Pontificia Universidad Católica del Perú

Escuela de Posgrado

Tesis

Irregularly Sampled Data in the Design of a Soft Sensor System: Some Preliminary Results

Para obtener el grado de:

Magíster en Ingeniería de Control y Automatización

Presentada por:	Fritjof Griesing-Scheiwe
Fecha de nacimiento:	11/11/1993, Hanover, Alemania
Profesor Responsable (TU Ilmenau):	Prof. Dr.-Ing Yuri Shardt
Miembro del jurado (TU Ilmenau):	Prof. Dr.-Ing Siegbert Hopfgarten
Profesor Responsable (PUCP):	Prof. Dr.-Ing Carlos Gustavo Pérez Zuñiga
Fecha y lugar:	29/10/2018, Ilmenau

Acknowledgements

I wish to thank, first and foremost Prof. Dr. Yuri Shardt for the support and guidance he has given me in the course of preparing and writing this thesis. He was always available, helping me to acquire the ability and knowledge necessary to write this thesis.

Furthermore, I would like to express my gratitude to Prof. Dr. Gustavo Pérez for his advice despite the big distance between Lima and Ilmenau.

I thank Stephan Henning for revising and commenting my thesis and Dr. Kai Wulff for his support in literature search.

This thesis would not have been possible without the love and support of my parents, my sister and my fiancé.

Statement of Authorship

I, Fritjof Griesing-Scheiwe, hereby declare that this bachelor thesis presented here is to the best of my knowledge and belief original and the result of my own investigations, unless otherwise acknowledged.

Formulations and ideas taken from other sources are cited as such.

This work has not been published and submitted, either in part or whole, for a degree at this or any other university.

Fritjof Griesing Scheiwe

Abstract

In modern industrial applications, sensors are an expensive part of installed systems. Nevertheless, many system variables cannot be measured sufficiently frequently or accurately. Thus, soft sensors have been developed to estimate those variables without the expense of additional hardware. The use of a soft sensor with a bias update term has shown to perform well for disturbed systems with time delays and multirate sampling times. In industrial application, the time delay and sampling times often vary. Yet, the case of variation of the time delay and sampling time in the bias update term has not been considered in previous publications. This thesis tests a soft sensor with bias update term in simulation and gives a modification yielding better performance. It is shown that the tested method gives unstable results. Hence, a more general method with a bias update term that considers all possible sampling times in each step is proposed, giving stable results in simulation. Furthermore, the stability of the general method is proven mathematically by building a state space representation and applying the Bauer-Premaratne-Durán theorem to the stability of switching systems.

Resumen

En aplicaciones industrialesmodernas, sensoressonunapartecostosadesistemas instaladas. De todas maneras, las medidas de muchas variables no son suficientemente frequente o preciso. Por lo tanto, sensores virtuales fueron desarolladosparaestimarestosvariablessinelgastoensensoresadicionales. El usodeunsensorvirtualconuncomponentedeactualizacindesesgofunciona bien para sistemas con retardo de tiempo y múltiples tiempos de muestreo. En aplicaciones industriales, es común que el retardo del tiempo y el tiempo demuestreosonvariables. Sinembargo, elcasoderetraso detiempovariable y tiempo de muestreo variable no fue considerado en publicaciones previas. Esta tesis testa un sensor virtual con componente de actualizacin de sesgo en simulación y desarolla una modificación que da mejores resultados. Se muestra que el método anterior resulta en una estimación inestable. Por eso, un método más general con un bias update term es presentado que considera todos los tiempos de muestreo posibles en todos los instantes de tiempo, que da resultados estables. Se prueba la estabilidad matemáticamente, construiendo una representación de matrices para cada tiempo de muestreo y aplicando el teorema de Bauer-Premaratne-Durán estabilidad de sistemas cambiantes.

Zusammenfassung

In modernen industriellen Anwendungen sind Sensoren ein kostenintensiver Teil von Installationen. Dennoch können viele Variablen nicht ausreichend häufig oder präzise gemessen werden. Daher wurden virtuelle Sensoren entwickelt, um diese Variablen ohne den Aufwand zusätzlicher Sensoren zu schätzen. Die Nutzung von virtuellen Sensoren mit einer Bias-Update-Komponente hat sich für Systeme mit Totzeit und unterschiedlichen Abtastzeiten bewährt. In industriellen Anwendungen kommt es oft vor, dass sowohl Totzeiten als auch Abtastzeiten variieren. Dennoch wurde der Fall von variablen Totzeiten und Abtastzeiten in vorherigen Veröffentlichungen nicht berücksichtigt. Diese Arbeit testet einen viruellen Sensor mit einer Bias-Update-Komponente in dieser Situation und schlägt eine Modifikation des Algorithmus vor, die bessere Ergebnisse erzielt. In den Tests zeigt der alte Ansatz ein instabiles Verhalten der Schätzung. Daher wird ein allgemeinerer Ansatz vorgeschlagen, der in jedem Zeitschritt alle möglichen Schrittlängen der Abtastzeit berücksichtigt. Dieser Ansatz führt in Simulationen zu guten Ergebnissen. Darüber hinaus kann über eine Zustandsraumdarstellung und den Satz von Bauer-Premaratne-Durán über schaltende Systeme ein mathematischer Beweis gegeben werden, dass gesuchte Variablen korrekt geschätzt werden.

Notation

$\mathbb{R}$	Field of the real numbers
$\|a\|$	Absolute value if a is a number cardinality if a is a set
$\vec{x} \in \mathbb{R}^n$	A column vector with n real entries
$\|\|\vec{x}\|\|$	The Euclidian norm of a vector
$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$	A matrix with m rows and n columns
A^T	Transpose of the matrix $A \in \mathbb{R}^{m \times n}$
A^{-1}	Inverse of a square matrix $A \in \mathbb{R}^{n \times n}$
$\|\|A\|\|_\infty$	Maximum norm of the matrix A
$G(z^{-1})$	Transfer function
z^{-d}	Time delay of d steps
$Var(x) = \sigma_x^2$	Variance of the random variable x
$[a, b]$	The set of all integer values in the range between a and b, including a and b
$\hat{a}$	The estimated value of the variable a
$\lim_{z^{-1} \to 1} (G)$	The limit of the transfer function $G(z^{-1})$ as z^{-1} approaches 1
$\sum_{i=a}^{b} A_i$	The sum of all $A_i \in \mathbb{R}^{m \times n}$ for $i \in [a, b]$
$\prod_{i=a}^{b} A_i$	The product of all $A_i \in \mathbb{R}^{n \times n}$ for $i \in [a, b]$
$x \sim N(\mu, \sigma^2)$	A random variable with normal distribution, with mean μ and variance σ^2
$x \sim \chi^2_m$	A random variable with χ^2 distribution, where the degree of freedom is m
$p(x\|y)$	The probability of the event x given the event y
$E(x\|y)$	The expected value of the random variable x given the event y

Table of Contents

List of Figures

List of Tables

1 Introduction

Knowledge about the states of industrial systems is key for effective control or supervision. However, in many applications, high sampling times or measurement delays cause bad measurement performance or the need of expensive sensors. A method to deal with those problems in the estimation of the states is soft sensors. Soft sensors use the data of some of the available sensors combined with mathematical models to estimate variables which are hard to measure.

Soft sensors are described in Fortuna et al. (2007); Kadlec et al. (2009) and general information is provided about their different structures and application. Soft sensors require a model whose construction will be explained. Some of the possible model structures used in soft sensor development are presented in Shardt (2015). To determine the parameters for the model of the soft sensor correctly, several steps are required. First, the data must be aquired. This can be done by a design of experiment (Shardt, 2015; Montgomery, 2009). The data is then processed. For the detection of outliers, the 3σ rule and the Hampel identifier are given as simple detection methods (Fortuna et al., 2007; Kadlec et al., 2009; Lin et al., 2006). The **p**rincipal **c**omponent **a**nalysis (PCA) (Fortuna et al., 2007; Mu Zhu, 2006) as well as the Q-statistic and T^2-statistic (Zhang et al., 2016; Ding, 2014) are more powerful, but require more computational effort. Several strategies for dealing with missing values are presented. Some simple strategies are given in Khatibisepehr and Huang (2008). The **n**onlinear **i**terative **p**artial **l**east **s**quares algorithm (NIPALS) (Geladi and Kowalski, 1985; Miyashita et al., 1990; Nelson et al., 1996) and the **e**xpectation **m**aximisation algorithm (EM) (Dempster et al., 1977; Chen and Gupta, 2010; Borman, 2004) are more involved and generally lead to better results. With both model structure and data given, the parameters are determined using regression (Shardt, 2015). The model is then validated to ensure that it represents the data adequatly (Shardt, 2015). In addition to the system model, the soft sensor's bias update term that corrects its estimate is considered (Shardt and Huang, 2012a). In this context, an introduction of tracking is given Shardt (2015). Further-

more, three performance metrics are presented. The metrics are the absolute error (Shardt and Huang, 2012a), the sum of squares error (Jin et al., 2012) and Pearson's coefficient of regression (Shardt, 2015).

The aim of Shardt and Huang (2012a,b) is the design of a soft sensor in the presence of time delay and multirate sampling time. The results are reviewed and extended, considering variation in the sampling time and time delay of the measurements. Therefore, a new approach is proposed and its adequacy shown using the Bauer-Premaratne-Durán theorem about the stability of switching systems (Lin and Antsaklis, 2009; Bauer et al., 1993). Both approaches are tested on a **c**ontinuous **s**tirred **t**ank **r**eactor (CSTR) presented in Morningred et al. (1990) and used as a benchmark in several other publications (Shardt and Huang, 2012a,b; Huang et al., 2000; Zhang et al., 2016).

This thesis focuses on the application of soft sensors in the presence of variable time delay and variable sampling time. In the course of working on this topic, 4 objectives are considered:

1. The soft sensor method applicable in the presence of multirate sampling and time delay presented in Shardt and Huang (2012a,b) is reviewed.
2. The method is tested in simulation.
3. An alternative approach is proposed which is designed to perform better for variable sampling times and variable time delays.
4. Mathematical proof for the adequacy of the modified method is given.
5. The convergence of the method is shown in simulation.

2 Soft Sensors

2.1 Properties of Soft Sensors

Soft sensors are a method to determine the unknown values of systems. They use the measurements of sensors and knowledge about the given system to

give an estimate of unknown values (Kadlec et al., 2009; Fortuna et al., 2007; Jin et al., 2012).

Numerous applications of soft sensors exist. They can be used as a backup for given sensors. In some cases, they can replace some sensors completely. Furthermore, it is possible to use soft sensors to detect faults like parameter changes or hardware malfunctions.

Soft sensors can be distinguished between model-driven soft sensors and data-driven soft sensors. In model-driven soft sensors the system information is given as a first principle model, *i.e.* the system model is built from the theoretical information about the system. Data-driven methods obtain their system information from measurements on the system in observation, historical data or experiments and construct a model that fits the data. This thesis has its emphasis on data-driven methods.

The soft sensors defined in this thesis follow a structure given in Shardt and Huang (2012a,b). The methods described is applied to an open loop process as given in Figure 1.

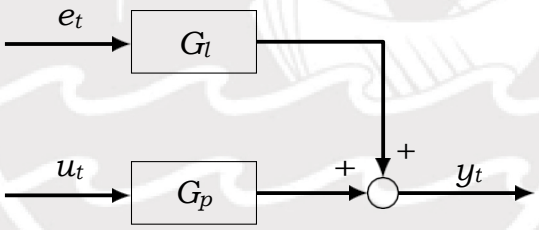

Figure 1: Open-loop process

The variable u_t is the input of the system, y_t an output and e_t an error term. The block G_p describes the transfer function of the system, G_l is the transfer function of the error. Transfer functions will be described in section 2.2.1. In order to give an estimate of the output y_t, the transfer function G_p is modelled. The resulting transfer function is $\hat{G}_p$. Since models always differ from the system they model, a feedback is used to correct estimation errors. The transfer function in the feedback is called bias update term and given by G_B (Shardt and Huang, 2012a,b). The system is built as shown in Figure

2. Note that the transfer functions G_l, G_B, G_p and $\hat{G}_p$ may be nonlinear in general.

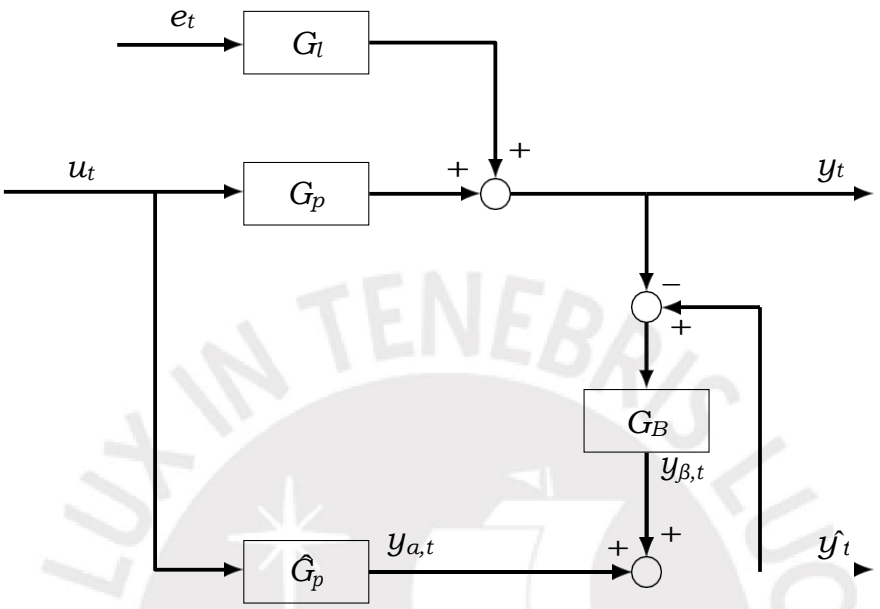

Figure 2: Open-loop soft sensor with bias update term

The soft sensor model presented in Figure 2 has several components. The quality of its estimation is determined by the model of the system $\hat{G}_p$ and by the choice of the bias update term G_B.

The model of the system $\hat{G}_p$ gives a simulation of the systems behaviour in the soft sensor. Therefore, its output $y_{a,t}$ reflects the behaviour of the system well and gives an estimate of the real output y_t. However, due to errors in the model and the error term e_t, an estimation purely based on simulation does not perform well. Since the errors in the estimate of the inner state of the system are passed on to future estimates, the estimation error can increase over time. If a linear model is used for a nonlinear system, as it is common practice in many applications, the estimate is likely to become biased even for steady state estimations.

Therefore, the bias update term G_B is used to compare the estimate $\hat{y}_t$ with the available measurements y_t and correct it using a feedback loop. For the use of the bias update term, it is assumed that some measurement of

the true output is given. Often, the measurement is not easily available, for example, if it is delayed in time or sampled infrequently. Thus, measuring alone does not provide a good estimate. Therefore, a soft sensor is used.

2.2 Construction of the System Model

The construction of $\hat{G}_p$ is done using basic system identification methods. An overview sufficient for the construction of simple models is given in this section.

2.2.1 Model Structure

In the application of soft sensors, numerous different model structures are used. Some models for data-driven methods are presented in this section (Shardt, 2015).

The most common model type for the considered process are discrete models. They are used frequently, since they are designed to work on digitally sampled data, which is commonly used for soft sensors. Two different model types are considered. Models based on transfer functions and models based on the impulse response.

The models based on the transfer function are derived from the prediction error model. It is given by

$$A(z^{-1})y_t = \frac{B(z^{-1})}{F(z^{-1})}u_t + \frac{C(z^{-1})}{D(z^{-1})}e_t$$

where $A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$, $D(z^{-1})$ and $F(z^{-1})$ are polynomials in z^{-1}. The polynomials have the form

$$\theta_0 + \sum_{i=1} \theta_i z^{-i}$$

where n is the maximal order of the polynomials, θ_i are the parameters and $\theta_0 = 0$ for $B(z^{-1})$, $\theta_0 = 1$ for the other polynomials.

The most popular models are the Box-Jenkins model, the ARMAX model.

In the Box-Jenkins model, the parameters θ_i for $i > 0$ in $A(z^{-1})$ are set to 0, leading to $A(z^{-1}) = 1$. In the ARMAX model, $F(z^{-1}) = D(z^{-1}) = 1$ holds.

