatrix} \tag{2.3}$$

2.1.1 Least-squares estimation

The least-squares criterion is a parameter estimation approach, which aims to minimize the sum of the squared errors between the output and the observations [Johansson, 1993].

$$\underset{\hat{\theta}}{argmin}\ \frac{1}{2}\sum_{i=1}^{K} e_i^2 = \frac{1}{2}(Y_K - \Phi_K\hat{\theta})^T(Y_K - \Phi_K\hat{\theta}) \tag{2.4}$$

After some computation, the optimal parameter estimation $\hat{\theta}$ is obtained from Equation 2.5

$$\hat{\theta} = (\Phi_K^T\Phi_K)^{-1}\Phi_K Y_K \tag{2.5}$$

2.1.2 Partial least squares

Partial least squares (PLS) [Ferrer et al., 2008] is used when analyzing data with more variables than observations. PLS works in the same manner as principal component analysis (PCA) [Bro and Smilde, 2014], but the resulting model will instead try to maximize the covariance between the input X and output Y. PLS tries to maximize the covariance between X and Y with the help of the scores and loadings from X and Y.

$$Y = u_1 q_1^T + \cdots + u_R q_R^T = UQ^T + F = \hat{Y} + F \tag{2.6}$$

The matrix U corresponds to the score matrix, Q matrix is the loading matrix and F is the residual matrix. Depending on how many score vectors one choose, the better the resulting approximation $\hat{Y}$ will be.

The different score matrices U and T will be used when trying to predict the output instead of measuring it. The prediction quality will be dependent on how well the different components describe the data.

2.2 Data unfolding

In order to use information from multiple data sets, one would need to arrange the data in a specific way. Each data set will contain J variables and K observations. An extra dimension I will describe the number of data sets. Together they will form a three-way matrix called $\underline{X}$. There are different ways of unfolding a three-way matrix [Eriksson et al., 2013].

One way of unfolding the three-way matrix, is to keep the direction of the variables. The unfolded matrix X is depicted in Figure 2.1.

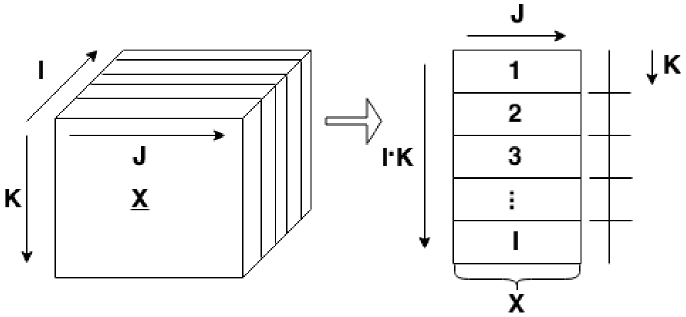
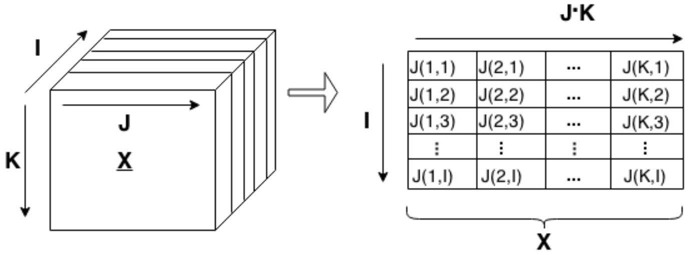


Figure 2.1 How to unfold a three-way matrix into a two-way matrix and keeping the direction of the variables. J is the number of variables in the data set, K is the number of observations and I is the number of data sets.

Another way to unfold the matrix is to preserve the direction of the different data sets. In this master thesis work data sets correspond to number of batches. The preservation is done by keeping each data set on separate rows, and is described more in detail in Figure 2.2.



J(1,1)	J(2,1)	...	J(K,1)
J(1,2)	J(2,2)	...	J(K,2)
J(1,3)	J(2,3)	...	J(K,3)
⋮	⋮	⋮	⋮
J(1,I)	J(2,I)	...	J(K,I)

Figure 2.2 How to unfold a three-way matrix into a two-way matrix and keeping the direction of the data. Where J(o,d) corresponds to J variables at observation o of data set d.

The unfolded matrix X will be an $J \cdot K \times I$ matrix. The first J columns will be the variables at the first sample, and the following columns will be the J variables at the next sample. The number of rows depends on the number of data sets I.

2.3 Singular value decomposition

Let X be an arbitrary matrix, of the size $J \times K$, with real- or complex valued elements. Then there exists a factorization on the form

$$X = U\Sigma V^{*} \tag{2.7}$$

Where U and V^{*} are unitary matrices of the size $J \times J$ and $K \times K$. Σ is a diagonal matrix of the size $J \times K$, where the diagonal elements are known as the singular values of X. The diagonal entries σ_i in Σ are non-negative. The singular values in Σ are structured in a descending order. [Johansson, 1993]

2.4 Hankel matrix

A Hankel matrix is a symmetric matrix whose elements are constant along its anti diagonals. Given a sequence of data

$$X_k \in \mathbb{R}^{J \times K} \; \forall \; 0, 1, \ldots, k, k+1, \ldots, m \tag{2.8}$$

where J and K is the number of rows and columns in X_k.
By defining two variables α and β, the Hankel matrix is defined as

$$H_{k|\alpha} = \begin{bmatrix} x_k & x_{k+1} & \cdots & x_{k+\beta-1} \\ x_{k+1} & x_{k+2} & \cdots & x_{k+\beta} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k+\alpha-1} & x_{k+\alpha} & \cdots & x_{k+\alpha+\beta-2} \end{bmatrix} \in \mathbb{R}^{\alpha \cdot J \times \beta \cdot K} \tag{2.9}$$

α and β are the number of m- and n-block rows and columns in $H_{k|\alpha}$ [David Di Ruscio, 1995].

2.5 Kalman filter

The famous Kalman Filter can be used in a number of different areas [Johansson, 1993]. Some examples are noise reduction, prediction and estimation of a certain property. The filter is based on a state space model, which can be seen in Equation 2.10.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k) \\ y(k) &= Cx(k) + Du(k) + e(k) \end{aligned} \tag{2.10}$$

In Equation 2.10, A is the state transition matrix, B the input matrix, C the output matrix and D is the feed-through matrix. v is the noise affecting the system, and e is the noise affecting the measurement. The filter is divided in to two steps, one prediction step and one measurement update step.

To exclude any mean terms changing the system dynamics from the input signals, a differentiation was done. The differentiation procedure is described in Equation 2.11.

$$\begin{aligned}\Delta x(k+1) &= x(k+1) - x(k) \\ \Delta u(k) &= u(k) - u(k-1) + u_0 - u_0 \\ \Delta x(k+1) &= A\Delta x(k) + B\Delta u(k) + \Delta v(k) \\ \Delta y(k) &= C\Delta x(k) + e(k)\end{aligned} \tag{2.11}$$

The differentiated state space is used recursively to predict k-step ahead. Since the input is unknown at k-step ahead prediction, the mean of future inputs at each time lag was used, from a number of batches. The recursive k-step predictor is described in Equation 2.12

$$\Delta\hat{y}(k+s) = C(A^s\Delta x(k) + AB\sum_{i=0}^{s-2}(\Delta u_{mean}(k+i)) + B\Delta u_{mean}(k+s-1)) \tag{2.12}$$

3

Method

In the Method Chapter, the different methods used in the thesis are explained. At first an explanation of data retrieval and structuring is described. This is followed by two methods within batch statistical process control, called batch evolution model and batch level model. Within the system identification concept there is two methods described, that are named multivariable subspace identification and least-squares estimation.

3.1 Data retrieval and structure

The different batch data sets were fetched from a database, and structured in Excel based on tag names of a given number of different variables in the process. The data were structured by using start- and end times of a batch, representing the batch duration, by using valve values as reference points. The different valve values used describes when the tanks are opened and closed. To ensure that the whole course of a unit operation was fetched properly an hour before and after batch start- and completion was added.

The initial chemical composition in the blend, was fetched, structured and saved for all batches. The starting quality for each batch is dependent on up to three different tanks. In Table 3.1, the product composition for each tank is measured. The start composition of a batch is calculated by using composition data and the total amount taken from each tank. In case of a missing initial product composition from one of the tanks, as seen in Table 3.2, the initial condition was removed. The batches that had a missing initial condition were still used when deriving a model that was only based on the process matrix.

Table 3.1 How calculated composition is stored when initial conditions from each tank is available.

Tank	Date	Info
α	2011-01-27	Composition in tank α
β	2011-01-31	Composition in tank β
γ	2011-01-31	Composition in tank γ
	2011-01-31	Resulting calculated composition from tank α, β and γ

Table 3.2 How calculated composition is stored when initial conditions from tank γ was missing.

Tank	Date	Info
α	2011-02-07	Composition in tank α
β	2011-02-07	Composition in tank β
γ	2011-02-07	Missing value in tank γ
	2011-02-07	Non calculated composition

3.2 Pre-processing

There are many advantages with pre-processing the data before using any methodologies on it. By smoothing, mean centering, normalizing and filling out missing data points, a pre-processed data set might remove unwanted behavior when creating a model. Since the methodologies used in this context are derived in two different programs, the pre-processing varies.

3.2.1 Batch statistical process control

The observations used were mean centered and scaled to unit variance, and all the valve values with a value of zero were removed in the same manner as in Section 3.2.2, before using any regression methods on it. By using PLS regression, as described in Section 2.1.2, on a unfolded process matrix X together with a matrix Y as time, and visualizing it in a scatter plot, together with Hotelling's T^2 tolerance ellipse, it is possible to detect deviating data point that does not coincide with the rest of the data points. A contribution plot, indicating which of the variables that contributed to a unexpected change in the process, was also used to notice any unwanted behavior [Eriksson et al., 2013].

3.2.2 System identification

Since the different batch data sets had extra time added to the start- and end times of a batch, the data sets had to be trimmed accordingly. This was done with the help of valves values and when they were opened and closed. The start- and end time of a batch, is when the valves that lead to the end tank are opened. This was done by looping through the data, searching for valve values equal to one, and then removing all of the measurements when the valves were closed. The valve value was equal to zero when the valve was closed.

3.3 Batch statistical process control

A batch process differ from a continuous process in the sense that there is a finite duration of the process, with initialization and completion. With this in mind, the data structuring of a batch process will form a three-way matrix, rather than a normal two-way matrix as in continuous- or discrete processes. Several different techniques has been developed over the years to handle the three-way matrix problem [Eriksson et al., 2013]. In this context, the data unfolding will be done as in Section 2.2. The BSPC concept is divided into two parts and described more in detail below.

3.3.1 Batch evolution model

The batch evolution model is used to evaluate and visualize the variability within a batch, thus creating a model that can accomplish early fault detection and predict batch maturity. The data unfolding is done as in Figure 2.1. To follow the maturity within a batch, the X matrix is regressed against time with PLS. The results from PLS will describe the maturity within a batch at each time point. PLS is described in Section 2.1.2.

The derived model will also be able to visualize an evolution trace of a normal batch. If a batch deviates from its normal operating conditions, the model should be able to recognize this as a deviating batch.

One way to determine if the model is sufficient, and if a correlation pattern has been found, is to check for the following; The cumulative sum of squares of the Y matrix explained by the components ($R^2Y(\text{cum})$) and the cumulative fraction of the total variation of X and Y for the component (Q^2). The cumulative sum of squares of the matrix X describes how much of the process data that can be described with a chosen number of components [Eriksson et al., 2013].

If the resulting model is accurate, it should be able to predict an inflection point for when a batch has reached its specifications and thus should be terminated. One way to improve the model further is to lag either the variables in X, or Y, creating dynamics and auto regression in the model by using previous data as well.

3.3.2 Batch level model

The batch level model uses the batch process data matrix X and the final product purity value Y to search for a correlation pattern between matrices. A new matrix called X_A is created, where the initial conditions Z and the process data is stored. The X_A matrix is then used to create a model whom might be able to predict Y.

To accomplish this, the data will be unfolded as in Figure 2.2, where the batch direction is preserved. The matrix structure for batch level model is seen in 3.1, where each row in X_A and Y correspond to information from one batch. The matrix X_A is regressed against product purity with PLS.

The structure stated above, can also be done in an hierarchical structure. The initial condition matrix Z and the process matrix X is regressed separately towards the end point product purity, creating two base models. Each base model is then summarized by one or more score vectors which are transferred to a top level model, which then describes product purity from two base models

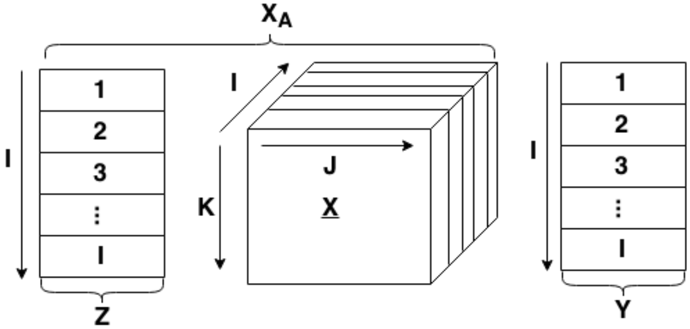


Figure 3.1 Structure of matrix X_A, with initial conditions from each batch in Z, batch data in $\underline{X}$ and Y, with product quality data from each batch.

3.3.3 Early fault detection

Based on a batch evolution model it is possible to achieve fault detection. This is based on the complexity of the model and how well the model estimates a certain value. The fault detection is divided into two parts. The first part evaluates the residual between the estimated and the real value. The distance is called Q and is orthogonal to PC (Principal Component). The second part is based on the direction of the score vector and detect deviations in the systematic part of the data. The distance between the estimated value and the centre of the main cluster is called D and is parallel with PC. In Figure 3.2 an example of an estimated value with a large residual, corresponding to Q, and a deviating point with a distance D to the centre of the main cluster.