A different, but in many cases effective approach is the impulse response model. It uses the impulse response of the process h and considers both the input and disturbance as a series of impulses. It is described by the equation

$$y_t = \sum_{i=0}^{\infty} h_i u_{t-i} + \sum_{j=0}^{\infty} h_j e_{t-j}.$$

The sum in this equation goes to infinity. In case of a finite impulse response all summands are 0 for sufficiently high i, j.

2.2.2 Design of Experiment

Experiments are one of the best ways to obtain the necessary data for regression. The planning and analysis of the experiments can impact the quality of the estimations and the cost of the experiments greatly. In this section, an overview of how experiments are planned and analysed is given (Shardt, 2015; Montgomery, 2009).

In any experiment, variables are measured that are to be used in regression. Generally, the system is assumed to be of the form

$$f(\vec{x}, \vec{\theta}) = \vec{y}$$

where $\vec{\theta}$ are the parameters of the system, $\vec{y}$ is the output, $\vec{x}$ are the regressors and $f(\cdot)$ a function, describing the system behaviour. Typically, the regressors $\vec{x}$ are fixed at a desired value to perform the experiment and the outputs $\vec{y}$ are measured. The parameters are calculated using regression as described in section 2.2.5. However, to get a good estimate of the parameters, a good choice of regressors and parameters must be taken. An analysis of this choice is given in section 2.2.6.2.

Given a model with several parameters and their respective regressor, the regressors are only adjusted at some discrete levels in planning the experiments. In the simplest case, two levels are considered for the level of the

regressors. Hence, each regressor can be applied using its low value or using its high level. The fact that the i^{th} regressor is given at the low level is denoted as $x_i = -1$, the high level is denoted as $x_i = +1$. It is possible to consider more levels for the regressors. However, more levels require more experiments to be covered adequately. Assume that each regressor corresponds to one parameter.

When planning the experiments, it is important to be able to consider the influence of each parameter separately. The simplest approach to do this is the one-factor-at-a-time approach. In the one-factor-at-a-time approach, all regressors are fixed on a baseline. Assume that the lower level -1 is the baseline. After making the baseline experiment, an experiment is done where one regressor is $+1$ and all other regressors are -1. This is done for each regressor. With this approach, all individual influences of the regressors on the outputs are considered. However, interactions between the regressors cannot be identified with this experiment.

A way that does consider the interactions between the regressors is the factorial design. Even though more levels of the regressors are possible, a two level design is assumed. Then, all possible combinations of regressor values are considered. Since these are 2^m combinations, where m is the number of regressors, and since the full set of combinations is considered, this is called a 2^m full factorial design. With this preparation, not only first order interactions like $\beta_1 x_1$ can be considered. Second order interactions like $\beta_{12} x_1 x_2$ and interactions up to the order m like $\beta_{12\ldots m} x_1 x_2 \ldots x_m$ can be considered.

2.2.3 Outlier Detection

In order to build an effective soft sensor, process data is needed. In order to maximise the usability of the data and thus the performance of the soft sensor, preprocessing, such as the elimination or replacement of outliers and missing values is necessary. If outliers are present in the soft sensor design, the sensor does not only predict the process behaviour, it also predicts the behaviour of the outliers given in the data set. One distinguishes between

obvious and nonobvious outliers (Kadlec et al., 2009). Obvious outliers can easily be detected, since they consist of measurements that are impossible to occur or to be measured.

Nonobvious outliers can be detected with several techniques. Some of them are presented in this section.

Some simple rules for the detection of outliers are the 3σ rule and the Hampel identifier (Fortuna et al., 2007; Kadlec et al., 2009; Lin et al., 2006). These rules detect outliers based on the univariate distribution of the variables.

The 3σ rule uses the mean value μ and the standard deviation σ to identify outliers. Values within $\mu \pm 3\sigma$ are considered normal, whereas values not included are interpreted as outliers.

As the mean, the median intents to give an estimate for the centre of data (Lin et al., 2006; Shardt, 2015). Assume a data set $X \in \mathbb{R}^m$ consisting of m measurements of a variable. The median is defined in a way that half the values in the data set are bigger than the median and half the values are smaller. If the cardinality of the data set is odd, the median is the middle value of the ordered data set. If it is even, the median is the mean between the two middle values.

The MAD describes the spread of data. It is calculated as

$$\text{MAD} = 1.4826\ \text{median}\{|x_i - \text{median}(X)|\}$$

where the values of the data set are given as x_i and the absolute value of $(x_i - \text{median}(X))$ is calculated. It is used analogously to the standard deviation.

Since the mean and standard deviation can be strongly influenced by the outliers, the Hampel identifier is proposed in Lin et al. (2006). Instead of the mean and the variance, it uses the robust median and MAD. Any value x_j fulfilling $|x_j - x^y| > 3\text{MAD}$ is considered an outlier.

A more elaborate approach for outlier detection is given with help of a **p**rincipal **c**omponent **a**nalysis (PCA). It is applied to multidimensional data.

Since some covariance matrices must be calculated, a short introduction to the covariance matrix is given.

The covariance matrix is the generalisation of the variance for the multidimensional case (Shardt, 2015). Given multidimensional data of m measurements for n different variables. The data is described by the matrix $X = \begin{bmatrix} \vec{x}_1 & \vec{x}_2 & \dots & \vec{x}_n \end{bmatrix} \in \mathbb{R}^{m \times n}$ with the mean $\bar{x} = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \dots & \bar{x}_n \end{bmatrix}^T$.

The notation $(\dots)^T$ is the transpose of the respective matrix. The covariance matrix is defined as

$$C = \begin{bmatrix} (\vec{x}_1 - \bar{x}_1)^T(\vec{x}_1 - \bar{x}_1) & (\vec{x}_1 - \bar{x}_1)^T(\vec{x}_2 - \bar{x}_2) & \dots & (\vec{x}_1 - \bar{x}_1)^T(\vec{x}_n - \bar{x}_n) \\ (\vec{x}_1 - \bar{x}_1)^T(\vec{x}_2 - \bar{x}_2) & (\vec{x}_2 - \bar{x}_2)^T(\vec{x}_2 - \bar{x}_2) & \dots & (\vec{x}_2 - \bar{x}_2)^T(\vec{x}_n - \bar{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\vec{x}_1 - \bar{x}_1)^T(\vec{x}_n - \bar{x}_n) & (\vec{x}_2 - \bar{x}_2)^T(\vec{x}_n - \bar{x}_n) & \dots & (\vec{x}_n - \bar{x}_n)^T(\vec{x}_n - \bar{x}_n) \end{bmatrix}.$$

The covariance matrix is a symmetric matrix in $\mathbb{R}^{n \times n}$.

For outlier detection with PCA, a T^2-statistic or Q-statistic must be used. Therefore, a short introduction to the statistics is given. The Q-statistic and T^2-statistic determine, if a value belongs to a distribution of values or if it is an outlier (Zhang et al., 2016; Ding, 2014). Therefore, these statistics are used for outlier detection in combination with algorithms such as the principal component analysis. The T^2-distribution generally yields better results than the Q-distribution. The Q-distribution has the advantage, that a matrix inversion is necessary using the T^2-distribution and unnecessary using the Q-distribution. In case of an ill-conditioned matrix, the inversion necessary for the T^2-statistic would lead to inaccurate results.

The values of the test statistics for all data points are defined by two equations

$$J_Q = Y^T Y J_T$$
$$^2 = Y^T \hat{\Sigma}^{-1} Y_Y$$

where Y is the data taken from $\mathbb{R}^{m\times n}$ with n data points of dimension m, $(\cdot)^{-1}$ the matrix inverse and $\hat{\Sigma}_Y$ the estimated covariance matrix of the data. Note that calculating the inverse of the covariance of the data does lead to some additional complexity, but usually improves the statistic, since the spread of the data is considered in the prominent directions only.

To use the values of the statistics, thresholds are calculated. For both the Q-statistic and T^2-statistic, one way to calculate the threshold is shown:

$$J_{th,T^2} = \chi^2_{n,a},$$
$$J_{th,Q} = trace(\hat{\Sigma}_Y)\chi^2_{1,a}$$

where n is the dimension of the data points and $\chi^2_{n,a}$ the χ^2-distribution with degree of freedom n satisfying $p(\chi^2_n > \chi^2_{n,a}) = a$. Note that the χ^2_n distribution is used, since it gives the distribution of a random variable given by

$$(\chi^2_1 + \chi^2_2 + \ldots + \chi^2_n) \sim \chi^2_n$$

where $\chi_i \sim N(0, 1)$ is normally distributed. Assuming that the data points are normally distributed, the χ^2_n distribution gives an accurate threshold.

Using this information, outlier detection using PCA can be explained and illustrated (Fortuna et al., 2007; Ding, 2014). Assume that the data considered consists of m measurements of n variables and can be written as $X \in \mathbb{R}^{m\times n}$. First, the main directions of the distribution of the data, called principal components, are identified. In order to calculate the principal components, the covariance matrix of the data is calculated. Then, the eigenvectors of the covariance matrix are determined, being the principal components. If only few outliers are present, they will give a good representation of the distribution of the normal values. In many cases, not all principal components are needed. The components can be reduced by discarding the eigenvectors with small eigenvalues.

An overview of methods to choose how many principal components are

part of the PCA is given in Mu Zhu (2006). Here, a method called scree plot is presented. To use the scree plot, all eigenvalues are ordered starting with the largest eigenvalue and then plotted. Observing the plot, there usually is some eigenvalue after which the eigenvalues are much smaller. This leads to some 'big gap' in the sizes of the eigenvalues. The proposed method then chooses all eigenvectors with eigenvalues before the 'big gap' as principal components. Such a scree plot is illustrated in Figure 3.

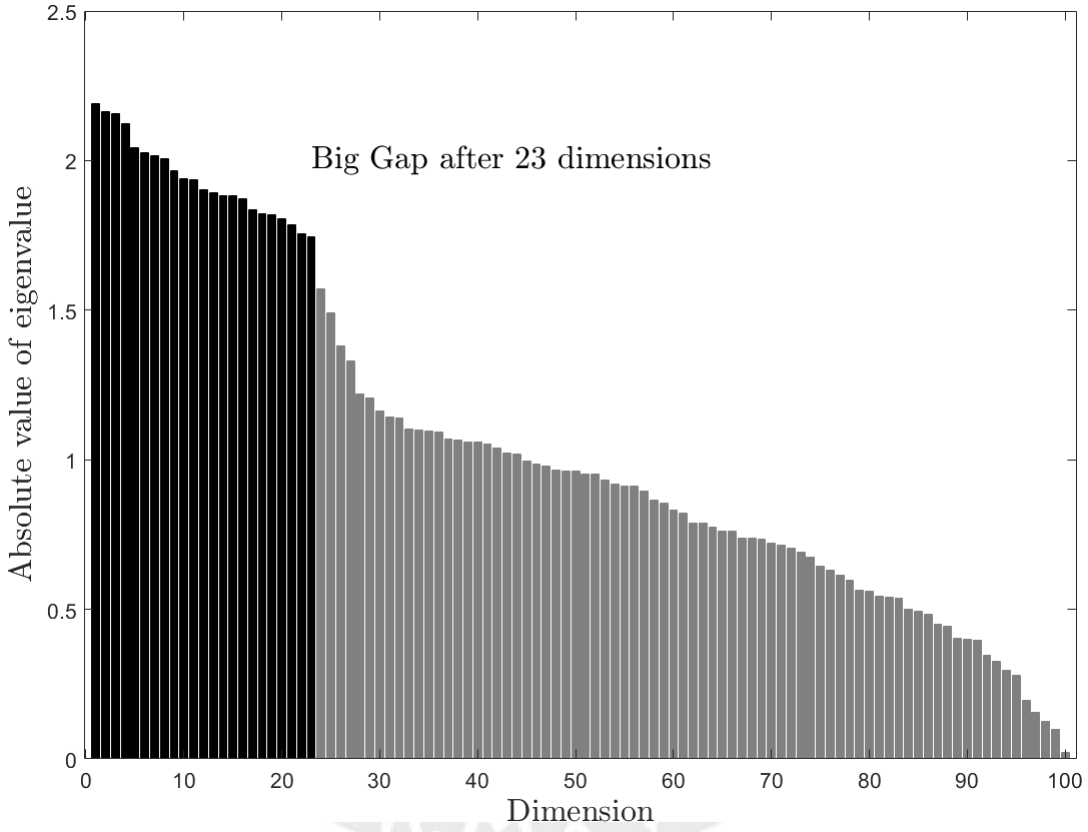

Figure 3: Scree plot

The data X can be transformed to a linear combination of the principal components. In order to achieve this, the principal components are used to form a matrix $V = \begin{pmatrix} \vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_n \end{pmatrix}$. For sake of simplicity, the algorithm is given for the case that no principal components were discarded. Then, the data X is multiplied with the inverse of that matrix, yielding the transformed

data $Y = XV^{-1}$.

Other than the deviation of the normal values, the deviation of the outliers does not follow the principal components. This leads to an unusually high value in Y . Therefore, it is possible to apply a Q-statistic or T^2-statistic to the data Y to determine, which data points are outliers (Zhang et al., 2016). A cost value is given to every measurement. The further the variables are away from the mean of the data, the higher the cost is. If the value deviates in a direction which is typical for the data present, the cost rises less than when the value deviates in an atypical direction.

To evaluate the cost values, a threshold is set. This can be done using the x^2 distribution with a confidence interval a. Data points whose cost exceed the threshold are considered outliers. All other values are considered normal values. The method can be visualised by a multidimensional ellipsis which center is the mean of the considered data. All values inside the ellipsis are considered normal, all values outside are considered outliers.

Figure 4 shows the algorithm, which is summarised as:

1. Calculate the mean and use it to make the data zero-mean.
2. Calculate the covariance matrix of the data.
3. Calculate the eigenvectors. Use the eigenvectors with of the covariance matrix chosen with the help of a Scree plot as principal components.
4. Transform the data using the principal components with the formula $Y = X \begin{matrix} \vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_n \end{matrix}_1$, where $\vec{v}_i$ are the principal components.
5. Calculate threshold using a x^2 distribution where the degree of freedom is the number of variables per data point n
6. Calculate the T^2-statistic of data points and compare it to the threshold
7. Mark every data point whose deviation exceeds the threshold as outlier, otherwise mark it as normal values.

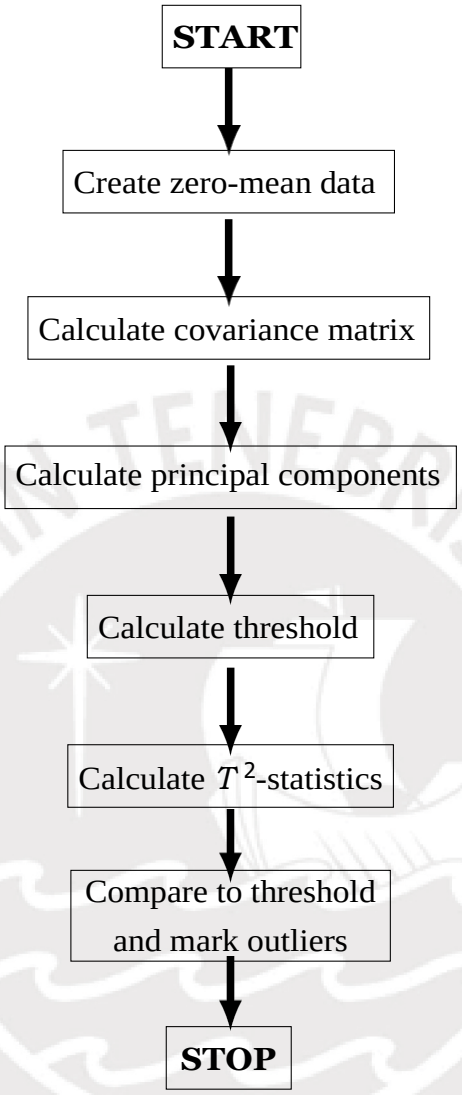

Figure 4: Principal component analysis

An alternative to PCA is using the T^2 statistic directly on the data. Using the covariance of the data to directly apply the statistic yields similar results as the method including PCA. However, other benefits of PCA such as the possible reduction in the data dimension are missed out if the detection is done with the T^2-statistic only.

A similar method to the PCA is the **n**onlinear **i**terative **p**artial **l**east

squares (NIPALS) method. Since it determines the principal components as well, it can be used for outlier detection, as the PCA. Furthermore, it can be used to estimate missing values.

2.2.4 Missing Values

Missing values are values of variables that are not available (Khatibisepehr and Huang, 2008). If outliers are detected, they are considered like missing values. There are three categories of missing values. First there are values which are missing completely at random. In this category, the probability of a value to miss is absolutely independent of both the observed and unobserved values. If values are missing at random, the probability of a value to miss depends on the observed values. If the values are not missing at random, the probability of a value to miss depends on observed and missing values. The right choice of how to deal with the missing values often depends on their category.

For efficient use of data, several strategies are used to deal with missing values. Some of the most common approaches to deal with missing values are presented.

One way to deal with missing data is the deletion of the missing or damaged values. Even though the deleted values are unknown, this method still leads to a loss of information, since the other process values at the time of measurement are deleted, as well. However, in case of completely randomly missing data, unbiased estimates are still possible on the given data.

A straight forward method to replace missing values is the mean substitution. All missing values of a variable are replaced by the mean of the variable. Thus, the size of the data is preserved, even though some estimations based on the data become biased.

The **l**ast **o**bservation **c**arried **f** orward method (LOCF) replaces missing values with the last known value.

To use regression imputation a model of the process is built via regression. Missing values are then replaced by applying the model to the preceding measurements.

The **n**onlinear **i**terative **pa**rtial **l**east **s**quares (NIPALS) method is similar to the PCA described in section 2.2.3 (Nelson et al., 1996; Geladi and Kowalski, 1985; Miyashita et al., 1990). Unlike PCA, the principal components are not identified at once. Instead, they are identified iteratively. The NIPALS method is implemented on data $X \in \mathsf{R}^{m \times n}$. The data consists of m samples, each including n variables.

Dealing with missing values with NIPALS requires a model to be built. The model consist of the loadings $\vec{p}_i \in \mathsf{R}^{1 \times n}$, which act analogously to the principal components in PCA and the scores $\vec{t}_i \in \mathsf{R}^{m \times 1}$. If both the scores and loadings are determined sufficiently well, the complete data $X \in \mathsf{R}^{m \times n}$ can be estimated as the sum $X = \sum_{i=1}^{k} \vec{t}_i \vec{p}_i$, where k is the number of components. The algorithm works as follows.

In the initial step of the algorithm, an iteration variable $i = 1$ is defined. In each step, the residual of the data E_i is used to extract remaining information about the data. Initially, the residual consists of the complete data $E_1 = X$.

In each iteration, one loading $\vec{p}_i$ is determined. It is determined in a way that minimises $E_i - \vec{t}_i \vec{p}_i$.

To find a good combination of score and loading, a loop is implemented in each iteration. Usually, a column of the data matrix is used as the initial score. It is used to calculate the initial estimate of the loading using least square estimation as described in section 2.2.5. The formula for the calculation is $\vec{p}_{d,i} = (\vec{t}_i^T \vec{t}_i)^{-1} \vec{t}_i^T E_i$. The loading is changed to a loading with the length 1 by calculating $\vec{p}_i = \frac{\vec{p}_{d,i}}{||\vec{p}_{d,i}||}$, where $||\cdot||$ is the Euclidian norm for vectors. Since the score is not adequate for the new loading, another least square estimation for the score is done. The new score is $\vec{t}_i = E_i \vec{p}_i^T (\vec{p}_i \vec{p}_i^T)^{-1}$. If the new score is unequal to the old score, the inner loop is started anew to calculate another loading and score until the score remains close enough to its last value.

When this happens, the loading and score that describe the actual residual are found. The loading is used as a principal component. To find the next component, the residual is updated using the current loading and score. Therefore, the new residual is given by $E_{i+1} = E_i - \vec{t}_i \vec{p}_i$. If this residual is

sufficiently small using a matrix norm of choice, the algorithm terminates. If it is not, the algorithm goes on with the next iteration as described in the previous paragraph. For the next iteration, the iteration variable is set to $i = i + 1$.

The algorithm is summed up in the following enumeration. An illustration of the algorithm is given in Figure 5.

1. Set $E_1 = X$.
2. Choose a score $\vec{t}_i$ from the columns of E_i.
3. Calculate the loading direction $\vec{p}_{d,i}$: $\vec{p}_{d,i} = (\vec{t}_i^T \vec{t}_i)^{-1} \vec{t}_i^T E_i$ (using a least squares estimation as described in section 2.2.5).
4. Normalise $\vec{p}_{d,i}$, giving a unit vector $\vec{p}_i = \frac{\vec{p}_{d,i}}{||\vec{p}_{d,i}||}$.
5. Recalculate the score due to change in $\vec{p}_i$: $\vec{t}_i = E_i \vec{p}_i^T (\vec{p}_i \vec{p}_i^T)^{-1}$. (using a least squares estimation as described in section 2.2.5).
6. If $\vec{t}_i$ changed in step 4, go to step 3. If not, go to the next step.
7. Set $i = i + 1$. Calculate the residual to be minimised in the next iteration $E_i = E_{i-1} - \vec{t}_{i-1} \vec{p}_{i-1}$.
8. If the residual is small enough, terminate the algorithm. Else, go to step 2.

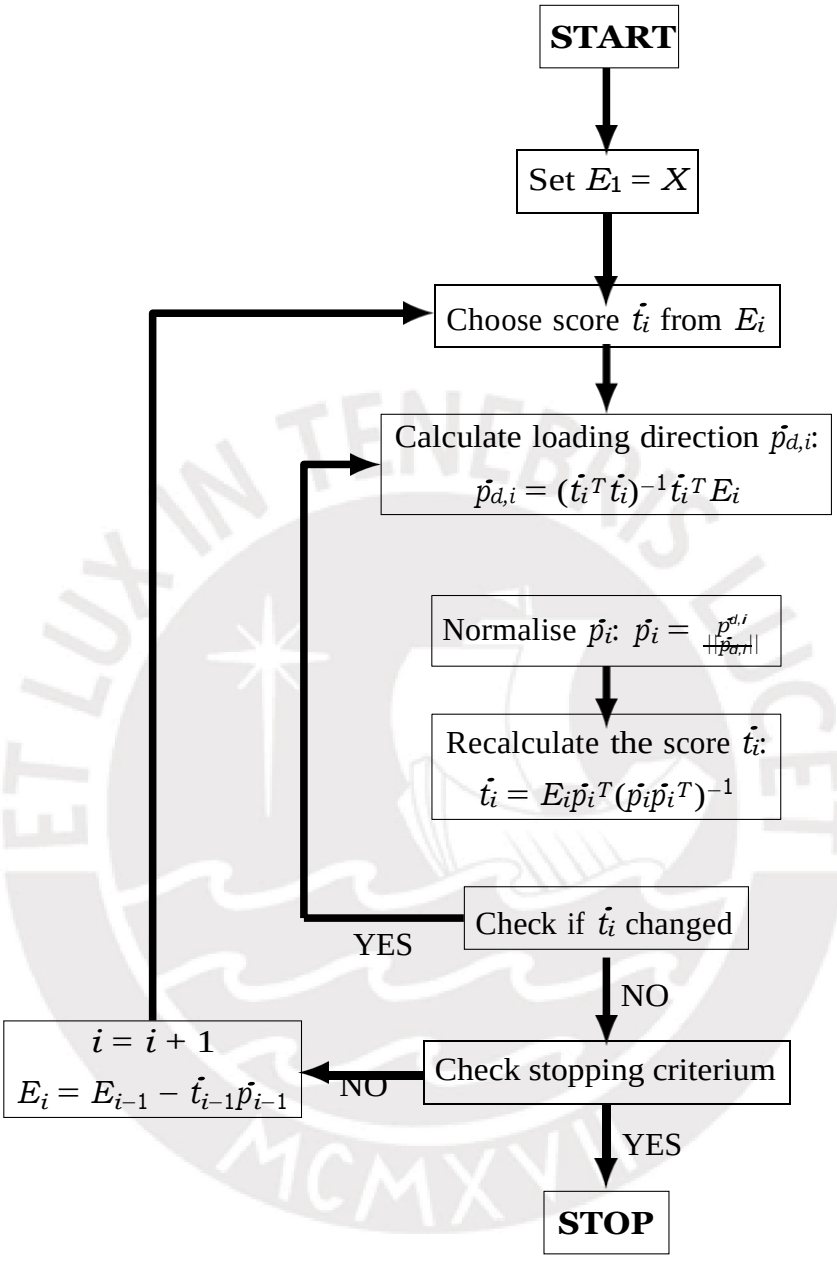

Figure 5: NIPALS algorithm

The algorithm gives the principal components of the data *X*. With the help of these principal components the missing values are chosen in a way that they fit the distribution of the data.

Another common algorithm to estimate missing data is the **e**xpectation-**m**aximisation (EM) algorithm (Dempster et al., 1977; Borman, 2004; Chen and Gupta, 2010). The algorithm uses the principle of likelihood to find the best possible distribution of missing values. The principle of likelihood is explained at hand of the likelihood function.

The likelihood describes, how probable a set of parameters is if the output of an experiment is known (North, 1968). It is a common situation in system identification that measurements of a system are given, but the parameters causing this outcome are unknown. In such a situation, it is desirable to find the parameters which are most likely to give the observed outcome.

Assuming the outcome of the experiment is the data X and the parameters of the system of interest are described by the vector $\vec{\theta}$. Given the correct parameters $\vec{\theta}^*$ exist, the probability distribution function of the outcomes of experiments is described by the conditional probabilities $p(X|\vec{\theta}^*)$. However, the case of a system identification, the data is given and the parameters are to be determined. Thus, the function $p(X|\vec{\theta})$ is considered as a function in the parameters $\vec{\theta}$ with a fixed X. It is called the likelihood function and defined as $L(\vec{\theta}) = p(X|\vec{\theta})$.

In estimation it is common to seek the parameters that maximise the likelihood function. An estimation fulfilling this criterion is called maximum likelihood estimation.

The algorithm assumes complete data to exist and describes it as a real valued vector $\vec{x}$. The data is divided into the observed data $\vec{y}$ and the missing data $\vec{z}$. Furthermore, a function $T(\vec{x}) = \vec{y}$ is defined. The function T is deterministic and depends on parameters $\vec{\theta}$.

The algorithm has its name since it consists of two alternating steps which iterate until a stationary solution is found. They are called the expectation step and the maximisation step. In the expectation step, given parameters are used to estimate the missing values. In the maximisation step, the esti- mated missing values and the observed values are used to give a parameter set for the next iteration. Each step is explained in more detail and Figure 6 is presented to illustrate the algorithm.

The observed values $\vec{y}$ and the current parameters $\vec{\theta}_i$ are given, as well as

the structure of $T(\cdot)$. It is assumed that the densities $p(\vec{y}|\vec{\theta})$ and $p(\vec{x}|\vec{\theta})$ are known. In the expectation step, it is necessary to find a distribution $p(\vec{x}|\vec{y}, \vec{\theta})$ for the current iteration. It is needed for the construction of the Q-function $Q(\vec{\theta}|\vec{\theta}_i)$. The Q-function serves to maximise the likelihood of the parameters for the next iteration. Therefore, it is defined as

$$Q(\vec{\theta}|\vec{\theta}_i) = E(\log(p(\vec{x}|\vec{\theta}))|\vec{y}, \vec{\theta}_i)$$

where $E(\cdot|\cdot)$ is the conditional expectation and $\log(\cdot)$ the natural logarithm. Note that the logarithm is included rather for mathematical simplicity than due to necessity. It does not affect the monotonicity of the Q-function. The value of the Q-function is non-decreasing for an increasing number of iterations. Furthermore, an increase in the Q-function implies an increase of the likelihood of the parameters in a way that

$$Q(\vec{\theta}_{i+1}|\vec{\theta}_i) > Q(\vec{\theta}_i|\vec{\theta}_i) \implies L(\vec{\theta}_{i+1}) > L(\vec{\theta}_i).$$

In the application of the algorithm for the determination of missing values, the Q-function is written as

$$Q(\vec{\theta}|\vec{\theta}_i) = \int_Z \log(p(\vec{y}, \vec{z}|\vec{\theta}))p(\vec{z}|\vec{y}, \vec{\theta}_i)dz$$

(Chen and Gupta, 2010). The maximisation step then maximises the Q-function with respect to $\vec{\theta}$. This maximum is used as $\vec{\theta}_{i+1}$. The algorithm terminates, if both the Q-function and $p(\vec{x}|\vec{y}, \vec{\theta})$ have not changed since the last iteration. The next iteration of the algorithm is started using the estimate $\vec{\theta}_{i+1}$.

The algorithm leads to non-decreasing likelihood of the parameters in each iteration. With more accurate parameters, the estimate of the missing values improves. Furthermore, the algorithm only terminates in stationary points of the likelihood function $L(\vec{\theta})$. These points can be local maxima, global maxima, saddle points or, in unlikely cases, minima.

1 Make an initial estimation of the parameters $\vec{\theta}_0$. Set $i = 0$.

2 Expectation step:

   (a) Estimate the distribution of all values $p(\vec{x}|\vec{y}, \vec{\theta})$ (Assuming $\vec{x}, \vec{y}, \vec{\theta}_i, p(\vec{y}|\vec{\theta})$ and $p(\vec{x}|\vec{\theta}$ are known).
   (b) Construct the Q-function: $Q(\vec{\theta}|\vec{\theta}_i) = \int_Z \log(p(\vec{y}, \vec{z}|\vec{\theta}))p(\vec{z}|\vec{y}, \vec{\theta}_i)dz$.

3 Maximisation step:

   Use a maximisation algorithm to find the parameters $\vec{\theta}$ that maximise $Q(\vec{\theta}|\vec{\theta}_i)$.

4 Check stopping criterium. If $Q(\vec{\theta}|\vec{\theta}_i) \neq Q(\vec{\theta}|\vec{\theta}_{i-1})$ or if $p(\vec{x}|\vec{y}, \vec{\theta}_i) \neq p(\vec{x}|\vec{y}, \vec{\theta}_{i-1})$, set $i = i + 1$ and go to the expectation step.

5 Declare $\vec{\theta}_i$ as the parameters found by the algorithm. Assume the missing values to have the distribution $p(\vec{x}|\vec{y}, \vec{\theta}_i)$.

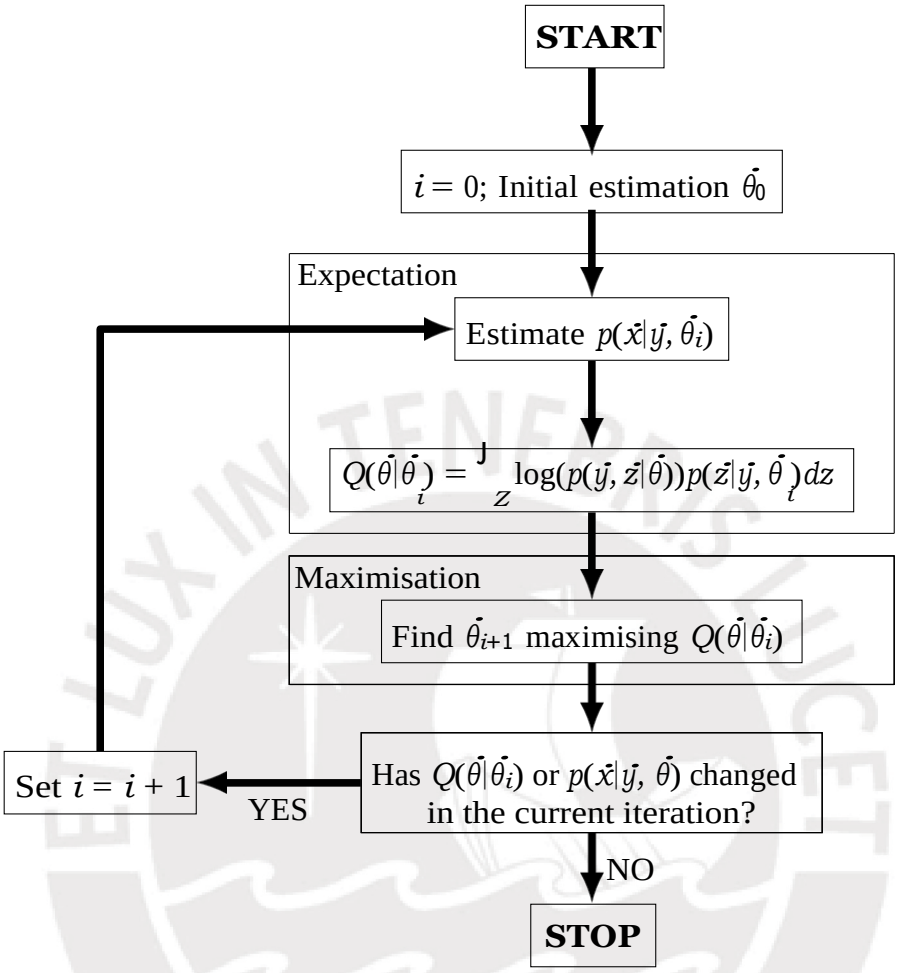

Figure 6: Expectation-maximisation algorithm

2.2.5 Regression

After choosing a model for a soft sensor, the parameters of the model are determined. This is done using regression methods (Montgomery et al., 2012; Shardt, 2015).