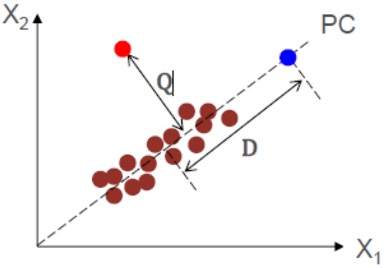


Figure 3.2 A picture describing the fault detection method, with a distance Q that is orthogonal to PC. Distance D describes the distance between a value and the centre of the main cluster and is parallel with PC.

3.4 System identification

In control theory, many different techniques have been developed over the years, to describe an observed input-output behavior with a mathematical model [Johansson, 1993; Ljung, 2010]. This is known as system identification, and is a very broad topic, due to a number of different variants in regards of real world observations, and the will to describe them.

3.4.1 Multivariable subspace identification

A multivariable subspace identification model [Qin, 2006] was derived with the X- and the Y matrix. The particular subspace identification method used, is called Multivariable Output Error State sPace (MOESP). In this context, the X matrix is the same as the U matrix . The procedure uses input- and output matrices to find an estimate of the different state space matrices A, B, C and D. The resulting state space model is shown in Equation 3.1.

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= Cx(k)+Du(k)+e(k) \end{aligned} \tag{3.1}$$

The first step is to structure the input- and output data as Hankel matrices. Hankel matrix structuring is described in Section 2.4. Each input u_k and y_k will be stored in Hankel matrices U and Y. The next step is to do a QR-factorization. This is done as in [Johansson, 1993], but since this will play a vital part in the system identification step, it is described below.

$$\begin{bmatrix} U^{(1)}_{r,N-r+1} \\ Y^{(1)}_{r,N-r+1} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{3.2}$$

The column space of the submatrices in the R-matrix is computed by the means of singular value decomposition. This will result in singular values, and an estimate of the extended observability matrix [Johansson, 1993; Verhaegen, 1994]. By using the extended observability matrix, an estimate of the matrices $\hat{A}$ and $\hat{C}$ was computed.

The last step, in the system identification procedure, was to estimate the matrices $\hat{B}$ and $\hat{D}$. This was done by transforming the equation as in [Johansson, 1993]. The transformation equations used, can be seen in Equation 3.3 and 3.4.

$$Y^{(1)}_{r,N-r+1} = CX^{(1)}_{r,N-r+1} + DU^{(1)}_{r,N-r+1} = R_{21}Q_1 + R_{22}Q_2 \tag{3.3}$$

into

$$(U_n^{\perp})^T D - (U_n^{\perp})^T R_{21}R_{11}^{-1} = 0 \tag{3.4}$$

The corresponding estimated state space model is seen in Equation 3.5

$$\begin{aligned} x(k+1) &= \hat{A}x(k) + \hat{B}u(k) \\ y(k) &= \hat{C}x(k) + \hat{D}u(k) + e(k) \end{aligned} \tag{3.5}$$

3.4.2 Least square estimation of product purity.

One approach used in the project, when trying to estimate product purity, was to use linear regression analysis. The idea is to use linear regression, and least-squares estimation, to find a set of parameters that can estimate the end product purity of a batch, with the help of a number of input parameters. One thing to notice, and keep in mind, when doing the linear regression analysis, is to notice that the end product purity values are dependent on the time average of the input signals. By exploiting the number of batches to get a lot of end product purity values, the task was then to find a set of parameters that describe the output as a function of the time average of the input signals, as in Equation 3.6.

$$\begin{aligned} y_{purity,end(1)} &= f(u_{mean}) + v \\ &\vdots \\ y_{purity,end(N)} &= f(u_{mean}) + v \end{aligned} \tag{3.6}$$

Equation 3.6 was rewritten as in Section 2.1, and the parameter estimation was done as in Section 2.1.1. When an $\hat{\theta}$ estimate had been calculated, and Equation 3.6 had been written in matrix notation, it was possible to get an estimation of the product purity.

$$Y_{purity} = U^T_{mean}\hat{\theta} + e \tag{3.7}$$

4

Results

The Results Chapter has been divided into the two methodologies that were examined. These sections are further divided into Product A and Product B.

4.1 Batch statistical process control

4.1.1 Product A

Pre-processing

A total of 19 batch data sets were evaluated with the BSPC concept. The Z matrix were not available for Product A, and the derived models were not based on the initial product composition. A PLS on the X matrix with all variables, regressed against time, before and after removal of outliers and unwanted behavior is seen in Figure 4.1. Corresponding R^2X, R^2Y and Q^2 before and after removing outliers can be seen in Table 4.1.

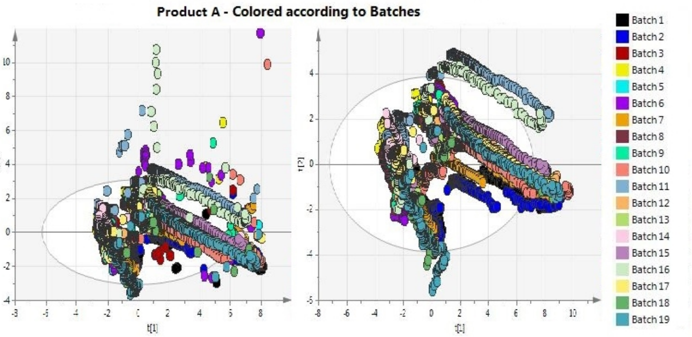


Figure 4.1 PLS score scatter plot before and after removal of outliers, based on Product A batch trajectories. The left subfigure is before removing any outliers, and the right subfigure is after removal. The scatter plot is colored according to batches.

Table 4.1 PLS of raw data of Product A with time as Y, with and without any removal of outliers or variables. R^2X, R^2Y and Q^2 describes how good the model estimate Y and X.

Removal	**Components**	R^2X	R^2Y	Q^2
Before	2	0.531	0.674	0.674
After	2	0.781	0.681	0.681

Estimation of product purity

A BLM was computed with the X matrix, with the product purity end points as the Y matrix. The results can be seen in Table 4.2.

Table 4.2 R^2X, R^2Y and Q^2 values of BLM of Product A with product purity as Y and pre-processed data as X.

Components	R^2X	R^2Y	Q^2
2	0.632	0.344	0.03
3	0.700	0.453	-0.287

Same procedure was done, but with the score vectors of the data set, after removal of unwanted behavior and outliers, as the X matrix. The results can be seen in Table 4.3.

Table 4.3 R^2X, R^2Y and Q^2 values of a BLM of Product A with product purity as Y and score vectors as X.

Components	R^2X	R^2Y	Q^2
2	0.727	0.178	0.178
3	0.786	0.290	-0.287

None of the product purity models above worked in a satisfying manner, and the reason for this is discussed more in detail in the Discussion Chapter.

Variable simulation and reduction

The operators use temperature, pressure and density as reference points during batch distillation. As seen in Table 4.4, the temperature variable is best explained due to having the highest R^2Y value. The X matrix contains all the variables. In Table 4.4, it can also be seen that a static temperature model is as good as a dynamic model, based on lagging the process variables backwards in time (auto regression).

Table 4.4 R^2X, R^2Y and Q^2 based on PLS regression, for three different variables. One static (1,5,9), and three dynamic models (2-4,6-8,10-12) with lag 0,1,3 and 5. The X and Y, in the lag column, describes which part is lagged. Number of components for all of the models were set to 2.

No	Variable	Lag	R^2X	R^2Y	Q^2
(1)	Temperature	0	0.766	0.964	0.964
(2)	Temperature	1 (X,Y)	0.781	0.971	0.971
(3)	Temperature	3 (X,Y)	0.781	0.972	0.972
(4)	Temperature	5 (X,Y)	0.772	0.974	0.974
(5)	Pressure	0	0.769	0.791	0.791
(6)	Pressure	1 (X,Y)	0.774	0.794	0.794
(7)	Pressure	3 (X,Y)	0.775	0.795	0.795
(8)	Pressure	5 (X,Y)	0.774	0.796	0.796
(9)	Density	0	0.676	0.767	0.767
(10)	Density	1 (X,Y)	0.679	0.800	0.800
(11)	Density	3 (X,Y)	0.679	0.817	0.817
(12)	Density	5 (X,Y)	0.669	0.844	0.844

All temperature variables except the temperature variable in the Y matrix was removed, due to high correlation. R^2X, R^2Y and Q^2 values are seen in Table 4.5, after removing all of the temperature variables from the X matrix.

Table 4.5 R^2X, R^2Y and Q^2 based on PLS regression against temperature, with other temperature variables removed in X.

Components	R^2X	R^2Y	Q^2
2	0.757	0.906	0.906

Model reduction was done, to make the model less complex and to remove variables that are non-significant without losing predictive power. As seen in Figure 4.2, there are 4 variables that has a low weight, after removing correlated temperature variables.

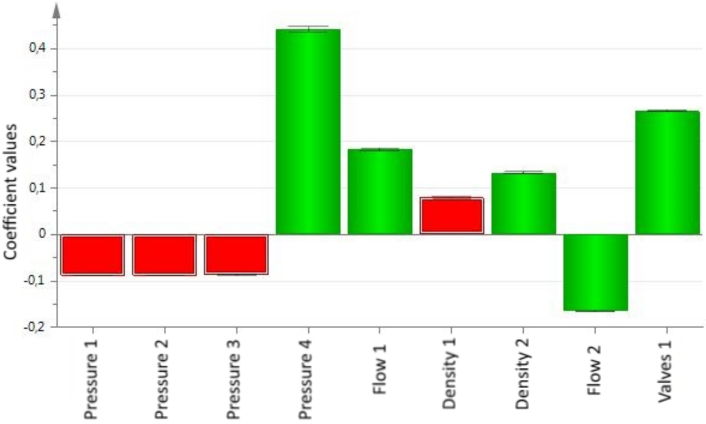


Figure 4.2 Variable weights describing the temperature of Product A, without other temperature variables in X. The red variables (1-3,6) has a low contribution when trying to estimate the temperature.

The final static temperature model is dependent on a flow and a pressure variable and can be seen in Figure 4.3. Higher temperature correspond to higher pressure and lower flow. The resulting R^2X, R^2Y and Q^2 values can be seen in Table 4.6.

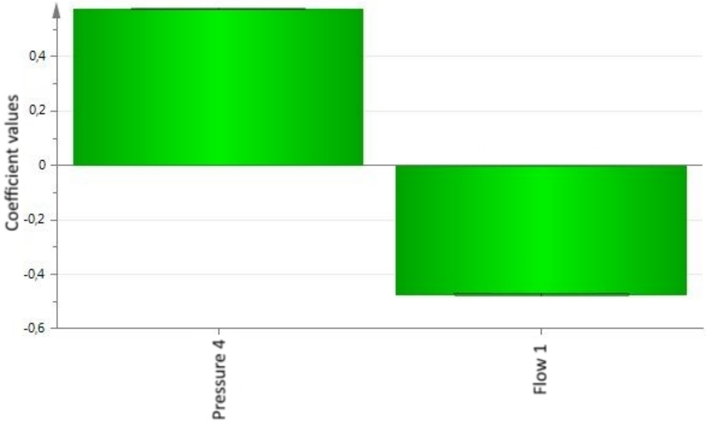


Figure 4.3 Final reduced temperature model of Product A. Model is only dependent on a flow and a pressure variable.

Table 4.6 R^2X, R^2Y and Q^2 for the reduced temperature model of Product A, with two components. Where the Y in the lag column indicates that it is this variable that has been lagged.

Lag	R^2X	R^2Y	Q^2
0	1	0.907	0.907
1 (Y)	0.840	0.957	0.957

Early fault detection

The derived model, in Table 4.1 after removal of outliers, is used to recognize a deviating batch. The derived model was used together with a deviating batch, and a visualization of the resulting score vector, and distance to model can be seen in Figure 4.4 and 4.5. A contribution plot was used, to visualize the reason for the deviating behavior, which is further discussed in the Discussion chapter. This can be seen in Figure 4.6.

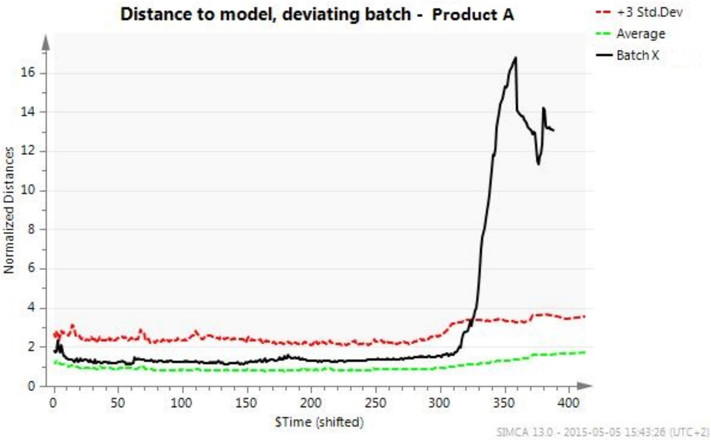


Figure 4.4 Distance to the non-reduced model with time as Y, after removal of outliers, of Product A for a deviating batch. Detects a fault at 320 minutes. The green line is the average and the red is ±3 standard deviation for the temperature model.