2.2.5.1 Sum of Squares estimation

The sum of squares estimation is the most popular regression method (Shardt, 2015). It has its name, since it minimises the sum of squares of the error

between the model output and the process output. The equation is

$$\sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{1}$$

where y_i is the process output and $\hat{y}_i$ is the model output. Assume the given model is linear in the parameters. This is the case if the model can be written in the form

$$\vec{y} = A\vec{\beta} + \vec{\varepsilon}$$

where $\vec{y}$ is the process output, $\vec{\varepsilon}$ an error term, $\vec{\beta}$ the parameters and A is the regressor matrix. The regressor matrix includes inputs or states of the process. They are known during the model build. One assumes that $\vec{\varepsilon}$ is zero-mean, normally distributed and its variance is $Var(\vec{\varepsilon}) = \sigma_y^2 I$. Some examples of models whose parameters may be determined by the sum of squares estimation are given:

- $y = x_1\beta_1 + \varepsilon_1$
- $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1^2 & x_1 \\ x_2 & x_1 x_2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix}$

Note that the models may include nonlinear behaviour, as long as they are linear in the parameters. As shown in Montgomery et al. (2012), the parameters that minimise the sum of squares described in equation (1) are given by

$$(A^T A)^{-1} A^T \vec{y} = \hat{\vec{\beta}}.$$

The sum of squares estimator has several useful properties.

- It is unbiased, so the expected values of the parameters are the parameters minimising the sum of squares described in equation (1).
- Its covariance matrix is easily obtained using the covariance of $\vec{y}$. It is $\sigma_{\vec{y}}^2 (A^T A)^{-1}$.

- It estimates the parameters with a minimum variance.
- It is a maximum likelihood estimation.

2.2.6 Validation

After regression, the model is validated to ensure it represents the process correctly (Shardt, 2015). If the analysis of the model gives unsatisfying results, the model is modified. The modelling procedure is illustrated in Figure 7.

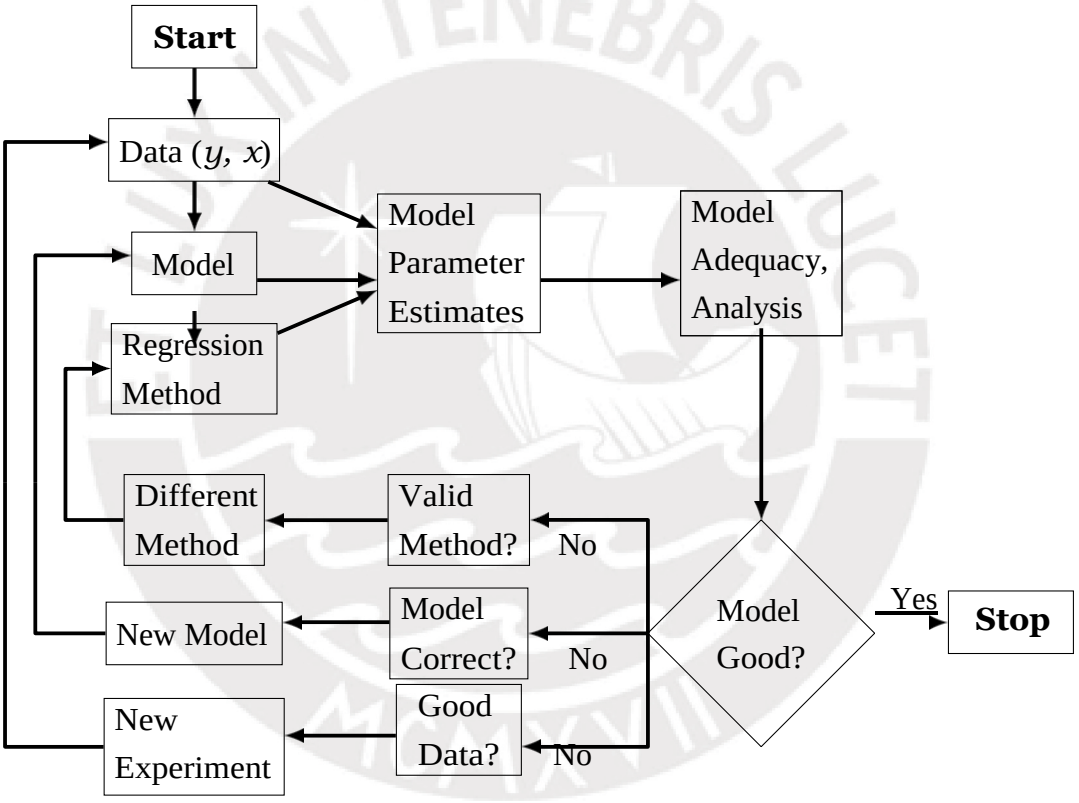

Figure 7: Modelling procedure (after Shardt (2015))

2.2.6.1 Residual Analysis

A validation method to test the general model quality is residual analysis. The residual is the difference between the process output and the model

output. If the process is modelled perfectly, the residual is zero-mean, white noise. Furthermore, the residuals of different outputs are independent. The mean of the residual is easily determined and compared to zero. To evaluate whether the residual is normally distributed and whether its variables are independent a residual analysis can be done by creating scatter plots and evaluating them (Shardt, 2015). This procedure requires some experience in residual analysis.

2.2.6.2 Parameter and Regressor Analysis

Beside residual analysis, the analysis of the parameters and regressors can show, which regressors contribute to the model quality. Furthermore, such analyses give a deeper insight into the model behaviour and hence help to find adequate corrections to the model.

Possible inadequate properties of the model may be a model overfitting. In case of a model overfitting some of the model parameters are redundant. Therefore, they are 0. Overfitting can be detected by calculating the confidence interval of the parameters. If any confidence interval includes 0, the parameter is fixed to 0, this way the model is simplified.

Another situation in which the model should be simplified is the case in which some of the parameters are equal. An equality of two coefficients is shown or discarded via a hypothesis test (Shardt, 2015).

Conversely, the model can also be too simple to perform adequately. This results in a big value of the residuals. There are several methods that indicate a lack of complexity. Pearson's coefficient of regression described in 2.3.2.3 serves this purpose. It compares the variation of the output with the variation of the model output. A value of $R^2 << 1$ indicates that the model can still be improved. If the model is too simple, additional regressors can be taken into account. In some situations a change of model structure can improve the model quality, as well. It requires expert knowledge to correctly predict, which steps lead to a better model.

An obvious model error is a wrong sign of a parameter. Such an error can occur if a relevant regressor is missed out. A high parameter variance

for a relatively small parameter can have the same effect. Another possible reason are computer errors. Multicollinearity, also known as near linear dependence, can lead to wrong signs. Multicollinearity occurs when two highly correlated regressors are used to build a model. Assume that the parameters are calculated via the formula $(A^T A)^{-1} A^T \vec{y} = \beta$, the sum of squares estimation described in section 2.2.5. Two nearly linear dependent columns in A lead to a very small eigenvalue in $A^T A$ and thus a very big eigenvalue in $(A^T A)^{-1}$. This big eigenvalue implies that small changes in y have a big impact on the estimate of $\vec{\beta}$.

2.3 Choice of the Bias Update Term

The bias update term G_B is essential for a the estimation of the output to stay close to the true value(Shardt and Huang, 2012a,b). To illustrate how the bias update term is used in the soft sensor, the graphic of the soft sensor presented in the beginning of section 2 is shown again.

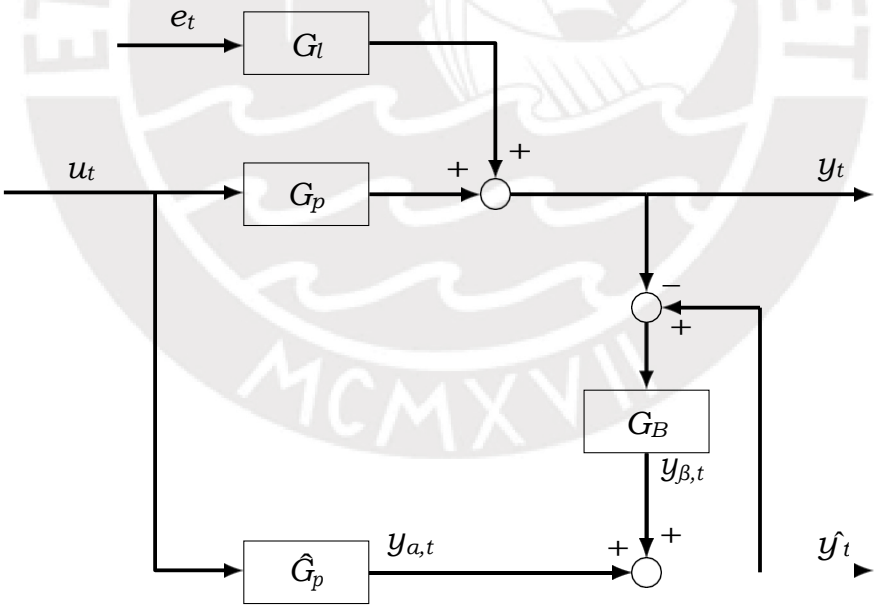

Figure 8: Open-loop soft sensor with bias update term

If no time delay or multirate sampling is present, G_B is a transfer function $\hat{G}_B$ as described in section 2.2.1. If a time delay and multirate sampling is present in the measurements, G_B is modelled as shown in Figure 9.

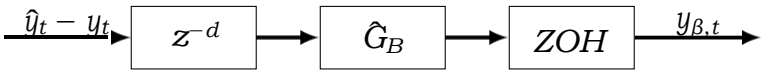

Figure 9: G_B if time delay is present

z^{-d} is a time delay block, $\hat{G}_B$ is a transfer function and ZOH is a zero order hold, sampling every N steps. This structure remains the same if the values of d and N are not constant in time.

Before determining the bias update $\hat{G}_B$, some preliminaries are set. In most real systems, the time delay present is relatively small. Thus, it is reasonable to assume that $d < N$, where d is the time delay and a measurement is taken every N samples. The purpose of the bias update term is to ensure that the soft sensor provides good tracking.

2.3.1 Tracking

The better the estimation of a soft sensor is, the better is the tracking it provides. Some measures for the quality of the tracking are presented in this section.

A straight-forward method of analysing the quality of the tracking of some soft sensor is plotting the correct values against the estimated values (Shardt, 2015). In such a plot, the real values form the x-axis and their respective estimation the y-axis. Therefore, the soft sensor provides good tracking if the plot is close to the $x = y$ line. This plot can identify problems such as single outliers or clusters of outliers. Such phenomena usually occur due to bad data quality. In some applications, the majority of the estimates of a soft sensor is inaccurate. This can be indicated by an incorrect slope of the plot. In this case, the plot lies around the $ax = y$ line, where $a \neq 1$. Sometimes, the estimates differ from the true value by a fixed bias. In that case, the plot lies around $x + b = y$, where $b \neq 0$. If an error in slope or

bias occurs, the model is inadequately chosen or its parameters were not determined correctly. The method that was presented only gives qualitative results about the tracking performance.

Some more quantitative knowledge about a soft sensor can be made using a performance metric on the soft sensor and some other method and comparing the performance of their estimation.

2.3.2 Performance Metrics

Three different performance metrics are considered. The absolute error, the sum of squared error and Pearson's coefficient of regression.

2.3.2.1 Absolute Error

The absolute error is calculated using the absolute value of the difference between measured value y and the estimated value $\hat{y}$. If N measurements are taken, it is calculated as

$$e = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i|.$$

Note that this is the averaged absolute error, due to the division by N. In Shardt and Huang (2012a), this error is called the average forecast error.

2.3.2.2 Sum of Squares Error

The sum of squares error is calculated using the measured value and the estimated value. There are several versions of the sum of squares error. They are multiplied with different scalars, but give the same quality of comparison in application. Here, the **m**ean **s**quare **e**rror (MSE) is presented (Jin et al., 2012). If N measurements are taken, it is calculated as

$$e = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2.$$

2.3.2.3 Pearson's Coefficient of Regression

Pearson's coefficient of regression, also known as R^2 (Shardt, 2015). To calculate that coefficient, one needs the **t**otal **s**um of **s**quares (TSS), given by

$$TSS = \sum_{i=1}^{N} (y_i - \bar{y})^2$$

where N is the number of measurements and $\bar{y}$ the mean of the estimated variable $\hat{y}$. Furthermore, the **s**um of **s**quares due to **r**egression (SSR) is needed to calculate the coefficient:

$$SSR = \sum_{i=1}^{N} (\hat{y}_i - \bar{y})^2$$

Pearson's coefficient of regression is then calculated by the formula

$$R^2 = \frac{SSR}{TSS}.$$

Since the TSS and SSR are both calculated in comparison to the mean of $\hat{y}$, TSS must equal or bigger than SSR. Equality is only given if the estimated value is the same as the measurement. Thus, the coefficient takes values between 0 and 1, where a higher value implies a better estimation.

2.3.3 State of the Art

In Shardt and Huang (2012a), open-loop soft sensors are considered. The paper was published together with Shardt and Huang (2012b), a paper that deals with soft sensors applied to a closed loop system. Since this thesis only deals with open-loop soft sensors, only Shardt and Huang (2012a) is presented.

Using equations based on the structure of Figure 8, a choice of the bias

update term can be made that provides good tracking. The equations are

$$\begin{aligned} y_t &= G_p u_t + G_l e_t, \\ y_{a,t} &= \hat{G}_p u_t, \\ y_{\beta,t} &= G_B(\hat{y}_t - y_t), \\ \hat{y}_t &= y_{a,t} + y_{\beta,t}. \end{aligned}$$

They may be used to form the equation

$$\hat{y}_t = \frac{\hat{G}_p - G_B G_p}{1 - G_B} u_t + \frac{-G_B G_l}{1 - G_B} e_t$$

The assumption of a bias update term without time delay or multirate sampling time is considered, yielding $G_B = \hat{G}_B$. Under that assumption, the bias update term can be chosen as a constant $\hat{G}_B = K_B$ and the equation $\hat{y}_t = y_t$ holds, if $K_B \to \pm\infty$. This can be shown evaluating:

$$\begin{aligned} \hat{y}_t &= \lim_{K_B \to \pm\infty} \left(\frac{\hat{G}_p - G_B G_p}{1 - G_B} u_t + \frac{-G_B G_l}{1 - G_B} e_t \right) \\ &= \lim_{K_B \to \pm\infty} \left(\frac{\hat{G}_p - K_B G_p}{1 - K_B} u_t \right) + \lim_{K_B \to \pm\infty} \left(\frac{-K_B G_l}{1 - K_B} e_t \right) \\ &= G_p u_t + G_l e_t = y_t \end{aligned}$$

Perfect tracking is provided for $K_B \to \pm\infty$. Therefore, in application, the bias update term must be chosen as big as possible, if no time delay and multirate sampling time is present.

The paper states that this choice of bias update term does not provide good tracking in the presence of time delay. In case of delay, $G_B = z^{-d}\hat{G}_B$ holds. Therefore, the bias update shows unstable behaviour for $K_B \to \infty$ and does not converge to the desired limit.

$$\frac{G_B}{1 - G_B} = \frac{K_B z^{-d}}{1 - K_B z^{-d}}$$

Analysing the denominator of the function it can be shown that $K_B \frac{1}{B} z$ yields $1 - K_B z^{-d} = 0$. Thus, it is a root which is outside the unit circle for $K_B \to \infty$. This implies that the transfer function and hence the soft sensor estimate is unstable. Therefore, another structure of the bias update term must be found.

Such a structure was proposed in Shardt and Huang (2012a). First, some conditions to the structure of the bias update term are stated. It is assumed that the term is a transfer function of the form

$$G_B = \frac{\sum_{i=0}^{n_a} a_i z^{-i}}{\sum_{i=0}^{n_b} \beta_i z^{-i}}$$

To ensure that the estimation $\hat{y}_t$ converges to y_t, the steady state behaviour is considered. It is assumed that all variables converge and the values from past time steps become equal to current values. This assumption yields $z^{-1} \to 1$. For $t \to \infty$, the equation

$$\frac{G_B}{1 - G_B} = -1$$

must hold for the error transfer function of $\hat{y}$ being the same as the error transfer function of y. This implies, that

$$\frac{G_B}{1 - G_B} = \frac{\sum_{i=0}^{n_a} a_i}{\sum_{i=0}^{n_b} \beta_i - \sum_{i=0}^{n_a} a_i} = \frac{\sum_{i=0}^{n_a} a_i}{\sum_{i=0}^{n_a} \beta_i - \sum_{i=0}^{n_a} a_i}$$

$$\lim_{z^{-1} \to 1} \frac{1 - G_B}{\sum_{i=0}^{n_a} \beta_i - \sum_{i=0}^{n_a} a_i} = -1. \qquad (2)$$

This equation implies, that $\sum_{i=0}^{n_a} \beta_i = 0$. Any bias update term must fulfil

this equation to provide good tracking. Furthermore, another condition was stated to minimise the effort to find the roots. It is

$$a_k - \beta_k \quad 0 \quad \exists\ k \text{ such that } 0 < k < max(n_a, n_b),$$

$$a_l - \beta_l = 0 \qquad \forall\ l \neq k. \qquad (3)$$

Using the results from equation (2) and (3), a bias update term stabilising

a system with time delay is given. It is

$$G_B = \frac{z^{-d}}{1 - z^{-d}}.$$

Another bias update term is proposed that provides good tracking for multirate sampling times. It is

$$G_B = \frac{1}{1 - z^{-N}}.$$

Both the soft sensor with time delay and the soft sensor with multirate sampling time were tested in simulation. The simulation shows good performance of both soft sensors. This performance is evaluated mathematically. Proof is given that the two soft sensors perform well in their respective case. The proof for multirate sampling times provided in the paper was extended and explained in detail in section 3. Regarding the proof of a case with a time delay and no multirate sampling time, Shardt and Huang (2012a) stated that

$$\begin{aligned}
\lim_{z^{-1}\to 1}(\hat{y}_t) &= \lim_{z^{-1}\to 1}\left[\frac{\hat{G}_p - G_B G_p}{1 - G_B}u_t + \frac{-G_B}{1 - G_B}e_t\right] \\
&= \lim_{z^{-1}\to 1}\left[\frac{\hat{G}_p}{1 - G_B}u_t\right] + \lim_{z^{-1}\to 1}\left[\frac{-G_B}{1 - G_B}G_p u_t\right] + \lim_{z^{-1}\to 1}\left[\frac{-G_B}{1 - G_B}G_l e_t\right] \\
&= 0 + \lim_{z^{-1}\to 1}(G_p u_t + G_l e_t) \\
&= \lim_{z^{-1}\to 1}(y_t).
\end{aligned}$$

Stability of the estimate can be shown by evaluating the roots.

$$1 - G_B = 1 - \frac{z^{-d}}{1 - z^{-d}} = \frac{1 - z^{-d}}{1 - z^{-d}} = 1$$

Regarding the case of time delay and multirate sampling time, it was stated, that for a delay d and a sampling of the bias update term N, several cases must be considered. For the case that $d \leq N$, the transfer function for

the multirate case

$$G_B = \frac{1}{1 - z^{-N}}$$

is used. Proof that this choice of bias update term is adequate is given in section 3. Performance of such a soft sensor is shown in section 4.2.

For the case that $d > N$, no general solution is given. However, it is proposed that if the delay is a multiple of the sample time, the bias update term of the delay

$$G_B = \frac{z^{-d}}{1 - z^{-d}}$$

is used.

2.3.4 Bias Update Termfor Parameter Variation

This thesis does consider the presence of a time delay and multirate sampling times. Furthermore, it is assumed that both the sampling time and the time delay are variable in time. The time delay is assumed to be in the range $d \in [d_a, d_\Omega]$. The sampling time is assumed to be in the range $N \in [N_a, N_\Omega]$, where the range is referred to as $\mathsf{S} := [N_a, N_\Omega]$. Note that everything that holds for variable sampling time and delay does hold for constant sampling time and delay since the upper and lower bound can be chosen equal.

It is assumed that $d_a \leq d_\Omega < N_a \leq N_\Omega$, so the sampling time is always bigger than the time delay. It holds $qN_a < N_\Omega \leq (q + 1)N_a$, where q is an integer value.

A linear transfer function is used as $\hat{G}_B$. If N is constant, the transfer function is

$$\hat{G}_B = -\frac{1}{1 - z^{-N}}. \tag{4}$$

If N is variable, the transfer function is

$$\hat{G}_B = -\frac{1}{|\mathsf{S}| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i}} \tag{5}$$

where $|\mathsf{S}|$ is the cardinality of the set S including all sampling times.

3 Proof of Adequacy of Bias Update Term

To show the influence of G_B in the presence of time delay and multirate sampling times, the behaviour of $\hat{y}_t$ is shown at a time step where a measurement is taken. In this step the equation

$$\hat{y}_t = \hat{G}_p u_t + G_B(\hat{y}_t - y_t) \tag{6}$$

holds. Since G_B includes a time delay, the values $(\hat{y}_t - y_t)$ are delayed by d steps. Thus, the equation is

$$\hat{y}_t = \hat{G}_p u_t + \hat{G}_B(\hat{y}_{t-d} - y_{t-d}).$$

Note that $\hat{y}_t = y_{a,t} + y_{\beta,t}$ and the value $y_{\beta,t}$ is part of the feedback loop. Hence, it is sampled with a zero order hold. Since $d < N$ at any point of time, $y_{\beta,t-d} = y_{\beta,t-N}$. So the equation (6) is

$$y_{a,t} + y_{\beta,t} = \hat{G}_p u_t + \hat{G}_B(y_{a,t-d} + y_{\beta,t-N} - y_{t-d}).$$

This equation is re-organised. First all terms including $y_{\beta,t}$ are put to one side of the equation and all other terms to the other side.

$$y_{\beta,t} - \hat{G}_B y_{\beta,t-N} = \hat{G}_p u_t + \hat{G}_B(y_{a,t} - y_{t-d}) - y_{a,t}$$

Now, the delay of some variables is not described in the index, but with a delay term.

$$y_{\beta,t} - \hat{G}_B z^{-N} y_{\beta,t} = \hat{G}_p u_t - \hat{G}_B y_{t-d} + \hat{G}_B z^{-d} y_{a,t} - y_{a,t}$$

The equation is re-organised:

$$(1 - \hat{G}_B z^{-N}) y_{\beta,t} = \hat{G}_p u_t - \hat{G}_B y_{t-d} + (\hat{G}_B z^{-d} - 1) y_{a,t}$$

Both sides are divided by $(1 - \hat{G}_B z^{-N})$.

$$y_{\beta,t} = \frac{\hat{G}_p}{1 - \hat{G}_{B_{z^{-N}}}} u_t - \frac{\hat{G}_B}{1 - \hat{G}^B z^{-N}} y_{t-d} + \frac{\hat{G}_B z^{-d} - 1}{1 - \hat{G}^B z^{-N}} y^{a,t}$$

Due to the system equation from Figure 1, $y_{t-d} = G_p u_{t-d} + G_l e_{t-d}$. This is used to further re-organise equation (6).

$$y_{\beta,t} = \frac{\hat{G}_p}{1 - \hat{G}_{B_{z^{-N}}}} u_t - \frac{\hat{G}_B}{1 - \hat{G}^B z^{-N}} y_{t-d} + \frac{\hat{G}_B z^{-d} - 1}{1 - \hat{G}^B z^{-N}} y^{a,t}$$

Since $y_t = G_p u_t + G_l e_t$, this implies

$$y_{\beta,t} = \frac{\hat{G}_p}{1 - \hat{G}_{B_{z^{-N}}}} u_t - \frac{\hat{G}_B G_p}{1 - \hat{G}^B z^{-N}} u_{t-d} + \frac{\hat{G}_B G_l}{1 - \hat{G}^B z^{-N}} e_{t-d} + \frac{\hat{G}_B z^{-d} - 1}{1 - \hat{G}^B z^{-N}} y^{a,t}.$$

With $u_{t-d} = z^{-d} u_t$ and $e_{t-d} = e_t z^{-d}$, the term can be simplified:

$$y_{\beta,t} = \frac{\hat{G}_p - \hat{G}_B G_p z^{-d}}{1 - \hat{G}_{B_{z^{-N}}}} u_t - \frac{\hat{G}_B G_l z^{-d}}{1 - \hat{G}^B z^{-N}} e_t + \frac{\hat{G}_B z^{-d} - 1}{1 - \hat{G}_B z^{-N}} y_{a,t} \qquad (7)$$

This equation is now used to evaluate what choice of $\hat{G}_B$ is adequate. The bias update term must ensure that the estimate converges to the correct value. This is shown by demonstrating that for $t \to \infty$ the estimation is exact and by showing that the estimation has a stable behaviour.

3.1 Behaviour for $t \to \infty$

To ensure good tracking of a soft sensor, the behaviour of the estimation for $t \to \infty$ is considered. It is shown that the bias update term for both the case of constant sampling time (4) and the case of variable sampling time (5) are chosen in a way that $\lim_{t\to\infty} \hat{y}_t = \lim_{t\to\infty} y_t$. For a constant input, the output of the bias update term converges to this value and remains stationary if it reaches the value. For a stationary value, the past outputs are equal to the present output and all past inputs are assumed equal to the present input. Thus, the equation $y_{k-1} = y_k$ and $u_{k-1} = u_k$ hold for any k in the stationary

state. Therefore, it is assumed that $z^{-1} = 1$ and $z^{-k} = 1$ for all integer values in k. Thus, the limit the transfer function converges to is determined by $\lim_{z^{-1}\to 1}(G(z^{-1}))$.

Lemma 3.1

For the bias update term G_B taken from (4) two equations hold:

$$\frac{\hat{G}_B z^{-d}}{1 - \hat{G}_B z^{-N}} = -1$$

$$\frac{1}{1 - \hat{G}_B z^{-N}} = 0$$

Proof :

To prove this, the bias update term

$$\frac{1}{1 - z^{-N}}$$

is inserted into the equation. Then, the equation is simplified, leading to the desired result.

$$\lim_{z^{-1}\to 1} \left(\frac{\hat{t}_B z^{-d}}{1 - \hat{t}_B z^{-N}} \right) = \lim_{z^{-1}\to 1} \left(\frac{\frac{z^{-d}}{1 - z^{-N}}}{1 + \frac{z^{-N}}{1 - z^{-N}}} \right)$$

$$= \lim_{z^{-1}\to 1} \left(\frac{-z^{-d}}{1 - z^{-N} + z^{-N}} \right) = \lim_{z^{-1}\to 1} \left(-z^{-d} \right) = -1$$

The same thing is done to show the second equation.

$$\lim_{z^{-1}\to 1} \left(\frac{1}{1 - \hat{t}_B z^{-N}} \right) = \lim_{z^{-1}\to 1} \left(\frac{1}{1 + \frac{z^{-N}}{1 - z^{-N}}} \right)$$

$$= \lim_{z^{-1}\to 1} \left(\frac{1 - z^{-N}}{1 - z^{-N} + z^{-N}} \right) = \lim_{z^{-1}\to 1} \left(1 - z^{-N} \right) = 0$$

■

Lemma 3.2

For the bias update term G_B taken from (5) two equations hold:

$$\frac{\hat{G}_B z^{-d}}{1 - \hat{G}_B z^{-N}} = -1$$

$$\frac{1}{1 - \hat{G}_B z^{-N}} = 0$$

Proof :

To prove this, the bias update term

$$\hat{tt}_B = -\frac{1}{|S| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i}}$$

is inserted into the equation. Then, the equation is simplified, leading to the desired result.

$$\lim_{z^{-1}\to 1} \left(\frac{\hat{t}_B z^{-d}}{1 - \hat{t}_B z^{-N}} \right) = \lim_{z^{-1}\to 1} \left(\frac{-\frac{z^{-d}}{|S| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i}}}{1 - \left(-\frac{1}{|S| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i}}\right) z^{-N}} \right)$$

$$= \lim_{z\to 1} \left(\frac{-z^{-d}}{|S| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i} + z^{-N}} \right) = \frac{-1}{|S| - |S| + 1} = -1$$

The same thing is done to show the second equation.

$$\lim_{z^{-1}\to 1} \left(\frac{1}{1 - \hat{tt}_B z^{-N}} \right) = \lim_{z^{-1}\to 1} \left(\frac{1}{1 + \frac{1}{|S| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i}} z^{-N}} \right)$$

$$= \lim_{z^{-1}\to 1} \left(\frac{|S| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i}}{|S| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i} + z^{-N}} \right) = \frac{0}{|S| - |S| + 1} = 0 \quad \blacksquare$$

Theorem 3.1

For both bias update term proposed for the case of constant time delay given in (4) and the bias update term proposed for a variable time delay (5), holds:

$$\lim_{t\to\infty} \hat{y}_t = \lim_{t\to\infty} y_t$$

Proof :

The equation (7) can be converted to the desired form using Lemmata 3.1 and 3.2, if $t \to \infty$ is assumed.

$$y_{\beta,t} = \frac{}{1 - \hat{t}_{B\,Z^{-N}}} u_t - \frac{}{1 - \hat{t}^{BZ\ \ N}} \frac{}{1 - \hat{t}_B}$$

The equations are re-organised.

$$y_{\beta,t} = \frac{1}{1 - \hat{t}_{BZ^{-N}}} \hat{tt}_p u_t - \frac{\hat{t}_{BZ^{-d}}}{1 - tt_{B\,Z^{-N}}} \hat{tt}_p u_t - \frac{\hat{t}_{BZ^{-d}}}{1 - tt^{BZ^{-N}}} tt_l e_t + \frac{\hat{t}_{BZ^{-d}}}{1 - \hat{tt}_{BZ^{-N}}} y_{a,t} - \frac{1}{1 - \hat{t}_{\ B^{Z^{-N}}}} y_{a,t}$$

Now Lemma 3.1 and Lemma 3.2 are used. Since for $t \to \infty$

$$\frac{\hat{t}_{BZ^{-d}}}{1 - \hat{t}_{B\,Z^{-N}}} = -1,$$
$$\frac{1}{1 - \hat{t}^{B\,Z^{-N}}} = 0$$

the equation above can be simplified:

$$y_{\beta,t} = 0\hat{t}_p u_t + \frac{}{\hat{t}}\ {}_p u_t + tt_l e_t - y_{a,t} - 0y_{a,t}$$

If $y_{a,t}$ is added to both sides of the equation, the equation simplifies, yield- ing the sought result.

$$y_{a,t} + y_{\beta,t} = tt_p u_t + tt_l e_t = \hat{y}_t = y_t$$

■

To show the adequacy of the bias update term, theorem 3.1 was proven. It was shown that if the estimation behaves asymptotically stable, it will converge to the true value without bias or slope. The stability of the estimation must still be proven to show that the soft sensor provides good tracking.