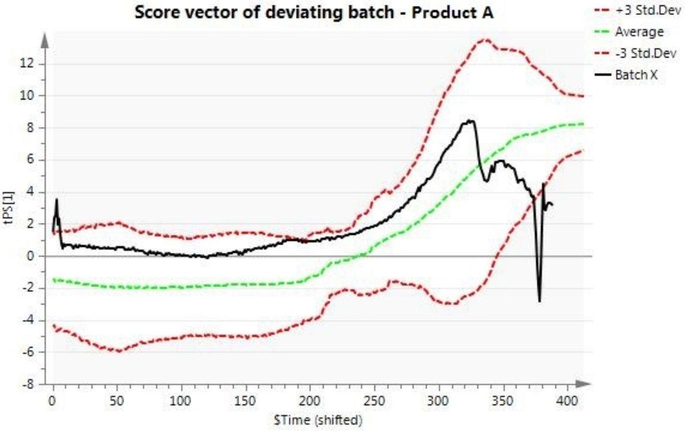


Figure 4.5 Scores vector from the deviating batch. The green line is the average and the red is ±3 standard deviation from the score vectors of all batches.

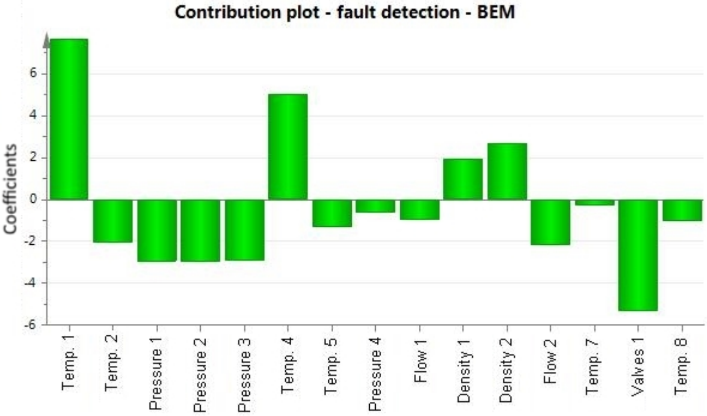


Figure 4.6 Contribution plot visualizing which of the variables that is causing the deviation.

Prediction of inflection point

Temperature is one of the set points used to determine batch termination. The time when a batch has reached its temperature break point was saved, and used together with the non-reduced temperature model to create a BLM. Partial models were created, to visualize how well the model could calculate an estimated end point, depending on how long the batch has gone. The results can be seen in Table 4.7. A visualization of a partial model, at 100% is seen in Figure 4.7.

Table 4.7 Partial models of end point time prediction with BLM. Standard deviation σ of batch end time was 43.9 minutes. Three components where used to describe X. Root-mean-square error of prediction in minutes of three batches, from a total of 20 batches, used as prediction set where the mean length of a batch is 374 minutes.

Time	*RMSEP*	R^2X	R^2Y	Q^2
20%	21.27	0.873	0.824	0.665
40%	22.31	0.830	0.863	0.693
60%	19.80	0.796	0.872	0.689
80%	22.74	0.714	0.948	0.743
100%	14.39	0.650	0.966	0.779

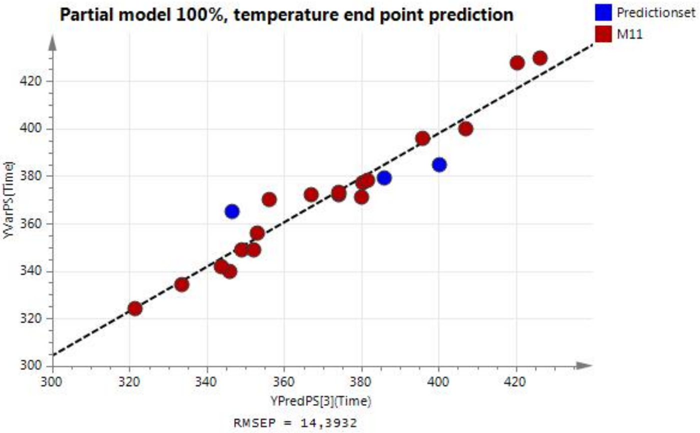


Figure 4.7 Visualization of partial model for Product A, 100% prediction end point. All the data points was used in the model. Predicted values on x-axis and real values on y-axis, where the red dots are from the estimation and the blue are for prediction, where *RMSEP* describes the prediction error.

To get a more robust model, only dependent on two variables, the same end point prediction procedure was done with the reduced static model. The results can be seen in Table 4.8.

Table 4.8 Partial models of end point time prediction, with heavily reduced static model, with BLM. Standard deviation σ of batch end time was 43.9 minutes. Three components used to describe the X matrix. Root-mean-square error of prediction in minutes of three batches used as prediction set where the mean length of a batch is 374 minutes.

Time	*RMSEP*	R^2X	R^2Y	Q^2
20%	14.10	0.859	0.829	0.469
40%	10.58	0.858	0.879	0.573
60%	12.13	0.834	0.909	0.684
80%	15.99	0.821	0.964	0.573
100%	13.32	0.812	0.981	0.913

To visualize what is causing the increase in RMSEP at 80% in Table 4.7 and 4.8, the first score vector for three batches was computed. See the Discussion chapter for more info.

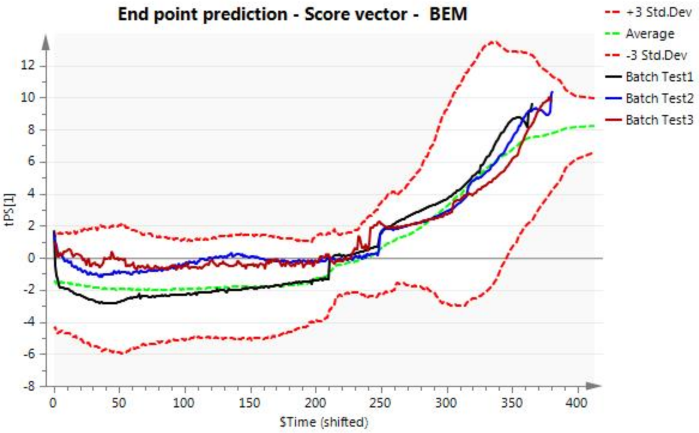


Figure 4.8 Three validation batches with the corresponding first score vector for each batch for Product A.

4.1.2 Product B

Pre-processing

A total of 43 batches were evaluated with the BSPC concept. Observations, without any removal of outliers or variables can be seen in Figure 4.9. R^2X,R^2Y and Q^2 values of the raw data can be seen in Table 4.9.

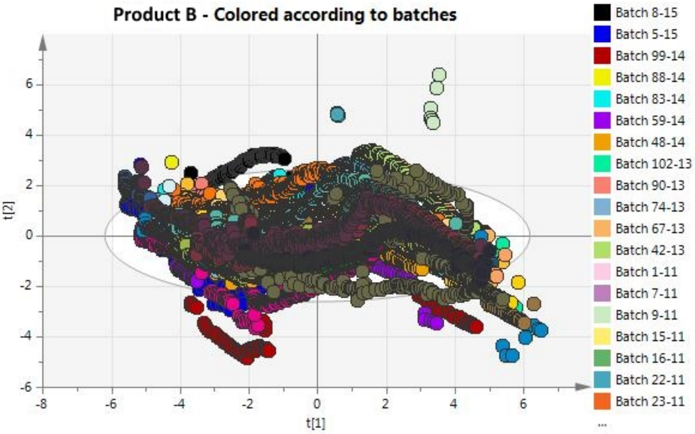


Figure 4.9 PLS score scatter plot to detect outliers and such, with the raw data as X and the time as Y, for all Product B batches.

The different Product B batch data sets were much smoother, compared to Product A, without a large set of outliers. The data points that were removed came from excluding a few points on variable level, and were confirmed by using contribution plots (spikes in the variables). The final R^2X, R^2Y and Q^2 values for the model, after removing any unwanted behavior can be seen in Table 4.9.

Table 4.9 PLS on raw data of Product B as X and time as Y, before and after removing any outliers. R^2X, R^2Y and Q^2 describes how good the model estimate Y and X.

Removal	**Components**	R^2X	R^2Y	Q^2
Before	6	0.857	0.805	0.805
After	6	0.861	0.812	0.812

Estimation of product purity - None hierarchical modeling

Estimation of product purity was done, according to BLM methodology described in Section 3.3.2. The total number of batches used when deriving the BLM was

35, with 7 batches used as a validation set. The initial product composition was not known. Any constituents with a negative Q^2 value was removed, and the resulting Y matrix contained six different product quality attributes for each batch. As seen in Figure 4.10, the loading scatter plot suggests that three different models are created. The models paired together are XIV and X, V and VI, as well as XIII and II. The reason for pairing XIII and II is because of their negative correlation towards each other. Corresponding $RMSE$, R^2X, R^2Y, and Q^2 values can be seen in Table 4.10.

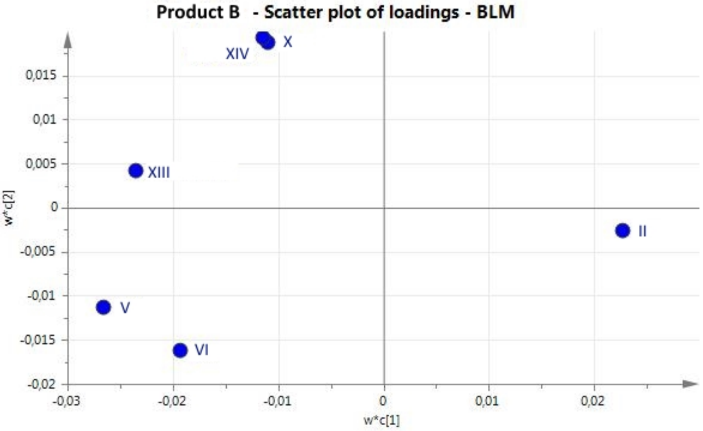


Figure 4.10 Loading scatter plot of a BLM model for Product B, with six different product attributes. Three different models were formed; 1) XIV and X, 2) V and VI and 3) XIII and II.

Table 4.10 Mean, standard deviation σ, $RMSEP$, R^2X, R^2Y and Q^2 values for different BLM models when predicting quality. The number of batches used as a prediction set were 7 and the number of batches used for training the models were 35.

Constituents	**Comp**	**Mean**	σ	$RMSEP$	R^2X	R^2Y	Q^2
II	3	69.1	5.85	1.96	0.483	0.780	0.396
V	2	0.69	0.34	0.16	0.648	0.910	0.689
VI	3	7.65	2.95	0.62	0.648	0.910	0.689
X	2	14.1	2.67	0.97	0.379	0.588	0.216
XIV	2	148.37	25.11	24.2	0.379	0.588	0.216
XIII	3	595.67	35.57	11.89	0.483	0.780	0.396

Estimation of product purity - Hierarchical modeling

The product purity was also estimated using the Z matrix, containing 22 batches. The different initial compositions was stored in the Z matrix. Three out of the 22 batches was used as validation batches. The resulting model was then divided into three models, after removing all Y variables with a negative Q^2 value. The loading plot had two distinct clusters of Y variables, and two of the variables XIII and II had a negative correlation towards each other. The different variables paired was VI and V, II and XIII, as well as XIV and X. The resulting $RMSEP$, R^2X, R^2Y and Q^2 values for the different models can be seen in Table 4.11. The loading plot can be seen in Figure 4.11 where the blue dots correspond to the different compositions and the green dots are the score vectors from the base models created between the different matrices.

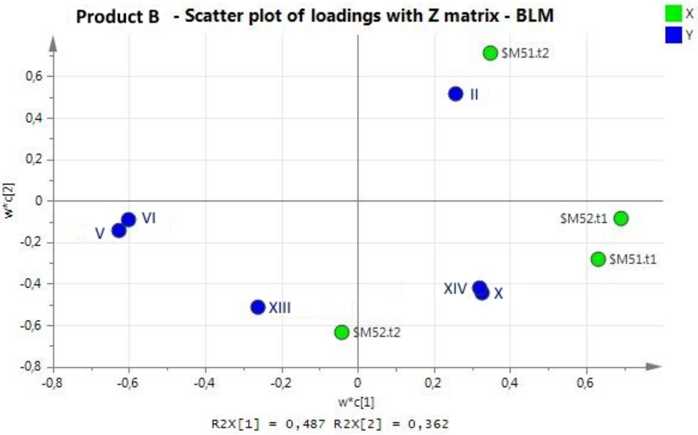


Figure 4.11 Loading scatter plot of BLM model for Product B quality with Z matrix and X matrix included. The green dots M51 is the scores from X and M52 is the scores from Z, where t1 is score one and t2 is score two. The blue dots are the different attributes. Three different models were formed; 1) XIV and X, 2) V and VI and 3) XIII and II.

Table 4.11 Mean, standard deviation σ, $RMSEP$, R^2X, R^2Y and Q^2 values for different BLM models, with Z matrix, when predicting the quality. The number of batches used as prediction set was set to 3.

Constituents	Comp	Mean	σ	$RMSEP$	R^2X	R^2Y	Q^2
II	4	69.7	4.32	2.26	1.000	0.704	0.484
V	1	0.66	0.32	0.09	0.485	0.759	0.672
VI	1	7.69	3.17	1.72	0.485	0.759	0.672
X	2	13.49	2.06	1.20	0.731	0.555	0.271
XIV	2	148.37	25.11	12.4	0.731	0.555	0.271
XIII	4	589.71	26.00	13.37	1.000	0.704	0.484

II and XIII was of a higher importance than the other product purity values, the resulting prediction can be seen in Figure 4.12 and 4.13.