3.2 Stability

To ensure stability, the stability of the transfer function $\frac{\hat{t}_B z^{-d}}{1-\hat{t}_B z^{-N}}$ and $\frac{1}{1-\hat{t}_B z^{-N}}$ must be shown. Assuming stability of $G_p u_t$, $\hat{G}_p u_t$, $G_l e_t$ and $y_{a,t}$, stability of $\frac{\hat{t}_B z^{-d}}{1-\hat{t}_B z^{-N}}$ and $\frac{1}{1-\hat{t}_B z^{-N}}$ implies stability of $y_{\beta,t}$ and hence, stability of the estimation $\hat{y}_t$.

To show that the estimate is stable, proof is given that $\frac{\hat{t}_B z^{-d}}{1-\hat{t}_B z^{-N}}$ and $\frac{1}{1-\hat{t}_B z^{-N}}$ are stable. This is sufficient to guarantee stability of the estimation for a system with multirate sampling time and time delay. Two separate proofs are given for the case of a constant sampling time and the case of a varying sampling time. The time delay is assumed variable for both proofs. Stability of $\hat{y}_t$ is shown using equation (7).

3.2.1 Stability for Constant Sampling Time

To show stability for a constant sampling time, the roots of the two transfer functions $\frac{\hat{t}_B z^{-d}}{1-\hat{t}_B z^{-N}}$ and $\frac{1}{1-\hat{t}_B z^{-N}}$ must be shown to be stable. Since both transfer functions share the same roots, it is sufficient to show that $\frac{\hat{t}_B z^{-d}}{1-\hat{t}_B z^{-N}}$ only has stable roots. With $\hat{G}_B$ given in equation (4), the roots of $\hat{G}_B$ are determined. Since

$$\frac{\hat{G}_B z^{-d}}{1-\hat{G}_B z^{-N}} = \frac{-\frac{z^{-d}}{1-z^{-N}}}{1+\frac{z^{-N}}{1-z^{-N}}} = \frac{-z^{-d}}{1-z^{-N}+z^{-N}} = -z^{-d}$$

$\hat{G}_B$ has no unstable roots. Therefore, the soft sensor provides good tracking.

3.2.2 Stability for Variable Sampling Time

The variable sampling times convert the linear system to a switched linear system. Since it is not sufficient to show that the estimation stays stable for

each fixed sampling time $N \in \mathsf{S}$, analysing the roots of the transfer functions is not sufficient (Lin and Antsaklis, 2009). To show the asymptotic stability of the switched system, a theorem from Bauer et al. (1993) is used.

Theorem 3.2 (*Bauer-Premaratne-Durán Theorem*)

A switched linear system $x_{k+1} = A_{\sigma(k)}x_k$, where $A_{\sigma(k)} \in \{A_1, A_2, \ldots, A_M\}$, is asymptotically stable under arbitrary switching if and only if there exists a finite integer n such that

$$||A_{i1}A_{i2} \ldots A_{in}||_\infty < 1$$

for all n-tuple $A_{ij} \in \{A_1, A_2, \ldots, A_M\}$, where $j = 1, \ldots, n$.

The norm used by the Bauer-Premaratne-Durán theorem is the maximum norm. It is the biggest absolute value of the matrix entries. Proof of the theorem is given in Bauer et al. (1993).

To verify whether the Bauer-Premaratne-Durán theorem applies to the system, a state space realisation of the transfer functions $\frac{\hat{t}_B z^{-d}}{1-\hat{t}_B z^{-N}}$ and $\frac{1}{1-\hat{t}_B z^{-N}}$ must be found for every combination of the current sampling time N and sampling times for the past time steps. Since the state space realisation of the two transfer functions is the same, it is sufficient to analyse $\frac{\hat{t}_B z^{-d}}{1-\hat{t}_B z^{-N}}$. The sampling time at the current sampling step is N_0. The sampling time of the last sampling step is N_1, the sampling time before that N_2 and the prior sampling times are described analogously, up to N_q. The bias update term is given as

$$\hat{G}_B = -\frac{1}{|\mathsf{S}| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i}}$$

where $|\mathsf{S}|$ is the cardinality of the set S including all sampling times. Therefore, the matrix A is built by analysing

$$\frac{\hat{G}_B z^{-d}}{1 - \hat{G}_B z^{-N_0}} = -\frac{z^{-d}}{|\mathsf{S}| - \sum_{i=N_\alpha}^{N_\Omega} z^{-i} + z^{-N_0}} = \frac{y_B}{u_B}.$$

Both sides are multiplied with the denominators. The sum is divided into

the sum before the subtracted value and after the subtracted value z^{-N_0}.

$$|S| - \sum_{i=N_a}^{N_0-1} z^{-i} - \sum_{i=N_0+1}^{N_\Omega} z^{-i}\, y_B = -z^{-d} u_B$$

Now, the term including the current output of the bias update term is brought to one side of the equation and all other terms to the other side.

$$|S|\, y_B = \sum_{i=N_a}^{N_0-1} z^{-i} y_B + \sum_{i=N_0+1}^{N_\Omega} z^{-i} y_B - z^{-d} u_B$$

Both sides of the equation are divided by $|S|$.

$$y_B = \sum_{i=N_a}^{N_0-1} \frac{z^{-i}}{|S|} y_B + \sum_{i=N_0+1}^{N_\Omega} \frac{z^{-i}}{|S|} y_B - z^{-d} u_B \tag{8}$$

All values considered by the bias update term were taken at the discrete sampling points. For example, the values between the last measurement N_0 steps in the past and the current time step are held constant and are $z^{-i} y_B = z^{-N_0} y_B$ for all values $i \in [N_a, N_0]$. The variance of the sampling time is assumed high. The range of sampling times is defined as $qN_a < N_\Omega \leq (q + 1)N_a$, where q is an integer value.

Therefore, the considered measurements can come from several time steps in the past. For $i \in [N_0 + 1, N_\Omega]$, it is possible that $\sum_{j=0}^{r-1} N_j < i \leq \sum_{j=0}^{r} N_j$, with $r \in [0, q]$. Hence, the measurement considered by the bias update term can come from up to $q + 1$ different measurements steps in the past. Note that $z^{-i} y_B = z^{-\sum_{j=0}^{r} N_j} y_B$ due to sampling.

This is used to re-organise equation (8). This leads to

$$
\begin{aligned}
y_B &= \sum_{i=N_a}^{N_1} \frac{z^{-i}}{|S|} y_B + \sum_{i=N+1}^{N_\Omega} \frac{z^{-i}}{|S|} y_B - z^{-d} u_B \\
&= (N_0 - N_a) \frac{z^{-N_0}}{|S|} y_B + \sum_{i=1}^{r-1} N_i \frac{\bar{z}^{\sum_{j=0}^{i} N_j}}{|S|} y_B + \\
&\quad + (N_\Omega - \sum_{j=0}^{r-1} N_j) \frac{\bar{z}^{\sum_{j=0}^{r} N_j}}{|S|} y_B - z^{-d} u_B
\end{aligned}
$$

where r is the biggest integer value with $\sum_{j=0}^{r-1} N_j < N_\Omega \leq \sum_{j=0}^{r} N_j$. Defining the state of the state space as $\dot{x}^T = (y_B \; y_B z^{-1} \; \dots \; y_B z^{-(2N_\Omega - 1)})$, a state space realisation is given as

$$
x_{m+1} = A\dot{x}_m + Bu_m
$$

where A is the state space matrix and $B = -(0 \cdots 0\,1\,0 \cdots 0)^T$ the input matrix, with the value equal 1 in the d^{th} row. The output is irrelevant for stability, so C and D are not given. $A \in \mathbb{R}^{(2N_\Omega - 1) \times (2N_\Omega - 1)}$ is constructed as

$$
A = \left[\begin{array}{ccccccc|c}
0^{N_0-1\times 1} & \frac{\overline{N_0 - N_a}}{|S|} & 0^{N_1-1\times 1} & \frac{\overline{N_1}}{|S|} & \cdots & \frac{\overline{N_{r-1}}}{|S|} & 0^{N_{r-1}-1\times 1} & \frac{\overline{N_\Omega - \sum_{j=0}^{r-1} N_j}}{|S|} & 0 \dots 0 \\
\hline
1 & 0 & & & \cdots & & & 0 & 0 \\
 & & & & \ddots & & & \vdots & \vdots \\
0 & 1 & & & & & & \vdots & \vdots \\
0 & & & & \cdots & & & 1 & 0
\end{array}\right]
\tag{9}
$$

To apply the Bauer-Premaratne-Durán theorem to the system of interest,

the matrices A are considered as the sum of two matrices, being

$$A = \Lambda_i + M_0 = \begin{pmatrix} 0^{N_0-1\times 1} & \frac{N_0-N_\alpha}{|S|} & 0^{N_1-1\times 1} & \frac{N_1}{|S|} & \cdots & \frac{N_{r-1}}{|S|} & 0^{N_{r-1}-1\times 1} & \frac{N_\Omega-\sum_{j=0}^{r-1} N_j}{|S|} & 0 \cdots 0 & 0 \\ \hline 0 & & & & \cdots & & & & 0 & 0 \\ \vdots & & & & \ddots & & & & \vdots & \vdots \\ 0 & & & & \cdots & & & & 0 & 0 \end{pmatrix}$$

$$+ \begin{pmatrix} 0 & & \cdots & & 0 \\ \hline 1 & 0 & \cdots & 0 & 0 \\ 0 & & & & \\ \vdots & & \ddots & \vdots & \vdots \\ & & & 0 & \\ 0 & & \cdots \ 0 & 1 & 0 \end{pmatrix}.$$

Note that Λ_i only has entries in the first row which are all positive and add up to less than one. M_0 is the subdiagonal shifting matrix. A property of the matrix Λ_i is shown in Lemma 3.3.

Lemma 3.3

Given Λ_i as above. For any matrix $X \in \mathbb{R}^{(2N_\Omega-1)\times(2N_\Omega-1)}$ holds the inequation

$$||\Lambda_i X||_\infty < ||X||_\infty.$$

Proof :

The Lemma is proven by converting a general form of $||\Lambda_i X||_\infty$ to a form that obviously is smaller than $||X||_\infty$ using upwards estimations.

First, the multiplication of the matrices Λ_i and X is done. The entries of X are $x_{i,j}$, where i is the number of the row of the entry and j the

number of the column.

$$\Lambda_i X = \begin{bmatrix} \frac{N_0 - N_a}{|S|} x_{N_0,j} + \sum_{i=1}^{r-1} \frac{N_i}{|S|} x_{(\sum_{k=0}^{i} N_k),j} + \frac{(N_\Omega - \sum_{j=0}^{r-1} N_j)}{|S|} x_{(\sum_{k=0}^{r-1} N(k)),j} & \cdots \\ \vdots & \\ 0 & \cdots \\ \vdots & \ddots \\ 0 & \cdots \end{bmatrix}$$

Now, the norm of this matrix is calculated. It is the maximum of the absolute value of all entries. Since all rows except the first are zero,

$$||\Lambda_i X||_\infty = max_j(|\frac{N_0 - N_a}{|S|} x_{N_0,j} + \sum_{i=1}^{r-1} \frac{N_i}{|S|} x_{(\sum_{k=0}^{i} N_k),j} + \frac{(N_\Omega - \sum_{j=0}^{r-1} N_j)}{|S|} x_{(\sum_{k=0}^{r-1} N(k)),j}|).$$

By splitting the equation, an estimation upward is made.

$$||\Lambda_i X||_\infty \leq max_j(|\frac{N_0 - N_a}{|S|} x_{N_0,j}|) + max_j(|\sum_{i=1}^{r-1} \frac{N_i}{|S|} x_{(\sum_{k=0}^{i} N_k),j}|) + max_j(|\frac{(N_\Omega - \sum_{j=0}^{r-1} N_j)}{|S|} x_{(\sum_{k=0}^{r-1} N(k)),j}|)$$

The constants are all bigger than zero and are not affected by the maximum function.

$$||\Lambda_i X||_\infty \leq \frac{N_0 - N_a}{|S|} max_j(|x_{N_0,j}|) + \sum_{i=1}^{r-1} \frac{N_i}{|S|} max_j(|x_{(\sum_{k=0}^{i} N_k),j}|) + \frac{(N_\Omega - \sum_{j=0}^{r-1} N_j)}{|S|} max_j(|x_{(\sum_{k=0}^{r-1} N(k)),j}|)$$

Since all entries in X are smaller than its maximum norm, an estimation upward for the maximum terms can be made.

$$||\Lambda_i X||_\infty \leq \left(\frac{N_0 - N_a}{|S|} + \sum_{i=1}^{r-1} \frac{N_i}{|S|} + \frac{(N_\Omega - \sum_{j=0}^{r-1} N_j)}{|S|} \right) ||X||_\infty$$

The sum of the constant is calculated.

$$\frac{N_0 - N}{|S|}\, a + \sum_{i=1}^{N_\Omega - N} \frac{N_i}{|S|} + \frac{(N_\Omega - \sum_{j=0}^{r-1} N_j)}{|S|} = \frac{N - N}{|S|}\, a_\Omega = \frac{|S| - 1}{|S|}$$

Since $\frac{|S|-1}{|S|} < 1$, the Lemma is proven.

$$||\Lambda_i X||_\infty \leq \frac{|S| - 1}{|S|} ||X||_\infty < ||X||_\infty$$

■

Now the stability of the switched system is shown.

Theorem 3.3

The switched system with the sampling time

$$N \in S,\ d_a \leq d_\Omega < N_a \leq N_\Omega < (q+1)N_a$$

$$\hat{G}_B = -\frac{1}{|S| - \sum_{i=N_a}^{N_\Omega N_a} z^{-i}}$$

is asymptotically stable for arbitrary switching N.

Proof :

Asymptotic stability of the system is shown using theorem 3.2. To apply it, it is shown by induction and application of Lemma 3.3, that the considered system has state matrices that have the property $||\prod_{i=1}^{2N_\Omega - 1} A_{ij}||_\infty < 1$ for any choice of A_{ij}.

Given $A_{ij}, j = 1, \ldots, n$ being n matrices in the form described as in equation (9). Choose $n = 2N_\Omega - 1$. It is shown by induction that the first k lines of the matrix product $\prod_{i=1}^{k} A_{ij}$ only consist of entries < 1 and all other values are ≤ 1.

$k = 1$: trivial

$k > 1$:

$$\prod_{i=1}^{k} A_{ik} = A_{i1} \prod_{i=2}^{k} A_{ik}$$

Since the matrices are arbitrary, new indices can be chosen.

$$A_{i0} \prod_{i=1}^{k-1} A_{ik} = (M_0 + \Lambda_0) \prod_{i=1}^{k-1} A_{ik} = M_0 \prod_{i=1}^{k-1} A_{ik} + \Lambda_0 \prod_{i=1}^{k-1} A_{ik}.$$

Any matrix multiplied with the shifting matrix from the left is shifted one row below. Thus, due to the induction hypothesis, $\prod_{i=1}^{k-1} M_0 A_{ik}$ has only values < 1 in the first k lines and values ≤ 1 in the other lines. Due to Lemma 3.3,

$$||\Lambda_0 \prod_{i=1}^{k-1} A_{ik}||_\infty < || \prod_{i=1}^{k-1} A_{ik}||_\infty \leq 1$$

$$\Rightarrow ||\Lambda_0 \prod_{i=1}^{k-1} A_{ik}||_\infty < 1.$$

Note that Λ_0 only has entries in the first row and M_0 no entries in the first row. Hence, the induction hypothesis holds.

By choosing $k = n$, it is shown that $|| \prod_{i=1}^{k} A_{ik}||_\infty < 1$. Thus, the Bauer-Premaratne-Durán theorem 3.2 can be applied and the system is asymptotically stable for arbitrary switching. ■

4 Application

The method of the latter chapter is applied to a simulation of a system. The simulation is done under various conditions.

4.1 Simulation Setup

The system to which the soft sensor described in section 2 is applied to is called the **c**ontinuous **s**tirred **t**ank **r**eactor (CSTR), proposed in Morningred et al. (1990). It is used as a benchmark in several publications (Huang et al., 2000; Zhang et al., 2016; Shardt and Huang, 2012a,b). The CSTR is a nonlinear system, modelling an irreversible, exothermic reaction, where the substrate A reacts to the product B. The reactor is cooled by a single coolant stream and modelled by the differential equations

$$\dot{C}_A = \frac{\dot{V}}{V}(C_{A0} - C_A) - k_0 C_A \exp\left(-\frac{E}{RT}\right)$$

$$\dot{T} = \frac{\dot{V}}{V}(T_0 - T) - \frac{(-\Delta H) k_0 C_A}{\rho c_p} \exp\left(-\frac{E}{RT}\right) + \frac{\rho_c c_{pc} \dot{V}_c}{\rho c_p V} \left(1 - \exp\left(-\frac{hA}{\dot{V}_c \rho_c c_{pc}}\right)\right)(T_{c0} - T)$$

where the nominal values are given in table 1.

The system is implemented in MATLAB/SIMULINK and simulated. The input of the system in the implementation is the coolant flow rate $\dot{V}_c$. It is chosen as a step function in simulation. The inlet flow rate $\dot{V}$ is considered a disturbance and implemented as coloured noise. It is created from a transfer function that is applied to white noise. The behaviour of the substrate concentration C_A of the CSTR in simulation is displayed in Figure 10. The goal of the implemented soft sensors is to estimate this variable at all points of time.

Table 1: Nominal CSTR parameter values

product concentration	C_A	0.1 mol/l
reactor temperature	T	438.54 K
coolant flow rate	$\dot{V}_c$	103.41 l/min
process flow rate	$\dot{V}$	100 l/min
feed concentration	C_{A0}	1 mol/l
feed temperature	T_0	350 K
inlet coolant temperature	T_{C0}	350 K
CSTR volume	V	100 l
heat transfer term	hA	7×10^5 cal/min/K
reaction rate constant	k_0	$7.2 \times 10^{10}\text{min}^{-1}$
activation energy term	E/R	1×10^4 K
heat of reaction	ΔH	-2×10^5 cal/mol
liquid densities	ρ, ρ_c	1×10^3 g/l
specific heats	c_p, c_{pc}	1 cal/g/K

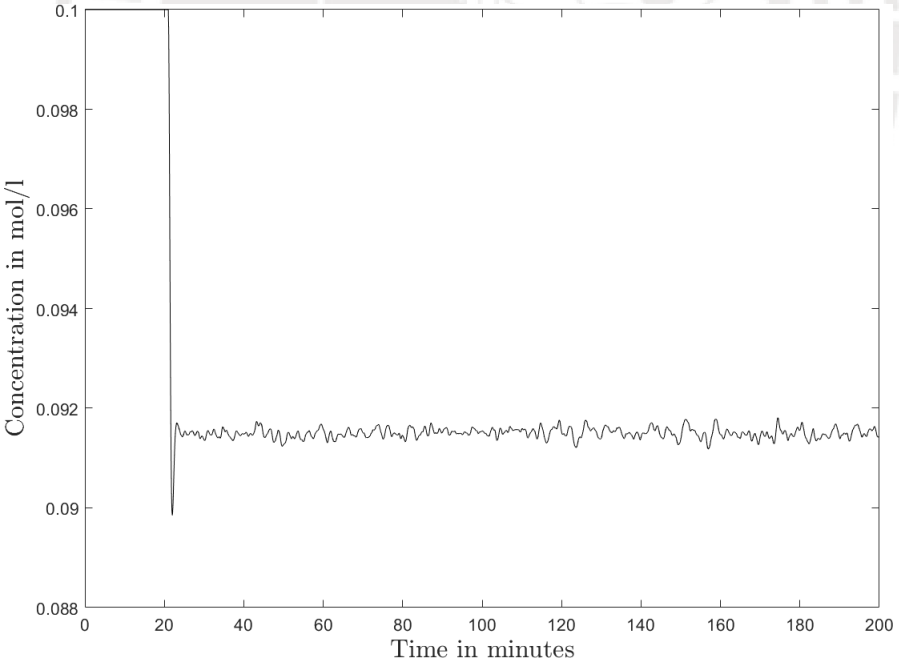

Figure 10: Substrate concentration C_A of the CSTR

4.2 Simulation

In this section, the model of the CSTR is used to test the performance of the traditional soft sensor from section 2.3.3 and its modification from section 2.3.4. They will be simulated one at a time to compare their performance once all approaches were simulated. The simulink implementation is visualised in the appendix in Figure 18.

The case of a constant time delay and a constant sampling rate of the bias update term is assumed in the traditional approach as described in section 2.3.3 and Shardt and Huang (2012a). For constant N and d, the method proposed in section 2.3.4 proposes the same bias update term. Therefore, no comparison of the methods can be done in this case. Still, the results are presented to show how the methods deal with a soft sensor with time delay and multirate sampling time. For a time delay $d = 1$ and a sampling time $N = 3$, a bias update term is used.

$$G_B = z^{-d}\hat{G}_B = z^{-1}\frac{1}{1 - z^{-3}}$$

To evaluate the accuracy of the estimation with a bias update term, three variables from the simulation are displayed. One of the variables is the measured concentration C_A of the substrate A, the value that shall be estimated. It is denoted as 'real output' and displayed as the green line. The next variable displayed is the output of the system model. It is denoted as 'simulated output' and displayed as a red line. The third and last variable displayed is the soft sensor estimate of y. It is denoted as 'estimated output' and displayed as a blue line. The parameters chosen for the soft sensor are given in the caption. This notation will be used for all graphics displayed in this section.

The results for the simulation of a soft sensor with a time delay $d = 1$ and a sampling time $N = 3$ are displayed in Figure 11.

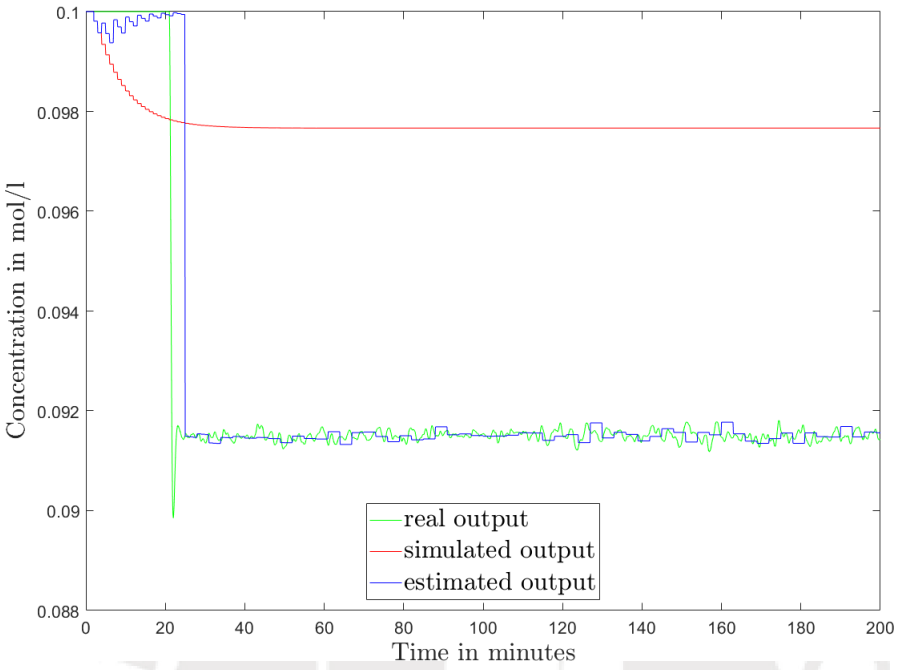

Figure 11: Soft sensor estimation of C_A for $d = 1$ and $N = 3$

During the first 20 time steps the system does not react to the input due to a time delay. The linear model does react to the input and converges to its steady state for the given input. The model output and true output do not converge to the same value due to non-linearity. However, the estimation by the soft sensor is quite accurate.

Assume the time delay $d \in [1, 2]$ and the sampling time $N = 3$. In this case, the premises of the traditional soft sensor are not fulfilled. However, the bias update term proposed by the method from section 2.3.4 is the same as for the traditional soft sensor. Due to the sampling time being higher than the time delay, the soft sensor for $d \in [1, 2]$ behaves very similarly to the case that $d = 1$. The simulation results are shown in Figure 12.

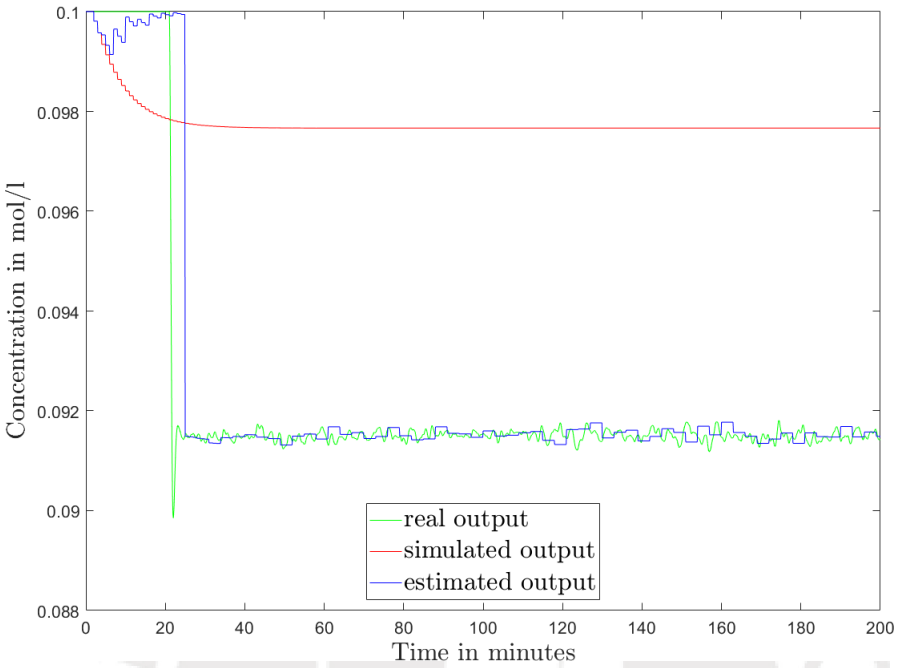

Figure 12: Soft sensor estimation of C_A for $d \in [1, 2]$ and $N = 3$

By direct comparison, one notices that the estimation in the first time steps is slightly worse for $d \in [1, 2]$ than for $d = 1$. However, the behaviour for $d \in [1, 2]$ appears to be very similar to the behaviour for $d = 1$. Looking at all simulations, the qualitative results for choosing $d = 1$ and $d \in [1, 2]$ are the same, even though the estimation is less accurate for $d \in [1, 2]$. Since in all implementations, the distinguishing features between the traditional approach and the proposed approach are more visible for $d \in [1, 2]$, only those results are shown. Assuming the time delay $d \in [1, 2]$ and the sampling time $N \in [3, 5]$, a bias update term according to the method given in section 2.3.4 is used:

$$G_B = z^{-d}\hat{G}_B = -z^{-d}\frac{1}{3 - z^{-3} - z^{-4} - z^{-5}}.$$

As shown in Figure 13, the method gives an unbiased estimate of the substrate concentration.

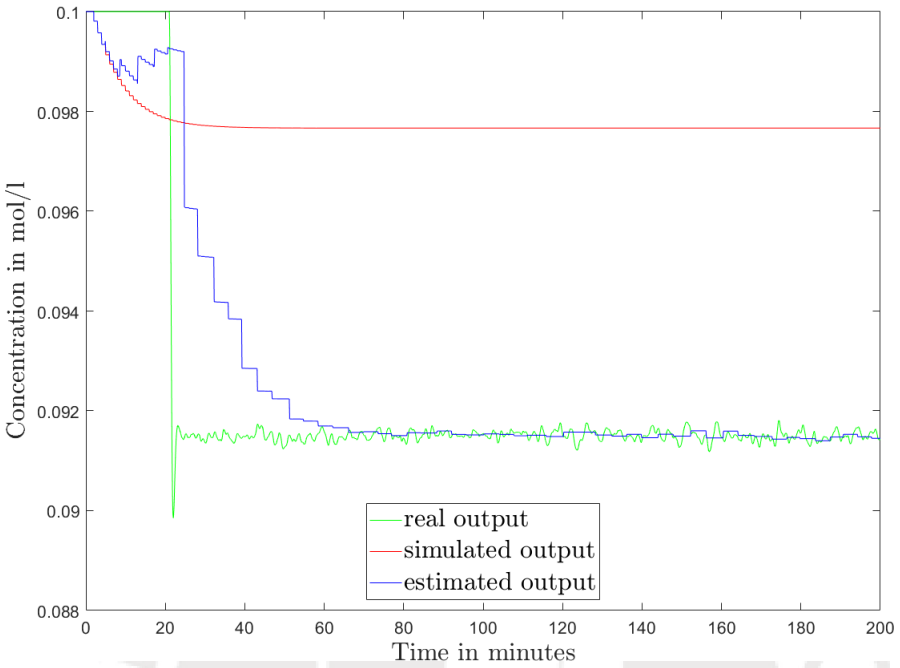

Figure 13: Soft sensor estimation of C_A for $d \in [1, 2]$ and $N \in [3, 5]$

Due to the variable sampling time and the widespread consideration of past values in $\hat{G}_B$, the estimate of the soft sensor converges to the correct value slower than in the previous cases. However, it is seen that it reaches the correct level of the output values and stays stable in its estimation.

To use the traditional soft sensor on the problem, one has to decide which sampling time to pick for the bias update term. The possible sampling times to be chosen are 3, 4 and 5.

By choosing $\hat{N} = 3$, the bias update term of the traditional approach is $\hat{G}_B = \frac{1}{1-z^{-3}}$. The simulation for a system with a delay $d \in [1, 2]$ and sampling time $N \in [3, 5]$ is shown in Figure 14.

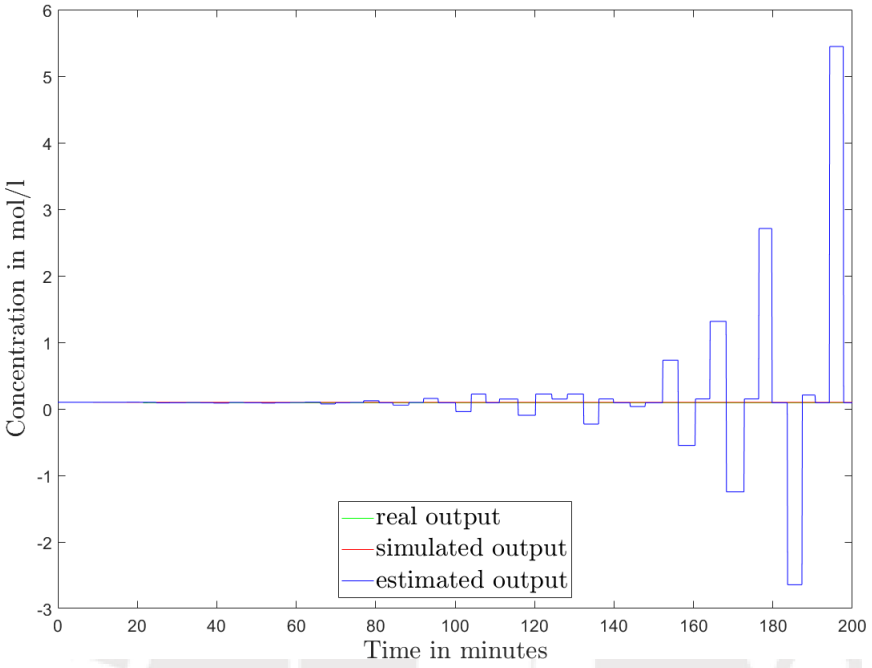

Figure 14: Soft sensor estimation of C_A for $d \in [1, 2]$ and $N = [3, 5]$, using $\hat{N} = 3$ for the soft sensor

The estimation of the traditional approach with $\hat{N} = 3$ is unstable. the value of the estimation diverges rapidly.

By choosing $\hat{N} = 4$, the bias update term of the traditional approach is $\hat{G} = \frac{1}{1-z^{-4}} B$. The simulation for a system with a delay $d \in [1, 2]$ and sampling time $N \in [3, 5]$ is shown in Figure 15 and Figure 16. Since the results are not conclusive about the stability of the estimation, one simulation with 200 steps is made and one simulation with 2000 steps.