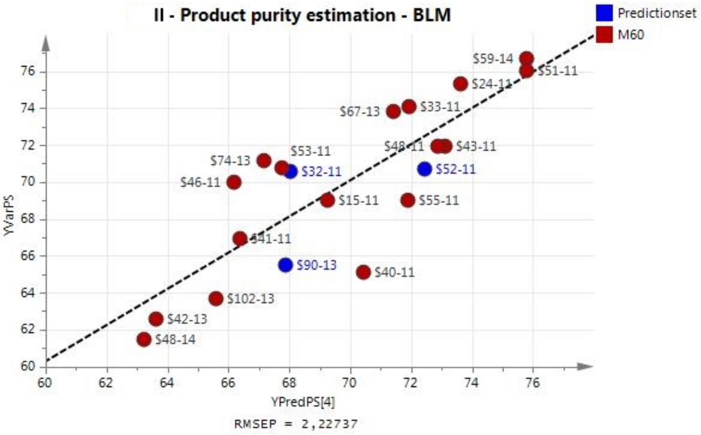


Figure 4.12 Predicted II content in Product B, using BLM. Predicted values on the x-axis and observed values on the y-axis, where the red dots represents the training set and the blue ones the prediction set.

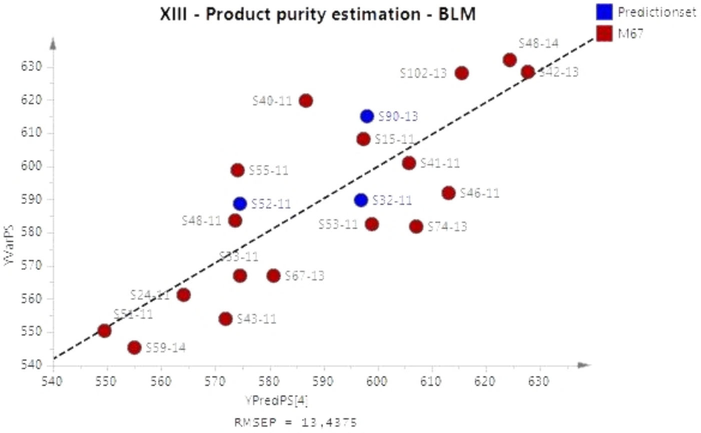


Figure 4.13 Predicted XIII content in Product B, using BLM. Predicted values on the x-axis and observed values on the y-axis, where the red dots correspond to the training set and the blue ones represents the prediction set.

The hierarchical model structure and the initial composition were used when trying to improve the prediction of the end product purity. The importance of the initial composition can be noticed by interpreting the contribution plot for the *Z* matrix when estimating a low- compared to a high XIII value. A higher amount of constituents of V and VI result in a higher XIII value. This is seen in Figure 4.14.

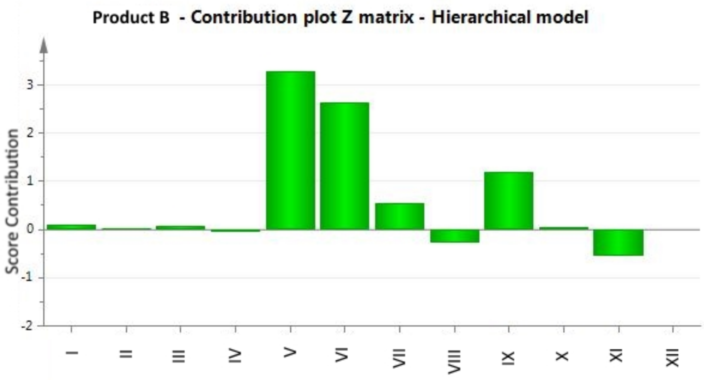


Figure 4.14 Variable weights describing the importance of initial conditions for XIII content of Product B.

Variable simulation and reduction

The operators use density, to decide when the mixture composition is within the specification range, when manufacturing Product B. Two models were derived, one static and one dynamic, to see how the different models explain the density variable. The results for the models using all variables can be seen in Table 4.12.

Table 4.12 Variance explained with PLS, for density variable. One static, and one dynamic model for the model with all variables (1-2) and the heavily reduced models (4-5). A static reduced model without the density variables (3). Number of components for all of the models were set to 3. The X and Y in column lag corresponds to which part is lagged.

No	Model	Lag	R^2X	R^2Y	Q^2
(1)	All variables	0	0.714	0.862	0.862
(2)	All variables	1 (X,Y)	0.720	0.870	0.870
(3)	Without density	0	0.759	0.681	0.681
(4)	Heavily reduced	0	1.000	0.684	0.684
(5)	Heavily reduced	1 (X,Y)	0.757	0.862	0.862

As seen in Table 4.12, the difference between a static and the dynamic models using all variables is low, almost none. Model reduction was done in order to reduce complexity.

All of the density variables were removed from the data set, since they are highly correlated to the Y variable explained. After removing the density variables from the X matrix, the resulting R^2X, R^2Y and Q^2 values can be seen in Table 4.12 and the model is shown in Figure 4.15.

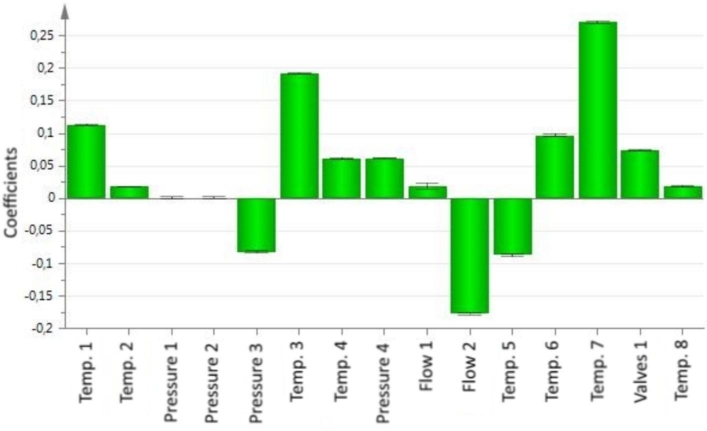


Figure 4.15 Variable weights from PLS when describing the density of Product B using BEM and all variables.

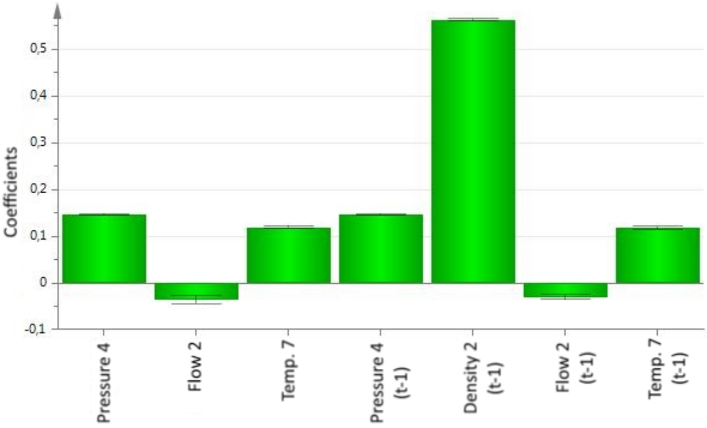


Figure 4.16 Final reduced density model of Product B. Model is dependent on a flow, a pressure, a temperature variable and a lagged variable backwards in time one step. The density described is also dependent on a lagged version of itself.

To decrease the complexity of the density model even more, and to be able to interpret and explain the model by its physical meaning, further model reduction was done. The final reduced model contained one temperature, one flow and one pressure variable. To catch the dynamics the above variables together with density were lagged backwards in time one step. The resulting R^2X, R^2Y and Q^2 values can be seen in Table 4.12 and the model is shown in Figure 4.16.

Early fault detection

The derived model, in Table 4.9 after removal of outliers, should be able to notice a deviating batch. A deviating batch was used with the derived non-reduced density model, and the first score vector and the residuals (distance to model) can be seen in Figure 4.17 and 4.18. A contribution plot was used, to visualize the reason for the deviating behavior. This can be seen in Figure 4.19 and is further debated in the Discussion chapter.

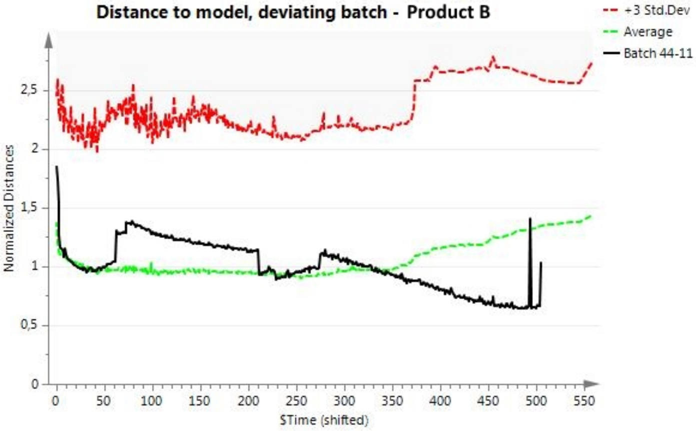


Figure 4.17 Distance to the non-reduced model with time as Y of Product B for a deviating batch, in order to detect a fault. The green line is the average and the red is ±3 standard deviation from the density model of all batches.

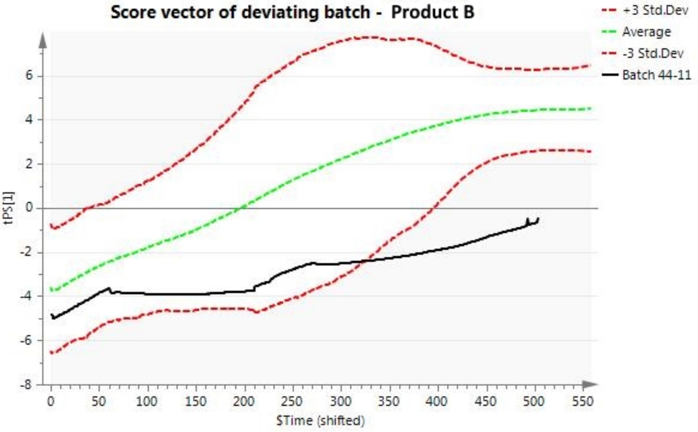


Figure 4.18 First score vector from the deviating batch. The green line is the average and the red is ±3 standard deviation from the score vectors of all batches.

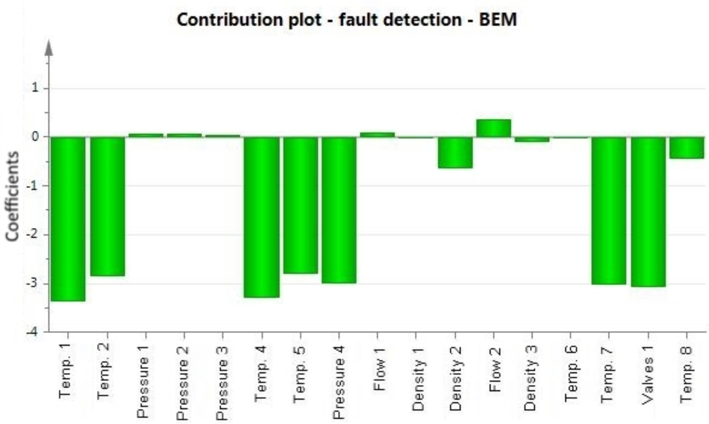


Figure 4.19 Contribution plot visualizing which of the variables that is causing the deviation.

Prediction of inflection

The density is used as an inflection point, when deciding batch termination. The batch time when the density variable has reached its specifications was saved, and used together with the non-reduced density model (1) in Table 4.12, to create a BLM. Partial models were created, to visualize how well the model could calculate an estimated end-point, depending on how long the batch has gone. A total of 38 batches was used, to derive the models. The results can be seen in Table 4.13 and in Figure 4.20 the predicted values are plotted against the observed values.

Table 4.13 Partial models of end point time prediction, using all variables, with BLM. Four components used to describe X. Standard deviation σ was 99.45 minutes. Root-mean-square error of prediction in minutes of four batches used as prediction set where the mean length of a batch is 380 minutes.

Time	*RMSEP*	R^2X	R^2Y	Q^2
20%	38.17	0.766	0.865	0.562
40%	33.94	0.716	0.908	0.752
60%	35.39	0.631	0.937	0.796
80%	43.58	0.605	0.962	0.845
100%	15.38	0.609	0.979	0.905

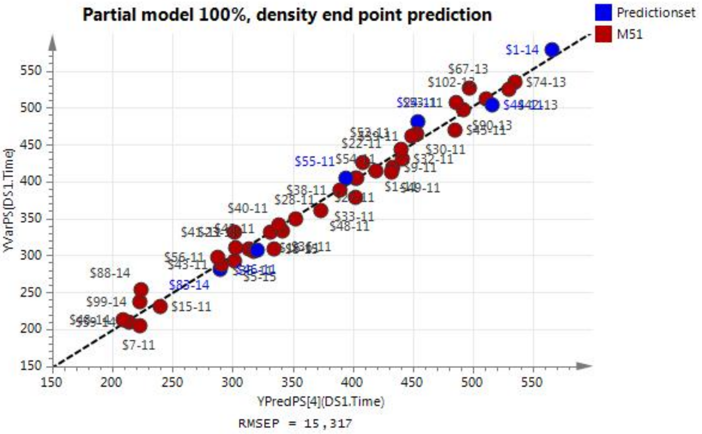


Figure 4.20 Visualization of partial model for Product B, 100% prediction end point. Predicted values on the x-axis and observed values on the y-axis, where the red dots represent the training set and the blue ones the prediction set. *RMSEP* describes the prediction error.

To see if it was possible to derive a more robust model when predicting batch termination time, the reduced static density model (4) in Table 4.12 for Product B was used. The results can be seen in Table 4.14. To visualize what is causing the increase in RMSEP at 60% and 80% in Table 4.13 and 4.14, the temperature for five batches was plotted in Figure 4.21.

Table 4.14 Partial models of end point time prediction, with reduced density model, for Product B, with BLM. Four components used to describe X. Standard deviation σ was 99.45 minutes. Root-mean-square error in minutes of four batches used as prediction set where the mean length of a batch is 380 minutes.

Time	$RMSEP$	R^2X	R^2Y	Q^2
20%	48.60	0.646	0.695	0.931
40%	42.74	0.727	0.820	0.599
60%	46.23	0.819	0.859	0.707
80%	43.03	0.683	0.921	0.776
100%	31.34	0.672	0.979	0.931

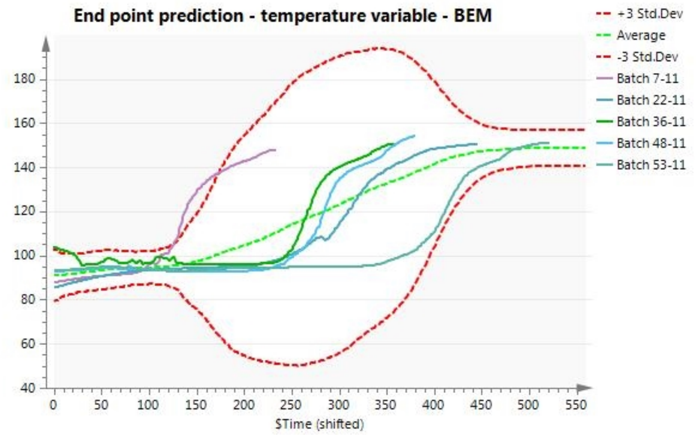


Figure 4.21 Temperature variable for five batches to see how the temperature varies over time for Product B.

4.2 System identifcation

4.2.1 Product A

Pre-processing

Before deriving any models. Some issues had to be dealt with in regards of pre-processing. One of the first problems encountered when deriving a model, was that the batches were not of equal length. This was solved by merging all the data sets. To ensure that the merged data sets were trimmed according to the valve values, as described in Section 3.1, an if statement was used. In terms of pre-processing, this was the only thing that was done to the data.

Estimation of product purity

The least-squares estimation was done in three different ways. The first way, done with a non-reduced model, as in Section 2.1.1, with the estimation technique described in Section 2.1.1. The resulting parameters for each of the variables is shown in Figure 4.22. The product purity estimation, when using the time average, can be seen in Figure 4.23. The first 17 batches were used for estimation and model derivation, and the last three for validation.

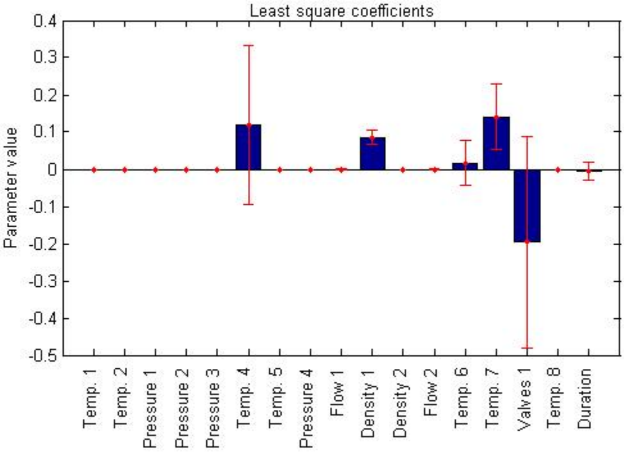


Figure 4.22 The estimated parameters from least square, with a 95% confidence interval, using the time average of the Product A batches and the quality.

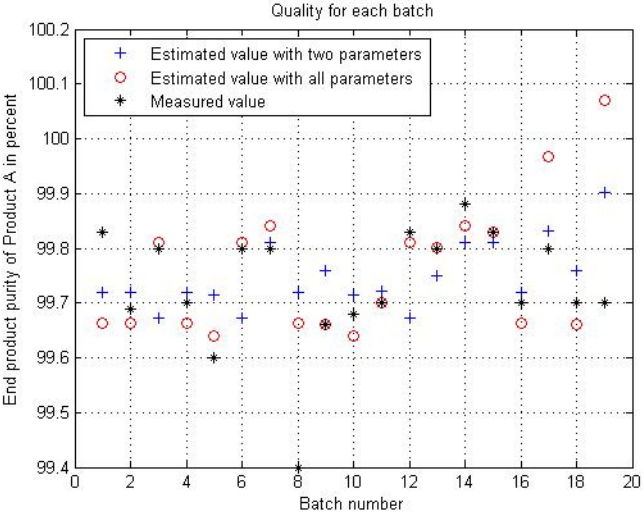


Figure 4.23 The estimated product quality for the two least square models using the time average of the batch and the measured quality. The circles are the estimated value using all the parameters and the plus is when only using two parameters. The star corresponds to the measured value. The last three batches are validation batches.

The second way, was done by using the two most significant parameters. The two most significant parameters can be seen in Figure 4.24. The resulting product quality estimation, using the reduced model, is shown in Figure 4.23. The residuals for the two different model approximations can be seen in Figure 4.25. The last three batches in Figure 4.25 were validation batches.

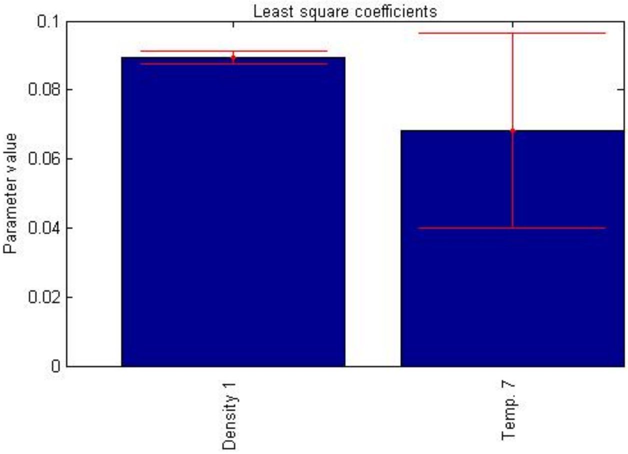


Figure 4.24 The two estimated parameters from least-squares estimation, with a 95% confidence interval, using the time average of the Product A batches and the quality.

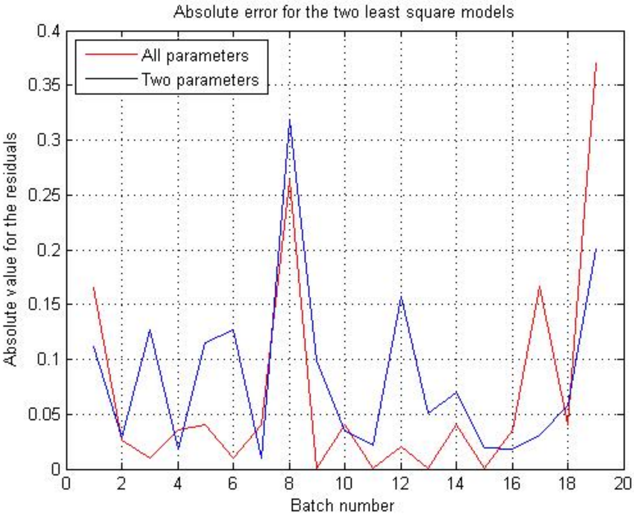


Figure 4.25 The residuals for the least-squares models with all parameters and two parameters. The red is the residuals using all parameters and the blue is when only using two.

The third was to take the whole X matrix, and unfold it as in Figure 2.2. The numbers of parameters were now equal to the numbers of variables times the length of the batches. The estimated parameters are shown in Figure 4.26.

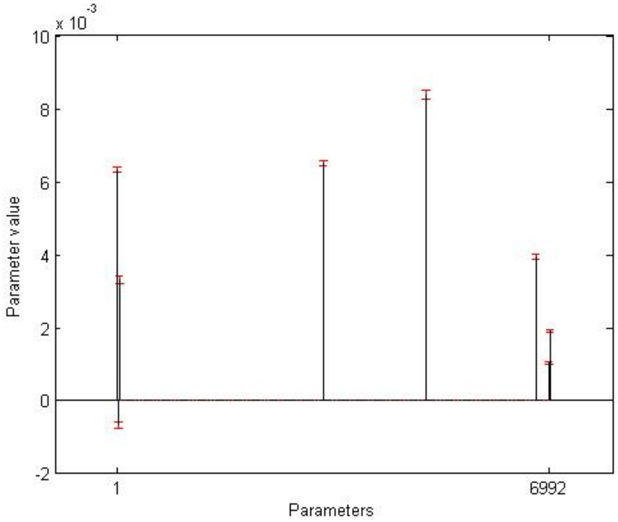


Figure 4.26 The estimated parameters from least-squares estimation, with a 95% confidence interval, using all the observations of the Product A batches and the quality.

The non-zero parameters are displayed in Figure 4.27 and the estimated product purity is seen in Figure 4.28.

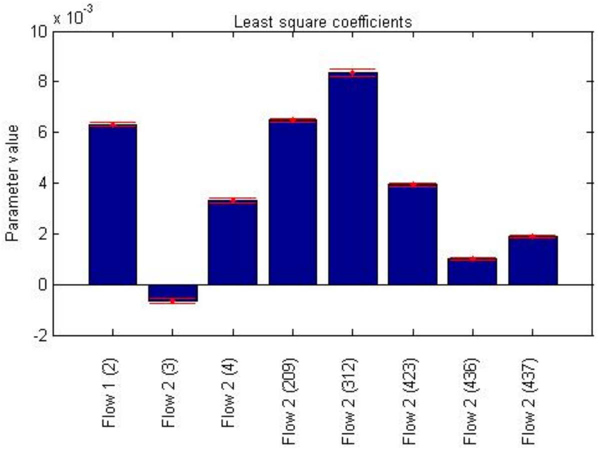


Figure 4.27 The estimated parameters from least-squares estimation, using all the observations of the Product A batches and the quality, that are none zeros, with a 95% confidence interval. The parenthesis in the variable name corresponds to what time the observation occurred.

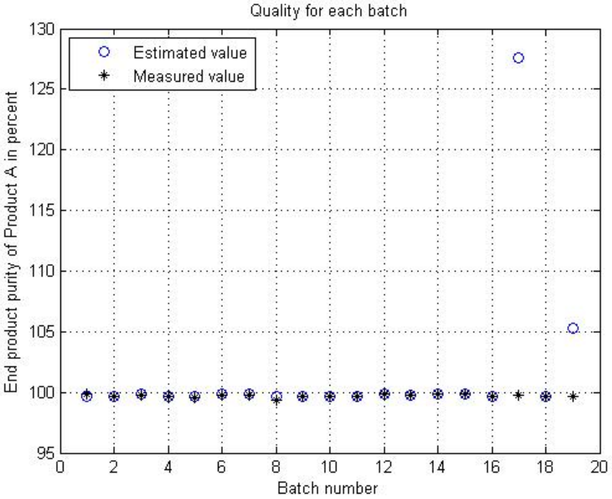


Figure 4.28 The estimated product quality for the least-squares model using all observations for the Product A batch and the measured quality. The circles are the estimated value and the starts correspond to the measured value. The last three batches are validation batches.

k-step prediction of temperature

The reference points that are used to terminate the batch are the temperature variable. The temperature model was created as in Section 3.4.1. Since there is no direct feed-through between the input and the output, the D matrix was set to zero. A differentiated Kalman filter, as described in Section 2.5, was used when trying to predict the temperature. The model order, was chosen dependent on the singular values. The singular values are seen in Figure 4.29. The R^2X values of the different model orders are seen in Table 4.15.

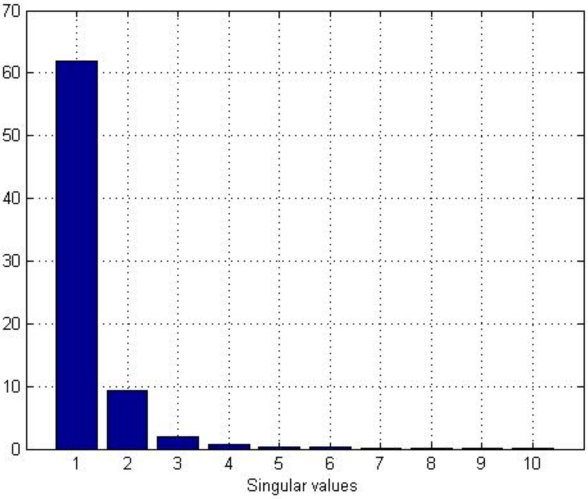


Figure 4.29 The singular values from all the Product A batches.

Table 4.15 Regression value for the state-space model, built on the temperature for Product A, of order 1, 2 and 3.

Model order	R^2X
1	93.38
2	98.01
3	97.61

A simulation of a model of second order can be seen in Figure 4.30. The different model parameters are displayed in Figure 4.31-4.33

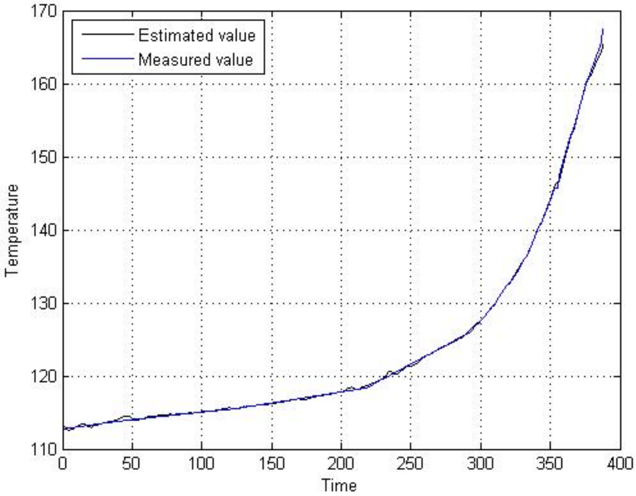


Figure 4.30 Simulation of the temperature for the second-order model and the measured temperature. The black line is the estimated temperature and the blue is the measured value.