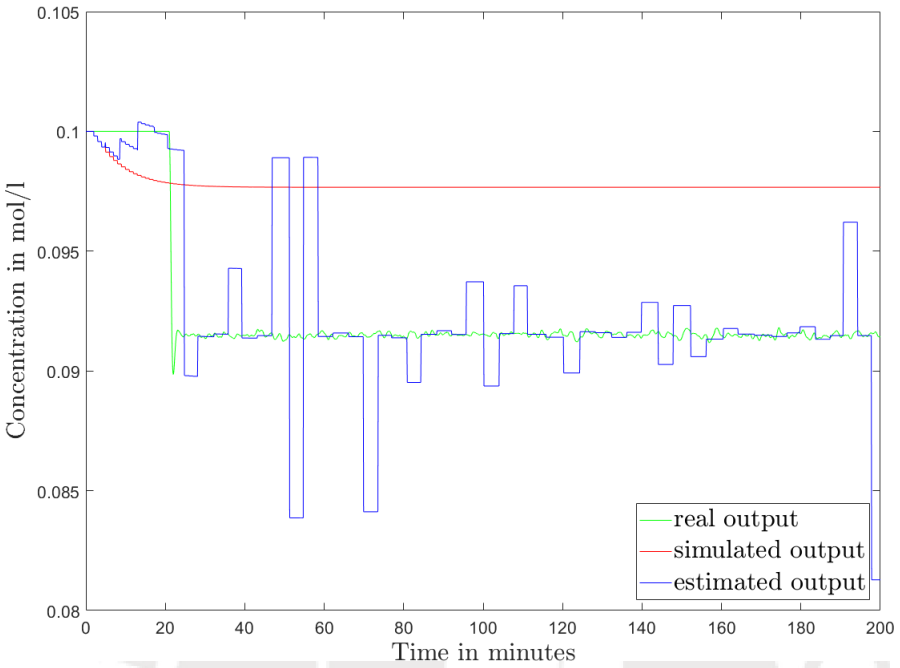

Figure 15: Soft sensor estimation of C_A for $d \in [1, 2]$ and $N = [3, 5]$, using $\hat{N} = 4$ for the soft sensor

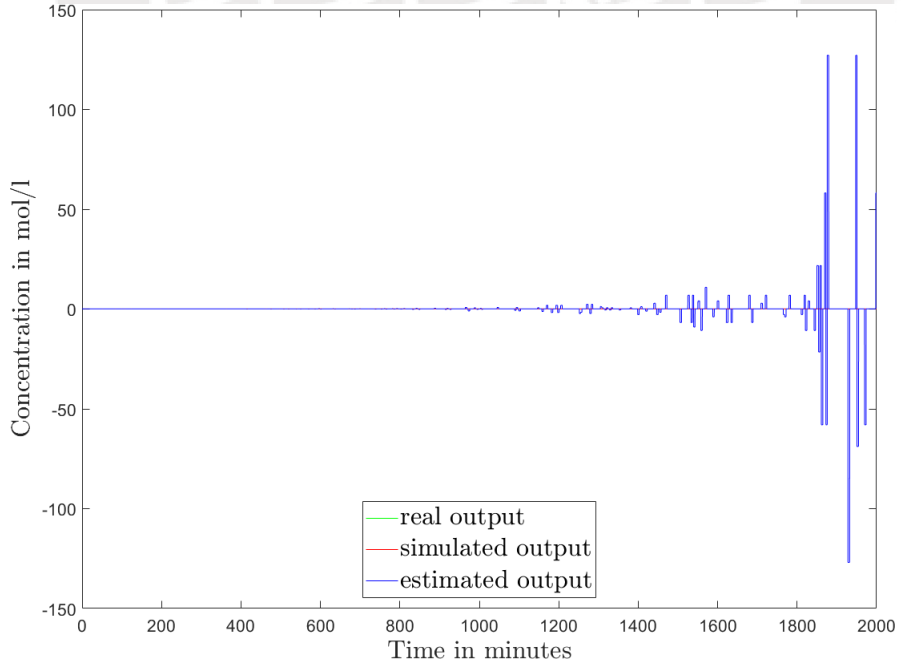

Figure 16: Soft sensor estimation of C_A for $d \in [1, 2]$ and $N = [3, 5]$, using $\hat{N} = 4$ for the soft sensor and simulating 2000 steps

The results seem better than the results for $\hat{N} = 3$ on first sight. When simulated for 200 steps, the estimation is generally close to the correct value. However, there are some peaks with a high estimation error.

Those peaks become bigger with increasing estimation time. Thus, this estimation shows to be unstable, as well. This is shown by simulating the estimation for 2000 steps instead of 200 steps.

When choosing $\hat{N} = 5$, the bias update term of the traditional approach is $\hat{G}_B = \frac{1}{1-z^{-5}}$. The simulation for a system with a delay $d \in [1, 2]$ and sampling time $N \in [3, 5]$ is shown in Figure 17.

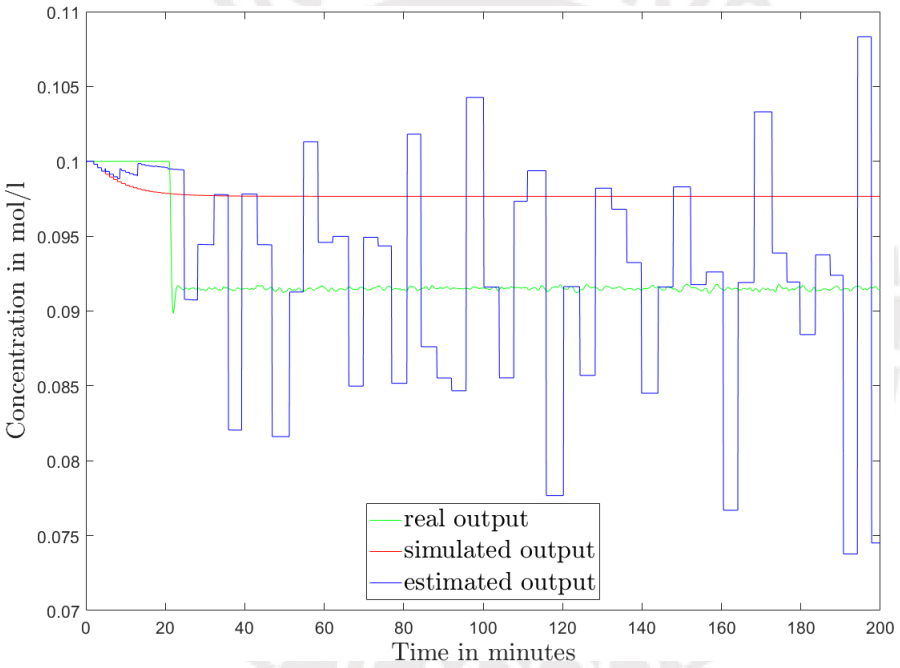

Figure 17: Soft sensor estimation of C_A for $d \in [1, 2]$ and $N = [3, 5]$, using $\hat{N} = 5$ for the soft sensor

Regarding the first 200 steps, the estimation shows strong oscillation. If only the 200 steps are considered, the instability becomes evident by comparing the magnitude of the oscillation in the first steps and in the last steps. The instability becomes more visible when looking at a simulation including more time steps.

Apparently, the traditional soft sensor does not succeed to give a stable estimate of the substrate concentration if the sampling time is variable. On the other hand, the method proposed in section 2.3.4 provides good tracking in simulation.

5 Conclusion

This thesis proposed a method for designing a soft sensor that estimates variables in the presence of variable time delay and variable multirate sampling times. The objectives of the thesis were given as

1. The soft sensor method applicable in the presence of multirate sampling and time delay presented in Shardt and Huang (2012a,b) is reviewed.
2. The method is tested in simulation.
3. An alternative approach is proposed which is designed to perform better for variable sampling times and variable time delays.
4. Mathematical proof for the adequacy of the modified method is given.
5. The convergence of the method is shown in simulation.

To objective 1: Details to the soft sensor method for multirate sampling time and time delay are given in section 2.3.3. The method considered uses a linear model of the system of interest in combination with a bias update term (Shardt and Huang, 2012a,b).

To objective 2: Section 4.2 gives results of the simulation of the soft sensor from section 2.3.3 on a nonlinear system. The soft sensor gives a stable estimation for a constant time delay and a constant sampling time. The performance hardly worsens if the time delay is assumed variable. However, for a variable sampling time the estimation of the soft sensors is unstable in simulation, regardless of the choice of time delay.

To objective 3: An alternative approach to estimate variables in the presence of variable multirate sampling time and variable time delay was proposed in section 2.3.4. The method proposed is a generalisation of the method given by Shardt and Huang (2012a,b). It uses a linear system model and a bias update term. However, the bias update term does not only consider one sampling time, but all possible sampling times in each step. The time delay is assumed smaller than the sampling time, since this situation is favoured by the literature proposing the algorithm and common in application.

To objective 4: The tracking abilities of the soft sensor are shown in section 3. It is proven that the method gives an asymptotically stable estimate converging to the true value.

To objective 5: Simulation results confirming the convergence of the modified method were shown in section 4.2. The test of the method in simulation gives stable results for both constant sampling times and variable sampling times, regardless whether the time delay is constant or variable.

Future work will consider the application of the proposed method in a closed-loop implementation. Furthermore, the restriction on the time delay can be removed, showing whether the system remains stable. Simulation on the CSTR gives reason to assume that the proposed method gives good results for closed loop implementations. In case of a relaxation of the time delay, the stability can only be observed in some cases. For a general proof of stability for arbitrary time delays, some more modifications of the algorithm must be considered.

References

Bauer, P., K.Premaratne, and J.Duran (1993). A necessary and sufficient condition for robust asymptotic stability in time-varying systems. *IEEE Trans. Automat. Contr.*, 38(9):1427–1430.

Borman, S. (2004). The expectation maximization algorithm – a short tutorial. Introduces the Expectation Maximization (EM) algorithm and fleshes out the basic mathematical results, inclusing a proof of convergence. The Generalized EM algorithm is also introduced.

Chen, Y. and Gupta, M. R. (2010). EM demystified: An expectation-maximization tutorial. Technical Report 2, University of Washington.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Ding, S. X. (2014). *Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems*. Springer.

Fortuna, L., Rizzo, A., and Xibilia, M. G. (2007). *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer.

Geladi, P. and Kowalski, B. R. (1985). Partial least-squares regression: A tutorial. *Analytica Chimica Acta*, 185:1–17.

Huang, Y., Lou, H. H., Gong, J., and Edgar, T. F. (2000). Fuzzy model predictive control. *IEEE Transactions on Fuzzy Systems*, 8(6):665–678.

Jin, X., Wang, S., Huang, B., and Forbes, F. (2012). Multiple model based lpv soft sensor development with irregular/missing process output measurement. *Control Engineering Practice*, 20(2):165–172.

Kadlec, P., Gabrys, B., and Strandt, S. (2009). Data-driven soft sensors in the process industry. *Computer and Chemical Engineering*, 33(4):795–814.

Khatibisepehr, S. and Huang, B. (2008). Dealing with irregular data in soft sensors: Bayesian method and comparative study. *Industrial & Engineering Chemistry Research*, 47(22):8713–8723.

Lin, B., Recke, B., K.H.Knudsen, J., and Jorgensen, B. B. (2006). A systematic approach for soft sensor development. *Computer and Chemical Engineering*, 31:419–425.

Lin, H. and Antsaklis, P. J. (2009). Stability and stabilizability of switched linear systems: A survey of recent results. *IEEE Transactions on Automatic Control*, 54(2):15.

Miyashita, Y., Itozawa, T., Katsumi, H., and Sasaki, S.-I. (1990). Comments on the nipals algorithm. *Journal of Chemometrics*, 4:97–100.

Montgomery, D. C. (2009). *Design and Analysis of Experiment*. John Wiley & Sons, Inc.

Montgomery, D. C., Peck, E. A., and Vining, G. G. (2012). *Introduction to Linear Regression*. John Wiley & Sons, Inc.

Morningred, J. D., Padent, B. E., Seborg, D. E., and Mellichamp, D. A. (1990). An adaptive nonlinear predictive controller. In *American Control Conference*. IEEE.

Mu Zhu, A. G. (2006). Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis*, 51:918–930.

Nelson, P. R., Taylor, P. A., and MacGregor, J. F. (1996). Missing data methods in pca and pls: Score calculations with incomplete observations. *Chenometrics and Intelligent Laboratory Systems*, 35:45–65.

North, D. W. (1968). A tutorial introduction to decision theory. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):200–210.

Shardt, Y. A. (2015). *Statistics for Chemical and Process Engineers*. Springer.

Shardt, Y. A. W. and Huang, B. (2012a). Tuning a soft sensors bias update term. 1. the open-loop case. *Industrial and Engineering Chemistry Research*, 51(2):4968–4981.

Shardt, Y. A. W. and Huang, B. (2012b). Tuning a soft sensors bias update term. 2. the closed-loop case. *Industrial and Engineering Chemistry Research*, 51(2):4958–4967.

Zhang, K., Ding, S. X., Shardt, Y. A. W., Chen, Z., and Peng, K. (2016). Assessment of t- and q-statistics for detecting additive and multiplicative faults in multivariate statistical process monitoring. *Journal of the Franklin Institute*, 354(2):668–688.

Appendix

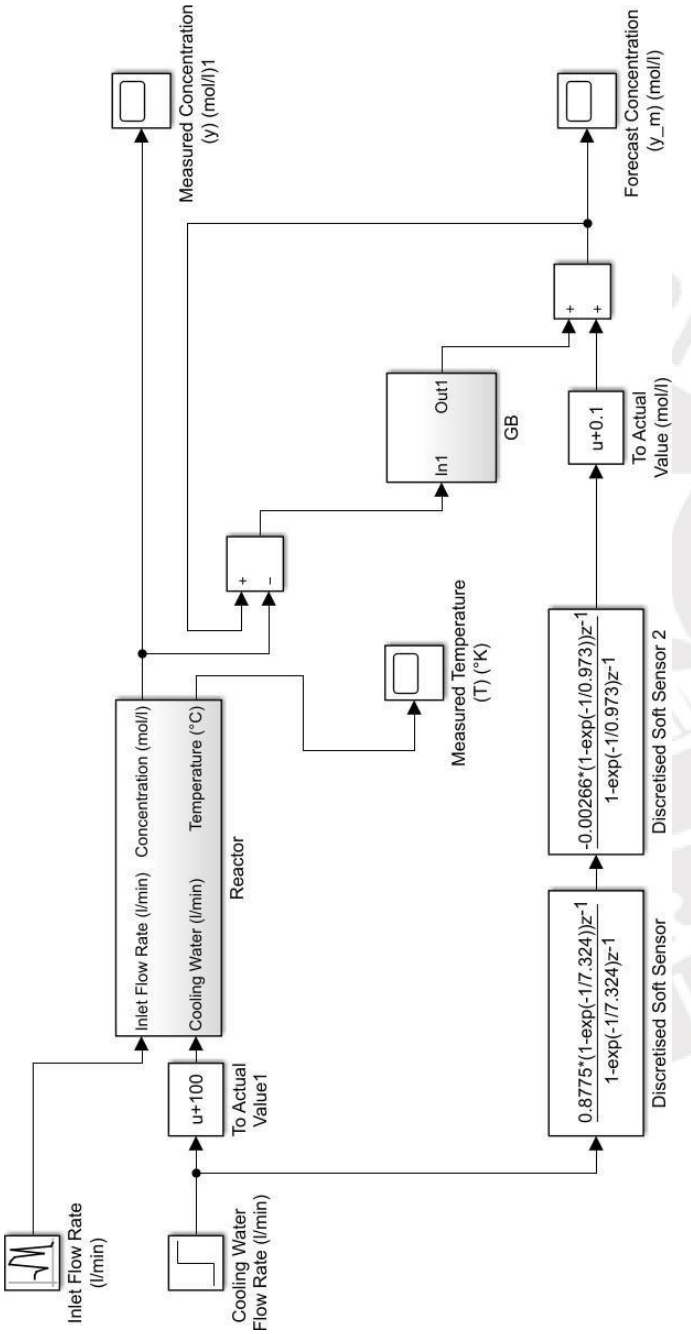

Figure 18: Simulink model