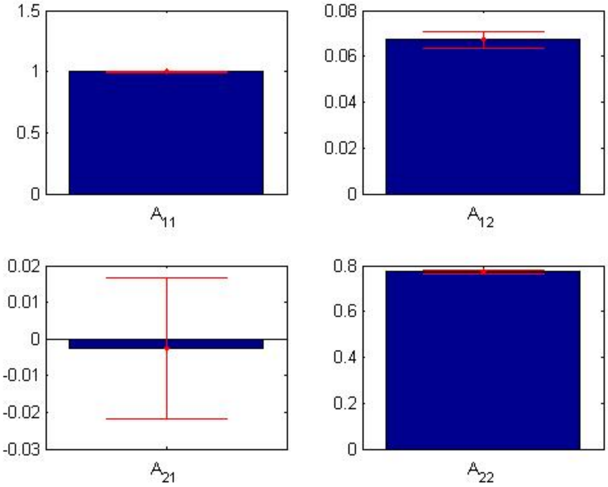


Figure 4.31 The estimated *A* parameters,using MOESP with the temperature and data from the Product A batches, with a 95% confidence interval.

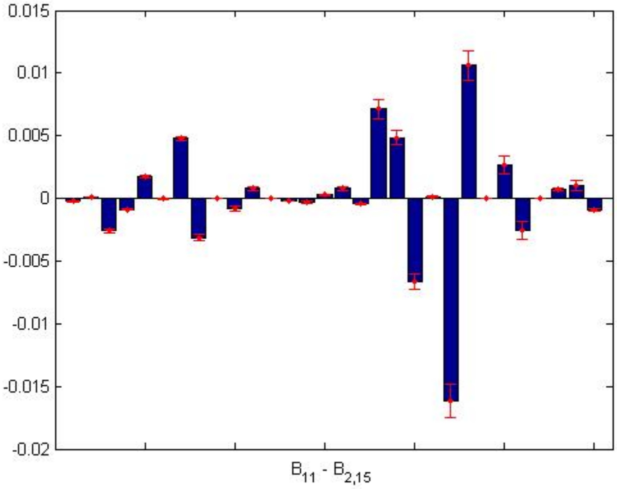


Figure 4.32 The estimated *B* parameters,using MOESP with the temperature and data from the Product A batches, with a 95% confidence interval.

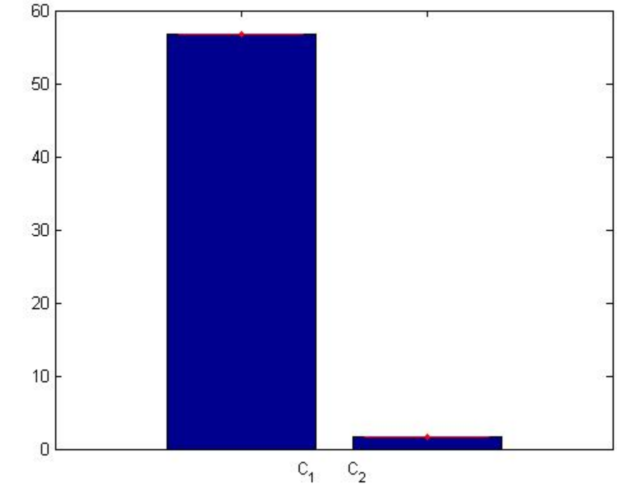


Figure 4.33 The estimated *B* parameters, using MOESP with the temperature and data from the Product A batches, with a 95% confidence interval.

The resulting model was then used when trying to predict the temperature. The differentiated Kalman filter prediction is seen in Figure 4.34. The R^2 values of the different k-step predictions are displayed in Figure 4.35.

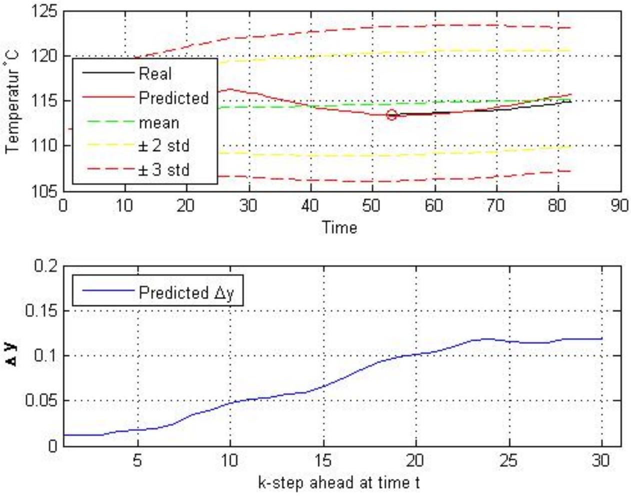


Figure 4.34 30-steps prediction of the temperature and the temperature difference. The black line is the real value, the red is the predicted, the green is the mean, the yellow is ±2 standard deviation and the red is ±3 standard deviation. The red circle is at time t. The blue line is the predicted temperature difference.

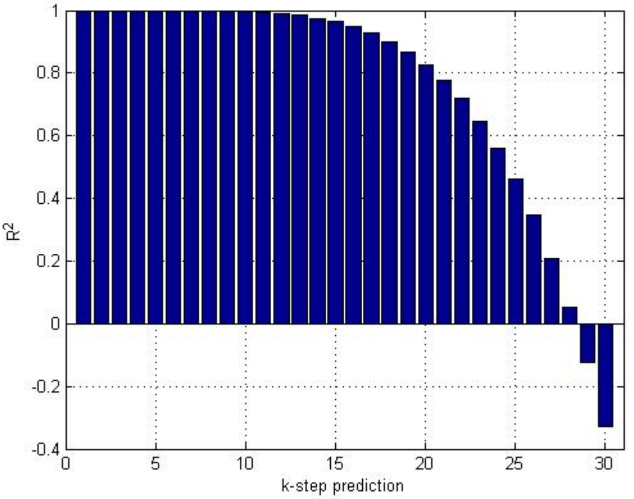


Figure 4.35 R^2 values for each prediction of the temperature.

4.2.2 Product B

Pre-processing

The same problem, regarding different batch lengths, was solved in the same way for Product B as it was done for Product A. By merging the different batch data sets to one object, and then trimming the data in regards of valve values, it was possible to derive a model. The Product B data set had only a few outliers on variable level. These were kept, even though the outliers might disrupt the model.

Estimation of product purity

The least-squares analysis was done in the same way as for the Product A by taking the time average and taking all the data points. The resulting parameters, using the mean value and the time average, to predict the II content in Product B is shown in Figure 4.36. The II content estimated by the parameters is shown in Figure 4.37.

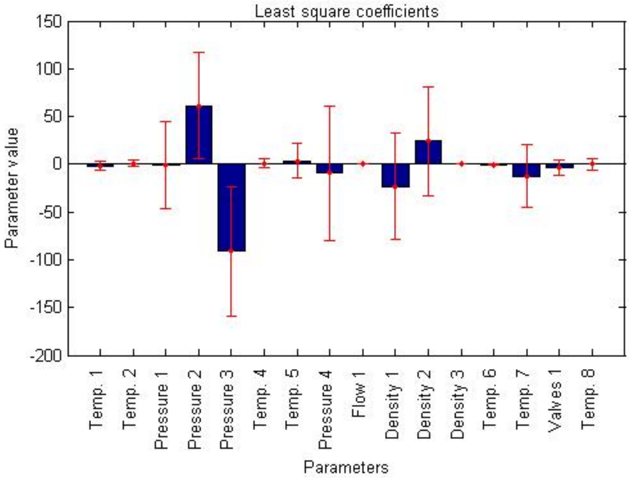


Figure 4.36 The estimated parameters, with a 95% confidence interval, using the II content in Product B and the time average of the Product B batches.

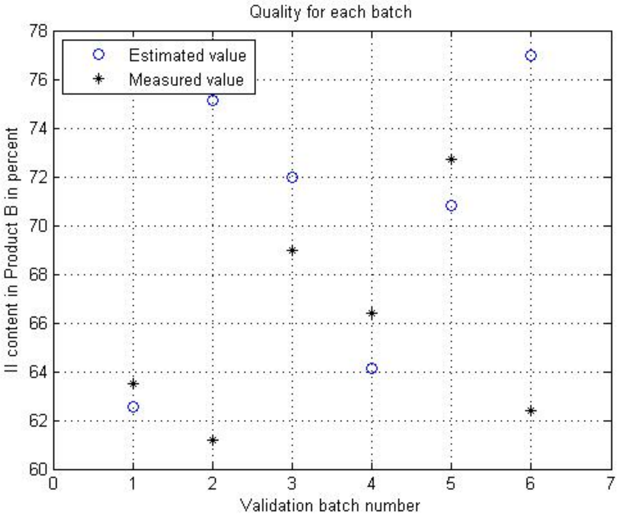


Figure 4.37 Estimated II content in Product B, with time average of six validation batches. The circles are the estimated value and the start is the measured value.

The parameter estimation, for all the different contents, was done as above with the same results. In Table 4.16 one can see the mean residuals from the least-squares models when trying to estimate the different contents in relation to the mean value of each substance. The mean residuals are from the six validation batches used to try to estimate the quality.

Table 4.16 Mean value and mean residuals for six validation batches using the least-squares models.

Constituents	**Mean**	σ	**Residual**
I	5.1829	2.3491	0.9682
II	68.9496	5.8590	4.3947
III	0.1791	0.1665	0.1826
IV	0.0144	0.0173	0.0256
V	0.6807	0.3386	0.0682
VI	7.6138	2.9536	1.3987
VII	0.7367	0.5490	0.2611
VIII	0.0107	0.0031	0.0039
IX	2.4244	2.1944	2.9705
X	14.1813	2.6745	3.8823
XI	0.0162	0.0183	0.0300
XII	0.0102	0.0014	0.0004
XIII	593.0570	35.5676	32.5121

k-step prediction of temperature

For Product B, the operator uses the density to decide when to terminate the batch. The model, that will then be used to predict the density, will be created as described in Section 3.4.1. The model, together with a Kalman filter, will then predict the density. The order of the model is chosen as the singular values and how good the model fit to the output, which is shown in Figure 4.38 and Table 4.17.

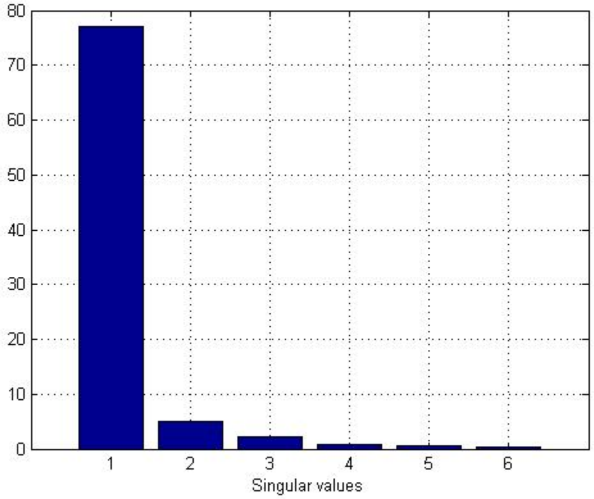


Figure 4.38 Singular values for all Product B batches.

Table 4.17 Regression value for the state-space model, built on the density for Product B, of order 1, 2 and 3.

Model order	R^2X
1	59.50
2	88.16
3	94.58

The simulation of the model, that will be used to predict the density, is shown in Figure 4.39 and the model is shown in Figure 4.40-4.42.

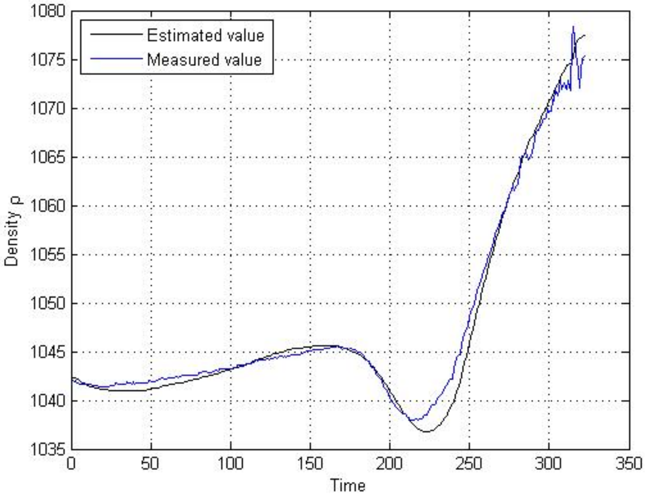


Figure 4.39 Simulation of the density for the second order model and the measured density. The black line is the estimated value and the blue is the measured value.

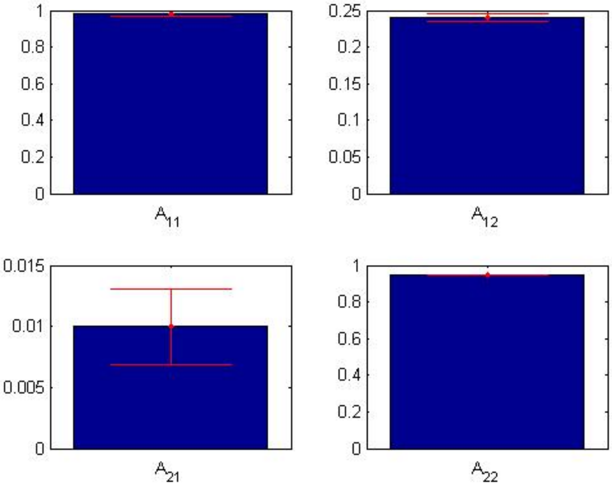


Figure 4.40 The estimated *A* parameters, using MOESP with the density and data from the Product B batches, with a 95% confidence interval.

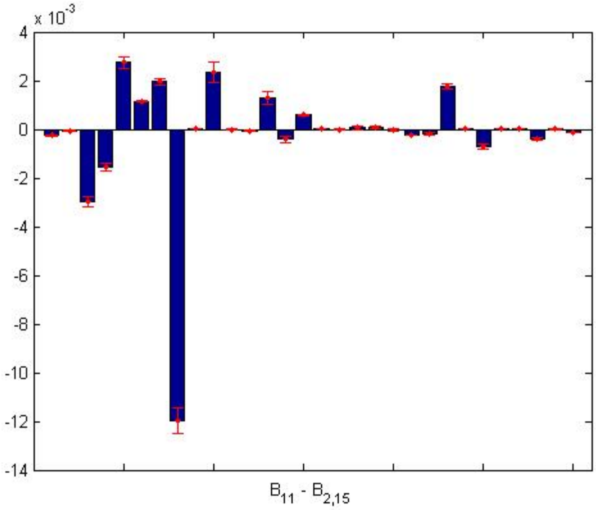


Figure 4.41 The estimated *B* parameters, using MOESP with the density and data from the Product B batches, with a 95% confidence interval.

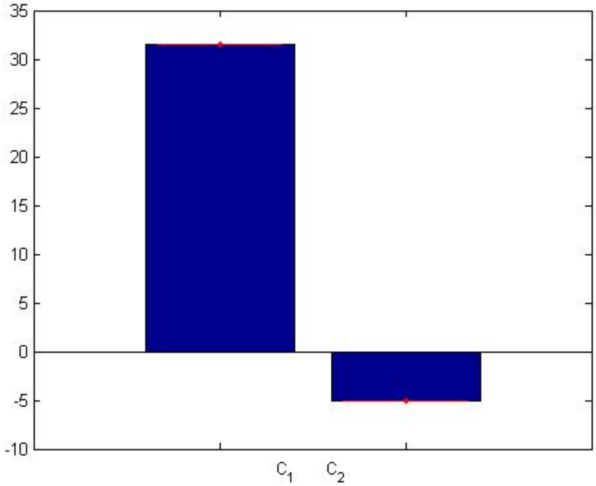


Figure 4.42 The estimated *C* parameters, using MOESP with the density and data from the Product B batches, with a 95% confidence interval.

This model was then used to try to predict the density k-step ahead. Figure 4.43 shows the prediction 45-steps ahead and in Figure 4.44 the resulting regression values are shown for each prediction step.

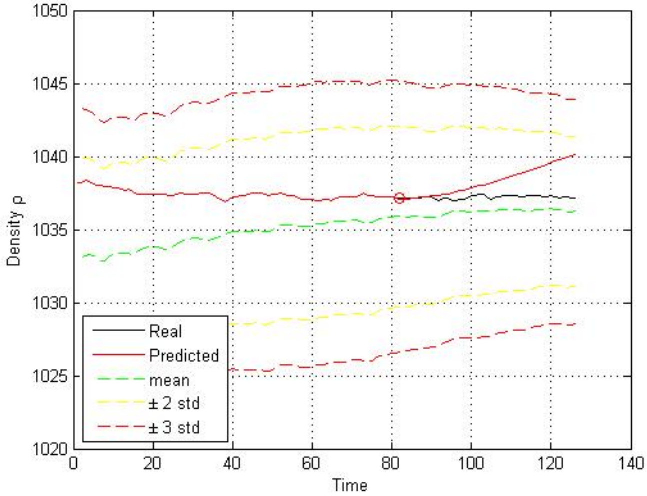


Figure 4.43 45-steps prediction of the density. The black line is the real value, the red is the predicted, the green is the mean, the yellow is ±2 standard deviation and the red is ±3 standard deviation. The red circle is at time t. The black line is the predicted temperature difference.

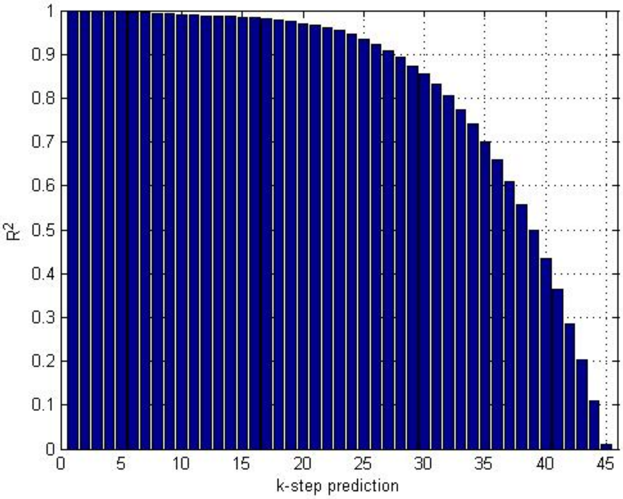


Figure 4.44 R^2 values for each prediction of the density.

5

Discussion

The Discussion Chapter has been divided into two sections, based on the two methodologies examined. The methodology sections have been further divided into Product A and Product B.

5.1 Batch statistical process control

One vital part, when deriving all of the models in the project is to stop and reflect to see if the derived model is sufficient to describe the desired behavior. If this is not done, the resulting models might become to complex, and hard to interpret. When working with the BSPC concept, it was quite easy to pursue a R^2 value very close to one. When analyzing and working on a model, it was very easy to overexclude data points, to raise the R^2 value, since if the R^2 value is one, the model describes the data set perfectly.

5.1.1 Product A

Pre-processing

When searching for abnormal behavior, in regards of outliers and deviating data points, one has to interpret the reason for the deviation before excluding the data points. By using a contribution plot, it was possible to find the reason for the abnormal behavior. If not extra precaution is taken, it is possible to remove important behavior in the data set. When the models were derived, some important information might have been excluded and thus losing vital parts of the process. Since Product A got only a few constituents, only two components was used to describe the behavior in the data set. Even though the R^2X value increased when excluding some of the data points, the resulting model might not have been more accurate than before the data exclusion. This was kept to a minimum by having a continuous discussion when interpreting the data. When comparing the subplots in Figure 4.1, one notice that the data set appears to be a lot smoother, after the obvious outliers had been excluded. Since the process is a batch process, and there are many different unit operations, one of the possible reasons for the deviating behavior might be that the batch phase examined has not yet started, even though it should have.

As seen in Table 4.1, the R^2X value is 0.531, when the raw data were examined with PLS regression. After the removal of outliers and unwanted behavior, the model is able to explain 20% more of the data set. Since the data points removal was done with caution, the 20% gained is considered as a positive result.

Estimation of product purity

The main reason for not finding any correlation pattern between the X matrix, and the product purity end points in the Y matrix, is thought to be the lack of variability within the product purity data set. If one examines the product purity data set, it varies in the first and second decimal. Since the X matrix varies a lot, but the Y matrix varies little to none, the Q^2 values in Tables 4.2 and 4.3, is thought of as noise and not a correlation pattern.

Another reason might be that the resulting models is trying to find a correlation pattern between a large X matrix, containing all of the batch data sets, and a product purity end-point for each batch. If more product purity data points were available during the course of a batch, it might have been possible to find a correlation pattern between the process data and the product purity values.

Variable simulation and reduction

As seen in Table 4.4, the temperature variable was the easiest one to describe of the different variables used as a break point, when deciding batch termination. Since the temperature is used as the Y matrix, all of the other temperature variables were excluded from the X matrix, to see if the R^2Y value decreased. As seen in Table 4.5, the possibility to describe the temperature with the two static models decreased with about 5%. The resulting R^2Y value of 0.906 is considered good, even though the other temperatures were excluded. The conclusion drawn from this, is that the temperature is not only dependent on other temperature variables during the course of a batch, and it is possible to describe the temperature with properties that describes the physical state of the process within the batch process at the same time lag.

To be able to interpret and describe the model from its physical properties, a strongly reduced model was computed. The final model was derived both as a static, and as a dynamic model that had the Y variable lagged once. The temperature was dependent on a flow and a pressure variable. As seen in Table 4.6, the difference between a static, and a dynamic model is about 5%. By deriving a dynamic model, it is possible to gain the information lost when reducing the complex static model.

Early fault detection

The early fault detection model was supposed to recognize a deviating batch, that was not in the model derivation data set. As seen in Figure 4.4, the data set starts to deviate after 300 minutes. If one look at the score vector in Figure 4.5, the deviation is not that clear, until a distinct drop at around 370 minutes. Since the score vector is within ±3 standard deviation between the 300- and 370 minute mark, it is not possible to notice a deviating batch just by looking at the corresponding score vector. By using a combination of distance to model, and the score vector of a batch, it is possible to notice a deviating batch with ease.

The deviating batch had a different chemical composition, compared to the composition of a regular batch, with a higher concentration of heavy chemicals. The increased amount of heavy chemicals required a higher evaporation temperature and this is what mainly causes the deviation. But there are also other variables that contribute, like for instance valves 1. The deviation can be seen in Figure 4.6. The conclusion drawn from this, is that the model derived for early fault detection handles and recognizes if a batch differs from a mean batch.

Prediction of inflection

Since the temperature is one of the break points used when deciding batch termination, a time end point prediction was done, to be able to give the operators a hint when batch termination is due to. As seen in Table 4.7, the root-mean-square error of three prediction batches is around 20 minutes. The RMSEP is 20 minutes even though only 20% of a batch has been completed. The conclusion drawn from this, is that the model is able to estimate a time end point for batch completion quite well independent of how long the batch has gone.

The heavily reduced static model was also used when trying to estimate the termination time. The RMSEP value was 5-7 minutes lower than the non-reduced static model. The reason might be that the non-reduced model is overfitted, when trying to estimate an end-point. Since the variation in regards of how the batch is run is very small, and due to that the model has to many variables to take into account.

One thing to notice, and keep in mind, is at the 80% mark, the RMSEP for both the derived models increase. The reason for this is visualized in Figure 4.8 where it can be seen that up to 60%, corresponding to 225 min (based on average batch duration) not much seems to happen. After 60% the first score vector increases, which causes increased variation and is reflected in higher RMSEP at 80% (300 min).

5.1.2 Product B

Pre-processing

When examining and searching for abnormal behavior in the data set, the score scatter plot was examined. The reason for choosing 6 components to describe the variation in the data set, is because Product B is a more complex blend with a higher number of constituents compared to Product A. As seen in Figure 4.9, the score scatter plot contained fewer obvious outliers than the Product A data set. By examining each variable by itself together with the corresponding contribution vector, a few outliers were excluded. The reason for the lack of outliers, is thought to be that there is a larger variation, including more noise, in the higher dimensions for Product B than in Product A.

Estimation of product purity

The first concept used was a BLM with the product attributes as the Y matrix. The different attributes with a negative Q^2 value was removed, due to a low variation between the values, or that the values were of such a low numerical value that they were insignificant. After the different attributes with a negative Q^2 value was removed, the total Q^2 value for the resulting model was raised from a negative value to a positive value.

By analyzing the loading scatter plot in Figure 4.10, one notice that the different Y variables got different weights and some of the variables are correlated to each other. By splitting the BLM model into three submodels, the hope was to raise the Q^2 value, thus finding a better correlation pattern between the matrices. As seen in Table 4.10, the resulting models are able to find a correlation pattern between the X matrix, and the different product attributes. The different models were able to reduce the standard deviation of the different attributes by about half, when comparing to the RMSEP value.

By introducing a new matrix Z, the BLM procedure was done once again. As seen in 4.11, the resulting correlation pattern, for the different models, is quite similar to the BLM model computed without the Z matrix.

One thing to keep in mind when drawing conclusions between the different model structures is that the number of batches used when validating the models is different, and the standard deviation σ of the different attributes are not the same. The number of batches used for validation in the first BLM model was 7, and the number of validation batches used in the BLM model with the Z matrix was three. With this in mind, it is not possible to tell which of the procedures that describes the attributes best, due to the difference in standard deviation σ.

The upside of using an hierarchical structure, is that it is possible to use the contribution plots from the different base models, to see where the variation and deviation originate from. If the deviation originates from the initial composition matrix Z it is possible to see which of the initial product constituents that is causing the deviation. In Figure 4.14 the contribution from Z is shown, and it is possible to see which of the product constituents that is causing the deviation. This implies that it is possible to increase constituents V and VI, to get a higher XIII value.

The downside of hierarchical modeling is that the link between the base models and the top model is described with score vectors. If the score vectors does not describe enough variability in the data set, the resulting product quality estimation in the top model might become poor.

Variable simulation and reduction

As said before, the operators use density, to decide when the current mixture composition is within specification range. As seen in Table 4.12, the difference between a static (1) and the dynamic (2) models when using all variables are low, almost none. The conclusion drawn from this, is that the model is able to explain the density by using only variables at the same time lag. One thing to keep in mind though, is that there are other density variables that help explaining the density, due to high correlation. Also seen in Table 4.12, the resulting R^2Y and Q^2 values decrease with about 20%, when the other density variables were excluded. This implies that the excluded density variables helped a lot when trying to explain the density as the Y variable.

To decrease the complexity of the density model, a static model (4) and a dynamic model (5) was derived. As seen in Table 4.12, the difference between a static and a dynamic model is about 20%. This implies that the dynamic model of order one, collects about 20% of its information in time lag t-1. The conclusion drawn from this, is that the density variable is harder to explain and requires a dynamic model with the variables and itself lagged.

Early fault detection

The model derived was tested, in order to see how good it detects deviating batches. In Figure 4.18, one can see how the first score vector from the deviating batch starts to drift from the mean of the score vectors after about 300 minutes, but in Figure 4.17 the distance to model is still good. By comparing the two fault detection models for the different products, the argument stated in the Product A section, that a combination of both looking at the score vector and the distance to model is correct, as seen in Figure 3.2 where both D and Q is shown.

According to the contribution plot in Figure 4.19 the deviating batch was run at lower temperature which was indicated by several temperature variables. Also Pressure 4 and Valves 1 were different.

Prediction of inflection

The prediction of end point was based on reaching a certain density threshold. As seen in Table 4.13, the RMSEP value, when a batch has gone 20% of its total length, is about 38 minutes. By comparing this to the mean length of a batch, which is about 380 minutes, the model is able to predict the time end point quite well. When the batch has gone 40% of its total length, the RMSE value has gone down to about 27 minutes. By comparing the standard deviation σ, that was 99.45 minutes, the derived model is considered very good. One thing to notice, is that the RMSEP values in Table 4.13 and 4.14 increase around 60% and 80%. The reason for this was that the variation was higher in the data set at the end of a batch. At 60%, corresponding to 228 min based on a average batch duration, the temperature increase is just about to start. At 80% (304 min) the first score vector increases quite rapidly. This trend is also valid for batches with short and long batch duration. See Figure 4.21.

To see if it was possible to decrease the RMSEP value, the same end point time prediction procedure was done, with the heavily reduced static model. As seen in Table 4.14, the RMSEP value for the heavily reduced model is much higher than in the non-reduced model. This implies, that the model needs all of the different variables to estimate the time end point.

5.2 System identification

5.2.1 Product A

Since the model derived, in means of system identification, was computed in Matlabs environment, much of the work was done as functions and scripts. Most of the scripts, and functions are hard-coded from scratch. With this in mind, the functions and scripts used in the model derivations might contain some small errors. The resulting models might result in good estimations, even though the coding contain small error. By continuous interpretation and discussion regarding the hard-coding, the script and function errors were kept to a minimum.

Pre-processing

Not removing any outliers can be a risky thing to do, but when computing the model, the confidence interval of the different parameters estimated was very low. This implies that if there were any outliers in the data set, the derived model did not have a problem to describe them. One way of excluding outliers in the data set, would have been to remove them, and running the data set through a Kalman filter to get an estimated value, instead of the outlier. This was something that was thought of but not done in this project.

Estimation of product purity

The least-squares regression analysis done in Section 4.2.1, is thought of as inconclusive, since the difference between the estimated, and the real product quality values differ too much between the batches. Since the specification for Product A is very strict, the estimations must be very precise. The estimated quality might be considered good enough and the results might be used, if the company wants a very rough estimation of what the product purity value is from a certain batch. Since the parameters computed are depending on how many batches there are in the data set, the method used might become a lot better, if more batch data sets were available.

When discussing the least-squares regression in Section 4.2.1 and the resulting estimations in Figures 4.23 and 4.28, is that the estimated product purity values for the first seventeen batches were included in the model derivation data set, and the three last batches were used for validation. As one can see in Figure 4.23 and 4.28, the model that was computed with the time average give a better estimation than using all of the values. However, the estimated parameters in Figure 4.22 has a very large confidence interval, and is therefore not considered trustworthy.

In Figure 4.22, there were two variables that had a smaller confidence interval than the rest of the parameters. The procedure was done once again, with these two parameters, and the resulting residuals can be seen in Figure 4.25. By comparing the validation batches with each other, it is possible to notice that the procedure that used only two parameters is better, than using all of the parameters. The procedure using two parameters only is a more robust model, but still not considered good enough in regards of product purity estimation due to the strict Product A specifications.

It is hard to tell whether the first or the second model is any good, since the different parameters in the first model have a big confidence interval and the second model does not use all of the information from the time averaged variables. If some important information happened in the variables that were excluded in the second model, the resulting model will not produce a correct estimation.

As seen in Figure 4.26, there are only a few parameters contributing to the product purity estimation. By examining the different parameters estimated in Figure 4.27, one may notice that the whole contribution is coming from a single physical property. With this in mind, the idea is either that it is possible to estimate the product purity with the flow variables alone, or that the flow variables are large numerical numbers, as compared to the other variables, and this results in an error in the parameter estimation. To rule out the first idea, the estimation procedure was done once again, with the flow variables alone. This was inconclusive, and is not described any further in the report.

Linear regression analysis, combined with the least-squares estimation, is a good method to find a relationship with an input- and output observation. Since the product purity estimation is thought of as inconclusive, a k-step predictor was implemented, to see if it was possible to predict a break-point variable.

k-step prediction of temperature

Since the product quality estimation was inconclusive, the project focus was directed to temperature inflection point prediction. This was done with an estimated state space model and a Kalman filter. The Kalman filter is described more in detail in Section 2.5. The feed-through matrix D in the state space was set to zero, since the input signal did not have a direct influence on the output. The order of the model was set to two, due to the model indication seen in Table 4.15. A simulation of the estimated model is seen in Figure 4.30. The estimate parameters, used for the Kalman filter is seen in Figure 4.31-4.33. In these figures, it is possible to notice that almost all the estimated parameters are good, in terms of confidence interval, except A_{21}. By testing the resulting model, it was decided that estimated parameter did not influence the model in a negative way, and was still used.

The Kalman-filter prediction of the temperature, seen in Figure 4.34, works descent up to 20-steps of prediction. As seen in Figure 4.35, the R^2 value starts to decrease after about 20 steps. The conclusion drawn is that the derived model is able to predict 15-20 steps into the future and still maintain a low prediction error.

5.2.2 Product B

Pre-processing

To ensure that the outliers in the data set did not disrupt the model, the parameters estimated were controlled by comparing the model estimation to the real values. Since the derived model is able to follow the real density values, the derived model is considered good enough. This can be seen in Figure 4.39.

Estimation of product purity

Once again the estimated parameters were poor in terms of confidence interval, as seen in Figure 4.36. This indicates that the estimated qualities for the content in Product B was poor. The estimated II content for six validation batches, in Product B, can be seen in 4.37. The residuals for the other quantities can be seen in 4.16. One thing to keep in mind, when interpreting the residuals, is that the percentage of the content varies depending on what you are looking at.

The residuals for II is 4.4% and the mean value of the II content is 68.9%. This indicates that the derived model is not sufficient enough compared to σ, which is 2.34. The parameter for all of the models looks like in Figure 4.22 where no significant parameters were found. The initial conditions were also added to the derived model, to see if it was possible to improve the model, but the resulting estimated ones were worse, and the model was not saved or described further in detail.

k-step prediction of temperature

The product quality estimation did not work as expected, and the focus was shifted to prediction of a termination point. The variable used was a density variable, since the operator terminate the batch after the density has reached a certain value. The density prediction was slightly better, due to the regression value indication seen in Figure 4.44. The regression value is around 90% at 27-steps, compared to the regression value done in for Product A which was at 80% at 20-steps. The conclusion drawn is that the density in Product B does not change as much over time as the temperature does in Product A, and therefore the model is able to predict further into the future.

5.3 Methodology comparison

Since the two different methodologies, system identification and BSPC, were done in Matlab and SIMCA, the comparison between the two different methodologies will also contain differences and similarities between the programs.

To visualize the BSPC concept, SIMCA was used. The program is based on a set of different estimation and visualization methods, such as Distance to model, Hotelling's T^2-test and a number of other visualization techniques. The downside of this, is that the flexibility is greatly reduced, since the different techniques to visualize the data set is already decided. The upside of this, is that the user does not have to hard code all the different techniques in order to visualize the data set, and the possibility to get an overview of the data set is done with ease.

Since the system Identification concept is done in Matlab, the flexibility regarding wanted techniques and sub methodologies is much larger than in SIMCA. The downside of this, is that much of the scripts and functions in Matlab require the user to hard code everything from scratch. This usually takes some time, and the possibility to create errors in the code might occur.

The batch level model exploits the number of parameters estimated to find any correlation pattern between the X matrix and the Y matrix. The idea of estimating a large set of parameters to describe a behavior goes against the system identification concept, in regards of keeping the order of a system as low as possible. Since the number of estimated parameters is equal to the number of variables times the number of time steps of a batch, the resulting parameter estimations is very hard to interpret and analyze.

At first the batch evolution model is derived as a static model. SIMCA creates the possibility to lag variables in time, thus creating a dynamic time series model. If the user lags variables, the resulting model will be closely related to a normal time series model, depending on how the variables are lagged. By lagging variables, the resulting model in SIMCA is closely related to the models derived with the system identification concept. SIMCA does not visually depict the resulting model as a formula, so the user have to understand that the lagged model is a time-series model by themselves.

6

Conclusion and Future work

The hardest part in this master thesis was to find and compute models that are able to handle the difficulties regarding three-way matrix structure and still be able to find a correlation pattern between a large set of input data points and a few output end data points. Since most of the known methodologies require an equal amount of input- and output data points when deriving a model, the thesis students had to ensure that the methodologies chosen were able to handle the stated issue.

The quality for both products could be estimated to an extent, depending which of the methodologies that were used. The quality for Product A could be estimated using system identification and for Product B the quality could be estimated using batch statistical process control. One idea of how to move forward is to install a sensor in the plant. The idea is to utilize vibrational spectroscopy in the IR range to quantify the chemical composition.

The concept of estimating batch termination time for both products was quite successful, as was the early fault detection methodology. For future projects the different termination time models could be implemented in real-time.

The prediction of termination variables used for both products worked well with the system identification concept.

Since the work is based on empirical modeling of input and output data, the models are only valid within the space defined by the data sets which were used in the master thesis work.

Bibliography

Bro, R. and A. K. Smilde (2014). “Principal component analysis”. *Anal. Methods* **6** (9), pp. 2812–2831. DOI: 10.1039/C3AY41907J. URL: http://dx.doi.org/10.1039/C3AY41907J.

David Di Ruscio, D. ing. (1995). *Subspace system identification: Theory and applications*. http://home.hit.no/~hansha/documents/control/theory/subspace_systemidentification.pdf. Visited on 2015-03-02.

Eriksson, L., T. Byrne, E. Johansson, J. Trygg, and C. Vikström (2013). *Multi- and Megavariate Data Analysis Basic Principles and Applications.* Umetrics Academy, Umeå. ISBN: 9789197373050.

Ferrer, A., D. Aguado, S. Vidal-Puig, J. M. Prats, and M. Zarzo (2008). “Pls: a versatile tool for industrial process improvement and optimization”. *Applied Stochastic Models in Business and Industry* **24**:6, pp. 551–567. ISSN: 1526-4025. DOI: 10.1002/asmb.716. URL: http://dx.doi.org/10.1002/asmb.716.

Johansson, R. (1993). *System Modeling and Identification*. Information and system sciences series. Prentice Hall, Englewood Cliffs, NJ. ISBN: 9780134823089. URL: https://books.google.se/books?id=FZ7gAAAAMAAJ.

Ljung, L. (2010). “Perspectives on system identification”. *Annual Reviews in Control* **34**:1, pp. 1 –12. ISSN: 1367-5788. DOI: http://dx.doi.org/10.1016/j.arcontrol.2009.12.001. URL: http://www.sciencedirect.com/science/article/pii/S1367578810000027.

Mathworks. *Matlab*. http://se.mathworks.com/products/matlab/. Visited on 2015-03-04.

Perstorp. https://www.perstorp.com/. Visited on 2015-03-04.

Qin, S. J. (2006). “An overview of subspace identification”. *Computers & Chemical Engineering* **30**:10–12. Papers form Chemical Process Control {VII} {CPC} {VII} Seventh international conference in the Series, pp. 1502 –1513. ISSN: 0098-1354. DOI: http://dx.doi.org/10.1016/j.compchemeng.2006.05.045. URL: http://www.sciencedirect.com/science/article/pii/S009813540600158X.

Umetrics. *Simca*. `http://www.umetrics.com/products/simca`. Visited on 2015-03-04.

Verhaegen, M. (1994). "*Identification of the deterministic part of MIMO state space models given in innovations form from input-output data*". *Automatica* **30**:1, pp. 61 –74.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER´S THESIS

Date of issue
June 2015

Document Number
ISRN LUTFD2/TFRT--5974--SE

Author(s)
Sebastian Larsson
Eric Grundén

Supervisor
Johan Rönnberg, Perstorp
Robert Zuban, Perstorp
Rolf Johansson, Dept. of Automatic Control, Lund University, Sweden
Charlotta Johnsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Sponsoring organization

Title and subtitle

Evaluation of a batch process by means of batch statistical process control and system identification

Abstract

Batch processes play an important role in the production of high quality specialty chemicals. Examples include the production of polymers, pharmaceuticals and formulated products. In this master thesis, the study of transformation of materials, by batch distillation and mixing is studied. The study is done by means of batch statistical process control and system identification methods in order to build soft sensors that can predict product quality and end-point but also to use the batch trajectory features for early fault detection.

In contrast to a continuous process, a batch process is a finite duration process, from initialization to completion. The physical state of the process is derived from measured variables, for example, temperatures, pressures and flows and comes from on-line measurements of the on-going process.

Since there are many variables, in terms of inputs and outputs, multivariate data analysis is a suitable choice for extracting systematic information which is used to find a relationship among the variables but also to visualize the batch trajectories and deviations from normal batch evolution.

The results suggest that the end point can be predicted during distillation and mixing and it seems like it is possible to separate normal batches from different batches by means of batch statistical process control strategies. However, estimating product purity during distillation was not possible due to limited variation in the output data. Instead, system identification methodologies were a better choice.

Product quality after mixing was poorly estimated with system identification tools due to the lack of variability within the time average of the different variables used, but was better predicted with batch statistical process control.

Keywords
System identification, Sub space identification, Kalman filter, Batch statistical process control, Partial least squares, Batch process

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title	*ISBN*
0280-5316	

Language	*Number of pages*	*Recipient's notes*
English	1-62	
Security classification		

http://www.control.lth.se/publications/